Electronic Colloquium on Computational Complexity, Report No. 188 (2020)

Hard QBFs for Merge Resolution*

Olaf Beyersdorff
Institut für Informatik, Friedrich-Schiller-Universität Jena, Jena, Germany
olaf.beyersdorff@uni-jena.de

Joshua Blinkhorn
Institut für Informatik, Friedrich-Schiller-Universität Jena, Jena, Germany
joshua.blinkhorn@uni-jena.de

Meena Mahajan
The Institute of Mathematical Sciences (HBNI), Chennai, India
meena@imsc.res.in

Tomáš Peitl
Institut für Informatik, Friedrich-Schiller-Universität Jena, Jena, Germany
tomas.peitl@uni-jena.de

Gaurav Sood
The Institute of Mathematical Sciences (HBNI), Chennai, India
gauravs@imsc.res.in

Abstract

We prove the first proof size lower bounds for the proof system Merge Resolution (MRes [6]), a refutational proof system for prenex quantified Boolean formulas (QBF) with a CNF matrix. Unlike most QBF resolution systems in the literature, proofs in MRes consist of resolution steps *together* with information on countermodels, which are syntactically stored in the proofs as merge maps. As demonstrated in [6], this makes MRes quite powerful: it has strategy extraction by design and allows short proofs for formulas which are hard for classical QBF resolution systems.

Here we show the first *exponential lower bounds for MRes*, thereby uncovering limitations of MRes. Technically, the results are either transferred from bounds from circuit complexity (for restricted versions of MRes) or directly obtained by combinatorial arguments (for full MRes). Our results imply that the MRes approach is *largely orthogonal to other QBF resolution models* such as the QCDCL resolution systems QRes and QURes and the expansion systems $\forall\text{Exp}+\text{Res}$ and IR.

2012 ACM Subject Classification Theory of computation → Proof complexity

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

Funding *Olaf Beyersdorff*: John Templeton Foundation (grant no. 60842), Carl Zeiss Foundation
Tomáš Peitl: Grant J-4361 of the Austrian Science Fund FWF.

Acknowledgements Part of this work was done during the Dagstuhl Seminar ‘SAT and Interactions’ (Seminar 20061).

1 Introduction

Proof complexity aims to provide a theoretical understanding of the ease or difficulty of proving statements formally. It also aims to explain the success stories of, as well as the obstacles faced by, algorithmic approaches to hard problems such as satisfiability (SAT) and Quantified Boolean Formulas (QBF) [19, 29]. While propositional proof complexity, the study of proofs of unsatisfiability of propositional formulas, has been around for decades [20, 27], the area of *QBF proof complexity* is relatively new, with theoretical studies gaining traction

* A preliminary version of this article appeared in the proceedings of the 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science – FSTTCS 2020 [8].

ISSN 1433-8092

only in the last decade or so [2, 7, 10, 11]. While inheriting and using a wealth of techniques from propositional proof complexity [12, 14, 25], QBF proof complexity has also given several new perspectives specific to QBF [5, 24, 35], and these perspectives and their connections to QBF solving [32, 39] as well as their practical applications [34] have driven the search for newer proof systems [1, 11, 22, 28, 30].

Many of the currently known QBF proof systems are built on the best-studied propositional proof system *resolution* [17, 33]. Broadly speaking, resolution has been adapted to handle the universal variables in QBFs in two intrinsically different ways. The first is an *expansion-based approach:* universal variables are eliminated at the outset by implicitly expanding the universal quantifiers into conjunctions, creating annotated copies of existential variables. The systems $\forall$Exp + Res, IR, and IRM [11, 24] are of this type. The second is a *reduction-rule approach:* under certain conditions, resolution may be blocked, and also under certain conditions, universal variables can be deleted from clauses. The conditions are formulated to preserve soundness, ensuring that if a QBF is true, then so is the QBF resulting from adding a derived clause. The systems QRes, QURes, CP + $\forall$Red [13, 26, 37] are of this type.

A central role in QBF proof complexity is played by the *two-player evaluation game* on QBFs, and the existence of winning strategies for the universal player in false QBFs. For many QBF resolution systems, such strategies were used to construct proofs and demonstrate completeness, and soundness was demonstrated by extracting such strategies from proofs [1, 11, 21]. The *strategy extraction* procedures build partial strategies at each line of the proof, with the strategies at the final line forming a complete countermodel. These extraction procedures are based on the fact that in each application of a rule in the proof system, any winning strategies of the existential player are not destroyed.

In the systems QRes [26] and QURes [37], the soundness of the resolution rule is ensured by enforcing a very simple side-condition: variables other than the pivot cannot appear in both polarities in the antecedents. It was observed early on that this is often too restrictive. The *long-distance resolution proof system* LD-QRes [1, 39] arose from efforts to have less restrictive but still sound rules. In this system, a universal variable could appear in both polarities and get merged in the consequent, provided it was to the right of the pivot in the quantifier prefix. This preserves soundness, but the strategy extraction procedures become notably more complex.

The system LD-QRes, while provably better than QRes [21], is still needlessly restrictive in some situations. In particular, by checking a very simple syntactic prefix-ordering condition, it fails to exploit the fact that soundness is not lost even if universal variables to the left of the pivot are merged in both antecedents, provided the partial strategies built for them in both antecedents are identical. A *new system Merge Resolution* (*MRes*) was introduced last year [6] by a subset of the current authors, precisely to address this point. In MRes, partial strategies are explicitly represented within the proof, in a particular representation format called merge maps – these are essentially deterministic branching programs (DBPs). In this format, isomorphism checking can be done efficiently, and this opens the way for enabling sound applications of resolution that would have been blocked in LD-QRes (and QRes). In [6], it was shown that this brought a rich pay-off: there is a family of formulas, the SquaredEquality formulas, with short (linear-size) proofs in MRes, even in its tree-like and regular versions, but requiring exponential size in QRes, QURes, CP + $\forall$Red, $\forall$Exp + Res, and IR. It is notable that the hardness of SquaredEquality in these systems stems from a certain semantic cost associated with these formulas and a corresponding lower bound [4, 5]. Thus the results of [6] show that such semantic costs are not a barrier for MRes.

In this paper, we explore the price paid for overcoming the semantic cost barrier. We

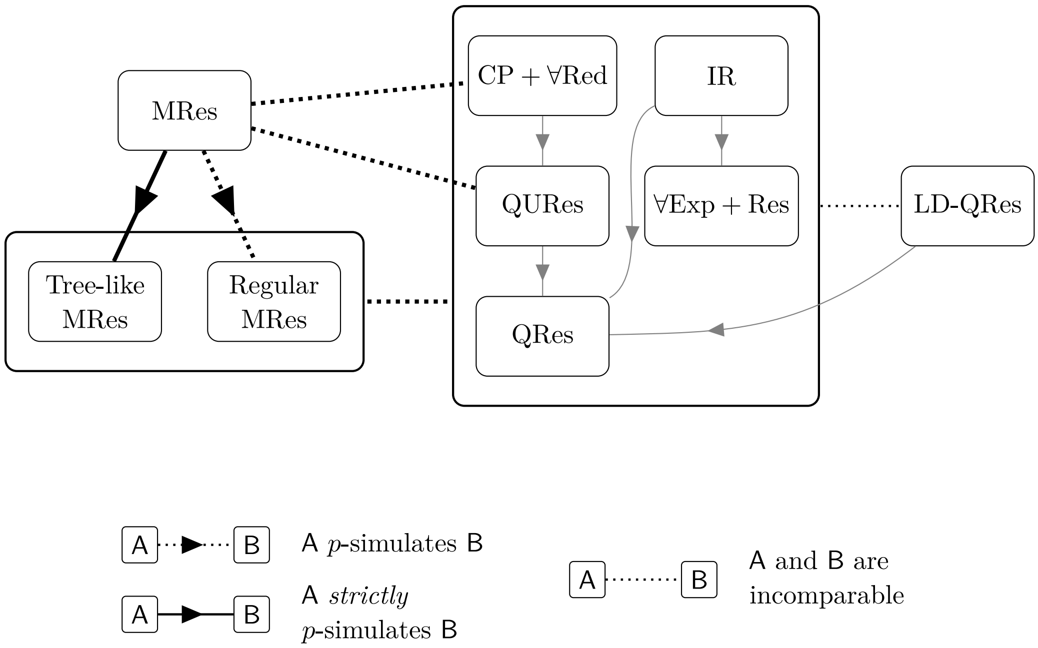

Figure 1 Visual summary of the proof complexity landscape, with new results shown in bold. Tree-like and regular MRes are also incomparable with the tree-like versions of the five systems in the big box.

show that (expectedly) MRes is not an unqualified success story. Building strategies into proofs via merge maps, and screening out unsoundness only through isomorphism tests, comes at a fairly heavy price.

(A) Lower bounds from circuit complexity for restricted versions of MRes. Since the strategies are explicitly represented inside the proofs, computational hardness of strategies immediately translates to proof size lower bounds. While computational hardness of strategies is a known source of hardness in all reduction-based proof systems admitting efficient strategy extraction [9,11], the computational model relevant for MRes is one for which no unconditional lower bounds are known. For tree-like and regular MRes, the relevant models are decision trees and read-once DBPs, where lower bounds are known. Using this approach, we show:

1. Tree-like MRes is exponentially weaker than MRes.
   The QParity formulas witness the separation (Theorem 8) as their unique countermodel is the parity function which requires large decision trees.
2. Tree-like MRes is incomparable with the dag-like and tree-like versions of QRes, QURes, CP + ∀Red, ∀Exp + Res and IR.
   One direction was shown in [6] via the Equality formulas: these formulas are easy for tree-like MRes but hard for dag-like QRes, QURes, CP+∀Red, ∀Exp+Res, IR. The other direction is witnessed by the Completion Principle formulas, easy in tree-like versions of QRes and ∀Exp + Res [23, 24], but exponentially hard for tree-like MRes (Theorem 11). Unlike the QParity formulas, these formulas do not have unique countermodels. However, we show that every countermodel requires large decision-tree size, and hence obtain the lower bound for tree-like MRes.

(B) Combinatorial lower bounds for full MRes. Even when winning strategies are unique and easy to compute by DBPs, the formulas can be hard for MRes. We establish such hardness in three cases, obtaining more incomparabilities.

1. The LQParity formulas, easy in $\forall$Exp + Res [11], are exponentially hard for regular MRes (Theorem 15). Hence regular MRes is incomparable with $\forall$Exp + Res and IR.
2. The Completion Principle formulas, easy in tree-like versions of QRes and $\forall$Exp + Res [23, 24], are exponentially hard for regular MRes (Theorem 20). Hence regular MRes is incomparable with the dag-like and tree-like versions of QRes, QURes, CP + $\forall$Red, $\forall$Exp + Res and IR.
3. The KBKF-lq formulas, easy in QURes [2], are exponentially hard for MRes (Theorem 24). Hence MRes is incomparable with QURes and CP + $\forall$Red.

The third hardness result above for the KBKF-lq formulas provides the first lower bound for the full system of MRes, for which previously no lower bounds were known.

It may be noted that for existentially quantified QBFs, all the QBF proof systems mentioned in this paper coincide with Resolution (or in case of CP + $\forall$Red, with Cutting Planes). Therefore lower bounds for these propositional proof systems trivially lift to the corresponding QBF proof system. In particular, the separations of tree-like and regular MRes from MRes and other systems follow from the propositional case. However, such lower bounds do not tell us much about the limitations of the QBF proof system other than what is known from the underlying propositional proof system. Therefore, in QBF proof complexity, we are interested in 'genuine' QBF lower bounds, i.e. lower bounds that do not follow from propositional lower bounds (cf. [15] on how to formally define the notion of 'genuine' lower bounds). The lower bounds we establish here are of this nature.

Figure 1 depicts the *simulation order and incomparabilities* we establish involving MRes and its refinements. Amongst the five systems in the big box, all relationships not directly implied by depicted connections are known to be incomparabilities [11, 13, 24].

2 Preliminaries

Let $[n] = \{1, 2, \ldots, n\}$ and $[m, n] = \{m, \ldots, n\}$. We represent clauses by sets of literals.

The *resolution rule* derives, from clauses $C \vee x$ and $D \vee \neg x$, the clause $C \vee D$. We say that $C \vee D$ is the resolvent, x is the pivot, and denote this by $C \vee D = \text{res}(C \vee x, D \vee \neg x, x)$.

The *propositional proof system Resolution* proves that a CNF formula F is unsatisfiable by deriving the empty clause through repeated applications of the resolution rule.

Quantified Boolean formulas. A *Quantified Boolean formula* (QBF) in *prenex conjunctive normal form* is denoted $\Phi \coloneqq Q \cdot \phi$, where (a) $Q = Q_1 Z_1 Q_2 Z_2 \ldots Q_k Z_k$ is the quantifier prefix, in which Z_i are pairwise disjoint finite sets of Boolean variables, $Q_i \in \{\exists, \forall\}$ for each $i \in [k]$ and $Q_i \neq Q_{i+1}$ for each $i \in [k-1]$, and (b) the matrix ϕ is a CNF over $\text{vars}(\Phi) \coloneqq \cup_{i \in [k]} Z_i$.

The existential (resp. universal) variables of Φ, typically denoted X or $X_\exists$ (resp. U or $X_\forall$) is the set obtained as the union of Z_i for which $Q_i = \exists$ (resp. $Q_i = \forall$). The prefix Q defines a binary relation $<_Q$ on $\text{vars}(\Phi)$, such that $z <_Q z'$ holds iff $z \in Z_i$, $z' \in Z_j$, and $i < j$, in which case we say that z' is right of z and z is left of z'. For each $u \in U$, we define $L_Q(u) \coloneqq \{x \in X \mid x <_Q u\}$, i.e. the existential variables left of u.

For a set of variables Z, let $\langle Z \rangle$ denote the set of assignments to Z. A *strategy* h for a QBF Φ is a set $\{h^u \mid u \in U\}$ of functions $h^u \colon \langle L_Q(u) \rangle \to \{0, 1\}$ (for each $\alpha \in \langle X \rangle$, $h^u(\alpha\restriction_{L_Q(u)})$ and $h(\alpha)$ should be interpreted as a Boolean assignment to the variable u and the variable set U respectively). Additionally h is *winning* if, for each $\alpha \in \langle X \rangle$, the restriction of ϕ by the assignment $(\alpha, h(\alpha))$ is false. We use the terms "winning strategy" and "countermodel" interchangeably. A QBF is called false if it has a countermodel, and true if it does not.

The semantics of QBFs is also explained by a *two-player evaluation game* played on a QBF. In a run of the game, two players, the existential and the universal player, assign values to the variables in the order of quantification in the prefix. The existential player wins if the assignment so constructed satisfies all the clauses of ϕ; otherwise the universal player wins. Assigning values according to a countermodel guarantees that the universal player wins no matter how the existential player plays; hence the term "winning strategy".

2.1 The formulas

We describe the formulas we will use throughout the paper.

The QParity and LQParity formulas [11]. Let $\text{parity}^c(y_1, y_2, \ldots, y_k)$ be a shorthand for the following conjunction of clauses: $\bigwedge_{S\subseteq[k],\,|S|\equiv 1 \,(\text{mod } 2)} \left((\vee_{i\in S}\overline{y_i}) \vee (\vee_{i\notin S} y_i)\right)$. Thus $\text{parity}^c(y_1, y_2, \ldots, y_k)$ is equal to 1 iff $y_1 + y_2 + \cdots + y_k \equiv 0 \pmod 2$. QParity_n is the QBF $\exists x_1, \ldots, x_n, \forall z, \exists t_1, \ldots, t_n.\ \left(\bigwedge_{i\in[n+1]} \phi_n^i\right)$ where

$$\phi_n^1 = \text{parity}^c(x_1, t_1); \quad \forall i \in [2, n],\ \phi_n^i = \text{parity}^c(t_{i-1}, x_i, t_i); \quad \phi_n^{n+1} = (t_n \vee z) \wedge \left(\overline{t_n} \vee \overline{z}\right).$$

The QBFs are false: they claim that there exist $x_1, \ldots, x_n$ such that $x_1 + \cdots + x_n$ is neither congruent to 0 nor 1 modulo 2. Note that the only winning strategy for the universal player is to play z satisfying $z \equiv x_1 + \cdots + x_n \pmod 2$.

Similarly, let $\widehat{\text{parity}}^c(y_1, y_2, \ldots, y_k, z)$ abbreviate $\bigwedge_{C\in\text{parity}^c(y_1,y_2,\ldots,y_k)} \left((C \vee z) \wedge (C \vee \overline{z})\right)$. LQParity_n is the QBF $\exists x_1, \ldots, x_n, \forall z, \exists t_1, \ldots, t_n.\ \left(\bigwedge_{i\in[n+1]} \phi_n^i\right)$ where

$$\phi_n^1 = \widehat{\text{parity}}^c(x_1, t_1, z); \quad \forall i \in [2, n], \phi_n^i = \widehat{\text{parity}}^c(t_{i-1}, x_i, t_i, z); \quad \phi_n^{n+1} = (t_n \vee z) \wedge \left(\overline{t_n} \vee \overline{z}\right).$$

For both QParity_n and LQParity_n, for $i, j \in [n+1], i \leq j$, we let $\phi_n^{[i,j]}$ denote $\bigwedge_{k\in[i,j]} \phi_n^k$. Also, $X = \{x_1, \ldots, x_n\}$ and $T = \{t_1, \ldots, t_n\}$.

▶ **Observation 1.** *For both $QParity_n$ and $LQParity_n$:* (a) *for each $i \in [n]$, and each $C \in \phi_n^i$, $\{x_i, t_i\} \subseteq var(C)$; and* (b) *for each $i \in [n+1] \setminus \{1\}$, and each $C \in \phi_n^i$, $\{t_{i-1}\} \subseteq var(C)$.*

The Completion Principle formulas CR_n [24]. The QBF CR_n is defined as follows:

$$\text{CR}_n = \underset{i,j\in[n]}{\exists} x_{ij}, \forall z, \underset{i\in[n]}{\exists} a_i, \underset{j\in[n]}{\exists} b_j.\ \left(\bigwedge_{i,j\in[n]} (A_{ij} \wedge B_{ij})\right) \wedge L_A \wedge L_B$$

where $A_{ij} = x_{ij} \vee z \vee a_i$, $B_{ij} = \overline{x_{ij}} \vee \overline{z} \vee b_j$, $L_A = \overline{a_1} \vee \cdots \vee \overline{a_n}$, and $L_B = \overline{b_1} \vee \cdots \vee \overline{b_n}$. Let X, A, B denote the variable sets $\{x_{ij} : i, j \in [n]\}$, $\{a_i : i \in [n]\}$, and $\{b_j : j \in [n]\}$. It is convenient to think of the X variables as arranged in an $n \times n$ matrix.

Intuitively, the formulas describe a completion game, played on the matrix

$$\begin{pmatrix} a_1 & \ldots & a_1 & \ldots & a_n & \ldots & a_n \\ b_1 & \ldots & b_n & \ldots & b_1 & \ldots & b_n \end{pmatrix}$$

where the $\exists$-player first deletes exactly one cell per column and the $\forall$-player then chooses one row. The $\forall$-player wins if his row contains all of A or all of B (cf. [24]).

The KBKF-lq$[n]$ formulas [2]. Our last QBFs are a variant of the formulas introduced by Kleine Büning et al. [26], which in various versions appear prominently throughout the

QBF literature [2,5,11,21,37]. For $n > 1$, the nth member of the KBKF-lq[n] family consists of the prefix $\exists d_1, e_1, \forall x_1, \exists d_2, e_2, \forall x_2, \dots, \exists d_n, e_n, \forall x_n, \exists f_1, f_2, \dots, f_n$ and clauses

$$
\begin{array}{llll}
A_0 = \{\overline{d_1}, \overline{e_1}, \overline{f_1}, \dots, \overline{f_n}\} & & & \\
A_i^d = \{d_i, x_i, \overline{d_{i+1}}, \overline{e_{i+1}}, \overline{f_1}, \dots, \overline{f_n}\} & A_i^e = \{e_i, \overline{x_i}, \overline{d_{i+1}}, \overline{e_{i+1}}, \overline{f_1}, \dots, \overline{f_n}\} & \forall i \in [n-1] \\
A_n^d = \{d_n, x_n, \overline{f_1}, \dots, \overline{f_n}\} & A_n^e = \{e_n, \overline{x_n}, \overline{f_1}, \dots, \overline{f_n}\} & \\
B_i^0 = \{x_i, f_i, \overline{f_{i+1}}, \dots \overline{f_n}\} & B_i^1 = \{\overline{x_i}, f_i, \overline{f_{i+1}}, \dots \overline{f_n}\} & \forall i \in [n-1] \\
B_n^0 = \{x_n, f_n\} & B_n^1 = \{\overline{x_n}, f_n\} &
\end{array}
$$

Note that the existential part of each clause in KBKF-lq[n] is a Horn clause (at most one positive literal), and except A_0, is even strict Horn (exactly one positive literal).

We use the following shorthand notation. Sets of variables: $D = \{d_1, \dots, d_n\}$, $E = \{e_1, \dots, e_n\}$, $F = \{f_1, \dots, f_n\}$, and $X = \{x_1, \dots, x_n\}$. Sets of literals: For $Y \in \{D, E, X, F\}$, set $Y^1 = \{u \mid u \in Y\}$ and $Y^0 = \{\overline{u} \mid u \in Y\}$. Sets of clauses:

$$
\begin{array}{llll}
\mathcal{A}_0 = \{A_0\} & & & \\
\mathcal{A}_i = \{A_i^d, A_i^e\} & \forall i \in [n] & \mathcal{B}_i = \{B_i^0, B_i^1\} & \forall i \in [n] \\
\mathcal{A}_{[i,j]} = \cup_{k \in [i,j]} \mathcal{A}_k & \forall i, j \in [0, n], i \leq j & \mathcal{B}_{[i,j]} = \cup_{k \in [i,j]} \mathcal{B}_k & \forall i, j \in [n], i \leq j \\
\mathcal{A} = \mathcal{A}_{[0,n]} & & \mathcal{B} = \mathcal{B}_{[1,n]} &
\end{array}
$$

We use the following property of these formulas:

▶ **Proposition 2.** *Let h be any countermodel for KBKF-lq[n]. Let α be any assignment to D, and β be any assignment to E.*
For each $i \in [n]$, if $\alpha_j \neq \beta_j$ for all $1 \leq j \leq i$, then $h^{x_i}\big((\alpha, \beta)\restriction_{LQ(x_i)}\big) = \alpha_i$.

In particular, if $\alpha_j \neq \beta_j$ for all $j \in [n]$, then the countermodel computes $h(\alpha, \beta) = \alpha$.

Proof. Let h be any countermodel for KBKF-lq[n]. For $i \in [n]$, let α^i be an assignment to $\{d_1, \dots, d_i\}$, and β^i be an assignment to $\{e_1, \dots, e_i\}$. For $j \leq i$, let α_j^i (resp. β_j^i) be the assignment to d_j (resp. e_j) set by the assignment α_j^i (resp. β_j^i). We will show that for each $i \in [n]$, if $\alpha_j^i \neq \beta_j^i$ for all $1 \leq j \leq i$, then $h^{x_i}(\alpha^i, \beta^i) = \alpha_i^i$. This implies the claimed result.

Fix some $i \in [n]$. Assume to the contrary that $\alpha_j^i \neq \beta_j^i$ for all $1 \leq j \leq i$ and $h^{x_i}(\alpha^i, \beta^i) \neq \alpha_i^i$. We will give a winning strategy for the existential player. Note that all clauses in $\mathcal{A}[0, i-1]$ are satisfied by the partial assignment (α^i, β^i). The existential player sets $d_j = e_j = 1$ for all $j > i$ and sets $f_j = 1$ for all $j \in [n]$. This satisfies all the remaining clauses, irrespective of the strategy of the universal player. Therefore the existential player wins. This contradicts the assumption that h is a countermodel for KBKF-lq[n]. ◀

2.2 The Merge Resolution proof system [6]

The formal definition of the *Merge Resolution proof system*, denoted MRes, is rather technical and can be found in [6]. Here we present a somewhat informal description.

First, we describe the *idea behind the proof system.* MRes is a line-based proof system. Each line L has a clause C with only existential literals, and a partial strategy h^u for each universal variable u. The idea is to maintain the invariant that for each existential assignment α, if α falsifies C, then α extended by the partial universal assignment setting each u to $h^u(\alpha)$ falsifies at least one of the clauses used to derive L. Thus the set of functions $\{h^u\}$ gives a partial strategy that wins whenever the existential player plays from the set of assignments falsifying C. The goal is to derive a line with the empty clause; the corresponding strategy at that line will be a complete winning strategy, a countermodel. Along the way, resolution is used on the clauses. If the pivot is x, then for universal variables u right of x, the partial strategies can be combined with a branching decision on x. However, for u left of x, in the evaluation game, the value of u is already set when x is to be assigned. Thus already existing non-trivial partial strategies for u cannot be combined with a branching decision, and so this resolution step is blocked. However, if both the strategies are identical, or if one of them is trivial (unspecified), then the non-trivial strategy can be carried forward while maintaining the desired invariant. Checking whether strategies are identical can itself be hard, making verification of the proof difficult. In MRes, this is handled by choosing a particular representation called merge maps, where isomorphism checks are easy.

Now we can describe the proof system itself. First we describe *merge maps*. Syntactically, these are deterministic branching programs, specified by a sequence of instructions of one of the following two forms:

- $\langle \mathsf{line}\ \ell\rangle : b$ where $b \in \{*, 0, 1\}$.[1]
  Merge maps containing a single such instruction are called simple. In particular, if $b = *$, then they are called trivial.
- $\langle \mathsf{line}\ \ \ell\rangle$: If $x = 0$ then go to $\langle \mathsf{line}\ \ell_1\rangle$ else go to $\langle \mathsf{line}\ \ell_2\rangle$, for some $\ell_1, \ell_2 < \ell$. In a merge map M for u, all queried variables x must precede u in the quantifier prefix.
  Merge maps with such instructions are called complex.

(All line numbers are natural numbers.) The merge map M^u computes a partial strategy for the universal variable u starting at the largest line number (the leading instruction) and following the instructions in the natural way. The value $*$ denotes an undefined value.

Two merge maps M_1, M_2 are said to be consistent, denoted $M_1 \bowtie M_2$, if for every line number i appearing in both M_1, M_2, the instructions with line number i are identical. Two merge maps M_1, M_2 are said to be isomorphic, denoted $M_1 \simeq M_2$, if there is a bijection between the line numbers in M_1 and M_2 that transforms M_1 to M_2 in the natural way.

For the remainder of this section let $\Phi = Q \cdot \phi$ be a QBF with existential variables X and universal variables U. The *proof system MRes* has the following rules:

1. *Axiom:* For a clause A in the matrix ϕ, let C be the existential part of A. For each universal variable u, let b_u be the value u must take to falsify A; if $u \notin \text{var}(A)$, then $b_u = *$. For any natural number i, the line $(C, \{M^u : u \in U\})$ where each M^u is the simple merge map $\langle i\rangle : b_u$ can be derived in MRes.
2. *Resolution:* From lines $L_a = (C_a, \{M_a^u : u \in U\})$ for $a \in \{0, 1\}$, in MRes, the line $L = (C, \{M^u : u \in U\})$ can be derived, where for some $x \in X$,
   - $C = \text{res}(C_0, C_1, x)$, and
   - for each $u \in U$, either M_a^u is trivial and $M^u = M_{1-a}^u$ for some a, or $M^u = M_0^u \simeq M_1^u$, or x precedes u and M^u has a leading instruction that builds the complex merge map If $x = 0$ then $\langle M_0^u\rangle$ else $\langle M_1^u\rangle$.

A *refutation* is a derivation using these rules and ending in a line with the empty existential clause. The size of the refutation is the number of lines. In the rest of this paper, we will denote refutations by the Greek letter Π.

▶ **Example 3.** We reproduce from [6] a small example to illustrate how MRes operates. The formulas to be refuted are the Equality formulas from [5], defined as follows: The *equality family* is the QBF family whose nth instance has the prefix $\exists x_1, \ldots, x_n, \forall u_1, \ldots, u_n, \exists t_1, \ldots, t_n$ and the matrix consisting of the clauses $\{x_i, u_i, t_i\}, \{\overline{x}_i, \overline{u}_i, t_i\}$ for $i \in [n]$, and $\{\bar{t}_1, \ldots, \bar{t}_n\}$.

[1] In [6], the notation used is $b \in \{*, u, \overline{u}\}$; $u, \overline{u}, *$ denote $u = 1, u = 0$, undefined respectively.

In [6] (Example 3), linear-size reductionless LDQRes refutations are described for these formulas, and later, MRes is shown to simulate reductionless LDQRes. Here, we directly present the implied linear-size MRes refutations.

First, we download the axioms. Line 0 downloads the long clause, with all trivial merge maps. The next $2n$ lines download the short axiom clauses. Letting $i \in [n]$, we define these lines as follows:
Line $2i-1$ is the clause $\{x_i, t_i\}$ with merge map 0 for u_i and all other merge maps are trivial.
Line $2i$ is the clause $\{\overline{x}_i, t_i\}$ with merge map 1 for u_i and all other merge maps are trivial.

For $i \in [n]$, line $2n+i$ is obtained by applying the merge resolution rule on lines $2i-1$ and $2i$. This gives the clause $\{t_i\}$; the merge maps for $j \neq i$ are trivial, and the merge map for u_i has the instruction: If $x_i = 0$ then go to $\langle$line $2i-1\rangle$ else go to $\langle$line $2i\rangle$.

At line $3n+1$, applying merge resolution on lines 0 and $2n+1$, we obtain the clause $\{\bar{t}_2, \ldots, \bar{t}_n\}$. The merge map for u_1 is taken from line $2n+1$, since at line 0 it is trivial.

Now for $i \in [2, n]$, line $3n+i$ is obtained by applying merge resolution on lines $2n+i$ and $3n+i-1$. This gives the clause $\{\bar{t}_{i+1}, \ldots, \bar{t}_n\}$. The merge map for u_i is taken from line $2n+i$ since at line $3n+i-1$ it is trivial. For $j < i$, the merge map for u_j is taken from line $3n+i-1$ since at line $2n+i$ it is trivial. Effectively, at this line, for all $j \leq i$, the merge map for u_j is from line $2n+j$, and for all $j > i$, the merge map for u_j is trivial.

Line $4n$ derives the empty clause and the strategy computing, for each $i \in [n]$, $u_i = x_i$. This completes the refutation. ⌟

As shown in [6], the merge maps at the final line compute a countermodel for the QBF. To establish this, some stronger properties of the derivation are established and will be useful to us. We restate the relevant properties here.

▶ **Lemma 4** (Extracted/adapted from [6] Section 4.3, (Proof of Lemma 21))**.** *Let $\Phi = Q \cdot \phi$ be a QBF with existential variables X and universal variables U. Let $\Pi \stackrel{\text{def}}{=} L_1, \ldots, L_m$ be an MRes refutation of Φ, where each $L_i = (C_i, \{M_i^u \mid u \in U\})$. Further, for each $i \in [m]$,*

- *let α_i be the minimal partial assignment falsifying C_i,*
- *let A_i be the set of assignments to X consistent with α_i,*
- *for each $u \in U$, let h_i^u be the function computed by M_i^u,*
- *for each $\alpha \in A_i$, let $h_i(\alpha)$ be the partial assignment which sets variable u to $h_i^u(\alpha\restriction_{L_Q(u)})$ if $h_i^u(\alpha\restriction_{L_Q(u)}) \neq *$, and leaves it unset otherwise.*

Then for each $\alpha \in A_i$, the (partial) assignment $(\alpha, h_i(\alpha))$ falsifies at least one clause of ϕ used in the sub-derivation of L_i.

Let G_Π be the derivation graph corresponding to Π (with edges directed from the antecedents to the consequent, hence from the axioms to the final line).

▶ **Proposition 5** ([6])**.** *For all $u \in U$, M_m^u is isomorphic to a subgraph of G_Π (up to path contraction).*

Let S be a subset of the existential variables X of Φ. We say that an MRes refutation of Φ is *S-regular* if for each $x \in S$, there is no leaf-to-root path that uses x as pivot more than once. An X-regular proof is simply called a *regular proof.* If G_Π is a tree, then we say that Π is a *tree-like proof.*

3 Lifting branching program lower bounds

We now start to explore lower bounds for MRes, first for its version where proofs are tree-like. The following lemma is an immediate consequence of Proposition 5.

▶ **Lemma 6.** *Let $\Pi \stackrel{\text{def}}{=} L_1, \ldots, L_m$ be an MRes refutation. If Π is tree-like (resp. regular), then for all $u \in U$, M^u_m is a decision tree (resp. read-once branching program). Moreover, the size of Π is lower bounded by the size of M^u_m.*

This lemma allows us to lift lower bounds for decision trees (resp. read-once branching programs) to lower bounds for tree-like (resp. regular) Merge Resolution.

For QParity_n and LQParity_n, the only winning strategy for the universal player is to set z such that $z \equiv x_1 + x_2 + \cdots + x_n \pmod 2$.

▶ **Proposition 7** (Folklore)**.** *The decision-tree size complexity of the parity function is 2^n.*

From Lemma 4, Lemma 6, and Proposition 7, we obtain the desired lower bound.

▶ **Theorem 8.** $\mathit{size}_{\mathit{MResTree}}(\mathit{QParity}_n) = 2^{\Omega(n)}$ *and* $\mathit{size}_{\mathit{MResTree}}(\mathit{LQParity}_n) = 2^{\Omega(n)}$.

▶ **Corollary 9.** *Tree-like MRes is exponentially weaker than MRes.*

Proof. Theorem 8 shows that QParity requires exponential-size refutations in tree-like MRes. It has polynomial-size refutations in reductionless LD-QRes [31] (and hence also in MRes). The result follows. ◀

For the QBF CR_n, the winning strategy for the universal player (countermodel) is not unique. However, we show that all countermodels require large decision trees.

▶ **Lemma 10.** *Every countermodel for CR_n has decision tree size complexity at least 2^n.*

Proof. We prove the size bound by showing that in every decision tree for every countermodel, all root-to-leaf paths query at least n variables, and hence the decision tree has at least 2^n nodes.

Assume to the contrary that some countermodel h is computed by a decision tree M that has a root-to-leaf path p querying less than n variables. Then there exist $k, \ell \in [n]$ such that no variable from Row k and no variable from Column ℓ is on this path. Let ρ_p be the minimal partial assignment that takes this path in M, and let ρ' be an arbitrary extension of ρ_p to variables in $\{x_{ij} \mid i \neq k, j \neq \ell\}$. Consider the following extension of ρ' to variables in $(X \setminus \{x_{k\ell}\}) \cup T$, giving assignment σ:
Set all variables in row k (other than $x_{k,\ell}$) to 1.
Set all variables in column ℓ (other than $x_{k,\ell}$) to 0.
Set a_k and b_ℓ to 0 and all other a_i, b_j variables to 1.

For $n \geq 2$, σ satisfies all the clauses of CR_n except $A_{k\ell}$ and $B_{k\ell}$, which get restricted to $x_{k\ell} \vee z$ and $\overline{x_{k\ell}} \vee \overline{z}$ respectively.

Let $\alpha_0 = \sigma \cup \{x_{k\ell} = 0\}$ and $\alpha_1 = \sigma \cup \{x_{k\ell} = 1\}$. Since both α_0 and α_1 extend ρ_p, they follow path p, therefore $h(\alpha_0) = h(\alpha_1)$. If $h(\alpha_0) = h(\alpha_1) = 0$, then $(\alpha_1, h(\alpha_1))$ satisfies all clauses of CR_n. On the other hand, if $h(\alpha_0) = h(\alpha_1) = 1$, then $(\alpha_0, h(\alpha_0))$ satisfies all clauses of CR_n. Thus in either case, h is not a countermodel for CR_n. ◀

From Lemmas 4, 6, and 10, we obtain the desired lower bound.

▶ **Theorem 11.** $\mathit{size}_{\mathit{MResTree}}(\mathit{CR}_n) = 2^{\Omega(n)}$.

▶ **Corollary 12.** *Tree-Like MRes is incomparable with the tree-like and general versions of QRes, QURes, CP + ∀Red, ∀Exp + Res, and IR.*

Proof. We showed in Theorem 11 that the Completion Principle CR_n requires exponential-size refutations in tree-like Merge Resolution. It has polynomial-size refutations in tree-like QRes [23] (and hence also in QURes and CP+∀Red) and tree-like ∀Exp+Res [24] (and hence also in IR). (While [24] does not explicitly mention tree-like proofs, the proof provided there for CR_n is tree-like.) On the other hand, the Equality formulas have polynomial-size tree-like MRes refutations [6] but require exponential-size refutations in QRes, QURes, CP + ∀Red [5], ∀Exp + Res, IR [4] (cf. [3] on how to apply the lower bound technique from [4] to the Equality formulas). ◀

We now show how to lift lower bounds for read-once branching programs to those for regular MRes. This follows the method used, for instance, in [11] (Section 4.1) and [31] (Section 6). Let $f\colon X \to \{0,1\}$ be a Boolean function, let C_f be a Boolean circuit encoding f, and let u be a variable not in X. Using Tseitin transformation [36], we can construct a CNF formula $\phi(X,u,Y)$ such that $\exists Y.\phi(X,u,Y)$ is logically equivalent to $C_f(X) \neq u$. Therefore, $\Phi := \exists X \forall u \exists Y.\phi(X,u,Y)$, called the QBF encoding of f, is a false QBF formula with f as the unique winning strategy. Moreover, the size of Φ is polynomial in the size of C_f. Choosing a function f that can be computed by polynomial-size Boolean circuits but requires exponential-size read-once branching programs gives the desired lower bound. Many such functions are known [38]. For instance, we can use the following result:

▶ **Theorem 13** ([18]). *There is a Boolean function f in n variables that can be computed by a Boolean circuit of size $O(n^{3/2})$ but requires read-once branching programs of size $2^{\Omega(\sqrt{n})}$.*

▶ **Corollary 14.** *There is a Boolean function f in n variables with a QBF encoding Φ of size polynomial in n such that any regular MRes refutation of Φ has size $2^{\Omega(\sqrt{n})}$.*

4 Lower bounds for Regular Merge Resolution

In this section, we prove Regular MRes lower bounds for formulas whose countermodels can be computed by polynomial-size read-once branching programs.

4.1 LQParity formulas

Our first result concerns the long-distance versions of the parity formulas [11] (cf. Section 2.1), which are known to be hard for LD-QRes. We establish that they are hard for regular Merge Resolution as well.

▶ **Theorem 15.** $\mathit{size}_{\mathit{MResReg}}(\mathit{LQParity}_n) = 2^{\Omega(n)}$.

This follows from a stronger result that we prove below: any T-regular refutation of $\mathrm{LQParity}_n$ in MRes must have size $2^{\Omega(n)}$ (Theorem 19).

The proof proceeds as follows: Let Π be a T-regular MRes refutation of $\mathrm{LQParity}_n$. Since every axiom has a variable from T while the final clause in Π is empty, there is a maximal "component" of the proof leading to and including the final line, where all clauses are T-free. The clauses in this component involve only the X variables. We show that the "boundary" of this component is large, by showing in Lemma 18 that each clause here must be wide. (This idea was used in [31] to show that CR is hard for reductionless LD-QRes.) To establish the width bound, we note that no lines have trivial strategies. Since the pivots at the boundary are variables from T, the merge maps incoming into each boundary resolution must be isomorphic. By carefully analysing which axiom clauses can and must be used to derive lines just above the boundary (Lemma 17), we conclude that the merge maps must be simple, yielding the lower bound. To fill in all the details, we first describe some properties (Lemma 16) of Π that will be used in obtaining this result.

The lines of Π will be denoted by L, L', L'' etc. For lines L and L' the respective clause, merge map and the function computed by the merge map will be denoted by C, M, h and C', M', h' respectively. Let G_Π be the derivation graph corresponding to Π (with edges directed from the antecedents to the consequent, hence from the axioms to the final line). We will refer to the nodes of this graph by the corresponding line. For $L, L' \in \Pi$, we will say $L \rightsquigarrow L'$ if there is a path from L to L' in G_Π.

For a line $L \in \Pi$, let Π_L be the minimal sub-derivation of L, and let G_{Π_L} be the corresponding subgraph of G_Π with sink L. Define $\text{UsedConstraints}(\Pi_L) = \{\phi_n^i \mid i \in [n+1], \text{leaves}(G_{\Pi_L}) \cap \phi_n^i \neq \emptyset\}$, and $\text{Uci}(\Pi_L) = \{i \in [n+1] \mid \phi_n^i \in \text{UsedConstraints}(\Pi_L)\}$. (Uci stands for UsedConstraintsIndex.) Note that for any leaf L, $\text{Uci}(\Pi_L)$ is a singleton.

Define $\mathcal{S}'$ to be the set of those lines in Π where the clause part has no T variable and furthermore there is a path in G_Π from the line to the final empty clause via lines where all the clauses also have no T variables. Let $\mathcal{S}$ denote the set of leaves in the subgraph of G_Π restricted to $\mathcal{S}'$; these are lines that are in $\mathcal{S}'$ but their parents are not in $\mathcal{S}'$. Note that no leaf of Π is in $\mathcal{S}'$ because all leaves of G_Π contain a variable in T.

▶ **Lemma 16.** *Let $L = (C, M)$ be a line of Π. Then $\text{Uci}(\Pi_L)$ is an interval $[i, j]$ for some $1 \leq i \leq j \leq n+1$. Furthermore, (below i, j refer to the endpoints of this interval)*

1. *For all $k \in [i, j-1]$, $t_k \notin var(C)$.*
2. *If $i > 1$, then $t_{i-1} \in var(C)$.*
3. *If $j \leq n$, then $t_j \in var(C)$.*
4. *$|var(C) \cap T| = 1$ iff $[i, j]$ contains exactly one of $1, n+1$.*
   $var(C) \cap T = \emptyset$ iff $[i, j] = [1, n+1]$.
5. *For all $k \in [i, j] \cap [1, n]$, $x_k \in var(C) \cup var(M)$.*

Proof. Let $I = \text{Uci}(\Pi_L)$. Assume, to the contrary, that I is not an interval; for some $k \in [2, n]$, I contains an index $i < k$ and an index $j > k$, but does not contain k. Let L' be the first line in Π such that $\text{Uci}(\Pi_{L'})$ intersects both $[1, k-1]$ and $[k+1, n+1]$. Since leaves have singleton Uci sets, L' is not a leaf. Say $L' = \text{res}(L'', L''', v)$. Assume that $\text{Uci}(\Pi_{L''}) \subseteq [1, k-1]$ and $\text{Uci}(\Pi_{L'''}) \subseteq [k+1, n+1]$; the argument for the other case is identical. So $v \in \text{var}_\exists(\text{UsedConstraints}(\Pi_{L''})) \subseteq \text{var}_\exists(\phi_n^{[1,k-1]})$, and $v \in \text{var}_\exists(\text{UsedConstraints}(\Pi_{L'''})) \subseteq \text{var}_\exists(\phi_n^{[k+1,n+1]})$. But $\text{var}_\exists(\phi_n^{[1,k-1]})$ and $\text{var}_\exists(\phi_n^{[k+1,n+1]})$ are disjoint, a contradiction.

Fixing i, j so that $I = \text{Uci}(\Pi_L) = [i, j]$, we now prove the remaining statements in the Lemma.

1. Fix any $k \in [i, j-1]$. Note that $\{k, k+1\} \subseteq \text{Uci}(\Pi_L)$. Let L' be the first line in Π_L such that $\{k, k+1\} \subseteq \text{Uci}(\Pi_{L'})$. Say L' is obtained as $\text{res}(L'', L''', v)$. Assume that $\text{Uci}(\Pi_{L''})$ contributes k and $\text{Uci}(\Pi_{L''})$ contributes $k+1$; the other case is symmetric. Since $\text{Uci}(\Pi_{L''})$ must also be an interval, and since it contains k but not $k+1$, $\text{Uci}(\Pi_{L''}) \subseteq [1, k] \cap \text{Uci}(\Pi_L) = [i, k]$. Similarly, $\text{Uci}(\Pi_{L'''}) \subseteq [k+1, j]$. The pivot variable v must thus belong to both $\phi_n^{[i,k]}$ and $\phi_n^{[k+1,j]}$; the only such existential variable is t_k. Hence each t_k is used as a pivot in Π_L.
   Since Π is T-regular, and since t_k is used as a pivot to derive L' inside Π_L, it cannot reappear in any line on any path from (including) L' to the final clause. Hence it does not appear in L.
2. Let $i > 1$. By Observation 1, t_{i-1} appears in at least one axiom used in Π_L. Assume to the contrary that $t_{i-1} \notin \text{var}(C)$. Let ρ_C be the minimal partial assignment falsifying C. By assumption, ρ_C does not set t_{i-1}, and by item 1 above, ρ_C does not set any variable t_k with

$i \leq k < j$. Extend ρ_C arbitrarily to all unassigned variables in $(X \cup T) \setminus \{t_{i-1}, \ldots, t_{j-1}\}$ to get ρ_1. Since the merge map M does not depend on variables in T, the partial assignment ρ_1 is sufficient to evaluate M and h. Define the value y as follows:

$$y = \begin{cases} \rho_1(t_j) & \text{if } j \leq n \\ h(\rho_1) & \text{if } j = n+1 \end{cases}$$

For $b \in \{0,1\}$, let ρ_1^b denote the extension of ρ_1 by $t_{i-1} = b$. Exactly one of ρ_1^0, ρ_1^1 satisfies the equation $t_{i-1} + x_i + x_{i+1} + \ldots + x_j + y \equiv 0 \bmod 2$; let this extension be ρ_2. Then there is a unique extension α of ρ_2 to $X \cup T$ such that

- if $j \leq n$, then α satisfies the existential part of all clauses in $\phi_n^{[i,j]}$;
- if $j = n+1$, then $(\alpha, h(\rho_1))$ satisfies all clauses in $\phi_n^{[i,j]}$. (That is, assigning $X \cup T$ according to α and assigning z the value $h(\rho_1)$ satisfies $\phi_n^{[i,j]}$.)

(To find α, work backwards from y to determine the appropriate values of $t_{j-1}, t_{j-2}, \ldots, t_i$ to satisfy $\phi_n^j, \phi_n^{j-1}, \ldots, \phi_n^i$.)
Note that $h(\rho_1) = h(\rho_2) = h(\alpha)$. So $(\alpha, h(\alpha))$ falsifies C (since it extends ρ_C) and satisfies all axiom clauses used to derive L. This contradicts Lemma 4.

3. Let $j \leq n$. Assume to the contrary that $t_j \notin \text{var}(C)$. The argument is identical to that in item 2 (only the indices differ): ρ_C falsifies C; ρ_1 extends it arbitrarily to all unassigned variables in $(X \cup T) \setminus \{t_i, \ldots, t_j\}$; ρ_2 is the extension of ρ_1 obtained by setting t_j so as to satisfy the equation $t_{i-1} + x_i + x_{i+1} + \ldots + x_j + t_j \equiv 0 \bmod 2$; (Here, if $i = 1$, discard t_0 from the equation; i.e. assume $t_0 = 0$); α is the unique extension of ρ_2 to $X \cup T$ satisfying $\phi_n^{[i,j]}$ (To obtain α, work forwards obtaining $t_i, t_{i+1}, \ldots, t_{j-1}$). Now $(\alpha, h(\alpha))$ contradicts Lemma 4.
4. Since $\text{UCI}(\Pi_L) = [i,j]$, variables t_k for $k \notin [i-1, j]$ do not appear in any of the used axioms (Observation 1) and hence do not appear in C. By the preceding three items, $\text{var}(C) \cap T$ does not include any t_k with $k \in [i, j-1]$, includes t_{i-1} whenever $i > 1$, and includes t_j whenever $j < n+1$. The claim follows.
5. Assume to the contrary that for some $k \in [i,j]$, $x_k \notin \text{var}(C) \cup \text{var}(M)$. The argument is similar to that in item 2: ρ_C falsifies C; ρ_1 extends it arbitrarily to all unassigned variables in $(X \setminus \{x_k\}) \cup (T \setminus \{t_i, \ldots, t_{j-1}\})$; y is the value of t_j if $j \leq n$ and the value of h otherwise (since $x_k \notin \text{var}(M)$, ρ_1 is sufficient to evaluate h); ρ_2 is the extension of ρ_1 obtained by setting x_k so as to satisfy the equation $t_{i-1} + x_i + x_{i+1} + \ldots + x_j + y \equiv 0 \bmod 2$; (Here, if $i = 1$, discard t_0 from the equation; i.e. assume $t_0 = 0$); α is the unique extension of ρ_2 to $X \cup T$ satisfying $\phi_n^{[i,j]}$ (To obtain α, work forwards from t_i towards t_{j-1}). Now $(\alpha, h(\alpha))$ contradicts Lemma 4. ◀

▶ **Lemma 17.** *Let $L \in \mathcal{S}$ be derived in Π as $L = res(L', L'', t_k)$. Then* $\text{UCI}(\Pi_L) = [1, n+1]$, *and* $\text{UCI}(\Pi_{L'}), \text{UCI}(\Pi_{L''})$ *partition* $[1, n+1]$ *into* $[1,k], [k+1, n+1]$.

Proof. Since $L \in \mathcal{S}$, L has no variable from T. By Lemma 16(4), $\text{UCI}(\Pi_L) = [1, n+1]$.

Since $L = \text{res}(L', L'', t_k)$, $\text{var}(C') \cap T = \text{var}(C'') \cap T = \{t_k\}$. By Lemma 16(2,3,4), $\text{UCI}(\Pi_{L'}), \text{UCI}(\Pi_{L''}) \in \{[1,k], [k+1, n+1]\}$.

If both $\text{UCI}(\Pi_{L'}), \text{UCI}(\Pi_{L''})$ equal $[k+1, n+1]$, then $\text{UCI}(\Pi_L) = [k+1, n+1]$, contradicting $\text{UCI}(\Pi_l) = [1, n+1]$.

If both $\text{UCI}(\Pi_{L'}), \text{UCI}(\Pi_{L''})$ equal $[1,k]$, then $\text{UCI}(\Pi_L) = [1,k]$. Since t_k is a pivot variable, $k \leq n$, contradicting $\text{UCI}(\Pi_l) = [1, n+1]$.

Hence one each of $\text{UCI}(\Pi_{L'}), \text{UCI}(\Pi_{L''})$ equals $[1,k]$ and $[k+1, n+1]$ as claimed. ◀

▶ **Lemma 18.** *For all $L \in \mathcal{S}$, $width(C) = n$.*

Proof. Let $L \in \mathcal{S}$ be derived in Π as $L = \text{res}(L', L'', t_k)$. Since all axioms create non-trivial strategies, neither M' nor M'' equals $*$. By the rules of MRes, $M' = M'' = M \neq *$. We will show that in fact M must be a constant strategy, $M \in \{0, 1\}$.

By definition of $\mathcal{S}$, $\text{var}(C) \cap T = \emptyset$, and hence $\text{var}(C') \cap T = \text{var}(C'') \cap T = \{t_k\}$. By Lemma 17, $\textsc{Uci}(\Pi_L) = [1, n+1]$ is partitioned by $\textsc{Uci}(\Pi_{L'})$ and $\textsc{Uci}(\Pi_{L''})$ into $[1, k], [k+1, n+1]$.

Assume $\textsc{Uci}(\Pi_{L'}) = [1, k]$, $\textsc{Uci}(\Pi_{L''}) = [k+1, n+1]$; the argument in the other case is identical. Then $\text{var}(M) = \text{var}(M') \subseteq \text{var}(\phi^{[1,k]}) \cap X = \{x_1, \ldots, x_k\}$, and $\text{var}(M) = \text{var}(M'') \subseteq \text{var}(\phi^{[k+1,n+1]}) \cap X = \{x_{k+1}, \ldots, x_n\}$. The only way both these conditions can be satisfied is if $\text{var}(M) = \emptyset$; that is, M is a constant strategy.

Since $\textsc{Uci}(\Pi_L) = [1, n+1]$ and $\text{var}(M) = \emptyset$, Lemma 16(5) implies that $X \subseteq \text{var}(C)$. Therefore $\text{width}(C) = n$. ◀

▶ **Theorem 19.** *Every T-regular refutation of $LQParity_n$ in MRes has size $2^{\Omega(n)}$.*

Proof. Let Π be a T-regular refutation of LQParity_n in MRes. Let $\mathcal{S}', \mathcal{S}$ be as defined in the beginning of this sub-section. By definition, for each $L = (C, M) \in S'$, $\text{var}(C) \subseteq X$. Let $\widehat{\Pi} = \{C \mid L = (C, M) \in S'\}$. Then $\widehat{\Pi}$ contains a propositional resolution refutation of $\mathcal{C} = \{C \mid L = (C, M) \in S\}$. Therefore $\mathcal{C}$ is an unsatisfiable CNF formula over the n variables in X. By Lemma 18, each clause in $\mathcal{C}$ has width n and so is falsified by exactly one assignment. Therefore, to ensure that each of the 2^n assignments falsifies some clause, (at least) 2^n clauses are required. Therefore $|\mathcal{C}| \geqslant 2^n$. Hence $|\Pi| \geqslant 2^n$. ◀

4.2 Completion Principle formulas

Our second hardness result for regular Merge Resolution is for the completion principle formulas, introduced in [24] (cf. Section 2.1).

▶ **Theorem 20.** *Every $(A \cup B)$-regular refutation of CR_n in MRes has size $2^{\Omega(n)}$.*

The proof proceeds as follows: Let Π be a $(A \cup B)$-regular MRes refutation of CR_n. Since every axiom has a variable from $A \cup B$ while the final clause in Π is empty, there is a maximal "component" of the proof leading to and including the final line, where all clauses are $(A \cup B)$-free. The clauses in this component involve only the X variables. We show that the "boundary" of this component is large, by showing in Lemma 21 that each clause here must be wide. (This idea was used in [31] to show that CR is hard for reductionless LD-QRes.)

To establish the width bound, we first note that except for the axioms L_A, L_B, no lines have trivial strategies. Since the pivots at the boundary are variables from $A \cup B$, which are all to the right of z, the merge maps incoming into each boundary resolution must be isomorphic. By analysing what axiom clauses cannot be used to derive lines just above the boundary, we show that many variables are absent in the corresponding merge maps, and invoking soundness of MRes, we show that they must then be present in the boundary clause, making it wide.

Proof. (of Theorem 20) Let Π be an $(A \cup B)$-regular refutation of CR_n (for $n \geq 2$) in MRes.

Define $\mathcal{S}'$ to be the set of those lines in Π where the clause part has no variable from $A \cup B$, and furthermore there is a path in G_Π from the line to the final empty clause via lines where all the clauses also have no variables from $A \cup B$. Let $\mathcal{S}$ denote the set of leaves in the subgraph of G_Π restricted to $\mathcal{S}'$; these are lines that are in $\mathcal{S}'$ but their parents are not in $\mathcal{S}'$. Note that no leaf of Π is in $\mathcal{S}'$ because all leaves of G_Π contain a variable in $A \cup B$.

By definition, for each $L = (C, M^z) \in S'$, $\text{var}(C) \subseteq X$. The sub-derivation $\widehat{\Pi} = \{C \mid \exists L = (C, M^z) \in S'\}$ contains a propositional resolution refutation of the conjunction of clauses $F = \{C \mid \exists L = (C, M^z) \in S\}$. Hence F is an unsatisfiable CNF formula over the n^2 variables in X. We show below, in Lemma 21, that each clause in F has width at least $n-1$. Hence it is falsified by at most $2^{n^2-(n-1)}$ assignments. Therefore, to ensure that each of the 2^{n^2} assignments falsifies some clause, at least 2^{n-1} clauses are required. Therefore $|F| \geqslant 2^{n-1}$. Hence $|\Pi| = 2^{\Omega(n)}$. ◀

▶ **Lemma 21.** *For all $L = (C, M^z) \in S$, width$(C) \geq n-1$.*

Proof. Since $\text{var}(C) \cap (A \cup B) = \emptyset$, L is not a leaf of Π. Say $L = \text{res}(L_1, L_2, v)$ where $L_1 = (C_1, M_1^z)$ and $L_2 = (C_2, M_2^z)$. Since $\text{var}(C_1) \cap (A \cup B) \neq \emptyset$ and $\text{var}(C_2) \cap (A \cup B) \neq \emptyset$, we have $v \in A \cup B$. Consider the case when $v \in A$; the argument for the case when $v \in B$ is symmetrically identical. Without loss of generality, assume that $v = a_n$; and $a_n \in C_1$ and $\overline{a_n} \in C_2$.

Since Π is $(A \cup B)$-regular, a_n does not occur as a pivot in the sub-derivation Π_{L_1}. Therefore $L_A \notin \text{leaves}(G_{\Pi_{L_1}})$ (otherwise $\overline{a_n} \in C_1$, and therefore C_1 would be tautological clause, a contradiction). This implies that the sub-derivation Π_{L_1} cannot use any axiom that contains a positive A literal other than a_n, since such a literal would have to be eliminated by resolution before reaching C_1, requiring the corresponding negated literal, and L_A is the only axiom with negated literals from A. That is, Π_{L_1} does not use any of the axioms A_{ij} for $i \in [n-1]$. The positive literal x_{ij} appears only in A_{ij}. Hence for $i \in [n-1]$, $j \in [n]$, x_{ij} is not a pivot in Π_{L_1} and hence does not appear in M_1^z. On the other hand, M_1^z is not trivial since some A_{nj} clause is used.

C_2 contains $\overline{a_n}$, but no other $\overline{a_i}$. So C_2 is not the axiom L_A. Hence M_2^z is not trivial.

Since the pivot a_n at the step obtaining line L is to the right of z, by the rules of MRes, M_1^z and M_2^z are isomorphic. Hence for each $i \in [n-1]$, and each $j \in [n]$, $x_{ij} \notin \text{var}(M_2^z)$. We claim the following:

▷ Claim 22. Either for all $i \in [n-1]$, C_2 has a variable of the form x_{i*}, or for all $j \in [n]$, C_2 has a variable of the form x_{*j}.

In either case, C_2 has at least $n-1$ variables.

It remains to prove the claim.

Proof. (of Claim) We know that $\overline{a_n} \in C_2$, and for all $i \in [n-1]$, for all $j \in [n]$, $x_{ij} \notin \text{var}(M_2^z)$. Aiming for contradiction, suppose that there exist $i \in [n-1]$ and $j \in [n]$ such that for all $\ell \in [n]$, $x_{i\ell} \notin \text{var}(C_2)$, and for all $k \in [n]$, $\text{var}(x_{kj}) \notin C_2$. Fix such an i, j.

Let ρ be the minimum partial assignment falsifying C_2. Then

- ρ sets $a_n = 1$, leaves all other variables in $A \cup B$ unset.
- ρ does not set any $x_{i\ell}$ or x_{kj}.

For $c \in \{0, 1\}$, extend ρ to α_c as follows: Set $a_i = 0, b_j = 0$, set all other unset variables from $A \cup B$ to 1. Set $x_{ij} = c$. All $x_{i\ell}$ other than x_{ij} set to 1. All x_{kj} other than x_{ij} set to 0. Set remaining variables arbitrarily (but in the same way in α_0 and α_1).

The common part of α_0 and α_1 satisfies all axiom clauses except A_{ij} and B_{ij}, and does not falsify any axiom. The extensions α_c satisfy one more axiom, and still do not falsify the remaining axiom (it has a universal literal z or $\overline{z}$). They both falsify C_2, since they extend ρ.

Since α_0 and α_1 agree everywhere except on x_{ij}, and since $x_{ij} \notin \text{var}(M_2^z)$, it follows that $M_2^z(\alpha_0) = M_2^z(\alpha_1) = d$, say.

By Lemma 4, both (α_0, d) and (α_1, d) should falsify some axiom. However, $(\alpha_{\bar{d}}, d)$ actually satisfies all axioms, a contradiction. ◀

With the claim established, the proof of the lemma is complete. ◀

▶ **Corollary 23.** *Regular MRes is incomparable with the tree-like and general versions of QRes, QURes, CP + ∀Red, ∀Exp + Res, and IR.*

Proof. Let $S \in \{\text{QRes}, \text{QURes}, \text{CP} + \forall\text{Red}, \forall\text{Exp} + \text{Res}, \text{IR}\}$.

The CR_n formulas are easy in tree-like S but hard for regular MRes, so regular MRes does not simulate tree-like or general versions of S.

The Equality formulas are hard for S but easy in regular MRes, so S (and hence also tree-like S) does not simulate regular MRes. ◀

5 A lower bound for Merge Resolution

In this section we turn towards the full system of Merge Resolution and consider the KBKF-lq formulas (cf. Section 2.1). Similarly as the LQParity formulas, these formulas were originally introduced as hard principles for LD-QRes [2]. Here we show that they are hard for the full system of Merge Resolution, thus making it our strongest lower bound in the paper. This constitutes the first lower bound for unrestricted MRes in the literature.

▶ **Theorem 24.** $size_{MRes}(\textit{KBKF-lq}[n]) = 2^{\Omega(n)}$.

Proof idea

We will show that, in any MRes refutation of the KBKF-lq formulas, the literals over the variables in $F = \{f_1, f_2, \ldots, f_n\}$ must be removed before the strategies become 'very complex'. From this we conclude that there must be exponentially many lines.

To argue that literals over F must be removed before the strategies become 'very complex', we look at the form of the lines containing literals over F. If any such line has a 'very complex' strategy (by which we mean that for some $i \in [n]$, u_i depends on either d_i or e_i), then the literals over F cannot be removed from the clause.

Elaborating on the roadmap of the argument: Let Π be an MRes refutation of KBKF-lq$[n]$. Each line in Π has the form $L = (C, M^{x_1}, \ldots, M^{x_n})$ where C is a clause over D, E, F, and each M^{x_i} is a merge map computing a strategy for x_i.

Define $\mathcal{S}'$ to be the set of those lines in Π where the clause part has no F variable and furthermore the line has a path in G_Π to the final empty clause via lines where all the clauses also have no F variables. Let $\mathcal{S}$ denote the set of leaves in the subgraph of G_Π restricted to $\mathcal{S}'$; these are lines that are in $\mathcal{S}'$ but their parents are not in $\mathcal{S}'$. Note that by definition, for each $L = (C, \{M^{x_i} \mid i \in [n]\}) \in \mathcal{S}'$, $\text{var}(C) \subseteq D \cup E$. No line in $\mathcal{S}'$ (and in particular, no line in $\mathcal{S}$) is an axiom since all axiom clauses have variables from F.

Recall that the variables of KBKF-lq$[n]$ can be naturally grouped based on the quantifier prefix: for $i \in [n]$, the ith group has d_i, e_i, x_i, and the $(n+1)$th group has the F variables. By construction, the merge map for x_i does not depend on variables in later groups, as is indeed required for a countermodel. We say that a merge map for x_i has self-dependence if it does depend on d_i and/or e_i.

We show that every merge map at every line in $\mathcal{S}'$ is non-trivial (Lemma 29). Further, we show that at every line on the boundary of $\mathcal{S}'$, i.e. in $\mathcal{S}$, no merge map has self-dependence (Lemma 30). Using this, we conclude that $\mathcal{S}$ must be exponentially large, since in every countermodel the strategy of each variable must have self-dependence (Proposition 2).

In order to show that lines in $\mathcal{S}$ do not have self-dependence, we first establish several properties of the sets of axiom clauses used in a sub-derivation (Lemmas 25, 26, 27, 28).

Detailed proof

For a line $L \in \Pi$, let Π_L be the minimal sub-derivation of L, and let G_{Π_L} be the corresponding subgraph of G_Π with sink L. Let $\textsc{Uci}(\Pi_L) = \{i \in [0, n] \mid \text{leaves}(G_{\Pi_L}) \cap \mathcal{A}_i \neq \emptyset\}$. (Uci stands for UsedConstraintsIndex). Note that we are only looking at the clauses in $\mathcal{A}$ to define Uci.

▶ **Lemma 25.** *For every line $L = (C, \{M^{x_i} \mid i \in [n]\})$ of Π,*

1. $\textsc{Uci}(\Pi_L) = \emptyset$ *if and only if* $C \cap F^1 \neq \emptyset$ *if and only if* $|C \cap F^1| = 1$.
2. $\textsc{Uci}(\Pi_L) \neq \emptyset$ *if and only if* $C \cap F^1 = \emptyset$.

Proof. Since the existential part of each clause in KBKF-lq$[n]$ is a Horn clause, and since the resolvent of Horn clauses is also Horn, $|C \cap F^1| \leq 1$ for each line of Π. It thus suffices to prove that $\forall L \in \Pi,\ \ \textsc{Uci}(\Pi_L) = \emptyset \iff C \cap F^1 \neq \emptyset$.

($\Rightarrow$): For an arbitrary line $L \in \Pi$, suppose $\textsc{Uci}(\Pi_L) = \emptyset$, so L is derived from $\mathcal{B}$. Since $\text{var}_\exists(\mathcal{B}) = F$, $\text{var}(C) \subseteq F$. The existential part of these clauses is strict Horn, and the resolvent of strict Horn clauses is also strict Horn, so C is strict Horn. So $C \cap F^1 \neq \emptyset$.

($\Leftarrow$): The statement $C \cap F^1 \neq \emptyset \Rightarrow \textsc{Uci}(\Pi_L) = \emptyset$ holds at all axioms. Assume to the contrary that it does not hold everywhere in Π. Pick a highest L (closest to the axioms) for which this statement fails. That is, $C \cap F^1 \neq \emptyset$, and $\textsc{Uci}(\Pi_L) \neq \emptyset$. Let L', L'' be the parents of L in Π; by choice of L, both L' and L'' satisfy the statement. Let f_j be the positive literal in C (unique, because C is Horn). Without loss of generality, $f_j \in C'$. Since L' satisfies the statement, $\textsc{Uci}(\Pi_{L'}) = \emptyset$. So $\text{var}(C') \subseteq F$, and since C' is Horn, $C' \setminus \{f_j\} \subseteq F^0$. Since $f_j \in C$, the pivot at this step is not f_j, so it must be an f_k for some $\overline{f_k} \in C'$. So $f_k \in C''$. Since L'' satisfies the statement, $\textsc{Uci}(\Pi_{L''}) = \emptyset$. But then $\textsc{Uci}(\Pi_L) = \textsc{Uci}(\Pi_{L'}) \cup \textsc{Uci}(\Pi_{L''}) = \emptyset$, contradicting our choice of L. Hence our assumption was wrong, and the statement holds for all L in Π. ◀

▶ **Lemma 26.** *A line $L = (C, \{M^{x_i} \mid i \in [n]\})$ of Π with $\textsc{Uci}(\Pi_L) = \emptyset$ has these properties:*

1. *$var(C) \subseteq F$; for all $i \in [n]$, $M^{x_i} \in \{*, 0, 1\}$;*
2. *For some $j \in [n]$, $f_j \in C$ and $M^{x_j} \in \{0, 1\}$;*
3. *For $1 \leq i < j$, $f_i \notin var(C)$ and $M^{x_i} = *$;*
4. *For $j < i \leq n$, if $f_i \notin var(C)$, then $M^{x_j} \in \{0, 1\}$.*

Proof.
1. Since $\textsc{Uci}(\Pi_L) = \emptyset$, $\text{var}(C) \subseteq \text{var}_\exists(\mathcal{B}) = F$.
All pivots in Π_L are from F, and all universal variables are left of F in the quantifier prefix. So no step in Π_L can use the merge operation to update merge maps; all steps in Π_L use only the select operation, which does not create any branching.
2. By Lemma 25, $|C \cap F^1| = 1$, so there is a unique j with the literal $f_j \in C$. This literal appears only in the clauses of $\mathcal{B}_j$, both of which create a non-trivial strategy for x_j. So $M^{x_j} \neq *$. By item (1) proven above, $M^{x_j} \in \{0, 1\}$.
3. Let k be the least index such that Π_L uses an axiom from $\mathcal{B}_k$. Since the positive literal f_j is in C and appears only in $\mathcal{B}_j$, $k \leq j$. Assume $k < j$. The axiom from $\mathcal{B}_k$ introduces the positive literal f_k into Π_L, and by choice of k, no axiom in Π_L has the literal $\overline{f_k}$. Hence f_k cannot be removed by resolution, and so $f_k \in C$, contradicting the fact that C is Horn. So in fact $k = j$. This means that no axiom introduces the variables f_i, $i < j$, into Π_L, so $f_i \notin \text{var}(C)$. Furthermore, amongst all the axioms in $\mathcal{B}$, only the axioms in $\mathcal{B}_i$ have a non-trivial merge map for x_i. Hence for $i < j$, no non-trivial merge map for x_i is created.
4. Since $f_j \in C$, Π_L uses an axiom from $\mathcal{B}_j$. This axiom introduces the literals $\overline{f_i}$, for $j < i \leq n$, into Π_L.

If $\overline{f_i}$ is removed (by resolution) in Π_L, then an axiom from $\mathcal{B}_i$ must be used to introduce the positive literal f_i. This axiom created a non-trivial merge map for x_i, so the merge map for x_i at L is also non-trivial. ◀

▶ **Lemma 27.** *Let $L = (C, \{M^{x_i} \mid i \in [n]\})$ be a line of Π with $\text{UCI}(\Pi_L) \neq \emptyset$. Then $\text{UCI}(\Pi_L)$ is an interval $[a, b]$ for some $0 \leq a \leq b \leq n$. Furthermore, (in the items below, a, b refer to the endpoints of this interval), it has the following properties:*

1. *For $k \in [n] \cap [a, b]$, $M^{x_k} \neq *$.*
2. *If $a \geq 1$, then $|\{d_a, e_a\} \cap C| = 1$. If $a = 0$, then C does not have any positive literal.*
3. *If $b < n$, then $\overline{d_{b+1}}, \overline{e_{b+1}} \in C$.*
4. *For all $k \in [n] \setminus [a, b]$,* (i) *$d_k, e_k \notin var(M^{x_k})$, and* (ii) *if $M^{x_k} = *$ then $\overline{f_k} \in C$.*

Proof. Assume to the contrary that $\text{UCI}(\Pi_L)$ is not an interval. Then there exist $0 \leq a < c < b \leq n$ such that $a, b \in \text{UCI}(\Pi_L)$ but $c \notin \text{UCI}(\Pi_L)$. Let L_1 be the first line in Π_L such that $\text{UCI}(\Pi_{L_1})$ intersects both $[0, c-1]$ and $[c+1, n]$ (note that L_1 exists). Since leaves have singleton UCI sets, L_1 is not a leaf. Say $L_1 = \text{res}(L_2, L_3, v)$. By our choice of L_1, exactly one each of $\text{UCI}(\Pi_{L_2})$ and $\text{UCI}(\Pi_{L_3})$ is a non-empty subset of $[0, c-1]$ and of $[c+1, n]$. So $v \in \text{var}_\exists(\mathcal{A}_{[0,c-1]})$ and $v \in \text{var}_\exists(\mathcal{A}_{[c+1,n]})$. But $\text{var}_\exists(\mathcal{A}_{[0,c-1]}) \cap \text{var}_\exists(\mathcal{A}_{[c+1,n]}) = F$, and by Lemma 25, both C_2 and C_3 contain variables of F only in negated form. So no variable from F can be a resolution pivot, a contradiction. It follows that $\text{UCI}(\Pi_L)$ is an interval.

1. For $k \in [n] \cap [a, b]$, some axiom from $\mathcal{A}_k$ has been used to derive L. Both these axioms create non-trivial strategies for x_k. Subsequent MRes steps cannot make a non-trivial strategy trivial.
2. Consider first the case $a \geq 1$. Since C is a Horn clause, C can contain at most one of the literals d_a, e_a.
   Since $a \in \text{UCI}(\Pi_L)$, at least one of A_a^d, A_a^e appears in leaves(Π_L), so at least one of the literals d_a, e_a is introduced into Π_L. Since A_{a-1}^d and A_{a-1}^e are the only axioms that contain $\overline{d_a}$ or $\overline{d_a}$, and since neither of these is used in Π_L, therefore the positive literals d_a, e_a, if introduced, cannot be removed through resolution. Hence at least one of them is in C. It follows that C has exactly one of d_a, e_a.
   If $a = 0$, Π_L uses the clause A_0 which has only negative literals. The resolvent of such a clause and a Horn clause also has only negative literals. Following the sequence of resolutions on the path from a leaf using A_0 to C shows that C has only negative literals.
3. Since $b < n$ and $b \in \text{UCI}(\Pi_L)$, some clause from $\mathcal{A}_b$ is used in Π_L and introduces the literals $\overline{d_{b+1}}, \overline{e_{b+1}}$ into Π_L. Since $b + 1 \notin \text{UCI}(\Pi_L)$, no leaf of Π_L contains the positive literals d_{b+1}, e_{b+1}. So $\overline{d_{b+1}}$ and $\overline{e_{b+1}}$ cannot be removed through resolution.
4. For $k > b$, no leaf in Π_L contains the positive literals d_k, e_k. For $k < a$, no leaf in Π_L contains the negative literals $\overline{d_k}, \overline{e_k}$. Thus, for $k \notin [a, b]$, the variables d_k, e_k are not used as resolution pivots anywhere in Π_L, and hence are not queried in any of the merge maps. Each negative literal $\overline{f_k}$ is present in every clause of $\mathcal{A}$, and hence is introduced into Π_L. If $M^{x_k} = *$, then $B_k^0, B_k^1 \notin \text{leaves}(\Pi_L)$ (both of them have non-trivial merge maps for x_k). Since these are the only clauses with the positive literal f_k, the literal $\overline{f_k}$ cannot be removed in Π_L; hence $\overline{f_k} \in C$. ◀

▶ **Lemma 28.** *For any line $L = (C, \{M^{x_i} \mid i \in [n]\})$ in Π, and any $k \in [n]$, if $\{d_k, e_k\} \cap var(M^{x_k}) \neq \emptyset$, then $\text{UCI}(\Pi_L) = [a, n]$ for some $a \leq k - 1$.*

Proof. Since $\{d_k, e_k\} \cap \text{var}(M^{x_k}) \neq \emptyset$, either d_k or e_k must be used as a pivot in Π_L, and hence must appear in both polarities in Π_L. The variables d_k, e_k appear positively only in $\mathcal{A}_k$, and negatively only in $\mathcal{A}_{k-1}$. Hence $a \leq k - 1$.

Suppose $b < n$. By Lemma 27 (3), both $\overline{d_{b+1}}$ and $\overline{e_{b+1}}$ are in C. Consider any path ρ in Π from L to the final line $L_\Box$. At every line on this path, the merge map for x_k queries at least one of d_k, e_k since it is at least as complex as the merge map M^{x_k}. Along this path, both d_{b+1} and e_{b+1} must appear as pivots, since the negated literals are eventually removed. Pick the first such step on ρ, and assume without loss of generality that the pivot is d_{b+1} (the other case is symmetric). So $\overline{d_{b+1}}$ is present in the line, say L_1, on ρ, and d_{b+1} is present in the clause L_2 with which it is resolved to obtain $L_3 = \text{res}(L_2, L_1, d_{b+1})$ on ρ. By Lemma 27 (2), $\textsc{Uci}(\Pi_{L_2}) = [b+1, b']$ for some $b' \geq b+1$. Hence by Lemma 27 (4), $d_k, e_k \notin \text{var}((M_2)^{x_k})$. However, $\{d_k, e_k\} \cap \text{var}((M_1)^{x_k}) \neq \emptyset$. Since this resolution on d_{b+1} is not blocked, it must be the case that $(M_2)^{x_k} = *$. Hence, by Lemma 27 (4), $\overline{f_k} \in C_2$ and so $\overline{f_k} \in C_3$. To remove this literal, at some later point along ρ, f_k must appear as pivot. However, at that point, the line from ρ has a complex merge map for x_k, while the line with the positive literal f_k has a non-trivial constant merge map (by Lemma 26 (2)). Hence the resolution on f_k is blocked, a contradiction.

It follows that $b = n$. ◀

▶ **Lemma 29.** *For all $L \in \mathcal{S}'$, for all $k \in [n]$, $M^{x_k} \neq *$.*

Proof. Consider a line $L = (C, \{M^{x_i} \mid i \in [n]\}) \in \mathcal{S}'$. Since $L \in \mathcal{S}'$, $\text{var}(C) \cap F = \emptyset$, so $C \cap F^1 = \emptyset$. By Lemma 25, $\textsc{Uci}(\Pi_L) \neq \emptyset$. Since every clause in $\mathcal{A}$ contains all literals in F^0, for each $k \in [n]$, Π_L has a leaf where the clause contains $\overline{f_k}$. This literal is removed in deriving L, so Π_L also has a leaf where the clause contains the positive literal f_k. That is, it uses an axiom from $\mathcal{B}_k$; this leaf has a non-trivial merge map for x_k. Since a step in MRes cannot make a non-trivial merge map trivial, the merge map for x_k at L is non-trivial. ◀

▶ **Lemma 30.** *For all $L \in \mathcal{S}$, for all $k \in [n]$, $d_k, e_k \notin var(M^{x_k})$.*

Proof. Consider a line $L \in \mathcal{S}$; $L = (C, \{M^{x_i} \mid i \in [n]\})$. Assume to the contrary that for some $k \in [n]$, $\{d_k, e_k\} \cap \text{var}(M^{x_k}) \neq \emptyset$.

Line L is obtained by performing resolution on two non-$\mathcal{S}'$ clauses with a pivot from F. Let $L = \text{res}(L', L'', f_\ell)$ for some $\ell \in [n]$; $f_\ell \in C'$ and $\overline{f_\ell} \in C''$. Since L has no variable in F, f_ℓ is the only variable from F in $\text{var}(C')$ and $\text{var}(C'')$.

Since C' has the literal $f_\ell \in F^1$, by Observation 25, $\textsc{Uci}(\Pi_{L'}) = \emptyset$ and L' is derived exclusively from $\mathcal{B}$. Since $D \cup E$ and $\text{var}(\mathcal{B})$ are disjoint, all the merge maps in L' have no variable from $D \cup E$. So M^{x_k} gets its $D \cup E$ variables from $(M'')^{x_k}$. Since this does not block the resolution step, $(M')^{x_k}$ must be trivial and $M^{x_k} = (M'')^{x_k}$. Since $\text{var}(C') \cap F = f_\ell$, by Lemma 26 (2),(3),(4), $k < \ell$.

The line L'' has no literal from F^1, so by Observation 25, $\textsc{Uci}(\Pi_{L''}) \neq \emptyset$. It has a merge map for x_k involving at least one of d_k, e_k, so by Lemma 28, $\textsc{Uci}(\Pi_{L''}) = [a, n]$ for some $a \leq k-1$. Thus we have $a \leq k-1 < k < \ell \leq n$.

Consider the resolution of L' with L''. By Lemma 26 (2), $(M')^{x_\ell} \in \{0,1\}$, and by Lemma 27 (1), $(M'')^{x_\ell} \neq *$. To enable this resolution, $(M'')^{x_\ell} = (M')^{x_\ell}$. The clauses A^d_ℓ and A^e_ℓ give rise to different constant strategies for x_ℓ. So the derivation of L'' uses exactly one of these two clauses. Assume it uses A^d_ℓ; the other case is symmetric. Since $a < \ell$, the derivation of L'' uses a clause from $A_{\ell-1}$, introducing literals $\overline{d_\ell}$ and $\overline{e_\ell}$. Since the only clause containing positive literal e_ℓ is not used, $\overline{e_\ell}$ survives in C''. Going from L'' to L removes only $\overline{f_\ell}$, so $\overline{e_\ell} \in C$.

To summarize, at this stage we know that $L \in \mathcal{S}$, $\overline{e_\ell} \in C$, $\{d_k, e_k\} \cap \text{var}(M^{x_k}) \neq \emptyset$, $M^{x_\ell} \in \{0,1\}$ and $1 \leq k < \ell \leq n$.

Fix any path ρ in G_Π from L to $L_\Box$. Along this path, e_ℓ appears as the pivot somewhere, since the literal $\overline{e_\ell}$ is eventually removed. Consider the resolution step at that point, say $C_1 = \text{res}(C_2, C_3, e_\ell)$, with C_3 being the clause at the line on ρ. At the corresponding line L_3, the strategies are at least as complex as those at L. Hence $\text{var}(M_3^{x_k}) \cap \{d_k, e_k\} \neq \emptyset$. On the other hand, C_2 has the positive literal e_ℓ. By Lemma 27, for the corresponding line L_2, $\text{UCI}(\Pi_{L_2}) = [\ell, c]$ for some $c \geq \ell$. Since $k < \ell$, by Lemma 27, $\{d_k, e_k\} \cap \text{var}(M_2^{x_k}) = \emptyset$. However, the path from L_2 to L_1 and thence to $L_\Box$ along ρ witnesses that $L_2 \in \mathcal{S}'$, so by Lemma 29, $(M_2)^{x_k} \neq *$. Thus $M_2^{x_k}$ and $M_3^{x_k}$ are non-trivial but not isomorphic, and this blocks the resolution on e_ℓ.

Thus our assumption that $\{d_k, e_k\} \cap \text{var}(M^{x_k}) \neq \emptyset$ must be false. The lemma is proved. ◀

Proof. (of Theorem 24) Let Π be a refutation of KBKF-lq$[n]$ in MRes. Let $\mathcal{S}', \mathcal{S}$ be as defined in the beginning of this section. Let the final line of Π be $L_\Box = (\Box, \{s^{x_i} \mid i \in [n]\})$, and for $i \in [n]$, let h_i be the functions computed by the merge map s^{x_i}. By soundness of MRes, the functions $\{h_i\}_{i \in [n]}$ form a countermodel for KBKF-lq$[n]$.

For each $a \in \{0,1\}^n$, consider the assignment α to the variables of $D \cup E$ where $d_i = a_i$, $e_i = \overline{a_i}$. Call such an assignment an anti-symmetric assignment. Given such an assignment, walk from $L_\Box$ towards the leaves of Π as far as is possible while maintaining the following invariant at each line $L = (C, \{M^{x_i} \mid i \in [n]\})$ along the way:

1. α falsifies C, and
2. for each $i \in [n]$, $h_i(\alpha) = M^{x_i}(\alpha)$.

Clearly this invariant is initially true at $L_\Box$, which is in $\mathcal{S}'$. If we are currently at a line $L \in \mathcal{S}'$ where the invariant is true, and if $L \notin \mathcal{S}$, then L is obtained from lines L', L''. The resolution pivot in this step is not in F, since that would put L in $\mathcal{S}$. So both L' and L'' are in $\mathcal{S}'$, and the pivot is in $D \cup E$. Let the pivot be in $\{d_\ell, e_\ell\}$ for some $\ell \in [n]$. Depending on the pivot value, exactly one of C', C'' is falsified by α; say C' is falsified. By Lemma 29, for each $i \in [n]$, both $(M')^{x_i}$ and $(M'')^{x_i}$ are non-trivial. By definition of the MRes rule,

- For $i < \ell$, $(M')^{x_i}$ and $(M'')^{x_i}$ are isomorphic (otherwise the resolution is blocked), and $M^{x_i} = (M')^{x_i} = (M'')^{x_i}$.
- For $i \geq \ell$, there are two possibilities:
  (1) $(M')^{x_i}$ and $(M'')^{x_i}$ are isomorphic, and $M^{x_i} = (M')^{x_i}$.
  (2) M^{x_i} is a merge of $(M')^{x_i}$ and $(M'')^{x_i}$ with the pivot variable queried. By definition of the merge operation, since C' is falsified by α, $M^{x_i}(\alpha) = (M')^{x_i}(\alpha)$.

Thus in all cases, for each i, $h_i(\alpha) = M^{x_i}(\alpha) = (M')^{x_i}(\alpha)$. Hence L' satisfies the invariant.

We have shown that as long as we have not encountered a line in $\mathcal{S}$, we can move further. We continue the walk until a line in $\mathcal{S}$ is reached. We denote the line so reached by $P(\alpha)$. Thus P defines a map from anti-symmetric assignments to $\mathcal{S}$.

Suppose $P(\alpha) = P(\beta) = (C, \{M^{x_i} \mid i \in [n]\})$ for two distinct anti-symmetric assignments obtained from $a, b \in \{0,1\}^n$ respectively. Let j be the least index in $[n]$ where $a_j \neq b_j$. By Lemma 30, M^{x_j} depends only on $\{d_i, e_i \mid i < j\}$, and α, β agree on these variables. Thus we get the equalities $a_j = h_j(\alpha) = M^{x_j}(\alpha) = M^{x_j}(\beta) = h_j(\beta) = b_j$, where the first and last equalities follow from Proposition 2, the third equality from by Lemma 30 and choice of j, and the second and fourth equalities by the invariant satisfied at $P(\alpha)$ and $P(\beta)$ respectively. This contradicts $a_j \neq b_j$.

We have established that the map P is one-to-one. Hence, $\mathcal{S}$ has at least as many lines as anti-symmetric assignments, so $|\Pi| \geq |\mathcal{S}| \geq 2^n$. ◀

▶ **Corollary 31.** *MRes is incomparable with QURes and CP + ∀Red.*

Proof. Theorem 24 shows that the KBKF-lq[n] formula requires exponential-size refutations in MRes. It has polynomial-size refutations in QURes [2], and also in CP + ∀Red (since CP + ∀Red simulates QURes [13]). The other direction follows from the Equality formulas, as already mentioned in the proofs of Corollaries 12, 23. ◀

6 Conclusions and Future Work

The proof system MRes was introduced in [6], using the novel idea of building strategies directly into the proof and using them to enable additional sound applications of resolution. In [6], the strengths of the proof system were demonstrated. In this paper, we complement that study by exposing some limitations of MRes. We obtain hardness for tree-like MRes by transferring computational hardness of the countermodels in decision trees, and for regular and general MRes by ad hoc combinatorial arguments.

Several questions still remain.

1. One of the driving goals behind the definition of MRes was overcoming a perceived weakness of LD-QRes: its criterion for blocking unsound applications of resolution also blocks several sound applications. However, whether MRes actually overcomes this weakness is yet to be demonstrated. In [6], MRes is shown to be more powerful than the reductionless variant of LD-QRes (introduced in [16] and further investigated in [6, 31]). However, we still do not have an instance of a formula hard for LD-QRes but easy for MRes. A natural candidate is LQParity, for which we only have a lower bound in regular MRes. Another natural candidate is SquaredEquality. The other direction, whether there is a formula easy for LD-QRes but hard for MRes, is also open. One possible candidate for this separation might appear to be KBKF, which is easy for LD-QRes [21] (that paper uses the name φ_t). However the KBKF formulas can be shown to have short refutations in MRes as well, and hence cannot be used for this purpose.
2. In the propositional case, regular resolution simulates tree-like resolution. This relation may not hold in the case of MRes, and even if it does, it will need a different proof. The trick used in the propositional case – (i) interpret the proof tree as a decision tree for search, (ii) make the decision tree read-once, (iii) then return from the search tree to a refutation, – does not work here because when we prune away parts of the decision tree to get a read-once tree, we may end up destroying isomorphism of strategies of blocking variables.

References

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

2 Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In Carsten Sinz and Uwe Egly, editors, *Theory and Applications of Satisfiability Testing – SAT 2014*, pages 154–169, Cham, 2014. Springer International Publishing.

3 Olaf Beyersdorff and Joshua Blinkhorn. Formulas with large weight: a new technique for genuine QBF lower bounds. *Electron. Colloquium Comput. Complex.*, 24:32, 2017.

4 Olaf Beyersdorff and Joshua Blinkhorn. Lower bound techniques for QBF expansion. *Theory of Computing Systems*, 64(3):400–421, 2020. doi:10.1007/s00224-019-09940-0.

5 Olaf Beyersdorff, Joshua Blinkhorn, and Luke Hinde. Size, Cost, and Capacity: A Semantic Technique for Hard Random QBFs. *Logical Methods in Computer Science*, Volume 15, Issue 1, February 2019. doi:10.23638/LMCS-15(1:13)2019.

6 Olaf Beyersdorff, Joshua Blinkhorn, and Meena Mahajan. Building Strategies into QBF Proofs. *Journal of Automated Reasoning*, 2020. Preliminary version in 36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019). `doi:10.1007/s10817-020-09560-1`.

7 Olaf Beyersdorff, Joshua Blinkhorn, and Meena Mahajan. Hardness characterisations and size-width lower bounds for QBF resolution. In *Proc. ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 209–223. ACM, 2020.

8 Olaf Beyersdorff, Joshua Blinkhorn, Meena Mahajan, Tomáš Peitl, and Gaurav Sood. Hard QBFs for Merge Resolution. In *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*, volume 182 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:15, 2020. `doi:10.4230/LIPIcs.FSTTCS.2020.12`.

9 Olaf Beyersdorff, Ilario Bonacina, and Leroy Chew. Lower bounds: From circuits to QBF proof systems. In *ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 249–260, 2016.

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

11 Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. New Resolution-Based QBF Calculi and Their Proof Complexity. *ACM Trans. Comput. Theory*, 11(4), September 2019. `doi:10.1145/3352155`.

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

13 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Understanding cutting planes for QBFs. *Information and Computation*, 262:141–161, 2018.

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

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

16 Nikolaj Bjørner, Mikolás Janota, and William Klieber. On conflicts and strategies in QBF. In Ansgar Fehnker, Annabelle McIver, Geoff Sutcliffe, and Andrei Voronkov, editors, *20th International Conferences on Logic for Programming, Artificial Intelligence and Reasoning LPAR 2015*, volume 35 of *EPiC Series in Computing*, pages 28–41. EasyChair, 2015.

17 A. Blake. *Canonical expressions in Boolean algebra.* PhD thesis, University of Chicago, 1937.

18 Beate Bollig and Ingo Wegener. A very simple function that requires exponential size read-once branching programs. *Information Processing Letters*, 66(2):53 – 57, 1998. `doi:https://doi.org/10.1016/S0020-0190(98)00042-8`.

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

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

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 Marijn Heule, Martina Seidl, and Armin Biere. A unified proof system for QBF preprocessing. In *IJCAR*, pages 91–106, 2014.

23 Mikoláš Janota. On Q-Resolution and CDCL QBF solving. In Nadia Creignou and Daniel Le Berre, editors, *Theory and Applications of Satisfiability Testing – SAT 2016*, pages 402–418, Cham, 2016. Springer International Publishing.

24 Mikoláš Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theoretical Computer Science*, 577:25 – 42, 2015. `doi:https://doi.org/10.1016/j.tcs.2015.01.048`.

25 Manuel Kauers and Martina Seidl. Short proofs for some symmetric quantified Boolean formulas. *Inf. Process. Lett.*, 140:4–7, 2018.

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

27 Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, volume 60 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, 1995.

28 Florian Lonsing, Uwe Egly, and Martina Seidl. Q-resolution with generalized axioms. In Nadia Creignou and Daniel Le Berre, editors, *Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Proceedings*, volume 9710 of *Lecture Notes in Computer Science*, pages 435–452. Springer, 2016.

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

30 Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Long-distance Q-resolution with dependency schemes. *J. Autom. Reasoning*, 63(1):127–155, 2019.

31 Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Proof complexity of fragments of long-distance Q-resolution. In Mikolás Janota and Inês Lynce, editors, *Theory and Applications of Satisfiability Testing - SAT 2019 - 22nd International Conference, SAT, Proceedings*, volume 11628 of *Lecture Notes in Computer Science*, pages 319–335. Springer, 2019.

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

33 John Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12:23–41, 1965.

34 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.

35 Friedrich Slivovsky and Stefan Szeider. Soundness of Q-resolution with dependency schemes. *Theoretical Computer Science*, 612:83–101, 2016.

36 G. S. Tseitin. On the complexity of derivation in propositional calculus. In Jörg H. Siekmann and Graham Wrightson, editors, *Automation of Reasoning: 2: Classical Papers on Computational Logic 1967–1970*, pages 466–483. Springer Berlin Heidelberg, Berlin, Heidelberg, 1983. doi:10.1007/978-3-642-81955-1_28.

37 Allen Van Gelder. Contributions to the theory of practical quantified Boolean formula solving. In *Proc. Principles and Practice of Constraint Programming (CP'12)*, pages 647–663, 2012.

38 Ingo Wegener. *Branching Programs and Binary Decision Diagrams*. Society for Industrial and Applied Mathematics, 2000. doi:10.1137/1.9780898719789.

39 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.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il