Extending Merge Resolution to a Family of Proof Systems

Sravanthi Chede and Anil Shukla

Department of Computer Science and Engineering, IIT Ropar, India
{sravanthi.20csz0001,anilshukla}@iitrpr.ac.in

Abstract. Merge Resolution (MRes [4]) is a recently introduced proof system for false QBFs. Unlike other known QBF proof systems, it builds winning strategies for the universal player within the proofs. Every line of this proof system consists of existential clauses along with countermodels. MRes stores the countermodels as merge maps. Merge maps are deterministic branching programs in which isomorphism checking is efficient as a result MRes is a polynomial time verifiable proof system.

In this paper, we introduce a family of proof systems MRes-\(\mathcal{R}\) in which, the information of countermodels are stored in any pre-fixed complete representation \(\mathcal{R}\), instead of merge maps. Hence corresponding to each possible complete representation \(\mathcal{R}\), we have a sound and refutationally complete QBF-proof system in MRes-\(\mathcal{R}\). To handle arbitrary representations for the strategies, we introduce consistency checking rules in MRes-\(\mathcal{R}\) instead of isomorphism checking in MRes. As a result these proof systems are not polynomial time verifiable. Consequently, the paper shows that using merge maps is too restrictive and can be replaced with arbitrary representations leading to several interesting proof systems.

The paper also studies proof theoretic properties of the family of new proof systems MRes-\(\mathcal{R}\). We show that eFrege+ϕred simulates all valid refutations from proof systems in MRes-\(\mathcal{R}\). Since proof systems in MRes-\(\mathcal{R}\) may use arbitrary representations, in order to simulate them, we first represent the steps used by the proof systems as a new simple complete structure. As a consequence, the corresponding proof system belonging to MRes-\(\mathcal{R}\) is able to simulate all proof systems in MRes-\(\mathcal{R}\). Finally, we simulate this proof system via eFrege+ϕred using the ideas from [10].

On the lower bound side, we lift the lower bound result of regular MRes ([5]) for all regular proof systems in MRes-\(\mathcal{R}\). To be precise, we show that the completion principle formulas from [17] which are shown to be hard for regular MRes in [5], are also hard for any regular proof system in MRes-\(\mathcal{R}\). Thereby, the paper lifts the lower bound of regular MRes to an entire class of proof systems, which use some complete representation, including those undiscovered, instead of merge maps.

1 Introduction

Proof complexity is a sub-branch of computational complexity in which the main focus is to understand the complexity of proving (refuting) theorems (contradictions) in various proof systems. Informally, a proof system is a polynomial time
computable function which maps proofs to theorems. Several propositional proof systems like resolution [23], Cutting planes [13], and Frege [15] have been developed for proving (refuting) propositional formulas. The relative strength of these proof systems has been well studied [24]. In the literature, several proof systems which are not polynomial time computable (verifiable) have also been well studied. For example, semantic cutting planes [14].

Quantified Boolean formulas (QBFs) extend propositional logic by adding quantifications $\exists$ (there exists) and $\forall$ (for all) to the variables. Several QBF proof systems like Q-Res [19], QU-Res [26], LD-Q-Res [2], $\forall$Exp+Res [18], IR-calc, and IRM-calc [7] have been proposed in the literature. These are all either CDCL (Conflict Driven Clause Learning)-based, or expansion-based QBF proof systems. Cutting planes proof systems has also been extended for QBFs ($\text{CP+}^\forall\text{red}$) [8].

A new proof system Merge resolution (MRes) [4] has been developed recently. It follows a different QBF-solving approach. In MRes, winning strategies for the universal player are explicitly represented within the proof in the form of deterministic branching programs, known as merge maps [4]. MRes builds partial strategies at each line of the proof such that the strategy at the last line (corresponding to the empty clause) forms the complete countermodel for the input QBF. As a result, MRes admits strategy extraction by design. Before applying the refutation rules, MRes needs to check the partial strategies of the hypothesis to be isomorphic. Note that the isomorphism checking in ‘merge maps’ is efficient, hence MRes is a polynomial time verifiable proof system.

In this paper, we extend MRes to a family of sound and refutationally complete QBF proof systems MRes-$\mathcal{R}$. We observe that the representation of strategies in the proofs as merge maps is not relevant for the soundness and completeness of the proof system. Strategies can be depicted by any complete representation and by slightly modifying the refutation rules to include arbitrary complete representations, the soundness and completeness of the proof system remains intact. To be precise, we change the isomorphism checking rule in MRes to ‘consistency’ checking (Section 3.1) defined for Dependency Quantified Boolean Formulas (DQBFs) in [9]. This leads to the definition of a new proof system (Say $\mathcal{P}$) for each complete representation. All these new proof systems together form the family of proof systems denoted by MRes-$\mathcal{R}$. However, due to the consistency checking rules, the proof systems in MRes-$\mathcal{R}$ are not polynomial time verifiable. In literature, many interesting non-polynomial time verifiable proof systems have been studied, for example, semantic cutting planes for QBFs (SemCP+$\forall$red) [8]. Because of the introduction of such powerful consistency checking rules, proof systems in MRes-$\mathcal{R}$ allow a few forbidden resolution steps of MRes (ref. Example 2).

The paper also studies in detail the strength of these new proof systems. We show that eFrege+$\forall$red is powerful enough to simulate valid refutations of the proof systems in MRes-$\mathcal{R}$. Since these systems admit strategy extraction by design, we show the said simulations by using the ideas from [10]. Furthermore, the paper lifts the lower bound results from [5] of regular MRes to every regular proof system $\mathcal{P} \in \text{MRes-} \mathcal{R}$. We explain our contributions in detail in the following section.
1.1 Our Contributions

1. **Introducing a new family of proof systems MRes-R:** MRes [4] uses merge maps to store the countermodels within proofs. We observe that merge maps are not important for the soundness and completeness of the proof system. They just make the proof system polynomial time verifiable. However, at the same time they are too restrictive. In this paper, we propose a family of proof systems MRes-R, one for each arbitrary complete representation of strategies into proofs (instead of merge maps). In order to make these proof systems sound and complete, we only need to modify the rules of MRes slightly. To be precise, we check the consistency relation instead of isomorphism among the strategies before applying the resolution rules (ref. Section 3). Although, this modification makes the proof systems not polynomial time verifiable; however, makes them very interesting, since the representations of strategies can be arbitrary. We only need that the representations be complete, in the sense that every finite function has at least one representation in the same. We need this for proving the completeness of our proposed proof systems (in Claim 7).

To be precise, for proving completeness of MRes-R, we consider the MRes-M proof system in MRes-R which uses merge maps as the representation for strategies. We then prove that MRes-M system p-simulates the MRes proof system (which is known to be complete) by showing that every rule of MRes is also valid in MRes-M (Theorem 6). We then show how any MRes-M-proof can be (non-efficiently) converted to P-proof for any $P \in MRes-R$ (Claim 7).

The soundness proof of MRes-R follows from proving that every line of the proof gives a partial falsifying strategy for the universal player.

2. **Proving a lower bound for Regular MRes-R:** The Completion Formulas $CR_n$ were first introduced in [17], to show that level-ordered Q-Res cannot p-simulate $\forall\exp+\res$. They were also used to show that level-ordered $Q\res$ cannot p-simulate tree-like $Q\res$ [22]. It has been shown recently in [5], that $CR_n$ formulas are even hard for regular MRes. In this paper, we lift this lower bound of the Completion Formulas $CR_n$ to all regular proof systems in MRes-R.

That is, we show that any regular proof system $P \in MRes-R$, takes exponential time to refute the $CR_n$ formulas.

For this lower bound proof, we mostly follow the proof from [5, Theorem 9] where they have used the fact that most of the clauses in the MRes proof are going to be free of all the literals from right (in quantifier prefix) of the only universal variable $z$ of $CR_n$. They showed that the number of such clauses are exponential in $n$ proving the required lower bound.

We established the similar argument for every regular proof system in MRes-R. In [5, Theorem 9], the major part of the proof relied on the fact that MRes uses isomorphism so they could rule out the variables not in one hypothesis merge map of a resolution step, as also not to be present in the other. However, this is not the case in MRes-R. So we provide a new Claim (ref. Claim 14) that even though MRes-R insists on consistency rather than isomorphism, the above property holds. That is, the clauses in CR, make it such that the variables not in
one hypothesis strategy cannot be present in the other as well when consistency is maintained in the resolution steps.

3. **eFrege+∀red simulates MRes-R**: We show that eFrege+∀red simulates valid refutations in every proof system belonging to MRes-R. Since proof systems in MRes-R can use arbitrary representations, simulating the same is difficult even for powerful proof systems. However, MRes-R family uses simple rules for refuting, which can even be detected by just observing the clauses of the lines, without exploring the representation parts. If one can come up with a complete representation which can represent the rules performed by any MRes-R proof system, then one can show the required simulation.

![Diagram](image)

**Fig. 1**: Various QBF proof systems and efficient simulations. New results and proof systems are written in bold. MRes-T belongs to MRes-R. Regular MRes-R are below the 'known lower bound' dashed line, due to Theorem 16. MRes-R p-simulates MRes due to Proposition 9. For the simulations of MRes, IRM-calc, and LQU+-Res by eFrege+∀red, and other known simulations refer [10, Fig. 1].

We use this observation and define a new complete representation, denoted as the T-representation (ref. Section 5.1). To handle all the MRes-R rules, we came up with this hybrid representation consisting of both circuits and branching programs. It is capable of representing all the refutation rules allowed in any MRes-R proof. To achieve this we introduce a new type of node, namely ‘#’ node, which deals with the new consistency checking property (ref. Fig 2a). We also show that T representation is a complete representation, therefore the corresponding MRes-T proof system (5.1) belongs to MRes-R.
The idea of the simulation is to convert every valid \( P \)-proof (\( P \in \text{MRes-} \mathcal{R} \)) efficiently into an \( \text{MRes-} \mathcal{T} \)-proof as explained in Theorem 8. Then, as \( \text{MRes-} \mathcal{T} \) admits strategy extraction by definition, we use the idea from [10] to simulate the valid \( \text{MRes-} \mathcal{T} \)-proof in \( \text{eFrege+} \forall \text{red} \). Thereby proving that \( \text{eFrege+} \forall \text{red} \) simulates any valid refutations from proof system in \( \text{MRes-} \mathcal{R} \).

This simulation result is a way forward towards uniform certification in QBF [10]. In [10], they used a very distinguished technique that allows \( \text{eFrege+} \forall \text{red} \) to simulate few proof systems that admit strategy extraction. Using this technique, they showed that \( \text{eFrege+} \forall \text{red} \) can simulate proof systems \( \text{MRes}, \text{IRM-calc} \) and \( \text{LQU}^+ \)-\text{Res}. We use the same technique and prove that \( \text{eFrege+} \forall \text{red} \) even simulates the family of proof systems \( \text{MRes-} \mathcal{R} \).

Observe that the proposed \( \text{eFrege+} \forall \text{red} \) simulation algorithm (Section 5) of the proof systems \( P \) in \( \text{MRes-} \mathcal{R} \) assumes that the given \( P \)-refutations are valid. As a result one cannot use this simulation algorithm to efficiently verify the correctness of the given \( P \)-refutations. That is, even if the resulting \( \text{eFrege+} \forall \text{red} \) proof is valid (which is efficiently verifiable), one cannot infer if or not the initial \( P \)-refutation is valid. However, the proposed simulation algorithm always produces a valid \( \text{eFrege+} \forall \text{red} \) proof for a given valid \( P \) proof.

Further, for the current simulation order among QBF proof systems refer Fig 1.

1.2 Organization of the paper

We present important notations and preliminaries used in this paper in Section 2. In Section 3, we present the new family of proof systems \( \text{MRes-} \mathcal{R} \). We prove the soundness and refutational completeness of proof systems in \( \text{MRes-} \mathcal{R} \) in Section 4. In Section 5, we show that \( \text{eFrege+} \forall \text{red} \) simulates proof systems in \( \text{MRes-} \mathcal{R} \). We establish the lower bound results for every regular proof system in \( \text{MRes-} \mathcal{R} \) in Section 6. Finally, we conclude and present a few open problems in Section 7.

2 Notations and Preliminaries

For a Boolean variable \( x \), its literals can be \( x \) and \( \overline{x} \). A clause \( C \) is a disjunction of literals and a conjunctive normal form (CNF) \( F \) is a conjunction of clauses. We denote the empty clause by \( \bot \). \( \text{vars}(C) \) is a set of all variables in \( C \) and \( \text{width}(C) = |\text{vars}(C)| \).

A proof system [11] for a non-empty language \( L \subseteq \{0,1\}^* \) is a polynomial time computable function \( f : \{0,1\}^* \rightarrow \{0,1\}^* \) such that \( \text{Range}(f) = L \). For string \( x \in L \), we say a string \( w \in \{0,1\}^* \) is an \( f \)-proof of \( x \) if \( f(w) = x \). A proof system \( f \) for \( L \) is complete if and only if for every \( x \in L \) we have a corresponding \( f \)-proof for \( x \). A proof system \( f \) for \( L \) is sound if and only if the existence of an \( f \)-proof for \( x \) implies that \( x \in L \). Informally, a proof system is a function \( f \) which maps proofs to theorems (or contradictions).

A proof system \( f \) p-simulates (polynomially simulates) another proof system \( g \), if every \( g \)-proof of input \( x \) can be translated into an \( f \)-proof for the same input in polynomial time w.r.t the size of the \( g \)-proof. We denote this as \( f \leq_p g \). Proof
systems for $L = \text{FQBFs/ TQBFs}$ are said to be QBF proof systems where, FQBFs (TQBFs) denote the set of all false (true) QBFs.

**Quantified Boolean formulas:** QBFs are an extension of the propositional Boolean formulas where each variable is quantified with one of $\{\exists, \forall\}$, the symbols having their general semantic definition of existential and universal quantifier respectively.

In this paper, we assume that QBFs are in closed prenex form with CNF matrix i.e., we consider the form $Q_1 X_1 \ldots Q_k X_k. \phi(X)$, where $X_i$ are pairwise disjoint sets of variables; $Q_i \in \{\exists, \forall\}$ and $Q_i \neq Q_{i+1}$, and $\phi(X)$ is in CNF form over variables $X = X_1 \cup \cdots \cup X_k$, called the matrix of the QBF. We denote QBFs as $F := Q.\phi$ in this paper, where $Q$ is the quantifier prefix. If $x \in X_i$ then we denote $Q(x)$ to be equal to $Q_i$. For a variable $x$ if $Q(x) = \exists$ (resp. $Q(x) = \forall$), we call $x$ an existential (resp. universal) variable. If a variable $x$ is in the set $X_i$, any $y \in X_j$ where $j < i$, we say that $y$ occurs to the left of $x$ in the quantifier prefix and write $y \leq_Q x$. On the other hand, if $j > i$ we say that $y$ occurs to the right of $x$ in the quantifier prefix and write $y \geq_Q x$. The set of existential variables to the left of a universal variable $u$ will be denoted by $L_Q(u)$ in this paper.

Let $C \in \phi$ and $Q(u) = \forall$, then the ‘falsifying $u$-literal’ is defined to be 0 if $u \in C$, and 1 if $u \in C$ and * if $u \notin \text{vars}(C)$. Also, the existential subclause of $C$ is the clause formed by only the existential literals from $C$.

If $S$ is any set of variables, a complete assignment of $S$ will be an assignment which assigns every variable in $S$ to either 1 or 0. Similarly, a partial assignment is an assignment which assigns a subset of variables in $S$ to either 1 or 0. Note that the $\text{vars}(S)$ that have not been assigned to 1 or 0 in a particular partial assignment of $S$ are denoted as having an assignment of ‘*’. We denote $\langle S \rangle$ as the set of all possible complete assignments of $S$ and $\langle\langle S \rangle\rangle$ as the set of all possible partial assignments of $S$.

**QBFs as a game:** [1] Given a QBF $F = \exists X_1 \forall X_2 \ldots \exists X_n.\phi$ we may view it as game between universal and existential player. The rules of the game being that according to the quantification sequence the players assign values to the sets $X_i$s alternatively. At the end, when substituting the complete assignment to all variables in $\phi$, if $\phi$ evaluates to 1 (resp. 0) the existential (resp. universal) player wins.

For a QBF $Q.\phi$, a strategy of universal player is a decision function that returns the assignment to all universal variables of $Q$, where the decision for each $u$ depends only on the variables in $L_Q(u)$. If $H^u$ is the strategy for the universal variable $u$ then, $\text{vars}(H^u)$ is the subset of existential variables from $L_Q(u)$ which are actually used in building the strategy $H^u$.

**Winning strategy** for the universal player is a strategy which gives an assignment to all universal variables of the given QBF for every possible assignment of existential variables, such that the substitution of this complete assignment falsifies the QBF. Winning strategy of the universal player is also called a countermodel in case of a false QBF. A QBF is false iff there exists a winning strategy for the universal player [1].
We say that a QBF proof system \( f \) admits **strategy extraction** if for any given valid \( f \)-proof of a false QBF \( F \), one can compute a winning strategy for the universal player in the time polynomial to the size of the \( f \)-proof.

As said earlier, strategies are basically decision functions. For the portrayal of the same, many representations can be used like truth tables, directed acyclic graphs (DAGs), merge maps, etc. A complete representation is the one in which every possible finite decision function can be represented.

**Resolution** [23] is the most studied redundancy rule in both SAT and QBF worlds, we define the same below:

\[
\frac{C \lor x}{C \lor D}, \quad \frac{D \lor \overline{\sigma}}{C \lor \overline{\sigma}},
\]

where \( C, D \) are clauses and \( x \) is the pivot variable. The clause \( (C \lor D) \) is called the resolvent. We denote this step as ‘\( \text{res}(C \lor x, D \lor \overline{\sigma}, x) \)’ throughout the paper.

Next, we define a few QBF proof systems that we require in this paper.

**Q-Res** [19] is one of a basic QBF proof system. It is an extension of the resolution proof system for QBFs. It allows the resolution rule defined above with the pivot variable being existential. For dealing with the universal variables, it defines a ‘universal reduction’ rule as follows:

The **Universal Reduction** (UR) rule of **Q-Res** allows dropping of a universal variable \( u \) from a clause \( C \) in the QBF, provided no existential variable \( x \in C \) appears to the right of \( u \) in the quantifier prefix.

### 2.1 MRes

**MRes** is a proof system for false QBFs introduced in [4]. We describe **MRes** briefly in this section, please refer to [4] for its formal definition.

For a false QBF \( Q.\phi \), an **MRes** refutation will be a sequence of lines of the form \( L_i = (C_i, \{M_u^i\}) \); where \( C_i \) is a clause consisting of only existential literals and \( \{M_u^i\} \) is the set of merge maps of each universal variable \( u \in Q \).

The **Merge maps** represent the partial strategies for each universal variable at any line. The merge map \( M_u^n \) is a decision branching graph with definite strategies \( \{0, 1, \ast\} \) at the leaves nodes (\( \ast \) is used when no strategy for \( u \) exists till that line). The intermediate nodes of merge map \( M_u^n \) branch on some existential variable (say \( x \)) \in L_Q(u). That is, if \( L_i = \text{res}(L_a, L_b, x) \) for some \( a, b < i \), then \( M_u^n \) will get connected to \( M_u^a \) with an edge label of \( \overline{x} \) and to \( M_u^b \) with an edge label of \( x \).

An important property used in **MRes** refutation rules is defined below:

**Isomorphism:** Two merge maps are isomorphic if and only if there exists a bijection mapping from the nodes of one to that of another. In other words, two isomorphic merge maps represent exactly the same strategy.

Two operations needed for **MRes** refutations are defined below:

**Select** operation is defined on two merge maps. If they are isomorphic, then it outputs one of them. Or, if one of them is trivial (i.e. \( \ast \)), then it outputs the other.

**Merge**\( (M_u^n, M_v^m, n, x) \) operation is defined when \( a, b < n \), and it returns a new merge map where the new root node is connected to \( M_u^n \) with \( \overline{x} \) and to \( M_v^m \) with \( x \). Also, if any common line nodes exist in \( M_u^n, M_v^m \), it merges them into a single node.
Now we define the MRes proof system:
For a false QBF $Q.\phi$, the MRes proof $\Pi := L_1, L_2, ..., L_m$ where every line $L_i := (C_i, \{M^u_i : \text{for every universal variable } u \in Q\})$ is derived using either an ‘Axiom’ step or a ‘Resolution’ step. In the axiom step, $C_i$ will be the existential subclause of some $C \in \phi$ and every $M^u_i$ will be a leaf node with the falsifying $u$-literal of $C$. In the resolution step, $C_i$ is obtained from $\text{res}(C_a, C_b, x)$ where $x$ is an existential variable and $a, b < i$. For this step to be valid, each $M^u_i$ must either be equal to $\text{select}(M^u_a, M^u_b)$ or if $x < \langle u \rangle$ then it can be equal to $\text{merge}(M^u_a, M^u_b, i, x)$.

The final line $L_m$ is the conclusion of $\Pi$, and $\Pi$ is a refutation of $Q.\phi$ iff $C_m = \bot$. $G_\Pi$ be the derivation graph corresponding to $\Pi$ with edges directed from the hypothesis to the resolvent (i.e from the axioms to the final line). A refutation $\Pi$ is said to be regular if no leaf-to-root path in $G_\Pi$ has any existential variable $x$ as a pivot more than once. For some given line $L$, $\Pi_L$ is defined as the sub-derivation of $\Pi$ deriving the line $L$.

2.2 eFrege+$\forall$red

Frege systems are fundamental proof systems of propositional logic. Lines in a Frege proof are formulas inferred from the previous lines via few sound rules. The rules are not important as all Frege systems are p-equivalent, the only condition is that a Frege system needs to be sound and complete. So w.l.o.g, we can assume that ‘modus ponens’ is the only rule in a Frege system. The modus ponens is defined as: if $A \rightarrow B$ and $A$ are present in the hypothesis then $B$ can be logically implied by the hypothesis. For a detailed definition and explanation refer [20].

Extended Frege (eFrege) [12] is an extention of Frege systems which allows introduction of new variables not present in previous lines of the proof. This rule allows lines of the form $v \leftrightarrow f(S)$ where $v$ is a new variable and $f$ can be any formula on the set of variables $S$, where $v \notin S$.

For QBF’s, eFrege is modified to be eFrege+$\forall$red (Extended Frege + $\forall$ reduction) [6] which requires that the extension variable must be added in the prefix and quantified to the right of the variables used to define it. To deal with the universal variables, the universal reduction (UR) rule as defined in Q-Res is introduced into eFrege+$\forall$red. The formal definition is as follows:

$$L_j \rightarrow L_i = L_j[u/B] \quad (\forall red)$$

where $L_j$ is some previous line in the eFrege+$\forall$red proof, $u$ is a universal variable that is also rightmost in the prefix among all variables in $L_j$ and $B$ is the Herbrand function of $u$ [6]. That is, a universal variable $u$ in a formula can be replaced by 0 or 1 when no other variable in that formula are to the right of $u$ in the prefix.

For the rest of known QBF-proof systems depicted in Fig 1, refer to [25]. We define the following formulas needed for proving lower bound later in the paper.
2.3 Completion Principle Formulas [17]

The QBFs \( CR_n \) are defined as follows:
\[
CR_n = \exists \ i, j \in [n] \ x_{ij} \land \forall \ z \in [n] \ \exists \ a_i \land \exists \ b_j \left( \bigwedge_{i, j \in [n]} (A_{ij} \land B_{ij}) \right) \land L_A \land L_B
\]
where,
\[
A_{ij} = x_{ij} \lor z \lor a_i \\
B_{ij} = \overline{x_{ij}} \lor z \lor b_j \\
L_A = \overline{a_1} \lor \cdots \lor \overline{a_n} \\
L_B = \overline{b_1} \lor \cdots \lor \overline{b_n}
\]
For any \( CR_n \) formula, we define the sets \( A := \{a_1, a_2, ..., a_n\} \) and \( B := \{b_1, b_2, ..., b_n\} \) as the set of all \( a, b \) variables respectively.

3 MRes-\( \mathcal{R} \): A new family of proof systems for false QBFs

We define a family of proof systems MRes-\( \mathcal{R} \), inspired from the MRes proof system. In MRes ([4]), strategies are built within the proof and are represented by merge maps. We observed that merge maps or any specific representations of strategies are not important for the soundness or completeness of the proof system. Since, isomorphism problem is efficient in merge maps, they make the proof systems polynomial time verifiable.

Based on this observations, we define a family of MRes-\( \mathcal{R} \) where every proof system \( \mathcal{P} \in \text{MRes-}\mathcal{R} \) has it’s own complete representation to represent the strategies. To allow the use of arbitrary representations in MRes-\( \mathcal{R} \), we introduce consistency checking rules for strategies which are not as efficient as checking isomorphism for MRes. As a result, our proof systems are not polynomial time verifiable. However their soundness & completeness doesn’t depend on their representations, which makes them interesting.

We use the idea of consistency checking from [9], which uses the same for DQBFs. For simplicity, we use the same notations from [9] whenever possible. We begin by defining some important notations and operations needed before actually defining the MRes-\( \mathcal{R} \) systems.

3.1 Important notations used in MRes-\( \mathcal{R} \)

To begin, let us define what consistency means for any two assignments of a set of variables.

**Definition 1 ([9]).** Let \( X \) be any set of variables and \( \varepsilon, \delta \in \langle (X) \rangle \). We say that \( \varepsilon \) and \( \delta \) are consistent, denoted by \( \varepsilon \simeq \delta \), if for every \( x \in X \) for which \( \varepsilon(x) \neq \ast \) and \( \delta(x) \neq \ast \) we have \( \varepsilon(x) = \delta(x) \).

Let \( H_u \) and \( H'_u \) be individual strategy functions for the universal variable \( u \), the consistency between two strategies is defined as follows:

We say that \( H_u \) and \( H'_u \) are consistent (written \( H_u \simeq H'_u \)) when \( H_u(\varepsilon) \simeq H'_u(\varepsilon) \) for each \( \varepsilon \in \langle (L_Q(u)) \rangle \). Recall that \( L_Q(u) \) are the existential variables to the left of \( u \) in the prefix. In other words \( H_u \) and \( H'_u \) are consistent, if the \( u \)-assignment given by \( H_u(\varepsilon) \) and \( H'_u(\varepsilon) \) should be consistent for every possible \( L_Q(u) \)-assignment \( \varepsilon \).
By a change in notation, we can see (partial) assignments as both functions and sets of literals, i.e., an assignment \( \varepsilon \) corresponds to the set of literals it satisfies. For example, \( \{ x_1, x_2, \overline{x_3}, \overline{x_4} \} \) represents an assignment which sets 1 to the variables \( x_1 \) and \( x_2 \) and 0 to \( x_3 \) and \( x_4 \). In this notation as sets of literals, a union (\( \cup \)) of assignments \( \varepsilon, \delta \) is defined when \( \varepsilon \simeq \delta \) and it is equal to \( \varepsilon \cup \delta \).

We now define a union operation (\( \circ \)) on two consistent strategies \( H_u \) and \( H'_u \).

**Definition 2** ([9]). Given two consistent strategies \( H_u \) and \( H'_u \) (i.e., \( H_u \simeq H'_u \)), we define the union strategy \( H''_u \) of \( H_u \) and \( H'_u \), denoted by \( H''_u = H_u \circ H'_u \), as follows:

\[
H''_u(\varepsilon) = H_u(\varepsilon) \cup H'_u(\varepsilon) \quad \text{for each} \quad \varepsilon \in (L_Q(u)).
\]

For example, if \( H_u \& H'_u \) be defined as below, then \( H''_u = H_u \circ H'_u \) will be:

\[
H_u = \begin{cases} 1 : & x \\ * : & \overline{x} \end{cases} \quad \text{and} \quad H'_u = \begin{cases} * : & x \\ 0 : & \overline{x} \end{cases}
\]

\[
H''_u = \begin{cases} 1 \cup * = 1 : & x \\ * \cup 0 = 0 : & \overline{x} \end{cases}
\]

We now define an if-else operation (\( \triangledown \)) on any two strategies \( H_u \) and \( H'_u \).

**Definition 3** ([9]). Given any two strategies \( H_u \) and \( H'_u \) and an existential variable \( x \), we define the if-else operation of \( H_u \) and \( H'_u \) on \( x \) to give the strategy \( H''_u \), denoted by \( H''_u = H_u \triangledown H'_u \), for each \( \varepsilon \in (L_Q(u)) \) as follows:

\[
H''_u(\varepsilon) = \begin{cases} H_u(\varepsilon) & : \ \varepsilon(x) = 1 \\ H'_u(\varepsilon) & : \ \varepsilon(x) = 0 \end{cases}
\]

For example, if \( H_u \& H'_u \) be defined as below, then \( H''_u = H_u \triangledown H'_u \) will be:

\[
H_u = \begin{cases} 1 : & y \\ * : & \overline{y} \end{cases} \quad \text{and} \quad H'_u = \begin{cases} * : & z \\ 0 : & \overline{z} \end{cases}
\]

\[
H''_u = \begin{cases} 1 : & x y \\ * : & x \overline{y} \\ 0 : & \overline{x z} \end{cases}
\]

Note that the input strategies \( H_u, H'_u \) need not be consistent for an '\( \triangledown \)' operation, but they must be in case of an '\( \circ \)' operation.

### 3.2 Definition of MRes-\( \mathcal{R} \)

Let \( \Phi = Q, \phi \) be a QBF with existential variables \( X \) and universal variables \( U \). A MRes-\( \mathcal{R} \) derivation of \( L_m \) from \( \Phi \) is sequence \( \pi = L_1, ..., L_m \) of lines where each \( L_i = (C_i, \{ H^u_i : u \in U \}) \) in which at least one of the following holds for \( i \in [m] \):

a. **Axiom.** There exists a clause in \( C \in \phi \) such that \( C_i \) is the existential subclause of \( C \), and for each \( u \in U \), \( H^u_i \) is the strategy function for \( u \) mapping it to the falsifying \( u \)-literal for \( C \) or,

b. **Resolution.** There exist integers \( a, b < i \) and an existential pivot \( x \in X \) such that \( C_i = res(C_a, C_b, x) \) and for each \( u \in U \):

i. if \( x > Q u \), then \( H^u_i = H^u_a \triangledown H^u_b \)

ii. else if \( x < Q u \), then \( H^u_i = H^u_a \circ H^u_b \).
π is a refutation of \( \Phi \) iff \( C_m = \bot \). Size of \( \pi \) is the number of lines i.e \( |\pi| = m \). Observe that similar to MRes, proof systems in MRes-R have only existential literals in the clause part of the lines in a proof.

Let \( S \) be a subset of the existential variables \( X \) of a false QBF \( \mathcal{F} \). We say that a \( \mathcal{P} \)-refutation \( \pi \) of \( \mathcal{F} \) (where \( \mathcal{P} \in \text{MRes-R} \)) is \( S \)-regular if for each \( x \in S \), there is no leaf-to-root path in \( G_\pi \) that uses \( x \) as pivot more than once. An \( X \)-regular proof is simply called a regular proof.

## 4 Soundness and Completeness of MRes-R

In this section, we show that each proof system in MRes-R is sound and refutationally complete for false QBFs.

### 4.1 Soundness

The soundness proof of MRes-R follows closely to that of the MRes proof system. The following lemma proves the soundness of MRes-R family of proof systems.

**Lemma 4.** Let \( \mathcal{P} \in \text{MRes-R} \) be any proof system. Let \( \pi = L_1, L_2, \ldots, L_m \) be a valid \( \mathcal{P} \)-proof of QBF \( \Phi = Q.\phi \). Then, the strategy functions \( \{H^u_{\pi} : u \in U\} \) in the conclusion line \( L_m \) of \( \pi \), will form a countermodel for \( \Phi \).

**Proof:** Given \( \pi := L_1, \ldots, L_m \) be an \( \mathcal{P} \)-refutation of a QBF \( \Phi = Q.\phi \). Let each \( L_i = (C_i, \{H^u_{\pi} : u \in U\}) \) and \( X, U \) are sets of all existential and universal variables in \( Q \) respectively. Further, for each \( i \in [m] \),

- let \( \alpha_i := \{l : l \in C_i\} \) be the smallest assignment falsifying \( C_i \),
- let \( A_i := \{\alpha \in \langle X \rangle : C_i \cap \alpha = \emptyset\} \) be all complete assignments to \( X \) consistent with \( \alpha_i \),
- for each \( \alpha \in A_i \), let \( l^u_{\pi}(\alpha) := H^u_{\pi}(\alpha) \) and \( H_{\pi}(\alpha) := \{l^u_{\pi}(\alpha) : u \in U\} \setminus \{\ast\} \).

**Induction statement:** By induction on \( i \in [m] \), we show, for each \( \alpha \in A_i \), that the restriction of \( \phi \) by \( \alpha \cup H_{\pi}(\alpha) \) contains the empty clause.

**Proof:** For the base case \( i = 1 \), let \( \alpha \in A_1 \). As \( L_1 \) is introduced as an axiom, there exists a clause \( C \in \phi \) such that \( C_1 \) is the existential subclause of \( C \), and each \( H^u_{\pi} \) is the function outputting the falsifying \( u \)-literal for \( C \). Hence, for each \( u \in U \), \( l^u_{\pi}(\alpha) \) is the falsifying \( u \)-literal for \( C \), so \( C\alpha \cup H_{1}(\alpha) = \emptyset \).

For the inductive step, let \( i \geq 2 \) and let \( \alpha \in A_i \). The case where \( L_i \) is introduced as an axiom is identical to the base case, so we assume that \( L_i \) was derived by resolution. Then there exist integers \( a, b < i \) and an existential pivot \( x \in X \) such that \( C_i = res(C_a, C_b, x) \)

(1) Suppose that \( x \in \alpha \), each \( u \in U \) has to satisfy either:

(i) \( x <_Q u \) and \( H^u_{\pi} = H^u_{a} \odot H^u_{b} \in X \) In which case, \( l^u_{\pi}(\alpha) = l^u_{a}(\alpha) \).

(ii) \( x >_Q u \) and \( H^u_{\pi} = H^u_{a} \odot H^u_{b} \). In which case, \( l^u_{\pi}(\alpha) = \{l^u_{a}(\alpha) \cup l^u_{b}(\alpha)\} \).
It follows that \( l_i^a \neq l_i^b \) only if \( l_i^a = * \), and hence \( H_a(\alpha) \subseteq H_i(\alpha) \). Since \( C_a \setminus \{x\} \subseteq C_i \), we have \( \alpha \in A_u \), so the restriction of \( \phi \) by \( \alpha \cup H_i(\alpha) \) contains the empty clause by the inductive hypothesis that \( \alpha \cup H_a(\alpha) \) contains the empty clause.

(2) Suppose that \( x \in \alpha \). A similar argument shows that \( H_b(\alpha) \subseteq H_i(\alpha) \).

Since \( \alpha_m \) is the empty assignment, we have \( A_m = \{X\} \) (i.e. all complete assignments to \( X \)). We therefore prove the lemma at the final step \( i = m \), as we show that \( \{H_m^u : u \in U\} \) is a countermodel for \( \Phi \).

\[ \square \]

### 4.2 Completeness

One would notice that a major change of \textbf{MRes-R} from \textbf{MRes} is the usage of ‘consistency’ check instead of ‘isomorphism’ check. Note that the relation between them is as such: isomorphism \( \Rightarrow \) consistency but not vice-versa. We use this in our proofs for completeness of \textbf{MRes-R}.

We show the completeness by first showing that \textbf{MRes-M} p-simulates \textbf{MRes} (Theorem 6). Here, \textbf{MRes-M} is a proof system in \textbf{MRes-R} which uses merge maps as the representation. Further, we show in Claim 7 that a \textbf{MRes-M}-proof can be transformed into any \textbf{MRes-R}-proof in exponential time. Nevertheless, completeness is guaranteed as \textbf{MRes} is complete and any \textbf{MRes}-proof can be transformed into a \textbf{MRes-M}-proof which in-turn can be transformed as any \textbf{MRes-R}-proof.

We will need the following remark from the paper introducing \textbf{MRes} [4].

\textbf{Remark 5.} \textit{[4, Proposition 10]} Any two isomorphic merge maps compute the same function.

\textbf{Theorem 6.} \textbf{MRes-M} p-simulates \textbf{MRes}.

\textbf{Proof.} Given a QBF \( \Phi \) and its \textbf{MRes}-proof \( \pi = L_1, ..., L_m \), where every line \( L_i = \{C_i, \{M_i^u : u \in U\}\} \). We intend to build an \textbf{MRes-M}-proof \( \Pi = L'_1, ..., L'_m \) for \( \Phi \), where each \( L'_i = \{C'_i, \{H_i^u : u \in U\}\} \).

For every line \( L_i \) in \( \pi \) starting from \( i = 1 \) to \( m \), if \( L_i \) is an axiom step then directly \( C'_i = C_i \) and \( H_i^u = M_i^u \) for all \( u \in U \). Otherwise, if \( L_i \) is an resolution step i.e. for some \( a, b < i, C_i = \text{res}(C_a, C_b, x) \); then set \( C'_i = C_i \) and for each \( u \in U \) if \( x < Q u \) then set \( H_i^u = H_b^u \bowtie H_a^u \) else set \( H_i^u = H_a^u \circ H_b^u \). We see that these are sound steps as resolution in \textbf{MRes} can be of the following types:

\begin{enumerate}
  \item \( x > Q u \) and \( M_i = \text{select}(M_a^u, M_b^u) \); in this case we set \( H_i^u = H_a^u \circ H_b^u \) which holds given the Remark 5 and that isomorphism \( \Rightarrow \) consistency.
  \item \( x < Q u \) and \( M_i = \text{merge}(M_a^u, M_b^u, i, x) \); in this case we set \( H_i^u = H_b^u \bowtie H_a^u \) which is same as the merge function of \textbf{MRes}.
  \item \( x < Q u \) and \( M_i = \text{select}(M_a^u, M_b^u) \); in this case we set \( H_i^u = H_b^u \bowtie x H_a^u \) which is allowed as \textbf{MRes} did the isomorphism test on \( M_a^u \) and \( M_b^u \), but we need no such check for \( \bowtie \) in \textbf{MRes-R}. (ref. the note just after Definition 3).
\end{enumerate}
In case-(iii) above it remains to note that adding a $\triangleleft\triangleleft$ to two isomorphic maps or when one of them is $\ast$, doesn’t add any new strategy: it just dilutes the strategy represented by the corresponding merge map. That is, we are adding an if-else condition where both the outcomes are same or one of them is $\ast$. Hence doesn’t affect future consistency checks which may arise in the proof. (For further clarity, one is suggested to look at Example 1 but it is not needed for the proof).

Therefore, one can clearly see that the proof $\Pi$ constructed in this process is a valid $\text{MRes-M}$-proof for $\Phi$. Hence this proves the above theorem.

Claim 7. Every $\text{MRes-M}$-proof can be transformed into an $\text{MRes-R}$-proof for any representation $R$ in exponential time.

Proof. Given a QBF $\Phi$ and its $\text{MRes-M}$-proof $\pi = L_1, ..., L_m$, where every line $L_i = \{C_i, \{M_u^i : u \in U\}\}$. We intend to build an $\text{MRes-R}$-proof $\Pi = L'_1, ..., L'_m$ for $\Phi$, where each $L'_i = \{C'_i, \{H_u^i : u \in U\}\}$.

For every line $L_i$ in $\pi$, we keep the clause part intact while we convert the merge maps into plain functions. Further as $R$ is a complete representation, these functions should have a corresponding representation in $R$; we extensively search for the same. This search terminates at some point owing to $R$ being a complete representation. (This is the place where we used the property that $R$ is a complete representation).

The result is an $\text{MRes-R}$-proof for $\Phi$. This process is not polynomial in time but regardless still proves completeness for the family of proof systems $\text{MRes-R}$. □

Let us consider example 1 (below) which was referred in Theorem 6. This example considers the situation corresponding to the case-(iii) of Theorem 6. That is, two isomorphic merge maps can be combined with an if-else and the resulting strategy will still output the same as input merge maps. Or when one of the input merge map being $\ast$, makes the resulting strategy diluted in the sense that for half the assignments it gives a $\ast$ and for others the same as the non-trivial input merge map.

Example 1. Let $M_u^1 = M_u^2 = 1$ be leaf nodes in $\text{MRes}$ proof. It implies that corresponding $H_u^1 = 1$ and $H_u^2 = 1$ in $\text{MRes-R}$ proof. Now say $\text{MRes}$ performs a resolution on pivot variable $x$ which is to the left of $u$, resulting in $M_u^3 = \text{select}(M_u^1, M_u^2)$. Whereas the corresponding $\text{MRes-R}$ rule needs to be a $H_u^3 = H_u^1 \not\triangleleft H_u^2$ from case(iii) (ref. Theorem 6). That is, $H_u^3$ in function form would be defined as follows:

$$H_u^3 = \begin{cases} 1 : & x \\ 1 : & \overline{x} \end{cases}$$

Notice how this is just a diluted way of writing the strategy $H_u^3 = 1$. Hence when in the next line of $\text{MRes}$ if a $M_u^4 = 1$ which is isomorphic to $M_u^3$ is encountered; the corresponding $H_u^3 = 1$ in $\text{MRes-R}$ will still remain to be consistent with $H_u^3$ (though they might seem to be structurally different).

In the same example if $M_u^2 = \ast$ (i.e. trivial), the strategy $H_u^3$ would have been:

$$H_u^3 = \begin{cases} 1 : & x \\ \ast : & \overline{x} \end{cases}$$
Notice how this is another way of diluting the strategy and is still consistent with $H^u_4 = 1$.

So far, we showed that each proof system in MRes-$\mathcal{R}$ is sound and refutationally complete for false QBFs. Next, we present an example of MRes-$\mathcal{R}$ allowing few resolution steps which are not allowed in MRes. Such examples may be useful for the separation results between the proof systems in MRes-$\mathcal{R}$ and the MRes proof system.

**Example 2.** Consider any proof system $\mathcal{P}$ in MRes-$\mathcal{R}$ which uses some complete $R$ representation for strategies. The following Table 1 is a $\mathcal{P}$-refutation of the false QBF: $\exists x \forall u \exists y \left( (y \lor x \lor u) \land (y \lor \neg x) \land (\neg y \lor x) \land (\neg y \lor \neg x \lor \neg u) \right)$

| Line | Rule | $C_i$ | $H^u_i$ |
|------|------|-------|--------|
| $L_1$ | axiom | $\{y, x\}$ | 0 |
| $L_2$ | axiom | $\{y, \neg x\}$ | * |
| $L_3$ | res($L_1, L_2, x$) | $\{y\}$ | $H^u_x \triangleright H^u_4$ |
| $L_4$ | axiom | $\{\neg y, x\}$ | |
| $L_5$ | axiom | $\{\neg y, \neg x\}$ | 1 |
| $L_6$ | res($L_4, L_5, x$) | $\{\neg y\}$ | $H^u_5 \triangleright H^u_4$ |
| $L_7$ | res($L_3, L_6, y$) | $\{\neg y\}$ | $H^u_3 \circ H^u_6$ |

Table 1: $\mathcal{P}$-refutation, where $\mathcal{P} \in$ MRes-$\mathcal{R}$, of the false QBF in Example 2

The strategies $H^u_3$ and $H^u_6$ in function format are as follows:

$$H^u_3 = \{0: x = 0, *: x = 1\}$$
$$H^u_6 = \{*: x = 0, 1: x = 1\}$$

One can see that these strategies are consistent (but not isomorphic), hence the resolution of $L_3, L_6$ on $y$ is allowed in the $\mathcal{P}$-refutation. But the analogous resolution would be blocked in MRes since the corresponding merge maps $M^u_3, M^u_6$ will not be isomorphic.

### 5 eFrege+$\forall$red simulates MRes-$\mathcal{R}$

In this section, we show that eFrege+$\forall$red can efficiently simulate any valid refutation from proof system in MRes-$\mathcal{R}$. Therefore, the stronger proof systems like QRAT (Quantified Resolution Asymmetric Tautologies) [16] and G (Gentzen/Sequent Calculus) [21] can also simulate the same.

However, proof systems in MRes-$\mathcal{R}$ can have arbitrary representations. Simulating the same is a nightmare for even the strongest proof systems. But observe that the rules to construct the strategies in any representation are the same as defined in Section 3.2. We capture these rules as a new tree structure $T$. That is, given a valid proof $\pi$ of any proof system $\mathcal{P} \in$ MRes-$\mathcal{R}$, $\pi = (C_1, R_1), (C_2, R_2), \ldots, (C_m, R_m)$, we construct a proof $\pi' := (C_1, T_1), (C_2, T_2), \ldots, (C_m, T_m)$ in such a way that the representation $T_i$ captures the rules that have been used to construct the strategy $R_i$. 

We show that the representation $T$ is also a complete representation for finite functions. Therefore, $\text{MRes-T}$ also belongs to the family $\text{MRes-R}$. Further, we show that any valid $P$-proof $\pi$ (where $P \in \text{MRes-R}$) can be efficiently converted to an $\text{MRes-T}$-proof $\pi'$. Finally using the ideas from [10], we show that $\text{eFrege+∀red}$ can efficiently simulate $\text{MRes-T}$. This shows that $\text{eFrege+∀red}$ can simulate any proof systems in $\text{MRes-R}$. We now proceed and define the proof system $\text{MRes-T}$.

5.1 $\text{MRes-T}$ proof system

Given a false QBF $F$, a $\text{MRes-T}$ proof $\pi$ of $F$ is a sequence of lines

$$(C_1, T_1), (C_2, T_2), \ldots, (C_m, T_m)$$

where $C_m = \bot$ and each $T_i$ is constructed as follows: if $(C_i, T_i)$ is an axiom step, then $T_i$ is constructed as in the $\text{MRes}$ proof system. Otherwise if $(C_i, T_i)$ is a Resolution step on a pivot left of the universal variable in question (i.e if-else step (‘$\triangleleft$’)) of $\text{MRes-R}$), then $T_i$ is constructed, as a merge node is constructed in $\text{MRes}$. Further, if $(C_i, T_i)$ is constructed from a resolution step on $(C_j, T_j)$ and $(C_k, T_k)$ with pivot being right of the universal variable in question and both $T_j$ and $T_k$ are consistent (i.e union step (‘$\circ$’) of $\text{MRes-R}$), then $T_i$ is constructed by adding a new type of node called the # node (defined below) with inputs $T_j$ and $T_k$.

The # node is defined assuming both its inputs are consistent, and it outputs the result of a union operation on them; more clearly, its truth table is shown in the Fig. 2a.

| A | B | A # B |
|---|---|-------|
| 1 | 1 | 1     |
| 0 | 0 | 0     |
| * | 0/1| 0/1   |
| 0/1| *  | 0/1   |
| * | *  | *     |

(a) Truth table for # operator.
(It assumes inputs to be consistent.)

(b) $T_{13}^u$ graph for Example 3

Fig. 2: Truth table of # operator is shown in Fig. 2a and its use in $\text{MRes-T}$ depicted by an example QBF in Fig.2b.

Note that $A = 1, B = 0$ and vice-versa cannot happen in a valid $\text{MRes-R}$ proof owing to the definition of union(‘$\circ$’) which needs the input strategies to be consistent. Therefore, the corresponding rows are omitted from the # node truth table in Fig 2a. Let us illustrate a $\text{MRes-T}$-proof below for an example QBF.
Example 3. Let \( \Phi := \exists x, y, \forall u, \exists a, b \ (x, y, u, a) \land (x, y, \bar{u}, a) \land (x, y, u, a) \land (x, y, \bar{u}, b) \land (x, y, u, b) \land (y, \bar{b}) \). The MRes-\( T \) proof of \( \Phi \) is shown below in Table 2:

| Line | Rule | \( C_i \) | \( T_i^m \) | Type of node |
|------|------|-----------|-------------|--------------|
| \( L_1 \) | axiom | \( \{ x, y, a \} \) | 1 | Leaf |
| \( L_2 \) | axiom | \( \{ x, y, a \} \) | * | Leaf |
| \( L_3 \) | \( res(L_1, L_2, x) \) | \( \{ y, a \} \) | \( T_2^u \vdash \land T_1^u \) | if-else |
| \( L_4 \) | axiom | \( \{ x, y, \pi \} \) | * | Leaf |
| \( L_5 \) | axiom | \( \{ x, y, \pi \} \) | 1 | Leaf |
| \( L_6 \) | \( res(L_5, L_4, x) \) | \( \{ y, \pi \} \) | \( T_6^u \vdash \land T_4^u \) | if-else |
| \( L_7 \) | \( res(L_3, L_6, a) \) | \( \{ y \} \) | \( T_3 \circ T_6 \) | # |
| \( L_8 \) | axiom | \( \{ x, y, b \} \) | 1 | Leaf |
| \( L_9 \) | axiom | \( \{ x, y, b \} \) | 0 | Leaf |
| \( L_{10} \) | \( res(L_9, L_8, x) \) | \( \{ y, b \} \) | \( T_9^u \vdash \land T_8^u \) | if-else |
| \( L_{11} \) | axiom | \( \{ y, \bar{b} \} \) | * | Leaf |
| \( L_{12} \) | \( res(L_{10}, L_{11}, b) \) | \( \{ y \} \) | \( T_{10} \circ T_{11} \) | # |
| \( L_{13} \) | \( res(L_{12}, L_{7}, y) \) | \( \{ \} \) | \( T_{12}^u \vdash \land T_{12}^u \) | if-else |

Table 2: A MRes-\( T \) refutation of the false QBF in Example 3

The final \( T \)-graph of winning strategy for the only universal variable \( u \) from Example 3 is shown in Figure 2b. One can see that this graph is a hybrid structure of both branching programs and circuits. Since it has both ‘branching’ nodes (\( \triangledown \) nodes) and ‘circuit’ nodes (\( \# \) nodes).

Observe that the proposed \( T \) representation is complete. That is, any valid finite function can be represented by a \( T \) graph. This follows since, merge maps are a subset of \( T \)-graphs (i.e without \( \# \) nodes) which are just branching programs, but known to be complete for all valid functions. Since \( T \) representations are complete, MRes-\( T \) is a member of MRes-\( \mathcal{R} \) proof systems. Therefore this is a sound and complete proof system. Also note that MRes-\( T \) is not claimed to be polynomial time verifiable.

5.2 Conversion of MRes-\( \mathcal{R} \) proofs into MRes-\( T \) proofs

In this section we show how to convert a valid \( \mathcal{P} \)-proof \( \pi \) into a valid MRes-\( T \)-proof \( \pi' \), where \( \mathcal{P} \) be any proof system in MRes-\( \mathcal{R} \). Let \( \pi = (C_1, R_1), (C_2, R_2), \ldots, (C_m, R_m) \) be a valid \( \mathcal{P} \) proof of a QBF \( \mathcal{F} \). We show how to convert \( \pi \) into a valid MRes-\( T \)-proof \( \pi' = (C_1, T_1), (C_2, T_2), \ldots, (C_m, T_m) \) of the same QBF \( \mathcal{F} \). Note that here \( T_i \) is not the representation of \( R_i \), but \( T_i \) is capturing how \( R_i \) has been constructed from some hypothesis \( R_j, R_k \) with \( j, k < i \) using rules from Section 3.2. For this we do not need to interpret \( R_i \)'s, but we can extract the required information from the clauses \( C_j, C_k \) and \( C_i \) of \( \pi \).

It is also useful to note that, during this conversion, one doesn’t need to check if the two strategies \( R_j, R_k \) are consistent or not. The conversion is smooth and simple as it assumes \( \pi \) to be a valid \( \mathcal{P} \)-proof of \( \mathcal{F} \). We now proceed to give a detailed method for the same.
**Theorem 8.** Any valid $P$-proof ($P \in \text{MRes-R}$) can be converted efficiently into an $\text{MRes-T}$ proof.

**Proof.** For a false QBF $F$, proofs of proof systems belonging to $\text{MRes-R}$ can have arbitrary representations for the strategies computed. However, the rules allowed to construct a strategy $R_i$ using any strategies $R_j$ and $R_k$ (where $j, k < i$) are fixed. They must follow the rules mentioned in Section 3.2. $\text{MRes-T}$ proof $\pi'$ captures these rules only.

To be precise, given a $P$-proof $\pi$ of $F$ where $\pi = (C_1, P_1), (C_2, P_2), \ldots, (C_m, P_m)$, we construct $\text{MRes-T}$-proof $\pi'$ as follows:

- From the clause part of the proof i.e $C_1, \ldots, C_m$ (in this sequence) based on what step is being followed (axiom, or resolution where pivot is on left, or resolution where pivot is on right), we build the corresponding $T$-maps as shown in the Figure 3.

![Figure 3](image)

**Fig. 3:** Rules to construct $T$-graphs. In Figure 3a, $c_u$ is the falsifying strategy of $u$ for the axiom clause $C_i$. In Figure 3b, $C_i = \text{res}(C_j, C_k, x)$ and $x$ is left of $u$ in prefix i.e $T_i^u = T_k^u \otimes T_j^u$. In Figure 3c, $C_i = \text{res}(C_j, C_k, x)$ and $x$ is right of $u$ in prefix i.e $T_i^u = T_j^u \circ T_k^u$. Note that the truth table of the ‘# gate’ is defined in Figure 2a

After following this procedure for all lines in $\pi$, the sequence of lines so formed i.e $\pi' = (C_1, T_1), (C_2, T_2), \ldots, (C_m, T_m)$ is a valid $\text{MRes-T}$ proof as the clauses $C_1, \ldots, C_m$ are the same as in the original $\text{MRes-R}$ proof hence we know that $C_m$ is definitely $\perp$ and that $T_1, \ldots, T_m$ are built using the same rules as used when building the valid $P$-proof $\pi$. Therefore $T_m$ is a countermodel as it is building the same strategy as in $R_m$. \qed

Now we proceed to show that $\text{MRes-T}$ proof system can be efficiently simulated by $\text{efrege+\forallred}$. However before proving the same, observe that $\text{MRes-T}$ efficiently simulates $\text{MRes}$ proof system: due to Theorem 8, $\text{MRes-T}$ simulates any $\text{MRes-R}$ proof system, and therefore, it also simulates efficiently the $\text{MRes-M} \in \text{MRes-R}$ proof system, which is known to simulate the $\text{MRes}$ proof system efficiently (Theorem 6). Thus we have the following:

**Proposition 9.** $\text{MRes-T}$ efficiently simulates $\text{MRes}$. 

5.3 eFrege+$\forall$red simulates MRes-$\mathcal{T}$

In this section, we show that eFrege+$\forall$red efficiently simulates valid MRes-$\mathcal{T}$ refutations. We use the ideas from [10, Theorem 1] which shows how eFrege+$\forall$red efficiently simulates MRes. Let us briefly explain the idea from [10, Theorem 1]: Given an MRes-proof $\pi = (C_1, M_1), (C_2, M_2), ..., (C_m, M_m)$ of a false QBF $\mathcal{F}$, we know that if $\pi$ is valid then the merge map $M_m$ in the last line gives a winning strategy $S$ for the universal player of $\mathcal{F}$. That is, if we assign values of the universal variables based on $S$, it falsifies $\mathcal{F}$. In [10] they derived an eFrege+$\forall$red proof $\pi'$ from $\pi$ efficiently in two phases: in the first phase, they derived $\mathcal{F} \models_{\text{eFrege}} (S \rightarrow \bot)$ using $\pi$. This is equivalent to $\mathcal{F} \models_{\text{eFrege}} S$. This first phase was purely propositional. Later in the second phase, they used universal reduction to prove $\mathcal{F} \models_{\text{eFrege}, \forall\text{red}} \bot$. Implying from both phases that $\mathcal{F} \models_{\text{eFrege}, \forall\text{red}} \bot$. We also use the same tricks for simulating MRes-$\mathcal{T}$ with eFrege+$\forall$red. Hence, we also simulate the same in two phases. However, in the first phase, they used a double induction in which the second induction depicted how to handle ‘Select’ and ‘Merge’ nodes of MRes. We extend this to MRes-$\mathcal{T}$ by introducing ‘# nodes’ and giving a method to handle those in the second induction. We now prove this in detail.

**Theorem 10.** eFrege+$\forall$red efficiently simulates MRes-$\mathcal{T}$.

**Proof.** Phase-1:
Given a valid MRes-$\mathcal{T}$ proof $\pi := (C_1, T_1), (C_2, T_2), ..., (C_m, T_m)$ of a false QBF $\mathcal{F}$, we create new extension variables for each node in every strategy appearing in the proof. That is, $s^u_{i,t}$ is created for the node $t$ in the strategy $T^u_i$ for the universal variable $u$.

We define $s^u_{i,t}$ based on whether the corresponding $T^u_i(t)$ is an axiom node, if-else node or # node as follows:

$$
\begin{align*}
\text{if } u_i \text{ is an axiom node:} \\
\{1/0/\} & \quad T^u_i(t) = \{1/0/\} \\
(y \land s^u_{i,b}) \lor (\overline{y} \land s^u_{i,c}) & \quad T^u_i(t) = T^u_i(b) \lor T^u_i(c) \\
\text{if } u_i \text{ is an if-else node:} \\
s^u_{i,b} \neq s^u_{i,c} & \quad T^u_i(t) = T^u_i(b) \circ T^u_i(c)
\end{align*}
$$

In the quantifier prefix, we place the newly created variables $s^u_{i,t}$ to the immediate left of $u$ to maintain the soundness of the proof, as strategies for $u$ depends on these variables.

We now prove the outer induction in ‘Induction 1’ which assumes that ‘Induction 2’ is valid and hence can derive the clause $C_i$ by assigning local strategies to universal variables through a simple resolution for every line $L_i$.

**Induction 1** Consider the $i^{th}$ line of $\pi$, that is, $(C_i, T_i)$. It is easy for eFrege+$\forall$red to prove $\wedge_{u \in U_i}(u \rightarrow s^u_{i,r(u,i)}) \rightarrow C_i$, where $r(u,i)$ is the index of the root node of $T^u_i$. $U_i$ is the subset of $\bar{U}$ for which $T^u_i$ is non-trivial.

**Proof:**
Base case: Axiom: Suppose $C_i$ is derived by axiom download of some clause $C \in \mathcal{F}$. If $u$ has a non-trivial strategy, it is because it appears in the clause $C$ and...
Starting with \( \text{Inductive step:} \) Resolution: If \( C_j \) is resolved with \( C_k \) to get \( C_i \) with pivots \( \pi \in C_j \) and \( \pi \in C_k \), where \( j, k < i \). From the induction hypothesis, we have \( \bigwedge_{u \in U_i} (u \leftrightarrow s_{i,r(u,j)}^u) \rightarrow C_j \) and \( \bigwedge_{u \in U_k} (u \leftrightarrow s_{k,r(u,k)}^u) \rightarrow C_k \). Observe that using these clauses, we cannot prove the required statement. However, note that if on the left hand side of these clauses, one changes the \( j \) and \( k \) respectively to \( i \), then using resolution we can derive \( C_i \) on the right hand side. We show in the Induction 2 (below) how to achieve the same. To be precise, from Induction 2 we prove that, \( \bigwedge_{u \in U_i} (u \leftrightarrow s_{i,r(u,i)}^u) \rightarrow C_j \) and \( \bigwedge_{u \in U_i} (u \leftrightarrow s_{i,r(u,i)}^u) \rightarrow C_k \) holds. We then resolve these together to derive \( C_i \). This proves Induction 1.

Now in Induction 2 below, we prove what we claimed before in Induction 1 i.e. given \( \bigwedge_{u \in U(U \cap U_i)} (u \leftrightarrow s_{j,r(u,j)}^u) \) \( \land \) \( \bigwedge_{u \in U(W \cap U_i)} (w \leftrightarrow s_{w,r(w,i)}^u) \rightarrow C_j \). Recall that \( U_i \) is the subset of \( U \) for which \( T_i^u \) is non-trivial.

**Proof:**

**Base case:** Initially \( W \) is empty and as strategies cannot go back to be trivial \( U_j \subseteq U_i \). Hence the statement to prove is exactly the statement given above in the hypothesis. Therefore, base case is trivially true.

**Inductive step:**

Starting with \( \bigwedge_{u \in U \cap U_j} (u \leftrightarrow s_{j,r(u,j)}^u) \land \bigwedge_{w \in U \cap U_i} (w \leftrightarrow s_{w,r(w,i)}^u) \rightarrow C_j \).

We pick a \( u \in V \) to adjust into \( i \)-terms, i.e. we show the following:

\( (u \leftrightarrow s_{u,i}^u) \land \bigwedge_{v \in U \cap U_j} (v \leftrightarrow s_{j,r(v,j)}^u) \land \bigwedge_{w \in U \cap U_i} (w \leftrightarrow s_{w,r(w,i)}^u) \rightarrow C_j \).

We have three cases based on the rule used to derive the line \( L_i = (C_i, T_i) \):

- **Case i:** \( T_j^u = * \)
  - **Case ii:** \( T_j^u \neq * \), \( T_i^u = T_j^u \times \sim T_i^u \)
  - **Case iii:** \( T_j^u \neq * \), \( T_i^u = T_j^u \circ T_i^u \)

In case (i) we can easily adjust the universal variable \( u \). That is, we can simply add the following: \( (u \leftrightarrow s_{u,i}^u) \). This is sound because the clause \( (u \leftrightarrow s_{j,r(u,j)}^u) \) has never appeared before in the left hand side of the hypothesis but still we were able to derive \( C_j \). Therefore, adding \( (u \leftrightarrow s_{u,i}^u) \) to the left hand side of the hypothesis, will still be able to derive \( C_j \).

In case (ii) we prove inductively that for each node \( t \) in \( T_j^u \), \( (s_{i,t}^u \leftrightarrow s_{j,t}^u) \) holds. This is true for all leaf and intermediate nodes of \( T_j^u \) as we are only going to connect two \( T \)-graphs (i.e. \( T_j^u \), \( T_k^u \)) by an extra \( i \)-else node in \( T_i^u \), i.e. all nodes of \( T_j^u \) are present in \( T_i^u \). Hence eventually at the root of \( T_i^u \), we will have \( s_{i,r(u,j)}^u \leftrightarrow s_{j,r(u,j)}^u \). However, to prove the induction statement (Induction 2), we need to show this relation between...
roots of $T_u$ and $T_j$ i.e, $s_{i,r(u,i)}^u \iff s_{j,r(u,j)}^u$. For this we use the definition of merging that $x \rightarrow (s_{i,r(u,i)}^u \iff s_{j,r(u,j)}^u)$ and so we have $(s_{i,r(u,i)}^u \iff s_{i,r(u,j)}^u) \lor \overline{\exists}$. We almost got the relation we needed but only $\overline{\exists}$ is the extra literal. But note that $\overline{\exists}$ is already in $C_j$. So, the $\overline{\exists}$ is absorbed by the $C_j$ in right hand side of the implication.

In case (iii) using a similar induction as used in case (ii), we can derive $s_{i,r(u,i)}^u \iff s_{i,r(u,j)}^u$; because we are not deleting any strategies just adding an $\# \bar{\exists}$ gate. By the definition of the $\# \bar{\exists}$ gate, $s_{i,r(u,i)}^u \neq s_{i,r(u,j)}^u$ only when $s_{i,r(u,j)}^u \iff \ast$, in which case it is directly case-(i) above. That is, we can directly add $u \iff s_{i,r(u,i)}^u$ to the given hypothesis and we are done. In the other case when $s_{i,r(u,i)}^u = s_{i,r(u,j)}^u$, we can simply add $s_{i,r(u,i)}^u \iff s_{i,r(u,j)}^u$ which directly proves the induction step. \[\square\]

**Phase-2:**

At this point, from the Induction 1, we have derived:

$$\mathcal{F} \models \forall u \in U_m \left( u \iff s_{m,r(u,m)}^u \rightarrow \bot \right)$$

In other words, we have derived the winning strategy (say $S$) for the universal player in the QBF $\mathcal{F} \models \mathcal{F} \models (S \rightarrow \bot)$. Equivalently, $\mathcal{F} \models S$. Also, observe that so far we are only in the propositional world. Using the ideas from [10], now from the negation of the strategies for the universal player (i.e., $\overline{S}$), we can easily derive the empty clause using the universal reduction steps.

That is, we have the following $\overline{S} := \bigvee_{i=1}^{n-1} (u_i \oplus s_{m,r(u,m)}^u)$, where $U_m = \{u_1, u_2, \ldots, u_n\}$ in this order in the prefix. Observe the following property for some $k = 1$ to $k = n$ in this order:

$$\bigvee_{i=1}^{n-k+1} (u_i \oplus s_{m,r(u,m)}^u)$$

From the above formula, just pull out the last term and we have:

$$\bigvee_{i=1}^{n-k} (u_i \oplus s_{m,r(u,m)}^u) \lor (u_{n-k+1} \oplus s_{m,r(u_{n-k+1},m)}^u)$$

Performing the universal reduction step on $u_{n-k+1}$ is the same as:

$$\bigvee_{i=1}^{n-k} (u_i \oplus s_{m,r(u,m)}^u) \lor (0 \oplus s_{m,r(u_{n-k+1},m)}^u) \land \bigvee_{i=1}^{n-k} (u_i \oplus s_{m,r(u,m)}^u) \lor (1 \oplus s_{m,r(u_{n-k+1},m)}^u)$$

Which is same as:

$$\bigvee_{i=1}^{n-k} (u_i \oplus s_{m,r(u,m)}^u) \lor (s_{m,r(u_{n-k+1},m)}^u) \land \bigvee_{i=1}^{n-k} (u_i \oplus s_{m,r(u,m)}^u) \lor (s_{m,r(u_{n-k+1},m)}^u)$$

Note that we can perform universal reduction on $u_{n-k+1}$ as the only existential new variable appearing in the clause is to the left of it in the prefix.
We can resolve these two to get the following:

\[
\bigvee_{i=1}^{n-k} (u_i \oplus s_{m,r(u_i,m)})
\]

Note that we used the following rule above: \(0 \oplus x = x\) and \(1 \oplus x = \bar{x}\). We continue reducing all \(u_i\)’s to derive the \(\bot\) at the end.

The proof of Theorem 9 can be concluded by combining the results of Phase-1 and Phase-2 i.e., \(F_{\text{efrege}} S\) and \(S_{\text{efrege+red}} \bot\). Implying from both phases that \(F_{\text{efrege+red}} \bot\).

Note that given a valid MRes-T proof, by the simulation in Theorem 10 one can obtain a valid eFrege+∀red proof. But the validness of the resultant eFrege+∀red proof cannot be used to determine if the initial MRes-T proof was valid or not. That is, an invalid MRes-T proof may result into a valid eFrege+∀red proof. Therefore as claimed before, these MRes-R proof systems are not polynomial time verifiable even after being simulated by the powerful eFrege+∀red proof system. From the above discussions, Theorem 8 and Theorem 10 imply the following:

**Theorem 11.** eFrege+∀red efficiently simulates valid refutations from proof systems in MRes-R.

### 6 Lower Bound for Regular MRes-R

In this section, we lift the lower bound of Completion Formulas (CR\(_n\), Section 2.3) for Regular-MRes [5, Theorem 9], to Regular-MRes-R. We state the CR\(_n\) formulas once again for ease of reference.

\[
\text{CR}_{n} = \exists_{i,j \in [n]} x_{ij}, \forall z, \exists_{i \in [n]} a_i, \exists_{j \in [n]} b_j \left( \bigwedge_{i,j \in [n]} (A_{ij} \land B_{ij}) \right) \land L_A \land L_B
\]

where,

\[
\begin{align*}
A_{ij} &= x_{ij} \lor z \lor a_i \\
B_{ij} &= \bar{x}_{ij} \lor z \lor b_j \\
L_A &= a_1 \lor \cdots \lor a_n \\
L_B &= \bar{b}_1 \lor \cdots \lor \bar{b}_n
\end{align*}
\]

The lower bound follows from a stronger result that we prove below in Theorem 12 that any (A ∪ B)-regular refutation of CR\(_n\) in any proof system belonging to MRes-R must have size \(2^{O(n)}\). We use the ideas from [5] to prove the lower bound. We try to maintain the same notations wherever possible for simplicity.

Before presenting the lower bound proof in detail, we present the basic idea for the same. The proof setup is depicted in Figure 4. As every clause in CR\(_n\) has a variable from the set \(A \cup B\), but the refutation should derive a \(\bot\) at the final line; there must be a ‘section’ of the proof (See shaded region \(S\)' in Fig 4) which only has \(X\) variables in all its clauses. This section also includes the final line. The set of clauses at the ‘border’ (See the bold line \(S\) in Fig 4) of this section of the proof is shown to be wide (in terms of number of literals) in Lemma 13. Using this and the argument that the conjunction of clauses in \(S\) itself forms a false CNF formula,
we show in Theorem 12 that the number of clauses in $S$ is large (exponential in $n$). This directly implies that the size of the $\text{MRes-R}$-proof is also large.

To establish the width bound, we note that the pivots which are used while deriving clauses in $S$ are variables from $A \cup B$ and that they are all to the right of $z$. Meaning that the corresponding resolutions must all be union steps i.e the incoming strategies must be consistent (not isomorphic as is the case in $\text{MRes}$). This especially makes it difficult to directly lift the lower bound proof of $\text{MRes}$ from [5]. However we successfully overcome this issue in Claim 14 by arguing how $L_A, L_B$ are the only clauses with trivial strategies and how any other clause which resolves with these will mask this trivial-ness with its own definitive strategy. Further, by analysing what axiom clauses cannot be used in the derivation of the clauses in $S$, we show that many variables cannot be resolved before these lines. Hence, these variables will still be present in the clause $\in S$, making it wide. We now clearly state and prove the theorem for the lower bound result.

**Theorem 12.** Every $(A \cup B)$-regular refutation of $\text{CR}_n$ in any proof system belonging to $\text{MRes-R}$ has size $2^{\Omega(n)}$.

**Proof.** For $\mathcal{P} \in \text{MRes-R}$, let $\Pi$ be a $\mathcal{P}$-refutation of $\text{CR}_n$ (for $n > 2$). Let the set of lines $S, S'$ be defined as follows:

- $S'$: This set consists of all the lines $L = (C, H^z)$ from $\Pi$ such that $\text{vars}(C) \cap \{A \cup B\} = \emptyset$ and there exists a path from $L$ to $\bot$ in $G_\Pi$ consisting of lines only from $S'$.
- $S$: This set contains all the lines $L \in S'$ such that $L = \text{Res}(L_1, L_2, v)$ where $L_1, L_2 \notin S'$. Observe that the pivot variable $v$ must belong to $\{A \cup B\}$.

Fig. 4: Lower bound proof illustration. Given any $\mathcal{P} \in \text{MRes-R}$, a $\text{CR}_n$ formula and it’s $\mathcal{P}$-proof $\Pi$, this figure shows the graph $G_\Pi$. Claim 14 illustrates that $x_{ij} \notin \text{var}(H^z_j)$ for $i \in [n-1], j \in [n]$. Claim 15 illustrates that $|\text{vars}(C_2)| \geq n - 1$. Lemma 13 shows that $|\text{vars}(C)| \geq n - 1$. Theorem 12 proves that $|S| \geq 2^{\Omega(n)}$.
Let $F = \bigwedge_{(C,H^2) \in S} C$. Note that $F$ is a false CNF formula because there exists a subderivation $\hat{\Pi} = \{C | \exists L = (C, H^2) \in S\}$ which derives a $\bot$ given $F$. The variables in $F$ are only $x_{ij}$’s where $i, j \in [n]$, therefore it consists of $n^2$ variables. In Lemma 13 we prove that each clause in $F$ has width $\geq n - 1$. That is each clause can be falsified by setting at least $n-1$ variables to 0. Hence the number of complete assignments of $X$ that can falsify a clause $\in F$ will be at most $2^{n^2-(n-1)}$. Since $F$ is a false CNF formula, all assignments to $X$ should falsify some clause of $F$. Therefore, the number of clauses in $F$ should be $\geq 2^{n^2}$. This implies that the number of lines in $\Pi$ must also be exponential in $n$. \hfill \Box

Now it remains to prove Lemma 13 which we prove below.

**Lemma 13 ([5]).** For all $L = (C, H^2) \in S$, $\text{width}(C) \geq n - 1$.

**Proof.** Observe that $L$ is not an axiom as all axioms of $\text{CR}_n$ have a variable from $A \cup B$ and so they cannot belong to $S$. So, let $L = res(L_1, L_2, v)$ where $L_1, L_2 \notin S'$. Since two lines not belonging in $S'$ resolve to make the resultant $\in S'$, the pivot (i.e $v$) should be from $A \cup B$. Assume $v \in A$, a similar argument can be made when $v \in B$. Without loss of generality, assume that $v = a_n^1$; and $a_n \in C_1$ and $\overline{a_n} \in C_2$.

Since $\Pi$ is $(A \cup B)$-regular, $a_n$ does not occur as a pivot in the sub-derivation $\Pi_{L_1}$. It implies that the axiom clause $L_A$ cannot be used in deriving $L_1$, because otherwise $C_1$ will have both $a_n \& \overline{a_n}$ which makes it a tautology. That implies, axioms with other positive literals $a_i$’s cannot be used in $\Pi_{L_1}$ as the negated literals $\overline{a_i}$’s are only available in $L_A$ which in-turn cannot be used in $\Pi_{L_1}$. Positive literals of $a_i$’s only $\in A_{ij}$ for all $j \in [n]$. Hence, axioms $A_{ij}$ for $i \in [n-1], j \in [n]$ also cannot be used in deriving the line $L_1$. Now, we know $x_{ij}$ only occur in $A_{ij}$; so $H^\pi_1$ has no $x_{ij}$ variable for $i \in [n-1], j \in [n]$. Also, $H^\pi_1$ is not a trivial strategy as some $A_{nj}$ for $j \in [n]$ has been used because $a_n \in C_1$.

Since the pivot $a_n$ at the resolution step obtaining line $L$ is to the right of $z$, by the rules of $\text{MRes-\ R}$, $H^\pi_1$ and $H^\pi_2$ are consistent. In Claim 14, we prove that even though $\text{MRes-\ R}$ only insists on consistency, it still holds that for each $i \in [n-1]$, and each $j \in [n]$, $x_{ij} \notin \text{var}(H^\pi_2)$. Using this result we prove in Claim 15 below, that $C_2$ will have at least $n - 1$ variables (including $\overline{a_n}$). Therefore, at least $n - 2$ variables from $C_2$ belong to $C$.

Also, observe that $x_{nj} \in C_1$ for some $j \in [n]$; Since some clause $A_{nj}$ for $j \in [n]$ was used in $\Pi_{L_1}$, the literal $x_{nj}$ is introduced into the proof and resolution of $x_{nj}$ is not possible before $L_1$. This is because, the clause $B_{nj}$ needed to resolve it, brings with it literal $b_j$ which needs to be resolved before $L_1$ (as $L_1$ cannot have any $A \cup B$ literals other than $a_n$). To resolve this $b_j$, one needs to introduce the clause $L_B$, but $L_B$ brings all $b$’s which is an inconsistency which cannot be further resolved as the $B$-clauses needed for the same do not have consistent strategies anymore. That is, because of the use of $A_{nj}$ the resolvent has a 0 strategy for some assignment to $X$

\footnote{Note that here $a_n$ is used only for ease in dividing the set $A$ into partitions. nowhere in the proof we use the fact that $a_n$ is the last variable in $A$. Hence it is indeed w.l.o.g}
variables, but $B$-clauses have a constant strategy of 1 hence these strategies will not be consistent to resolve further.

Hence, we know $x_{nj} \in C_1$ for some $j \in [n]$. It implies that $x_{nj} \in C$ as well. This $x_{nj}$ cannot $\in C_2$ as the corresponding axiom clause needed for the same has $a_n$ in it, which would make $C_2$ a tautology. Using the three results above, we can derive that $\text{width}(C) \geq n - 1$.

\textbf{Claim 14.} For $i \in [n - 1]$, and each $j \in [n]$, $x_{ij} \notin \text{var}(H^2_z)$.

\textit{Proof.} At the point of use of this claim in the proof of Lemma 13, we definitely know that for $i \in [n - 1] \& j \in [n]$: $x_{ij} \notin H^1_z$. That is, if $f_1$ is the function representing the strategy $H^1_z$, then for any assignment $\sigma$ of $x_{nj}$’s and $i \in [n - 1], j \in [n]$, it implies that:

$$f_1(\sigma, x_{ij} = 0) = f_1(\sigma, x_{ij} = 1)$$

(1)

Let $f_2$ be the function representing the strategy $H^2_z$. Since $a_n$ is to the right of $z$, we know that $H^1_z$ and $H^2_z$ are consistent, i.e. for any assignment $\sigma'$ (an extension of $\sigma$) and for $i \in [n - 1], j \in [n]$, it implies that:

$$f_2(\sigma', x_{ij} = 0) \simeq f_1(\sigma', x_{ij} = 0)$$

(2)

$$f_2(\sigma', x_{ij} = 1) \simeq f_1(\sigma', x_{ij} = 1)$$

(3)

Only remaining question is if $f_2(\sigma', x_{ij} = 0) = f_2(\sigma', x_{ij} = 1)$? Observe that if this equality holds, then $f_2$ will be independent of $x_{ij}$’s, which implies that $x_{ij} \notin H^2_z$ for $i \in [n - 1], j \in [n]$. Now, we are heading towards proving the equality holds. Note that if none of the terms in equation 2 and equation 3 give a ‘$\ast$’ for any assignment of $X$, the equality in question definitely holds. So, now we prove that none of them can give a ‘$\ast$’ for any given assignment.

The only axiom clauses of $\text{CR}_n$ with trivial strategies are $L_A, L_B$ and these axioms only contain variables of $A \cup B$, which are all to the right of $z$. Hence if any other clause is to be resolved with these clauses, the pivot has to be in $A \cup B$ i.e. a union step needs to be performed. At this point the trivial-ness of $L_A$ (or $L_B$) is masked and does not show up in the final strategy of the resultant line; this is because union of any strategy with a trivial strategy will be the strategy itself. The only case by which a ‘$\ast$’ can be in the resulting strategy is if $L_A$ is resolved with $L_B$, which can clearly not happen as they have no common variable.

Since $C_1, C_2$ are definitely not the axiom clauses $L_A$ (or $L_B$), using the above argument it is simply not possible for the functions $f_1$ (or $f_2$) to output a ‘$\ast$’ for any input assignment provided. This means the equality in question above holds; meaning that $H^2_z$ also doesn’t depend on $x_{ij}$’s when $i \in [n - 1], j \in [n]$ i.e $x_{ij} \notin \text{vars}(H^2_z)$.

Now we prove claim 15 which was used in Lemma 13.

\textbf{Claim 15 ([5])}. Either for all $i \in [n - 1]$, $C_2$ has a variable of the form $x_{i\ast}$, or for all $j \in [n]$, $C_2$ has a variable of the form $x_{\ast j}$.
Proof. At this point in the proof of Lemma 13, we definitely know that \( \overline{a}_n \in C_2 \), and for all \( i \in [n - 1] \), for all \( j \in [n] \), \( x_{ij} \notin \text{var}(H^2) \). We prove this claim by contradiction. Suppose the claim is wrong i.e, there exists some \( u \in [n - 1] \) where for all \( l \in [n] \) \( x_{ul} \notin \text{var}(C) \) and some \( v \in [n] \) where for all \( k \in [n] \) \( x_{kv} \notin \text{var}(C) \).

Let \( \rho \) be the minimum partial assignment falsifying \( C_2 \). Then we know that:

- \( \rho \) sets \( a_n = 1 \), leaves all other variables in \( A \cup B \) unset, since they \( \notin C_2 \).
- \( \rho \) does not set any \( x_{ul} \) or \( x_{kv} \), since by our assumptions they all are not in \( C_2 \).

Now, extend \( \rho \) to assignment \( \alpha \) by setting:

- \( a_u = b_v = 0 \) and rest all unset variables from \( A \cup B \) to 1.
- Also except \( x_{uv} \), set \( x_{us} = 1 \) and \( x_{sv} = 0 \).

Observe that the assignment \( \alpha \) satisfies all axiom clauses except \( A_{uv} \) and \( B_{uv} \), and does not falsify any axiom.

Now extend \( \alpha \) to \( \alpha_0 \) and \( \alpha_1 \) by setting \( x_{uv} = 0 \) and 1 respectively.

The extension \( \alpha_0 \) satisfies one more axiom i.e. \( B_{uv} \); similarly \( \alpha_1 \) satisfies one more axiom i.e. \( A_{uv} \). Note that they still do not falsify the remaining axiom. That is, \( \alpha_0 \) does not falsify \( A_{uv} \) and similarly, \( \alpha_1 \) does not falsify \( B_{uv} \).

\( \alpha_0 \) and \( \alpha_1 \) agree everywhere except on \( x_{ij} \), and since \( x_{ij} \notin \text{var}(H^2) \), it follows that \( H^2 (\alpha_0) = H^2 (\alpha_1) \), say this value is equal \( d \).

From the proved Induction in Lemma 4, the partial strategy of universal player at every line combined with the extension of the existential assignment falsifying it’s clause part, should falsify some axiom of the QBF. Also, \( \alpha_0 \) and \( \alpha_1 \) falsify \( C_2 \), since they extend \( \rho \). Hence, it is a contradiction that \( (\alpha_0, d) \) satisfies all axioms. Therefore, the claim needs to be true.

\( \square \)

From the above discussions and due to Theorem 12, we have the following:

**Theorem 16.** Every \( MRes-R \)-regular refutation of \( CR_n \) has size \( 2^{\Omega(n)} \).

7 Conclusion and Future work

\( MRes \) proof system introduced recently in [4] builds strategies into proofs for false QBFs. We extend this proof system to a new family of sound, refutationally complete but not polynomial time verifiable proof systems \( MRes-R \). For each complete representation \( R \), we have a proof system in \( MRes-R \). We also define a complete representation \( T \), and it’s proof system \( MRes-T \) belonging to \( MRes-R \). We show how this \( MRes-T \) proof system efficiently simulates the before-mentioned \( MRes \) proof system. We also prove that \( eFreg+\text{red} \) can simulate every valid refutation from proof systems belonging to \( MRes-R \). Further, we establish a lower bound of Completion Formulas (\( CR_n \)) for every regular-proof system in \( MRes-R \). Refer Fig 1 for the resulting landscape of QBF-proof systems with efficient simulations.

Still several open problems remain in the scope of this paper. We would like to end our discussions by pointing out a few of them.
The simulation relation between proof systems in MRes-R and MRes is still open. Since proof systems in MRes-R uses strong consistency checking rules as compared to the isomorphism rule in MRes, we believe that there exists a family of QBFs which are easy for proof systems in MRes-R but hard for MRes. For the motivation of the same refer Example 2. It presents the resolution steps forbidden in MRes but allowed in MRes-R.

Another important open problem, is to establish a lower bound for proof systems in MRes-R. Note that whether KBKF-lq formulas from [3], is hard or easy for proof systems in MRes-R is still open. These formulas have been shown to be hard for the MRes proof system in [5].

MRes proof system is inspired from the LD-Q-Res proof system. It allows some forbidden resolution steps of LD-Q-Res. It has already been shown that MRes efficiently simulates the reduction-less LD-Q-Res proof system [4]. However, it is still open whether MRes and LD-Q-Res are incomparable, or if one can simulate the other.

8 Acknowledgements

We would like to thank Gaurav Sood for many important discussions, comments and suggestions regarding this paper.

References

1. Sanjeev Arora and Boaz Barak. Computational Complexity - A Modern Approach. Cambridge University Press, 2009.
2. Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. Formal Methods in System Design, 41(1):45–65, August 2012.
3. Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In Theory and Applications of Satisfiability Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings, volume 8561 of Lecture Notes in Computer Science, pages 154–169. Springer, 2014.
4. Olaf Beyersdorff, Joshua Blinkhorn, and Meena Mahajan. Building strategies into QBF proofs. J. Autom. Reason., 65(1):125–154, 2021.
5. Olaf Beyersdorff, Joshua Blinkhorn, Meena Mahajan, Tomás Peitl, and Gaurav Sood. Hard qbs for merge resolution. In 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2020, December 14-18, 2020, BITS Pilani, K K Birla Goa Campus, Goa, India (Virtual Conference), volume 182 of LIPIcs, pages 12:1–12:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
6. Olaf Beyersdorff, Ilario Bonacina, Leroy Chew, and Ján Pich. Frege systems for quantified boolean logic. J. ACM, 67(2):9:1–9:36, 2020.
7. Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. On unification of QBF resolution-based calculi. In Mathematical Foundations of Computer Science (MFCS), pages 81–93, 2014.
8. Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Understanding cutting planes for qbs. Inf. Comput., 262:141–161, 2018.
9. Joshua Blinkhorn, Tomáš Peitl, and Friedrich Slivovsky. Davis and putnam meet henkin: Solving DQBF with resolution. In Chu-Min Li and Felip Manyà, editors, Theory and Applications of Satisfiability Testing - SAT 2021 - 24th International Conference, Barcelona, Spain, July 5-9, 2021, Proceedings, volume 12831 of Lecture Notes in Computer Science, pages 30–46. Springer, 2021.

10. Leroy Chew and Friedrich Slivovsky. Towards uniform certification in QBF. Electron. Colloquium Comput. Complex., page 144, 2021.

11. Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. J. Symb. Log., 44(1):36–50, 1979.

12. Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. J. Symb. Log., 44(1):36–50, 1979.

13. William J. Cook, Collette R. Coullard, and György Turán. On the complexity of cutting-plane proofs. Discrete Applied Mathematics, 18(1):25–38, 1987.

14. Yuval Filmus, Pavel Hrubes, and Massimo Lauria. Semantic versus syntactic cutting planes. In Nicolas Ollinger and Heribert Vollmer, editors, 33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France, volume 47 of LIPIcs, pages 35:1–35:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

15. Gottlob Frege. Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache der reinen Denkens, Halle 1879. 1967. English translation in: from Frege to Godel, a source book in mathematical logic (J. van Heijenoord editor), Harvard University Press, Cambridge.

16. Marijn J. H. Heule, Martina Seidl, and Armin Biere. Solution validation and extraction for QBF preprocessing. J. Autom. Reason., 58(1):97–125, 2017.

17. Mikoláš Janota and João Marques-Silva. Expansion-based QBF solving versus q-resolution. Theor. Comput. Sci., 577:25–42, 2015.

18. Mikoláš Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. Theoretical Computer Science, 577:25–42, 2015.

19. Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified Boolean formulas. Information and Computation, 117(1):12–18, 1995.

20. Jan Krajícek. Proof Complexity. Cambridge University Press, 2019.

21. Jan Krajícek and Pavel Pudlák. Quantified propositional calculi and fragments of bounded arithmetic. Math. Log. Q., 36(1):29–46, 1990.

22. Meena Mahajan and Anil Shukla. Level-ordered Q-resolution and tree-like Q-resolution are incomparable. Inf. Process. Lett., 116(3):256–258, 2016.

23. John Alan Robinson. Theorem-proving on the computer. J. ACM, 10(2):163–174, 1963.

24. Nathan Segerlind. The complexity of propositional proofs. Bull. Symb. Log., 13(4):417–481, 2007.

25. Anil Shukla. On proof complexity for quantified Boolean formulas. PhD thesis, The Institute of Mathematical Sciences, Chennai, 2017.

26. Allen Van Gelder. Contributions to the theory of practical quantified Boolean formula solving. In Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming (CP), pages 647–663, 2012.