Electronic Colloquium on Computational Complexity, Report No. 172 (2018)

Building Strategies *into* QBF Proofs

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

Joshua Blinkhorn
Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany
scjlb@leeds.ac.uk

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

Abstract

Strategy extraction is of paramount importance for quantified Boolean formulas (QBF), both in solving and proof complexity. It extracts (counter)models for a QBF from a run of the solver resp. the proof of the QBF, thereby allowing to certify the solver's answer resp. establish soundness of the system. So far in the QBF literature, strategy extraction has been algorithmically performed *from* proofs. Here we devise the first QBF system where (partial) strategies are built *into* the proof and are piecewise constructed by simple operations along with the derivation.

This has several advantages: (1) lines of our calculus have a clear semantic meaning as they are accompanied by semantic objects; (2) partial strategies are represented succinctly (in contrast to some previous approaches); (3) our calculus has strategy extraction by design; and (4) the partial strategies allow new sound inference steps which are disallowed in previous central QBF calculi such as Q-Resolution and long-distance Q-Resolution.

The last item (4) allows us to show an exponential separation between our new system and the previously studied reductionless long-distance resolution calculus, introduced to model QCDCL solving.

Our approach also naturally lifts to dependency QBFs (DQBF), where it yields the first sound and complete CDCL-type calculus for DQBF, thus opening future avenues into DQBF CDCL solving.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases QBF, DQBF, resolution, proof complexity

1 Introduction

Proof complexity investigates the resources for proving logical theorems, focussing foremost on the minimal size of proofs needed in a particular calculus. Since its inception the field has enjoyed strong connections to computational complexity (cf. [14, 17]) and to first-order logic [16, 26]).

During the past decade, proof complexity has emerged as a key tool to model and analyse advances in the algorithmic handling of hard problems such as SAT and beyond. While traditionally perceived as a computationally hard problem, SAT solvers have been enormously successful in tackling huge industrial instances [29, 39] and hard combinatorial problems [21]. As each run of a solver on an unsatisfiable formula can be understood as a proof of unsatisfiability, each solver implicitly defines a proof system. This connection turns proof complexity into the main theoretical approach towards understanding the power and limitations of solving, with bounds on proof size directly corresponding to bounds on solver running time [14, 30].

ISSN 1433-8092

The algorithmic success story of solving has not stopped at SAT, but is currently extending to even more computationally complex problems such as *quantified Boolean formulas* (QBF), which is PSPACE complete, and *dependency QBFs* (DQBF), which is even NEXP complete. While quantification does not increase expressivity, (D)QBFs can encode many problems far more succinctly, including application domains such as automated planning [15, 18], verification [5, 28], synthesis [20, 27] and ontologies [25].

The past 15 years have seen *huge advances in QBF solving*, which currently reaches the point of industrial applicability. While some of the main innovations in SAT solving, including the development of conflict-driven clause learning (CDCL), revolutionised SAT in the late 1990s [37], this development in QBF is happening *now*. Consequently, QBF proof complexity has received considerable attention in recent years.

Compared with QBF, solving in DQBF is at its very beginnings, both in implementations (2018 was the first year that saw a DQBF track in the QBF competition [1]) as well as in its accompanying theory [36].

Strategy extraction is one of the distinctive features of QBF and DQBF, manifest in both solving and proof complexity. For solving it guarantees that together with the true/false answer the (D)QBF solver can produce a model (resp. countermodel) of the (D)QBF, thus *certifying* the correctness of the answer.

On the proof complexity side, this implies that proof calculi modelling QBF solving should allow strategy extraction in the sense that from a refutation of false QBF, a countermodel of the QBF can be efficiently constructed. This feature — without analogue in the propositional domain — enables strong lower bound techniques in QBF proof complexity [8, 9, 11], exploiting the fact that formulas requiring hard strategies cannot have short proofs in calculi with efficient strategy extraction.

As in SAT versus propositional proof complexity, one of the prime challenges in QBF and DQBF is to create compelling proof-theoretic models that capture central features of (D)QBF solving and at the same time remain amenable to a proof-theoretic analysis. While there exist several orthogonal approaches in QBF solving with quite different associated proof calculi, we will focus here on the paradigm of conflict-driven clause learning in QBF (QCDCL) [40]. Proof-theoretically its most basic model is Q-Resolution [23], which as in propositional resolution operates on clauses (of prenex QBFs).

Q-Resolution (Q-Res) uses the resolution rule of propositional resolution and augments this with a universal reduction rule that allows to eliminate universal variables from clauses. Combining these two rules requires some technical care: without any side-conditions the two rules result in an unsound system. Typically this is circumvented by prohibiting the derivation of universal tautologies. It was noted early on, that in solving this is needlessly prohibitive [40] and universal tautologies can be permitted under certain side-conditions. Later formalised as the proof system *long-distance Q-Resolution* (LD-Q-Res) [3], it was even shown that LD-Q-Res exponentially shortens proofs in comparison to Q-Res [19], thus demonstrating the appeal of the approach for solving. In fact, when enabling long-distance steps in QBF solving, universal reduction is not strictly needed and this reductionless approach was adopted in the QBF solver GhostQ [24]. To model this solving paradigm, Bjørner, Janota, and Klieber [13] introduced the calculus of *reductionless* LD-Q-Res.

The interplay between long-distance resolution and universal reduction steps becomes even more intriguing in DQBF. In [2] it was shown that lifting Q-Res (using the rules of resolution and universal reduction) to DQBF results in an incomplete proof system, whereas lifting LD-Q-Res (using long-distance resolution steps together with universal reduction) becomes unsound [12].

Naturally, the intriguing question of why and how deriving 'universal tautologies' in long-distance steps might help solving has attracted attention among theoreticians and practitioners alike. Instead of a universal tautology $u \vee \bar{u}$, most formalisations of long-distance resolution actually use the concept of a 'merged' literal u^*. While it is clear (and implicit in the literature) that merged literals u^* correspond to partial strategies for u rather than universal tautologies, a formal semantic account of long-distance steps (and stronger calculi using merging [10]) was only recently given by Suda and Gleiss [38], where partial strategies are constructed for each individual proof inference. However, as already noted in [38], the models considered in [38] fail to have efficient strategy extraction in the sense that the constructed (partial) strategies may need exponential-size representations.

Our contributions

A. The new calculus of Merge Resolution. Starting from the reductionless LD-Q-Res system of [13] and its role of modelling QCDCL solving, we develop a new calculus that we call Merge Resolution (M-Res). Like reductionless LD-Q-Res, the system M-Res only uses a resolution rule and does not permit universal reduction steps. Reductionless LD-Q-Res and M-Res are therefore both refutational calculi that finish as soon as they derive a purely universal clause.

As the prime novel feature of M-Res we build partial strategies *into* proofs. We achieve this by computing explicit representations of strategies in a variant of binary decision diagrams (called *merge maps* here), which are updated and refined at each proof step by simple operations. These merge maps are part of the proof. As a consequence, M-Res has efficient strategy extraction by design.

This is in contrast to all previous existing QBF calculi in the literature, where strategies are algorithmically constructed *from* proofs. In particular, this also applies to the approaches taken in [19, 38] for LD-Q-Res and in [13] for reductionless LD-Q-Res. But also the choice of our representation as merge maps matters: as [13, 38] both represent (partial) strategies as trees, the constructed strategies may grow exponentially in the proof size, thus losing the property of efficient strategy extraction desired for practice. In contrast, in our model merge maps are always linear in the size of the clause derivations.

B. Exponential separation of M-Res from reductionless LD-Q-Res. Including merge maps explicitly into proofs also has another far-reaching advantage: it allows resolution steps not only forbidden in Q-Res, but even disallowed in LD-Q-Res. In a nutshell, LD-Q-Res allows resolution steps only when universal variables quantified left of the pivot have *constant and equal* strategies in both parent clauses. In M-Res we have explicit representations of strategies and thus can allow resolution steps as long as the strategies in both parent clauses are *isomorphic* to each other, a property that we can check efficiently for merge maps.

This last mentioned advantage of allowing resolution steps in M-Res forbidden in (reductionless) LD-Q-Res manifests in shorter proofs. We show this by explicitly giving an example of a family of QBFs that admit linear-size proofs in M-Res (Theorem 27), but require exponential size in reductionless LD-Q-Res (Theorem 26). The separating formulas are a variant of the equality formulas introduced in [8]. While the original formulas from [8] are hard for Q-Res, but easy in LD-Q-Res, we here consider a 'squared' version, for which we naturally use resolution steps for clauses with associated non-constant winning strategies, allowed in M-Res, but forbidden in LD-Q-Res.

This shows that M-Res is exponentially stronger than reductionless LD-Q-Res, thus also pointing towards potential improvements in QCDCL solving. While the simulation of reductionless LD-Q-Res by M-Res is almost immediate and also the upper bound in M-Res is comparatively straightforward, the lower bound is a technically involved argument specifically tailored towards the squared equality formulas.

C. A sound and complete CDCL calculus for DQBF. As our final contribution we show that the new QBF system of M-Res naturally lifts to a sound and complete calculus for DQBF. As shown in [2], the lifting of Q-Res to DQBF is incomplete, whereas the combination of universal reduction and long-distance steps presents soundness issues, both in DQBF [12] as well as in the related framework of dependency schemes [6, 7].

Here we show that our framework of M-Res overcomes both these soundness and completeness issues and therefore has exactly the right strength for a natural DQBF resolution calculus. In fact, it is the first DQBF CDCL-type system in the literature[1] and as such paves the way towards CDCL solving in DQBF. Again, by design our DQBF system has efficient strategy extraction.

2 Preliminaries

Propositional logic. Let $\mathcal{Z}$ be a countable set of Boolean variables. A *literal* is a Boolean variable $z \in \mathcal{Z}$ or its negation $\bar{z}$, a *clause* is a set of literals, and a *CNF* is a set of clauses. For a literal l, we define $\mathrm{var}(l) := z$ if $l = z$ or $l = \bar{z}$; for a clause C, we define $\mathrm{vars}(C) := \{\mathrm{var}(l) : l \in C\}$; for a CNF ϕ we define $\mathrm{vars}(\phi) := \cup_{C\in\phi}\mathrm{vars}(C)$. An assignment to a set $Z \subseteq \mathcal{Z}$ of Boolean variables is a function $\rho : Z \to \{0, 1\}$, conventionally represented as a set of literals in which z (resp. $\bar{z}$) represents the assignment $z \mapsto 1$ (resp. $z \mapsto 0$). The set of all assignments to Z is denoted $\langle Z\rangle$. Given a subset $Z' \subseteq Z$, $\rho\restriction_{Z'}$ is the restriction of ρ to Z'. The CNF $\phi[\rho]$ is obtained from ϕ by removing any clause containing a literal in ρ, and removing the negated literals $\{\bar{l} : l \in \rho\}$ from the remaining clauses. We say that ρ *falsifies* ϕ if $\phi[\rho]$ contains the empty clause, and that ϕ is *unsatisfiable* if it is falsified by each $\rho \in \langle Z\rangle$.

Given two clauses R_1 and R_2 and a literal l such that $l \in R_1$ and $\bar{l} \in R_2$, we define the resolvent $\mathrm{res}(R_1, R_2, l) := (R_1 \setminus \{l\}) \cup (R_2 \setminus \{\bar{l}\})$. (Note that $\mathrm{res}(R_1, R_2, l) = \mathrm{res}(R_2, R_1, \bar{l})$.) A *resolution refutation* of a CNF ϕ is a sequence $C_1, \dots, C_k$ of clauses in which C_k is the empty clause and, for each $i \in [k]$, either (a) $C_i \in \phi$ or (b) $C_i = \mathrm{res}(C_a, C_b, z)$ for some $a, b < i$ and $z \in \mathrm{vars}(\phi)$.

Quantified Boolean formulas. A *quantified Boolean formula* (QBF) in *prenex conjunctive normal form* (PCNF) is denoted $\Phi := \mathcal{Q} \cdot \phi$, where (a) $\mathcal{Q} := \mathcal{Q}_1 Z_1 \cdots \mathcal{Q}_n Z_n$ is the *quantifier prefix*, in which the $Z_i \subset \mathcal{Z}$ are pairwise disjoint finite sets of Boolean variables, $\mathcal{Q}_i \in \{\exists, \forall\}$ for each $i \in [n]$, and $\mathcal{Q}_i \neq \mathcal{Q}_{i+1}$ for each $i \in [n-1]$, and (b) the *matrix* ϕ is a CNF over $\mathrm{vars}(\Phi) := \bigcup_{i=1}^n Z_i$.

The existential (resp. universal) variables of Φ, typically denoted X (resp. U), is the set obtained as the union of the Z_i for which $\mathcal{Q}_i = \exists$ (resp. $\mathcal{Q}_i = \forall$). The prefix $\mathcal{Q}$ defines a binary relation $<_\mathcal{Q}$ on $\mathrm{vars}(\Phi)$, such that $z <_\mathcal{Q} z'$ 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 $L_\mathcal{Q}(u) := \{x \in X : x <_\mathcal{Q} u\}$, i.e. the existential variables left of u.

---

[1] Previous DQBF resolution systems either use expansion [12] or extension variables [34].

QBF semantics. Semantics for QBFs are neatly described by the *two-player evaluation game*. Over the course of a game, the variables of a QBF $\mathcal{Q} \cdot \phi$ are assigned 0/1 values in the order of the prefix, with the $\exists$-player ($\forall$-player) choosing the values for the existential (universal) variables. When the game concludes, the players have constructed a total assignment ρ to the variables. The $\forall$-player wins iff ρ falsifies ϕ.

A *strategy* dictates how the $\forall$-player should respond to every possible move of the $\exists$-player. A strategy h for a QBF Φ is a set $\{h_u : u \in U\}$ of functions $h_u : \langle L_{\mathcal{Q}}(u)\rangle \to \{u, \bar{u}\}$. Additionally h is *winning* if, for each $\alpha \in \langle X\rangle$, the restriction of ϕ by $\alpha \cup \{h_u(\alpha\restriction_{L_{\mathcal{Q}}(u)}) : u \in U\}$ contains the empty clause. 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.

QBF proof systems. We deal with line-based refutational QBF systems that typically employ axioms and inference rules to prove the falsity of QBFs. We say that P is *complete* if there exists a P refutation of every false QBF, *sound* if there exists no P refutation of any true QBF. We call P a *proof system* if it is sound, complete, and polynomial-time checkable. Given two QBF proof systems P_1 and P_2, P_1 *p-simulates* P_2 if there exists a polynomial-time procedure that takes a P_2-refutation and outputs a P_1-refutation of the same QBF [17].

3 Reductionless long-distance Q-Resolution

In this section we recall the definition of reductionless LD-Q-Res, prove that it is refutationally complete, and demonstrate that it does not have polynomial-time strategy extraction in either of the computational models of [13, 38]. The system appeared first in [13, Fig. 1], where it was referred to as Q^w-resolution.

▶ **Definition 1** (reductionless LD-Q-Res [13])**.** A *reductionless* LD-Q-Res *derivation* from a QBF $\Phi := \mathcal{Q} \cdot \phi$ is a sequence $\pi := C_1, \dots, C_k$ of clauses in which at least one of (a) or (b) holds for each $i \in [k]$:

(a) **Axiom.** C_i is a clause from the matrix ϕ;

(b) **Long-distance resolution.** There exist integers $a, b < i$ and an existential pivot $x \in X$ such that $C_i = \mathrm{res}(C_a, C_b, x)$ and, for each $u \in \mathrm{vars}_\forall(C_a) \cap \mathrm{vars}_\forall(C_b)$, if $u <_{\mathcal{Q}} x$, then $\{u, \bar{u}\} \nsubseteq C_i$.

The final clause C_k is the *conclusion* of π, and π is a *refutation* of Φ iff C_k contains no existential variables.

A pair of complementary universal literals $\{u, \bar{u}\}$ appearing in a clause is referred to singly as a *merged literal*. It is clear from a wealth of literature[2] that merged literals are 'placeholders' for partial strategies, the exact representation left implicit in the structure of the derivation.

We illustrate the rules of the calculus by showing that the equality formulas [8] have linear-size refutations.

▶ **Definition 2** (equality formulas [8])**.** The *equality family* is the QBF family whose n^{th} instance has prefix $\exists\{x_1, \dots, x_n\}\forall\{u_1, \dots, u_n\}\exists\{t_1, \dots, t_n\}$ and matrix consisting of the clauses $\{x_i, u_i, t_i\}, \{\bar{x}_i, \bar{u}_i, t_i\}$ for $i \in [n]$, and $\{\bar{t}_1, \dots, \bar{t}_n\}$.

[2] The notion is evident to a greater or lesser degree in all of the papers [4, 7, 19, 31, 33, 38].

▶ **Example 3.** We construct linear-size reductionless LD-Q-Res refutations in two stages. First, resolve each pair $\{x_i, u_i, t_i\}$, $\{\bar{x}_i, \bar{u}_i, t_i\}$ of clauses over pivot x_i to obtain $C_i := \{u_i, \bar{u}_i, t_i\}$. Note that it is allowed to introduce the merged literal $\{u_i, \bar{u}_i\}$ since variable u_i is right of the pivot x_i. Second, resolve the C_i successively against the long clause $\{\bar{t}_1, \dots, \bar{t}_n\}$ over pivot t_i, to obtain a full set of merged literals $C := \{u_i, \bar{u}_i : i \in [n]\}$. Here, even though u_i is left of the pivot t_i, the appearance of the merged literal $\{u_i, \bar{u}_i\}$ in the resolvent is allowed, since variable u_i is absent from one of the antecedents. The derivation is a refutation since the conclusion C contains no existential literals.

Given a false QBF Φ with a countermodel h, we construct a canonical reductionless LD-Q-Res refutation based on the 'full binary tree' representation of a countermodel [35]. For each $\alpha \in \langle X \rangle$, there exists some C_α in the matrix falsified by $\alpha \cup h(\alpha)$. The set of all such C_α may be successively resolved over existential pivots in reverse prefix order, finally producing a clause containing no existentials. Merged literals never block resolution steps in this construction, as they only ever appear to the right of the pivot variable.

▶ **Lemma 4.** *Every false QBF has a reductionless* LD-Q-Res *refutation.*

Proof. Let $\Phi := \mathcal{Q} \cdot \phi$ be a false QBF with countermodel h. Denote the existential variables of Φ by $\{x_1, \dots, x_n\}$, such that whenever $i < j$ holds, there is no universal u such that $x_j <_{\mathcal{Q}} u <_{\mathcal{Q}} x_i$. Let $\alpha_1, \dots, \alpha_{2^n}$ define the natural lexicographic ordering of the total assignments to X, as in

$$\begin{array}{ccccc}
\alpha_1 & = & \bar{x}_1 \cdots \bar{x}_{n-2}\, \bar{x}_{n-1}\, \bar{x}_n & \approx & 0 \cdots 000\,, \\
\alpha_2 & = & \bar{x}_1 \cdots \bar{x}_{n-2}\, \bar{x}_{n-1}\, x_n & \approx & 0 \cdots 001\,, \\
\alpha_3 & = & \bar{x}_1 \cdots \bar{x}_{n-2}\, x_{n-1}\, \bar{x}_n & \approx & 0 \cdots 010\,, \\
\alpha_4 & = & \bar{x}_1 \cdots \bar{x}_{n-2}\, x_{n-1}\, x_n & \approx & 0 \cdots 011\,, \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\alpha_{2^n} & = & x_1 \cdots x_{n-2}\, x_{n-1}\, x_n & \approx & 1 \cdots 111\,.
\end{array}$$

We define a sequence $\pi := \pi_n \circ \cdots \circ \pi_0$ in which each $\pi_i := C^i_1, \dots, C^i_{2^i}$, and the clauses C^i_j are defined recursively as follows: For $j \in [2^n]$, C^n_j is any clause in ϕ falsified by $\alpha_j \cup h(\alpha_j)$ (at least one such clause exists by definition of countermodel); for $i \in [n]$ and $j \in [2^{i-1}]$, $C^{i-1}_j := \mathrm{res}(C^i_{2j-1}, C^i_{2j}, x_i)$ if this resolvent exists, otherwise

$$C^{i-1}_j := \begin{cases} C^i_{2j-1}\,, & \text{if } x_i \notin C^i_{2j-1}\,, \\ C^i_{2j}\,, & \text{if } \bar{x}_i \notin C^i_{2j}\,. \end{cases}$$

It is readily verified by downwards induction on $i \in [n]$ that each C^i_j contains no complementary universal literals in variables left of x_i. Moreover, it is easy to see that the conclusion C^0_1 contains no existential literals. Removing duplicate clauses from π produces a reductionless LD-Q-Res refutation of Φ. ◀

Soundness and polynomial-time checkability of reductionless LD-Q-Res are immediate, as the system uses a subset of the rules of the classical long-distance Q-resolution proof system [3].

The computational model of Bjørner et al. [13]. In tandem with reductionless LD-Q-Res, the authors of [13] introduced a computational model based on tree-like branching programs. The model is used to explicitly construct the partial strategies represented implicitly by merged literals.

We demonstrate that tree-like branching programs constructed in this way cannot represent strategies efficiently; that is, the system does not have polynomial-time strategy extraction in the associated model (even for partial strategies). The following example shows a linear-size derivation whose explicit strategy grows exponentially large.

▶ **Example 5.** Consider the following proof fragment, in the reductionless LD-Q-Res proof system, with a prefix $\exists v \exists x \exists w \forall u \exists y \exists z$. Alongside each proof line is the strategy for the universal variable u, as built by the `Build` function in [13].

Line	Obtained as	Clause	Strategy as built in [13].
C_1	axiom	$\{w, x, u\}$	0
C_2	axiom	$\{\bar{w}, x, \bar{u}\}$	1
C_3	$\text{res}(C_1, C_2, w)$	$\{x, u, \bar{u}\}$	$w ? 1 : 0$
C_4	axiom	$\{\bar{x}, u, y\}$	0
C_5	$\text{res}(C_3, C_4, x)$	$\{u, \bar{u}, y\}$	$x ? 0 : [w ? 1 : 0]$
C_6	axiom	$\{v, \bar{y}\}$	$*$
C_7	$\text{res}(C_5, C_6, y)$	$\{v, u, \bar{u}\}$	$x ? 0 : [w ? 1 : 0]$
C_8	axiom	$\{\bar{x}, z\}$	$*$
C_9	$\text{res}(C_3, C_8, x)$	$\{u, \bar{u}, z\}$	$w ? 1 : 0$
C_{10}	axiom	$\{\bar{v}, \bar{z}\}$	$*$
C_{11}	$\text{res}(C_9, C_{10}, z)$	$\{\bar{v}, u, \bar{u}\}$	$w ? 1 : 0$
C_{12}	$\text{res}(C_7, C_{11}, v)$	$\{u, \bar{u}\}$	$v ? (w ? 1 : 0) : (x ? 0 : [w ? 1 : 0])$

Observe that the final strategy at line 12 represents the strategy corresponding to line 3 twice. By nesting such a proof fragment from lines C_3 to C_{12} with fresh copies of the existential variables (v, x, y, z) k times, we can construct a reductionless LD-Q-Res proof fragment with $O(k)$ lines, where the strategy built by the `Build` function from [13] has size exponential in k.

The computational model of Suda and Gleiss [38]. The authors of [38] proposed a model of partial strategies based on so-called *policies*. They noted that the equality formulas have linear-size refutations in the strong QBF system IRM-calc [10], whereas policies witnessing their falsity must be exponentially large, therefore IRM-calc does not admit polynomial-time strategy in policies. The same is true for reductionless LD-Q-Res, since Example 3 shows that the equality formulas also have linear-size refutations there.

The computational model of policies is not even suitable for strategy extraction in the weak system level-ordered Q-Res [22].[3] Versions of the equality formulas in which the prefix is rearranged ($\exists x_1 \forall u_1 \exists t_1 \cdots \exists x_n \forall u_n \exists t_n$) have linear-size level-ordered Q-Res refutations, whereas winning strategies represented as policies must be large. The argument is the same as for the equality formulas [38], and derives from the implicit use of tree-like structures.

That neither model is suitable for efficient strategy extraction shows that using either *inside* the derivation would result in an artificial, exponential size blow-up. The root of the issue is tree-like models versus DAG-like proofs. The DAG-like computational model that we introduce in the following section is tightly knitted to the refutation, yielding linear-time strategy extraction for free.

[3] Reductionless LD-Q-Res p-simulates level-ordered Q-Res by means of a simple construction, and is exponentially separated by the equality formulas [8].

4 Merge Resolution

In this section we introduce Merge Resolution (M-Res, Subsection 4.2), and prove that it is sound and complete for QBF (Subsection 4.3). The salient feature of M-Res is the built-in partial strategies, represented as *merge maps*. Given the problems with the computational models of [13, 38], the principal technical challenge is to find a suitable way to define and combine partial strategies devoid of an artifical proof-size inflation.

4.1 Merge maps

Our computational model. A merge map is a branching program that queries a set of existential variables and outputs an assignment to some universal variable, i.e. a literal in $\{u, \bar{u}, *\}$, where $*$ stands for 'no assignment'. As we intend to tie the DAG structure of the merge maps to the DAG structure of the proof, we will label query nodes with natural numbers based on the proof line indexing (we elaborate on this later). Hence, from a technical standpoint it makes sense to define a merge map as a function from the index set of its nodes.

▶ **Definition 6** (merge map). A *merge map* M for a Boolean variable u over a finite set X of Boolean variables is a function from a finite set N of natural numbers satisfying, for each $i \in N$, either $M(i) \in \{u, \bar{u}, *\}$ or $M(i) \in X \times N_{<i} \times N_{<i}$, where $N_{<i} := \{i' \in N : i' < i\}$.

A triple of the form $(x, a, b) \in X \times N_{<i} \times N_{<i}$ represents the instruction 'if $x = 0$ then goto a else goto b', whereas the literals $\{u, \bar{u}, *\}$ represent output values. The exact computation is formalised below.

▶ **Definition 7** (computed function). Let M be a merge map for u over X with domain N. The *function computed by* M is the function $h : \langle X \rangle \to \{u, \bar{u}, *\}$ mapping α to the output of the following algorithm:

$i := \max(N)$
while $M(i) \notin \{u, \bar{u}, *\}$
$(x, a, b) := M(i)$
  if $\bar{x} \in \alpha$ **then** $i := a$ **else** $i := b$
return $M(i)$

We depict merge maps pictorially as DAGs. The nodes are the domain elements, and the leaf nodes as well as the directed edges are labelled by literals. In a merge map M, if $M(i)$ is a literal l, then node i is labeled l. If $M(i) = (x, a, b)$, then the DAG has the edge $i \to a$ labeled $\bar{x}$ and the edge $i \to b$ labeled x. The DAG naturally describes a deterministic branching program computing a Boolean function.

Figure 1 shows a merge map represented as a function, and its corresponding depiction as a branching program.

Relations. Merge Resolution uses two relations to determine preconditions for the binary operations. Firstly, we give M-Res the power to identify merge maps with equivalent representations, up to indexing. We term equivalent representations 'isomorphic'.

▶ **Definition 8** (isomorphism). Two merge maps M_1 and M_2 for u over X with domains N_1 and N_2 are *isomorphic* (written $M_1 \simeq M_2$) iff there exists a bijection $f : N_1 \to N_2$ such that the following hold for each $i \in N_1$:

$$
\begin{array}{rcrcl}
M & : & 1 & \mapsto & u \\
 & & 2 & \mapsto & \bar{u} \\
 & & 3 & \mapsto & (w, 1, 2) \\
 & & 4 & \mapsto & * \\
 & & 5 & \mapsto & (w, 4, 2) \\
 & & 6 & \mapsto & (v, 5, 3)
\end{array}
$$

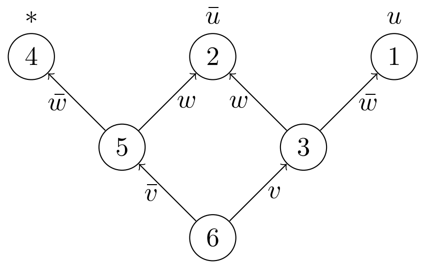


Figure 1 Function and branching program representations of a merge map M.

(a) if $M_1(i)$ is a literal in $\{u, \bar{u}, *\}$ then $M_2(f(i)) = M_1(i)$;
(b) if $M_1(i)$ is the triple (x, a, b) then $M_2(f(i)) = (x, f(a), f(b))$.

▶ Proposition 9. Any two isomorphic merge maps compute the same function.

Proof. Let M_1 and M_2 be merge maps, let f be a bijection satisfying the properties of Definition 8, and let $\alpha \in \mathrm{dom}(M_1)$. The computation of $M_2(\alpha)$ as in Definition 7 is identical to that of M_1, except that each natural number $i \in \mathrm{dom}(M_1)$ is replaced with $f(i)$. The proposition follows. ◀

Our second relation, consistency, simply identifies whether or not two merge maps agree on the intersection of their domains.

▶ **Definition 10** (consistency)**.** Two merge maps M_1 and M_2 for u over X with domains N_1 and N_2 are *consistent* (written $M_1 \bowtie M_2$) iff $M_1(i) = M_2(i)$ for each $i \in N_1 \cap N_2$.

It is easy to see that both relations can be computed in time linear in $\max(N_1 \cup N_2)$.

Operations. M-Res uses two binary operations to build merge maps for the resolvent based on those of the antecedents. We define the operations and give some intuition on their role in M-Res. Concrete examples follow the definition of the system in the next subsection.

The *select* operation identifies equivalent merge maps by means of the isomorphism relation. It also allows a *trivial* merge map to be discarded; we call a merge map trivial iff it is isomorphic to $1 \mapsto *$. (The operations is undefined if the merge maps are neither isomorphic nor do they contain a trivial map.)

▶ **Definition 11** (select)**.** Let M_1 and M_2 be merge maps for which $M_1 \simeq M_2$ or one of M_1, M_2 is trivial. Then $\mathrm{select}(M_1, M_2) := M_2$ if M_1 is trivial, and $\mathrm{select}(M_1, M_2) := M_1$ otherwise.

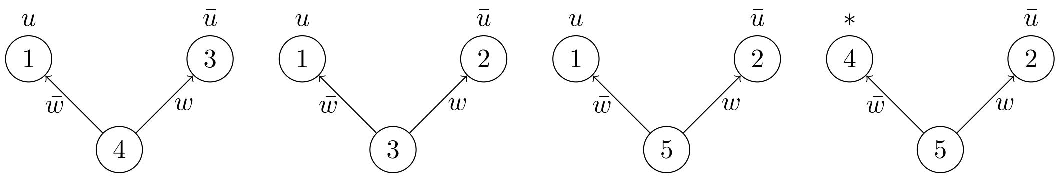
The *merge* operation allows two consistent merge maps to be combined as the children of a fresh query node. Antecedent maps are only ever merged for universal variables right of the pivot x. The inclusion of a natural number n allows the new query node to be identified with the resolvent, via its index in the proof sequence. In this way, query nodes are shared between later merge maps, rather than being duplicated; the result is a DAG-like structure which faithfully follows that of the derivation.

▶ **Definition 12** (merge)**.** Let M_1 and M_2 be consistent merge maps for u over X with domains N_1 and N_2, let $n > \max(N_1 \cup N_2)$ be a natural number, and let $x \in X$. Then merge(M_1, M_2, n, x) is the function from $N_1 \cup N_2 \cup \{n\}$ defined by

$$\text{merge}(M_1, M_2, n, x)(i) := \begin{cases} (x, \max(N_1), \max(N_2)) & \text{if } i = n, \\ M_1(i) & \text{if } i \in N_1, \\ M_2(i) & \text{if } i \in N_2 \setminus N_1. \end{cases}$$

▶ **Example 13.** For the merge maps depicted in Figure 2, isomorphism and consistency (or lack thereof) are as given in the table below. Furthermore, note that select$(A, B) =$ select$(A, C) = A$ and merge$(D, B, 6, v)$ gives the merge map from Figure 1.

relation	isomorphic	not isomorphic
consistent	$A \bowtie C$; $A \simeq C$	$B \bowtie D$; $B \not\simeq D$
not consistent	$A \not\bowtie B$; $A \simeq B$	$C \not\bowtie D$; $C \not\simeq D$



(a) Merge map A **(b)** Merge map B **(c)** Merge map C **(d)** Merge map D

Figure 2 Relations and operations on merge maps.

4.2 Definition of M-Res

We are now ready to put down the rules of Merge Resolution. Given a non-tautological clause C and a Boolean variable u, the *falsifying u-literal for C* is $\bar{l}$ if there is a literal $l \in C$ with var$(l) = u$, and $*$ otherwise.

▶ **Definition 14** (merge resolution). Let $\Phi := \mathcal{Q} \cdot \phi$ be a QBF with existential variables X and universal variables U. A *merge resolution* (M-Res) *derivation* of L_k from Φ is a sequence $\pi := L_1, \ldots, L_k$ of lines $L_i := (C_i, \{M_i^u : u \in U\})$ in which at least one of the following holds for each $i \in [k]$:

(a) **Axiom.** There exists a clause in $C \in \phi$ such that C_i is the existential subclause of C, and, for each $u \in U$, M_i^u is the merge map for u over $L_{\mathcal{Q}}(u)$ with domain $\{i\}$ mapping i to the falsifying u-literal for C;

(b) **Resolution.** There exist integers $a, b < i$ and an existential pivot $x \in X$ such that $C_i = \text{res}(C_a, C_b, x)$ and, for each $u \in U$, either (i) $M_i^u = \text{select}(M_a^u, M_b^u)$, or (ii) $x <_{\mathcal{Q}} u$ and $M_i^u = \text{merge}(M_a^u, M_b^u, i, x)$.

The final line L_k is the *conclusion* of π, and π is a *refutation* of Φ iff $C_k = \emptyset$. The *size* of π is $|\pi| = k$.

Note that the order of the indexes a and b in merge(M_a, M_b, i, x) matches that of res(C_a, C_b, x). This is why we interpret the triple (x, a, b) as 'if $x = 0$ then goto a else goto b'. Using the conventional 'if $x = 1$' entails swapping the order of the arguments M_a and M_b.

We illustrate the rules of M-Res with two examples. The first demonstrates that labelling branching nodes with proof-line indexes sidesteps the exponential blow-up in the computational model of [13].

▶ **Example 15.** The reductionless LD-Q-Res proof fragment in Example 5 can be viewed as a proof in M-Res if we attach appropriate merge maps at each line.

Line	Rule	C_i	M_i	Query
L_1	axiom	$\{w, x\}$	$1 \mapsto \bar{u}$	
L_2	axiom	$\{\bar{w}, x\}$	$2 \mapsto u$	
L_3	$\text{res}(L_1, L_2, w)$	$\{x\}$	$\text{merge}(M_1, M_2, 3, w)$	$3 \mapsto (w, 1, 2)$
L_4	axiom	$\{\bar{x}, y\}$	$4 \mapsto \bar{u}$	
L_5	$\text{res}(L_3, L_4, x)$	$\{y\}$	$\text{merge}(M_3, M_4, 5, x)$	$5 \mapsto (x, 3, 4)$
L_6	axiom	$\{v, \bar{y}\}$	$6 \mapsto *$	
L_7	$\text{res}(L_5, L_6, y)$	$\{v\}$	$\text{select}(M_5, M_6) = M_5$	
L_8	axiom	$\{\bar{x}, z\}$	$8 \mapsto *$	
L_9	$\text{res}(L_3, L_8, x)$	$\{z\}$	$\text{select}(M_3, M_8) = M_3$	
L_{10}	axiom	$\{\bar{v}, \bar{z}\}$	$10 \mapsto *$	
L_{11}	$\text{res}(L_9, L_{10}, z)$	$\{\bar{v}\}$	$\text{select}(M_9, M_{10}) = \text{select}(M_3, M_{10}) = M_3$	
L_{12}	$\text{res}(L_7, L_{11}, v)$	$\{\}$	$\text{merge}(M_7, M_{11}, 12, v)$	$12 \mapsto (v, 5, 3)$
			$= \text{merge}(M_5, M_3, 12, v)$	

In lines L_7, L_9 and L_{11}, the use of select is allowed, since in each case one of the antecedent merge maps is trivial (i.e. isomorphic to $1 \mapsto *$). Notice that at line L_7, we could also have chosen M_7 to be $\text{merge}(M_5, M_6, 7, y)$; this would result in a larger merge map.

Now, consider the final merge map M_{12}. The corresponding branching program has isolated nodes numbered 6, 8, and 10; these can be removed, giving the pruned merge map shown in Figure 3. Notice how the size blow-up from Example 5 is avoided here; since M_3 and M_5 are consistent, node 12 simply points to both of them, and the shared part (that is, the branching program M_3 containing nodes 1, 2, and 3) is represented just once.

$$
M_{12} \;:\; \begin{array}{rcl} 1 & \mapsto & \bar{u} \\ 2 & \mapsto & u \\ 3 & \mapsto & (w, 1, 2) \\ 4 & \mapsto & \bar{u} \\ 5 & \mapsto & (x, 3, 4) \\ 12 & \mapsto & (v, 5, 3) \end{array}
$$

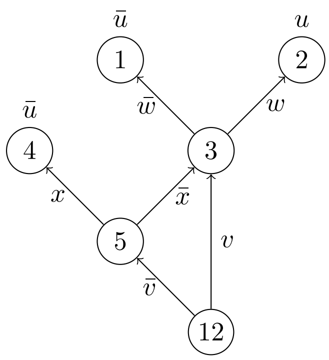


Figure 3 Function and branching program representations of M_{12} from Example 15.

Our second example illustrates how the explicit representation of strategies, in tandem with the isomorphism relation, gives M-Res access to resolution steps that are disallowed in reductionless LD-Q-Res.

▶ **Example 16.** Consider the following M-Res refutation of the QBF with prefix $\exists x \forall u \exists t$ and matrix consisting of the clauses $\{x, u, t\}$, $\{\bar{x}, \bar{u}, t\}$, $\{x, u, \bar{t}\}$ and $\{\bar{x}, \bar{u}, \bar{t}\}$.

Line	Rule	C_i	M_i	Query
L_1	axiom	$\{x, t\}$	$1 \mapsto \bar{u}$	
L_2	axiom	$\{\bar{x}, t\}$	$2 \mapsto u$	
L_3	$\mathrm{res}(L_1, L_2, x)$	$\{t\}$	$\mathrm{merge}(M_1, M_2, 3, x)$	$3 \mapsto (x, 1, 2)$
L_4	axiom	$\{x, \bar{t}\}$	$4 \mapsto \bar{u}$	
L_5	axiom	$\{\bar{x}, \bar{t}\}$	$5 \mapsto u$	
L_6	$\mathrm{res}(L_4, L_5, x)$	$\{t\}$	$\mathrm{merge}(M_4, M_5, 6, x)$	$6 \mapsto (x, 4, 5)$
L_7	$\mathrm{res}(L_3, L_6, t)$	$\{\}$	$\mathrm{select}(M_3, M_6) = M_3$	

As shown in Figure 4, M_3 and M_6 are isomorphic, so $\mathrm{select}(M_3, M_6)$ is defined and equal to M_3. For this reason, the resolution of antecedents L_3 and L_6 into L_7 is allowed, and the final merge map M_7 is simply a copy of M_3. The analogous resolution would be disallowed in reductionless LD-Q-Res because the pivot t is right of u, and the non-constant merge maps M_3 and M_6 would appear as merged literals $\{u, \bar{u}\}$ in the antecedent clauses.

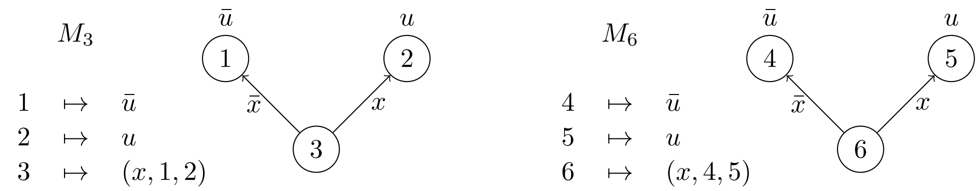


Figure 4 Functions and branching programs for merge maps M_3 and M_6 from Example 16.

We conclude this subsection by showing that the number of lines really is the correct size measure for Merge Resolution. The justification lies in the fact that the domain of the merge map at line i is a subset of $[i]$.

▶ Proposition 17. Let $(C_1, \{M_1^u : u \in U\}), \ldots, (C_k, \{M_k^u : u \in U\})$ be an M-Res refutation of $\mathcal{Q} \cdot \phi$. For each $u \in U$, $M_1^u, \ldots, M_n^u$ are pairwise consistent merge maps for u over $L_{\mathcal{Q}}(u)$ with $\max(\mathrm{dom}(M_i^u)) \leq i$ for each $i \in [n]$.

Proof. The proposition follows straightforwardly from three observations: (1) each M_i^u introduces at most one node, which is labelled i; (2) if L_i is an axiom, then each M_i^u is a merge map over $L_{\mathcal{Q}}(u)$; (3) the merge operation is only applied when $x \in L_{\mathcal{Q}}(u)$. ◀

4.3 Soundness and completeness of M-Res

The soundness of M-Res comes down to the fact that the merge maps at a given line form a partial strategy for the input QBF, in the technical sense of [38]. This means that any total existential assignment that falsifies the clause C_i will falsify the matrix when extended by the output of the merge maps M_i^u. Our proof of soundness is an induction on the proof structure with exactly this invariant. At the conclusion, all existential assignments falsify the empty clause C_k, and hence the M_k^u compute a countermodel. A trivial corollary, then, is that M-Res has linear strategy extraction in merge maps. Our formal proof of soundness is preceded by a preliminary proposition.

▶ Proposition 18. Let M_1 and M_2 be consistent merge maps for u over X with domains N_1 and N_2, let $n > \max(N_1 \cup N_2)$ be a natural number, let $x \in X$ and let $\alpha \in \langle X \rangle$. Further, let h_1, h_2 and h be the functions computed by M_1, M_2 and $\text{merge}(M_1, M_2, x, n)$. Then $h(\alpha) = h_1(\alpha)$ if $\bar{x} \in \alpha$, and $h(\alpha) = h_2(\alpha)$ if $x \in \alpha$.

Proof. Let $M := \text{merge}(M_1, M_2, n, x)$, and suppose that $\bar{x} \in \alpha$. By Definition 12, $M(n) = (x, \max(N_1), \max(N_2))$ and $M(i) = M_1(i)$ for each $i \in N_1$. Hence, the computation of $h(\alpha)$ from the second iteration of the while loop is identical to the computation of $h_1(\alpha)$ from the first iteration, and it follows that $h(\alpha) = h_1(\alpha)$. Suppose instead that $x \in \alpha$. By Definition 12, $M(i) = M_2(i)$ for each $i \in N_2 \setminus N_1$; by Definition 10, $M_1(i) = M_2(i)$ for each $i \in N_1 \cap N_2$. Then $M(i) = M_2(i)$ for each $i \in N_2$, and the proposition follows as in first case. ◀

▶ **Lemma 19.** *Let $(\emptyset, \{M^u : u \in U\})$ be the conclusion of an* M-Res *refutation of a QBF Φ. Then the functions computed by $\{M^u : u \in U\}$ form a countermodel for Φ.*

Proof. Let $\pi := L_1, \dots, L_k$ be an M-Res refutation of a QBF $\Phi := \mathcal{Q} \cdot \phi$, where each $L_i = (C_i, \{M_i^u : u \in U\})$. Further, for each $i \in [k]$,

- let $\alpha_i := \{\bar{l} : l \in C_i\}$ be the smallest assignment falsifying C_i,
- let $A_i := \{\alpha \in \langle X \rangle : C_i \cap \alpha = \emptyset\}$ be all 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 $l_i^u(\alpha) := h_i^u(\text{proj}(\alpha, L_{\mathcal{Q}}(u)))$ and $h_i(\alpha) := \{l_i^u(\alpha) : u \in U\} \setminus \{*\}$.

(Note that Proposition 17 guarantees that each h_i^u is defined.) By induction on $i \in [k]$, we show, for each $\alpha \in A_i$, that the restriction of ϕ by $\alpha \cup h_i(\alpha)$ contains the empty clause. Since α_k is the empty assignment, we have $A_k = \langle X \rangle$. We therefore prove the lemma at the final step $i = k$, as we show that $\{h_k^u : u \in U\}$ is a countermodel for Φ.

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 M_1^u is the merge map from $\{i\}$ mapping i to the falsifying u-literal for C. Hence, for each $u \in U$, $l_1^u(\alpha)$ 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 $C_i = \text{res}(C_a, C_b, x)$, and each $u \in U$ satisfies either (i) $M_i^u = \text{select}(M_a^u, M_b^u)$, or (ii) $x \in L_{\mathcal{Q}}(u)$, and $M_i^u = \text{merge}(M_a^u, M_b^u, i, x)$. Now, suppose on the one hand that $\bar{x} \in \alpha$, and let $u \in U$. If u satisfies (i) and M_a^u is non-trivial, then $l_i^u(\alpha) = l_a^u(\alpha)$, and if u satisfies (ii) then $l_i^u(\alpha) = l_a^u(\alpha)$ by Proposition 18. It follows that $l_i^u \neq l_a^u$ only if $l_a^u = *$, and hence $h_a(\alpha) \subseteq h_i(\alpha)$. Since $C_a \cup \{x\} \subseteq C_i$, we have $\alpha \in A_a$, so the restriction of ϕ by $\alpha \cup h_i(\alpha)$ contains the empty clause by the inductive hypothesis. Supposing, on the other hand, that $x \in \alpha$, a similar argument shows that $h_i(\alpha) \subseteq h_b(\alpha)$. Note that, in this case, if u satisfies (i) and M_b^u is non-trivial, then $M_a^u \simeq M_b^u$ and $l_i^u = l_a^u = l_b^u$ by Proposition 9. ◀

Completeness of M-Res is shown via the p-simulation of reductionless LD-Q-Res. The simulation copies precisely the structure of the reductionless LD-Q-Res refutation, while replacing merged literals by merge maps in the natural way.

▶ **Theorem 20.** M-Res *p-simulates reductionless* LD-Q-Res.

Proof. Let $\Phi := \mathcal{Q} \cdot \phi$ be a QBF with existential variables X and universal variables Y, and let $\pi := C_1, \ldots, C_k$ be a reductionless LD-Q-Res refutation of Φ. We define a sequence $\pi' := L_1, \ldots, L_n$, in which each $L_i := (C'_i, \{M_i^u : u \in U\})$, and prove that it is an M-Res refutation of Φ.

For each $i \in [k]$, we define C'_i to be the existential subclause of C_i. For each $u \in U$, the merge maps are defined recursively as follows: If C_i is an axiom, M_i^u is defined as the merge map over $L_{\mathcal{Q}}(u)$ with domain $\{i\}$ mapping i to the falsifying u-literal for C_i (note that this covers the definition of M_1^u). If C_i is derived by resolution, say $C_i = \text{res}(C_a, C_b, x)$ with $a, b < i$, then

$$M_i^u := \begin{cases} \text{select}(M_a^u, M_b^u)\,, & \text{if select}(M_a^u, M_b^u) \text{ is defined}\,, \\ \text{merge}(M_a^u, M_b^u, i, x)\,, & \text{otherwise}\,. \end{cases}$$

Now, by induction on $i \in [n]$, we prove that, for each $u \in U$,

(a) if $\{u, \bar{u}\} \nsubseteq C_i$, then M_i^u is isomorphic to $1 \mapsto l$, where l is the falsifying u-literal for C_i,
(b) L_i can be derived from previous lines in π' using an M-Res rule.

Both are established trivially when C_i is an axiom; hence it remains to show the inductive step in the case where C_i was derived by resolution. In this case $C_i = \text{res}(C_a, C_b, x)$ for some $a, b < i$ and some $x \in X$.

(a) Suppose that $\{u, \bar{u}\} \nsubseteq C_i$, and let l_i, l_a, l_b be the falsifying u-literals for C_i, C_a, C_b. By definition of resolution, either (1) $l_i = l_a = l_b$, or (2) exactly one of l_a, l_b is trivial (l_b, say), the other is equal to l_i. In the former case, M_a^u and M_b^u are both isomorphic to $1 \mapsto l_i$, by the inductive hypothesis; in the latter case, M_a^u is isomorphic to $1 \mapsto l_i$ and M_b^u is trivial. Either way we get $M_i^u = \text{select}(M_a^u, M_b^u) = M_a^u$, and the inductive step follows.
(b) By Proposition 17, for each $u \in U$, M_a^u and M_b^u are consistent merge maps for u over $L_{\mathcal{Q}}(u)$, so $\text{merge}(M_a^u, M_b^u, i, x)$ is defined for any case. Hence, if we can show that $\text{select}(M_a^u, M_b^u)$ is defined whenever $u <_{\mathcal{Q}} x$, then it is clear that L_i can be derived by resolution from L_a and L_b. To that end, let u be left of x. If $\{u, \bar{u}\} \nsubseteq C_i$, then $\text{select}(M_a^u, M_b^u)$ is defined by (a). Otherwise, we must have $u \notin \text{vars}(C_a) \cap \text{vars}(C_b)$, so the falsifying u-literal for one of C_a and C_b is $*$ By the inductive hypothesis, one of M_a^u and M_b^u is trivial, and $\text{select}(M_a^u, M_b^u)$ is defined.

This completes the induction. Since C_n contains only universal variables, C'_k is the empty clause, and π' is a refutation. ◀

With soundness and completeness established by Lemma 19 and Theorem 20, it remains to show that M-Res refutations can be checked in polynomial time. This is easy to see, since the isomorphism and consistency relations are computable in linear time.

▶ **Theorem 21.** M-Res *is a QBF proof system.*

5 Proof complexity: Merge Resolution vs Reductionless LD-Q-Res

In this section we exponentially separate M-Res from reductionless LD-Q-Res. The separating formulas are a kind of 'squaring' of the equality formulas from Definition 2.

▶ **Definition 22** (squared equality formulas). The *squared equality family* is the QBF family whose n^{th} instance $\mathrm{EQ}^2(n) := \mathcal{Q}(n) \cdot \mathrm{eq}^2(n)$ has prefix

$$\mathcal{Q}(n) := \exists\{x_1, y_1, \dots, x_n, y_n\}\forall\{u_1, v_1, \dots, u_n, v_n\}\exists\{t_{i,j} : i, j \in [n]\},$$

and CNF matrix $\mathrm{eq}^2(n)$ consisting of the clauses

$$\begin{array}{lll} \{x_i, y_j, u_i, v_j, t_{i,j}\}, & \{x_i, \bar{y}_j, u_i, \bar{v}_j, t_{i,j}\}, & \text{for } i, j \in [n], \\ \{\bar{x}_i, y_j, \bar{u}_i, v_j, t_{i,j}\}, & \{\bar{x}_i, \bar{y}_j, \bar{u}_i, \bar{v}_j, t_{i,j}\}, & \text{for } i, j \in [n], \\ \{\bar{t}_{i,j} : i, j \in [n]\}. & & \end{array}$$

The only winning strategy for the universal player is to set $u_i = x_i$ and $v_j = y_j$ for each $i, j \in [n]$. At the final block, the existential player is faced with the full set of $\{t_{i,j}\}$ unit clauses, and to satisfy all of them is to falsify the square clause $\{\bar{t}_{i,j} : i, j \in [n]\}$. No other strategy can be winning, as it would fail to produce all n^2 unit clauses.

5.1 $\mathbf{EQ}^2(n)$ lower bound for reductionless LD-Q-Res

We first give a formal definition of a refutation *path*; that is, a sequence of consecutive resolvents beginning with an axiom and ending at the conclusion.

▶ **Definition 23** (path). Let π be a reductionless LD-Q-Res refutation. A *path* from a clause C in π is a subsequence $C_1, \dots, C_k$ of π in which:

- $C = C_1$ is an axiom of π;
- C_k is the conclusion of π;
- for each $i \in [k-1]$, there exists a literal p_i and a clause R_i occurring before C_{i+1} in π such that $C_{i+1} = \mathrm{res}(C_i, R_i, p_i)$.

The lower-bound proof is based upon two facts: (1) every total existential assignment corresponds to a path, all of whose clauses are consistent with the assignment (Lemma 24); (2) every path from the square clause contains a 'wide' clause containing either all the x_i or all the y_j variables (Lemma 25). It is then possible to deduce the existence of exponentially many wide clauses, i.e. by considering the set of assignments for which each $x_i = y_i$ and each $t_{i,j} = 0$, all of whose corresponding paths begin at the square clause (proof of Theorem 26).

▶ **Lemma 24.** *Let π be a reductionless* LD-Q-Res *refutation of a QBF Φ, and let A be a clause with* $\mathrm{vars}(A) = \mathrm{vars}_\exists(\Phi)$. *Then there exists a path in π in which no existential literal outside of A occurs.*

Proof. We describe a procedure that constructs a sequence $P := C_k, \dots, C_1$ of clauses in reverse order as follows: To begin with, let the 'current clause' C_1 be the conclusion of π. As soon as the current clause C_i is in an axiom, the procedure terminates. Whenever necessary, obtain C_{i+1} as follows: find clauses R_1 and R_2 occurring before C_i in π and a literal $p \in A$ such that C_i is $\mathrm{res}(R_1, R_2, p)$, and set $C_{i+1} := R_1$ as the current clause. P is clearly a path in π by construction. By induction one shows that the existential subclause of C_i is a subset of A, for each $i \in [n]$: The base case $i = 1$ holds trivially since there are no existential literals in the conclusion C_1 of π. For the inductive step, observe that $C_{i+1} = C' \cup \{p\}$, for some subset $C' \subseteq C_i$ and literal $p \in A$. ◀

The second lemma is more technical, and its proof more involved. The proof works directly on the definition of path, the rules of reductionless LD-Q-Res, and the syntax of the squared equality formulas, to show the existence of the wide clause.

▶ **Lemma 25.** *Let $n \geq 2$, and let π be a reductionless* LD-Q-Res *refutation of* $\mathrm{EQ}^2(n)$. *On each path from $\{\bar{t}_{i,j} : i,j \in [n]\}$ in π, there occurs a clause C for which either $\{x_1, \dots, x_n\} \subseteq \mathrm{vars}(C)$ or $\{y_1, \dots, y_n\} \subseteq \mathrm{vars}(C)$.*

Proof. Put $X := \{x_1, \dots, x_n\}$ and $Y := \{y_1, \dots, y_n\}$. Call a clause R in π a *p-resolvent* if there exist earlier clauses R_1 and R_2 such that $R = \mathrm{res}(R_1, R_2, p)$.

Let $P := C_1, \dots, C_k$ be a path from $\{\bar{t}_{i,j} : i,j \in [n]\}$ in π. With each C_l we associate an $n \times n$ matrix M_l in which $M_l[i,j] := 1$ if $\bar{t}_{i,j} \in C_i$ and $M_l[i,j] := 0$ otherwise. Let l be the least integer such that M_l has either a 0 in each row or a 0 in each column. Note that $l \geq 2$ since M_1 has no zeros. We prove the lemma by showing that either $X \subseteq \mathrm{vars}(C_l)$ or $Y \subseteq \mathrm{vars}(C_l)$ must hold.

Suppose that M_l has a 0 in each row. Since $n \geq 2$, we now show that every row in M_l also has at least one 1. To see this, suppose on the contrary that M_l contains a full 0 row r. Note that by definition of resolution there can be at most one element that changes from 1 in M_{l-1} to 0 in M_l. Since M_{l-1} does not have a 0 in every column, it does not contain a full zero row. Hence it must be the case that the unique element that went from 1 in M_{l-1} to 0 in M_l is in row r. Since $n \geq 2$, we deduce that M_{l-1} has a 0 in each row, contradicting the minimality of l. Hence, for each $i \in [n]$ there exists $j_i \in [n]$ such that $\bar{t}_{i,j_i} \in C_l$. Fixing $i \in [n]$ and putting $j := j_i$, we make the following claims:

(1) for each clause C on P, if $\bar{t}_{i,j} \in C$ then $\{u_i, \bar{u}_i\} \nsubseteq C$;
(2) each x_i-resolvent in π contains $\{u_i, \bar{u}_i\}$ as a subset.
(3) for each $t_{i,j}$-resolvent R in π, if $x_i \notin \mathrm{vars}(R)$ then $\{u_i, \bar{u}_i\} \subseteq R$;

From claim (1) it follows that $\{u_i, \bar{u}_i\} \nsubseteq C_l$. Moreover, as universal literals accumulate along the path, this means that $\{u_i, \bar{u}_i\} \nsubseteq C_m$ for each $m \leq l$. Since the i^{th} row in M_l contains a 0, there exists $j' \in [n]$ such that $\bar{t}_{i,j'} \notin C_l$. As $\bar{t}_{i,j'} \in C_1$, there must be a $t_{i,j'}$-resolvent $C_{l'}$ on P with $l' \leq l$. Then we have $x_i \in \mathrm{vars}(C_{l'})$ by claim (3). Also, for each $m \leq l$, C_m is not an x_i-resolvent by claim (2). It follows that $x_i \in \mathrm{vars}(C_l)$. Since $i \in [n]$ was chosen arbitrarily, we have $X \subseteq \mathrm{vars}(C_l)$.

Suppose on the other hand that M_l does not contain a 0 in each row. Then M_l contains a 0 in each column. A symmetrical argument then shows that $Y \subseteq \mathrm{vars}(C_l)$.

It remains to prove the three claims.

(1) Observe that each clause in π containing the positive literal $t_{i,j}$ also contains the variable u_i (this holds for every axiom and universal literals are never removed). Let C be a clause on the path P for which $\bar{t}_{i,j} \in C$, and, for the sake of contradiction, suppose that $\{u_i, \bar{u}_i\} \subseteq C$. Since $u_i <_{\mathcal{Q}(n)} t_{i,j}$, there cannot be $t_{i,j}$-resolvent on P following C, as such a resolution step is explicitly forbidden in the rules of reductionless LD-Q-Res. This means that $\bar{t}_{i,j}$ occurs in C_k, the final clause of P. This is a contradiction, since C_k is the conclusion of π, which contains no existential literals. Therefore $\{u_i, \bar{u}_i\} \nsubseteq C$.

(2) Observe that each clause in π containing x_i (resp. $\bar{x}_i$) also contains u_i (resp. $\bar{u}_i$) (again, this holds for every axiom and universal literals are never removed). Let R be an x_i-resolvent of R_1 and R_2 in π. Since $x_i \in R_1$ and $\bar{x}_i \in R_2$, we must have $u_i \in R_1$ and $\bar{u}_i \in R_2$. It follows immediately that $\{u_i, \bar{u}_i\} \subseteq R$.

(3) Observe that each axiom in π containing the positive literal $t_{i,j}$ contains variable x_i. Hence, any clause in π that contains literal $t_{i,j}$ but not variable x_i must appear after an x_i-resolvent on some path, and therefore contains $\{u_i, \bar{u}_i\}$ by Claim (2). Now, let R be a $t_{i,j}$-resolvent of R_1 and R_2 in π. Suppose that $x_i \notin \mathrm{vars}(R)$, which implies that $x_i \notin \mathrm{vars}(R_1)$. Since $t_{i,j} \in R_1$, we have $\{u_i, \bar{u}_i\} \subseteq R_1$, and it follows that $\{u_i, \bar{u}_i\} \subseteq R$. ◀

It remains to prove the lower bound formally from the preceding lemmata.

▶ **Theorem 26.** *The squared equality family requires exponential-size reductionless* LD-Q-Res *refutations.*

Proof. Let $n \in \mathbb{N}$, and let π be a reductionless LD-Q-Res refutation of $\mathrm{EQ}^2(n)$. We show that $|\pi| \geq 2^{n-1}$. The size bound is trivially true for $n = 1$, so we assume $n \geq 2$. Put $X := \{x_1, \dots, x_n\}$ and $Y := \{y_1, \dots, y_n\}$, and let $L := \{\bar{t}_{i,j} : i, j \in [n]\}$ be the long clause from $\mathrm{eq}^2(n)$. We call a non-tautological clause S *symmetrical* iff $\mathrm{vars}(S) = X \cup Y$ and $x_i \in S \Leftrightarrow y_i \in S$ for each $i \in [n]$. (A symmetrical clause represents a total assignment to $X \cup Y$). Note that there are 2^n distinct symmetrical clauses.

By Lemma 24, for each symmetrical clause S, there exists a path P_S in π in which all existential literals are contained in $S \cup L$. Moreover, each P_S begins at clause L, since every other clause in $\mathrm{eq}^2(n)$ contains some positive $t_{i,j}$ literal that does not occur in $S \cup L$. By Lemma 25, on each path P from L in π there exists a clause C for which either $X \subseteq \mathrm{vars}(C)$ or $Y \subseteq \mathrm{vars}(C)$. It follows that we can define a function f that maps each symmetrical assignment S to a clause $f(S)$ in π for which either $\mathrm{proj}(S, X) \subseteq f(S)$ or $\mathrm{proj}(S, Y) \subseteq f(S)$. Moreover, since distinct symmetrical clauses S_1 and S_2 satisfy $\mathrm{proj}(S_1, X) \neq \mathrm{proj}(S_2, X)$ and $\mathrm{proj}(S_1, Y) \neq \mathrm{proj}(S_2, Y)$, each $f(S)$ is the image of at most two distinct symmetrical clauses. Hence, π contains at least 2^{n-1} clauses. ◀

Close inspection of the lower-bound proof reveals that particular resolution steps are blocked due to the appearance of merged literals in the antecedents (see claim (1) in the proof of Lemma 25). As we noted in Example 16, such steps remain blocked even if both merged literals implicitly represent the same (non-constant) function, in which case the resolution step is actually perfectly sound. As we will see, the M-Res upper-bound construction makes crucial use of the isomorphism of non-constant merge maps.

5.2 Short M-Res refutations of EQ$^2(n)$

Here we construct short M-Res refutations of the squared equality formulas. The approach is as follows. First, for each $i, j \in [n]$, obtain a line $(\{t_{i,j}\}, M_{i,j})$ by resolving the axioms for the four clauses in $\mathrm{eq}(n)^2$ that contain $\{t_{i,j}\}$. By the natural application of the merge and select operations, one obtains merge maps $M_{i,j}$ in which the merge map for u_i outputs x_i with a single query, the merge map for v_j outputs y_j with a single query, and all other maps are trivial. Notice that all the non-trivial merge maps for a given universal variable are isomorphic, so these n^2 unit clauses can all be resolved against the square clause, utilising the select operation. It is precisely this final step which is unavailable in reductionless LD-Q-Res.

▶ **Theorem 27.** *The squared equality family has $O(n^2)$-size* M-Res *refutations.*

Proof. Let $n \in \mathbb{N}$. We construct a refutation in two stages. In the first stage we explicitly construct an M-Res derivation $\pi := L_1, \dots, L_k$ from $\mathrm{EQ}^2(n)$, where $k = 2n^2$. In the second stage, we show that π can be extended to a refutation with a further $n^2 + 1$ lines.

Stage one. For each $h, i, j \in \mathbb{N}$ we let $\delta(h, i, j) := (h - 1)n^2 + (i - 1)n + j$ and use $L(h, i, j)$ as an alias for $L_{\delta(h,i,j)}$. Similarly, we let $C(h, i, j)$ be the clause, $U(h, i, j)$ be the merge map for u_i, and $V(h, i, j)$ be the merge map for v_j appearing on line $L(h, i, j)$. These $U(h, i, j)$ and $V(h, i, j)$ are the only merge maps in π we define explicitly; we consider all others to be defined implicitly as the appropriate trivial merge map.

Letting $i, j \in [n]$, we define the first $4n^2$ lines with

$$\begin{array}{llllll}
C(0,i,j) := \{x_i, y_j, t_{i,j}\} & U(0,i,j) := \delta(0,i,j) \mapsto \bar{u}_i & V(0,i,j) := \delta(0,i,j) \mapsto \bar{v}_j\,, \\
C(1,i,j) := \{\bar{x}_i, y_j, t_{i,j}\} & U(1,i,j) := \delta(1,i,j) \mapsto u_i & V(1,i,j) := \delta(1,i,j) \mapsto \bar{v}_j\,, \\
C(2,i,j) := \{x_i, \bar{y}_j, t_{i,j}\} & U(2,i,j) := \delta(2,i,j) \mapsto \bar{u}_i & V(2,i,j) := \delta(2,i,j) \mapsto v_j\,, \\
C(3,i,j) := \{\bar{x}_i, \bar{y}_j, t_{i,j}\} & U(3,i,j) := \delta(3,i,j) \mapsto u_i & V(3,i,j) := \delta(3,i,j) \mapsto v_j\,,
\end{array}$$

and observe that each of these lines can be introduced as an axiom.

The next $2n^2$ lines are the result of the natural resolutions over y_j. For each $i, j \in [n]$ we define

$$\begin{array}{lllllll}
C(4,i,j) & := & \{x_i, t_{i,j}\} & U(4,i,j) & := & U(0,i,j)\,, \\
C(5,i,j) & := & \{\bar{x}_i, t_{i,j}\} & U(5,i,j) & := & U(1,i,j)\,,
\end{array}$$

$$\begin{array}{lll}
V(4,i,j) & := & \delta(4,i,j) \mapsto (y_j, \delta(0,i,j), \delta(2,i,j)) \\
& & \delta(2,i,j) \mapsto v_j \\
& & \delta(0,i,j) \mapsto \bar{v}_j\,, \\
V(5,i,j) & := & \delta(5,i,j) \mapsto (y_j, \delta(1,i,j), \delta(3,i,j)) \\
& & \delta(3,i,j) \mapsto v_j \\
& & \delta(1,i,j) \mapsto \bar{v}_j\,.
\end{array}$$

Each line $L(4,i,j)$ can be derived by resolution from $L(0,i,j)$ and $L(2,i,j)$; to see this, note that $U(0,i,j)$ is clearly isomorphic to $U(2,i,j)$ and $V(0,i,j)$ is trivially consistent with $V(2,i,j)$ (their domains are disjoint), therefore $U(4,i,j) = \text{select}(U(0,i,j), U(2,i,j))$ and $V(4,i,j) = \text{merge}(V(0,i,j), V(2,i,j), \delta(4,i,j), y_j)$. A similar argument shows each that $L(5,i,j)$ can be derived by resolution from $L(1,i,j)$ and $L(3,i,j)$.

The final n^2 lines are the result of the natural resolutions over x_i. For each $i, j \in [n]$ we define

$$\begin{array}{llllll}
C(6,i,j) & := & \{t_{i,j}\} & V(6,i,j) & := & V(4,i,j)\,,
\end{array}$$

$$\begin{array}{lll}
U(6,i,j) & := & \delta(6,i,j) \mapsto (x_i, \delta(0,i,j), \delta(1,i,j)) \\
& & \delta(1,i,j) \mapsto u_i \\
& & \delta(0,i,j) \mapsto \bar{u}_i\,.
\end{array}$$

It is easy to see that each $L(6,i,j)$ can be derived by resolution from $L(4,i,j)$ and $L(5,i,j)$, since $V(4,i,j)$ is clearly isomorphic to $V(5,i,j)$ (an isomorphism is $l \mapsto l+n^2$) and $V(0,i,j)$ is trivially consistent with $V(1,i,j)$ (disjoint domains).

Stage two. We now show how π can be extended to a refutation. Let $L_6 := \{L(6,i,j) : i, j \in [n]\}$ denote the final n^2 lines of π, in each of which appears some unit clause $\{t_{i,j}\}$. We observe that, for each $a, b, i \in [n]$, $U(6,i,a)$ is isomorphic to $U(6,i,b)$ (an isomorphism is $l \mapsto l + b - a$); that is, amongst the lines L_6, the non-trivial merge maps for u_i are pairwise isomorphic. Similarly, for each $j \in [n]$, the non-trivial merge maps for v_j appearing in L_6 are pairwise isomorphic.

Now, a line T, consisting of the clause $\{\bar{t}_{i,j} : i, j \in [n]\}$ and a full set of trivial merge maps, can be introduced as an M-Res axiom in a derivation from $\text{EQ}^2(n)$. From T and L_6, in a further n^2 steps we obtain a refutation by successively resolving each line in L_6 against T, removing a literal $\bar{t}_{i,j}$ each time. All such resolution steps are valid, since the merge map for u_i (v_j) in any line can be defined as $\text{select}(M_a, M_b)$, where M_a and M_b are the merge maps for u_i appearing in the antecedent lines. The isomorphism of non-trivial merge maps for u_i (v_j) is preserved, and ensures that $\text{select}(M_a, M_b)$ is defined. ◀

The separation follows immediately from Theorems 26 and 27.

▶ **Theorem 28.** LD-Q-Res *does not p-simulate* M-Res *on QBF.*

6 Extending Merge Resolution to DQBF

In this section, we show that M-Res extends naturally to a DQBF proof system with the addition of a single weakening rule.

An *H-form dependency quantified Boolean formula* (DQBF) is denoted $\Phi := \mathcal{Q} \cdot \phi$. Similarly to QBF, the matrix ϕ is a CNF, but the quantifier prefix $\mathcal{Q}$ has a more general specification that allows variable dependencies to be written explicitly. Formally, $\mathcal{Q} := (X, U, L_{\mathcal{Q}})$, in which $X \subset \mathcal{Z}$ and $U \subset \mathcal{Z}$ are finite sets called the existential and universal variables of Φ, and $L_{\mathcal{Q}} : U \to \wp(X)$ is the *support set function*.

This is not the conventional notation for DQBF (cf. [2]), but it coincides conveniently with our QBF notation. In particular, our definition of 'countermodel' need not change, and we call a DQBF false if it has a countermodel, and true if it does not. We redefine $<_{\mathcal{Q}}$ as a binary relation on $X \times U$ such that $x <_{\mathcal{Q}} u$ holds iff $x \in X$, $u \in U$ and $x \in L_{\mathcal{Q}}(u)$.

To lift M-Res to DQBF, we take Φ to be a DQBF in Definition 14 and add an extra case:

(c) **Weakening.** There exists an integer $a < i$ such that C_i is an existential superclause of C_a and, for each $u \in U$, either (i) $M_i^u = M_a^u$, or (ii) M_a^u is trivial and $M_i^u := i \mapsto l$ for some literal $l \in \{u, \bar{u}\}$.

By 'existential superclause' it is meant that $\text{vars}(C_i) \subseteq X$ and $C_a \subseteq C_i$.

Weakening is, in a clear sense, the simplest rule with which one extends M-Res to DQBF. Its function is merely to represent exactly the paths of the countermodel on which the canonical completeness construction is based. In general, the countermodel needs to be represented in full since merge maps must be isomorphic in order to apply the select operation. Note that the DQBF analogue of Proposition 17 is proved easily with an additional case for the weakening rule.

6.1 Soundness and Completeness

Soundness of M-Res for DQBF is proved in the same way as for QBF, i.e. by showing that the concluding merge maps compute a countermodel. We need only prove that weakening preserves partial strategies.

▶ **Lemma 29.** *Let $(\emptyset, \{M^u : u \in U\})$ be the conclusion of an* M-Res *refutation of a DQBF Φ. Then the functions computed by $\{M^u : u \in U\}$ form a countermodel for Φ.*

Proof. We add an additional case to the inductive step in the proof of Lemma 19. Suppose that L_i was derived by weakening. Then there exists an integer $a < i$ such that $C_a \subseteq C_i$ and, for each $u \in U$, either (i) $M_i^u = M_a^u$, or (ii) M_a^u is trivial and $M_i^u := i \mapsto l$ for some literal $l \in \{u, \bar{u}\}$. Here $A_i \subseteq A_a$, so $\alpha \in A_a$. For each $u \in U$, if u satisfies (i) then $l_i^u(\alpha) = l_a^u(\alpha)$, and if u satisfies (ii) then $l_a^u(\alpha) = * \notin h_i(\alpha)$. Hence we have $h_a(\alpha) \subseteq h_i(\alpha)$. It follows that the restriction of ϕ by $\alpha \cup h_i(\alpha)$ contains the empty clause by the inductive hypothesis. ◀

Completeness, on the other hand, cannot be established with an analogue of Theorem 20; DQBF is strictly larger than QBF, and hence simulation of reductionless LD-Q-Res does not guarantee completeness. Our proof rather extends the method by which completeness of reductionless LD-Q-Res was proved in Lemma 4; namely, the construction of a 'full binary tree' of resolution steps based on the countermodel, following the prefix order of existential variables.

We give an overview of the construction. Let $\Phi := (X, U, L_{\mathcal{Q}}) \cdot \phi$ be a false DQBF with a countermodel h. For each $\alpha \in \langle X \rangle$, the assignment $\alpha \cup h(\alpha)$ falsifies some clause $C_\alpha \in \phi$ by definition of countermodel. Now, consider the M-Res line whose clause is the largest existential clause falsified by α and whose merge maps are constant functions computing $h(\alpha)$. Each such line can be derived in two M-Res steps, by weakening the axiom corresponding to C_α. Moreover, the clauses $\{C_\alpha : \alpha \in \langle X\rangle\}$ form the leaves of a full binary tree resolution refutation which can be completed using an arbitrary order of the existential pivots X. The merge maps are constructed by merging over the pivot x iff $x \in L_{\mathcal{Q}}(u)$; otherwise the select operation takes the merge map from either antecedent, since the full binary tree structure *guarantees* that they are isomorphic.

As merge maps essentially represent the structure of resolution steps in the subderivation, it is no surprise that the merge maps in our construction also have a full binary tree structure. This structure is captured by the following definition.

▶ **Definition 30** (binary tree merge map). A *binary tree merge map* for a variable u over a set of variables $X := \{x_1, \dots, x_n\}$ is a function M with domain $[2_{n+1} - 1]$ and rule

$$M(i) := \begin{cases} (x_{\lfloor \log i\rfloor+1}, 2i, 2i+1) & \text{if } 1 \leq i < 2^n\,, \\ l_i & \text{if } 2^n \leq i < 2^{n-1}\,, \end{cases}$$

where each $l_i \in \{u, \bar{u}\}$.

The construction itself is defined recursively in the proof of completeness, combining full binary tree refutations for $\Phi[x]$ and $\Phi[\bar{x}]$ for some $x \in X$ with a single resolution step. For the corresponding induction, we also need to restrict the countermodel $h = \{h_u : u \in U\}$. Given a literal l with $\text{var}(l) = x$, $h[l] := \{h_u[l] : u \in U\}$ is the set of functions $h_u[l] : \langle L_{\mathcal{Q}}(u) \setminus \{x\}\rangle \to \{u, \bar{u}\}$ defined by $h_u[l](\alpha) := h_u((\alpha \cup \{l\})\restriction^{L_{\mathcal{Q}}(u)})$. Restrictions preserve countermodels in the following sense.

▶ Proposition 31. Let h be a countermodel for a DQBF $\Phi := (X, U, L_{\mathcal{Q}}) \cdot \phi$ and let l be a literal with $\text{var}(l) \in X$. Then $h[l]$ is a countermodel for $\Phi[l]$.

As the final precursor to the completeness proof, we show that a derivation of the negated literal $\bar{l}$ and the restricted countermodel $h[l]$ can be obtained easily from a refutation of the restricted DQBF $\Phi[l]$

▶ Proposition 32. Let $\Phi := (X, U, L_{\mathcal{Q}}) \cdot \phi$ be a false DQBF, let l be a literal with $\text{var}(l) \in X$, and let $(\emptyset, \{M_u : u \in U\})$ be the conclusion of be an M-Res refutation of $\Phi[l]$. Then there exists an M-Res derivation of $(\{\bar{l}\}, \{M_u : u \in U\})$ from Φ.

Proof. Let π be the refutation with the given conclusion. The desired derivation may be obtained from π simply by adding the literal $\{\bar{l}\}$ to each clause, applying weakening where necessary, and adjusting the indexing of the merge maps to account for the extra weakening steps. ◀

▶ **Lemma 33.** *Every false H-form DQBF has an* M-Res *refutation.*

Proof. Let $\Phi := (X, U, L_{\mathcal{Q}}) \cdot \phi$ be a false DQBF, and let $X := \{x_1, \dots, x_n\}$ where the x_i are pairwise distinct. For any M-Res refutation π with conclusion $(C_k, \{M_k^u : u \in U\})$, let $\{h_u : u \in U\}$ be the *concluding countermodel* for π, where the h_u are the functions computed by the *concluding merge maps* M_k^u. A merge map for $u \in U$ over $L_{\mathcal{Q}}(u)$ is said to be *complete* if it is isomorphic to a binary tree merge map for u over $L_{\mathcal{Q}}(u)$. By induction on the number n of existential variables, we show that, for each countermodel h for Φ, there exists an M-Res refutation whose concluding countermodel is h and whose concluding merge maps are complete. To that end, let $h := \{h_u : u \in U\}$ be an arbitrary countermodel for Φ.

For the base case $|X| = 0$, observe that each h_u is a constant function with some singleton codomain $\{l_u\}$. By definition of countermodel, there exists a clause $C \in \phi$ such that $C = \{\bar{l}_u : u \in \text{vars}(C)\}$. Applying the axiom rule to C, one obtains a derivation of the line $(\emptyset, \{M^u : u \in U\})$ in which M^u computes the constant function h_u if $u \in \text{vars}(C)$, and is trivial otherwise. With a single weakening step, each trivial M^u can be swapped for a merge map isomorphic to $1 \mapsto l_u$. Then each M^u is trivially complete and computes the constant function h_u.

For the inductive step, let $n \in \mathbb{N}$. Combining Propositions 31 and 32 with the inductive hypothesis, we deduce that there exist M-Res derivations π and π' of the lines $(\{\bar{x}_n\}, \{M_u : u \in U\})$ and $(\{x_n\}, \{M'_u : u \in U\})$ from Φ in which the M_u and M'_u are complete merge maps computing $h_u[x_n]$ and $h_u[\bar{x}_n]$. Assume that the lines of π are indexed from 1 to $|\pi|$ and that those of π' are indexed from $|\pi|+1$ to $|\pi|+|\pi'|$. For each $u \in U$, the domains of M_u and M'_u are disjoint, so $M_u \bowtie M'_u$. If $x_n \notin L_{\mathcal{Q}}(u)$, then $h_u[x_n] = h_u[\bar{x}_n]$, and we must have $M_u \simeq M'_u$ since complete merge maps computing the same function must be isomorphic. It follows that the line $(\emptyset, \{M''_u : u \in U\})$ can be derived from Φ, where

$$M''_u := \begin{cases} \text{merge}(M_u, M'_u, |\pi| + |\pi'| + 1, x_i) & \text{if } x_i \in L_{\mathcal{Q}}(u), \\ M_u & \text{if } x_i \notin L_{\mathcal{Q}}(u). \end{cases}$$

It is easy to see that the M''_u are complete merge maps computing the h_u. ◀

The weakening rule is clearly polynomial-time checkable. Thus the following is immediate from Lemmata 29 and 33.

▶ **Theorem 34.** M-Res *is a proof system for H-form DQBF.*

7 Conclusions

We argue that building strategies *into* proofs is the natural way to deal with incompleteness for DQBF CDCL-systems [2]. The other approach, known as Fork Resolution [34], uses extension variables, and is not known to correspond to an existing implementation [36].

We also suggest that H-form (rather than S-form) DQBFs may be more suitable for CDCL-style solving, since associated proof systems 'prove the existence of Herbrand functions'. In the QBF realm, this is of course equivalent to proving the non-existence of Skolem functions, but that does not carry over to DQBF (in a precise technical sense [2]). From this standpoint, it is natural to refute H-form DQBFs by finding the Herbrand functions that certify falsity. Moreover, it is unnatural to refute S-form DQBFs — which amounts to proving the non-existence of Skolem functions — by looking for Herbrand functions that may exist *even if the formula is true*. We suggest that this notion is the source of the soundness issues [12] associated with CDCL systems for DQBF.

Explicit representations may also be relevant for QBF solving. In dependency learning [32], variable dependencies are ignored until clause learning is blocked by an illegal merge. Our work demonstrates that many 'illegal' merges are perfectly sound inferences; moreover, Merge Resolution provides a mechanism for identifying such cases based on isomorphism.

Particular implementations may want to fine-tune the details. Isomorphism is an easy way to determine the equivalence of two Boolean functions, but in general it seems unlikely that two equivalent functions will have identical representations. This points towards efficient (approximate) equivalence testing as the key to a successful implementation of M-Res.

References

1 QBFEVAL homepage. `http://www.qbflib.org/index_eval.php`. Accessed: 2018-09-26.

2 Valeriy Balabanov, Hui-Ju Katherine Chiang, and Jie-Hong R. Jiang. Henkin quantifiers and Boolean formulae: A certification perspective of DQBF. *Theoretical Computer Science*, 523:86–100, 2014.

3 Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. *Formal Methods in System Design*, 41(1):45–65, 2012.

4 Valeriy Balabanov, Jie-Hong Roland Jiang, Mikoláš Janota, and Magdalena Widl. Efficient extraction of QBF (counter)models from long-distance resolution proofs. In Blai Bonet and Sven Koenig, editors, *Conference on Artificial Intelligence (AAAI)*, pages 3694–3701. AAAI Press, 2015.

5 Marco Benedetti and Hratch Mangassarian. QBF-based formal verification: Experience and perspectives. *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, 5(1-4):133–191, 2008.

6 Olaf Beyerdorff, Joshua Blinkhorn, Leroy Chew, Renate Schmidt, and Martin Suda. Reinterpreting dependency schemes: Soundness meets incompleteness in DQBF. *Journal of Automated Reasoning (in press)*, 2018.

7 Olaf Beyersdorff and Joshua Blinkhorn. Dependency schemes in QBF calculi: Semantics and soundness. In Michel Rueher, editor, *Principles and Practice of Constraint Programming (CP)*, volume 9892 of *Lecture Notes in Computer Science*, pages 96–112. Springer, 2016.

8 Olaf Beyersdorff, Joshua Blinkhorn, and Luke Hinde. Size, cost and capacity: A semantic technique for hard random QBFs. In Anna R. Karlin, editor, *ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

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

10 Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. On unification of QBF resolution-based calculi. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 8635 of *Lecture Notes in Computer Science*, pages 81–93. Springer, 2014.

11 Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. Proof complexity of resolution-based QBF calculi. In Ernst W. Mayr and Nicolas Ollinger, editors, *International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 76–89. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

12 Olaf Beyersdorff, Leroy Chew, Renate A. Schmidt, and Martin Suda. Lifting QBF resolution calculi to DQBF. In Nadia Creignou and Daniel Le Berre, editors, *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 9710 of *Lecture Notes in Computer Science*, pages 490–499. Springer, 2016.

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

14 Samuel R. Buss. Towards NP-P via proof complexity and search. *Annals of Pure and Applied Logic*, 163(7):906–917, 2012.

15 Michael Cashmore, Maria Fox, and Enrico Giunchiglia. Partially grounded planning as quantified Boolean formula. In Daniel Borrajo, Subbarao Kambhampati, Angelo Oddi, and Simone Fratini, editors, *International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI, 2013.

16 Stephen A. Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, Cambridge, 2010.

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

18 Uwe Egly, Martin Kronegger, Florian Lonsing, and Andreas Pfandler. Conformant planning as a case study of incremental QBF solving. *Annals of Mathematics and Artificial Intelligence*, 80(1):21–45, 2017.

19 Uwe Egly, Florian Lonsing, and Magdalena Widl. Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In Kenneth L. McMillan, Aart Middeldorp, and Andrei Voronkov, editors, *International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR)*, volume 8312 of *Lecture Notes in Computer Science*, pages 291–308. Springer, 2013.

20 Peter Faymonville, Bernd Finkbeiner, Markus N. Rabe, and Leander Tentrup. Encodings of bounded synthesis. In Axel Legay and Tiziana Margaria, editors, *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 10205 of *Lecture Notes in Computer Science*, pages 354–370, 2017.

21 Marijn J. H. Heule and Oliver Kullmann. The science of brute force. *Communications of the ACM*, 60(8):70–79, 2017.

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

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

24 William Klieber, Samir Sapra, Sicun Gao, and Edmund M. Clarke. A non-prenex, non-clausal QBF solver with game-state learning. In Ofer Strichman and Stefan Szeider, editors, *Interenational Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 6175 of *Lecture Notes in Computer Science*, pages 128–142. Springer, 2010.

25 Roman Kontchakov, Luca Pulina, Ulrike Sattler, Thomas Schneider, Petra Selmer, Frank Wolter, and Michael Zakharyaschev. Minimal module extraction from DL-lite ontologies using QBF solvers. In Craig Boutilier, editor, *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 836–841. AAAI Press, 2009.

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

27 Andrew C. Ling, Deshanand P. Singh, and Stephen Dean Brown. FPGA logic synthesis using quantified boolean satisfiability. In Fahiem Bacchus and Toby Walsh, editors, *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 3569 of *Lecture Notes in Computer Science*, pages 444–450. Springer, 2005.

28 Hratch Mangassarian, Andreas G. Veneris, and Marco Benedetti. Robust QBF encodings for sequential circuits with applications to verification, debug, and test. *IEEE Transactions on Computers*, 59(7):981–994, 2010.

29 Joao Marques-Silva and Sharad Malik. Propositional SAT solving. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 247–275. Springer, 2018.

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

31 Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Long distance Q-resolution with dependency schemes. In Nadia Creignou and Daniel Le Berre, editors, *International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 9710 of *Lecture Notes in Computer Science*, pages 500–518. Springer, 2016.

32 Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Dependency learning for QBF. In Serge Gaspers and Toby Walsh, editors, *International Conference on Theory and Practice of Satisfiability Testing (SAT)*, volume 10491 of *Lecture Notes in Computer Science*, pages 298–313. Springer, 2017.

33 Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Polynomial-time validation of QCDCL certificates. In Olaf Beyersdorff and Christoph M. Wintersteiger, editors, *International Conference on Theory and Practice of Satisfiability Testing (SAT)*, volume 10929 of *Lecture Notes in Computer Science*, pages 253–269. Springer, 2018.

34 Markus N. Rabe. A resolution-style proof system for DQBF. In Serge Gaspers and Toby Walsh, editors, *International Conference on Theory and Practice of Satisfiability Testing (SAT)*, volume 10491 of *Lecture Notes in Computer Science*, pages 314–325. Springer, 2017.

35 Horst Samulowitz, Jessica Davies, and Fahiem Bacchus. Preprocessing QBF. In Frédéric Benhamou, editor, *International Conference on Principles and Practice of Constraint Programming (CP)*, volume 4204 of *Lecture Notes in Computer Science*, pages 514–529. Springer, 2006.

36 Christoph Scholl and Ralf Wimmer. Dependency quantified boolean formulas: An overview of solution methods and applications - extended abstract. In Olaf Beyersdorff and Christoph M. Wintersteiger, editors, *International Conference on Theory and Practice of Satisfiability Testing (SAT)*, volume 10929 of *Lecture Notes in Computer Science*, pages 3–16. Springer, 2018.

37 João P. Marques Silva, Inês Lynce, and Sharad Malik. Conflict-driven clause learning SAT solvers. In *Handbook of Satisfiability*, pages 131–153. IOS Press, 2009.

38 Martin Suda and Bernhard Gleiss. Local soundness for QBF calculi. In Olaf Beyersdorff and Christoph M. Wintersteiger, editors, *International Conference on Theory and Practice of Satisfiability Testing (SAT)*, volume 10929 of *Lecture Notes in Computer Science*, pages 217–234. Springer, 2018.

39 Moshe Y. Vardi. Boolean satisfiability: Theory and engineering. *Communications of the ACM*, 57(3):5, 2014.

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

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il