# On Proof Complexity for Quantified Boolean Formulas

*By*

Anil Shukla

MATH 10201104004

The Institute of Mathematical Sciences, Chennai

*A thesis submitted to the*

*Board of Studies in Mathematical Sciences*

*In partial fulfillment of requirements*

*For the Degree of*

DOCTOR OF PHILOSOPHY

*of*

HOMI BHABHA NATIONAL INSTITUTE

February, 2017

# Homi Bhabha National Institute

## Recommendations of the Viva Voce Board

As members of the Viva Voce Committee, we certify that we have read the dissertation prepared by Anil Shukla entitled "On Proof Complexity for Quantified Boolean Formulas" and recommend that it maybe accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

________________________________________________ Date:

Chair - Prof. V. Arvind

________________________________________________ Date:

Guide/Convener - Prof. Meena Mahajan

________________________________________________ Date:

Examiner - Prof. Rahul Santhanam

________________________________________________ Date:

Member 1 - Prof. R. Ramanujam

________________________________________________ Date:

Member 2 - Prof. Venkatesh Raman

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to HBNI.

I hereby certify that I have read this thesis prepared under my direction and recommend that it may be accepted as fulfilling the thesis requirement.

**Date:**

**Place:** Guide

# STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at Homi Bhabha National Institute (HBNI) and is deposited in the Library to be made available to borrowers under rules of the HBNI.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the Competent Authority of HBNI when in his or her judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

Anil Shukla

# DECLARATION

I, hereby declare that the investigation presented in the thesis has been carried out by me. The work is original and has not been submitted earlier as a whole or in part for a degree / diploma at this or any other Institution / University.

Anil Shukla

# List of Publications arising from the thesis

## Journal

1. Meena Mahajan and **Anil Shukla**. "Level-ordered $Q$-resolution and tree-like $Q$-resolution are incomparable", Information Processing Letters, 116(3):256–258, 2016.

## Conferences

1. Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and **Anil Shukla**. "Feasible interpolation for QBF Resolution calculi", Proceedings of 42nd International Colloquium on Automata, Languages, and Programming (ICALP), Part I, LNCS vol. 9134, Springer, 180–192, 2015.

2. Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and **Anil Shukla**. "Are short proofs narrow? QBF Resolution is not simple", Proceedings of 33rd Symposium on Theoretical Aspects of Computer Science (STACS), LIPIcs vol/ 47, 15:1–15:14, 2016.

3. Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and **Anil Shukla**. "Understanding Cutting Planes for QBFs". Proceedings of 36th Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS) LIPIcs vol. 65, 40:1–40:15, 2016.

Anil Shukla

# Contents

# SYNOPSIS

Propositional proof complexity–a sub-branch of computational complexity–is an extensively studied area, with a number of lower bound techniques for various propositional proof systems (for example Resolution, and Cutting Planes). The purpose of this thesis is to assess whether similar techniques are applicable for proof systems for quantified Boolean formulas (QBFs). The major contributions of this work are as follows:

1. We show that level-ordered $Q$-resolution and tree-like $Q$-resolution, two restrictions of $Q$-resolution system, are incomparable.

2. We establish the feasible interpolation technique, first introduced by Krajíček for Resolution [62], to all CDCL-based QBF Resolution calculi. This provides the first general lower bound method for CDCL-based QBF calculi and also extends the scope of the feasible interpolation technique.

3. We introduce a cutting planes system CP+$\forall$red for QBFs and analyse the proof-theoretical strength of this new calculus. We also establish the strategy extraction technique and feasible interpolation technique for the new calculi.

4. We show that both the size-width relation, established by Ben-Sasson and Wigderson in [9], and space-width relation, established by Atserias and Dalmau in [3], for the Resolution proof system drastically fail in $Q$-resolution, even in its weaker tree-like version.

# List of Figures

# Chapter 1

# Introduction

## 1.1 Proof Complexity

Proof complexity is a sub-branch of computational complexity, in which the main focus is to prove non-trivial lower bounds for complete and sound proof systems (Definition 2.1). To be precise, the problem is to find some hard theorems (resp. false statements), proving (resp. refuting) which in a particular proof system requires exponentially many steps with respect to the size of the theorem. Even more important is to establish techniques for proving lower bounds. Apart from having fun, proving lower bounds are closely related to the main open problem of complexity theory: NP vs coNP, in case of propositional proofs, and NP vs PSPACE in case of proof complexity for quantified Boolean formulas (QBFs). Cook and Reckhow in [36], proved that NP$\neq$ coNP iff for every propositional proof system, there is a polynomial-size family of tautologies that requires superpolynomial size proofs with respect to the size of the tautology. Since finding such family of tautologies are quite hard, the theory of proof complexity breaks this problem into smaller problems of proving such lower bounds for specific proof systems. Several propositional proof systems have been introduced in the literature, for example Resolution (Section

2.3.1), Cutting Planes (Section 2.3.2), and Frege proof systems (Section 2.3.3).

Another importance of proving lower bounds comes from the field of *automated theorem provers*. It is known that SAT solvers based on *conflict-driven clause learning* (CDCL) implicitly generate resolution proofs for unsatisfiable instances [74]. Therefore lower bounds for resolution proofs directly translate to the lower bounds for the running time of CDCL-based SAT solvers.

Since modern SAT solvers are so important, as it solves several hard industrial instances very efficiently, and Resolution is closely related to them, great efforts have been given for proving lower bounds for Resolution. As a result several lower bound techniques have been developed for Resolution. For example, *feasible interpolation* technique, first introduced by Krajíček in [62], is a very successful technique, which transfers circuit lower bounds to proof size lower bounds. It also applies to Cutting Planes proof system [67]. Another important lower bound technique for Resolution is the *size-width* relation, introduced by Ben-Sasson and Wigderson in [9]. Here size of a proof denotes the number of clauses in it, and width of a proof is the length of the biggest clause in it. The size-width relation allows us to prove size lower bounds via width lower bounds.

Yet another lower bound technique in propositional proof complexity is the *game theoretic methods* and the combinatorial characterizations of the hardness measures. For example, Pudlák in [68] characterizes the size of resolution proofs as games. Recently Pudlák's game have been used in [23] for improving size and width lower bounds. Atserias and Dalmau in [3], gave a combinatorial characterization of resolution width, and used it to show that even space is lower bounded by width in Resolution. Informally, the space complexity for refuting a formula in Resolution is the minimum number of clauses that must be kept in memory to refute the formula. Game theoretic methods are also useful for obtaining optimal bounds in tree-like resolution [18]. However we will not consider game theoretic methods in this thesis.

Interested readers are referred to Bonacina's PhD dissertation [23].

In Chapter 2, we revisit some important propositional proof systems, along with their lower bound techniques, which are relevant for the thesis.

The picture is more complex for proof systems for quantified Boolean formulas (QBFs), as there exist two different approaches for QBF solving based on Resolution: CDCL-based and expansion-based solving. A number of QBF proof systems have been designed to capture these approaches. The core CDCL-based QBF Resolution system is $Q$-resolution (Q-Res), introduced in [60]. This has been augmented to capture ideas from CDCL solving, leading to long-distance resolution (LD-Q-Res) [4], universal resolution (QU-Res) [78], or its combinations like $\mathsf{LQU}^{+}$-Res [6]. The core expansion-based proof system is $\forall$Exp+Res, introduced in [56]. Recently more powerful expansion-based proof systems have been developed in the form of IR-calc, and IRM-calc [12]. In Chapter 2, we present the simulation order (Definition 2.3) among these proof systems from [13]. Also, QBF proof systems based on Frege, introduced recently by Beyersdorff et al. in [11], will be presented.

Since QBF proof complexity is a relatively young field, very few lower bound techniques are known for it. Recently a lower bound for tree-like $Q$-resolution was obtained via a game theoretic characterization of proof size [17]. However, the most important lower bound technique developed for QBF systems is the *strategy extraction* technique (Section 2.5). We say that a QBF proof system $P$ has strategy extraction if given a refutation $\pi$ of a false QBF $\mathcal{F}$, it is possible to extract **efficiently** from $\pi$ the winning strategy of the universal player for $\mathcal{F}$. Beyersdorff et al. in [13], were the first to use strategy extraction as a lower bound technique for QBF proof systems QU-Res (and therefore Q-Res). Based on the fact that strategy extraction for QU-Res is possible in $\mathsf{AC}^0$ ( [4]), they constructed a hard formula $\mathrm{QP{\scriptsize ARITY}}_n$, such that the only winning strategy for the formula is the parity function. Since the parity function is known to be hard for $\mathsf{AC}^0$ circuits [50], $\mathrm{QP{\scriptsize ARITY}}_n$

must require exponential size proofs in QU-Res. Recently, Beyersdorff et al. in [11], used strategy extraction technique for proving lower bounds in QBF proof systems based on restricted Frege. We come back to the strategy extraction technique in Section 2.5. We dedicate Chapter 2 of the thesis for literature survey.

## 1.2 Our Contributions

The purpose of this thesis is to understand which lower bound techniques of propositional proof systems are effective for QBF proof systems. The main contributions of the thesis in the field of QBF proof complexity are as follows:

### 1.2.1 Incomparability Results

In Chapter 3, we show that level-ordered $Q$-resolution and tree-like $Q$-resolution, two restrictions of $Q$-resolution are incomparable (Theorem 3.1). That is neither can simulate (Definition 2.3) the other.

For showing that tree-like $Q$-resolution cannot simulate level-ordered $Q$-resolution, we use the family of false formulas, which we denote as $\phi_n$, defined by Janota and Marques-Silva in [56]. In [56], they showed that $\phi_n$ is hard for $\forall$Exp+Res, and since $\forall$Exp+Res p-simulates tree-like $Q$-resolution, $\phi_n$ is hard for tree-like $Q$-resolution as well. On the other side, $\phi_n$ was shown to be easy for $Q$-resolution [56], and we observe that the same proof is indeed level-oredered and hence $\phi_n$ is easy for level-ordered $Q$-resolution.

For proving that level-ordered $Q$-resolution cannot simulate tree-like $Q$-resolution, we use the family of false QBFs $CR_n$, defined again by Janota and Marques-Silva in [56]. They showed that $CR_n$ is hard for level-ordered $Q$-resolution, but here we show that $CR_n$ is in fact easy for tree-like $Q$-resolution. We prove this by giving a short tree-like $Q$-resolution refutation of $CR_n$ (Section 3.3).

This work was done jointly with Meena Mahajan. It has been published in the Journal; 'Information Processing Letters', 2016 [65].

### 1.2.2 Introduction of a New QBF Proof System based on Cutting Planes

In propositional case, Cutting Planes proof system, which works with linear inequalities, has been developed in [37]. It is well known that Cutting Planes proof system is in between Resolution and Frege, that is, it is exponentially stronger than Resolution, however is exponentially weaker than Frege (see Section 2.3.2). For QBFs a similar Cutting Planes system based on integer linear programming has been missing. In Chapter 4, we define a natural Cutting Planes system for QBFs and give a comprehensive analysis of its proof complexity. To be precise, we prove the following results in Chapter 4:

**1. Cutting Planes for QBFs.** We introduce a complete and sound QBF proof system CP+∀red for false QBFs, that works with quantified set of linear inequalities, where each variable is either quantified existentially or universally in a quantifier prefix. The lines in the CP+∀red proof systems are linear inequalities. The system CP+∀red extends the classical Cutting Planes system with one single ∀-reduction rule allowing manipulation of universally quantified variables. The definition of the system thus naturally aligns with the QBF Resolution systems Q-Res [60] and QU-Res [78] and the stronger QBF Frege systems [11] that likewise add universal reduction to their classical base systems.

Inspired by the recent work on semantic Cutting Planes [47] we also define a stronger system semCP+∀red where in addition to universal reduction all semantically valid inferences between inequalities are allowed (Section 4.4).

**2. Relations to Other QBF Proof Systems.** We compare our new system CP+∀red with previous QBF Resolution and Frege systems. In contrast to the classical setting, the emerging picture is somewhat more complex: while CP+∀red is strong enough to simulate the core CDCL QBF Resolution systems Q-Res and QU-Res and indeed is exponentially stronger than these systems (Theorem 4.8), CP+∀red is incomparable (under a natural circuit complexity assumption) to even the base system ∀Exp+Res of the expansion Resolution systems (Theorem 4.14).

On the other hand, CP+∀red turns out to be simulated by QBF Frege (Theorem 4.15) and QBF Frege is exponentially more powerful than CP+∀red (Corollary 4.16). While this separation could be achieved by lifting the classical separation [67] to QBF by considering purely existentially quantified formulas, we highlight that our separation also holds for natural QBFs expressing the clique-co-clique principle, which is not known to admit a succinct propositional representation.

**3. Lower Bound Techniques for CP+∀red.** Technically, our separations rely on two lower bound methods that we establish for CP+∀red: strategy extraction (Section 4.3) and feasible interpolation (Chapter 5, Section 5.2).

*Strategy extraction* as a lower bound technique was first devised for Q-Res, and QU-Res [13], and subsequently extended to QBF Frege systems [11, 20]. The technique applies to calculi that allow to efficiently extract winning strategies for the universal player from a refutation (or alternatively Skolem functions for the existential variables from a proof of a true QBF). Here we show that CP+∀red admits strategy extraction in $\mathsf{TC}^0$, thus establishing an appealing link between CP+∀red proofs (which can count) and the counting circuit class $\mathsf{TC}^0$ (Theorem 4.18). For each function $f \in \mathsf{P/poly}$ we construct false QBFs $Q\text{-}f_n$ where each winning strategy forces the universal player to compute $f$. Thus assuming the existence of $f \in \mathsf{P/poly} \setminus \mathsf{TC}^0$ we obtain lower bounds for $Q\text{-}f_n$ in CP+∀red (Theorem 4.14) and even semCP+∀red (Corollary 4.20), whereas however the same formulas are easy in ∀Exp+Res.

We establish feasible interpolation technique for CP+∀red in Chapter 5 and thereby obtain an unconditional lower bound result for the system CP+∀red.

This work was done jointly with Olaf Beyersdorff, Leroy Chew, and Meena Mahajan. It has been published in the proceeding of $36^{th}$ IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), 2016 [16].

### 1.2.3 Establishing Feasible Interpolation for QBF Proof Systems

For propositional proof systems, a number of lower bound techniques have been developed. One of the most successful techniques is the *feasible interpolation* technique first developed by Krajíček for Resolution [62]. The technique also applies to Cutting Planes [67]. The technique transfers circuit lower bounds to proof size lower bounds. In chapter 5, we establish feasible interpolation technique for all CDCL-based QBF Resolution proof systems. We do this by establishing the technique for the most powerful CDCL-based QBF proof system $\mathsf{LQU}^+$-Res (Section 5.1.2). As discussed above, we also establish this technique for CP+∀red (Section 5.2) and also for the stronger semCP+∀red (Section 5.3). This largely extends the scope of feasible interpolation technique.

As a consequence, we prove an unconditional exponential lower bound for the clique-co-clique formulas $\Phi_{n,k}$ for the systems $\mathsf{LQU}^+$-Res, CP+∀red and semCP+∀red proof systems (Sections 5.1.4, 5.2, and 5.3 respectively). The formulas $\Phi_{n,k}$ encode the obviously false statement that a given graph on $n$ vertices both has and does not have a $k$-clique (for definition of $\Phi_{n,k}$, see Section 4.2.3).

This work was done jointly with Olaf Beyersdorff, Leroy Chew, and Meena Mahajan. It has been published in the proceeding of $42^{nd}$ International Colloquium on

Automata, Languages, and Programming (ICALP), 2015 [14]. Results from Section 5.2, and 5.3 appear in [16].

### 1.2.4 Negative Results: Size-width and Space-width Relation of Resolution fails in $Q$-resolution

In their paper [9] ‘Short proofs are narrow – resolution made simple’, Ben-Sasson and Wigderson introduces the *size-width* relation, which is one of the important lower bound techniques for Resolution. It allows us to prove size lower bounds via width lower bounds. Also Atserias and Dalmau in [3] show that lower bounds for space in Resolution again can be obtained via lower bounds for width.

In chapter 6, we assess whether similar techniques are effective for Resolution calculi for quantified Boolean formulas (QBFs). We concentrate only on the following three QBF Resolution systems: Q-Res, ∀Exp+Res, and IR-calc. This choice is motivated by the fact that Q-Res and ∀Exp+Res form the base systems for CDCL and expansion-based solving, respectively, and IR-calc unifies both approaches in a natural way, as it simulates both Q-Res and ∀Exp+Res [12].

Though space and width have not been considered in QBF before, these notions straightforwardly apply to QBF Resolution systems. However, due to the ∀-reduction rule in Q-Res handling universal variables, it is relatively easy to enforce that universal literals accumulate in clauses of Q-Res proofs, thus always leading to large width, irrespective of size and space requirements (Lemma 6.4). This prompts us to consider *existential width* — counting only existential literals — as an appropriate width measure in QBF. This definition aligns both with Q-Res, resolving only on existential variables, as well as with ∀Exp+Res and IR-calc, which like all expansion-based systems only operate on existential literals. We show the following:

**1. Negative Results.** Our main results show that the size-width relation of [9] as well as the space-width relation of [3] dramatically *fail* for Q-Res, even when considering the tighter existential width. We first notice that the proof establishing the size-width result in [9] almost fully goes through, except for some very inconspicuous step that fails in QBF (Proposition 6.5). But it is not only the particular technique that fails: we prove that Tseitin transformations (see Section 2.1) of formulas expressing a natural completion principle from [56] have small size and space, but require large existential width in tree-like Q-Res (Theorem 6.6), thus refuting the size-width relation for tree-like Q-Res as well as the space-width relation for general dag-like Q-Res.

As the formulas for the completion principle have $O(n^2)$ variables, they do not rule out size-width relations in general Q-Res. However, we show that different formulas, hard for tree-like Q-Res [56], provide counterexamples for size-width relations in full Q-Res (Theorem 6.8).

Technically, our main contributions are width lower bounds for the above formulas, which we show by careful counting arguments. We complement these results by existential width lower bounds for parity-formulas $\mathrm{QP}\textsc{arity}_n$ from [13], providing an optimal width separation between Q-Res and ∀Exp+Res (Theorem 6.18).

**2. Positive Results and Width-space-preserving Simulations.** Though the negative picture above prevails, we prove some positive results for size-width-space relations for tree-like versions of the expansion-based Resolution systems ∀Exp+Res and IR-calc. Proofs in ∀Exp+Res can be decomposed into two clearly separated parts: an expansion phase followed by a classical resolution phase. This makes it easy to transfer almost the full spectrum of the classical relations to ∀Exp+Res (Theorem 6.19).

To lift these results to IR-calc (Theorem 6.20), we show a series of careful space and width-preserving simulations between tree-like Q-Res, ∀Exp+Res, and IR-calc. In particular, we show the surprising result that tree-like ∀Exp+Res and tree-like IR-calc are equivalent with respect to simulation (Lemma 6.15), thus providing a rare example of two proof systems that coincide in the tree-like, but are separated in the dag-like model [13]. The only other such example that we are aware of is regular-resolution vs. Resolution (although this is perhaps slightly less natural as regular-resolution is just a sub-system of Resolution). In addition, our simulations provide a simpler proof for the simulation of tree-like Q-Res by ∀Exp+Res (Corollary 6.17), shown in [56] via a more involved argument.

Our last positive result is a size-space relation in tree-like Q-Res (Theorem 6.20), which we show by a pebbling game analogous to the classical relation in [46]. Not surprisingly, this only positive result for Q-Res avoids any reference to the notion of width.

This work was done jointly with Olaf Beyersdorff, Leroy Chew, and Meena Mahajan. It has been published in the proceeding of $33^{rd}$ International Symposium on Theoretical Aspects of Computer Science (STACS), 2016 [15].

# Chapter 2

# Literature Survey

## 2.1 Quantified Boolean Formulas

A literal is a Boolean variable or its negation. For any variable $x$, we say the literal $x$ is complementary to the literal $\neg x$ ($\bar{x}$) and vice versa. A *clause* is a disjunction ($\vee$) of literals and a *term* is a conjunction ($\wedge$) of literals. We say a clause $C$ is a tautological clause if there exists a variable $x$ such that both the literals $x$ and $\neg x$ belongs to $C$. Otherwise the clause is non-tautological. We denote the empty clause by $\square$. A formula in *Conjunctive Normal Form (CNF)* is a conjunction of clauses. A *DNF* is a disjunction of terms. For convenience the clause $C$ is written simply as a set of literals and any CNF formula as a set of clauses. If a clause has at most $k$-literals, we call it a $k$-clause. A $k$-CNF formula is a set of $k$-clauses. Let SAT be the language of all satisfiable propositional Boolean formulas, and UNSAT be the set of all unsatisfiable propositional CNF formulas. For a literal $l = x$ or $l = \neg x$, we write $\mathrm{var}(l)$ for $x$ and extend this notation to $\mathrm{var}(C)$ for a clause $C$.

Let $\alpha$ be any partial assignment. For a clause $C$, we write $C|_\alpha$ for the clause obtained after applying the partial assignment $\alpha$ to $C$. For example, applying $\alpha: x_1 \leftarrow 0$ on the clause $C \equiv (x_1 \vee x_2 \vee x_3)$ yields $C|_\alpha \equiv (x_2 \vee x_3)$, and on applying $\alpha: x_1 \leftarrow 1$ on the same clause gives $C|_\alpha \equiv 1$. Let $F$ be a CNF formula, and $x$ is a variable in $F$. Then $F|_{x=1}$ is a CNF formula obtained from $F$ by removing all clauses containing positive $x$, and removing all occurrences of negative $x$. Let $A_1, \ldots, A_k$ and $B$ be some propositional formulas. Then we say that $A_1, \ldots, A_k \models B$ is valid, if any assignment $\alpha$ which satisfies $A_1 \wedge \cdots \wedge A_k$ also satisfies $B$.

Quantified Boolean Formulas (QBFs) extend propositional logic with Boolean quantifiers with the standard semantics that $\forall x.F$ is satisfied by the same truth assignments as $F|_{x=0} \wedge F|_{x=1}$ and $\exists x.F$ as $F|_{x=0} \vee F|_{x=1}$. We assume that QBFs are in *closed prenex form* with a CNF matrix, i.e, we consider the form $\mathcal{Q}_1 x_1 \cdots \mathcal{Q}_n x_n \,.\, \phi$ where each $\mathcal{Q}_i$ is either $\exists$ or $\forall$, and $\phi$ is a quantifier-free CNF formula in the variables $x_1, \ldots, x_n$. Any QBF can be efficiently converted to an equivalent QBF in this form, but note that restricting formulas to prenex form is not a restriction from a logical point of view. However, there is no unique prenex form, for a non-prenex formula, and the chosen prenex form may strongly influence the length of proofs. Such formulas are succinctly denoted as $\mathcal{Q} \,.\, \phi$, where $\phi$ is called the *matrix*, and $\mathcal{Q}$ is its *quantifier prefix.* The *index* $\text{ind}(y)$ of a variable $y$ is its position in the prefix $\mathcal{Q}$; for each $i \in [n]$, $\text{ind}(x_i) = i$. If $\text{ind}(x) < \text{ind}(y)$, we say that $x$ occurs *before* $y$, or *to the left of* $y$. Following scoping rules, the rightmost variable in $\mathcal{Q}$ is also called the *innermost* variable. The *quantification level* $\text{lv}(y)$ of a variable $y$ in $\mathcal{Q} \,.\, \phi$ is the number of alternations of quantifiers $y$ has on its left in the quantifier prefix of $\mathcal{Q} \,.\, \phi$. For instance, in a QBF $\exists x_1 \forall x_2 \forall x_3 \exists x_4 \phi$, $\text{lv}(x_1) = 1$, $\text{lv}(x_2) = \text{lv}(x_3) = 2$, and $\text{lv}(x_4) = 3$. Let $\mathcal{F} = \mathcal{Q}_1 x_1 \cdots \mathcal{Q}_n x_n \,.\, \phi$ be a QBF, then $\mathcal{Q}_1 x_1 \cdots \mathcal{Q}_n x_n \,.\, \phi|_{x_i=1}$ is a QBF with the CNF matrix $\phi|_{x_i=1}$ obtained from $\phi$ by removing all clauses containing positive $x_i$, and removing all occurrences of negative $x_i$. We denote this formula by $\mathcal{F}|_{x_i=1}$.

A QBF $\mathcal{Q}_1 x_1 \cdots \mathcal{Q}_k x_k \,.\, \phi$ can be seen as a game between two players: *universal* ($\forall$) and *existential* ($\exists$). In the $i^{th}$ step of the game, the player $\mathcal{Q}_i$ assigns a value to the variable $x_i$. The existential player wins if $\phi$ evaluates to 1 under the assignment constructed in the game. The universal player wins if $\phi$ evaluates to 0. A *strategy for* $x_i$ is a function from all variables of index $< i$ to $\{0,1\}$. A *strategy* for the universal player is a collection of strategies, one for each universally quantified variable. Similarly, a *strategy* for the existential player is a collection of strategies, one for each existentially quantified variable. A strategy for the universal player is a winning strategy if using this strategy to assign values to variables, the universal player wins any possible game, irrespective of the strategy used by the existential player. Winning strategies for the existential player are similarly defined. For any QBF, exactly one of the two players has a winning strategy. A QBF is false if and only if there exists a *winning strategy* for the universal player ( [52], [2, Section 4.2.2], [66, Chapter 19]).

**Tseitin (Tseytin) Transformations [76]** Given a Boolean formula $F(\vec{x})$, the Tseitin transformation converts it into 3-CNF formula $F'(\vec{x},\vec{y})$ such that $F(\vec{x})$ is satisfiable if and only if $F'(\vec{x},\vec{y})$ is satisfiable. The size of $F'(\vec{x},\vec{y})$ is polynomially related to the size of $F(\vec{x})$. Not just a formula, in fact, given any Boolean circuit, the Tseitin transformation converts it into a 3-CNF formula such that the size of the formula is linear in the size of the circuit. Moreover, the assignments which make the circuit evaluates to 1 are in 1-to-1 correspondence with the assignments that satisfy the 3-CNF formula. Briefly this is achieved by introducing a new variable for each gate, representing the value of the gate, and adding clauses to enforce that the gate values are correctly computed (for details see [38, Chapter 34]).

We also use the Tseitin transformations for the formulas with quantifier prefix. The newly introduced Tseitin variables are existential. We extend the original quantifier prefix in the following ways: if a Tseitin variable $t$ abbreviates a formula $f(x_1,\ldots,x_n)$ then $\exists t$ will occur somewhere right of $\mathcal{Q}_i x_i$ ( for all $1 \leq i \leq n$) in the quantifier prefix. It should be noted that the concrete placement of $\exists t$ may have a severe impact on the size of proofs.

## 2.2 Proof Systems

The concept of proof system was first defined by Cook and Reckhow in [36]. Consider the language **SAT** of all satisfiable Boolean formulas. Trivially SAT is in NP as there always exists a short proof (satisfying truth assignment) for formulas in SAT. However for formulas not in SAT how short proofs could be is not clear. This requires the definition of a proof system.

**Definition 2.1.** *[36] A proof system for a non-empty language $L \subseteq \{0,1\}^*$ is a polynomial time computable function $f : \{0,1\}^* \to \{0,1\}^*$ such that Range$(f) = L$. For string $x \in L$, we say a string $w \in \{0,1\}^*$ is an $f$-proof of $x$ if $f(w) = x$. We say a proof system for L is polynomially bounded if there exists a polynomial $p(x) \in \mathbb{N}[x]$ such that each $x \in L$ has an $f$-proof '$w$' of size $|w| \leq p(|x|)$.*

**Definition 2.2** (Completeness and soundness of a proof system [36])**.** *We say that a proof system $f : \{0,1\}^* \to \{0,1\}^*$ for a language $L$ is complete if $f(\{0,1\}^*) \supseteq L$ (i.e, for every $x \in L$, there exists an $f$-proof $w$), and is sound if $f(\{0,1\}^*) \subseteq L$ (i.e, if there exists an $f$-proof $w$ for $x$, then $x \in L$).*

**Definition 2.3** (Simulations [36])**.** *Given two proof systems $f_1$ and $f_2$ for the same language $L$, we say that $f_1$ simulates $f_2$, if there exists a function $g$ and a polynomial $p$ such that $f_1(g(w)) = f_2(w)$ and $|g(w)| \leq p(|w|)$ for all $w$. Thus $g$ translates a proof $w$ of $x \in L$ in the system $f_2$ into a proof $g(w)$ of $x \in L$ in the system $f_1$, with at most polynomial blow-up in proof-size. If there is such a $g$ that is also polynomial-time computable, then we say that $f_1$ p-simulates $f_2$.*

## 2.3 Propositional Proof Systems

A proof system for the language UNSAT is called **propositional proof system** (pps). Note that pps can also be defined for the languages of true propositional DNF formulas (TAUT), however here we consider pps for unsatisfiable formulas only.

From Definition 2.1, it is clear that NP is precisely the set of languages that have polynomially bounded proof systems. In fact, Cook and Reckhow proved in [36] that if one can find a polynomial-size family of tautologies that does not have polynomial size proofs then this will separate NP from coNP and thus separate P from NP. Since finding such family of tautologies is quite hard, the theory of proof complexity breaks this problem into smaller problems of proving such lower bounds for specific proof systems. We briefly discuss three important pps: Resolution, Cutting Planes, and Frege proof systems.

### 2.3.1 Resolution Proof System

Resolution (Res) is well studied propositional proof system introduced by Blake in [22] and proposed by Robinson in [73] as automated theorem proving. The lines in the resolution proofs are clauses. Given a CNF formula $F$, Resolution can infer new clauses according to the following inference (resolution) rule :

$$\frac{C \vee x \quad D \vee \neg x}{C \vee D},$$

where $C$ and $D$ are clauses and $x$ is a variable being resolved, called as pivot variable. Let $F$ be an unsatisfiable CNF formula. A resolution proof (refutation) $\pi$ of $F$ is a sequence of clauses $D_1, \ldots, D_l$ with $D_l = \Box$ and each clause in the sequence is either from $F$ or is derived from previous clauses in the sequence using the above resolution rule.

We can also view $\pi$ as a directed acyclic graph $G_\pi$, where the source nodes are the clauses from $F$, internal nodes are the derived clauses and the empty node is the unique sink. Edges in $G_\pi$ are from the hypotheses to the conclusion for each resolution step. In $G_\pi$, we say that a clause $C$ is descendant to a clause $D$ if there is a directed path from $C$ to $D$. If $G_\pi$ is a tree, we call $\pi$ a tree-like resolution proof ($\mathsf{Res}_\mathsf{T}$) of $F$. In other words, in tree-like resolution proofs one cannot reuse the derived clauses. We call $\pi$ a regular resolution proof if on any directed path in $G_\pi$ no variable appears twice in any resolution rule as the pivot variable.

### Complexity Measures for Resolution

For a CNF formula $F$, $|F|$ is the number of clauses in it, and $w(F)$ denotes the maximum number of literals in any clause of $F$. Let $\pi \vdash_{\overline{\mathsf{Res}}} F$ (resp. $\pi \vdash_{\overline{\mathsf{Res}_\mathsf{T}}} F$) denote that $\pi$ is a resolution proof (tree-like resolution proof, respectively) of the formula $F$.

The most important complexity measure for Resolution is the size. The size $|\pi|$ of a refutation $\pi$ is defined as the number of clauses in $\pi$. The size complexity $S(\vdash_{\overline{\mathsf{Res}}} F)$ (resp. $S(\vdash_{\overline{\mathsf{Res}_\mathsf{T}}} F)$) of refuting an unsatisfiable CNF formula $F$ in Resolution (resp. in tree-like resolution) is defined as $\min\{|\pi| \ : \ \pi \vdash_{\overline{\mathsf{Res}}} F\}$ (repectively $\min\{|\pi| \ : \ \pi \vdash_{\overline{\mathsf{Res}_\mathsf{T}}} F\}$).

The *width* of a clause $C$ is the number of literals in $C$, denoted by $w(C)$. The width $w(F)$ of a CNF formula $F$, is the maximum width of a clause in $F$. The width $w(\pi)$ of a proof $\pi$ is the maximum width of any clause appearing in $\pi$. The width $w(\vdash_{\overline{\mathsf{Res}}} F)$ (resp. $w(\vdash_{\overline{\mathsf{Res}_\mathsf{T}}} F)$) of refuting an unsatisfiable CNF formula $F$ in Resolution (resp. tree-like resolution) is defined as $\min\{w(\pi) \ : \ \pi \vdash_{\overline{\mathsf{Res}}} F\}$ (resp. $\min\{w(\pi) \ : \ \pi \vdash_{\overline{\mathsf{Res}_\mathsf{T}}} F\}$).

The third complexity measure for Resolution is *space*, first defined in [46]. In literature it is also called clause space, to distinguish it from variable space or total space, see for example, [8]. We consider only clause space in this thesis, and so we call it just space. Informally, it is the minimal number of clauses that must be kept simultaneously in memory to refute a formula. Instead of viewing a proof as a DAG, we view it as a sequence of CNF formulas $F_0, F_1, \dots, F_s$, where $F_0 = \emptyset$, $\Box \in F_s$, and each $F_{i+1}$ is obtained from $F_i$ by either erasing some clause, downloading an axiom, or adding a clause derived by resolution rule from clauses in $F_i$. In the latter case, one of the premises of the inference rule may also simultaneously be deleted. For such a proof $\sigma$, $\mathit{CSpace}(\sigma)$ is the maximum number of clauses in any $F_i$, $i \in [s]$. The space to refute $F$, denoted $\mathit{CSpace}(\vdash_{\mathsf{Res}} F)$, is the minimum $\mathit{CSpace}(\sigma)$ over all resolution refutations $\sigma$ for $F$.

If we modify the inference step so that the clause(s) used to obtain the inference are erased in the same step, then any derived clause can be used at most once and we obtain a tree-like space-oriented resolution proof. Correspondingly we define $\mathit{CSpace}(\vdash_{\mathsf{Res}_\mathsf{T}} F)$ as the minimum space used by any tree-like proof sequence refuting $F$.

We come back to these complexity measures in Chapter 6.

The main objective is to prove size lower bounds for Resolution. To be precise, come-up with a hard family of CNF formulas $F_n$ such that $S(\vdash_{\mathsf{Res}} F_n)$ is exponential in the size of $F_n$. Apart from theoretical interests such lower bounds are useful for practical purposes as well: Resolution is at the core of most of the SAT solvers since the introduction of the DPLL algorithm [41, 42] and its improvements to *Conflict Driven Clause Learning* (CDCL) algorithms, therefore size lower bounds on Resolution translate to time lower bounds for these algorithms. To be precise, a run of a SAT solver on some unsatisfiable formula, provides a proof of unsatisfiability of the input formula, and these proofs of unsatisfiability are closely related to resolution proofs. Interested readers are referred to Ashish Sabharwal's Ph.D dissertation [74].

### Lower Bound Techniques for Resolution

The first exponential size lower bound for Resolution has been proved by Haken in 1985 [53]. He considered the pigeonhole principle for his proof. The pigeonhole principle says that if we put $m$ pigeons into $n$ holes, where $m > n$, then at least one hole must contain more than one pigeon. (We discuss pigeonhole principle with precise Theorem statement proved in [53] in Section 2.3.2). Then Urquhart in 1987 [77], showed that refuting Tseitin contradictions, which captures the fact that for every graph, the sum of degrees of all vertices is even, requires exponential steps in Resolution. For the precise definition of Tseitin contradictions, see [9, Definition 4.1]. Then Chvátal and Szemerédi in their outstanding paper [35], showed that for $k \geq 3$, with high probability a random $k$-CNF formula is unsatisfiable and requires an exponential size refutation in Resolution. These lower bounds were achieved by ingenious counting and random restriction techniques. However, Ben-Sasson and Wigderson in [9] showed that all these exponential lower bounds can be achieved by showing the width lower bounds for resolution proofs. They actually introduced a new lower bound technique, commonly known as 'size-width' technique for Resolution.

### The Size-Width Relationship

In their pioneering paper [9], Ben-Sasson and Wigderson showed that resolution size lower bounds can be elegantly obtained by showing lower bounds to the *width* of resolution proofs. In particular, they prove the following Theorem:

**Theorem 2.4** ( [9]). *For all unsatisfiable CNFs $F$ in $n$ variables the following holds:*

$$\begin{aligned} S(\left|\overline{\scriptstyle \mathsf{Res}_T}\right. F) &\geq 2^{w\left(\left|\overline{\scriptstyle \mathsf{Res}}\right. F\right)-w(F)}, \quad \textit{and} \\ S(\left|\overline{\scriptstyle \mathsf{Res}}\right. F) &= \exp\left(\Omega\left(\frac{\left(w\left(\left|\overline{\scriptstyle \mathsf{Res}}\right. F\right)-w(F)\right)^2}{n}\right)\right). \end{aligned}$$

Using Theorem 2.4, Ben-Sasson and Wigderson gave simple and unified proofs for almost all known exponential lower bounds on size of resolution proofs. In addition they also gave some new lower bound results. Inspired from the size-width relationship, one natural question arises: do similar relations exist among other complexity measures in Resolution? The literature contains some positive answers. In their fundamental work, Atserias and Dalmau in [3] showed that lower bounds for space again can be obtained via lower bounds for width. In Chapter 6, we discuss these relations and also assess whether similar techniques are effective for Resolution calculi for quantified Boolean formulas (QBFs).

### Feasible Interpolation for Resolution

Using Craig's interpolation theorem [39,40], Krajíček in [62], has established a new lower bound technique for Resolution commonly known as *feasible interpolation* technique. The technique reduces the problem of proving size lower bounds on Resolution to proving lower bounds on the circuit size for explicit Boolean functions.

To be precise, let $F \equiv A(\vec{p}, \vec{q}) \wedge B(\vec{p}, \vec{r})$ be an unsatisfiable CNF formula, where $\vec{p}, \vec{q}$, and $\vec{r}$ are disjoint set of variables. $A(\vec{p}, \vec{q})$ is the set of clauses over variables $\vec{p}$ and $\vec{q}$ and $B(\vec{p}, \vec{r})$ is the set of clauses over variables $\vec{p}$ and $\vec{r}$. Thus $\vec{p}$ are the common variables among them. Let $\vec{a}$ be the assignment for $\vec{p}$ variables. Then Krajíček showed that from any resolution proof $\pi$ of $F$, one can extract a Boolean interpolating circuit $C(\vec{p})$ of size polynomially related to $|\pi|$, such that $C(\vec{a}) = 0 \implies A(\vec{a}, \vec{q})$ is false and $C(\vec{a}) = 1 \implies B(\vec{a}, \vec{r})$ is false. He further showed that, if $\vec{p}$ variables appears only positively in $A(\vec{p}, \vec{q})$ or only negatively in $B(\vec{p}, \vec{r})$, then one can in fact extract a monotone interpolating circuit from $\pi$ with similar properties. Thus the technique translates the lower bound problem for Resolution to lower bound problem on monotone circuits. Since clique functions are hard for monotone circuits [1], it turns out that a nice encoding of clique functions as CNF formulas (clique-colour formulas) is hard for Resolution proof system as well. We discuss *feasible interpolation* in Chapter 5.

## 2.3.2 Cutting Planes Proof System

After Resolution, Cutting Planes is one of the best known complete and sound proof systems for unsatisfiable CNF formulas. Cutting Planes was first proposed as a proof system in [37], and is designed to show that a given set of linear inequalities has no 0, 1-solutions. Each proof line in Cutting Planes is of the form

$$\sum_k c_k x_k \geq C,$$

where $c_k, C$ are integers. The variables $x_i$'s can take only integer values. However we restrict the variables to take only $0, 1$-values. We do this by adding additional inequalities (called Boolean axioms) $x_k \geq 0$ and $-x_k \geq -1$ for each variable $x_k$. The inference rules are as follows:

**Add:** from $\sum_k c_k x_k \geq C$ and $\sum_k d_k x_k \geq D$ derive $\sum_k (c_k + d_k) x_k \geq C + D$.

**Multiply:** from $\sum_k c_k x_k \geq C$ derive $\sum_k d c_k x_k \geq dC$, where $d \in \mathbb{Z}^+$.

**Divide:** from $\sum_k c_k x_k \geq C$ derive $\sum_k \frac{c_k}{d} x_k \geq \left\lceil \frac{C}{d} \right\rceil$, where $d \in \mathbb{Z}^+$ divides each $c_k$.

**Definition 2.5** (Cutting Planes proofs). *Let $\mathcal{I}$ be a set of inequalities. A Cutting Planes proof of an inequality $I$ from $\mathcal{I}$ is a sequence of inequalities $I_1, \ldots, I_l$ such that $I_l = I$ and for every $j \in \{1, \ldots, l\}$,*

- *$I_j \in \mathcal{I}$, or,*
- *$I_j$ is a Boolean axiom, or,*
- *$I_j$ is derived from earlier inequalities in the sequence via one of the inference rules: add, multiply, or divide.*

*A Cutting Planes refutation $\pi$ of an inconsistent set of inequalities $\mathcal{I}$ is a proof deriving $0 \geq C$ for some positive integer $C$. The length of $\pi$ (denoted $|\pi|$) is the number of lines in it, and the size of $\pi$ (denoted size$(\pi)$) is the bit-size of a representation of the proof (this depends on the number of lines and the binary length of the numbers in the proof).*

**Definition 2.6** (Encoding CNFs as inequalities). *We encode a CNF formula $\phi$ over variables $x_1, \ldots, x_n$ as a set of linear inequalities as follows: define $R(x) = x$, $R(\bar{x}) = 1 - x$. A clause $C \equiv (l_1 \vee \cdots \vee l_k)$ is translated into the linear inequality $R(C) \equiv \sum_{i=1}^{k} R(l_i) \geq 1$. A CNF formula $\phi = C_1 \wedge \cdots \wedge C_m$ is represented as the set* *of inequalities* $F_\phi = \{R(C_1), R(C_2), \ldots, R(C_m)\} \cup B$*, where* $B$ *is the set of Boolean axioms* $x \geq 0, -x \geq -1$ *for each variable* $x$*. We call this the standard encoding.*

We say that a $0,1$-assignment $\alpha$ satisfies the inequality $I \equiv \sum_{i=1}^{n} a_i x_i \geq b$ (i.e, $I|_\alpha = 1$), if $\sum_{i=1}^{n} a_i \alpha_i \geq b$, where $\alpha_i$ is the value given to the variable $x_i$ by $\alpha$. Observe that for any clause $C$, an assignment satisfies $C$ if and only if it satisfies $R(C)$. Hence CNF formula $\phi$ is satisifable iff $F_\phi$ is satisfiable.

After defining standard encoding one can talk about refuting unsatisfiable CNF formulas in Cutting Planes. It is not hard to see that Cutting Planes is at least as powerful as Resolution.

**Theorem 2.7** ( [37], also see [7])**.** *Cutting Planes p-simulates Resolution.*

*Proof.* We only need to simulate resolution rule. Let $\frac{(A \vee C \vee x) \quad (B \vee C \vee \bar{x})}{(A \vee B \vee C)}$ be a resolution step, where the literals in $A$ and $B$ are disjoint. By induction we have the inequalities $R(A \vee C \vee x)$ and $R(B \vee C \vee \bar{x})$. Add the two inequalities together with inequality $y \geq 0$ for each positive literal $y$ in $A$ or $B$ and $1 - y \geq 0$ for each negative literal $\bar{y}$ in $A$ or $B$. This results in an inequality in which the conversion of each literal in $(A \vee B \vee C)$ appears with the coefficient 2 and applying the division rule with $d = 2$ gives the inequality $R(A \vee B \vee C)$. □

As promised before, we now encode pigeonhole principle, with $m$ pigeons and $n$ holes, as a CNF formula $\mathsf{PHP}_n^m$, state Haken's result and revisit the short Cutting Planes refutation of $\mathsf{PHP}_n^m$ $(m > n)$, from [37].

Pigeonhole principle can be easily encoded as an unsatisfiable CNF formula $\mathsf{PHP}_n^m$ over variables $x_{i,j}, 1 \leq i \leq m, 1 \leq j \leq n$, which are supposed to be assigned TRUE if pigeon $i$ is put into hole $j$. $\mathsf{PHP}_n^m$ contains the following clauses:

- $(x_{i,1} \vee \ldots, \vee x_{i,n})$, for $i \in [m]$. This clause ensures that the $i^{th}$ pigeon is assigned to at least one hole.

- $\bar{x}_{i,j} \vee \bar{x}_{k,j}$, for $i, k \in [m], i \neq k, j \in [n]$. This clause ensures that the $j^{th}$ hole does not get both the $i^{th}$ and the $k^{th}$ pigeons.

Haken proves the following Theorem:

**Theorem 2.8.** *[53] For any $n \geq 2$, every resolution refutation of $\mathsf{PHP}_{n-1}^{n}$ has size at least $2^{n/20}$. That is $S(\vdash_{\mathsf{Res}} \mathsf{PHP}_{n-1}^{n}) \geq 2^{n/20}$.*

**Short Cutting Planes Proof for Pigeonhole Principle [37, Proposition 7]**

We have the following inequalities corresponding to the clauses in $\mathsf{PHP}_n^m$:

- $x_{i,1} + \cdots + x_{i,n} \geq 1$, for $i \in [m]$.
- $x_{i,j} + x_{k,j} \leq 1 (\equiv -x_{i,j} - x_{k,j} \geq -1)$, for $i, k \in [m], i \neq k, j \in [n]$.
- $x_{i,j} \geq 0$, $x_{i,j} \leq 1 (\equiv -x_{i,j} \geq -1)$, for $i \in [m], j \in [n]$.

By induction on $k$ from 2 to $m$, we derive $x_{1,j} + x_{2,j} + \cdots + x_{k,j} \leq 1$, for each $j \in [n]$. Note that the required inequalities for $k = 2$ is already present in the set of initial inequalities. This takes care of the base case.

Suppose one has derived $x_{1,j} + x_{2,j} + \cdots + x_{(k-1),j} \leq 1$. We do the following:

1. Add $(k-2)$ copies of $x_{1,j} + x_{2,j} + \cdots + x_{(k-1),j} \leq 1$ and one each of $x_{i,j} + x_{k,j} \leq 1$, with $1 \leq i \leq (k-1)$, to get $(k-1)x_{1,j} + (k-1)x_{2,j} + \cdots + (k-1)x_{k,j} \leq 2k-3$.
2. Apply division rule to get $x_{1,j} + x_{2,j} + \cdots + x_{k,j} \leq \lfloor \frac{2k-3}{k-1} \rfloor = 1$.

Summing these inequalities for all $j$ gives that the sum of all $x_{i,j}$'s is at most $n$. Moreover, summing up the first set of inequalities gives us that the sum of all $x_{i,j}$'s is at least $m$. Thus when $m > n$, we have a contradiction.

This along with Theorem 2.8, shows that Cutting Planes is exponentially more powerful than Resolution.

### Lower Bound Technique for Cutting Planes

*Feasible interpolation* is the only known lower bound technique for Cutting Planes. Pudlák in [67] generalises Krajíček's feasible interpolation technique for Resolution [62] to Cutting Planes, and proves the first exponential lower bound results for Cutting Planes. In Chapter 5, we discuss this technique in detail.

Now we define Frege proof systems, which are known to be exponentially more stronger than Cutting Planes [51, 67].

## 2.3.3 Frege Proof Systems

A Frege proof system is not just a single proof system, but it usually refers to a class of proof systems. A typical *Frege* proof system $\mathcal{G}$ has a finite set of axiom schemes and inference rules. For example $(P \wedge Q) \to P$ might be an axiom. Here $P$ and $Q$ are not just single formulas, but they are meta-symbols that can stand for any propositional formulas. The lines in a Frege proof are propositional formulas built over propositional variables $x_i$ and some finite set of functionally complete connectives. A Frege proof is a sequence of formulas where each formula is an (instance of an) axiom, or is inferred from previous formulas by a valid inference rule. We call such systems Frege systems, after Frege [49]. Apart from being sound and complete, Frege systems are also required to be implicationally complete. That is, for formulas $A_1, \ldots, A_k$, and $B$, if $A_1, \ldots, A_k \models B$ then there exists a Frege proof of $B$ from $A_1, \ldots, A_k$ (i.e, $A_1, \ldots, A_k \vdash_{\mathsf{Frege}} B$). The exact choice of the axiom schemes and rules does not matter as any two Frege systems can p-simulate each other [36]. Therefore in this thesis, we assume that $\mathcal{G}$ has only one inference rule: **modus ponens**,

$$\frac{A \quad A \to B}{B}$$

With this rule the system is sound and implicationally complete [61].

Usually Frege systems are defined as proof systems where the last formula is the proven formula. However, in this thesis we use the equivalent setting of refutational Frege systems where we start with the negation of the formula that we want to prove (that is, include the negation of the formula as an axiom) and derive the contradiction $\bot$.

**Note:** We know that Resolution proof system is sound and complete, but it is not implicationally complete: since we have $A \models A \vee B$ but we cannot derive $A \vee B$ from $A$ in Resolution. However, adding the weakening rule $\frac{A}{A \vee B}$ makes it implicationally complete while retaining soundness.

### Circuit Clases

We recall the definitions of standard circuit classes used in this thesis (cf. [79]). For every $n \in \mathbb{N}$ a Boolean circuit $C_n$ with $n$ inputs is a directed acyclic graph. It contains $n$ input nodes of in-degree 0 (no incoming edges) and a unique output node of outdegree 0 (no outgoing edges). All other nodes are called gates and are labeled with one of $\neg, \vee,$ or $\wedge$. Fan-in of a gate is the number of incoming edges, and fan-out is the number of outgoing edges. The size of a circuit is the number of nodes in it, and its depth is the maximal length of a path from an input node to the output node. The circuit is called a Boolean formula if each node has at most one outgoing edge. It is easy to see that a Boolean circuit implements a function $f_n : \{0,1\}^n \to \{0,1\}$. We say that a language $L \subseteq \{0,1\}^*$ is decidable by a circuit familiy $\{C_n\}_{n\in\mathbb{N}}$, if for all $x \in \{0,1\}^n$, $x \in L \iff C_n(x) = 1$.

$\mathsf{AC}^0$ is the class of languages decidable by circuit families $\{C_n\}_{n\in\mathbb{N}}$ of constant depth, polynomial size with respect to the input size $n$, and whose gates have unbounded fan-in. And when we restrict the depth by a constant $d$ we call the circuit class as $\mathrm{AC}^0_d$. $\mathsf{AC}^0[p]$ circuits are $\mathsf{AC}^0$ circuits augmented with $\mathrm{MOD}_p$ gates, which determines whether the sum of the inputs is 0 modulo $p$. And $\mathsf{TC}^0$ circuits are $\mathsf{AC}^0$ circuits augmented with threshold gates, which determines whether the sum of the inputs is at least some threshold $k$. $\mathsf{NC}^1$ circuits are of polynomial size with respect to the input size $n$, logarithmic depth ($O(\log n)$) and with bounded fan-in. Finally $\mathsf{P/poly}$ is the class of languages that are decidable by circuit families of polynomial size with respect to the input size. We use non-uniform circuit classes in this thesis. Interested readers are referred the book [2].

### $\mathcal{C}$-Frege Proof systems

In literature, several restricted versions of Frege proof systems have been studied. We know that Frege proofs consists of sequence of formulas. If we restrict that every formula in the proof must come from some circuit class $\mathcal{C}$, then such Frege proofs are called as $\mathcal{C}$-Frege proofs, and the proof systems, where only $\mathcal{C}$-Frege proofs are allowed are called as $\mathcal{C}$-Frege proof systems.

### Extended Frege (EF) Proof Systems

Let $\mathcal{G}$ be any Frege proof system. An extended Frege proof (EF), is a sequence of formulas $A_1, \ldots, A_n$ such that for all $i$, either $A_i$ is derived from earlier formulas using some inference rule of $\mathcal{G}$, or $A_i$ is an axiom instance of $\mathcal{G}$, or else $A_i$ is an extension formula of the form $r_i \equiv \varphi$, where $\varphi$ is any formula and $r_i$ is a fresh extension variable, (i.e, $r_i$ occurs neither in $\varphi$ nor in any of $A_1, \ldots, A_{i-1}$ nor in the last formula in the proof). The last formula $A_n$ are not allowed to contain any of the extension variables. Thus EF proof sytems are Frege systems with an extension rule.

Like Frege proof systems, it is known that any two EF systems p-simulate each other [36]. There are two important open problems regarding Frege systems:

1. Can Frege proof systems p-simulates EF systems?

2. Are EF proof systems polynomially bounded?

In 1979 [36], Cook and Reckhow gave polynomially sized EF proof of $\mathsf{PHP}^n_{n-1}$ and claimed that refuting $\mathsf{PHP}^n_{n-1}$ in Frege needs exponential size. However, Buss in

1987 [30] first showed that pigeonhole principle is easy for Frege as well and gave a polynomial size Frege proof of $\mathsf{PHP}^n_{n-1}$. However both proofs were very different, Cook and Reckhow use the inductive methods whereas Buss uses the counting method. Since both the techniques were so different it was believed that Frege can not simulate the inductive proof of $\mathsf{PHP}^n_{n-1}$ in polynomial steps. Moreover this was also taken as evidence that Frege cannot simulate EF. However recently in 2015, Buss gave a quasipolynomial sized inductive proof of $\mathsf{PHP}^n_{n-1}$ [29]. He showed how to mimic the inductive proof of EF (refuting $\mathsf{PHP}^n_{n-1}$) in Frege systems in quasipolynomial time, a major step towards problem 1 (above).

### Lower Bound Techniques for Frege Proof Systems

Unfortunately there exists no known lower bounds for Frege proof systems. In fact one of the biggest open problem in the field of proof complexity is: are Frege proof systems polynomially bounded?

**Comments:** All hard formulas that we have seen so far for Resolution and Cutting Planes, are known to be easy for Frege. Regarding simulations, we know that the resolution rule is a special case of modus ponens:

$$\frac{p \to q \quad p}{q} \equiv \frac{\bar{p} \vee q \quad p}{q},$$

and it has been shown in [51] that Frege p-simulates Cutting Planes. It has also been shown that the feasible interpolation technique which is known to be effective for Resolution and Cutting Planes, fails for Frege proof systems under plausible cryptographic and number-theoretic assumptions [24, 27, 63].

## 2.4 Proof Systems for QBFs

Proof systems for the language of false quantified Boolean formulas (QBFs) are called *QBF proof systems.* Equivalently, QBF proof systems can be defined for true QBFs as well. Since (using PSPACE-completeness of QBF) any QBF $\mathcal{Q} \,.\, \phi$ can be converted in polynomial time to another QBF $\mathcal{Q}' \,.\, \phi'$ such that exactly one of $\mathcal{Q} \,.\, \phi$ and $\mathcal{Q}' \,.\, \phi'$ is true, it suffices to consider only QBF proof systems for false QBFs.

Proof complexity for QBFs is a relatively young field, studying proof systems for quantified Boolean logic. However during the last decade there has been great interest for this. Again there are two reasons: first is its tight relation to the separation of complexity classes NP vs PSPACE and the second reason to study the lower bounds for proofs is the analysis of QBF solvers. We first revisit QBF proof systems based on Resolution.

### 2.4.1 QBF Resolution Calculi

QBF proof systems which are based on Resolution are called QBF Resolution calculi. There exists two main solving approaches utilizing CDCL solving and expansion-based solving. We first describe known QBF Resolution calculi from the literature. We start by describing the proof systems modelling *CDCL-based QBF solving*:

**CDCL-based QBF Resolution Calculi**

In this section we give a brief overview of CDCL-based Resolution calculi. We first define Q-Res which is the base system for CDCL solving, and a simple generalisation to QU-Res. We then define several extensions of Q-Res: long-distance Q-resolution (LD-Q-Res) and LQU$^{+}$-Res.

*Q-resolution (Q-Res)*, introduced by Kleine Büning et al. in [60], is a resolution-like calculus that operates on QBFs in prenex form where the matrix is a CNF. The lines in a Q-Res proof are clauses. It uses the resolution rule (Res) with the side condition that the pivot variable is existential and provided that the resolvent clause is not a tautology (i.e, contains a positive and negative literal at the same time). In addition Q-Res has a universal reduction rule which allows dropping a universal variable literal from a clause provided the clause has no existential variable to the right of the reduced variable. Note that we also forbid tautological clauses in the input. This is to preserve the soundness of the system: for example, consider a true formula $\forall x.\ (x \vee \neg x)$. The $\forall$-Red rule on the formula derives the empty clause, which is unsound. The inference rules of Q-Res are given in Figure 2.1. Similar to tree-like resolution we have tree-like Q-Res (i.e, Q-Res$_{\mathsf{T}}$). To be precise, if the underlying proof graph of Q-Res proof is a tree (that is, no derived clause is used more than once), then we have a Q-Res$_{\mathsf{T}}$ proof.

$$\frac{}{C}\ (\text{Axiom}) \qquad\qquad \frac{C_1 \cup \{x\} \qquad C_2 \cup \{\neg x\}}{C_1 \cup C_2}\ (\text{Res})$$

$C$ is a clause in the matrix. Variable $x$ is existential. If $z \in C_1$, then $\neg z \notin C_2$.

$$\frac{C \cup \{u\}}{C}\ (\forall\text{-Red})$$

Variable $u$ is universal. If $x \in C$ is existential, then $\text{lv}(x) < \text{lv}(u)$.

Figure 2.1: The rules of Q-Res [60]

*QU*-resolution (QU-Res) [78] removes the restriction from Q-Res that the resolved variable must be existential variable and allows resolution on universal variables as well. Thus QU-Res is classical Resolution augmented with a $\forall$-Red rule.

*Long-distance resolution (LD-Q-Res)* appears originally in the work of Zhang and Malik in [80] and was formalized into a calculus by Balabanov and Jiang in [4]. Observe that in Resolution proof system deriving a tautological clause containing a literal and its complement is redundant. One can easily ignore such resolution steps from the proof [70]. But in Q-Res such steps are prohibited, as it makes the system unsound (for an easy example see [56, Remark 1]). In other words, resolution step in Q-Res is allowed between the clauses with distance 1 (i.e, other than resolved existential variable, the clauses does not contain any other complementary literals). To be precise, two non-tautological clauses $C$ and $D$ are at distance 1 if there exists a variable $x$ (either existentially or universally quantified) such that $x \in C$ and $\neg x \in D$ or vice versa. In general two non-tautological clauses are at distance $k$ if there are $k$ variables $\{x_1, \ldots, x_k\}$ appearing in both the clauses but as a complementary literals.

Modern search-based QBF solvers perform resolution in essence. In such solvers a tautological clause containing both positive and negative literals of a (universal) variable may result from a resolution step [80]. Since resolution is performed between two clauses with distance greater than 1, it is referred to as long-distance resolution. Balabanov and Jiang introduced a new sound and complete QBF proof system (LD-Q-Res) corresponding to such solvers [4]. To make long-distance resolution step a sound rule, LD-Q-Res merges the complementary literals of a universal variable $u$ into a special literal $u^*$. In particular, different literals of a universal variable $u$ may be merged only if $lv(x) < lv(u)$, where $x$ is the resolved (pivot) variable. The rules are given in Figure 2.2.

$$\frac{}{C}\ \text{(Axiom)} \qquad \frac{D \cup \{u\}}{D}\ (\forall\text{-Red}) \qquad \frac{D \cup \{u^*\}}{D}\ (\forall\text{-Red}^*)$$

$C$ is a clause in the matrix. Literal $u$ is universal and $\mathrm{lv}(u) \geq \mathrm{lv}(l)$ for all $l \in D$.

$$\frac{C_1 \cup U_1 \cup \{x\} \qquad C_2 \cup U_2 \cup \{\neg x\}}{C_1 \cup C_2 \cup U}\ \text{(Res)}$$

Variable $x$ is existential. If for $l_1 \in C_1, l_2 \in C_2$, $\mathrm{var}(l_1) = \mathrm{var}(l_2) = z$ then $l_1 = l_2 \neq z^*$. $U_1, U_2$ contain only universal literals with $\mathrm{var}(U_1) = \mathrm{var}(U_2)$. For each $u \in \mathrm{var}(U_1)$ we require $\mathrm{lv}(x) < \mathrm{lv}(u)$. If for $w_1 \in U_1, w_2 \in U_2$, $\mathrm{var}(w_1) = \mathrm{var}(w_2) = u$ then $w_1 = \neg w_2$, $w_1 = u^*$ or $w_2 = u^*$. $U$ is defined as $\{u^* \mid u \in \mathrm{var}(U_1)\}$.

Figure 2.2: The rules of LD-Q-Res [4]

*LQU$^+$-Res* [6] extends LD-Q-Res by allowing short and long distance resolution pivots to be universal; however, the pivot is never a merged literal $z^*$, and the level restriction now must become an index restriction, to differentiate between universal variables on the same level. The rules are given in Figure 2.3.

$$\frac{}{C}\ (\text{Axiom}) \qquad \frac{D \cup \{u\}}{D}\ (\forall\text{-Red}) \qquad \frac{D \cup \{u^*\}}{D}\ (\forall\text{-Red}^*)$$

$C$ is a clause in the matrix. Literal $u$ is universal and $\text{lv}(u) \geq \text{lv}(l)$ for all $l \in D$.

$$\frac{C_1 \cup U_1 \cup \{x\} \qquad C_2 \cup U_2 \cup \{\neg x\}}{C_1 \cup C_2 \cup U}\ (\text{Res})$$

If for $l_1 \in C_1, l_2 \in C_2$, $\text{var}(l_1) = \text{var}(l_2) = z$ then $l_1 = l_2 \neq z^*$. $U_1, U_2$ contain only universal literals with $\text{var}(U_1) = \text{var}(U_2)$. For each $u \in \text{var}(U_1)$ we require $\text{ind}(x) < \text{ind}(u)$, as variable $x$ could be at the same level as $u$. If for $w_1 \in U_1, w_2 \in U_2$, $\text{var}(w_1) = \text{var}(w_2) = u$ then $w_1 = \neg w_2$, $w_1 = u^*$ or $w_2 = u^*$. $U$ is defined as $\{u^* \mid u \in \text{var}(U_1)\}$.

Figure 2.3: The rules of LQU$^+$-Res [6]

### Expansion-based QBF Resolution Calculi

Another main approach to QBF-solving is through expansion of quantifiers [10, 21, 55]. One approach will be to expand both existential and universal variables. However, Janota and Marques-Silva in [56] observed that this creates two main obstacles to developing a proof system using both expansion and resolution (apart from exponential growth). The first obstacle is that the result of the expansion is not in prenex form, but this can be overcome by prenexing the expansion. The second obstacle is that expanding existential quantifier does not yield a CNF. So they focussed only on the expansion of universal quantifiers, and came up with a sound and complete proof system $\forall$Exp+Res.

Expansion of universal quantifiers certainly decreases the number of quantifications but at the cost of increasing the size. Also for maintaining prenex normal form, one has to include fresh variables. For instance, consider the following *QBF* formula: $\exists x \forall y \exists z.\phi$. We can expand the universal variable $y$ and get $\exists x.(\exists z.\phi[y/0]) \wedge (\exists z.\phi[y/1])$. Observe that $z$ may depend on the universal variable $y$. Therefore we introduce fresh variables of $z$ for both the subformulas and get an equivalent formula $\exists x \exists z^{0/y} \exists z^{1/y}.\phi[y/0, z/z^{0/y}] \wedge \phi[y/1, z/z^{1/y}]$.

Inspired from the above discussions several calculi based on *instantiation* of universal variables were introduced: ∀Exp+Res [56], IR-calc, and IRM-calc [12]. All these calculi operate on clauses that comprise only existential variables from the original QBF, which are additionally *annotated* by a substitution to some universal variables, e.g. $\neg x^{0/u_1, 1/u_2}$. For any annotated literal $l^{\sigma}$, the substitution $\sigma$ must not make assignments to variables at a higher quantification level than $l$, i.e. if $u \in \mathsf{dom}(\sigma)$, then $u$ is universal and $\mathrm{lv}(u) < \mathrm{lv}(l)$. To preserve this invariant, we use the *auxiliary notation* $l^{[\sigma]}$, which for an existential literal $l$ and an assignment $\sigma$ to the universal variables filters out all assignments that are not permitted, i.e. $l^{[\sigma]} = l^{\{c/u \in \sigma \mid \mathrm{lv}(u) < \mathrm{lv}(l)\}}$. We say that an assignment is complete if its domain is the set of all universal variables. Likewise, we say that a literal $x^{\tau}$ is fully annotated if all universal variables $u$ with $\mathrm{lv}(u) < \mathrm{lv}(x)$ in the QBF are in $\mathsf{dom}(\tau)$, and a clause is fully annotated if all its literals are fully annotated.

The simplest expansion-based calculi is the calculus ∀Exp+Res. The calculus ∀Exp+Res works with fully annotated clauses on which resolution is performed. The rules are given in figure 2.4. Similar to other tree-like proofs, we have tree-like ∀Exp+Res proofs (denoted ∀Exp+Res$_{\mathsf{T}}$).

$$\frac{}{\{l^{[\tau]} \mid l \in C, l \text{ is existential}\} \cup \{\tau(l) \mid l \in C, l \text{ is universal}\}} \text{ (Axiom)}$$

$C$ is a clause from the matrix. $\tau$ is an assignment to all universal variables.

$$\frac{C_1 \cup \{x^{\tau}\} \qquad C_2 \cup \{\neg x^{\tau}\}}{C_1 \cup C_2} \text{ (Res)}$$

Figure 2.4: The rules of ∀Exp+Res [56]

We illustrate the axiom download step in ∀Exp+Res with an example: consider the following QBF formula with just one clause (for simplicity)

$$\exists e_1 \forall u_1 \exists e_2 \forall u_2 \exists e_3 \forall u_3.\ (e_1 \vee \bar{e}_2 \vee u_1 \vee e_3 \vee \bar{u}_3).$$

Let $\tau = u_1 \leftarrow 0, u_2 \leftarrow 1, u_3 \leftarrow 1$ and $\sigma = u_1 \leftarrow 1, u_2 \leftarrow 1, u_3 \leftarrow 1$ be two assignments to all universal variables. Then in $\forall$Exp+Res the clauses $(e_1 \vee \bar{e}_2^{0/u_1} \vee 0 \vee e_3^{0/u_1,1/u_2} \vee 0)$, and $(e_1 \vee \bar{e}_2^{1/u_1} \vee 1 \vee e_3^{1/u_1,1/u_2} \vee 0) \equiv 1$ can be downloaded with respect to $\tau$ and $\sigma$ respectively.

The calculus IR-calc [12] extends $\forall$Exp+Res by enabling partial assignments in annotations. The system IR-calc is more flexible in the sense that it uses 'delayed' expansion and can mix instantiation with resolution steps. Formally, IR-calc works with partial assignments on which we use auxiliary operations of *completion* and *instantiation*. For assignments $\tau$ and $\mu$, we write $\tau \veebar \mu$ for the assignment $\sigma$ defined as $\sigma(x) = \tau(x)$ if $x \in \mathsf{dom}(\tau)$, otherwise $\sigma(x) = \mu(x)$ if $x \in \mathsf{dom}(\mu)$. The operation $\tau \veebar \mu$ is called *completion* as $\mu$ provides values for variables not defined in $\tau$. For an assignment $\tau$ and an annotated clause $C$, the function $\mathsf{inst}(\tau, C)$ returns the annotated clause $\left\{l^{[\sigma \veebar \tau]} \mid l^{\sigma} \in C\right\}$. The rules of IR-calc are given in Figure 2.5. Similarly we have tree-like IR-calc proofs (denoted $\mathsf{IR_T}$-calc), where derived clauses cannot be reused. Unlike $\forall$Exp+Res, in IR-calc the assignment $\tau$ set values only to those universal variables which belongs to the clause (being downloaded) and in such a way

$$\frac{}{\{x^{[\tau]} \mid x \in C, x \text{ is existential}\}} \text{ (Axiom)}$$

$C$ is a non-tautological clause from the matrix. $\tau = \{0/u \mid u \text{ is universal in } C\}$, where the notation $0/u$ for literals $u$ is shorthand for $0/x$ if $u = x$ and $1/x$ if $u = \neg x$.

$$\frac{\{x^{\tau}\} \cup C_1 \qquad \{\neg x^{\tau}\} \cup C_2}{C_1 \cup C_2} \text{ (Resolution)} \qquad \frac{C}{\mathsf{inst}(\tau, C)} \text{ (Instantiation)}$$

$\tau$ is an assignment to universal variables.

Figure 2.5: The rules of IR-calc [12]

that the clause restricted to $\tau$ contains no universal variable. For example consider the same QBF formula with the single clause. For $\tau = u_1 \leftarrow 0, u_3 \leftarrow 1$, IR-calc downloads the following clause: $(e_1 \vee \bar{e}_2^{0/u_1} \vee e_3^{0/u_1})$. Note that the universal variable $u_2$ does not belongs to the domain of $\tau$. Also for this example the assignments which sets $u_1 \leftarrow 1$ are not allowed in IR-calc.

Our last expansion-based QBF Resolution calculi is the calculus IRM-calc [12]. We will not consider IRM-calc in this thesis, however we present it here for completeness. The calculi IRM-calc extends IR-calc by enabling annotations containing an assignment to the special symbol $*$. The symbol $*$ may be introduced by the merge rule, e.g. by collapsing $x^{0/u} \vee x^{1/u}$ into $x^{*/u}$. The rules of the calculus IRM-calc are presented in Fig. 2.6.

Axiom and instantiation rules as in IR in Figure 2.5.

$$\frac{\{x^{\tau \cup \xi}\} \cup C_1 \qquad \{\neg x^{\tau \cup \sigma}\} \cup C_2}{\mathsf{inst}(\sigma, C_1) \cup \mathsf{inst}(\xi, C_2)} \text{ (Resolution)}$$

$\mathsf{dom}(\tau)$, $\mathsf{dom}(\xi)$ and $\mathsf{dom}(\sigma)$ are mutually disjoint. $\mathsf{rng}(\tau) = \{0, 1\}$

$$\frac{C \cup \{b^{\mu}\} \cup \{b^{\sigma}\}}{C \cup \{b^{\xi}\}} \text{ (Merging)}$$

$\mathsf{dom}(\mu) = \mathsf{dom}(\sigma)$. $\xi = \{c/u \mid c/u \in \mu, c/u \in \sigma\} \cup \{*/u \mid c/u \in \mu, d/u \in \sigma, c \neq d\}$

Figure 2.6: The rules of IRM-calc [12]

Before presenting the QBF proof systems based on Frege, we briefly discuss the simulation orders among QBF Resolution calculi.

### The Simulation Order of QBF Resolution Systems

As we know that in QBF solving there exists two very different paradigms: CDCL-based and expansion-based solving. We first concentrate on just CDCL-based calculi and revist their simulation order. For simulation results in CDCL-based solvers, we need the family of false QBF formulas KBKF$(t)$ introduced by Kleine Büning et al. in [60].

**Definition 2.9** (Kleine Büning, Karpinski and Flögel [60])**.**

$$\begin{aligned}
\mathrm{KBKF}(t) \equiv\ & \exists c_0 \exists a_1 b_1 \forall x_1 \exists a_2 b_2 \forall x_2 \dots \exists a_t b_t \forall x_t \exists d_1 \dots d_t. \\
& \neg c_0 \land (c_0 \lor \neg a_1 \lor \neg b_1) \land \\
& (a_1 \lor x_1 \lor \neg a_2 \lor \neg b_2) \quad \land \quad (b_1 \lor \neg x_1 \lor \neg a_2 \lor \neg b_2) \land \\
& (a_2 \lor x_2 \lor \neg a_3 \lor \neg b_3) \quad \land \quad (b_2 \lor \neg x_2 \lor \neg a_3 \lor \neg b_3) \land \\
& \qquad \qquad \qquad \dots \\
& (a_{t-1} \lor x_{t-1} \lor \neg a_t \lor \neg b_t) \quad \land \quad (b_{t-1} \lor \neg x_{t-1} \lor \neg a_t \lor \neg b_t) \land \\
& (a_t \lor x_t \lor \neg d_1 \lor \dots \neg d_t) \quad \land \quad (b_t \lor \neg x_t \lor \neg d_1 \lor \dots \neg d_t) \land \\
& (x_i \lor d_i) \quad \land \quad (\neg x_i \lor d_i) \quad \textit{for } i \in [t]
\end{aligned}$$

It is easy to verify that KBKF$(t)$ is indeed a false QBF formula. It was shown in [13], that KBKF$(t)$ is hard for IR-calc and so is hard for Q-Res as well, since IR-calc p-simulates Q-Res [12]. But next we observe that KBKF$(t)$ has short refutation in QU-Res.

**Proposition 2.10** ( [78])**.** KBKF$(t)$ *has short refutation in QU-Res.*

*Proof.* Let us suppose that we have already derived both $a_1$ and $b_1$. Then we resolve $(c_0 \lor \neg a_1 \lor \neg b_1)$ and $a_1$ to get $(c_0 \lor \neg b_1)$. Then resolve it with $b_1$ and derive $c_0$. Finally resolve it with the axiom clause $\neg c_0$ and derive the empty clause $\Box$.

Next we show how to derive $a_i$ and $b_i$ from $a_{i+1}, b_{i+1}$ and the initial clauses in QU-Res. See Figure 2.7 for the derivations.

$$\dfrac{\dfrac{\dfrac{(a_i \lor x_i \lor \neg a_{i+1} \lor \neg b_{i+1}) \quad (a_{i+1})}{(a_i \lor x_i \lor \neg b_{i+1}) \quad (b_{i+1})}\, a_{i+1}}{(a_i \lor x_i)}\, b_{i+1}}{(a_i)}\, \forall\text{-red} \qquad \dfrac{\dfrac{\dfrac{(b_i \lor \neg x_i \lor \neg a_{i+1} \lor \neg b_{i+1}) \quad (a_{i+1})}{(b_i \lor \neg x_i \lor \neg b_{i+1}) \quad (b_{i+1})}\, a_{i+1}}{(b_i \lor \neg x_i)}\, b_{i+1}}{(b_i)}\, \forall\text{-red}$$

Figure 2.7: Deriving $a_i$ and $b_i$ from $a_{i+1}, b_{i+1}$, and the initial clauses

Observe that if we can derive $a_t, b_t$ from the initial clauses, then we are done. For this we first derive $d_i$, for all $i \in [t]$. We do this by resolving the clauses $(x_i \vee d_i)$ and $(\neg x_i \vee d_i)$ with $x_i$ as universal pivot variable. Recall that in QU-Res we can resolve on universal variables as well. Now we proceed as in Figure 2.8.

$$\begin{array}{c}
(a_t \vee x_t \vee \neg d_1 \vee \cdots \vee \neg d_t) \quad (d_t) \\ \hline
(a_t \vee x_t \vee \neg d_1 \vee \cdots \vee \neg d_{t-1}) \quad (d_{t-1}) \\ \hline
(a_t \vee x_t \vee \neg d_1 \vee \cdots \vee \neg d_{t-2}) \quad (d_{t-2}) \\ \hline
\vdots \\ \hline
(a_t \vee x_t) \\ \hline
(a_t)
\end{array}
\begin{array}{l}
\\ d_t \\ d_{t-1} \\ d_{t-2} \\ d_1 \\ \forall\text{-red}
\end{array}
\qquad
\begin{array}{c}
(b_t \vee \neg x_t \vee \neg d_1 \vee \cdots \vee \neg d_t) \quad (d_t) \\ \hline
(b_t \vee \neg x_t \vee \neg d_1 \vee \cdots \vee \neg d_{t-1}) \quad (d_{t-1}) \\ \hline
(b_t \vee \neg x_t \vee \neg d_1 \vee \cdots \vee \neg d_{t-2}) \quad (d_{t-2}) \\ \hline
\vdots \\ \hline
(b_t \vee \neg x_t) \\ \hline
(b_t)
\end{array}
\begin{array}{l}
\\ d_t \\ d_{t-1} \\ d_{t-2} \\ d_1 \\ \forall\text{-red}
\end{array}$$

Figure 2.8: Deriving $a_t$ and $b_t$ using $d_i$'s and the initial clauses.

The length of the proof is $O(t)$. □

Thus we conclude that Q-Res cannot simulate QU-Res. Also it was shown in [45, Proposition 1], that KBKF$(t)$ has an LD-Q-Res refutation of size polynomial in $t$ for $t \geq 1$. Therefore we also conclude that: Q-Res cannot simulate LD-Q-Res.

In [6, Section 3.1], incomparability results between LD-Q-Res and QU-Res were established. To establish this, Balabanov et al. considered the modification of KBKF$(t)$ formulas. By adding fresh variables in the clauses of KBKF$(t)$ they constructed a new family of false formulas which they call KBKF-qu$(t)$, and showed that KBKF-qu$(t)$ are hard for QU-Res but are still easy for LD-Q-Res. In a similar manner they again modified KBKF$(t)$ and constructed another family of false formulas which they call KBKF-lu$(t)$. They showed that KBKF-lu$(t)$ are hard for LD-Q-Res but are easy for QU-Res.

From the definition of LQU$^+$-Res, we know that LQU$^+$-Res p-simulates both LD-Q-Res and QU-Res, and since both proof systems are incomparable, we conclude that LQU$^+$-Res is exponentially stronger than both LD-Q-Res and QU-Res. This completes the entire simulation picture of CDCL-based solvers, except the simulation order of Q-Res$_\mathsf{T}$ with respect to other calculi, which we present after introducing an important family of false QBFs $\phi_n$ (see below).

Now let us start considering expansion-based proof systems as well. In [56], it has been shown that $\forall$Exp+Res p-simulates Q-Res$_\mathsf{T}$. However it turns out that $\forall$Exp+Res cannot simulate Q-Res. And also later it had been proved that even Q-Res cannot simulate $\forall$Exp+Res. We now discuss these incomparibility results in detail.

We start from the following results: $\forall$Exp+Res cannot simulate Q-Res. This result was established by Janota and Marques-Silva in [56, Theorem 4]. For this they constructed the following family of false formulas, which we denote as $\phi_n$:

$$\begin{aligned}\phi_n \equiv\ & \exists e_1 \forall u_1 \exists c_1 c_2 \ldots \exists e_n \forall u_n \exists c_{2n-1} c_{2n}. \\ & \bigwedge_{i \in [n]} (\neg e_i \vee c_{2i-1}) \wedge (\neg u_i \vee c_{2i-1}) \wedge (e_i \vee c_{2i}) \wedge (u_i \vee c_{2i}) \wedge \\ & \left( \bigvee_{i \in [2n]} \neg c_i \right)\end{aligned}$$

They proved that $\phi_n$ is hard for $\forall$Exp+Res:

**Proposition 2.11.** *[56, Proposition 3] Any $\forall$Exp+Res refutation of $\phi_n$ is exponential in $n$.*

At the same time they gave a polynomial sized Q-Res proof for $\phi_n$.

**Proposition 2.12.** *[56, Proposition 2] $\phi_n$ has a Q-Res refutation of size polynomial in $n$.*

*Proof.* For $k \in [n]$, let us denote $D_{2k} \equiv (\bigvee_{i \in [2k]} \neg c_i)$. Let $D_0$ be the empty clause. Starting from $D_{2n}$ we will derive $D_0$ in $n$ steps. In each step we derive $D_{2k-2}$ from $D_{2k}$ as shown in Figure 2.9.

□

As promised, we now immediately present the missing relation “Q-Res$_\mathsf{T}$ cannot

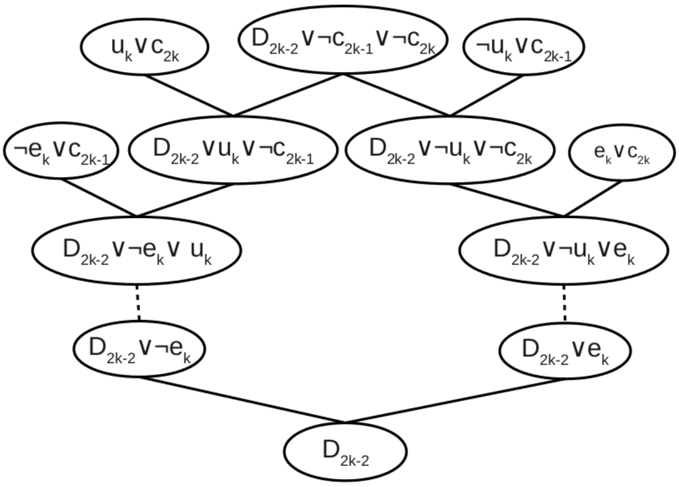

Figure 2.9: Proof of Proposition 2.12. Dashed line represents ∀-red steps. $D_{2k-2} = \neg c_1 \vee \cdots \vee \neg c_{2k-2}$.

simulate Q-Res": from Proposition 2.11, and the fact that ∀Exp+Res p-simulates Q-Res$_\mathsf{T}$ [56], we conclude that $\phi_n$ is hard for Q-Res$_\mathsf{T}$, but it is easy for Q-Res (Proposition 2.12).

Further in [13], Beyersdorff et al. established that even Q-Res can not simulate ∀Exp+Res. For this they showed that a formulation QParity$_n$ of the parity function $\oplus_n$ is hard for Q-Res [13, Section 4] but has a polynomial size proof in ∀Exp+Res [13, Lemma 15]. We will describe the formulation QParity$_n$ and its lower bound proof in Q-Res in Section 2.5.

In [12, Theorem 6, 7], it has been shown that IR-calc p-simulates both Q-Res and ∀Exp+Res. And since Q-Res and ∀Exp+Res are incomparable, we immediately conclude that IR-calc is exponentially stronger than both Q-Res and ∀Exp+Res.

In their paper [13], Beyersdorff et al. established an incomparability result between IR-calc and LD-Q-Res. For one direction they established hardness result of KBKF($t$) formulas in IR-calc. They showed that KBKF($t$) is hard for IR-calc, but it was known to be easy for LD-Q-Res [45]. So they showed that IR-calc can simulate neither LD-Q-Res nor QU-Res [13, Corollary 8]. Before discussing the other direction we point out that, in [12, Theorem 8], it has been shown that IRM-calc can p-simulate LD-Q-Res, and hence KBKF$(t)$ is also easy for IRM-calc. Thus IR-calc cannot simulate IRM-calc. (This also completes the entire simulation order within the expansion-based proof systems). For the other direction, i.e, for showing that LD-Q-Res cannot simulate IR-calc, they consider the variant of the parity function LQParity$_n$. They showed that LQParity$_n$ is hard for LD-Q-Res [13, Theorem 22], but easy for ∀Exp+Res [13, Proposition 18] and thus also easy for IR-calc. They further extend the lower bounds to the powerful LQU$^+$-Res. For this, they constructed a new family of formulas QUParity$_n$ from LQParity$_n$ by using the tricks from [6]. They showed that QUParity$_n$ requires exponential size refutations in LQU$^+$-Res [13, Theorem 23]. However, QUParity$_n$ remains easy for ∀Exp+Res. Thus we have: LQU$^+$-Res cannot simulate ∀Exp+Res, IR-calc, and IRM-calc. This completes the entire simulation results among QBF Resolution calculi. See Figure 2.10.

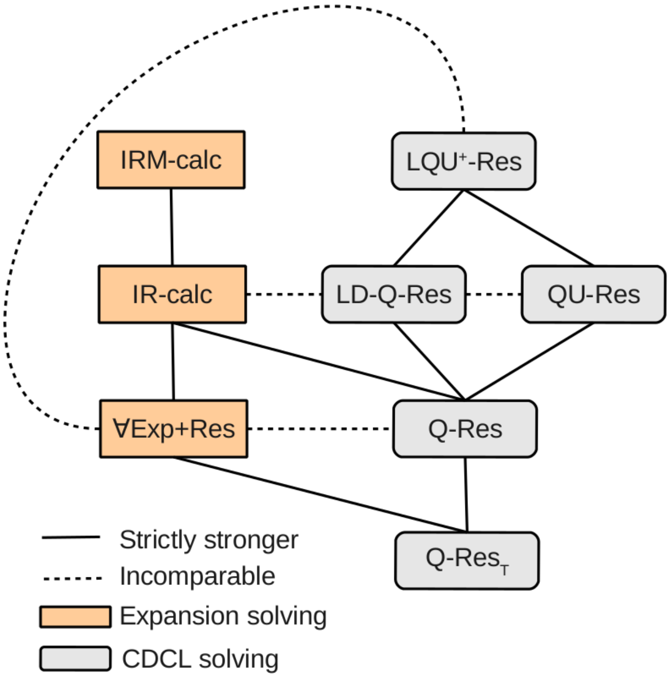

Figure 2.10: The simulation order of QBF Resolution calculi [13]

### 2.4.2 QBF Proof Systems Based on Frege

Recently Beyersdorff et al. in [11], introduced a new class of Frege systems for quantified Boolean formulas and showed strong lower bounds for restricted versions of these systems. After that Beyersdorff and Pich [20] gave a detailed analysis of the extended Frege systems for QBFs from [11], denoted EF+∀red. We next define QBF Frege from [11]. Let $\mathcal{C}$ be some circuit class.

**Definition 2.13** ($\mathcal{C}$-Frege+∀red [11])**.** *A $\mathcal{C}$-Frege+∀red refutation of a false QBF $\mathcal{Q} . \phi$ is a sequence of lines $L_1, \ldots, L_\ell$ where all lines are from the circuit class $\mathcal{C}$ and the last line is the contradiction $\perp$. Each $L_i$ in the proof is either an axiom instance, or is derived from some previous lines in the sequence using the inference rule of $\mathcal{C}$-Frege, or is derived using the ∀-red rule*

$$\frac{L_j}{L_j[u/B]}$$

*where $u$ is the innermost (highest index) variable among the variables of $L_j$, $B$ is a formula containing only variables left of $u$, and $L_j[u/B]$ is the formula obtained from $L_j$ by replacing each occurrence of $u$ in $L_j$ by $B$. Most importantly $L_j[u/B]$ belongs to the circuit class $\mathcal{C}$.*

It has been shown that $\mathcal{C}$-Frege+∀red is a complete and sound QBF proof system.

## 2.5 A Lower Bound Technique for QBFs: Strategy Extraction

Recall from Section 2.1 that a QBF $\mathcal{Q}_1 x_1 \cdots \mathcal{Q}_k x_k . \phi$ can be seen as a game between two players: *universal* (∀) and *existential* (∃). Given a universal variable $u$ with index $i$, a *strategy for $u$* is a function from all variables of index $< i$ to $\{0, 1\}$. A

QBF is false if and only if there exists a *winning strategy* for the universal player.

*Strategy extraction* is an important paradigm in QBF [5,12,45,52], which is highly desirable in practice. Winning strategies for the universal player can be very complex. But, we say that a QBF proof system has the strategy extraction property for a circuit class $\mathcal{C}$ if we can efficiently extract, from every refutation $\pi$ of a false QBF $\mathcal{F}$, $\forall$-player winning strategies in $\mathcal{C}$ for all universal variables.

Beyersdorff et al. in [13] were the first to use strategy extraction as a lower bound technique for the proof system $\mathsf{Q\text{-}Res}$ and $\mathsf{QU\text{-}Res}$. Before we present the proofs from [13], we need the following definition:

**Definition 2.14** (Decision lists [72]). *A* decision list *is a list $L$ of pairs*

$$(t_1, v_1), \ldots, (t_r, v_r),$$

*where each $t_i$ is a term (conjunction of literals) and $v_i$ is a value in $\{0,1\}$, and the last term $t_r$ is the constant term* **true** *(i.e, the empty term). A decision list $L$ defines a Boolean function as follows: for any assignment $\alpha$, $L(\alpha)$ equals $v_j$ where $j$ is the least index such that $t_j|_\alpha = 1$. (Such an item always exists, since the last term evaluates to $1$). We may think of a decision list as an extended* "if - then - elseif - . . . else" *rule.*

Observe that, if a function $f$ can be represented as a decision list $L$ of polynomial size, then $f \in \mathsf{AC}^0$ [13, Lemma 13]. In [11], Definition 2.14 has been generalised to $\mathcal{C}$-decision lists (for some circuit class $\mathcal{C}$), where instead of terms one can use circuits from $\mathcal{C}$. A $\mathcal{C}$-decision list of length $\ell$ can be converted to a circuit by noting that $f(x)$ equals $\bigvee_{i=1}^{r} \Big(v_i \wedge C_i(x) \wedge \bigwedge_{j<i} \neg C_j(x)\Big)$. In particular, for $\mathcal{C} \in \{\mathsf{AC}^0, \mathsf{TC}^0, \mathsf{NC}^1\}$ a polynomial-sized $\mathcal{C}$-decision list yields a circuit in $\mathcal{C}$.

### A Lower Bound for Q-Res Proof System via Strategy Extraction [13]

We begin by stating the fact that a winning strategy of the universal player can be extracted from a Q-Res proof in the form of decision lists very efficiently:

**Theorem 2.15** ( [4]). *Given a false QBF $\mathcal{Q} \,.\, \phi$, with $n$ variables, and a* Q-Res *proof $\pi$ of $\mathcal{Q} \,.\, \phi$ of size $|\pi|$, it is possible to extract from $\pi$ a winning strategy $\sigma_u$ for each universal variable $u \in \phi$, such that each $\sigma_u$ can be expressed as a decision list whose size is polynomial in $|\pi|$.*

*In particular, if $\mathcal{Q} \,.\, \phi$ can be refuted in* Q-Res *in $n^{O(1)}$ size, then the winning strategies can be computed in* $\mathsf{AC}^0$ *(from above discussion).*

The general idea of the lower bound technique is as follows: come up with a family of false QBFs $\varphi_f$, such that $\varphi_f$ contains a universal variable, say $z$, with a unique winning strategy $f_z \notin \mathsf{AC}^0$. Now if $\varphi_f$ has a polynomial sized Q-Res proof, then from Theorem 2.15, $f_z \in \mathsf{AC}^0$. A contradiction.

We know that the parity function $\textsc{Parity}(x_1, \ldots, x_n) = x_1 \oplus \cdots \oplus x_n$, cannot be computable in $\mathsf{AC}^0$.

**Theorem 2.16** (Furst, Saxe, Sipser [50]). $\textsc{Parity} \notin \mathsf{AC}^0$.

By keeping Theorem 2.16 in mind, in [13], Beyersdorff et al. constructed the required family of false QBFs $\varphi_f$: consider the family of sentences expressing the following: $\exists x_1 \ldots x_n \forall z. \textsc{Parity}(\vec{x}) \neq z$. If we want to encode this sentence in QBF form with CNF matrix, we need to use some extra existential variables, as $\textsc{Parity} \notin \mathsf{AC}^0$. Let $(x \equiv y \oplus z)$ be the set of clauses which encodes that $x$ is equal to $y \oplus z$. In particular, it consists of the following set of clauses: $\{(\neg x \vee \neg y \vee \neg z), (\neg x \vee y \vee z), (x \vee \neg y \vee z), (x \vee y \vee \neg z)\}$

They define the parity formula $QPARITY_n$ as follows:

$$\exists x_1 \ldots x_n \forall z \exists t_2 \ldots t_n.\ (t_2 \equiv x_1 \oplus x_2) \wedge \bigwedge_{i=3}^{n} (t_i \equiv x_i \oplus t_{i-1})\ \wedge (t_n \neq z)$$

Notice that without the $(t_n \neq z)$ clauses, the sentence is true, and each $t_i$ in a satisfying assignment must be $x_1 \oplus \cdots \oplus x_i$. Also note that the formula itself is of size $\theta(n)$. Clearly the only winning strategy for the universal variable $z$ is the function $\textsc{Parity}(\vec{x}) \notin \mathsf{AC}^0$. Thus if $\textsc{QParity}_n$ has a polynomial sized Q-Res proof, then $\textsc{Parity} \in \mathsf{AC}^0$ from Theorem 2.15. A contradiction. So we have:

**Theorem 2.17** ( [13]). *Any Q-Res refutation of* $\textsc{QParity}_n$ *is of exponential size in* $n$*.*

**A Lower Bound for $\mathcal{C}$-Frege+$\forall$red Proof Systems via Strategy Extraction** Fix some circuit class $\mathcal{C}$. In [11], Beyersdorff et al. defined QBF proof systems based on restricted Frege; denoted $\mathcal{C}$-Frege+$\forall$red, proved that $\mathcal{C}$-Frege+$\forall$red admits strategy extraction in the circuit class $\mathcal{C}$, and thereby transform circuit lower bounds to proof size lower bounds for these proof systems. We first state the strategy extraction theorem for $\mathcal{C}$-Frege+$\forall$red.

**Theorem 2.18** ( [11]). *Given a false QBF $\mathcal{Q} \,.\, \phi$ and a $\mathcal{C}$-Frege+$\forall$red refutation $\pi$ of $\mathcal{Q} \,.\, \phi$, it is possible to extract from $|\pi|$ a collection of $\mathcal{C}$-decision lists of sizes polynomial in $|\pi|$, which computes a winning strategy of the universal variables of $\phi$.*

*In particular if $|\pi| = \ell$, then each of the $\mathcal{C}$-decision list can be converted into a circuit $C \in \mathcal{C}$ of size $O(\ell)$.*

For proving lower bounds, they come up with a family of false QBFs $\mathcal{Q}$-$C_n$ based on some circuit $C_n$ computing a function $f(\vec{x})$, such that $\mathcal{Q}$-$C_n$ has a universal variable $u$ with a unique winning strategy $u \leftarrow C_n(\vec{x})$. Now from Theorem 2.18, we know that if $\mathcal{Q}$-$C_n$ has a $\mathcal{C}$-Frege+$\forall$red proof of size bounded by a function $q(n)$, then for every $n$, $C_n$ is actually equivalent to a circuit $C'_n$ of size $O(q(n))$ that uses the gates and depth allowed in $\mathcal{C}$. They proved something stronger for such QBFs. They showed that if $(C_n)_{n\in\mathbb{N}}$ is a polynomial size circuit family from $\mathcal{C}$ then $\mathcal{Q}$-$C_n$ have polynomial size refutation in $\mathcal{C}$-Frege+$\forall$red [11].

In particular (by combining the above two statements) they showed that a Boolean function $f$ is computable by a polynomial size $\mathcal{C}$ circuit iff $\mathcal{Q}$-$C_n$ have polynomial size $\mathcal{C}$-Frege+$\forall$red refutations for each choice of Boolean circuits $(C_n)_{n\in\mathbb{N}}$ computing $f$. Before using this fact to prove lower bound results we explicitly define the $\mathcal{Q}$-$C_n$ formulas from [11].

**Definition 2.19** ( [11]). *Let $C_n$ be a circuit with inputs $x_1, \dots, x_n$. We define*

$$Q\text{-}C_n \equiv \exists x_1 \cdots x_n \forall u \exists t_1 \cdots t_l.(t_l \neq u) \wedge \bigwedge_{i=1}^{l} (t_i \textit{ is consistent with the inputs to gate } i).$$

*The inner formula can be written as an $O(l)$-sized CNF. For example, if we have a gate $g_i = g_j \wedge g_k$ in the circuit, then the clauses equivalent to $t_i \equiv t_j \wedge t_k$ are added to $Q$-$C_n$, where each $t_i$ gets the value of the gate $g_i$ in the circuit $C_n$ on input $\vec{x}$. Observe that this is just a Tseitin transformation (see Section 2.1).*

Informally $Q$-$C_n$ expresses the false sentence that there exists an input $\vec{x}$ such that $C(\vec{x})$ evaluates to both 1 and 0. Obviously the unique winning strategy for $u$ is $C(\vec{x})$.

We now present an important result from [11], which uses strategy extraction and proves an exponential lower bound for $\mathsf{AC}^0[p]$-Frege+$\forall$red.

**Corollary 2.20** ( [11]). *Let $C_n$ be a family of polynomial-size circuits computing* PARITY$(x_1, \dots, x_n)$. *For each odd prime $p$ the QBFs $\mathcal{Q}$-$C_n$ require proofs of exponential size in $\mathsf{AC}^0[p]$-Frege+$\forall$red.*

*Proof.* If $\mathcal{Q}$-$C_n$ has polynomial size proofs in $\mathsf{AC}^0[p]$-Frege+$\forall$red, then from Theorem 2.18, and the discussions above, $C_n$ is equivalent to some circuit $C'_n \in \mathsf{AC}^0[p]$ computing PARITY. However this is impossible, because it has been shown that circuits belonging to $\mathsf{AC}^0[p]$ class cannot compute parity. To be precise, it has been proved that for each odd prime $p$ any family of bounded-depth circuits with MOD$_p$ gates computing PARITY must be of exponential size [71, 75]. □

For further applications of strategy extraction, interested readers are referred to [11, 20]. We end this Chapter with the following Comment:

**Comment:** In propositional case, proving exponential lower bounds for $\mathsf{AC}^0[p]$-Frege is an open problem. Thus Corollary 2.20 is an important result which translates circuit lower bounds for parity [71, 75] to QBF proof complexity lower bounds for $\mathsf{AC}^0[p]$-Frege+$\forall$red. Such transfer of lower bound results are missing for stronger systems in the propositional case.

# Chapter 3

# Level-ordered Q-Res and Tree-like Q-Res are Incomparable

In this Chapter, we prove that level-ordered $Q$-resolution and tree-like $Q$-resolution ($\mathsf{Q\text{-}Res_T}$), two restrictions of the $Q$-resolution system ($\mathsf{Q\text{-}Res}$) for proving false QBFs false, are incomparable. While the $\forall\mathsf{Exp{+}Res}$ system is known to p-simulate tree-like $Q$-resolution [56], we observe that it cannot simulate level-ordered $Q$-resolution. On the other hand, it is well known that level-ordered $Q$-resolution cannot simulate $\forall\mathsf{Exp{+}Res}$. Therefore we conclude that even $\forall\mathsf{Exp{+}Res}$ and level-ordered $Q$-resolution are incomparable.

We highlight that apart from theoretical interests the incomparability result has practical significance as well. To be precise, the result shows that the expansion-based QBF solvers, for example **RAReQS** [55], and the CDCL-based QBF solvers, for example **Evaluate**, introduced in [31] (also see [32]), are indeed orthogonal paradigms.

## 3.1 Introduction

As discussed in Chapter 2, Q-Res, the core CDCL-based proof system, and ∀Exp+Res, the core expansion-based proof system are incomparable. Looking at how incomparability were established, we see that two sub-classes of $Q$-resolution (Q-Res) are significant: tree-like proofs, where the graph underlying the resolution structure is a tree, and level-ordered proofs (see Section 3.2), where at each resolution step, the variable on which resolution is performed is at the rightmost level (quantifier block) among all existential variables in the clauses involved. The known results were established in the following chronological order.

1. ∀Exp+Res proof system cannot simulate Q-Res. As already mentioned in Chapter 2, this was established by Janota and Marques-Silva in 2013 [57] (also see [56]). The family of false QBFs witnessing this is $\phi_n$ (See Section 2.4.1).

2. Level-ordered $Q$-resolution cannot simulate ∀Exp+Res.

   This too was shown by Janota and Marques-Silva in [56]. They define a false QBF sentence $CR_n$ (Section 3.2) and proved that $CR_n$ is hard for level-ordered $Q$-resolution [56, Proposition 5] but has a polynomial size proof in ∀Exp+Res [56, Proposition 4].

3. Q-Res cannot simulate ∀Exp+Res.

   This was shown by Beyersdorff et al. in [13]. As seen in chapter 2, the family of false QBFs witnessing this is $\text{QP}{\scriptsize\text{ARITY}}_n$ (Section 2.5).

4. ∀Exp+Res can p-simulate tree-like $Q$-resolution.

   This was shown by Janota and Marques-Silva in 2013 [58, Section 3] (also see [56]). The converse direction is ruled out by the $\text{QP}{\scriptsize\text{ARITY}}_n$ formulas. As already discussed, since $\phi_n$ is hard for ∀Exp+Res, it follows that $\phi_n$ is hard for tree-like $Q$-resolution as well.

Now we prove the main Theorem of this Chapter.

**Theorem 3.1.** *Tree-like $Q$-resolution and level-ordered $Q$-resolution are incomparable.*

If we consider sentences with only existential quantifiers, then a Q-Res proof is just a proof in propositional Resolution. In fact, every resolution proof is level-ordered, since all variables are at the same level. Also note that for sentences with only existential quantifiers, ∀Exp+Res also becomes propositional Resolution. Results from propositional Resolution thus imply that there are sentences (with only existential quantifiers) where ∀Exp+Res, Q-Res and level-ordered $Q$-resolution are exponentially more powerful than tree-like $Q$-resolution [25]. However, for QBFs (i.e, not just with existential variables) the simulation order among the QBF proof systems becomes more interesting. Another refinement of propositional resolution proofs is ordered resolution, where the variables are resolved in a specified order. This is known to be incomparable with tree-like resolution [59] (see also [25,28]). In the context of QBFs, level-ordered is a weaker restriction than ordered, since no order is imposed on variables in the same quantifier block. We note that Theorem 3.1 has practical importance as well: it underlines the fact that QBF solvers limit themselves greatly by assigning variables in the prefix order.

One direction of Theorem 3.1 is obtained as follows: observe that the known polynomial size Q-Res proof of $\phi_n$ (Proposition 2.12, also mentioned in item (1) of the chronological order above) is also in fact level-ordered. Therefore $\phi_n$ has a short level-ordered $Q$-resolution proof. Since $\phi_n$ is hard for tree-like $Q$-resolution (item (4) above), and ∀Exp+Res (item (1) above), we conclude that tree-like $Q$-resolution cannot simulate level-ordered $Q$-resolution. Furthermore, we conclude that even ∀Exp+Res cannot simulate level-ordered $Q$-resolution.

For the other direction, we show in the rest of this Chapter that the sentences $CR_n$ (item (2) above) have polynomial size tree-like $Q$-resolution proofs (Section

3.3). As $CR_n$ is hard for level-ordered $Q$-resolution, we conclude that level-ordered $Q$-resolution cannot simulate tree-like $Q$-resolution.

This completes the entire picture of relations among the above mentioned proof systems. See Figure 3.1.

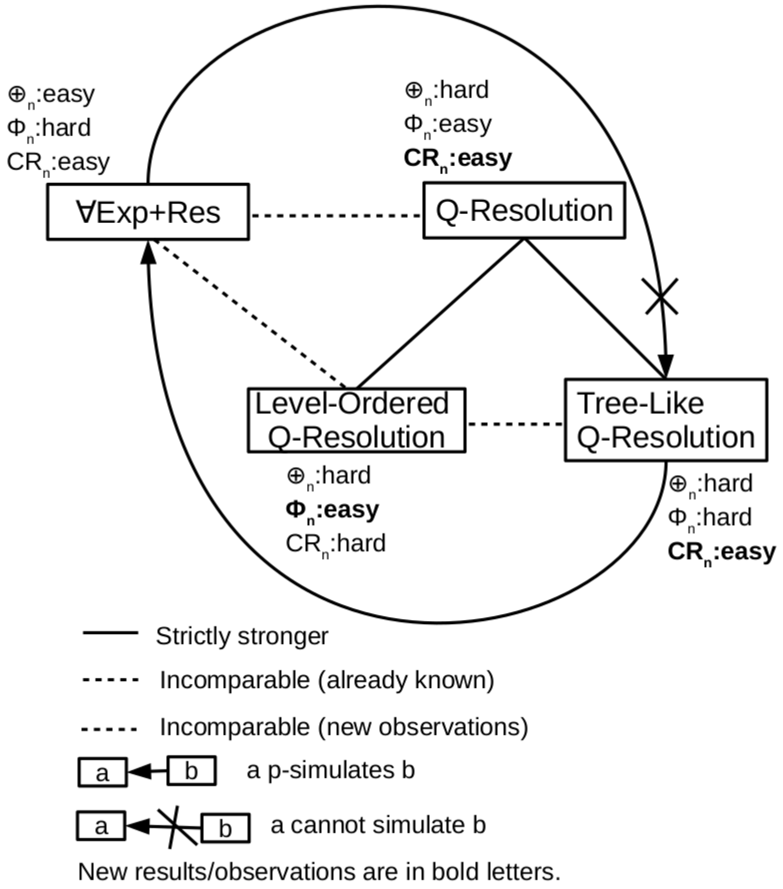

Figure 3.1: Relationships among some QBF Resolution systems

## 3.2 Definitions

We briefly describe level-ordered $Q$-resolution, and the sentence $CR_n$.

**Definition 3.2** (Level-ordered $Q$-resolution). *Any Q-Res proof $\pi$ is said to be level-ordered iff for every resolution step $\frac{(x \vee C) \quad (\neg x \vee D)}{C \vee D}$ in $\pi$ the following holds: $lv(y) \leq lv(x)$, for any existential variable $y \in \mathrm{var}(C \vee D)$.*

**Completion Principle and the Sentence $CR_n$ ( [56]) :**

Consider two sets $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$, and depict their cross product $A \times B$ as in Table 3.1. The following two-player game is played on Table 3.1.

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_1$ | $a_1$ | $\ldots$ | $a_1$ | $a_2$ | $a_2$ | $\ldots$ | $a_2$ | $\ldots$ | $\ldots$ | $a_n$ | $a_n$ | $\ldots$ | $a_n$ |
| $b_1$ | $b_2$ | $\ldots$ | $b_n$ | $b_1$ | $b_2$ | $\ldots$ | $b_n$ | $\ldots$ | $\ldots$ | $b_1$ | $b_2$ | $\ldots$ | $b_n$ |

Table 3.1: Completion Principle.

In the first round, player 1 deletes exactly one cell from each column. In the second round, player 2 chooses one of the two rows. Player 2 wins if the chosen row contains either the complete set $A$ or the set $B$; otherwise player 1 wins. It is well known that player 2 has a winning strategy: suppose, after player 1 plays, some $a_i$ is missing in the top row. Then the entire set $B$ below the $a_i$ chunk is present in the bottom row and so player 2 chooses the bottom row to win. Otherwise, no $a_i$ is missing in the top row, so player 2 can win by choosing the top row. This fact (that player 2 can always win) is called the completion principle.

Based on the completion principle, the false sentence CR$_n$ is formulated to express the notion that player 1 has a winning strategy. For each column $\begin{bmatrix} a_i \\ b_j \end{bmatrix}$ of the Table 3.1 (denote this the $(i, j)^{th}$ column), there is a boolean variable $x_{i,j}$. Let $x_{i,j} = 0$ denote that player 1 'deletes $b_j$' (i.e, keeps $a_i$) from the $(i, j)^{th}$ column, and $x_{i,j} = 1$ denotes that player 1 keeps $b_j$ in the $(i, j)^{th}$ column. There is a variable $z$ to denote the choice of player 2: $z = 0$ means 'choose top row'. The Boolean variables $a_i$, $b_j$, for $i, j \in [n]$ encode that for the chosen values of all the $x_{k,\ell}$, and the row chosen via $z$, at least one copy of the element $a_i$ and $b_j$ respectively is kept. (eg. $(x_{i,j} \wedge z) \Rightarrow b_j$). Let $\tilde{x}$, $\tilde{a}$ and $\tilde{b}$ stands for the vector of variables $\{x_{1,1}, x_{1,2}, \ldots, x_{n,n}\}$, $\{a_1, \ldots, a_n\}$, and $\{b_1, \ldots, b_n\}$ respectively. Now $CR_n$ can be framed as follows:

$$\exists \tilde{x}_{i,j} \; \forall z \; \exists \tilde{a} \exists \tilde{b} \; \left( (\tilde{a}, \tilde{b} \text{ consistent with } \tilde{x}, z) \; \wedge \bigvee_i \bar{a}_i \wedge \bigvee_j \bar{b}_j \right)$$

The inner formula can be expressed as the conjunction of the following clauses:

$$\text{For } i,j \in [n], C_{i,j}: \qquad (x_{i,j} \vee z \vee a_i) \tag{3.1}$$

$$\text{For } i,j \in [n], D_{i,j}: \qquad (\bar{x}_{i,j} \vee \bar{z} \vee b_j) \tag{3.2}$$

$$\bigvee_{i\in[n]} \bar{a}_i \tag{3.3}$$

$$\bigvee_{i\in[n]} \bar{b}_i \tag{3.4}$$

## 3.3 Tree-like $Q$-resolution Proof for $CR_n$

Observe that to begin we cannot apply the ∀-Red rule because the only universal variable $z$ has been blocked, in all clauses where it appears, by existential variables from $\tilde{a}$ and $\tilde{b}$. We also cannot resolve any $C_{i,j}$ and $D_{i,j}$ on variable $x_{i,j}$ because the resolvent is a tautology, which is not allowed in $Q$-resolution. We are thus forced to resolve on $\tilde{a}$ and $\tilde{b}$ variables initially.

We proceed as follows: We derive $\bar{z}$, and then apply a ∀-Red to derive $\Box$. To derive $\bar{z}$, we first derive each of the clauses $W_j = \bar{z} \vee b_j$ in a distinct tree $T_j$. Then we can put together these trees with the clause from (3.4), and in $n$ resolution steps, obtain $\bar{z}$, as follows: let $C_1$ denote the clause (3.4). For $\ell \in [n]$, resolve $C_\ell$ and $W_\ell$ (on variable $b_\ell$) to get $C_{\ell+1}$. Note that for $\ell > 1$, $C_\ell$ has the form $\bar{z} \vee \bigvee_{k\geq\ell} \bar{b}_k$. So $C_{n+1}$ is $\bar{z}$ as desired.

Now we describe the trees $T_j$ that derive $W_j = \bar{z} \vee b_j$. We first derive the clause $x_{1,j} \vee x_{2,j} \vee \ldots \vee x_{n,j} \vee z$ in a tree $T'_j$ described later. Now the ∀-Red rule is applicable, since all the $\tilde{x}$ variables are quantified before $z$. Thus we can obtain the clause $Y_{1,j} = x_{1,j} \vee x_{2,j} \vee \ldots \vee x_{n,j}$. Now for $\ell \in [n]$, resolve $Y_{\ell,j}$ with the clause $D_{\ell,j}$ from (3.2) (on variable $x_{\ell,j}$ ) to get $Y_{\ell+1,j}$. Note that for $\ell > 1$, $Y_{\ell,j}$ has the form $\bar{z} \vee b_j \vee \bigvee_{k\geq\ell} x_{k,j}$. So $Y_{n+1,j}$ is $\bar{z} \vee b_j$ as desired.

It remains to describe tree $T'_j$ deriving $x_{1,j} \vee x_{2,j} \vee \ldots \vee x_{n,j} \vee z$. This is similar to the above step, using clause 3.3 which we shall denote $Z_{1,j}$ along with the clauses $C_{\ell,j}$ from 3.1. For $\ell \in [n]$, resolve $Z_{\ell,j}$ and $C_{\ell,j}$ on variable $(a_\ell)$ to get $Z_{\ell+1,j}$. For $\ell > 1$, $Z_{\ell,j}$ has the form $z \vee \bigvee_{k<\ell} x_{k,j} \vee \bigvee_{k\geq\ell} \bar{a}_k$. So $Z_{n+1,j}$ is $z \vee \bigvee_{k\in[n]} x_{k,j}$ as desired.

**Size of the Refutation:** Each $T'_j$ has $n$ resolution steps. Each $T_j$ has $T'_j$, one $\forall$-reduction, and then $n$ more resolution steps. Once all $T_j$'s are constructed, we use another $n$ resolutions steps followed by one last $\forall$-reduction. Overall, there are $n(2n + 1)$ resolution steps and $n + 1$ $\forall$-reductions. Thus the total refutation size is $O(n^2)$.

# Chapter 4

# A New QBF Proof System Based on Cutting Planes

In this Chapter, we introduce a complete and sound QBF proof system CP+∀red that works with quantified set of linear inequalities, where each variable is either quantified existentially or universally in a quantifier prefix. The lines in CP+∀red proof are linear inequalities. The system CP+∀red extends the classical Cutting Planes system with one single ∀-reduction rule allowing manipulation of universally quantified variables.

Inspired by the recent work on semantic Cutting Planes [47] we also define a stronger system semCP+∀red where in addition to universal reduction all semantically valid inferences between inequalities are allowed (Section 4.4).

We compare our new system CP+∀red with previous QBF Resolution and Frege systems. We also establish *strategy extraction* technique for both CP+∀red and semCP+∀red. We establish the *feasible interpolation* technique for these systems in Chapter 5.

We start by defining our new proof system CP+∀red.

## 4.1 The CP+$\forall$red Proof System

In this section we define a QBF analogue of the classical Cutting Planes proof system by augmenting it with a reduction rule for universal variables. We denote this system by CP+$\forall$red. Consider a false quantified set of inequalities

$$\varphi \equiv \mathcal{Q}_1 x_1 \dots \mathcal{Q}_n x_n.\ F,$$

where $F$ is a set of linear inequalities of the form $\sum x_i a_i \geq A$ for integers $a_i$ and $A$, and $F$ includes the set of inequalities $B = \{x_i \geq 0, -x_i \geq -1 \mid i \in [n]\}$. The inequalities in $B$ are called the Boolean axioms, because they force any integer-valued assignment $\bar{a}$ to the variables to take only $0, 1$-values if it is to satisfy all inequalities in $F$. We point out that classical Cutting Planes proof systems (only existential variables) can refute any inconsistent set of linear inequalities over integers. However, once universal quantification is allowed, dealing with an unbounded domain is more messy. Since our primary goal in defining this proof system is to refute false QBFs, and since QBFs have only Boolean variables, we only consider sets of inequalities that contain $B$.

**Definition 4.1** (CP+$\forall$red proofs for inequalities). *Consider a set of quantified inequalities $\varphi \equiv \mathcal{Q}_1 x_1 \dots \mathcal{Q}_n x_n.\ F$, where $F$ also contains the Boolean axioms. A CP+$\forall$red refutation $\pi$ of $\varphi$ is a quantified sequence of linear inequalities*

$$\mathcal{Q}_1 x_1 \dots \mathcal{Q}_n x_n . [I_1, I_2, \dots, I_l]$$

*where the quantifier prefix is the same as in $\varphi$, $I_l$ is an inequality of the form $0 \geq C$ for some positive integer $C$, and for every $j \in \{1, \dots, l\}$,*

- *$I_j \in F$, or*

- *$I_j$ is derived from earlier inequalities in the sequence (for example, $I_{k_1}, I_{k_2}$,* *with $k_1, k_2 < j$) via one of the following inference rules:*

1. **Addition**: *From* $\sum_k c_k x_k \geq C$ *and* $\sum_k d_k x_k \geq D$ *derive* $\sum_k (c_k + d_k) x_k \geq C + D$.

2. **Multiplication**: *From* $\sum_k c_k x_k \geq C$ *derive* $\sum_k d c_k x_k \geq dC$, *where* $d \in \mathbb{Z}^+$.

3. **Division**: *From* $\sum_k c_k x_k \geq C$ *derive* $\sum_k \frac{c_k}{d} x_k \geq \left\lceil \frac{C}{d} \right\rceil$, *where* $d \in \mathbb{Z}^+$ *divides each* $c_k$.

4. $\forall$**-red**: *From* $\sum_{k \in [n] \setminus \{i\}} c_k x_k + h x_i \geq C$ *derive* $\begin{cases} \sum_{k \in [n] \setminus \{i\}} c_k x_k \geq C & \text{if } h > 0; \\ \sum_{k \in [n] \setminus \{i\}} c_k x_k \geq C - h & \text{if } h < 0. \end{cases}$

   *This rule can be used provided variable $x_i$ is universal, and provided all existential variables $y$ with nonzero coefficients in the hypothesis have* $\mathrm{ind}(y) < \mathrm{ind}(x_i)$. *(That is, if $x_j$ is existential and $c_j \neq 0$, then $j < i$. Observe that when $h > 0$, we are replacing $x_i$ by* 0*, and when $h < 0$, we are replacing $x_i$ by* 1*. We say that the universal variable $x_i$ has been reduced.*

*Each inequality $I_j$ is a line in the proof $\pi$. Note that proof lines are always of the form $\sum_k c_k x_k \geq C$ for integer-valued $c_k, C$. The length of $\pi$ (denoted $|\pi|$) is equal to the number of lines in it, and the size of $\pi$ (denoted size($\pi$)) is equal to the bit-size of a representation of the proof (this depends on the number of lines and the binary length of the numbers in the proof).*

In order to use CP+$\forall$red as a refutational system for QBFs in prenex form with CNF matrix, we must translates QBFs into quantified set of inequalities.

**Definition 4.2** (Encoding QBFs as inequalities)**.** *In Definition 2.6 we showed how to encode a CNF formula $\phi$ as a set of inequalities $F_\phi$. Recall we defined $R(x) = x$, $R(\bar{x}) = 1 - x$. A clause $C \equiv (l_1 \vee \cdots \vee l_k)$ is translated into the linear inequality*

*$R(C) \equiv \sum_{i=1}^{k} R(l_i) \geq 1$. A CNF formula $\phi = C_1 \wedge \cdots \wedge C_m$ is represented as the set of inequalities $F_\phi = \{R(C_1), R(C_2), \ldots, R(C_m)\} \cup B$, where $B$ is the set of Boolean axioms $x \geq 0, -x \geq -1$ for each variable $x$. Recall the encoding is called the standard encoding. For a QBF $\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ \phi$ with a CNF matrix $\phi$, the encoding is the quantified set of linear inequalities $\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.\ F_\phi$.*

We say that a 0, 1-assignment $\alpha$ satisfies the inequality $I \equiv \sum_{i=1}^{n} a_i x_i \geq b$ (i.e, $I|_\alpha = 1$), if $\sum_{i=1}^{n} a_i \alpha_i \geq b$, where $\alpha_i$ is the value given to the variable $x_i$ by $\alpha$. Observe that for any clause $C$, an assignment satisfies $C$ if and only if it satisfies $R(C)$. This observation along with the fact that the encoding includes all Boolean axioms immediately yields the following:

**Proposition 4.3.** *Let $\mathcal{Q} \,.\, \phi$ be a QBF in closed prenex CNF form, and let $\varphi = \mathcal{Q} \,.\, F_\phi$ be its encoding as a quantified set of linear inequalities. Then $\mathcal{Q} \,.\, \phi$ is false if and only if $\varphi$ is false.*

Analogous to QBFs, we can also play the 2-player game on the encoding $\varphi$ of a QBF. Players choose 0-1 values for their variables in the order defined in the prefix. The $\forall$ player wins if the assignment so constructed violates some inequality in $F$. As before, the universal player has a winning strategy exactly when $\varphi$ is false, and otherwise the existential player has a winning strategy.

**Definition 4.4** (CP+∀red proofs for QBFs)**.** *Let $\mathcal{Q} \,.\, \phi = \mathcal{Q}_1 x_1 \cdots \mathcal{Q}_n x_n \,.\, \phi$ be a false QBF in prenex CNF form, and let $\varphi$ be its encoding as a quantified set of linear inequalities. A CP+∀red (refutation) proof of $\mathcal{Q} \,.\, \phi$ is a CP+∀red proof of $\varphi$ as defined in Definition 4.1.*

It is worth noting that CP+∀red for inequalities, as defined in Definition 4.1, can start with encodings of QBFs, but can also start with quantified sets of inequalities that contain the Boolean axioms but do not correspond to any QBF, since the initial non-Boolean inequalities can have arbitrary integer coefficients.

Now we show that CP+∀red is a complete and sound proof system for false QBFs. That is, we show that if $\mathcal{F}$ is a false QBF, then there exists a CP+∀red refutation of $\mathcal{F}$ (completeness) and if there exists a CP+∀red refutation of $\mathcal{F}$, then $\mathcal{F}$ is false (soundness).

To show completeness, we first compare CP+∀red to the known QBF calculus QU-Res.

**Lemma 4.5.** *CP+∀red p-simulates QU-Res.*

*Proof.* We know that the rules of the propositional cutting planes system can p-simulate the resolution rule [37]. That is, if $C$ can be derived from $\phi$ in Res, then $R(C)$ can be derived from $F_\phi$ in Cutting Planes. Observe that the same simulation works independent of the quantifier prefix or the nature of the pivot variable. Now we show how CP+∀red simulates the ∀-red rule of QU-Res proof system. Consider a ∀-red step in QU-Res of the form $\frac{C \vee u}{C}$, where $u$ is universal and all existential variables in the clause $C$ come before $u$ in the prefix. By induction we have derived the inequality $R(C \vee u)$ for the clause $C \vee u$. Reducing $u$ from this inequality is valid. Clearly, the coefficient of $u$ in the inequality $R(C \vee u)$ is $+1$. Hence in the CP+∀red proof, using the ∀-red rule assigns $u = 0$ and hence derives $R(C)$. Similarly, for $\frac{C \vee \bar{u}}{C}$, the coefficient of $u$ in the inequality $R(C \vee \bar{u})$ is $-1$ (the variable $u$ contributes $(1-u)$ to $R(C \vee \bar{u})$), hence the ∀-red rule in CP+∀red sets $u = 1$ and again derives $R(C)$. □

Since QU-Res is known to be complete, we obtain completeness for CP+∀red for false QBFs.

Before proving the soundness of CP+∀red, we first show a normal form for proofs; this makes establishing soundness cleaner. Observe that in the ∀-red step of CP+∀red, if $u$ is the universal variable being reduced, then $u$ need not be the rightmost variable with a non-zero coefficient. There may be universal variables to the right of $u$

with non-zero coefficients. This is as allowed in QU-Res = Res + ∀-Red. Inspired by the ∀-red step defined for the Frege+∀red proof system in [11], let us consider a reduction step where we allow only the innermost (rightmost) universal variable to be reduced from any inequality. That is, reduce a universal variable $x_i$ from an inequality $I$ only if no variable to the right of $x_i$, **existential or universal**, has non-zero coefficient in $I$. We call a proof where the ∀-red steps are applied only to the innermost universal variables with non-zero coefficients a **normal-form** CP+∀red proof. We show below that any CP+∀red proof can be efficiently converted to one in normal form. In later sections we will often assume this normal form.

**Lemma 4.6.** *Any CP+∀red proof $\pi$ can be converted into a normal-form CP+∀red proof $\pi'$ in polynomial time.*

*Proof.* Let $\pi$ be any CP+∀red proof of a false QBF $\mathcal{F}$. We efficiently convert $\pi$ into a normal-form proof $\pi'$ using the Boolean axioms. Let inequality $I'$ be derived in $\pi$ from $I$ by a ∀-reduction step on $w$. If $w$ is the innermost universal variable in $I$, then nothing needs to be done. Otherwise, in any case, no existential variable right of $w$ can have non-zero coefficient in $I$. Let $(w =)w_0, w_1, \ldots, w_k$ be the universal variables right of (including) $w$ with non-zero coefficients $h_0, h_1, \ldots, h_k$ in $I$. We obtain $I'$ from $I$ via the following $(3k+1)$ steps:
For $j = k$ down to 0, reduce $w_j$.
For $j = 1$ up to $k$, if $h_j > 0$ then add $h_j(w_j \geq 0)$, else add $(-h_j)(-w_j \geq -1)$.
Observe that this proof fragment is in normal-form. □

Note that when using CP+∀red to simulate QU-Res as in Lemma 4.5, this proof fragment corresponds to a sequence of $k+1$ ∀-reductions followed by a sequence of $k$ weakenings (recall in the Resolution proof system using weakening rules one can include any number of positive or negative literals on the given clause, similarly here after $k+1$ ∀-reductions steps we are adding back the universal variables with the same coefficients as before to the given inequality).

Now we prove the soundness of $\mathsf{CP}+\forall\mathsf{red}$.

**Lemma 4.7.** *$\mathsf{CP}+\forall\mathsf{red}$ is a sound proof system for false QBFs.*

*Proof.* Let $\mathcal{Q} \,.\, \phi = \mathcal{Q}_1 x_1 \cdots \mathcal{Q}_n x_n \,.\, \phi$ be a QBF in closed prenex CNF form, and let $\varphi = \mathcal{Q}.\, F$ be its encoding as inequalities. Recall that $F$ also includes Boolean axioms. Let $\pi = \mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n . [I_1, I_2, \ldots, I_l]$ be any $\mathsf{CP}+\forall\mathsf{red}$ refutation (see Definition 4.1) of $\varphi$. We can assume (using Lemma 4.6) that $\pi$ is in normal form.

To prove soundness, we need to show that $\mathcal{Q} \,.\, \phi$ is false. From Proposition 4.3, it suffices to show that $\varphi$ is false. We do this by showing that the following is valid for each $j \in [l]$:

$$\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n .\ [F \wedge I_1 \wedge \cdots \wedge I_{j-1}] \implies \mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n .\ [F \wedge I_1 \wedge \cdots \wedge I_{j-1} \wedge I_j],$$

where $I_j$ is derived from some inequalities before it via an inference rule of $\mathsf{CP}+\forall\mathsf{red}$. Observe that the cases when $I_j$ is derived via Addition, Multiplication, or Division rules are straightforward, since every Boolean assignment satisfying $F \wedge I_1 \wedge \cdots \wedge I_{j-1}$ also satisfies $I_j$. We now concentrate on the $\forall$-red step.

Say $I_j$ is derived from $I_k$, $k < j$, via the $\forall$-red rule. Let $u = x_r$ be the universal variable reduced, and let $I_k$ be $\sum_s c_s x_s \geq C$ for some integers $c_1, \ldots, c_n, C$. Since $\pi$ is in normal form, for all $s > r$, $c_s = 0$.

Suppose the claimed statement is not valid. That is, $\varphi_{j-1} = \mathcal{Q} \,.\, F \wedge I_1 \wedge \cdots \wedge I_{j-1}$ is true but $\varphi_j = \mathcal{Q} \,.\, F \wedge I_1 \wedge \cdots \wedge I_j$ is false. The existential player has a winning strategy $\sigma_\exists$ for $\varphi_{j-1}$, while the universal player has a winning strategy $\sigma_\forall$ for $\varphi_j$. Let $\alpha$ be the assignment constructed when the players use these strategies for their variables. Then $\alpha$ satisfies $F \wedge I_1 \wedge \cdots \wedge I_{j-1}$, and in particular, $I_k$, but does not satisfy $I_j$. Define a new strategy $\sigma'_\forall$ for the universal player; it uses the same strategy as $\sigma_\forall$ for variables other than $x_r$, but flips the strategy of $\sigma_\forall$ for variable $x_r$. Let $\beta$ be the assignment constructed by strategies $\sigma_{\exists}$ and $\sigma'_{\forall}$. Then $\beta_s = \alpha_s$ for all $s < r$, and $\beta_r \neq \alpha_r$. These are the only values that matter for evaluating $I_k$. An examination of the ∀-red rule shows that it derives the tighter of the two inequalities $I_k|_{x_r=0}$ and $I_k|_{x_r=1}$ as $I_j$, and hence $I_k(\beta)$ equals $I_j(\alpha)$ and is false. Thus the existential player using strategy $\sigma_{\exists}$ does not win against the universal player using strategy $\sigma'_{\forall}$, and hence is not a winning strategy for $\varphi_{j-1}$, a contradiction.

Now let us assume that $\varphi$ is true, then we conclude that $\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_n x_n.[I_1, I_2, \ldots, I_l]$ is also true. A contradiction, as the last inequality $I_l \equiv 0 \geq C$ is always false. □

**Comments:** We have seen that the new QBF proof system CP+∀red is sound and complete for the language of false QBFs. We point out that CP+∀red is also sound and complete for the language of false quantified set of linear inqualities that contains the Boolean axiom, but do not corresponds to any QBF. Interested readers are referred to [34].

## 4.2 Relative Power of CP+∀red with Respect to Other QBF Proof Systems

In this section we relate the power of CP+∀red with other well known QBF proof systems.

### 4.2.1 CP+∀red is Exponentially Stronger than Q-Res and QU-Res

By Lemma 4.5, CP+∀red p-simulates QU-Res (and hence Q-Res). Now we show that in fact CP+∀red is exponentially stronger than both these systems. In other words;

**Theorem 4.8.** *Q-Res and QU-Res cannot simulate CP+∀red.*

*Proof.* From classical proof complexity we know that false CNF formulas based on the pigeonhole principle (PHP) are easy for Cutting Planes proof system [37] but are hard for Resolution [53]. Therefore CP+$\forall$red is exponentially more powerful than any QBF proof system based on Resolution (example, Q-Res, QU-Res, etc).

Note that the separating QBFs have only existential quantification, and this is not the effect one wants to study in QBF proof complexity (also see [33] for a discussion). We now consider an explicit family of false QBFs with universal quantifiers for which CP+$\forall$red is exponentially more powerful that Q-Res. In [13] it has been shown that the false QBFs KBKF($t$), introduced in [60], are hard for Q-Res. However, they are known to have a polynomial-size proofs in QU-Res (see Proposition 2.10), and by Lemma 4.5 in CP+$\forall$red as well. Therefore Q-Res cannot simulate CP+$\forall$red.

Now we consider yet another family of false QBFs from [43], which we denote as Q-PHP$_n$. The formula is based on the pigeonhole principle, and is shown in [43] to be hard for Q-Res. We observe that it is also hard for QU-Res, but easy for CP+$\forall$red, hence showing that QU-Res cannot simulate CP+$\forall$red. The formula Q-PHP$_n$ is define as follows: let CPHP$_n^{X_n}$ be the false CNF formula encoding pigeon hole principle on $n+1$ pigeon and $n$ holes, and over the variables in $X_n = \{x_1, \ldots, x_n\}$, as defined in Section 2.3.2. That is,

$$\mathrm{CPHP}_n^{X_n} = \left(\bigwedge_{i=1}^{n+1}(\bigvee_{j=1}^{n} x_{i,j})\right) \wedge \left(\bigwedge_{j=1}^{n} \bigwedge_{1\leq i_1<i_2\leq n+1} (\neg x_{i_1,j} \vee \neg x_{i_2,j})\right)$$

Now define

$$\mathrm{DPHP}_n^{X_n} = \neg\mathrm{CPHP}_n^{X_n}$$

Clearly DPHP$_n^{X_n}$ $\in$ TAUT and is in DNF if the negation sign is propagated to the leaves in the formula tree. Consider the following formula:

$$\exists X_n \forall Y_n . \mathrm{DPHP}_n^{Y_n} \wedge \mathrm{CPHP}_n^{X_n}$$

with $Y_n \cap X_n = \emptyset$. This is a false QBF because $\mathrm{CPHP}_n^{X_n}$ is unsatisfiable. However the matrix of the formula is not in CNF. We define $\text{Q-PHP}_n$ to be the equivalent of the above formula where the matrix is in CNF form. In [43], the DNF formula $\mathrm{DPHP}_n^{Y_n}$ is encoded into an equivalent CNF formula $\mathrm{TPHP}_n^{Y_n,Z}$ using additional variables $Z$, disjoint from $X_n$ and $Y_n$. To be precise,

$$\text{Q-PHP}_n = \exists X_n \forall Y_n \exists Z. \mathrm{TPHP}_n^{Y_n,Z} \wedge \mathrm{CPHP}_n^{X_n}$$

with $\mathrm{DPHP}_n^{Y_n} \equiv \exists Z. \mathrm{TPHP}_n^{Y_n,Z}$ and $\mathrm{TPHP}_n^{Y_n,Z}$ is in CNF (for detailed encoding, refer to [43], also see [44, Section 6]).

**$\text{Q-PHP}_n$ is hard for Q-Res and QU-Res [43], but easy for CP+$\forall$red**: In QU-Res no resolution step is possible between the clauses from $\mathrm{TPHP}_n^{Y_n,Z}$ and $\mathrm{CPHP}_n^{X_n}$, as the variable sets are disjoint. Also the refutation is possible only from the clauses of $\mathrm{CPHP}_n^{X_n}$, as $\forall Y_n \exists Z_n \mathrm{TPHP}_n^{Y_n,Z}$ is true. Since all the variables in $X_n$ are existential, the claim follows directly from the hardness result of the pigeon hole principle for Resolution [53], and the fact that the pigeon hole principle is easy for the Cutting-plane proof system [37, Proposition 7]. □

In fact, recently it has been shown in [44, Corollary 3 and Proposition 6], that $\text{Q-PHP}_n$ needs exponential sized proofs in all QBF Resolution calculi, whether CDCL-based or expansion-based.

### 4.2.2 CP+$\forall$red and $\forall$Exp+Res are Incomparable Unless P/poly = TC$^0$

Theorem 4.8 compares CP+$\forall$red to CDCL-based QBF Resolution proof systems. Now we relate CP+$\forall$red with the most basic expansion-based proof system $\forall$Exp+Res.

One direction is unconditional.

**Proposition 4.9.** *$\forall$Exp+Res cannot simulate CP+$\forall$red.*

*Proof.* In [56], Janota and Marques-Silva show that there exists a family of false QBFs ($\phi_n$, see Section 2.4.1) which are hard for $\forall$Exp+Res but easy to refute in Q-Res. As CP+$\forall$red p-simulates Q-Res (Lemma 4.5), we conclude that $\forall$Exp+Res cannot simulate CP+$\forall$red. □

By applying [44, Proposition 6], which shows that the Q-PHP$_n$ formula from [43] requires exponential sized proofs in IRM-calc, and using the fact that Q-PHP$_n$ has a short CP+$\forall$red proof, one can strengthen Proposition 4.9 to the following:

**Proposition 4.10.** *IRM-calc cannot simulate CP+$\forall$red.*

Next we show that if $\mathsf{P/poly} \not\subseteq \mathsf{TC}^0$, then CP+$\forall$red cannot simulate $\forall$Exp+Res. We follow the method first introduced in [13, Section 4], which we briefly describe below (also in Section 2.5).

For any Boolean function family $f_n$ computed by a circuit family $C_n$ of size $l(n)$, consider the family of sentences expressing the following: $\exists x_1 \cdots x_n \forall z. f(\vec{x}) \neq z$. We want to state this sentence as a QBF. To express $f(\vec{x})$, we use the circuit $C = C_n$. Let this circuit be of size $l = l(n)$. We use Tseitin transformation (Section 2.1). Associate a variable $t_i$ with each gate of $C$, and let $t_l$ be the variable associated with the output gate. Now the false sentence can be expressed as follows:

$$Q\text{-}f_n \equiv \exists x_1 \cdots x_n \forall z \exists t_1 \cdots t_l.(t_l \neq z) \wedge \bigwedge_{i=1}^{l} (t_i \text{ is consistent with the inputs to gate } i).$$

The inner formula can be written as an $O(l)$-sized CNF (see Section 2.5).

We now show that if $f_n \in \mathsf{P/poly} \setminus \mathsf{TC}^0$, then $Q\text{-}f_n$ cannot be refuted in CP+$\forall$red in size polynomial in $n$. (To be very precise, $L$ is a language in $\mathsf{P/poly} \setminus \mathsf{TC}^0$, and $f_n$ is the characteristic function of its $n$th slice.) We prove the contrapositive:

**Lemma 4.11.** *For $f_n \in \mathsf{P/poly}$, if $Q$-$f_n$ has a polynomial-size $\mathsf{CP}+\forall\mathsf{red}$ proof, then $f_n \in \mathsf{TC}^0$.*

To prove Lemma 4.11, we need the following definition which will also be used later.

**Definition 4.12.** *Let $\mathcal{Q}.\ \phi$ be a false QBF, encoded as quantified inequalities $\mathcal{Q}.\ F$ as per Definition 4.2. Let $\pi = \mathcal{Q}.\ [I_1, \ldots, I_l]$ be any $\mathsf{CP}+\forall\mathsf{red}$ proof of $\mathcal{Q}.\ \phi$ and $\mathcal{Q}.\ F$. Define $\pi_l = \emptyset$, and for $0 \le j < l$ define $\pi_j = \mathcal{Q}.\ [I_{j+1}, \ldots, I_l]$. Further, define $F_0 = F$, and for $j > 0$, $F_j = F \cup \{I_1, \ldots, I_j\}$ ( treat $F_j$ as sequences of inequalities).*

*Proof of Lemma 4.11.* Let $\mathcal{Q}.\ F$ be the encoding of $Q$-$f_n$ as inequalities, as per Definition 4.2. Recall that $F$ includes the Boolean axioms. Consider any $\mathsf{CP}+\forall\mathsf{red}$ refutation proof $\pi = \mathcal{Q}.\ [I_1, \ldots, I_l]$ of $\mathcal{Q}.\ F$. Let the size of the proof be $m$. By assumption, $l$ and even $m$ are polynomially bounded in $n$.

By downward induction on $j$, from $\pi_j$ we show how to compute, in $\mathsf{TC}^0$, a Boolean function $\sigma^j(\vec{x})$ such that for every assignment $\vec{a}$ to the $\vec{x}$ variables, if $z$ is set to the value $\sigma^j(\vec{a})$, then the statement $\exists\vec{t}.\ F_j \mid_{\vec{x}\leftarrow\vec{a}, z\leftarrow\sigma^j(\vec{a})}$ is false. (In other words, $\sigma^j(\vec{x})$ is a winning strategy for the universal player in the 2-player game played on $\mathcal{Q}.\ F_j$.) Observe that the only such choice for $\sigma^0(\vec{x})$ is $f(\vec{x})$, since only by setting $z$ to $f(\vec{x})$ does $\phi \mid_{z=f(\vec{x})}$ become unsatisfiable. Instead of giving the $\mathsf{TC}^0$ circuits directly, we provide polynomial-size $\mathsf{TC}^0$-decision lists. To be precise we show the following:

**Claim 4.13.** *For every $j \in [l]$, from $\pi_j$, one can extract a winning strategy for the universal player $\sigma^j(\vec{x})$ in the two player game played on $\mathcal{Q}.\ F_j$, such that $\sigma^j(\vec{x})$ can be computed by a $\mathsf{TC}^0$-decision lists of length $O(l-j)$.*

As already mentioned, we prove Claim 4.13 by downward induction on $j$. Observe that the initial axioms $F$ are already included on $F_j$, therefore we can avoid the axiom download steps of the $\mathsf{CP}+\forall\mathsf{red}$ proofs.

**Base case:** When $j = l$, define $\sigma^l(\vec{x}) \equiv 0$. Indeed $\sigma^l(\vec{x})$ can take any Boolean value as $F_l$ contains $I_l$ which is the contradiction $0 \geq 1$.

**Induction hypothesis:** Assume that the Claim 4.13 is true at the $j^{th}$ step.

**Induction step:** We define $\sigma^{j-1}(\vec{x})$ from $\sigma^j(\vec{x})$. Thus for every assignment $\vec{a}$ to $\vec{x}$ and $\vec{b}$ to $\vec{t}$, if $z$ is assigned $\sigma^j(\vec{a})$, then some inequality in $F_j$ is not satisfied.

1. If the inequality $I_j$ is derived using the addition, multiplication or division rule, then define $\sigma^{j-1}(\vec{x}) \equiv \sigma^j(\vec{x})$. Observe that if an assignment $(\vec{a}, \sigma^j(\vec{a}), \vec{b})$ does not falsify $I_j$, then it must falsify an $I_k \in F_j$ with $k < j$, that is, an $I_k \in F_{j-1}$. Otherwise, since it falsifies $I_j$ and since the inference rules are sound, it also falsifies at least one of the hypotheses $I_k$, $k < j$.

2. If $I_j$ is derived using a $\forall$-reduction step (see Definition 4.1), then $I_j = I_k|_{z=b_j}$ for some $k < j$ (here $b_j \in \{0, 1\}$, depending on the coefficient of the universal variable $z$ in $I_k$), and in $I_k$ all the $\vec{t}$ variables have coefficient 0. So $I_j$ is an inequality involving only the $\vec{x}$ variables. We define $\sigma^{j-1}(\vec{x})$ as follows: If $I_k|_{z=b_j}(\vec{a})$ is false, then $\sigma^{j-1}(\vec{a}) = b_j$, else $\sigma^{j-1}(\vec{a}) = \sigma^j(\vec{a})$. Using the inductive hypothesis, we see that any assignment $\alpha = (\vec{a}, \sigma^{j-1}(\vec{a}), \vec{b})$ falsifies some inequality in $F_{j-1}$.

   The decision list $D_{j-1}(\vec{x})$ for $\sigma^{j-1}(\vec{x})$ is constructed as follows: If $\neg(I_k|_{z=b_j}(\vec{x}))$ then $D_{j-1}(\vec{x}) = b_j$ else $D_{j-1}(\vec{x}) = D_j(\vec{x})$. Observe that $D_{j-1}(\vec{x})$ has just one more condition than $D_j(\vec{x})$. By assumption, the bit-size of $I_k$ is polynomially bounded in $n$, and hence one can check the **if** condition in $\mathsf{TC}^0$.

The decision list $D_0(\vec{x})$ has length $O(l)$ and each condition is checkable by a constant-depth threshold circuit of size polynomial in $m$. Hence $\sigma^0(\vec{x}) = f(\vec{x})$ can be computed in $\mathsf{TC}^0$. □

On the other hand, from [13, Proposition 28], we know that the formula $Q$-$f_n$ can be refuted in $\forall$Exp+Res in $O(n+l)$ steps. This, along with Lemma 4.11, yields the following desired separation:

**Theorem 4.14.** *If* $\mathsf{P/poly} \not\subseteq \mathsf{TC}^0$ *then CP+$\forall$red cannot simulate $\forall$Exp+Res.*

Note that the function $\sigma^0(\vec{x})$ in the proof of Lemma 4.11 is actually a strategy extraction for (the only) universal variable $z$ in the formula $Q$-$f_n$. Thus the obvious next question is whether we can lift this technique of strategy extraction for any false QBF $\mathcal{F}$. In Section 4.3 we answer this question positively.

### 4.2.3 Frege+$\forall$red p-simulates CP+$\forall$red

In this section we show that the Frege+$\forall$red proof system defined in [11] p-simulates the CP+$\forall$red proof system. That is:

**Theorem 4.15.** *Frege+$\forall$red p-simulates CP+$\forall$red.*

In classical (propositional) proof systems, Cook, Coullard and Turán [37] first showed that EF p-simulates Cutting Planes. Then Goerdt [51] showed that even Frege p-simulates Cutting Planes. Here we show that the same simulation goes through with minor modifications. We use the techniques from [30], [37], and [51] to prove Theorem 4.15.

*Proof of Theorem 4.15.* Let $\mathcal{F}$ be a false formula $\mathcal{F} = \mathcal{Q}x_0 \cdots \mathcal{Q}x_{N-1}.\ [C_1 \wedge \cdots \wedge C_p]$, and let $\varphi$ denote its standard encoding as described in Definition 4.2. Fix any CP+$\forall$red proof $\pi = \mathcal{Q}x_0 \cdots \mathcal{Q}x_{N-1}.\ [I_1, I_2, \ldots, I_m]$ of $\varphi$. By Lemma 4.6, we can assume that $\pi$ is in normal form. We need to represent each inequality $I$ as a propositional formula $\mathrm{Rep}(I)$, such that on each assignment $\alpha$ to the Boolean variables, $\mathrm{Rep}(I)(\alpha)$ is 1 if and only if $I|_\alpha$ is 1. We do this almost exactly as in [51].

We know that integer arithmetic is in $\mathsf{NC}^1$. Thus, for a string of $(n+1)L$ Boolean variables $\tilde{y}$ representing the bits of $n+1$ signed integers $a_1, a_2, \ldots, a_n, b$ with bit length $L$ each, and $n$ Boolean variables $x_1, \ldots, x_n$, there is a formula $F(\tilde{y}, \vec{x})$ of size polynomial in $n + L$ (and depth logarithmic in $nL$) with the following properties:

- For every assignments $\beta$ to the $\tilde{y}$ variables, $F(\beta, \vec{x})$ represents the inequality $\sum_i a_i x_i \geq b$.
- For every assignments $\alpha$ to the $\vec{x}$ variables, we have $F(\beta, \alpha)$ is true iff $\sum_i a_i \alpha_i \geq b$ is true.

Now to represent a specific inequality $I : \sum_i a_i x_i \geq b$, we append to the leaves of $F$ labeled from $\tilde{y}$ subformulas of the form $x \vee \bar{x}$ or $x \wedge \bar{x}$ depending on the bits of the $a_i$'s and $b$. The resulting formula has the variables $x_1, \ldots, x_n$ and is the representation $\text{Rep}(I)$.

Our simulating Frege+∀red proof will have the structure

$$\pi_1, \text{Rep}(I_1), \pi_2, \text{Rep}(I_2), \ldots, \pi_m, \text{Rep}(I_m), \pi_{m+1}, \text{ false}$$

where each $\pi_i$ is a sequence of formulas. That is, the simulating Frege+∀red proof is a sequence of formulas containing the subsequence

$$\text{Rep}(I_1), \text{Rep}(I_2), \ldots, \text{Rep}(I_m), \text{ false}$$

For each axiom clause $C$, we need to derive the formula $\text{Rep}(R(C))$ by a short (polynomial in $n$) Frege+∀red proof. Furthermore, inside $\text{Rep}(R(C))$, there will be explicit sub-formulas representing the bits of each coefficient, $a_{ij}$ and $b_j$ for $i \in [n]$, $j \in [L]$. (To handle carry overflows, we pad each coefficient with 0s to length $\theta(L)$ as in [51].) There will also be explicit sub-formulas for each $a_{ij} \wedge x_i$.

We also need to derive each $\text{Rep}(I_t)$ from $\text{Rep}(I_j)$, $j < t$, via short (polynomial in the size of proof $\pi$) Frege+∀red proofs.

The addition rule, multiplication rule, and the division rule can be simulated as in the classical case [51]: since integer arithmetic is in $\mathsf{NC}^1$, we have small formulas $G$ expressing the coefficients of the resulting inequality $I$ from the used inequalities $I'$ and $I''$. A Frege style proof can describe how values from the subformulas in $\mathrm{Rep}(I')$ and $\mathrm{Rep}(I'')$ propagate through $G$ to bits equivalent to the corresponding input bits of $\mathrm{Rep}(I)$.

Now we show the $\forall$-red step simulation.

Suppose the inequality $I_k$ is obtained from $I_j$ for some $j < k$ by applying the $\forall$-red rule, reducing universal variable $u$. Clearly, $u$ is the rightmost variable in $I_j$ with nonzero coefficient $h_u$. Inductively, we have already derived $Rep(I_j)$. Let $b_u = 0$ if $h_u > 0$, otherwise $b_u = 1$. We need to instantiate $u$ in $\mathrm{Rep}(I_j)$ with $b_u$. But $u$ is not the rightmost variable in $\mathrm{Rep}(I_j)$. However, for each variable $v$ to the right of $u$, we know that the coefficient $a_v$ of $v$ in $I_j$ is 0, and hence the sub-formulas evaluating to the bits $a_{vj}$, as well as the sub-formulas evaluating $a_{vj} \wedge v$, are all 0. In $\mathsf{Frege}+\forall\mathsf{red}$, we can transform the pair of sub-formulas, $a_{vj} \wedge v$, and $a_{vj} \equiv 0$, to the subformula $a_{vj} \wedge 0$, and thus eliminate $v$ (note that $v$ does not figure anywhere else in the formula). Once this is done for all variables right of $u$, we have the formula $R$ in which the $\forall$-reduction step is valid in $\mathsf{Frege}+\forall\mathsf{red}$. Performing this reduction gives the formula $R' = R \mid_{u=b_u}$. Now, a short Frege proof can allow us to derive $\mathrm{Rep}(I_j \mid_{u=b_u}) = \mathrm{Rep}(I_k)$. To see why such a proof exists, consider the case $b_u = 0$. Inside $R'$ we have subformulas for the bits $h_{uj}$ of the coefficient $h_u$ of $u$, and bits for $h_{uj} \wedge u$, and at $u$ we have attached a subformula evaluating to 0. What we want is subformulas where $u$ is still free, but the bits of the new coefficient of $u$ are all 0. That is, from $h_{uj} \wedge u$ and $u \equiv 0$, we want to derive $0 \wedge u$ (the reverse of what we did before the reduction for later variables $v$). This is easy in $\mathsf{Frege}+\forall\mathsf{red}$. The case when $b_u = 1$ is similar, with the added task of subtracting $h_u$ from the right-hand-side. This too can be tracked using an $\mathsf{NC}^1$ formula for subtraction. □

Since Frege is exponentially more powerful than Cutting Planes over propositional formulas (as witnessed by the clique-colour formulas [67]), the converse simulation fails, and CP+∀red and Frege+∀red are exponentially separated.

We now obtain an exponential separation between CP+∀red and Frege+∀red for QBF formulas with universal quantifiers as well (Corollary 4.16). We construct a family of false QBFs $\Phi_{n,k}$ which encode that a graph on $n$ vertices both has and does not have a $k$-clique, and are of size polynomial in $n$. We call them the clique-co-clique formulas: fix positive integers $n$ (indicating the number of vertices of the graph) and $k \leq n$ (indicating the size of the clique queried) and let $\vec{p}$ be the set of variables $\{p_{uv} \mid 1 \leq u < v \leq n\}$. An assignment to $\vec{p}$ picks a set of edges, and thus an $n$-vertex graph. Let $\vec{q}$ be the set of variables $\{q_{iu} \mid i \in [k], u \in [n]\}$. We use the following clauses.

$$\begin{aligned}
C_i &= q_{i1} \vee \cdots \vee q_{in} && \text{for } i \in [k] \\
D_{i,j,u} &= \neg q_{iu} \vee \neg q_{ju} && \text{for } i,j \in [k], i<j \text{ and } u \in [n] \\
E_{i,u,v} &= \neg q_{iu} \vee \neg q_{iv} && \text{for } i \in [k] \text{ and } u,v \in [n], u<v \\
F_{i,j,u,v} &= \neg q_{iu} \vee \neg q_{jv} \vee p_{uv} && \text{for } i,j \in [k], i<j \ \text{ and } \ u \neq v \in [n].
\end{aligned}$$

We can now express $\textsc{Clique}(n,k)$ as a polynomial-size QBF $\exists \vec{q}. A_{n,k}(\vec{p},\vec{q})$, where

$$A_{n,k}(\vec{p},\vec{q}) = \bigwedge_{i\in[k]} C_i \wedge \bigwedge_{i,j\in[k],i<j,u\in[n]} D_{i,j,u} \wedge \bigwedge_{i\in[k],u<v} E_{i,u,v} \wedge \bigwedge_{i,j\in[k],i<j,u\neq v} F_{i,j,u,v}.$$

Here the edge variables $\vec{p}$ appear only positively in $A_{n,k}(\vec{p},\vec{q})$.

Likewise co-$\textsc{Clique}(n,k)$ can be written as a QBF $\forall \vec{r} \exists \vec{t}. B_{n,k}(\vec{p},\vec{r},\vec{t})$ of polynomial size. We describe here one particular encoding. This encoding is convenient for us because it allows us to obtain a short Frege+∀red proof (Theorem 4.17 below). For $\vec{r}$, we have a variable $r_{iu}$ for every variable $q_{iu}$ and we let the set of variables of $\vec{t}$ be $\{t_K \mid K \in A_{n,k}\} \cup \{t\}$. For each clause $K$ in $A_{n,k}(\vec{p},\vec{q})$, we include an equivalence

$t_K \leftrightarrow K[r_{iu}/q_{iu}]$ in $B_{n,k}(\vec{p}, \vec{r}, \vec{t})$, which we represent as a set of clauses. We also introduce clauses for $t \leftrightarrow \bigwedge_{K \in A_{n,k}} t_K$, i.e., $t$ indicates whether the $\vec{r}$ variables encode a clique. Because we want to represent the co-clique formula we also include $\neg t$ in $B_{n,k}(\vec{p}, \vec{r}, \vec{t})$, which yields the CNF formula co-$\textsc{Clique}(n, k) = \forall \vec{r} \exists \vec{t}. B_{n,k}(\vec{p}, \vec{r}, \vec{t})$.

The clique-co-clique formulas $\Phi_{n,k}$ are $\exists \vec{p} \exists \vec{q} \forall \vec{r} \exists \vec{t}. A_{n,k}(\vec{p}, \vec{q}) \wedge B_{n,k}(\vec{p}, \vec{r}, \vec{t})$. These formulas encode the obviously false statement that a given graph on $n$ vertices both has and does not have a $k$-clique, and are of size polynomial in $n$ as promised.

In Chapter 5, we will show via feasible interpolation technique, that the formula $\Phi_{n,k}$ (for some $k$) needs exponential many steps to refute in $\mathsf{CP}+\forall\mathsf{red}$ (Corollary 5.12).

We now show that the formula $\Phi_{n,k}$ are in fact easy for $\mathsf{Frege}+\forall\mathsf{red}$ (Theorem 4.17) and thereby obtain the following Corollary:

**Corollary 4.16.** *$\mathsf{CP}+\forall\mathsf{red}$ does not simulate $\mathsf{Frege}+\forall\mathsf{red}$.*

**Theorem 4.17.** *The clique-co-clique formulas $\Phi_{n,k}$ have short proofs in $\mathsf{Frege}+\forall\mathsf{red}$.*

*Proof.* We use a result from [20, Theorem 8.1] which shows that a $\mathsf{Frege}+\forall\mathsf{red}$ superpolynomial lower bound must either come from a circuit lower bound or a classical $\mathsf{Frege}$ lower bound. More precisely, if false QBFs $\Phi_n$ do not admit polynomial-size $\mathsf{Frege}+\forall\mathsf{red}$ proofs, then either the universal player does not have $\mathsf{NC}^1$ winning strategies for the universal variables, or if small $\mathsf{NC}^1$ winning strategies exist, then the propositional formulas obtained by substituting the $\mathsf{NC}^1$ circuits for universal variables in $\Phi_n$ are hard for classical $\mathsf{Frege}$.

In the case of the clique co-clique formulas $\Phi_{n,k}$ there exist short winning strategies for the universal player, namely $\vec{r} = \vec{q}$. To see this, we just need to consider the case where the existential player chooses a graph $\vec{p}$ that contains a $k$-clique exhibited in the $\vec{q}$-variables, because otherwise the universal player immediately wins on

$A_{n,k}(\vec{p},\vec{q})$. In this case, choosing $\vec{r}=\vec{q}$ ensures that $B_{n,k}(\vec{p},\vec{r},\vec{t})$ fails as $\vec{r}$ indeed is a $k$-clique.

Substituting these winning strategies into $\Phi_{n,k}$, we obtain the false propositional formulas $A_{n,k}(\vec{p},\vec{q}) \wedge B_{n,k}(\vec{p},\vec{q},\vec{t})$, which admit short Frege refutations.

Using this intuition we can refute $\Phi_{n,k}$ in Frege+∀red with short proofs. For this we first derive the tautology $\neg(A_{n,k}(\vec{p},\vec{q}) \wedge B_{n,k}(\vec{p},\vec{q},\vec{t}))$ by demonstrating a way to find a contradiction in $A_{n,k}(\vec{p},\vec{q}) \wedge B_{n,k}(\vec{p},\vec{q},\vec{t})$. To do this we observe that for any clause $K \in A_{n,k}(\vec{p},\vec{q})$, we have the equivalences $(t_K \leftrightarrow K) \in B_n(\vec{p},\vec{q},\vec{t})$, so we derive all $t_K$. Then, because $(t \leftrightarrow \bigwedge_{K\in A_{n,k}} t_K) \in B_{n,k}(\vec{p},\vec{q},\vec{t})$, we obtain $t$. This means that with $\neg t \in B_{n,k}(\vec{p},\vec{q},\vec{t})$ we have a contradiction, thus proving the negation $\neg(A_{n,k}(\vec{p},\vec{q}) \wedge B_{n,k}(\vec{p},\vec{q},\vec{t}))$.

Moving forward to the next step, we derive in (polynomially many) Frege steps the implication $\bigwedge_{i\in[k],j\in[\binom{n}{2}]}(q_{i,j} \leftrightarrow r_{i,j}) \rightarrow \neg(A_{n,k}(\vec{p},\vec{q}) \wedge B_n(\vec{p},\vec{r},\vec{t}))$, from which together with the axiom $A_n(\vec{p},\vec{q}) \wedge B_n(\vec{p},\vec{r},\vec{t})$ we derive the disjunction $\bigvee_{i\in[k],j\in[\binom{n}{2}]}(r_{i,j} \neq q_{i,j})$.

Now we perform ∀-reduction, starting with the rightmost universal variable $r_{i_1,j_1}$ and instantiating it with both 0 and 1. Thus we obtain two lines:

$$(0 \neq q_{i_1,j_1}) \vee \bigvee_{i\in[k],i\neq i_1,j\in[\binom{n}{2}],j\neq j_1} (r_{i,j} \neq q_{i,j})$$

$$(1 \neq q_{i_1,j_1}) \vee \bigvee_{i\in[k],i\neq i_1,j\in[\binom{n}{2}],j\neq j_1} (r_{i,j} \neq q_{i,j})$$

We then use the tautology $(q_{i_1,j_1} \leftrightarrow 0) \vee (q_{i_1,j_1} \leftrightarrow 1)$ and the two instantiations to remove the disjunct $(r_{i_1,j_1} \neq q_{i_1,j_1})$ from the disjunction. Continuing this iteratively, we remove all disjuncts and are left with the empty disjunct, hence refuting $\Phi_{n,k}$ in polynomial size. □

Note that if we changed the quantification and used formula $\exists\vec{p}\forall\vec{r}\exists\vec{t}\exists\vec{q}.A_{n,k}(\vec{p},\vec{q}) \wedge$

$B_{n,k}(\vec{p}, \vec{r}, \vec{t})$ we would still be describing the same contradiction between clique and co-clique. However the above argument would not work for finding short Frege+$\forall$red proofs. This is because the strategies of the universal player cannot refer to the choices of $\vec{q}$ (since the universal player is restricted to using variables that appear left of the variable in question) but instead has to describe a $k$-clique expressed as the $\vec{r}$ variables as soon as the existential player plays in the graph variables. However the strategies that determine these clique are restricted to the $\vec{p}$ graph variable. Since cliques can be checked easily when found, this means that the universal strategies compute the NP-complete CLIQUE$(n, k)$ problem. So strategies are conjectured to be hard unless $\mathsf{NP} \subseteq \mathsf{NC}^1$. Because of the strategy extraction theorem from [11] $\mathsf{NP} \subseteq \mathsf{NC}^1$ will be a necessary condition for these modified formulas to have short proofs in Frege+$\forall$red.

## 4.3 Strategy extraction for CP+$\forall$red

Recall that a QBF $\mathcal{Q}_1 x_1 \cdots \mathcal{Q}_k x_k \,.\, \phi$ can be seen as a game between two players: *universal* ($\forall$) and *existential* ($\exists$). Given a universal variable $u$ with index $i$, a *strategy for $u$* is a function from all variables of index $< i$ to $\{0, 1\}$. A QBF is false if and only if there exists a *winning strategy* for the universal player.

Recall from Section 2.5 that, a QBF proof system has the strategy extraction property for a particular class of circuits $\mathcal{C}$ whenever we can efficiently extract, from every refutation $\pi$ of a QBF $\phi$, universal player strategies in circuit class $\mathcal{C}$ for all universal variables.

In the proof of Lemma 4.11 we saw how to extract, from a refutation of $Q$-$f_n$ in CP+$\forall$red, a winning strategy for the sole universal variable $z$. We now consider the more general version; for formulas with multiple universal variables, quantified anywhere in the prefix, we show how to extract winning strategies for the universal player from a refutation in CP+∀red.

**Theorem 4.18** (Strategy Extraction Theorem)**.** *Given a false QBF $\mathcal{F} = \mathcal{Q}.\ \phi$, with $n$ variables, and a CP+∀red refutation $\pi$ of $\mathcal{F}$ (to be precise,the refutation of the standard encoding $\varphi = \mathcal{Q}.\ F$ of $\mathcal{F}$, see Definition 4.2) of size $m$, it is possible to extract from $\pi$ a winning strategy $\sigma_u$ for each universal variable $u \in \varphi$, such that each $\sigma_u$ can be computed by Boolean circuits of $(m+n)^{O(1)}$ size, constant depth, with unbounded fanin AND, OR, NOT gates as well as threshold gates.*

*In particular, if $\varphi$ can be refuted in CP+∀red in $n^{O(1)}$ size, then the winning strategies can be computed in $\mathsf{TC}^0$.*

*Proof.* We adapt the technique from [11]. Let $\pi = \mathcal{Q}.\ [I_1, \ldots, I_l]$ be a normal-form CP+∀red proof of the standard encoding $\mathcal{Q}.\ F$ of $\mathcal{Q}.\ \phi$, of length $l$ and size $m \geq l$. For $j \in \{0, 1, \ldots, l\}$, $\pi_j$ and $F_j$ are as defined in Definition 4.12; $\pi_j = \mathcal{Q}.\ [I_{j+1}, \ldots, I_l]$ and $F_j = F \cup \{I_1, \ldots, I_j\}$ (note that $\pi_l = \emptyset$ and $F_0 = F$). By downward induction on $j$, from $\pi_j$ we show how to compute, for each universal variable $u$, a Boolean function $\sigma_u^j$ that maps each assignment to the variables quantified before $u$ to a bit $\{0, 1\}$. These functions satisfy the property that in a 2-player game played on the formula $\mathcal{Q}.\ F_j$, if the universal player chooses values for each universal variable $u$ according to $\sigma_u^j$, then finally some inequality in $F_j$ is falsified. We describe the functions $\sigma_u^j$ by decision lists of size $O(l - j)$, where each condition is checkable by a constant depth threshold circuit of size polynomial in $m$. Again we can skip the axiom download steps.

The strategy is as follows: $\sigma_u^l = 0$ for all $u$. For $j \leq l$, if $I_j$ is obtained by a classical rule, then $\sigma_u^{j-1} = \sigma_u^j$ for every universal variable $u$. If $I_j$ is derived using a ∀-red rule; that is $I_j = I_k|_{u=b_j}$ for some $k < j$, then for all $u' \neq u$, $\sigma_{u'}^{j-1} = \sigma_{u'}^j$. For $u$, if $I_k|_{u=b_j}(\vec{a}) = 0$, then $\sigma_u^{j-1}(\vec{a}) = b_j$, else $\sigma_u^{j-1}(\vec{a}) = \sigma_u^j(\vec{a})$. (The value $I_k|_{u=b_j}(\vec{a})$ can be determined since variables to the right of $u$ have zero coefficient in $I_k$.) It is easy to see that these functions so defined have the desired property. □

Lemma 4.11 yields a conditional lower bound for $\mathsf{CP}+\forall\mathsf{red}$: If $\mathsf{P/poly} \not\subseteq \mathsf{TC}^0$, then there is a family of false QBFs with no polynomial size proof in $\mathsf{CP}+\forall\mathsf{red}$. Using Theorem 4.18, we can obtain a similar lower bound from a weaker assumption; namely,

**Corollary 4.19.** *If $\mathsf{PSPACE/poly} \not\subseteq \mathsf{TC}^0$, then there exists a family of false QBFs $Q_{qbf}$-$f_n$ that requires super-polynomial size proofs in $\mathsf{CP}+\forall\mathsf{red}$.*

*Proof.* Let $f_n \in \mathsf{PSPACE/poly} \setminus \mathsf{TC}^0$. Consider the false sentence based on $f_n$:

$$\exists x_1 \dots x_n \forall z. \big[f(\vec{x}) \neq z\big].$$

Since $f_n$ is in $\mathsf{PSPACE/poly}$ and QBF is $\mathsf{PSPACE}$-complete, the value of $f_n$ can be compactly expressed by a QBF. That is, $f_n(\vec{x}) \equiv \mathcal{Q}_1 y_1 \dots \mathcal{Q}_r y_r.\psi_n(\vec{x}, \vec{y})$ where $r$ is polynomial in $n$ and $\psi_n(\vec{x}, \vec{y})$ is in $\mathsf{P/poly}$. Thus we have the false sentence

$$\exists x_1 \dots x_n \forall z. \big[\overbrace{(\mathcal{Q}_1 y_1 \dots \mathcal{Q}_r y_r.\psi_n(\vec{x}, \vec{y}))}^{f_n(\vec{x})} \leftrightarrow \neg z\big].$$

We now choose circuits $C_n$ computing $\psi_n$ and use additional variables $\vec{s}$ and $\vec{t}$ to represent the gate values in the $\mathsf{P/poly}$ circuits $C_n$ and $\neg C_n$, respectively. We obtain the QBF

$$\exists x_1 \dots x_n \forall z \mathcal{Q}_1 y_1 \dots \mathcal{Q}_r y_r \bar{\mathcal{Q}}_1 w_1 \dots \bar{\mathcal{Q}}_r w_r \exists \vec{s}, \vec{t}.\ \big[(C_n(\vec{x}, \vec{y}, \vec{s}) \vee z) \wedge \big(\neg C_n(\vec{x}, \vec{w}, \vec{t}) \vee \neg z\big)\big]$$

where $\bar{\mathcal{Q}} = \exists$ if $\mathcal{Q} = \forall$ and vice versa. We call this formula $Q_{qbf}$-$f_n$ and remark that it is a false prenex QBF with CNF matrix. ($C_n$ can be expressed as a CNF by applying Tseitin transformation to the circuit; then adding the literal $z$ to each clause expresses $C_n \vee z$. Similarly for $\neg C_n \vee \neg z$.)

As in Lemma 4.11, observe that in the two-player game on $Q_{qbf}$-$f_n$ or on its encoding as inequalities, the only winning strategy for the universal variable $z$ is the function $f_n(\vec{x})$ itself. Therefore if there exists a polynomial size CP+∀red proof for $Q_{qbf}$-$f_n$ (or equivalently for its standard encoding), then from Theorem 4.18, $f_n \in \mathsf{TC}^0$, a contradiction. □

## 4.4 Semantic cutting planes for QBFs

The classical cutting planes proof system Cutting Planes can be extended to the semantic Cutting Planes proof system by allowing the following semantic inference rule: from inequalities $I'$, $I''$, we can infer $I$ in one step if every Boolean assignment satisfying both $I'$ and $I''$ also satisfies $I$. In [47], it is shown that semantic Cutting Planes is exponentially more powerful than Cutting Planes. We now augment the system semantic Cutting Planes with the ∀-reduction rule as defined for CP+∀red, to obtain a QBF version denoted semCP+∀red. In fact, in this system we need only two rules, semantic inference and ∀-reduction, since the addition, multiplication and division rules of Cutting Planes are also semantic inferences, and the Boolean axioms can be semantically inferred from any inequality.

It is clear that semCP+∀red is sound and complete for false QBFs. However it is not possible to verify the semantic rule efficiently (unless P= NP).

As in CP+∀red, we call a semCP+∀red proof $\pi$ a normal-form proof if ∀-red is applied only to the innermost universal variable. Since one can use Boolean axioms in semCP+∀red; Lemma 4.6 is valid in semCP+∀red as well. That is one can convert any semCP+∀red proof $\pi$ into a normal form in polynomial time.

Clearly, SemCP+∀red is at least as powerful as CP+∀red. From classical proof complexity we known that semantic Cutting Planes is exponentially more powerful than Cutting Planes [47]. That is, in [47, Theorem 2], it has been shown that for every $n$, there exists a CNF formula $F_n$ which has a short semantic Cutting Planes refutation but needs $2^{n^{\Omega(1)}}$ lines to refute in Cutting Planes. Thus semCP+∀red is also exponentially more powerful that CP+∀red, as witnessed by these purely existentially quantified formulas.

In Lemma 4.11 and Theorem 4.18, we established strategy extraction from CP+∀red proofs. These results hold for semCP+∀red proofs as well; if $I_j$ is obtained by semantic inference, we do not change the strategy functions and let $\sigma_u^{j-1} = \sigma_u^j$ for every universal variable $u$. Thus all the conditional lower bounds on CP+∀red continue to hold:

**Corollary 4.20.** *1. If* P/poly $\not\subseteq$ $\mathsf{TC}^0$*, then semCP+∀red cannot p-simulate ∀Exp+Res. For any* $f_n \in$ P/poly $\setminus$ $\mathsf{TC}^0$*, the false QBF formula Q-*$f_n$ *requires super-polynomial size proofs in semCP+∀red.*

*2. If* PSPACE $\not\subseteq$ $\mathsf{TC}^0$*, then for any* $f_n \in$ PSPACE $\setminus$ $\mathsf{TC}^0$*, the false QBF* $Q_{qbf}$*-*$f_n$ *requires super-polynomial size proofs in semCP+∀red.*

As already mentioned, in Chapter 5, we establish feasible interpolation for semCP+∀red and prove an exponential lower bound for the formula $\Phi_{n,k}$ in semCP+∀red proof system (Corollary 5.14).

# Chapter 5

# Feasible Interpolation for QBF Proof Systems

Recall from Section 2.3.1, *feasible interpolation*, first introduced by Krajíček in [62], is a particular successful paradigm that transfers circuit lower bounds to proof size lower bounds. The technique has been shown to be effective for Resolution [62], Cutting Planes [67] and even strong Frege systems for modal and intuitionistic logics [54]. However, feasible interpolation fails for strong propositional systems such as Frege systems under plausible cryptographic and number-theoretic assumptions [24, 27, 63].

The following question naturally arises: does the feasible interpolation technique apply to QBF Resolution systems? In this Chapter, we answer the question positively, that is, we show that feasible interpolation applies to all CDCL-based QBF Resolution calculi. We do this by establishing the technique for the most powerful CDCL-based QBF proof system; $\mathsf{LQU}^+$-Res (Section 5.1.2). In fact, it has been shown in [14] (also see PhD dissertation of Leroy Chew [34]), that the technique works even for all expansion-based QBF Resolution calculi. We also establish the technique for our new QBF proof systems from Chapter 4; CP+∀red, and semCP+∀red

(Section 5.2 and 5.3).

As a consequence, we show that the clique-co-clique formulas $\Phi_{n,k}$ (see Section 4.2.3), are hard for the proof systems LQU$^+$-Res (Section 5.1.4), CP+∀red (Section 5.2) and semCP+∀red (Section 5.3).

We start by establishing feasible interpolation technique for the CDCL-based QBF Resolution calculi.

## 5.1 Feasible Interpolation for CDCL-based QBF Resolution Calculi

In this section we establish the feasible interpolation technique for all known CDCL-based QBF proof systems. After describing the required setting, we first revisit the feasible interpolation theorem established for Resolution by Krajíček in [62] and Pudlák in [67] in Section 5.1.1. We also show how to generalize the technique to Q-Res proof systems. Then in Section 5.1.2 we establish the feasible interpolation theorem for the most powerful CDCL-based proof system LQU$^+$-Res. We further establish monotone feasible interpolation for the system LQU$^+$-Res in Section 5.1.3, using which we show an exponential lower bounds for LQU$^+$-Res (Section 5.1.4).

### 5.1.1 The Setting

Consider a false QBF $\mathcal{F}$ of the form

$$\exists \vec{p} Q \vec{q} Q \vec{r}.\ \big[A(\vec{p},\vec{q}) \wedge B(\vec{p},\vec{r})\big],$$

where, $\vec{p}$, $\vec{q}$, and $\vec{r}$ are mutually disjoint sets of propositional variables, $A(\vec{p},\vec{q})$ is a CNF formula on variables $\vec{p}$ and $\vec{q}$, and $B(\vec{p},\vec{r})$ is a CNF formula on variables

$\vec{p}$ and $\vec{r}$. Thus $\vec{p}$ are the common variables between them. The $\vec{q}$ and $\vec{r}$ variables can be quantified arbitrarily, with any number of quantification levels. The QBF is equivalent to the following, not in prenex form

$$\exists\vec{p}\big[\mathcal{Q}\vec{q}.A(\vec{p},\vec{q}) \wedge \mathcal{Q}\vec{r}.B(\vec{p},\vec{r})\big].$$

**Definition 5.1.** *Let $\mathcal{F}$ be a false QBF of the form $\exists\vec{p}\mathcal{Q}\vec{q}\mathcal{Q}\vec{r}.\,[A(\vec{p},\vec{q}) \wedge B(\vec{p},\vec{r})]$. An* interpolation circuit *for $\mathcal{F}$ is a boolean circuit $C$ such that on every $0,1$ assignment $\vec{a}$ for $\vec{p}$ we have*

$$\begin{aligned} C(\vec{a}) = 0 &\implies \mathcal{Q}\vec{q}.A(\vec{a},\vec{q}) \text{ is false, and} \\ C(\vec{a}) = 1 &\implies \mathcal{Q}\vec{r}.B(\vec{a},\vec{r}) \text{ is false.} \end{aligned}$$

*We say that a QBF proof system $S$ has* feasible interpolation *if for any $S$-proof $\pi$ of a QBF $\mathcal{F}$ of the form above, we can extract from $\pi$ an interpolation circuit for $\mathcal{F}$ of size polynomial in the size of $\pi$.*

*We say that a QBF proof system $S$ has* monotone feasible interpolation *if the following holds: in the same setting as above, if $\vec{p}$ appears only positively in $A(\vec{p},\vec{q})$, then we can extract from $\pi$ a monotone interpolation circuit for $\mathcal{F}$.*

As our main results in this Section, we show that $\mathsf{LQU}^+$-$\mathsf{Res}$ has monotone feasible interpolation.

Before proving the interpolation theorems, we first outline the general idea for establishing feasible interpolation for any CDCL-based QBF Resolution calculus.

### Proof idea.

Fix a proof system $S \in \{$Q-Res, QU-Res, LD-Q-Res, $\mathsf{LQU}^+$-Res$\}$ and an $S$-proof $\pi$ of $\mathcal{F}$. Consider the following definition of a $\vec{q}$-clause and an $\vec{r}$-clause.

**Definition 5.2** ( [67]). *We call a clause $C$ in $\pi$ a $\vec{q}$-clause (resp. $\vec{r}$-clause), if $C$ contains only variables $\vec{p}, \vec{q}$ (resp. $\vec{p}, \vec{r}$). We also call $C$ a $\vec{q}$-clause (resp. $\vec{r}$-clause), if $C$ contains only $\vec{p}$ variables, but all its descendant clauses in the proof $\pi$ (all clauses with a directed path to $C$ in $\pi$) are $\vec{q}$ (resp. $\vec{r}$)-clauses. An initial clause $C$ containing only $\vec{p}$ variables is call a $\vec{q}$-clause (resp. $\vec{r}$-clause) if it belongs to $A(\vec{p}, \vec{q})$ (resp. $B(\vec{p}, \vec{r})$) part.*

Note that if we have not given an explicit partition of the initial clauses then we partition the clauses as follows: put the clauses containing $\vec{p}$ and $\vec{q}$ variables in $A(\vec{p}, \vec{q})$ part, clauses containing $\vec{p}$ and $\vec{r}$ variables in $B(\vec{p}, \vec{r})$ part, and we are free to put the clauses containing only $\vec{p}$ variables in either of the parts.

From $\pi$ we construct a circuit $C_\pi$ with the $\vec{p}$-variables as inputs: For each node $u$ with clause $C_u$ in the proof $\pi$, associate a gate $g_u$ (or a constant-size circuit) in the circuit $C_\pi$. We then construct, for any assignment $\vec{a}$ to the $\vec{p}$ variables, another proof-like structure $\pi'(\vec{a})$. For each node $u$ with clause $C_u$ in the proof $\pi$, associate a clause $C'_{u,\vec{a}}$ in the structure $\pi'(\vec{a})$. Finally, we obtain $\pi''(\vec{a})$ from the structure $\pi'(\vec{a})$ by instantiating $\vec{p}$ variables to the assignment $\vec{a}$ and doing some pruning, and show that $\pi''(\vec{a})$ is a valid proof in $S$. We then find that if $C_\pi(\vec{a}) = 0$, then $\pi''(\vec{a})$ uses only $\vec{q}$-clauses and thus is a refutation of $\mathcal{Q}\vec{q}.A(\vec{a}, \vec{q})$, and if $C_\pi(\vec{a}) = 1$, then $\pi''(\vec{a})$ uses only $\vec{r}$-clauses and thus is a refutation of $\mathcal{Q}\vec{r}.B(\vec{a}, \vec{r})$. Thus $C_\pi$ is the desired interpolant circuit.

To explain the idea more precisely, we need the following Definition:

**Definition 5.3.** *For clauses $C, D$ we write $C \preceq D$ if for any literal $l \in C$ we have $l \in D$ or $l^* \in D$ and for any $l^* \in C$ we have $l^* \in D$ (recall that $l^*$ is a merger literal, see the Definition of* LD-Q-Res *proof system, Figure 2.2). Note that, in proof systems, where special literals of the form $u^*$ are not present (ex.* Q-Res*), $\preceq$ is just the $\subseteq$ relation.*

More precisely the idea is to show (by induction on the height of $u$ in $\pi$) that:

1. $C'_{u,\vec{a}} \preceq C_u$.

2. $g_u(\vec{a}) = 0 \implies C''_{u,\vec{a}}$ is a $\vec{q}$-clause and can be obtained from the clauses of $A(\vec{a}, \vec{q})$ alone using the rules of $S$.

3. $g_u(\vec{a}) = 1 \implies C''_{u,\vec{a}}$ is an $\vec{r}$-clause and can be obtained from the clauses of $B(\vec{a}, \vec{r})$ alone using the rules of $S$.

From above, we have the following conclusion. Let $r$ be the root of $\pi$. Then on any assignment $\vec{a}$ to the $\vec{p}$ variables we have:

**(1)** $C'_{r,\vec{a}} \preceq C_r = \Box$, so $C'_{r,\vec{a}} = \Box$. Therefore, $C''_{r,\vec{a}} = C'_{r,\vec{a}}|_{\vec{a}} = \Box$.

**(2)** $g_r(\vec{a}) = 0 \implies \Box$ is a $\vec{q}$-clause and can be obtained from the clauses of $A(\vec{a}, \vec{q})$ alone using the rules of system $S$. Hence by soundness of $S$, $\mathcal{Q}\vec{q}.A(\vec{a}, \vec{q})$ is false.

**(3)** $g_r(\vec{a}) = 1 \implies \Box$ is an $\vec{r}$-clause and can be obtained from the clauses of $B(\vec{a}, \vec{r})$ alone using the rules of system $S$. Hence by soundness of $S$, $\mathcal{Q}\vec{q}.B(\vec{a}, \vec{r})$ is false.

Thus $g_r$, the output gate of the circuit, computes an interpolant.

### Interpolants from Resolution Proofs [62, 67]

Observe that when $\mathcal{F}$ has only existential quantification, $\pi$ is a classical resolution proof, and this is exactly the interpolant computed by Pudlák's method in [67]. For completeness we present his method from [67].

As mentioned in the proof idea, for a Resolution proof $\pi$ of $\mathcal{F}$, we first describe the circuit $C_\pi$ with input $\vec{p}$.

**Construction of the Circuit $C_\pi$.** The DAG underlying the circuit is exactly the same as the DAG underlying the proof $\pi$. For each node $u$ with clause $C_u$ in $\pi$ we associate a gate $g_u$ as follows:

$u$ **is a Leaf Node:** If $C_u \in A(\vec{p}, \vec{q})$ then $g_u$ is a constant 0 gate. If $C_u \in B(\vec{p}, \vec{r})$ then $g_u$ is a constant 1 gate.

$u$ **is an Internal Node:** We distinguish three cases.

**(1)** $u$ corresponds to a resolution step with an existential variable $x \in \vec{p}$ as pivot. Nodes $v$ and $w$ are its two children, i.e.

$$\frac{\overbrace{C_1 \vee x}^{\text{node } v} \qquad \overbrace{C_2 \vee \neg x}^{\text{node } w}}{\underbrace{C_1 \vee C_2}_{\text{node } u}}$$

In this case, put a selector gate $\mathsf{sel}(x, g_v, g_w)$ for $g_u$. Here, $\mathsf{sel}(x, a, b) = a$, when $x = 0$, and $\mathsf{sel}(x, a, b) = b$, when $x = 1$. That is, $\mathsf{sel}(x, a, b) = (\neg x \wedge a) \vee (x \wedge b)$.

**(2)** $u$ corresponds to a resolution step with $x \in \vec{q}$ as pivot. Put an OR gate for $g_u$.

**(3)** $u$ corresponds to a resolution step with $x \in \vec{r}$ as pivot. Put an AND gate for $g_u$.

This completes the description of the circuit $C_\pi$.

**Construction of $\pi'$ and $\pi''$.** Following our proof idea, we now construct a proof-like structure $\pi'(\vec{a})$. For each node $u$ in $\pi$ with clause $C_u$, we associate a clause $C'_{u,\vec{a}}$ in $\pi'(\vec{a})$.

**At Leaf Level:** Let node $u$ be a leaf in $\pi$. Then $C'_{u,\vec{a}} = C_u$; that is, we copy the clause as it is. Trivially, we have $C'_{u,\vec{a}} \subseteq C_u$ (for $\mathsf{Resolution}$, $\preceq$ is just $\subseteq$). By construction of $C_\pi$, the conditions concerning $g_u(\vec{a})$ and $C''_{u,\vec{a}}$ are satisfied.

At an internal node we distinguish three cases based on the rule that was applied.

**At an Internal Node with $\vec{p}$-resolution:** Let node $u$ in the proof $\pi$ correspond to a resolution step with pivot $x \in \vec{p}$. We have

$$\frac{C_v = \overbrace{C_1 \vee x}^{\text{node } v} \qquad \overbrace{C_2 \vee \neg x}^{\text{node } w} = C_w}{C_u = \underbrace{C_1 \vee C_2}_{\text{node } u}}.$$

In the assignment $\vec{a}$, if $x = 0$, then define $C'_{u,\vec{a}} = C'_{v,\vec{a}} \setminus \{x\}$ and if $x = 1$ then define $C'_{u,\vec{a}} = C'_{w,\vec{a}} \setminus \{\neg x\}$. By induction, we have $C'_{v,\vec{a}} \subseteq C_v$ and $C'_{w,\vec{a}} \subseteq C_w$. So, if $x = 0$, we have $C'_{u,\vec{a}} = C'_{v,\vec{a}} \setminus \{x\} \subseteq C_v \setminus \{x\} \subseteq C_u$. If $x = 1$, we have $C'_{u,\vec{a}} \subseteq C'_{w,\vec{a}} \setminus \{\neg x\} \subseteq C_w \setminus \{\neg x\} \subseteq C_u$.

In this case $g_u$ is a selector gate. If $x = 0$ in the assignment $\vec{a}$, then $g_u(\vec{a}) = g_v(\vec{a})$ and $C''_{u,\vec{a}} = C''_{v,\vec{a}}$. Since the conditions concerning $g_v(\vec{a})$ and $C''_{v,\vec{a}}$ are satisfied by induction, the conditions concerning $g_u(\vec{a})$ and $C''_{u,\vec{a}}$ are satisfied as well. Similarly, if $x = 1$, then $g_u(\vec{a}) = g_w(\vec{a})$ and $C''_{u,\vec{a}} = C''_{w,\vec{a}}$, and the statements that are inductively true at $w$ hold at $u$ as well.

**At an Internal Node with $\vec{q}$-resolution:** Let node $u$ in the proof $\pi$ correspond to a resolution step with pivot $x \in \vec{q}$. We have

$$\frac{C_v = \overbrace{C_1 \vee x}^{\text{node } v} \qquad \overbrace{C_2 \vee \neg x}^{\text{node } w} = C_w}{C_u = \underbrace{C_1 \vee C_2}_{\text{node } u}}, \quad x \in \vec{q}.$$

If $g_v(\vec{a}) = 1$ then define $C'_{u,\vec{a}} = C'_{v,\vec{a}}$. By induction, we know that $C''_{u,\vec{a}} = C''_{v,\vec{a}}$ is an $\vec{r}$-clause. Since $x$ is a $\vec{q}$-variable and is not instantiated by $\vec{a}$, it must be the case that $x \notin C'_{v,\vec{a}}$. Thus $C'_{u,\vec{a}} = C'_{v,\vec{a}} \subseteq C_v \setminus \{x\} \subseteq C_u$.

Else if $g_w(\vec{a}) = 1$, define $C'_{u,\vec{a}} = C'_{w,\vec{a}}$. By a similar analysis as above, $C'_{u,\vec{a}} = C'_{w,\vec{a}} \subseteq C_w \setminus \{\neg x\} \subseteq C_u$.

If $g_v(\vec{a}) = g_w(\vec{a}) = 0$, and if $x \notin C'_{v,\vec{a}}$, define $C'_{u,\vec{a}} = C'_{v,\vec{a}}$. Otherwise, if $\neg x \notin C'_{w,\vec{a}}$, define $C'_{u,\vec{a}} = C'_{w,\vec{a}}$. It follows from induction that $C'_{u,\vec{a}} \subseteq C_u$.

Else, define $C'_{u,\vec{a}}$ to be the resolvent of $C'_{v,\vec{a}}$ and $C'_{w,\vec{a}}$ on $x$. By induction, we know that $C'_{v,\vec{a}} \setminus \{x\} \subseteq C_1$ and $C'_{w,\vec{a}} \setminus \{\neg x\} \subseteq C_2$. Hence $C'_{u,\vec{a}} \subseteq C_1 \vee C_2 = C_u$.

We need to verify the conditions on $g_u(\vec{a})$ and $C''_{u,\vec{a}}$. The case when $g_u(\vec{a}) = 1$ is immediate, since $C''_{u,\vec{a}}$ copies a clause known by induction to be an $\vec{r}$-clause. So now consider the case when $g_u(\vec{a}) = 0$. By induction, we know that both $C''_{v,\vec{a}} = C'_{v,\vec{a}}|_{\vec{a}}$ and $C''_{w,\vec{a}} = C'_{w,\vec{a}}|_{\vec{a}}$ are $\vec{q}$-clauses and can be derived using $A(\vec{a}, \vec{q})$ alone in Resolution.

We have three cases. If $C'_{u,\vec{a}} = C'_{v,\vec{a}}$ or $C'_{u,\vec{a}} = C'_{w,\vec{a}}$, then by induction we are done. Otherwise, $C'_{u,\vec{a}}$ is obtained from $C'_{v,\vec{a}}$ and $C'_{w,\vec{a}}$ via a resolution step on pivot $x$. Since $\vec{a}$ is an assignment to the $\vec{p}$ variables and $x \notin \vec{p}$, $C''_{u,\vec{a}}$ can be derived from $C''_{v,\vec{a}}$ and $C''_{w,\vec{a}}$ via the same $\vec{q}$-resolution step.

**At an Internal Node with $\vec{r}$-resolution:** Let node $u$ in $\pi$ correspond to a resolution step with pivot $x \in \vec{r}$. This is dual to the case above.

This gives the interpolant from resolution proof and shows that Resolution admits feasible interpolation.

### Interpolants from Q-Res Proofs

Now we show how to lift Pudlák's method of finding interpolating circuit from resolution proofs to all CDCL-based QBF proofs. We first show how with some minute modifications the method can be easily extended for Q-Res. Let $\mathcal{F}$ has universal variables as well. However as already mentioned, the common $\vec{p}$ variables are purely existential and are before all other variables. Let $\pi$ be a Q-Res proof of $\mathcal{F}$. We need to extract the interpolating circuit from $\pi$. We again follow the above mentioned proof idea. The circuit $C_\pi$ is constructed as follows: if node $u$ of $\pi$ corresponds to the universal reduction step, put a no-operation gate for $g_u$ and for all other cases, define $C_\pi$ exactly as for resolution proofs. Similarly, all other cases, in the construction of $\pi'$ and $\pi''$, that is resolution steps on pivot variables $x \in \vec{p}$, or $x \in \vec{q}$, or $x \in \vec{r}$, are handled as before. Note that now $\vec{q}$ and $\vec{r}$ variables can be either existential or universal, but that does not makes any difference. We only need to handle the $\forall$-Red step.

**At an Internal Node with Universal Reduction:** Let node $u$ be an internal node in $\pi$ corresponding to a universal reduction step on some universal variable $x$. Let node $v$ be its only child. We have

$$\frac{C_v = \overbrace{D_v \vee x}^{\text{node } v}}{C_u = \underbrace{D_v}_{\text{node } u}}, \qquad x \text{ is a universal variable, if } l \in D_v \text{ is existential, } \operatorname{lv}(l) < \operatorname{lv}(x).$$

In this case, define $C'_{u,\vec{a}} = C'_{v,\vec{a}} \setminus \{x, \neg x\}$. By induction, $C'_{v,\vec{a}} \subseteq C_v = D_v \vee x$. Therefore, $C'_{u,\vec{a}} = C'_{v,\vec{a}} \setminus \{x, \neg x\} \subseteq D_v = C_u$.

If $g_u(\vec{a}) = 0$, then we know that $g_v(\vec{a}) = 0$ as $g_u(\vec{a}) = g_v(\vec{a})$. By the induction hypothesis, we know that $C''_{v,\vec{a}} = C'_{v,\vec{a}}|_{\vec{a}}$ is a $\vec{q}$-clause and can be derived using $A(\vec{a}, \vec{q})$ alone via Q-Res. Recall that $C'_{u,\vec{a}} = C'_{v,\vec{a}} \setminus \{x, \neg x\}$ in this case. Since $\vec{a}$ is an assignment to the $\vec{p}$ variables and $x \notin \vec{p}$, $C'_{u,\vec{a}}|_{\vec{a}} = C''_{u,\vec{a}}$ is a $\vec{q}$-clause and can be derived using $A(\vec{a}, \vec{q})$ alone via Q-Res. (Either $C''_{u,\vec{a}}$ already equals $C''_{v,\vec{a}}$, or $x$ needs to be dropped. In the latter case, the condition on $\operatorname{lv}(x)$ is satisfied at $C''_{u,\vec{a}}$ because it is satisfied at $C_v$ in $\pi$ and $C''_{v,\vec{a}} \subseteq C_v$. So we can drop $x$ from $C''_{v,\vec{a}}$ to get $C''_{u,\vec{a}}$.)

The situation is dual for the case when $g_u(\vec{a}) = 1$; we get $\vec{r}$-clauses.

This gives the interpolating circuit from Q-Res proofs. As promised, we now show that the above methods can be extended to the most powerful CDCL-based QBF proof system LQU$^+$-Res, and hence to all CDCL-based proof systems.

### 5.1.2 Interpolants from $\mathsf{LQU}^+$-Res Proofs

As mentioned in the proof idea, for an $\mathsf{LQU^+\text{-}Res}$ proof $\pi$ of $\mathcal{F}$, we construct a circuit $C_\pi$ with input $\vec{p}$, and proof-like structure $\pi'$ and proof $\pi''$.

**Construction of the Circuit $C_\pi$:** The circuit $C_\pi$ is defined exactly as for $\mathsf{Q\text{-}Res}$ proofs.

**Construction of $\pi'$ and $\pi''$:** Following our proof idea, we now construct a proof-like structure $\pi'(\vec{a})$, which depends on the assignment $\vec{a}$ to the $\vec{p}$ variables, the proof $\pi$ of $\mathcal{F}$, and the circuit $C_\pi$. For each node $u$ in $\pi$ with clause $C_u$, we associate a clause $C'_{u,\vec{a}}$ in $\pi'(\vec{a})$. We do this almost exactly as above, however note that in $\mathsf{LQU^+\text{-}Res}$ we need to handle the special literals $u^*$ as well.

As already mentioned, from the structure $\pi'(\vec{a})$, we get another structure $\pi''(\vec{a})$ by instantiating $\vec{p}$ variables by the assignment $\vec{a}$ in each clause of $\pi'(\vec{a})$, cutting away any edge out of a node where the clause evaluates to 1, and deleting nodes which now have no path to the root node. That is, for each survived node $u$ in $\pi''(\vec{a})$, the associated clause $C''_{u,\vec{a}}$ is equal to $C'_{u,\vec{a}}|_{\vec{a}}$.

We show (by induction on the height of $u$ in $\pi$) that:

1. $C'_{u,\vec{a}} \preceq C_u$.

2. $g_u(\vec{a}) = 0 \implies C''_{u,\vec{a}}$ is a $\vec{q}$-clause and can be obtained from the clauses of $A(\vec{a}, \vec{q})$ alone using the rules of system $\mathsf{LQU^+\text{-}Res}$.

3. $g_u(\vec{a}) = 1 \implies C''_{u,\vec{a}}$ is a $\vec{r}$-clause and can be obtained from the clauses of $B(\vec{a}, \vec{r})$ alone using the rules of system $\mathsf{LQU^+\text{-}Res}$.

As described in the proof outline, this suffices to conclude that $C_\pi$ computes an interpolant. We now present the construction details.

**At Leaf Level:** Let node $u$ be a leaf in $\pi$. Then $C'_{u,\vec{a}} = C_u$; that is, we copy the clause as it is. Trivially, we have $C'_{u,\vec{a}} \preceq C_u$. By construction of $C_\pi$, the conditions concerning $g_u(\vec{a})$ and $C''_{u,\vec{a}}$ are satisfied.

At an internal node we distinguish four cases based on the rule that was applied.

**At an Internal Node with Universal Reduction:** Let node $u$ be an internal node in $\pi$ corresponding to a universal reduction step on some universal variable $x$ or $x^*$. Let node $v$ be its only child. Here we consider only the case where the universal literal is $x$. The case of $x^*$ is identical. We have

$$\frac{C_v = \overbrace{D_v \vee x}^{\text{node } v}}{C_u = \underbrace{D_v}_{\text{node } u}}, \qquad x \text{ is a universal variable, } \forall l \in D_v, \mathrm{lv}(l) \leq \mathrm{lv}(x).$$

In this case, define $C'_{u,\vec{a}} = C'_{v,\vec{a}} \setminus \{x, \neg x, x^*\}$. By induction, $C'_{v,\vec{a}} \preceq C_v = D_v \vee x$. Therefore, $C'_{u,\vec{a}} = C'_{v,\vec{a}} \setminus \{x, \neg x, x^*\} \preceq D_v = C_u$.

If $g_u(\vec{a}) = 0$, then we know that $g_v(\vec{a}) = 0$ as $g_u(\vec{a}) = g_v(\vec{a})$. By the induction hypothesis, we know that $C''_{v,\vec{a}} = C'_{v,\vec{a}}|_{\vec{a}}$ is a $\vec{q}$-clause and can be derived using $A(\vec{a}, \vec{q})$ alone via $\mathsf{LQU^+\text{-}Res}$. Recall that $C'_{u,\vec{a}} = C'_{v,\vec{a}} \setminus \{x, \neg x, x^*\}$ in this case. Since $\vec{a}$ is an assignment to the $\vec{p}$ variables and $x \notin \vec{p}$, $C'_{u,\vec{a}}|_{\vec{a}} = C''_{u,\vec{a}}$ is a $\vec{q}$-clause and can be derived using $A(\vec{a}, \vec{q})$ alone via $\mathsf{LQU^+\text{-}Res}$. (Either $C''_{u,\vec{a}}$ already equals $C''_{v,\vec{a}}$, or $x$ needs to be dropped. In the latter case, the condition on $\mathrm{lv}(x)$ is satisfied at $C''_{u,\vec{a}}$ because it is satisfied at $C_v$ in $\pi$ and $C''_{v,\vec{a}} \preceq C_v$. So we can drop $x$ from $C''_{v,\vec{a}}$ to get $C''_{u,\vec{a}}$.)

The situation is dual for the case when $g_u(\vec{a}) = 1$; we get $\vec{r}$-clauses.

**At an Internal Node with $\vec{p}$-resolution:** Let node $u$ in the proof $\pi$ correspond to a resolution step with pivot $x \in \vec{p}$. Note that $x$ is existential, as $\vec{p}$ variables occur only existentially in $\mathcal{F}$. We have

$$\frac{C_v = \overbrace{C_1 \vee U_1 \vee x}^{\text{node } v} \qquad \overbrace{C_2 \vee U_2 \vee \neg x}^{\text{node } w} = C_w}{C_u = \underbrace{C_1 \vee C_2 \vee U}_{\text{node } u}}.$$

In the assignment $\vec{a}$, if $x = 0$, then define $C'_{u,\vec{a}} = C'_{v,\vec{a}} \setminus \{x\}$ and if $x = 1$ then define $C'_{u,\vec{a}} = C'_{w,\vec{a}} \setminus \{\neg x\}$. By induction, we have $C'_{v,\vec{a}} \preceq C_v$ and $C'_{w,\vec{a}} \preceq C_w$. So, if $x = 0$, we have $C'_{u,\vec{a}} = C'_{v,\vec{a}} \setminus \{x\} \preceq C_1 \vee U_1 \preceq C_u$. If $x = 1$, we have $C'_{u,\vec{a}} \preceq C'_{w,\vec{a}} \setminus \{\neg x\} \preceq C_2 \vee U_2 \preceq C_u$.

In this case $g_u$ is a selector gate. If $x = 0$ in the assignment $\vec{a}$, then $g_u(\vec{a}) = g_v(\vec{a})$ and $C'''_{u,\vec{a}} = C'''_{v,\vec{(a)}}$. Since the conditions concerning $g_v(\vec{a})$ and $C'''_{v,\vec{a}}$ are satisfied by induction, the conditions concerning $g_u(\vec{a})$ and $C'''_{u,\vec{a}}$ are satisfied as well. Similarly, if $x = 1$, then $g_u(\vec{a}) = g_w(\vec{a})$ and $C'''_{u,\vec{a}} = C'''_{w,\vec{a}}$, and the statements that are inductively true at $w$ hold at $u$ as well.

**At an Internal Node with $\vec{q}$-resolution:** Let node $u$ in the proof $\pi$ correspond to a resolution step with pivot $x \in \vec{q}$. Note that $x$ may be existential or universal. We have

$$\frac{C_v = \overbrace{C_1 \vee U_1 \vee x}^{\text{node } v} \qquad \overbrace{C_2 \vee U_2 \vee \neg x}^{\text{node } w} = C_w}{C_u = \underbrace{C_1 \vee C_2 \vee U}_{\text{node } u}}, \quad x \in \vec{q}.$$

In this case, we use the value of gate $g_u$ in circuit $C_\pi$ on input $\vec{a}$.

If $g_v(\vec{a}) = 1$ then define $C'_{u,\vec{a}} = C'_{v,\vec{a}}$. By induction, we know that $C'''_{u,\vec{a}} = C'''_{v,\vec{a}}$ is an $\vec{r}$-clause. Since $x$ is a $\vec{q}$-variable and is not instantiated by $\vec{a}$, it must be the case that $x \notin C'_{v,\vec{a}}$. Thus $C'_{u,\vec{a}} = C'_{v,\vec{a}} \preceq C_v \setminus \{x\} \preceq C_u$.

Else if $g_w(\vec{a}) = 1$, define $C'_{u,\vec{a}} = C'_{w,\vec{a}}$. By a similar analysis as above, $C'_{u,\vec{a}} = C'_{w,\vec{a}} \preceq C_w \setminus \{\neg x\} \preceq C_u$.

If $g_v(\vec{a}) = g_w(\vec{a}) = 0$, and if $x \notin C'_{v,\vec{a}}$, define $C'_{u,\vec{a}} = C'_{v,\vec{a}}$. Otherwise, if $\neg x \notin C'_{w,\vec{a}}$, define $C'_{u,\vec{a}} = C'_{w,\vec{a}}$. It follows from induction that $C'_{u,\vec{a}} \preceq C_u$.

Else, define $C'_{u,\vec{a}}$ to be the resolvent of $C'_{v,\vec{a}}$ and $C'_{w,\vec{a}}$ on $x$. (see below the 'note', why (LD)-resolution is valid here in $\pi'(\vec{a})$). By induction, we know that $C'_{v,\vec{a}} \setminus \{x\} \preceq C_1 \vee U_1$ and $C'_{w,\vec{a}} \setminus \{\neg x\} \preceq C_2 \vee U_2$. Hence $C'_{u,\vec{a}} \preceq C_1 \vee C_2 \vee U = C_u$.

We need to verify the conditions on $g_u(\vec{a})$ and $C''_{u,\vec{a}}$. The case when $g_u(\vec{a}) = 1$ is immediate, since $C''_{u,\vec{a}}$ copies a clause known by induction to be an $\vec{r}$-clause. So now consider the case when $g_u(\vec{a}) = 0$. By induction, we know that both $C''_{v,\vec{a}} = C'_{v,\vec{a}}|_{\vec{a}}$ and $C''_{w,\vec{a}} = C'_{w,\vec{a}}|_{\vec{a}}$ are $\vec{q}$-clauses and can be derived using $A(\vec{a}, \vec{q})$ alone via LQU$^+$-Res.

We have three cases. If $C'_{u,\vec{a}} = C'_{v,\vec{a}}$ or $C'_{u,\vec{a}} = C'_{w,\vec{a}}$, then by induction we are done. Otherwise, $C'_{u,\vec{a}}$ is obtained from $C'_{v,\vec{a}}$ and $C'_{w,\vec{a}}$ via a resolution step on pivot $x$. Since $\vec{a}$ is an assignment to the $\vec{p}$ variables and $x \notin \vec{p}$, $C''_{u,\vec{a}}$ can be derived from $C''_{v,\vec{a}}$ and $C''_{w,\vec{a}}$ via the same (LD)-resolution step.

**Note:** A simple observation is that $C'_{u,\vec{a}}$ is always a subset of $C_u$ with only one exception, which is that some special symbol $u^*$ in $C_u$ may be converted into $u$ in $C'_{u,\vec{a}}$. This leads us to define the relation $\preceq$. Also, the resolution step in $\pi''(\vec{a})$ is applicable in LQU$^+$-Res because

1. Every mergable universal variable in $C''_{v,\vec{a}}$ and $C''_{w,\vec{a}}$ was also mergable earlier in $C_v$ and $C_w$ in $\pi$.

2. Every common existential variable in $C''_{v,\vec{a}}$ and $C''_{w,\vec{a}}$ was also an existential variable in $C_v$ and $C_w$. Note that existential variables are not mergable.

3. Every non-mergable universal variable in $C''_{v,\vec{a}}$ and $C''_{w,\vec{a}}$ was also a non-mergable universal pair in $C_v$ and $C_w$.

4. The operations do not disturb the levels of variables, therefore if variable $x$ satisfies the level condition in $\pi$ it satisfies it in $\pi''(\vec{a})$ as well.

**At an Internal Node with $\vec{r}$-resolution:** Let node $u$ in $\pi$ correspond to a resolution step with pivot $x \in \vec{r}$. This is dual to the case above.

### 5.1.3 Monotone Interpolation for $\mathsf{LQU}^{+}$-$\mathsf{Res}$

To transfer known circuit lower bounds into size of proof bounds, we need a monotone version of the previous interpolation theorems, which we prove next.

**Theorem 5.4.** *LQU$^{+}$-Res, and therefore all CDCL-based QBF Resolution systems, have monotone feasible interpolation.*

*Proof.* In previous sections, we have shown that the circuit $C_\pi(\vec{p})$ is a correct interpolant for the QBF sentence $\mathcal{F}$. That is, if $C_\pi(\vec{p}) = 0$ then $\mathcal{Q}\vec{q}.A(\vec{a},\vec{q})$ is false, and if $C_\pi(\vec{p}) = 1$ then $\mathcal{Q}\vec{r}.B(\vec{a},\vec{r})$ is false.

However, if $\vec{p}$ occurs only positively in $A(\vec{p},\vec{q})$ then we construct a monotone circuit $C_\pi^{mon}(\vec{p})$ such that, on every $0,1$ assignment $\vec{a}$ to $\vec{p}$ we have

$$\begin{aligned} C_\pi^{mon}(\vec{a}) = 0 &\implies \mathcal{Q}\vec{q}.A(\vec{a},\vec{q}) \text{ is false, and} \\ C_\pi^{mon}(\vec{a}) = 1 &\implies \mathcal{Q}\vec{r}.B(\vec{a},\vec{r}) \text{ is false.} \end{aligned}$$

We obtain $C_\pi^{mon}(\vec{p})$ from $C_\pi(\vec{p})$ by replacing all selector gates $g_u = \mathsf{sel}(x, g_v, g_w)$ by the following monotone ternary connective: $g_u = (x \vee g_v) \wedge g_w$ where nodes $v$ and $w$ are the children of $u$ in $\pi$. We also change the proof-like structure $\pi'(\vec{a})$; the construction is the same as before except that at $\vec{p}$-resolution nodes, the rule for fixing $C'_{u,\vec{a}}$ is also changed to reflect the monotone function used instead.

More precisely, the functions $\mathsf{sel}(x, g_v, g_w)$ and $g_u = (x \vee g_v) \wedge g_w$ differ only when $x = 0$, $g_v(\vec{a}) = 1$, and $g_w(\vec{a}) = 0$. We set $C'_{u,\vec{a}}$ to $C'_{w,\vec{a}} \setminus \{\neg x\}$ if $x = 1$ or if $x = 0$, $g_v(\vec{a}) = 1$ and $g_w(\vec{a}) = 0$, and to $C'_{v,\vec{a}} \setminus \{x\}$ otherwise.

We need to show that at the differing setting, the inductive statements relating the modified $C'_{u,\vec{a}}$, $g_u(\vec{a})$ and $C''_{u,\vec{a}}$ continue to hold. The relation $C''_{u,\vec{a}} \preceq C_u$ holds by induction. Now consider the gate values.

We know by induction that $g_v(\vec{a}) = 1$ means that $C''_{v,\vec{a}}$ is an $\vec{r}$-clause and can be derived from $B(\vec{a}, \vec{r})$ alone. When $x = 0$, $C'_{u,\vec{a}} = C'_{v,\vec{a}}$ and the original selector gate would have output the value of $g_v(\vec{a})$ which is a 1. Hence $C''_{u,\vec{a}}$ is an $\vec{r}$-clause. However, observe that at this setting, $g_w(\vec{a}) = 0$, which means by induction that $C''_{w,\vec{a}}$ is a $\vec{q}$-clause and can be derived using $A(\vec{a}, \vec{q})$ clauses alone via the appropriate proof system. Thus by our assumption about $\vec{p}$ variables appearing only positively in $A$, the clause $C'_{w,\vec{a}}$ does not contain $\neg x$. Thus we can safely assign $C'_{u,\vec{a}} = C'_{w,\vec{a}}$. This completes the proof. □

### 5.1.4 Exponential Lower Bounds for $\mathsf{LQU}^+$-$\mathsf{Res}$

Consider the false clique-co-clique formulas $\Phi_{n,k}$ from Section 4.2.3. Recall that the formulas $\Phi_{n,k} \equiv \exists\vec{p}\exists\vec{q}\forall\vec{r}\exists\vec{t}.\ [A_{n,k}(\vec{p}, \vec{q}) \wedge B_{n,k}(\vec{p}, \vec{r}, \vec{t})]$ encode the obviously false statement that a given graph on $n$ vertices (encoded by $\vec{p}$ variables) both has and does not have a $k$-clique.

The lower bounds for the formulas $\Phi_{n,k}$ in the proof system $\mathsf{LQU}^+$-$\mathsf{Res}$ will be directly transferred from the following monotone circuit lower bound for the problem $\textsc{Clique}(n, k)$, asking whether a given graph with $n$ nodes has a clique of size $k$.

**Theorem 5.5** (Alon, Boppana 87 [1]). *All monotone circuits that compute* $\textsc{Clique}(n, n/2)$ *are of exponential size.*

Observe that the formula $\Phi_{n,n/2}$ has the unique interpolant $\textsc{Clique}(n, n/2)(\vec{p})$. But since all monotone circuits for this are of exponential size by Theorem 5.5 and monotone circuits of size polynomial in $\mathsf{LQU}^+$-$\mathsf{Res}$ proof size can be extracted by Theorem 5.4, all such proofs must be of exponential size, yielding:

**Theorem 5.6.** *The QBFs $\Phi_{n,n/2}$ require exponential-size proofs in* $\mathsf{LQU}^+$*-*$\mathsf{Res}$*.*

## 5.2 Feasible (Monotone) Interpolation for CP+∀red and Unconditional Lower Bounds

Pudlák in [67] established the feasible interpolation technique for Cutting Planes proof system and hence prove the first exponential lower bounds for Cutting Planes. In [67], he initially described the technique first established by Krajíček [62] for the Resolution proof system, and then extend it to the Cutting Planes proof system. In this section we show that our new QBF proof system CP+∀red (see Chapter 4) also admits feasible monotone interpolation, and as a consequence prove unconditional lower bounds for CP+∀red. We adapt the technique used by Pudlák in [67].

As in Section 5.1.1, consider a false QBF of the form

$$\mathcal{F} = \exists \vec{p} \mathcal{Q} \vec{q} \mathcal{Q} \vec{r}. \left[ A'(\vec{p}, \vec{q}) \wedge B'(\vec{p}, \vec{r}) \right]$$

where $\vec{p}$, $\vec{q}$, and $\vec{r}$ are mutually disjoint sets of propositional variables, $A'(\vec{p}, \vec{q})$ is a set of clauses using only the $\vec{p}$ and $\vec{q}$ variables, and $B'(\vec{p}, \vec{r})$ is a set of clauses using only the $\vec{p}$ and $\vec{r}$ variables. Thus $\vec{p}$ are the common variables between them. The $\vec{q}$ and $\vec{r}$ variables can be quantified arbitrarily, with any number of quantification levels. Since $\mathcal{F}$ is false, on any assignment $\vec{a}$ to the variables in $\vec{p}$, either $\mathcal{F}_{\vec{a},0} = \mathcal{Q}\vec{q}.\ A'(\vec{a}, \vec{q})$ or $\mathcal{F}_{\vec{a},1} = \mathcal{Q}\vec{r}.\ B'(\vec{a}, \vec{r})$ (or both) must be false. An interpolant for $\mathcal{F}$ is a Boolean function that, given $\vec{a}$, indicates which of $\mathcal{F}_{\vec{a},0}$, $\mathcal{F}_{\vec{a},1}$ is false.

Recall the definition of interpolating circuit for CDCL-based QBF Resolution calculi (Definition 5.1). Obviously it is a Boolean circuit. However, dealing with CP+∀red naturally gives rise to arithmetic rather than Boolean circuits, as was done in the classical case in [67]. Generalising this to the case of QBFs, we have the following definitions.

**Definition 5.7.** *[67] A monotone real circuit is a circuit which computes with real numbers and uses arbitrary non-decreasing real unary and binary functions as gates.*

*We say that a monotone real circuit computes a Boolean function (uniquely determined by the circuit), if for all inputs of* $0$*'s and* $1$*'s the circuit outputs* $0$ *or* $1$*.*

**Definition 5.8.** *A QBF proof system $S$ admits* monotone real feasible interpolation *if for any false QBF $\mathcal{F}$ of the form $\exists\vec{p}\mathcal{Q}\vec{q}\mathcal{Q}\vec{r}.\big[A'(\vec{p},\vec{q})\wedge B'(\vec{p},\vec{r})\big]$ where the $\vec{p}$ variables occur only positively in $A'$ or only negatively in the clauses of $B'$, and for any $S$-proof $\pi$ of $\mathcal{F}$, we can extract from $\pi$ a monotone real circuit $C$ of size polynomial in the length of $\pi$ and the number $n$ of $\vec{p}$ variables, such that $C$ computes a Boolean function, and on every $0,1$ assignment $\vec{a}$ for $\vec{p}$,*

$$\begin{aligned} C(\vec{a})=0 &\implies \mathcal{Q}\vec{q}.A'(\vec{a},\vec{q}) \text{ is false, and} \\ C(\vec{a})=1 &\implies \mathcal{Q}\vec{r}.B'(\vec{a},\vec{r}) \text{ is false.} \end{aligned}$$

*Such a $C$ is called a monotone real interpolating circuit for $\mathcal{F}$.*

Note that if $S$ admits monotone feasible interpolation then it also admits monotone real feasible interpolation. The converse may not be true: arbitrary non-decreasing real functions are allowed in monotone real feasible interpolation, but their conversions to Boolean functions may be non-monotone in the bit representation.

We prove that the CP+∀red proof system for false QBFs has this property:

**Theorem 5.9.** *CP+∀red for false QBFs admits monotone real feasible interpolation.*

To prove this, we will actually prove a stronger theorem, about interpolants for all false quantified sets of inequalities (not just those arising from false QBFs). That is, we prove the following theorem:

**Theorem 5.10.** *$\mathsf{CP}$+$\forall$$\mathsf{red}$ for inequalities (from Definition 4.1) admits monotone real feasible interpolation. That is, let $\varphi$ be any false quantified set of inequalities of the form $\exists\vec{p}\mathcal{Q}\vec{q}\mathcal{Q}\vec{r}.\big[A(\vec{p},\vec{q}) \wedge B(\vec{p},\vec{r})\big]$ where $A \cup B$ also includes all Boolean axioms, and where the coefficients of $\vec{p}$, are either all non-negative in $A$ or are all non-positive in $B$. If $\varphi$ has a $\mathsf{CP}$+$\forall$$\mathsf{red}$-proof $\pi$, of length $l$, then we can extract from $\pi$ a monotone real circuit $C$ of size polynomial in $l$ and the number $n$ of $\vec{p}$ variables in $\varphi$, such that $C$ computes a Boolean function, and on every $0,1$ assignment $\vec{a}$ for $\vec{p}$,*

$$C(\vec{a}) = 0 \implies \mathcal{Q}\vec{q}.A(\vec{a},\vec{q}) \text{ is false, and}$$
$$C(\vec{a}) = 1 \implies \mathcal{Q}\vec{r}.B(\vec{a},\vec{r}) \text{ is false.}$$

*Such a $C$ is called a monotone real interpolating circuit for $\varphi$.*

Before proving this theorem, let us see why it implies Theorem 5.9

*Proof.* (of Theorem 5.9.) Let $\mathcal{F}$ be the given false QBF of the form described above, that is,

$$\mathcal{F} = \exists\vec{p}\mathcal{Q}\vec{q}\mathcal{Q}\vec{r}.\big[A'(\vec{p},\vec{q}) \wedge B'(\vec{p},\vec{r})\big]$$

Encoding it as a quantified set of inequalities as per Definition 4.2, we get a quantified set of linear inequalities $\varphi = \mathcal{Q}.\ F$, of the form

$$\varphi = \exists\vec{p}\mathcal{Q}\vec{q}\mathcal{Q}\vec{r}.\big[A(\vec{p},\vec{q}) \cup B(\vec{p},\vec{r})\big]$$

Here, $A(\vec{p},\vec{q})$ contains inequalities $R(C)$ for all clauses $C \in A'$; these are of the form $\sum\limits_k e_k p_k + \sum\limits_i f_i q_i \geq b$. Similarly, $B(\vec{p},\vec{r})$ contains inequalities $R(C)$ for all $C \in B'$; these are of the form: $\sum\limits_k e_k p_k + \sum\limits_j g_j r_j \geq b$. The Boolean axioms corresponding to the $\vec{q}$ variables are included in $A$, those corresponding to the $\vec{r}$ variables are included in $B$. The Boolean axioms corresponding to the $\vec{p}$ variables also have to be included in $A \cup B$. They have both positive and negative coefficicents. If $\vec{p}$ occurs only positively in $A'$, we include these in $B$, otherwise we include them in $A$.

Since $\mathcal{F}$ is false, so is $\varphi$. On any assignment $\vec{a}$ to the variables in $\vec{p}$, either $\varphi_{\vec{a},0} = \mathcal{Q}\vec{q}.\ A(\vec{a},\vec{q})$ or $\varphi_{\vec{a},1} = \mathcal{Q}\vec{r}.\ B(\vec{a},\vec{r})$ (or both) must be false. Furthermore, for $b \in \{0,1\}$, $\varphi_{\vec{a},b}$ is false exactly when $\mathcal{F}_{\vec{a},b}$ is false. Thus a monotone real interpolating circuit for $\varphi$ is also a monotone real interpolating circuit for $\mathcal{F}$.

Note that if $\vec{p}$ occurs only positively in $A'$, then the coefficients $e_k$ in all the inequalities in $A$ are non-negative. Similarly, if $\vec{p}$ occurs only negatively in $B'$, then the coefficients $e_k$ in all the inequalities in $B$ are non-positive. Hence, invoking Theorem 5.10 on $\varphi$, we obtain the desired monotone real interpolating circuit for $\mathcal{F}$ and for $\varphi$. □

Now we get back to constructing interpolants for CP+∀red with inequalities.

*Proof.* (of Theorem 5.10.) Let $\pi = \exists\vec{p}\mathcal{Q}\vec{q}\mathcal{Q}\vec{r}.\ [I'_1, \dots, I'_l]$ be a CP+∀red refutation of $\varphi$. The idea, as in [67], is to associate with each inequality

$$I \equiv \sum_k e_k p_k + \sum_i f_i q_i + \sum_j g_j r_j \geq D$$

in $\pi$, two inequalities

$$I_0 \equiv \sum_i f_i q_i \geq D_0, \quad I_1 \equiv \sum_j g_j r_j \geq D_1$$

depending on the Boolean assignment $\vec{a}$ to the $\vec{p}$ variables, in such a way that

- $I_0$ and $I_1$ together imply $I|_{\vec{a}}$. (It suffices to ensure $D_0 + D_1 \geq D - \sum_k e_k a_k$.)
- $I_0$ can be derived solely from the $\mathcal{Q}\vec{q}.A(\vec{a},\vec{q})$ part in CP+∀red.
- $I_1$ can be derived solely from the $\mathcal{Q}\vec{r}.B(\vec{a},\vec{r})$ part in CP+∀red.

Then the inequalities corresponding to the last step of the proof, $I'_l$, are $0 \geq D_0$ and $0 \geq D_1$, with $D_0 + D_1 \geq 1$. Hence $D_0 > 0 \implies \vec{Q}\vec{q}.A(\vec{a}, \vec{q})$ is false, and $D_0 \leq 0 \implies D_1 > 0 \implies \vec{Q}\vec{r}.B(\vec{a}, \vec{r})$ is false. Note that we only need to compute one of the values $D_0$, $D_1$ to identify a false part of $\varphi$. Furthermore, we will show that if all the coefficients $e_k$ in $B(\vec{p}, \vec{r})$ are non-positive, then $D_1$ can be computed by a real monotone circuit of size $O(nl)$. If all the coefficients $e_k$ in $A(\vec{p}, \vec{q})$ are non-negative, then we will show that $-D_0$ can be computed by a real monotone circuit of size $O(nl)$. (The inputs to the circuit are an assignment $\vec{a}$ to the $\vec{p}$ variables.) Applying the unary non-decreasing threshold function $D_1 > 0$? or $-D_0 \geq 0$? to its output will then give a monotone real interpolating circuit for $\varphi$.

We first describe the computation of $D_0$ and $D_1$ at each inequality. These are computed by two circuits, both of which have exactly the structure of $\pi$.

Consider the case when all $e_k$ in $B(\vec{p}, \vec{r})$ are non-positive; the other case is analogous. All axioms are considered as either $A$-axioms or as $B$-axioms. The Boolean axioms concerning $\vec{p}$ variables are treated as $A$-axioms in this case.

The computation of $D_0$ and $D_1$ proceeds bottom-up as described in Table 5.1.

As in the proof argument from [67], a straightforward induction shows that with these computations, at each proof line $I$, the inequalities $I_0$ and $I_1$ together imply $I\mid_{\vec{a}}$, and that each $I_0$ can be derived from the $A$-axioms alone and each $I_1$ can be derived from the $B$-axioms alone.

All the operations required for the arithmetic and reduction steps compute non-decreasing functions. At the axioms, note that the dependence of the $D_1$ values on the assignment values $\vec{a}$ is always with non-negative coefficients $-e_k$; hence these functions are also non-decreasing. Thus we obtain a monotone real circuit for $D_1$, of size $O(nl)$. □

Using our monotone interpolation theorem (Theorem 5.9), we now prove an uncon-

Table 5.1: Computation of $D_0$ and $D_1$ in the proof of Theorem 5.10

| How inequality $I$ is obtained | $D_0$ | $D_1$ |
|---|---|---|
| Axioms: | | |
| $p_k \geq 0$ | $-a_k$ | $0$ |
| $-p_k \geq -1$ | $a_k - 1$ | $0$ |
| $-q_i \geq -1$ | $-1$ | $0$ |
| $-r_j \geq -1$ | $0$ | $-1$ |
| $q_j \geq 0$ or $r_j \geq 0$ | $0$ | $0$ |
| $\sum_k e_k p_k + \sum f_i q_i \geq D$ | $D - \sum e_k a_k$ | $0$ |
| $\sum_k e_k p_k + \sum g_j r_j \geq D$ | $0$ | $D - \sum e_k a_k$ |
| Arithmetic: | | |
| Addition $I = I' + I''$ | $D_0' + D_0''$ | $D_1' + D_1''$ |
| Multiplication $I = hI'$, $h > 0$ | $h \times D_0'$ | $h \times D_1'$ |
| Division $I = I'/c$, $c > 0$ | $\left\lceil \frac{D_0'}{c} \right\rceil$ | $\left\lceil \frac{D_1'}{c} \right\rceil$ |
| Reduction: $I = I' \mid_{u=b}$; coefficient of $u$ in $I'$ is $h$. | | |
| $h > 0$ | $D_0'$ | $D_1'$ |
| $h < 0$ and $u$ is a $\vec{q}$ variable | $D_0' - h$ | $D_1'$ |
| $h < 0$ and $u$ is an $\vec{r}$ variable | $D_0'$ | $D_1' - h$ |

ditional lower bound for the CP+∀red proof system.

Again consider the false clique-co-clique formulas $\Phi_{n,k}$ from Section 4.2.3. As already mentioned, the formulas $\Phi_{n,k} \equiv \exists \vec{p} \exists \vec{q} \forall \vec{r} \exists \vec{t}. A_{n,k}(\vec{p}, \vec{q}) \wedge B_{n,k}(\vec{p}, \vec{r}, \vec{t})$ encodes the obviously false statement that a given graph on $n$ vertices (encoded by $\vec{p}$ variables) both has and does not have a $k$-clique.

Now suppose $\Phi_{n,k}$ has a CP+∀red proof of length $l$. From Theorem 5.9, we obtain a monotone real circuit $C$ of size $O(l + n^2)$ computing a Boolean function, such that for every $0, 1$ input vector $\vec{p}$ of length $\binom{n}{2}$ encoding a graph $G_n$ on $n$ vertices, $C(\vec{p}) = 1 \iff G_n$ has a $k$ clique.

In [67], Pudlák showed the following exponential lower bound on the size of real monotone circuits interpolating the famous "clique-color" encodings.

**Theorem 5.11** ( [67]). *Suppose that the inputs for a monotone real circuit $C$ are $0, 1$ vectors of length $\binom{n}{2}$ encoding in the natural way graphs on an $n$-element set. Suppose that $C$ outputs $1$ on all cliques of size $k$ and outputs $0$ on all complete*

*$(k-1)$-partite graphs, where $k = \lfloor \frac{1}{8}(n/\log n)^{2/3} \rfloor$. Then the size of the circuit is at least $2^{\Omega((n/\log n)^{1/3})}$.*

In some earlier literature, clique-color has been referred to as clique-co-clique. However, this is misleading because the clique-color encoding is weaker than $\Phi_{n,k}$ in the following sense. The clique-color encoding says that there exists a graph which has a $k$-clique and is complete $(k-1)$-partite (maximal $(k-1)$-colorable). A graph may neither have a $k$-clique nor be complete $(k-1)$-partite, so both parts of the clique-color formula may be false. Our clique-co-clique formula, on the other hand, expresses that there exists a graph which has a $k$-clique and which does not have a $k$-clique. For every graph, exactly one part of the clique-co-clique formula is false.

Since complete $(k-1)$-partite graphs have no $k$-clique, the real monotone interpolating circuit $C$ we obtain from a proof of $\Phi_{n,k}$ also satisfies the premise of Theorem 5.11. Hence, $C$ must be of exponential size. But $C$ is polynomially related to the length of the $\mathsf{CP}+\forall\mathsf{red}$ proof of $\Phi_{k,n}$. We have thus obtained the following:

**Corollary 5.12.** *For $k = \lfloor \frac{1}{8}(n/\log n)^{2/3} \rfloor$, the false QBFs $\Phi_{n,k}$ require exponential-length proofs in the $\mathsf{CP}+\forall\mathsf{red}$ proof system.*

## 5.3 Feasible (Monotone) Interpolation for semCP+∀red and Unconditional Lower Bounds

In this section, we establish feasible monotone interpolation for $\mathsf{semCP}+\forall\mathsf{red}$ (for the definition, see Section 4.4). We adapt the corresponding proof technique used in the classical case from [47]. Using their technique for the semantic inference rule, and handling axioms and ∀-reduction rules as described in the proof of Theorem 5.10, everything goes through as desired.

**Theorem 5.13.** *SemCP+$\forall$red admits monotone real feasible interpolation for false QBFs.*

*Proof.* Let $\mathcal{F} = \exists\vec{p}Q\vec{q}Q\vec{r}(A'(\vec{p},\vec{q}) \wedge B'(\vec{p},\vec{r}))$ be a false QBF formula. Without loss of generality, the $\vec{p}$ variables appear only negatively in $B'(\vec{p},\vec{r})$. Consider the standard encoding $\varphi = \exists\vec{p}Q\vec{q}Q\vec{r}(A(\vec{p},\vec{q}) \wedge B(\vec{p},\vec{r}))$ of $\mathcal{F}$ (see Definition 4.2). Clearly the coefficient of $\vec{p}$ variables in $B$ are non-positive. As discussed before it is sufficient to extract a monotone real feasible interpolation for $\varphi$. Let $\pi$ be any semCP+$\forall$red proof of $\varphi$, and as in the proof of Theorem 5.10, we construct a real monotone interpolating $C$ to detect whether $D_1 > 0$. Axioms and the $\forall$-reduction rule are handled exactly as in Theorem 5.10. Now suppose that the inequality $I \equiv \sum_k e_k p_k + \sum_i f_i q_i + \sum_j g_j r_j \geq D$ is semantically inferred from $I'$ and $I''$. We define $I_0, I_1$ by defining $D_0$ and $D_1$.

$$D_0 = \min\left\{\sum_i f_i q_i|_\gamma : \gamma \in \{0,1\}^{|\vec{q}|}, \gamma \text{ satisfies } I_0', I_0''\right\}$$
$$D_1 = \min\left\{\sum_j g_j r_j|_\tau : \tau \in \{0,1\}^{|\vec{r}|}, \tau \text{ satisfies } I_1', I_1''\right\}$$

It suffices to show that $D_0 + D_1 \geq D - \sum_k e_k a_k$. For $D_0$, let the minimum be achieved at assignment $\gamma_0$, and for $D_1$, let the minimum be achieved at assignment $\tau_1$. Let $\rho$ be the assignment to the $\vec{q}$ and $\vec{r}$ variables setting $\vec{q}$ as in $\gamma_0$ and $\vec{r}$ as in $\tau_1$. Then $\rho$ satisfies $I_0'$, $I_0''$, $I_1'$, $I_1''$ (at $\vec{p} = \vec{a}$). Hence by induction, $\rho$ satisfies $I'$ and $I''$. Since $I$ is inferred semantically from $I'$ and $I''$, $\rho$ satisfies $I$ as well. Hence

$$\begin{aligned} D_0 + D_1 &= \sum_i f_i q_i|_{\gamma_0} + \sum_j g_j r_j|_{\tau_1} \\ &= \left(\sum_i f_i q_i + \sum_j g_j r_j\right)|_\rho \\ &\geq D - \sum_k e_k a_k, \qquad \text{as required.} \end{aligned}$$

Since $\vec{p}$ appears only negatively in $B(\vec{p},\vec{r})$, $D_1$ is a non-decreasing function of $D_1'$

and $D_1''$. (As the values of $D_1'$ and $D_1''$ increase, the set of assignments $\tau$ over which we take the minimum shrinks, and so the minimum value can only increase or stay the same.) $\square$

Now we can use our monotone feasible interpolation theorem for achieving an unconditional exponential lower bound for semCP+∀red. Similarly to Corollary 5.12, we have the following:

**Corollary 5.14.** *For $k = \lfloor \frac{1}{8}(n/\log n)^{2/3} \rfloor$, the false QBFs $\Phi_{n,k}$ require exponential-length proofs in the semCP+∀red proof system.*

# Chapter 6

# Are Short Proofs Narrow in QBF Resolution Calculi?

As discussed in Chapter 2, a number of ingenious techniques have been designed to show lower bounds for the *size of resolution proofs*, among them are *feasible interpolation*, established by Krajíček in [62], and the *size-width* relation, established by Ben-Sasson and Wigderson, in their paper 'Short proofs are narrow – resolution made simple' [9]. Another important measure for Resolution is *space* [46]. As already mentioned, Atserias and Dalmau in [3] demonstrated that also space is tightly related to width. Indeed, showing lower bounds for width serves again as the primary method to obtain space lower bounds.

In Chapter 5, we have seen that the *feasible interpolation* technique applies to all CDCL-based QBF proof systems (in fact it also applies to all expansion-based proof systems [14]). We also showed that the feasible interpolation technique also applies to our new QBF proof systems based on Cutting Planes: CP+∀red and semCP+∀red.

In this Chapter, we address the question whether *lower bound techniques via width*, which have revolutionised propositional proof complexity, are also effective for QBF Resolution systems? We concentrate only on the following QBF systems: Q-Res,

$\forall$Exp+Res, and IR-calc; even here the picture is rather complex.

As already mentioned, though space and width have not been considered in QBF before, these notions straightforwardly apply to QBF Resolution systems. However, due to the $\forall$-reduction rule in Q-Res handling universal variables, it is relatively easy to enforce that universal literals accumulate in clauses of Q-Res proofs, thus always leading to large width, irrespective of size and space requirements (Lemma 6.4). This prompts us to consider *existential width* — counting only existential literals — as an appropriate width measure in QBF. We had already discussed our findings in Chapter 1, however for ease of reference, we briefly mention it once again:

**1. Negative Results.** Our main results show that the size-width relation of [9] as well as the space-width relation of [3] dramatically *fail* for Q-Res, even when considering the tighter existential width. To be precise, we prove that Tseitin transformations (see Section 2.1) of formulas $CR_n$ from [56] (see Section 3.2) have small size and space, but require large existential width in tree-like Q-Res (Theorem 6.6), thus refuting the size-width relation for tree-like Q-Res as well as the space-width relation for general dag-like Q-Res.

As the formulas $CR_n$ have $O(n^2)$ variables, they do not rule out size-width relations in general Q-Res. However, we show that different formulas, hard for tree-like Q-Res [56], provide counterexamples for size-width relations in full Q-Res (Theorem 6.8).

**2. Positive Results and Width-space-preserving Simulations.** After negative results, we prove some positive results for size-width-space relations for tree-like versions of the expansion-based Resolution systems $\forall$Exp+Res and IR-calc. We lift all the relations from tree-like resolution to $\forall\text{Exp+Res}_\text{T}$ (Theorem 6.19).

To lift these results to $\text{IR}_\text{T}$-calc (Theorem 6.20), we show a series of careful space and width-preserving simulations between tree-like Q-Res, $\forall$Exp+Res, and IR-calc. In particular, we show the surprising result that tree-like $\forall$Exp+Res and tree-like

IR-calc are equivalent (Lemma 6.15), thus providing a rare example of two proof systems that coincide in the tree-like, but are separated in the dag-like model [13]. In addition, our simulations provide a simpler proof for the simulation of tree-like Q-Res by ∀Exp+Res (Corollary 6.17), shown in [56] via a more involved argument.

Our last positive result is a size-space relation in tree-like Q-Res (Theorem 6.20), which we show by a pebbling game analogous to the classical relation in [46].

We start by defining size, width, and space for QBF Resolution calculi.

## 6.1 Size, Width and Space in Resolution Calculi

Recall from Chapter 2, the definition of complexity measures size, width, and space for Resolution. In this section, we state their relations in Resolution, and also explain how to apply these measures to QBF Resolution systems. While this is straightforward for size and space, we need a more elaborate discussion on what constitutes a good notion of width for QBF Resolution systems.

### 6.1.1 Defining Size, Width, and Space for QBF Resolution Calculi

For ease of reference, we again define the complexity measures size, width, and space for Resolution. For a CNF formula $F$, $|F|$ denotes the number of clauses in it, and $w(F)$ denotes the maximum number of literals in any clause of $F$, and we extend the same notation to QBFs with a CNF matrix.

For $P$ one of the calculi Resolution (Res), Q-Res, ∀Exp+Res, IR-calc, let $\pi \vdash_{\mathsf{P}} F$ (resp. $\pi \vdash_{\mathsf{P_T}} F$) denote that $\pi$ is a $P$-proof (tree-like $P$-proof, respectively), of the formula $F$. For a proof $\pi$ of $F$ in system $P$, its size $|\pi|$ is defined as the number of clauses in

$\pi$. The **size** complexity $S(\vdash_{\mathsf{P}} F)$ of deriving $F$ in $P$ is defined as $\min\{|\pi| \ : \ \pi \vdash_{\mathsf{P}} F\}$. The tree-like size complexity, denoted $S(\vdash_{\mathsf{P_T}} F)$, is $\min\{|\pi| \ : \ \pi \vdash_{\mathsf{P_T}} F\}$.

Note that in Chapter 4, the size of the CP+∀red proof denotes the bit-size representation of the proof, and not the number of lines in the proof. Length of a CP+∀red proof actually denotes the number of lines in the proof. However here the size of any QBF Resolution calculi denotes the number of clauses in the proof.

A second complexity measure is the minimal **width**. The width of a clause $C$ is the number of literals in $C$, denoted $w(C)$. The width of a CNF formula $F$, denoted $w(F)$, is the maximum width of a clause in $F$; $w(F) = \max\{w(C) \ : \ C \in F\}$. The width $w(\pi)$ of a proof $\pi$ is defined as the maximum width of any clause appearing in $\pi$, i.e, $w(\pi) = \max\{w(C) \ : \ C \in \pi\}$. The width $w(\vdash_{\mathsf{P}} F)$ of refuting a CNF formula $F$ in $P$ is defined as $\min\{w(\pi) \ : \ \pi \vdash_{\mathsf{P}} F\}$. Again the same notation extends to QBFs with CNF matrix.

Note that for width in any calculus, whether the proof is tree-like or not is immaterial, since a proof can always be made tree-like by duplication without increasing the width. We therefore drop the T subscript when talking about proof width.

The third complexity measure for Resolution calculi is **space**. Recall the definition of space from Section 2.3.1. We can directly adapt this definition to QBF Resolution calculi.

**Definition 6.1** (Space-oriented proof sequences). *A false QBF sentence $\mathcal{F}$ can be refuted in system $P$ within space $k$ if there is a sequence $\sigma$ of QBFs $\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_s$, all having the same quantifier prefix as $\mathcal{F}$, and with matrix $F_0, F_1, \ldots, F_s$, respectively, such that $F_0 = \emptyset$, $F_1$ contains a subset of clauses obtained from the corresponding axiom download in the proof system $P$, $F_s = \{\square\}$ (the empty clause), each $F_i$ has at most $k$ clauses, and for each $i < s$, $F_{i+1}$ is obtained from $F_i$ by one of the following rules:*

1. **Erase**: *$F_{i+1} = F_i \setminus \{C\}$ for some clause $C \in F_i$.*

2. **Inference**: *$F_{i+1} \subseteq F_i \cup \{C\}$ for $C$ obtained by applying any inference rule of the proof system $P$. In this step, one of the hypotheses used in the inference rule may be erased.*

3. **Axiom Download**: *$F_{i+1} = F_i \cup \{C\}$ for some clause $C$ obtained by applying the axiom download rule of the proof system $P$.*

*For a proof written as a sequence $\sigma$ as above, the clause space of $\sigma$, denoted by $CSpace(\sigma)$, is $\max_{i\in[s]}\{|F_i|\}$. The clause space needed to refute a QBF $\mathcal{F}$ in the $P$-proof system, denoted by $CSpace(\mid\!\frac{\ }{P}\,\mathcal{F})$, is the minimum $CSpace(\sigma)$ over all sequences $\sigma$ refuting $\mathcal{F}$.*

*If we modify the inference step above so that the clause(s) used to obtain the inference are erased in the same step, then any clause $D$ can be used at most once and we obtain a tree-like space-oriented $\mathcal{P}$-proof. Correspondingly we can define $CSpace(\mid\!\frac{\ }{P_T}\,\mathcal{F})$ as the minimum space used by any tree-like proof sequence refuting $\mathcal{F}$.*

### 6.1.2 Relations in Classical Resolution

In this section, we state some of the main relations between size, width, and space for Resolution. In Chapter 2, we have stated the size-width relation (Theorem 2.4) established by Ben-Sasson and Wigderson in [9]. Here for ease of reference we state it once again.

**Theorem 2.4 [9].** *For all unsatisfiable CNFs $F$ in $n$ variables the following holds:*

$$\begin{aligned} S(\mid\!\frac{\ }{\mathsf{Res}_T}\,F) &\geq 2^{w\left(\mid\!\frac{\ }{\mathsf{Res}}\,F\right)-w(F)}, \quad \textit{and} \\ S(\mid\!\frac{\ }{\mathsf{Res}}\,F) &= \exp\left(\Omega\left(\frac{\left(w\left(\mid\!\frac{\ }{\mathsf{Res}}\,F\right)-w(F)\right)^2}{n}\right)\right). \end{aligned}$$

Space complexity was introduced by Esteban and Torán in [46] and relations between space, size and width are explored (cf. also [19, 64]).

**Theorem 6.2** ( [46]). *For all unsatisfiable CNFs $F$ the following relation holds:* $S(\vdash_{\mathsf{Res}_T} F) \geq 2^{CSpace\left(\vdash_{\mathsf{Res}_T} F\right)} - 1$.

The fundamental relation between space and width was obtained by Atserias and Dalmau in [3]; a more direct proof was given recently in [48].

**Theorem 6.3** ( [3]). *For all unsatisfiable CNFs $F$ the following relation holds:* $w(\vdash_{\mathsf{Res}} F) \leq CSpace(\vdash_{\mathsf{Res}} F) + w(F) - 1$.

### 6.1.3 Existential Width: What is the Right Width Notion for QBFs?

We wish to explore the possibility of a similar approach as used in [9] to prove an analogue of Theorem 2.4 when dealing with QBFs. The following simple example shows that the relationships in Theorem 2.4 and Theorem 6.3 do not carry over for the system Q-Res.

Consider the following false QBF $\mathcal{F}_n$ over $2n + 1$ variables:

$$\begin{aligned} \mathcal{F}_n =& \forall u_1 \dots u_n \exists e_0 \exists e_1 \dots e_n. \\ C_0 : \quad & (e_0) \wedge \\ \text{For } i \in [n], D_i : \quad & (\bar{e}_{i-1} \vee u_i \vee e_i) \wedge \\ D_{n+1} : \quad & (\bar{e}_n) \end{aligned}$$

**Lemma 6.4.** *$S(\vdash_{\mathsf{Q\text{-}Res}_T} \mathcal{F}_n) \in O(n)$ and $CSpace(\vdash_{\mathsf{Q\text{-}Res}_T} \mathcal{F}_n) \in O(1)$, but $w(\vdash_{\mathsf{Q\text{-}Res}} \mathcal{F}_n) \in \Omega(n)$.*

*Proof.* For the upper bounds consider the following proof. For $i \in [n]$, let $C_i =$

$(u_1 \vee \cdots \vee u_i \vee e_i)$. For $i \in [n]$ in sequence, resolving $C_{i-1}$ and $D_i$ on variable $e_{i+1}$ gives $C_i$. Resolving $C_n$ and $D_{n+1}$ on variable $e_n$ gives the clause $U = (u_1 \vee \cdots \vee u_n)$. Finally, applying $\forall$-Red on the clause $U$ yields the empty clause in $n$ more steps.

This is a tree-like proof of size $O(n)$. Further, each resolution step involves an axiom clause, so at each step we need to hold just two clauses, and so the space requirement is $O(1)$.

Concerning the width lower bound, by the order of quantification in $\mathcal{F}_n$, every existential literal in $\mathcal{F}_n$ blocks any $\forall$-reduction. Therefore, in any refutation, when a $\forall$-reduction is first used, the clause $C$ has only universal variables. At this point, the empty clause is derivable from $C$ by a series of $\forall$ reductions. Note that if any clause is dropped from $\mathcal{F}_n$, the resulting sentence is no longer false. Thus any refutation must use all clauses. Hence $C$ must have all universal variables in it; it must be $(u_1 \vee \cdots \vee u_n)$ as all $u_i$ variables have been accumulated, without being reduced. Then clause $C$ has width $n$. □

Noting that $w(\mathcal{F}_n) = 3$, Lemma 6.4 implies that the relationships from Theorem 2.4 and Theorem 6.3 do not hold for Q-Res and Q-Res$_\mathsf{T}$.

As the above example illustrates, it is easy to enforce that universal variables are accumulated in a clause, thus leading to large width. Hence the following question naturally arises: can we obtain size-width or space-width relations by using the tighter measure of only counting existential variables?

This aligns with the situation in the expansion systems $\forall$Exp+Res and IR-calc, where clauses contain only existential variables. In this respect, it is worth noting that the above example indeed does not demonstrate the failure of the size-width relationship in expansion-based calculi. For instance, in $\forall$Exp+Res, a tree-like refutation could download the existential variables of axioms annotated with $u_i/0$ for $i \in [n]$, and generate the empty clause in $O(n)$ steps with width just 2 at the leaves and 1 at the internal nodes.

Thus, to get a consistent and interesting width measure for QBF calculi, we consider the notion of **existential width** that just counts the number of existential literals. This approach is justified also for Q-Res as the calculus can only resolve on existential variables, and rules out the easy counterexamples above. Formally, we define the existential width of a clause $C$ to be the number of existential literals in $C$, and denote it by $w_\exists(C)$. Using $w_\exists$ instead of $w$ everywhere, we obtain the existential width of a formula $w_\exists(F)$, of a proof $w_\exists(\pi)$, and of refuting a false sentence $w_\exists(\vdash_{\mathsf{S}} \mathcal{F})$.

For the expansion systems ∀Exp+Res and IR-calc the notions of existential width and width coincide. (In particular, distinct annotations of the same existential variable in a single clause are counted as distinct literals.) Hence we can drop the $\exists$ subscript in width of proofs in these systems. For the width of the sentence itself, there is still a difference between $w$ and $w_\exists$.

## 6.2 Negative Results: Size-width and Space-width Relations Fail in Q-Res

In this section we show that in the Q-Res proof system, even replacing width by existential width, the relations to size or space as in classical Resolution (Theorems 2.4 and 6.3) no longer hold for both tree-like and general proofs.

Firstly, we point out where the technique of [9] fails. A crucial ingredient of their proof is the following statement: if a clause $A$ can be derived from $F|_{x=1}$ in width $w$, then the clause $A \vee \neg x$ can be derived from $F$ in width $w+1$ (possibly using a weakening rule at the end). We show that the statement no longer holds in Q-Res.

**Proposition 6.5.** *There are false sentences $\psi_n$ of the form $\vec{\mathcal{Q}}\vec{w}\exists b.\ F_n$, with an existential literal $b$ quantified at the innermost level, such that the sentence $\psi_n|_{b=1} \equiv \vec{\mathcal{Q}}\vec{w}.\ F_n|_{b=1}$ is false and has a small existential-width proof, but $\psi_n$ itself needs large existential width to refute in* Q-Res.

*Proof.* The sentence $\psi_n$ is constructed by taking the conjunction of two sentences with distinct variables. The first sentence is a very simple one: $\exists a \forall u \exists b \quad (a \vee u \vee \bar{b}) \wedge (\bar{a})$. It is a true sentence, but if $b$ is set to 1, it becomes false. The second sentence is a false sentence of the form $\exists \vec{x} G_n(\vec{x})$, where $G_n$ is any unsatisfiable CNF formula over the $\vec{x}$ variables, such that $G_n$ needs large width in classical Resolution. One such example is the CNF formula described by Bonet and Galesi [26], that we denote as $BG_n$. $BG_n$ is an unsatisfiable 3-CNF formula over $O(n^2)$ variables with $w(\vdash_{\mathsf{Res}} BG_n) = \Omega(n)$. Now define $\psi_n$ as:

$$\exists \vec{x} \exists a \forall u \exists b \quad (a \vee u \vee \bar{b}) \wedge (\bar{a}) \wedge BG_n(\vec{x}).$$

Note that the clauses $(a \vee u \vee \bar{b}) \wedge (\bar{a})$ contain a contradiction if and only if $b = 1$. Thus $\psi_n|_{b=1}$ can be refuted with existential width 1 using just these two clauses: a $\forall$-Red on $(a \vee u)$ yields $a$ which can be resolved with $\bar{a}$. On the other hand, to refute $\psi_n$, the contradiction in $BG_n$ must be exposed. Since all the variables involved are existential, Q-Res degenerates to classical Resolution, requiring (existential) width $\Omega(n)$. □

The example in the proof of Proposition 6.5 can be made 'less degenerate' by interleaving more existential and universal variables disjoint from $\vec{x}$ and putting them in the first sentence. All we need is that $b$ is quantified existentially at the end, the first sentence is true as a whole but false if $b = 1$, and this latter sentence can be refuted in Q-Res with small existential width.

We now show that it is not just the technique of [9] that fails for Q-Res. No other technique will work either, because the relation from Theorem 2.4 between size and existential width itself fails to hold. The same example also shows that the relation from Theorem 6.3 between space and existential width also fails to hold. We first give an example where the relation for tree-like proofs fails.

**Theorem 6.6.** *There is a family of false QBF sentences $CR'_n$ over $O(n^2)$ variables, such that $S(\vdash_{\textsf{Q-Res}_T} CR'_n) = n^{O(1)}$, $w_\exists(CR'_n) = 3$, $CSpace(\vdash_{\textsf{Q-Res}_T} CR'_n) = O(1)$, and $w_\exists(\vdash_{\textsf{Q-Res}_T} CR'_n) = \Omega(n)$.*

*Proof.* Consider the formulas $CR_n$, introduced by Janota and Marques-Silva in [56]. Recall that we used the same formula in Chapter 3, for showing that level-ordered $Q$-resolution cannot simulate tree-like $Q$-resolution. We now define it once again for ease of reference.

$$CR_n = \exists x_{1,1} \ldots x_{n,n} \ \forall z \ \exists a_1 \ldots a_n \exists b_1 \ldots b_n.$$

$$\begin{array}{rll}
(C_{i,j}) & (x_{i,j} \vee z \vee a_i), & i,j \in [n] \\
(D_{i,j}) & (\bar{x}_{i,j} \vee \bar{z} \vee b_j), & i,j \in [n] \\
(A) & \bigvee_{i \in [n]} \bar{a}_i & \\
(B) & \bigvee_{i \in [n]} \bar{b}_i. &
\end{array}$$

We know from Chapter 3, that $CR_n$ has short proof in $\textsf{Q-Res}_\textsf{T}$. However $CR_n$ has large existential width, and in order to prove Theorem 6.6, we need a formula with constant initial existential width. To achieve this we proceed similarly as in the Tseitin transformations, i.e., we introduce $2n+2$ new existential variables (i.e, $\vec{y}, \vec{p}$) at the innermost level in $CR_n$, and replace the two large clauses in $CR_n$ by any CNF formula which preserves their satisfiability. Let $CR'_n$ denote the modified formula

$$CR'_n = \exists x_{1,1} \dots x_{n,n}\ \forall z\ \exists a_1 \dots a_n \exists b_1 \dots b_n \exists y_0 \dots y_n \exists p_0 \dots p_n.$$

$$(C_{i,j}) \qquad (x_{i,j} \vee z \vee a_i), \qquad i,j \in [n] \tag{6.1}$$

$$(D_{i,j}) \qquad (\bar{x}_{i,j} \vee \bar{z} \vee b_j), \qquad i,j \in [n] \tag{6.2}$$

$$\bar{y}_0 \wedge \bigwedge_{i \in [n]} (y_{i-1} \vee \bar{a}_i \vee \bar{y}_i) \wedge y_n \tag{6.3}$$

$$\bar{p}_0 \wedge \bigwedge_{i \in [n]} (p_{i-1} \vee \bar{b}_i \vee \bar{p}_i) \wedge p_n. \tag{6.4}$$

Note that $w_\exists(CR'_n) = 3$.

It is clear that from type-(6.3) clauses of $CR'_n$, we can derive the large clause $\bigwedge_{i \in [n]} \bar{a}_i$ of $CR_n$ in $n+1$ resolution steps. Similarly we can derive the large clause $\bigwedge_{i \in [n]} \bar{b}_i$ of $CR_n$ from the type-(6.4) clauses in $n+1$ steps. The proof refuting $CR_n$ uses each of these large clauses $n$ times ; see below. Thus $S(\vdash_{\textsf{Q-Res}_\textsf{T}} CR'_n) \leq S(\vdash_{\textsf{Q-Res}_\textsf{T}} CR_n) + O(n^2) = O(n^2)$.

We briefly sketch the refutation of $CR_n$ from Chapter 3, to analyse its space requirement. The fragment $W_j$ starts with clause $A$, successively resolves it with clauses from $C_{*,j}$ to get $z \vee x_{1,j} \vee \ldots \vee x_{n,j}$, eliminates $z$ through a $\forall$-reduction, then successively resolves it with clauses from $D_{*,j}$ to get $W_j = \bar{z} \vee b_j$. It is easy to see that $O(1)$ space suffices to construct this fragment. The overall proof starts with the clause $B$, successively resolves it with $W_1, W_2, \ldots, W_n$ (reusing the space to construct successive $W_j$'s), and finally gets $\bar{z}$ which is eliminated through a $\forall$-reduction. Again $O(1)$ space suffices.

Finally, we show that $CR'_n$ needs large existential width.

Let $\pi$ be a proof in Q-Res, $\pi \vdash_{\textsf{Q-Res}} CR'_n$. List the clauses of $\pi$ in sequence, $\pi = \{D_0, D_1, \ldots, D_s = \Box\}$, where each clause in the sequence is either a clause from $CR'_n$, or is derived from clause(s) preceding it in the sequence using resolution or

$\forall$-Red. There must be at least one universal reduction step in $\pi$, since all the initial clauses are necessary for refuting $CR'_n$, some of them contain universal variables, and the only way to remove a universal variable in Q-Res is by $\forall$-Red. Let $t$ be the least index such that in the clause $D_t$, a $\forall$-Red step has been performed on the only universal variable. Without loss of generality, let the universal literal be the positive literal $z$; the argument for $\bar{z}$ is identical. As the existential variables, $\vec{a}, \vec{b}, \vec{y}$, and $\vec{p}$ all block the universal variable $z$, none of them is present in the clause $D_t$. We use this fact to show that $w_{\exists}(D_t) = \Omega(n)$. Our strategy is to associate some set with each clause in $\pi$ in a specific way, and use the set size to bound existential width.

We associate the following sets with the literals of $CR'_n$ and the clauses of $\pi$.

$$\begin{array}{lrcl}
 & \sigma(z) & = & \emptyset = \sigma(\bar{z}) \\
\forall i \in [n] & \sigma(a_i) & = & [n] \setminus \{i\} = \{1, \ldots, n\} \setminus \{i\} \\
\forall i \in [n] & \sigma(x_{i,j}) = \sigma(\bar{a}_i) & = & \{i\} \\
\forall i \in [n] & \sigma(\bar{y}_i) & = & [n] \setminus [i] = \{i+1, \ldots, n\} \\
\forall i \in [n] & \sigma(y_i) & = & [i] = \{1, \ldots, i\} \\
\forall j \in [n] & \sigma(b_j) & = & [n] \setminus \{j\} = \{1, \ldots, n\} \setminus \{j\} \\
\forall j \in [n] & \sigma(\bar{x}_{i,j}) = \sigma(\bar{b}_j) & = & \{j\} \\
\forall j \in [n] & \sigma(\bar{p}_j) & = & [n] \setminus [j] = \{j+1, \ldots, n\} \\
\forall j \in [n] & \sigma(p_j) & = & [j] = \{1, \ldots, j\} \\
\forall D \in \pi & \sigma(D) & = & \bigcup\limits_{l \in D} \sigma(l).
\end{array}$$

Note that for variables $v$ in $\vec{a}$, $\vec{b}$, $\vec{p}$, $\vec{y}$, the sets $\sigma(v)$ and $\sigma(\bar{v})$ form a partition of $[n]$.

For $D \in \pi$, let $\pi_D$ be the sub-DAG of $\pi$, rooted at $D$. Consider the sub-DAG $\pi_{D_t}$ of $\pi$. We have the following observations:

**Observation 1.** $\pi_{D_t}$ contains at least one type-(6.1) clause as a source; this is because $z \in D_t$, and the only initial clauses containing $z$ are the type-(6.1) clauses.

**Observation 2.** $\pi_{D_t}$ does not contain any clause of type-(6.2) : as $z \in D_t$, we know that $\bar{z} \notin D_t$. Therefore if some type-(6.2) clause is present in this sub-DAG, the only way to remove $\bar{z}$ is via $\forall$-Red. This reduction will take place before the reduction on $D_t$, contradicting our choice of index $t$. We also conclude that the literal $\bar{z}$ cannot appear anywhere in $\pi_{D_t}$.

**Observation 3.** $\pi_{D_t}$ does not contain any type-(6.4) clause: we know that $D_t$ does not contain $\vec{p}$ and $\vec{b}$ variables (because they block $z$). Any use of type-(6.4) clauses introduces $\vec{p}$ variables and possibly $\bar{b}$ literals. Removing $\vec{p}$ variables introduces $\bar{b}$ literals. But $\bar{b}$ can be removed only by resolving with $b$, which is only in type-(6.2) clauses. We have already seen that type-(6.2) clauses are not present in $\pi_{D_t}$.

**Observation 4.** No clause in $\pi_{D_t}$ contains a literal $\bar{x}_{i,j}$, since $\bar{x}_{i,j}$ are introduced only in type-(6.2) clauses which were already ruled out.

**Observation 5.** For any clause $C$ derived solely from type-(6.3) clauses, $\sigma(C) = [n]$. This is true for type-(6.3) clauses by definition of $\sigma$. Using only these clauses, the only resolution step possible is with a $y$ variable as pivot. The claim can be verified by induction on depth: since $\sigma(y_i)$ and $\sigma(\bar{y}_i)$ partition $[n]$, $[n] \setminus \sigma(y_i)$ and $[n] \setminus \sigma(\bar{y}_i)$ also partition $[n]$.

We show that all clauses in $\pi_{D_t}$ that are descendants of some type-(6.1) clause, (i.e, all clauses in $\pi_{D_t}$ with a directed path to some type-(6.1) clause), have large sets associated with them. In particular, we show:

**Claim 6.7.** *Every clause $D$ in $\pi_{D_t}$ such that $\pi_D$ contains a type-(6.1) clause has $\sigma(D) = [n]$.*

Deferring the proof briefly, we continue with our argument. From the Claim we conclude that $\sigma(D_t) = [n]$. Recall that the variables $\vec{a}, \vec{b}, \vec{y}, \vec{p}$ and the literals $\bar{x}_{i,j}$'s are not present in $D_t$. The only literals left are positive $x_{i,j}$'s. These literals are associated with singleton sets, and the variables $x_{i,j}$ for different values of $j$ give the same singleton set. So we conclude that for each $i \in [n]$, there must be some $x_{i,j} \in D_t$. Hence $w_{\exists}(D_t) = \Omega(n)$.

It remains to establish the claimed set size.

*Proof of claim 6.7.* We proceed by induction on the depth of descendants of type-(6.1) clauses in $\pi_{D_t}$. The base case is a type-(6.1) clause itself and follows from the definition of $\sigma$.

For the inductive step, let $D$ be obtained by resolving $(E \vee r)$ and $(F \vee \bar{r})$. There are two cases to consider: both are descendants of some type-(6.1) clauses, or only one of them, say $(E \vee r)$, is a descendant of a type-(6.1) clause. In the former case, by the induction hypothesis, $\sigma(E \vee r) = [n]$ and $\sigma(F \vee \bar{r}) = [n]$. In the latter case, $\sigma(E \vee r) = [n]$ by induction hypothesis, and $\sigma(F \vee \bar{r}) = [n]$ from the observations above. ($(F \vee \bar{r})$ is not a descendant of any type-(6.1) clause. But it belongs to $\pi_{D_t}$ which has only type-(6.1) and type-(6.3) clauses. So it must be a descendant of only type-(6.3) clauses, and hence has $[n]$ associated with it.)

Thus in both cases, we have $\sigma(E \vee r) = \sigma(F \vee \bar{r}) = [n]$. So we have $\sigma(E) \supseteq [n] \setminus \sigma(r)$ and $\sigma(F) \supseteq [n] \setminus \sigma(\bar{r})$. Observe that the pivot variable $r$ can only be either an $\vec{a}$ or a $\vec{y}$ variable. Thus $\sigma(r)$ and $\sigma(\bar{r})$ are disjoint, and hence $\sigma(E) \cup \sigma(F) = [n]$. Thus $\sigma(D) = \sigma(E) \cup \sigma(F) = [n]$ as claimed. □

This completes the proof of the Theorem. □

Since tree-like space is at least as large as space (from definition), Theorem 6.6 also rules out the space-width relation for general dag-like Q-Res proofs. However, observe that Theorem 6.6 cannot be used to show that the size-existential-width relationship for general dag-like proofs fails in Q-Res, because the sentences $CR'_n$ have $O(n^2)$ variables. However, we show via another example that the relation fails to hold in Q-Res as well. This example cannot be used for proving Theorem 6.6 because it is known to be hard for $\mathsf{Q\text{-}Res_T}$ [56]. (In [56] the hardness for $\forall$Exp+Res is shown, which implies hardness for $\mathsf{Q\text{-}Res_T}$, as $\forall$Exp+Res p-simulates $\mathsf{Q\text{-}Res_T}$.)

**Theorem 6.8.** *There is a family of false QBFs $\phi'_n$ in $O(n)$ variables such that $S(\vdash_{\textsf{Q-Res}} \phi'_n) = n^{O(1)}$, $w_\exists(\phi'_n) = 3$, and $w_\exists(\vdash_{\textsf{Q-Res}} \phi'_n) = \Omega(n)$.*

*Proof.* Consider the formulas $\phi_n$, described in Chapter 2 (Proposition 2.12, 2.11). Recall that the formula $\phi_n$ has been used in [56] to show that $\forall$Exp+Res cannot simulate Q-Res. As discussed in Chapter 2, $\phi_n$ has a short Q-Res proof (Proposition 2.12), but are hard for $\forall$Exp+Res (Proposition 2.11). We now use the same formula to prove Theorem 6.8. For ease of reference we present the formula once again.

$$\begin{aligned}\phi_n = {} & \exists e_1 \forall u_1 \exists c_1 c_2 \dots \exists e_n \forall u_n \exists c_{2n-1} c_{2n}. \\ & \bigwedge_{i\in[n]} \big((\bar{e}_i \vee c_{2i-1}) \wedge (\bar{u}_i \vee c_{2i-1}) \wedge (e_i \vee c_{2i}) \wedge (u_i \vee c_{2i})\big) \wedge \\ & \big(\bigvee_{i\in[2n]} \bar{c}_i\big).\end{aligned}$$

Observe that $\phi_n$ has large initial existential width, However, in order to prove Theorem 6.8, we need a formula with constant initial width. To achieve this we consider quantified Tseitin transformations of $\phi_n$, i.e. we introduce $2n+1$ new existential variables $x_i$ at the innermost quantification level in $\phi_n$, and replace the only large clause in $\phi_n$ by any CNF formula that preserves satisfiability. Let $\phi'_n$ denote the modified formula:

$$\begin{aligned}\phi'_n &= \exists e_1 \forall u_1 \exists c_1 c_2 \dots \exists e_n \forall u_n \exists c_{2n-1} c_{2n} \exists x_0 \dots x_{2n}. \\ &\bigwedge_{i\in[n]} \big( (\bar{e}_i \vee c_{2i-1}) \wedge (\bar{u}_i \vee c_{2i-1}) \wedge (e_i \vee c_{2i}) \wedge (u_i \vee c_{2i}) \big) \wedge \qquad (6.5) \\ &\bar{x}_0 \wedge \bigwedge_{i\in[2n]} (x_{i-1} \vee \bar{c}_i \vee \bar{x}_i) \wedge x_{2n}. \qquad (6.6)\end{aligned}$$

Note that $w_\exists(\phi'_n) = 3$.

We refer to the clauses in (6.6) as $x$-clauses. It is clear that from the $x$-clauses, we can derive the large clause of $\phi_n$ in $2n + 1$ resolution steps and get back $\phi_n$. Thus $S(\mid_{\overline{\text{Q-Res}}} \phi'_n) \leq S(\mid_{\overline{\text{Q-Res}}} \phi_n) + 2n + 1 \in n^{O(1)}$.

We now show that $\phi'_n$ needs large existential width. We follow the same strategy used in proving Theorem 6.6.

Let $\pi$ be a proof in Q-Res, $\pi \mid_{\overline{\text{Q-Res}}} \phi'_n$. List the clauses of $\pi$ in sequence, $\pi = \{D_0, D_1, \dots, D_s = \Box\}$, where each clause in the sequence is either a clause from $\phi'_n$, or is derived from clause(s) preceding it in the sequence using resolution or ∀-Red. There must be at least one universal reduction step in $\pi$, since all the initial clauses are necessary for refuting $\phi'_n$, some of them contain universal variables, and the only way to remove a universal variable in Q-Res is by ∀-Red. Let $i$ be the least index such that the clause $D_i$ is obtained by ∀-Red on $D_j$ for some $0 < i$. Since all $x$ variables block all $u$ variables, $D_j$ and $D_i$ cannot contain any $x$ variables. We use this fact to show that $w_\exists(D_i) = \Omega(n)$. Our strategy is to associate some set with each clause in $\pi$ in a specific way, and use the set size to bound existential width.

We associate the following sets with the literals of $\phi'_n$ and the clauses of $\pi$.

$$\begin{array}{lrcl}
 & \sigma(x_0) & = & \emptyset \\
\forall i \in [2n] & \sigma(x_i) & = & [i] = \{1, 2, \dots, i\} \\
 & \sigma(\bar{x}_0) & = & [2n] \\
\forall i \in [2n] & \sigma(\bar{x}_i) & = & [2n] \setminus [i] = \{i+1, \dots, 2n\} \\
\forall i \in [n] & \sigma(e_i) = \sigma(u_i) = \sigma(\bar{c}_{2i}) = \sigma(c_{2i-1}) & = & \{2i\} \\
\forall i \in [n] & \sigma(\bar{e}_i) = \sigma(\bar{u}_i) = \sigma(\bar{c}_{2i-1}) = \sigma(c_{2i}) & = & \{2i-1\} \\
\forall D \in \pi & \sigma(D) & = & \bigcup_{l \in D} \sigma(l).
\end{array}$$

Note that for any literal $\ell$, $\sigma(\ell)$ and $\sigma(\bar{\ell})$ are disjoint.

For $D \in \pi$, let $\pi_D$ be the sub-DAG of $\pi$, rooted at $D$.

**Claim 6.9.** *$\pi_{D_i}$ contains at least one $x$-clause (axiom clause of type* (6.6)*).*

*Proof.* The child $D_j$ of node $D_i$ contains a universal variable which is then removed through ∀-Red to get $D_i$. The universal variables appear only in clauses of type (6.5), but are blocked by the $c$-variables in every clause where they appear. Thus, before a reduction is permitted, a $c$ variable must be eliminated by resolution. Since all $c$ variables appear only positively in type (6.5) clauses, some $x$-clause must be used in the resolution. □

We show that all clauses in $\pi_{D_i}$ that are descendants of some $x$-clause have large sets associated with them. In particular, we show:

**Claim 6.10.** *Every clause $D$ in $\pi_{D_i}$ such that $\pi_D$ contains an $x$-clause has $\sigma(D) = [2n]$.*

Deferring the proof briefly, we continue with our argument. From the Claim we conclude that $\sigma(D_i) = [2n]$. Recall that none of the $x$ variables belongs to $D_i$. All other literals are associated with singleton sets, so $D_i$ must contains at least $2n$

literals in order to be associated with the complete set $[2n]$. Since Q-Res proofs prohibit a variable and its negation in the same clause, at most $n$ of the literals in $D_i$ can be universal variables. Thus $D_i$ has at least $n$ existential literals, hence $w_{\exists}(D_i) = \Omega(n)$.

It remains to establish the claimed set size.

*Proof of claim 6.10.* We proceed by induction on the depth of descendants of $x$-clauses in $\pi_{D_i}$. The base case is an $x$-clause itself and follows from the definition of $\sigma$.

For the inductive step, let $D$ be obtained by resolving $(E \vee z)$ and $(F \vee \bar{z})$. There are two cases to consider:

**Case 1:** Both $(E \vee z)$ and $(F \vee \bar{z})$ are descendants of $x$-clauses (not necessarily the same $x$-clause). Then by induction, $\sigma(E \vee z) = \sigma(F \vee \bar{z}) = [2n]$. So $\sigma(E) \supseteq [2n] \setminus \sigma(z)$ and $\sigma(F) \supseteq [2n] \setminus \sigma(\bar{z})$. Since $\sigma(z)$ and $\sigma(\bar{z})$ are disjoint, $\sigma(E) \cup \sigma(F) = [2n]$. Thus $\sigma(D) = \sigma(E) \cup \sigma(F) = [2n]$ as claimed.

**Case 2:** Exactly one of $(E \vee z)$ and $(F \vee \bar{z})$ is a descendant of an $x$-clause. Without loss of generality, let $F \vee \bar{z}$ be the descendant. Then $E \vee z$ is either a type-(6.5) clause or is derived solely from type-(6.5) clauses using resolution. However, observe that the only clauses derivable solely from type-(6.5) clauses via resolution, without creating tautologies as mandated in Q-Res, are of the form $(c_{2i-1} \vee c_{2i})$ for some $i$. It follows that $z$ is not an $x$ variable. Hence $\sigma(z)$ and $\sigma(\bar{z})$ are distinct singleton sets. Further, $z$ cannot be a $u$ variable either, since resolution on universal variables is not permitted in Q-Res.

Now note that for any type-(6.5) clause $C$, $\sigma(C) = \{2i-1, 2i\}$ for the appropriate $i$. Similarly, $\sigma(c_{2i-1} \vee c_{2i}) = \{2i-1, 2i\}$. So if $E \vee z$ is one of these clauses, then $\sigma(E \vee z) = \sigma(z) \cup \sigma(\bar{z})$ and $\sigma(E) = \sigma(\bar{z})$. Further, as in Case 1, by induction we know that $\sigma(F \vee \bar{z}) = [2n]$ and $\sigma(F) \supseteq [2n] \setminus \sigma(\bar{z})$. Hence, $\sigma(E \vee F) = [2n]$ as claimed. □

This completes the proof of the theorem. □

The above counterexamples are provided by formulas that require small size, but large existential width. We will now illustrate via another example that also *large size and large width* can occur. Consider the formulas $\mathrm{QP\textsc{arity}}_n$ defined in Section 2.5. As already mentioned in Theorem 2.17, these formulas are hard for Q-Res. We now complement the exponential size lower bound from [13] by a width lower bound.

**Theorem 6.11.** *$w_\exists(\vdash_{\textsf{Q-Res}} \mathrm{QP\textsc{arity}}_n) \geq n$.*

*Proof.* Consider the formula $\mathrm{QP\textsc{arity}}_n$ from Section 2.5. Observe that in $\mathrm{QP\textsc{arity}}_n$, the contradiction occurs semantically because of the clauses $z \vee t_n$, $\neg z \vee \neg t_n$ asserting $z \neq t_n$ (along with the fact that the values of $x$ variables uniquely determine the values of all $t$ variables, in particular, $t_n$). Thus, at least one of these clauses must be used in any proof, necessitating a $\forall$-reduction. In Q-Res we cannot reduce $z$ while any of the $t$ variables are present; and due to the restrictions in Q-Res we cannot resolve any descendants of $z \vee t_n$ with any descendants of $\neg z \vee \neg t_n$ until there is at least one $\forall$-reduction.

Consider a smallest Q-Res proof, and assume without loss of generality that a first (lowest) $\forall$ reduction happens on the positive literal $z$. Therefore before this $\forall$-reduction step we have essentially a resolution derivation $\pi$ from $\Gamma = (t_2 \equiv x_1 \oplus x_2) \cup \bigcup_{i=3}^{n}(t_i \equiv x_i \oplus t_{i-1}) \cup \{t_n \vee z\}$. The clause $D$ that occurs in $\pi$ immediately before the $\forall$-reduction must only contain variables from $\{x_1, \ldots, x_n\}$ apart from the literal $z$, else the reduction is blocked.

We now use the following observation.

**Claim 6.12.** *Suppose $x_1 \oplus \cdots \oplus x_n \vDash C$ for some clause $C$. Then $C$ is either a tautology or $C$ contains all variables $x_1, \ldots, x_n$.*

Assuming Claim 6.12, we continue with our argument. Any assignment to the $x$ variables satisfying $x_1 \oplus \cdots \oplus x_n$ has a unique extension to $z$ and the $t$ variables satisfying all clauses of the formula $\text{QP}\textsc{arity}_n$. This extension necessarily has $t_n = x_1 \oplus \cdots \oplus x_n = 1$ and $z = 0$. Since it satisfies all axioms, by soundness of resolution, it also satisfies $D$.

This, along with Claim 6.12, implies that $D$ is either a tautology or has all $x$ variables. Since it cannot be a tautology (it appears in the proof, and besides, at the very least it has the variable $z$), it must have all $x$ variables, and hence has existential width $n$.

It remains to prove the Claim.

*Proof of Claim 6.12.* Suppose the clause $C$ is not a tautology, but the variables $x_i$, $i \in I \neq \emptyset$, do not appear in $C$. Since $C$ is a non-tautological clause, there is exactly one partial assignment $\alpha$ falsifying $C$. By setting the variables $x_i$, $i \in I$, appropriately, we can increase $\alpha$ to an assignment satisfying $x_1 \oplus \cdots \oplus x_n$, but still falsifying $C$. Hence $x_1 \oplus \cdots \oplus x_n \nvDash C$. □

This completes the proof of the Theorem. □

## 6.3 Simulations: Preserving Size, Width, and Space Across Calculi

After these strong negative results, ruling out size-width and space-width relations in Q-Res and $\text{Q-Res}_\text{T}$, we aim to determine whether any positive results hold in the expansion systems ∀Exp+Res and IR-calc. Before we can do this we need to relate the measures of size, width, and space across the three calculi Q-Res, ∀Exp+Res, IR-calc. Of course, such a comparison in terms of refined simulations is also interesting in its own as it determines the relative strength of the different proof systems. As size corresponds to running time, and space to memory consumption of QBF solvers, such a comparison yields interesting insights into the power of QBF solvers using CDCL vs. expansion techniques.

It is known that IR-calc p-simulates ∀Exp+Res and Q-Res [12], and that ∀Exp+Res p-simulates $\mathsf{Q\text{-}Res_T}$ [56]. We revisit these proofs, with special attention to the width parameter, and also obtain simulating proofs that are tree-like if the original proof is tree-like. The relationships we establish are stated in the following theorem:

**Theorem 6.13.** *For all false QBFs $\mathcal{F}$, the following relations hold:*

1. $\frac{1}{2}S(\vdash_{\mathsf{IR_T\text{-}calc}} \mathcal{F}) \leq S(\vdash_{\forall\mathsf{Exp+Res_T}} \mathcal{F}) \leq S(\vdash_{\mathsf{IR_T\text{-}calc}} \mathcal{F}) \leq 3S(\vdash_{\mathsf{Q\text{-}Res_T}} \mathcal{F})$.
2. $w(\vdash_{\mathsf{IR\text{-}calc}} \mathcal{F}) = w(\vdash_{\forall\mathsf{Exp+Res}} \mathcal{F}) \leq w_{\exists}(\vdash_{\mathsf{Q\text{-}Res}} \mathcal{F})$.
3. $\mathit{CSpace}(\vdash_{\forall\mathsf{Exp+Res_T}} \mathcal{F}) = \mathit{CSpace}(\vdash_{\mathsf{IR_T\text{-}calc}} \mathcal{F}) \leq \mathit{CSpace}(\vdash_{\mathsf{Q\text{-}Res_T}} \mathcal{F})$.

These results follow from Proposition 6.14 and Lemmas 6.15, 6.16 that are stated and established below.

**Proposition 6.14** ( [12]). *Any proof in ∀Exp+Res of size $S$, width $W$, and space $C$ can be efficiently converted into a proof in IR-calc of size at most $2S$, width $W$, and space $C$. If the proof in ∀Exp+Res is tree-like, so is the resulting IR-calc proof.*

*Proof.* In IR-calc, when an axiom is downloaded, the existential literals in it are annotated partially. However in ∀Exp+Res, the annotations are *complete*; all universal variables at a lower level than a literal appear in its annotation. To convert a proof $\pi$ in ∀Exp+Res to one in IR-calc, all that is needed is to follow up each axiom-download with an instantiation that completes the annotations as in $\pi$. This introduces at most one extra step per leaf but does not increase width. Also observe that the space required has not changed: to instantiate a clause we can reuse the same space. □

**Lemma 6.15.** *$\forall$Exp+Res$_\mathsf{T}$ p-simulates IR$_\mathsf{T}$-calc while preserving its width, size, and space.*

*Proof.* Recall the main reason why IR$_\mathsf{T}$-calc proofs differ from those in $\forall$Exp+Res$_\mathsf{T}$: axioms are downloaded with partial rather than complete annotations, and annotations can be extended at any stage by the inst operation.

The idea is to systematically transform an IR$_\mathsf{T}$-calc proof, proceeding downwards from the top where we have the empty clause, and modifying annotations as we go down, so that when all leaves have been modified the resulting proof is in fact an $\forall$Exp+Res$_\mathsf{T}$ proof. This crucially requires that we start with a tree-like proof; if the underlying graph is not a tree, we cannot always find a way of modifying the annotations that will work for all descendants.

Let $\pi$ be an IR$_\mathsf{T}$-calc proof of a false QBF $\mathcal{F}$. Without loss of generality, we can assume that every resolution node has, as parent, an instantiation node. (If it does not, we introduce the dummy $\mathsf{inst}(\emptyset, *)$ node between it and its parent.) Since the proof is tree-like, we can also collapse contiguous instantiation nodes into a single instantiation node. Thus, as we move down a path from the root, nodes are alternately instantiation and resolution nodes. We consider each resolution node and its parent instantiation node to be at the same level.

Starting from the top, which we call level zero, we transform $\pi$ to another proof $\pi'$ in IR$_\mathsf{T}$-calc maintaining the following invariants: after the $i^{th}$ step, all the instantiated clauses up to level $i$ are fully annotated and the instantiating assignments are complete. Thus the instantiation steps become redundant. This further implies that after the last level (when we reach the axiom farthest from the top), the resulting proof is in fact a $\forall$Exp+Res$_\mathsf{T}$ proof.

- **At Level** $0$**:** The node at this level must be a resolution producing the empty clause, followed by a dummy instantiation with the empty assignment. Thus the clauses at this level are already fully annotated, but the instantiating assignment is far from complete. Pick an arbitrary complete assignment, say $\sigma$, and instantiate the empty clause with $\sigma$. Clearly the invariants hold now.

- Assume that the invariants holds after processing all nodes at level $i-1$.

- **At Level $i$:** Let $D$ be an instantiated clause at level $i-1$, obtained by instantiating some clause $C$ by an assignment $\sigma$. That is, $D = \mathsf{inst}(C, \sigma)$. By the induction hypothesis, $D$ is fully annotated and $\sigma$ is complete. Let $C$ be obtained by resolving $E = (G \vee x^\tau)$ and $F = (H \vee \neg x^\tau)$. We need to make $E$ and $F$ fully annotated. Let $E = \mathsf{inst}(I, \beta_1)$ and $F = \mathsf{inst}(J, \beta_2)$ in $\pi$. Replace $E$ by $E' = \mathsf{inst}(I, \beta_1 \veebar \sigma)$ and $F$ by $F' = \mathsf{inst}(J, \beta_2 \veebar \sigma)$. As $\sigma$ is complete, both $\beta_1 \veebar \sigma$ and $\beta_2 \veebar \sigma$ are complete, and hence both $E'$ and $F'$ are fully annotated. The resolution step is now performed on $x^{\tau'}$, where $\tau' = \tau \veebar \sigma$ is the resulting annotation on $x$. It is easy to see that the resolvent of $E'$ and $F'$ is $D$, so the intermediate instantiation step going from $C$ to $D$ becomes redundant.

It is clear that the simulation preserves width. It also does not increase size: we may introduce dummy instantiation nodes to make the proof 'alternating', but after the transformation, all instantiations — dummy and actual — are eliminated completely. It is also clear that the simulation preserves the space needed, since the structure of the proof is preserved. □

The simulation in Lemma 6.15 exhibits an interesting phenomenon: while it shows that the tree-like versions of $\forall$Exp+Res and IR-calc are p-equivalent, it was shown in [13] that in the dag-like versions, IR-calc is exponentially stronger than $\forall$Exp+Res. Thus $\forall$Exp+Res and IR-calc provide a rare example in proof complexity of two systems that coincide in the tree-like model, but are separated in the dag-like model.

**Lemma 6.16.** *$\mathsf{IR}_\mathsf{T}$-calc p-simulates Q-$\mathsf{Res}_\mathsf{T}$ while preserving space and existential* *width exactly and size upto a factor of 3. That is,*

$$\begin{aligned} S(\vdash_{\textsf{IR}_\textsf{T}\textsf{-calc}} \mathcal{F}) &\leq 3S(\vdash_{\textsf{Q-Res}_\textsf{T}} \mathcal{F}), \\ CSpace(\vdash_{\textsf{IR}_\textsf{T}\textsf{-calc}} \mathcal{F}) &\leq CSpace(\vdash_{\textsf{Q-Res}_\textsf{T}} \mathcal{F}), \quad \textit{and} \\ w(\vdash_{\textsf{IR-calc}} \mathcal{F}) &\leq w_\exists(\vdash_{\textsf{Q-Res}} \mathcal{F}) \end{aligned}$$

*Proof.* We use the same simulation as given in [12]. This simulation was originally for dag-like proof systems, but here we check that it also works for tree-like systems, and we observe that space and existential width are preserved.

Let $C_1, \ldots, C_k$ be a $\textsf{Q-Res}_\textsf{T}$ proof. We translate the clauses into clauses $D_1, \ldots, D_k$, which will form the skeleton of a proof in IR-calc.

- For an axiom $C_i$ in $\textsf{Q-Res}_\textsf{T}$ we introduce the same clause $D_i$ by the axiom rule of IR-calc, i.e., we remove all universal variables and add annotations.
- If $C_i$ is obtained via $\forall$-reduction from $C_j$, then $D_i = D_j$; we make no change.
- Consider now the case that $C_i$ is derived by resolving $C_j$ and $C_k$ with pivot variable $x$. Then $D_j = x^\tau \vee K_j$ and $D_k = \bar{x}^\sigma \vee K_k$. It is shown in [12] that the annotations $\tau$ and $\sigma$ are not contradictory; in fact, no annotations in the two clauses are contradictory. So if we define $D'_j = \textsf{inst}(\sigma, D_j)$ and $D'_k = \textsf{inst}(\tau, D_k)$, then the annotations of $x$ in $D'_j$ and $\bar{x}$ in $D'_k$ match, and we can resolve on this literal. Define $D'_i$ as the resolvent of $D'_j$ and $D'_k$. We can perform a further instantiation to obtain $D_i = \textsf{inst}(\eta, D_i)$, where $\eta$ is the set of all assignments to universal variables appearing anywhere in $D'_i$. $D_i$ has no more literals than $C_i$. For details, see [12].

Note that to complete this skeleton into a proof, we only add instantiation rules. Thus, if the original proof was tree-like, so is the new proof. If the original proof has size $S$, the new proof has size at most $4S$, since each resolution may now be preceded by two instantiations and followed by one instantiation. However, this is an overcount, since we are counting two instantiations per edge, one from the parent and one from the child, and contiguous instantiations can be collapsed. That is, every instantiation following a resolution step can be merged with the instantiation preceding the next resolution and need not be counted separately. The only exception is at the root, where there is nothing to collapse it with. However, at the root, the instantiation itself is redundant and can be discarded. Thus we obtain a new proof of size at most $3S$.

Further, if the original proof had existential width $w$, then the new proof has width $w$ since each $D_i$ has at most (annotated versions of) the existential literals of $C_i$.

Regarding space, observe that simulating axiom download and ∀-Red do not require additional space. At the resolution step, the simulation first performs additional instantiations. But instantiation does not need additional space. So the space bound remains the same. □

As a by-product, these simulations enable us to give an easy and elementary proof of the simulation of $\mathsf{Q\text{-}Res_T}$ by $\forall\mathsf{Exp{+}Res}$, shown in [56] via a more involved argument.

**Corollary 6.17** (Janota, Marques-Silva [56]). *$\forall\mathsf{Exp{+}Res_T}$ p-simulates $\mathsf{Q\text{-}Res_T}$.*

*Proof.* By Lemma 6.15, $\forall\mathsf{Exp{+}Res_T}$ p-simulates $\mathsf{IR_T\text{-}calc}$, which in turn p-simulates $\mathsf{Q\text{-}Res_T}$ by Lemma 6.16. □

Using again the width lower bound for $\mathrm{QP{\small ARITY}}_n$ (Theorem 6.11) we can show that item 2 of Theorem 6.13 cannot be improved, i.e. we obtain an optimal width separation between $\mathsf{Q\text{-}Res}$ and $\forall\mathsf{Exp{+}Res}$.

**Theorem 6.18.** *There exist false QBFs $\psi_n$ with $w_\exists(\vdash_{\textsf{Q-Res}} \psi_n) = \Omega(n)$, but $w(\vdash_{\forall\textsf{Exp+Res}} \psi_n) = O(1)$.*

*Proof.* We use the $\text{QP}\textsc{arity}_n$ formulas, which by Theorem 6.11 require existential width $n$ in Q-Res. To get the separation it remains to show $w(\vdash_{\forall\textsf{Exp+Res}} \text{QP}\textsc{arity}_n) = O(1)$. For this we use the following ∀Exp+Res proofs of $\text{QP}\textsc{arity}_n$ from [13]: the formulas $\text{QP}\textsc{arity}_n$ have exactly one universal variable $z$, which we expand in both polarities 0 and 1. This does not affect the $x_i$ variables, but creates different copies $t_i^{z/0}$ and $t_i^{z/1}$ of the existential variables right of $z$. Using the clauses of $t_i \equiv x_i \oplus t_{i-1}$, we can inductively derive clauses representing $t_i^{z/0} = t_i^{z/1}$. This lets us derive a contradiction using the clauses $t_n^{z/0}$ and $\neg t_n^{z/1}$.

Clearly, this proof only contains clauses of constant width, giving the result. □

## 6.4 Positive Results: Size, Width, and Space in Tree-like QBF Calculi

We are now in a position to show some positive results on size-width and size-space relations for QBF Resolution calculi. However, most of these results only apply to the rather weak tree-like proof systems.

### 6.4.1 Relations in the Expansion Calculi ∀Exp+Res and IR-calc

We first observe that for ∀Exp+Res almost the full spectrum of relations from classical Resolution remains valid.

**Theorem 6.19.** *For all false QBFs $\mathcal{F}$, the following relations hold:*

1. $S(\mid\!\overline{_{\forall \mathsf{Exp+Res_T}}}\ \mathcal{F}) \geq 2^{(w(\mid\overline{_{\forall \mathsf{Exp+Res}}}\mathcal{F}))-w_\exists(\mathcal{F}))}$.
2. $S(\mid\!\overline{_{\forall \mathsf{Exp+Res_T}}}\ \mathcal{F}) \geq 2^{CSpace(\mid\overline{_{\forall \mathsf{Exp+Res_T}}}\mathcal{F})} - 1$.
3. $CSpace(\mid\!\overline{_{\forall \mathsf{Exp+Res_T}}}\ \mathcal{F}) \geq CSpace(\mid\!\overline{_{\forall \mathsf{Exp+Res}}}\,\mathcal{F}) \geq w(\mid\!\overline{_{\forall \mathsf{Exp+Res}}}\,\mathcal{F}) - w_\exists(\mathcal{F}) + 1$.

*Proof.* This theorem follows from the analogous statements for classical Resolution. We just describe how to apply those results to ∀Exp+Res.

We know that in $\forall\mathsf{Exp+Res_T}$ proofs, leaves corresponds to the expanded clauses from $\mathcal{F}$. The expanded clauses contain only existential (annotated) literals and no universal literals. Let $\mathcal{G}$ be the QBF obtained after expanding $\mathcal{F}$ based on all possible assignments of universal variables. Clearly, $\mathcal{G}$ contains no universal variables and hence can be treated as a propositional CNF formula (all variables are only existentially quantified). That is, if $G$ is the matrix of clauses in $\mathcal{G}$, then $\mathcal{G}$ asserts that $G$ is satisfiable. Also, $w(G) = w(\mathcal{G}) = w_\exists(\mathcal{F})$.

Refutations of $\mathcal{F}$ in ∀Exp+Res (respectively, $\forall\mathsf{Exp+Res_T}$) are precisely refutations (resp. tree-like refutations) of $G$ in classical resolution; the size, space and width are exactly the same, by definition. That is, $S(\mid\!\overline{_{\mathsf{Res_T}}}\,G) = S(\mid\!\overline{_{\forall\mathsf{Exp+Res_T}}}\,\mathcal{F})$, $w(\mid\!\overline{_{\mathsf{Res}}}\,G) = w(\mid\!\overline{_{\forall\mathsf{Exp+Res}}}\,\mathcal{F})$, $CSpace(\mid\!\overline{_{\mathsf{Res}}}\,G) = CSpace(\mid\!\overline{_{\forall\mathsf{Exp+Res}}}\,\mathcal{F})$, and $CSpace(\mid\!\overline{_{\mathsf{Res_T}}}\,G) = CSpace(\mid\!\overline{_{\forall\mathsf{Exp+Res_T}}}\,\mathcal{F})$. Now the Theorem follows by applying Theorems 2.4, 6.2, and 6.3, on $G$. □

By the equivalence of $\forall\mathsf{Exp+Res_T}$ and $\mathsf{IR_T}$-calc with respect to all the three measures size, width, and space (Theorem 6.13) we can immediately transfer all results from Theorem 6.19 to $\mathsf{IR_T}$-calc.

**Theorem 6.20.** *For all false QBFs $\mathcal{F}$, the following relations hold:*

1. $S(\mid\!\overline{_{\mathsf{IR_T\text{-}calc}}}\,\mathcal{F}) \geq 2^{(w(\mid\overline{_{\mathsf{IR\text{-}calc}}}\mathcal{F}))-w_\exists(\mathcal{F}))}$.

2. $S(\vdash_{\textsf{IR}_T\textsf{-calc}} \mathcal{F}) \geq 2^{CSpace(\vdash_{\textsf{IR}_T\textsf{-calc}} \mathcal{F})} - 1$.

3. $CSpace(\vdash_{\textsf{IR}_T\textsf{-calc}} \mathcal{F}) \geq w(\vdash_{\textsf{IR-calc}} \mathcal{F}) - w_\exists(\mathcal{F}) + 1$.

### 6.4.2 The Size-space Relation in Tree-like $Q$-resolution (Q-Res$_\mathsf{T}$)

We finally return to Q-Res. Most relations were already ruled out in Section 6.2 for both Q-Res and Q-Res$_\mathsf{T}$. The only relation that we can still show to hold is the classical size-space relation (Theorem 6.2), which we transfer from Res$_\mathsf{T}$ to Q-Res$_\mathsf{T}$.

In Resolution, this relationship was obtained using pebbling games [46]. We observe that the same holds for Q-Res$_\mathsf{T}$ as well, giving the analogous relationship. That is, we show:

**Theorem 6.21.** *For a false QBF sentence $\mathcal{F}$,*

$$S(\vdash_{\textsf{Q-Res}_T} \mathcal{F}) \geq 2^{CSpace\left(\vdash_{\textsf{Q-Res}_T} \mathcal{F}\right)} - 1.$$

Before getting into the proof, we describe the pebbling game.

**Definition 6.22.** *(Pebbling Game) Let $G = (V, E)$ be a connected directed acyclic graph with a unique sink $s$, where every vertex of $G$ has fan-in at most 2. The aim of the game is to put a pebble on the sink of the graph following this set of rules:*

1. *A pebble can be placed on any source vertex, that is, on a vertex with no predecessors.*
2. *A pebble can be removed from any vertex.*
3. *A pebble can be placed on an internal vertex provided all of its children are pebbled. In this case, instead of placing a new pebble on it, one can shift a pebble from a child to the vertex.*

*The minimum number of pebbles needed to pebble the unique sink following the above rules is said to be the* **pebbling number** *of $G$.*

Consider the proof graph $G_\pi$ corresponding to a Q-Res proof $\pi$ of a false QBF $\mathcal{F}$. In $G_\pi$ clauses are the vertices and edges go from the hypotheses to the conclusion of inference rules (i.e, $\forall$-Red, resolution steps). Clearly $G_\pi$ is a DAG with initial clauses as sources and the empty clause as the unique sink. Also the in-degree of each vertex in $G_\pi$ is at most 2. Hence the pebbling game is well defined in $G_\pi$.

We now show that the space required to refute a false QBF sentence $\mathcal{F}$ (as per Definition 6.1) coincides with the minimum number of pebbles needed to play the pebble game on the graph of a Q-Res proof of $\mathcal{F}$. The relation holds for tree-like proofs as well.

**Lemma 6.23.** *Let $\mathcal{F}$ be a false QBF in prenex form. Then the following holds:*

1. $CSpace(\vdash_{\textsf{Q-Res}} \mathcal{F}) = \min \left\{ k : \begin{array}{l} \exists \text{ Q-Res } proof\ \pi\ of\ \mathcal{F},\ G_\pi\ can\ be\ pebbled\ with\ k \\ pebbles \end{array} \right\}$;
2. $CSpace(\vdash_{\textsf{Q-Res}_T} \mathcal{F}) = \min \left\{ k : \begin{array}{l} \exists \text{ Q-Res}_T\ proof\ \pi\ of\ \mathcal{F},\ G_\pi\ can\ be\ pebbled\ with\ k \\ pebbles \end{array} \right\}$.

*Proof Sketch.* The proof is exactly the same as in classical Resolution.

Let $\pi$ be a Q-Res proof whose proof graph $G_\pi$ can be pebbled with $k$ pebbles. (If $\pi$ is treelike, then $G_\pi$ is a tree.) Note that the vertices of $G_\pi$ are clauses in the proof. The space-oriented Q-Res (respectively Q-Res$_\mathsf{T}$) proof sequence with clause space $k$ is constructed by maintaining at each stage exactly the pebbled clauses. By the rules of the pebbling game, adding a clause to the current set is valid because the added clause is either at a source node and hence an axiom, or it has all predecessors pebbled and hence can be inferred. Further, if $\pi$ is tree-like, then it can be shown that there is a $k$-pebble sequence where no node is pebbled more than once (once a node is pebbled, no predecessor of the node need be pebbled again). So the above construction will yield a tree-like space-$k$ proof sequence.

In the other direction, given a space-$k$ proof as a sequence $\sigma$, we can construct a corresponding DAG $G$ with nodes for each clause appearing anywhere in $\sigma$, and edges reflecting how the clauses are used for inference in $\sigma$. Thus we obtain a proof $\pi$ with $G_\pi = G$ (it is the same proof as $\sigma$, just represented differently). We can pebble $G$ with $k$ pebbles by maintaining the invariant that at each stage, pebbles are placed on exactly the clauses present in the corresponding formula in the sequence $\sigma$. If $\sigma$ is a tree-like space-$k$ proof, we construct a corresponding tree with a distinct node for every copy of a clause introduced at some stage in $\sigma$, and then pebble it as above. We omit the details. □

We can now prove Theorem 6.21.

*Proof of Theorem 6.21.* This proof too is almost identical to the proof for classical Resolution [46]. We give a brief sketch.

Let $S(\vdash_{\overline{\mathsf{Q\text{-}Res_T}}} \mathcal{F}) = s$. Consider a tree-like Q-Res (Q-Res$_\mathsf{T}$) proof $\pi$ of $\mathcal{F}$ (i.e, $\pi \vdash_{\overline{\mathsf{Q\text{-}Res_T}}} \mathcal{F}$), of size $s$, and let $T$ be the underlying proof-tree.

In contrast to classical Resolution, a proof graph in Q-Res may have unary nodes corresponding to $\forall$-reductions. In particular, for a proof in Q-Res$_\mathsf{T}$, there may be paths corresponding to series of $\forall$-reductions. Once the lower end of such a path is pebbled, the same pebble can be slid up to the top of the path; no additional pebbles are needed. So without loss of generality we work with the tree $T'$ obtained by shortcutting all paths containing unary nodes.

Let $d_c(T)$ be the depth of the biggest complete binary tree that can be embedded in $T'$ or in $T$. (We say that a graph $G_1$ is embeddable in a graph $G_2$ if a graph isomorphic to $G_2$ can be obtained from $G_1$ by adding vertices and edges or subdividing edges of $G_1$.) Clearly, $2^{d_c(T)+1} - 1 \leq s$.

By induction on $|T'|$, we can show that $d_c(T)+1$ pebbles suffice to pebble $T'$. Hence, by the argument given above, $d_c(T)+1$ pebbles suffice to pebble $T$ as well. Now, using Lemma 6.23, we obtain $\mathit{CSpace}(\vdash_{\textsf{Q-Res}_\textsf{T}} \mathcal{F}) \leq d_c(T)+1$. Hence

$$2^{\mathit{CSpace}(\vdash_{\textsf{Q-Res}_\textsf{T}} \mathcal{F})} - 1 \leq 2^{d_c(T)+1} - 1 \leq s = S(\vdash_{\textsf{Q-Res}_\textsf{T}} \mathcal{F})$$

as claimed. □

# Chapter 7

# Conclusions and Open Problems

Beyersdorff et al. in [11], introduce a general method of transforming a propositional proof system $P$ into a QBF proof system P+∀red. In [11] they concentrate on exploring QBF proof systems $\mathcal{C}$-Frege+∀red based on restricted versions of Frege proof systems. Then Beyersdorff and Pich in [20] explore EF+∀red proof systems and prove some interesting results. In this thesis, we have introduced a new complete and sound QBF proof system CP+∀red based on Cutting Planes proof system (Section 4.1), and gave a comprehensive analysis of its proof complexity. As already stated, such proof systems have not been explored before. We point here that still, QBF proof systems based on algebraic propositional proof systems (for example, Hilbert's Nullstellensatz, Polynomial Calculus-PC, Polynomial Calculus with Resolution-PCR etc. are missing. (For their definitions, we recommend the survey [7]). It will be nice to introduce such QBF proof systems and give a detailed analysis of their proof complexities.

In Chapter 5, we have established *feasible interpolation* technique for all CDCL-based QBF Resolution calculi. In fact it has been shown in [14], that the technique also applies to all expansion-based QBF proof systems. Using the ideas from [67], we also established the technique for our new proof systems CP+∀red and semCP+∀red.

In propositional case, it is known that the feasible interpolation technique applies to some algebraic proof systems as well, for example Nullstellensatz, and PC [69]. It would be nice to lift the technique for QBF proof systems based on these systems (once introduced).

In Chapter 6, we show that the success story of width in Resolution needs to be rethought when moving to QBF. Indeed, the question arises: is width a central parameter in QBF Resolution? Is there another parameter that plays a similar role as classical width for understanding size and space for QBF Resolution ?

Our findings almost completely uncover the picture for size, space, and width for the most basic and arguably most important QBF Resolution systems Q-Res, $\forall$Exp+Res, and IR-calc. The most immediate open question arising from our investigation is whether size-width relations hold for general dag-like $\forall$Exp+Res or IR-calc proofs. The issue here is that in the classical size-width relation of [9] the number of variables enters the formula in a crucial way. For the instantiation calculi it is not clear what should qualify as the right count for this as different annotations of the same existential variable are formally treated as distinct variables (which is also clearly justified by the semantic meaning of expansions).

For further research it will also be interesting to settle whether size-width or space-width relations apply to any of the stronger QBF Resolution systems QU-Res [78], LD-Q-Res [4], or IRM-calc [12]. However, we conjecture that the negative picture also prevails for these systems.

# Bibliography

[1] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of Boolean functions. *Combinatorica*, 7(1):1–22, 1987.

[2] Sanjeev Arora and Boaz Barak. *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009.

[3] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, 2008.

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

[5] Valeriy Balabanov, Jie-Hong Roland Jiang, Mikolas Janota, and Magdalena Widl. Efficient extraction of QBF (counter) models from long-distance resolution proofs. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, pages 3694–3701, 2015.

[6] Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In *Proceeding of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 154–169, 2014.

[7] Paul Beame. Notes on proof complexity. Lectures by Paul Beame, scribed by Ashish Sabharwal, `http://www.cs.cornell.edu/~sabhar/publications/iaspcmi-proofcomplexity00.pdf`, 2000.

[8] Eli Ben-Sasson. Size space tradeoffs for resolution. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 457–464, 2002.

[9] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.

[10] Marco Benedetti. Evaluating QBFs via symbolic skolemization. In *Proceedings of the 11th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR)*, pages 285–300, 2004.

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

[12] 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.

[13] Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. Proof complexity of resolution-based QBF calculi. In *Proceedings of the 32nd International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 76–89. LIPIcs, 2015.

[14] Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Feasible interpolation for QBF resolution calculi. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 180–192. Springer, 2015.

[15] Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Are short proofs narrow? QBF resolution is not simple. In *Proceedings of the 33rd Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 15:1–15:14, 2016.

[16] Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Understanding Cutting Planes for QBFs. In *Proceedings of the 36th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 40:1–40:15, 2016.

[17] Olaf Beyersdorff, Leroy Chew, and Karteek Sreenivasaiah. A game characterisation of tree-like Q-resolution size. In *Proceeding of the 9th International Conference on Language and Automata Theory and Applications (LATA)*, pages 486–498. Springer, 2015.

[18] Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. A characterization of tree-like resolution size. *Information Processing Letters*, 113(18):666–671, 2013.

[19] Olaf Beyersdorff and Oliver Kullmann. Unified characterisations of resolution hardness measures. In *Proceeding of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 170–187, 2014.

[20] Olaf Beyersdorff and Ján Pich. Understanding Gentzen and Frege systems for QBF. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2016.

[21] Armin Biere. Resolve and expand. In *Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, 2004.

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

[23] Ilario Bonacina. *Space in Weak Propositional Proof Systems.* PhD thesis, Sapienza University of Rome, Faculty of Engineering, Computer Science and Statistics, 2015.

[24] Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldà, Alexis Maciel, and Toniann Pitassi. Non-automatizability of bounded-depth Frege proofs. *Computational Complexity*, 13(1–2):47–68, 2004.

[25] Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM Journal on Computing*, 30(5):1462–1484, 2000.

[26] Maria Luisa Bonet and Nicola Galesi. A study of proof search algorithms for resolution and polynomial calculus. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 422–432, 1999.

[27] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. On interpolation and automatization for Frege systems. *SIAM Journal on Computing*, 29(6):1939–1967, 2000.

[28] Joshua Buresh-Oppenheim and Toniann Pitassi. The complexity of resolution refinements. *Journal of Symbolic Logic*, 72(4):1336–1352, 2007.

[29] Sam Buss. Quasipolynomial size proofs of the propositional pigeonhole principle. *Theoretical Computer Science*, 576:77–84, 2015.

[30] Samuel R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52(4):916–927, 1987.

[31] Marco Cadoli, Andrea Giovanardi, and Marco Schaerf. An algorithm to evaluate Quantified Boolean Formulae. In *Proceedings of the 15th National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference (AAAI)*, pages 262–267, 1998.

[32] Marco Cadoli, Marco Schaerf, Andrea Giovanardi, and Massimo Giovanardi. An algorithm to evaluate Quantified Boolean Formulae and its experimental evaluation. *Journal of Automated Reasoning*, 28(2):101–142, 2002.

[33] Hubie Chen. Proof complexity modulo the polynomial hierarchy: Understanding alternation as a source of hardness. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP'16)*, pages 94:1–94:14, 2016.

[34] Leroy chew. *Proof Complexity for Quantified Boolean Formulas.* PhD thesis, School of Computing, University of Leeds, United Kingdom, 2017.

[35] Vasek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, 1988.

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

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

[38] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms (3. ed.).* MIT Press, 2009.

[39] William Craig. Linear reasoning. A new form of the herbrand-gentzen theorem. *Journal of Symbolic Logic*, 22(3):250–268, 1957.

[40] William Craig. Three uses of the herbrand-gentzen theorem in relating model theory and proof theory. *Journal of Symbolic Logic*, 22(3):269–285, 1957.

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

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

[43] Uwe Egly. On sequent systems and resolution for QBFs. In *Proceeding of the 15th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 100–113, 2012.

[44] Uwe Egly. On stronger calculi for QBFs. In *Proceeding of the 19th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 419–434, 2016.

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

[46] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001.

[47] Yuval Filmus, Pavel Hrubes, and Massimo Lauria. Semantic versus syntactic cutting planes. In *Proceedings of the 33rd Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 35:1–35:13, 2016.

[48] Yuval Filmus, Massimo Lauria, Mladen Miksa, Jakob Nordström, and Marc Vinyals. From small space to small width in resolution. In *Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 300–311, 2014.

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

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

[51] Andreas Goerdt. Cutting plane versus Frege proof systems. In *Proceedings of the 4th Workshop on Computer Science Logic (CSL)*, pages 174–194, 1990.

[52] Alexandra Goultiaeva, Allen Van Gelder, and Fahiem Bacchus. A uniform approach for generating proofs and strategies for both true and false QBF formulas. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 546–553, 2011.

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

[54] Pavel Hrubeš. On lengths of proofs in non-classical logics. *Annals of Pure and Applied Logic*, 157(2–3):194–205, 2009.

[55] Mikolás Janota, William Klieber, João Marques-Silva, and Edmund M. Clarke. Solving QBF with counterexample guided refinement. In *Proceeding of the 15th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 114–128, 2012.

[56] Mikolás Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theoretical Computer Science*, 577:25–42, 2015.

[57] Mikoláš Janota and Joao Marques-Silva. $\forall\mathsf{Exp{+}Res}$ does not p-simulate $Q$-resolution. In *International Workshop on Quantified Boolean Formulas*, pages 17–21, 2013. `http://fmv.jku.at/qbf2013/reportQBFWS13.pdf`.

[58] Mikoláš Janota and Joao Marques-Silva. On propositional $QBF$ expansions and $Q$-resolution. In *Proceedings of the 16th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 7962, pages 67–82, 2013.

[59] Jan Johannsen. Exponential incomparability of tree-like and ordered resolution. Manuscript, 2001.

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

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

[62] Jan Krajíček. Interpolation theorems, lower bounds for proof systems and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997.

[63] Jan Krajíček and Pavel Pudlák. Some consequences of cryptographical conjectures for $S^1_2$ and $EF$. *Information and Computation*, 140(1):82–94, 1998.

[64] Oliver Kullmann. Investigating a general hierarchy of polynomially decidable classes of CNF's based on short tree-like resolution proofs. *Electronic Colloquium on Computational Complexity (ECCC)*, 99(41), 1999.

[65] Meena Mahajan and Anil Shukla. Level-ordered $Q$-resolution and tree-like $Q$-resolution are incomparable. *Information Processing Letters*, 116(3):256–258, 2016.

[66] Christos H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[67] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–998, 1997.

[68] Pavel Pudlák. Proofs as games. *American Mathematical Monthly*, 107(6):541–550, 2000.

[69] Pavel Pudlák and Jiri Sgall. Algebraic models of computation and interpolation for algebraic proof systems. In P. W. Beame and S. R. Buss, editors, *Proof Complexity and Feasible Arithmetic*, volume 39 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 279–296. American Mathematical Society, 1998.

[70] Nicolas Rachinsky. The complexity of resolution refinements and satisfiablity algorithms. Master's thesis, Ludwig-Maximilians-Universität (LMU) M*ü*nchen, 2007.

[71] A. A. Razborov. Lower bounds on the size of bounded depth networks over a complete basis with logical addition. *Matematicheskie Zametki*, 41:598–607, 1987. In Russian. English translation in *Mathematical Notes of the Academy of Sciences of the USSR* 41:333–338, 1987.

[72] Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

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

[74] Ashish Sabharwal. *Algorithmic Applications of Propositional Proof Complexity.* PhD thesis, University of Washington, 2005.

[75] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the 19th Symposium on Theory of Computing (STOC)*, pages 77–82. ACM Press, 1987.

[76] G. S. Tseitin. On the complexity of derivations in propositional calculus. In A. O. Slisenko, editor, *Studies in Mathematics and Mathematical Logic, Part II*, pages 115–125. Springer-Verlag, Berlin Heidelberg, 1970.

[77] Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, 1987.

[78] 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.

[79] Heribert Vollmer. *Introduction to Circuit Complexity – A Uniform Approach.* Texts in Theoretical Computer Science. Springer Verlag, Berlin Heidelberg, 1999.

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