Reasons for Hardness in QBF Proof Complexity

by

Luke Peter Edward Hinde

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy

The University of Leeds
School of Computing
July 2019

Declarations

The candidate confirms that the work submitted is his/her/their own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Some parts of the work presented in this thesis have been published in the following articles:

Publications in Journals
List of Publications

1. O. Beyersdorff, J. Blinkhorn & L. Hinde. Size, cost and capacity: A semantic technique for hard random QBFs, In *Logical Methods in Computer Science, Vol. 15, 2019*
2. O. Beyersdorff & L. Hinde. Characterising tree-like Frege proofs for QBF, To appear in *Information and Computation*

Refereed Contributions in Conference Proceedings
List of Publications

3. O. Beyersdorff, L. Hinde & J. Pich. Reasons for hardness in QBF proof systems, In *Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), 2017*
4. O. Beyersdorff, J. Blinkhorn & L. Hinde. Size, cost and capacity: A semantic technique for hard random QBFs, In *Conference on Innovations in Theoretical Computer Science (ITCS), 2018*

The candidate confirms that the roles of the authors in above jointly-authored publications are as follows:

- Chapters 4, 5 and 6 contain work from [3]. I was the main author. The proofs were a result of discussion involving all authors.
- Chapter 7 contains work from [2]. I was the main author; the contribution of the other author was that of a primary supervisor.
- Chapters 8 and 9 contain work from [1,4] ([4] is the conference version of [1]). Blinkhorn and I were the main authors. I contributed the results in Sections 6,7 and 8 of [1]. The main result of Section 5 is included in the thesis with an alternative proof to that given in these publications.

Acknowledgements

First and foremost, I would like to thank my primary supervisor Olaf Beyersdorff. Olaf first introduced me to proof complexity and his advice and supervision has been invaluable throughout the last four years. He has pointed me in the direction of useful results, helped me ensure the rigour of my work, and given excellent feedback on my writing. I would not have been able to complete this thesis without this supervision. Thanks are also due to my second supervisor Kristina Vušković, in particular for her detailed advice on writing this thesis.

I must also express my gratitude to Ján Pich and Joshua Blinkhorn, with whom I have had the pleasure of co-authoring papers during my time as a PhD student. Much of the work presented here was inspired and improved as a result of stimulating conversations with Ján and Joshua. I have also benefitted greatly from sharing an office with Josh, Leroy Chew, Sarah Sigley and Judith Clymo, who were also working on proof complexity and often helped answer any questions I had. I was also lucky enough to learn from the experience of Meena Mahajan and other proof complexity researchers, both on their visits to Leeds and elsewhere.

The staff of the School of Computing have provided excellent support throughout, especially Judi Drew, who ensured that any travel arrangements, expenses and so on were handled with ease, and Sam Wilson, who encouraged me to be involved with teaching and proofread my thesis draft. I would also like to thank the PhD student communities in both Maths and Computing, particularly Richard, John, Richard, Fabio, Noleen, Bjarki, Polly and Jake, as well everyone at the Leeds Universities Catholic Chaplaincy, for their community and friendship throughout.

My parents and siblings have always been a source of inspiration and encouragement; I cannot thank them enough for this and for their support, financial and otherwise, to get to this point. Last but certainly not least I must thank Carolyn, whose love and companionship have seen me through even the most difficult parts of the last four years, all while she has been working on her own PhD.

Abstract

Quantified Boolean Formulas (QBF) extend the canonical **NP**-complete satisfiability problem by including Boolean quantifiers. Determining the truth of a QBF is **PSPACE**-complete; this is expected to be a harder problem than satisfiability, and hence QBF solving has much wider applications in practice. QBF proof complexity forms the theoretical basis for understanding QBF solving, as well as providing insights into more general complexity theory, but is less well understood than propositional proof complexity.

We begin this thesis by looking at the reasons underlying QBF hardness, and in particular when the hardness is propositional in nature, rather than arising due to the quantifiers. We introduce relaxing QU-Res, a previous model for identifying such propositional hardness, and construct an example where relaxing QU-Res is unsuccessful in this regard. We then provide a new model for identifying such hardness which we prove captures this concept.

Now equipped with a means of identifying 'genuine' QBF hardness, we prove a new lower bound technique for tree-like QBF proof systems. Lower bounds using this technique allows us to show a new separation between tree-like and dag-like systems. We give a characterisation of lower bounds for a large class of tree-like proof systems, in which such lower bounds play a prominent role.

Further to the tree-like bound, we provide a new lower bound technique for QBF proof systems in general. This technique has some similarities to the above technique for tree-like systems, but requires some refinement to provide bounds for dag-like systems. We give applications of this new technique by proving lower bounds across several systems. The first such lower bounds are for a very simple family of QBFs. We then provide a construction to combine false QBFs to give formulas for which we can show lower bounds in this way, allowing the generation of the first random QBF proof complexity lower bounds.

Table of Contents

List of Figures

List of Notation

$\mathbf{AC}^i[p]$	$\mathbf{AC}^i$ circuits in which $\mathsf{mod}\, p$ gates are permitted
$\mathrm{assign}(C)$	smallest assignment falsifying C
$\mathrm{clause}(\alpha)$	largest clause falsified by α
DEQ_n	QBFs with large strategy size
$\mathsf{dom}(\alpha)$	the domain of the assignment α
$\mathrm{EQ}(n)$	nth equality formula
KBKF_n	the nth formula of Kleine Büning et al.
KBKFd_n	the nth formula of Kleine Büning et al. with doubled universal variables
$\langle X \rangle$	the set of all possible assignments to the variables in X
lv	the level of a variable or formula in a quantifier prefix
$\mathbb{Q}$	the rational numbers
$\mathbb{N}$	the set of natural numbers
$\neg$	logical negation
$\oplus$	logical xor
$\otimes$	product formula
$\phi[\alpha]$	the substitution of the assignment α into ϕ
PHP_n	the nth pigeonhole principle formula
Π_k^b	containing k quantifier blocks with the leftmost block universally quantified
Π_k^{p}	class of problems reducible to a QBF with a Π_k^b-prefix
$\prec$	partial order on lines of a proof given by structure of deductions
$\mathrm{QPARITY}_n$	QBFs for which the winning strategy computes parity
$\mathsf{rng}(f)$	the range of the function f
$\rho(\Phi)$	strategy size of Φ
$\rightarrow$	logical implication
Σ_k^b	containing k quantifier blocks with the leftmost block existentially quantified
Σ_k^{p}	class of problems reducible to a QBF with a Σ_k^b-prefix
$\mathcal{U}$	the set of universal variables in a QBF prefix
$\mathrm{var}(\phi)$	the set of variables in the formula ϕ
$\vee$	logical disjunction
$\wedge$	logical conjunction
$\mathcal{X}$	the set of existential variables in a QBF prefix

$\mathbb{Z}$	the set of integers
$\mathbb{Z}_p$	the ring of integers modulo p
$H(\Phi, \Pi_k^b)$	set of axioms for relaxing QU-Res
L_i^α	root at ith round under α in round-based strategy extraction
p_α	path through a proof defined by α
$Q(n, m, c)$	a class of random QBFs
$\{0,1\}^*$	the set of all finite bit strings
$\mathbf{AC}_k^0$	$\mathbf{AC}^0$ circuits of depth at most k
$\mathbf{AC}^i$	the class of Boolean circuits with unbounded in-degree and depth $O(\log^i n)$
$\mathbf{coNP}$	class of languages whose complement is in $\mathbf{NP}$
CP	Cutting Planes proof system
eFrege	extended Frege; the $\mathbf{P/poly}$-Frege proof system
$\forall$Exp+Res	universal expansion and Resolution
Frege	the $\mathbf{NC}^1$-Frege proof system
$\mathcal{C}$-Frege	a Frege proof system with lines from the circuit class $\mathcal{C}$
IPS	Ideal Proof System
IR-calc	instantiation and resolution calculus
IRM-calc	instantiation, resolution and merging calculus
LD-Q-Res	long distance Q-Resolution
L-IPS	line-IPS
LQU$^+$-Res	long distance QU-Resolution
$\mathbf{NC}^i$	the class of Boolean circuits with in-degree 2 and depth $O(\log^i n)$
$\mathbf{NP}$	nondeterministic polynomial-time
PC	Polynomial Calculus
PCR	Polynomial Calculus with Resolution
$\mathbf{P}$	polynomial-time
$\mathbf{PH}$	the polynomial hierarchy
$\mathbf{P/poly}$	the class of functions with polynomial-size Boolean circuits
P+$\forall$red	the proof system P with $\forall$-reduction
$\mathbf{PSPACE}$	polynomial space
Σ_k^p-QU-Res	QU-Resolution with a Σ_k^p-oracle
$\mathbf{TC}^i$	$\mathbf{AC}^i$ circuits in which threshold gates are permitted
CDCL	conflict-driven clause learning
CEGAR	counterxample guided abstraction and refinement
CNF	Conjunctive Normal Form
dag	directed acyclic graph
DNF	disjunctive normal form
QBF	quantified Boolean formula
SAT	the language of satisfiable CNFs

Chapter 1

Introduction

We begin by discussing some general background on proof complexity, both for propositional logic and for quantified Boolean formulas. We also give a general overview of the contributions of this thesis, and outline how these are organised in the following chapters.

1.1 Background

SAT solving Given any computational task or problem, it is a natural question to ask: How efficiently can this problem be solved? The purpose of computational complexity is to provide a formal and theoretical answer to this question by considering the running time or the memory usage of algorithms solving the problem.

Perhaps the most famous and well studied such problem is the *satisfiability* (SAT) problem, of determining whether or not there is an assignment which satisfies a given Boolean formula. SAT is the canonical $\mathbf{NP}$-complete problem – all problems solvable in non-deterministic polynomial time can be efficiently translated to a SAT problem [42]. SAT solvers can therefore be applied to many other problems in $\mathbf{NP}$ such as bounded model checking [32] and some bounded planning problems [71]. The wide variety of applications has led to a great deal of development in SAT solving recently. Modern solvers such as MiniSat [51] and Glucose [7], Lingeling [31], and MapleCOMSPS and its derivatives [81, 84] compete on and regularly solve instances containing millions of variables [85].

Propositional proof complexity Given a SAT solver, one can view the run of the solver on an unsatisfiable instance as a *proof* that the instance is unsatisfiable. *Proof complexity* formalises the definition of a proof and a *proof system*, and has as its main focus the size of proofs; propositional proof complexity is concerned with proofs of the (un)satisfiability of Boolean formulas. There is a natural correspondence between propositional proof complexity and SAT solving [90]. By studying a proof system which corresponds to the running of a SAT solver, proof complexity helps to better understand the advantages and limitations of the solver, as well as point towards potential techniques worth exploring to improve such solvers.

The most studied propositional proof system is *Resolution*, a relatively simple proof system in which every line is a disjunction of literals and which contains only one inference rule for deriving new formulas. Much effort has been put into understanding Resolution since it provides insight into the effectiveness of solvers based on the classical DPLL algorithm [46, 47] and its variants. Many lower bounds have been shown for Resolution, using techniques such as the correspondence between the size of a proof and the size of the largest clause [13], or lower bounding scores in a two player game between a Prover and a Delayer on a CNF [96] (see [104] for a detailed survey of such techniques). This variety of lower bounds demonstrates that despite the recent success of SAT solvers based on DPLL and conflict-driven clause learning (CDCL), there still remain problems which cannot be efficiently decided. Indeed, such problems are very common and are straightforward to generate. Random 3-CNFs, in which kn clauses are chosen uniformly at random from the set of 3-clauses on n variables, have been shown to be hard for Resolution with high probability for suitable values of k [38].

Beyond Resolution, the substantially stronger Frege proof system is the classical 'textbook' proof system, in which lines consist of any logical formula, with a finite set of axiom and deduction rules. Frege is known to be strictly stronger than Resolution, in the sense that there are small proofs of more formulas. No lower bounds are currently known for the Frege proof system. The study of strong proof systems such as Frege is motivated not by their correspondence to current solvers, but by their relevance to longstanding complexity theory questions. The proof complexity of Frege systems, and subsystems with restricted classes of formulas, is believed to have close ties to circuit complexity. More generally, upper or lower bounds for sufficiently strong proof systems provide one possible method for resolving the question of $\mathbf{NP}$ vs $\mathbf{coNP}$ [43].

Proof systems based on algebraic or linear programming methods, rather than propositional logic, have also been developed. Cutting Planes (CP) [44] translates a CNF into a set of integer linear inequalities and uses integer linear programming methods to determine that there is no solution. There exist solvers which implement this approach to SAT solving [15], although this area is certainly less developed than CDCL solvers. Polynomial Calculus (PC) [2, 40] translates CNFs to polynomial equations, and by taking linear combinations of these equations, proves no solution exists by deriving the equation $1 = 0$. Both of these approaches strengthen Resolution, although they are incomparable with each other. This is illustrated in Figure 1, which describes the relative power of the proof systems mentioned.

QBF solving While the $\mathbf{NP}$-completeness of SAT allows a wide variety of problems to be reduced to a SAT instance, there are many problems we would like to solve which are not, or are not known to be, in $\mathbf{NP}$. *Quantified Boolean formulas* (QBFs) extend the language of propositional logic by introducing Boolean quantifiers. The substantially more expressive formulas in this language lift the complexity of determining the truth of QBFs to be $\mathbf{PSPACE}$-complete [107], the canonical problem solvable in polynomial space.

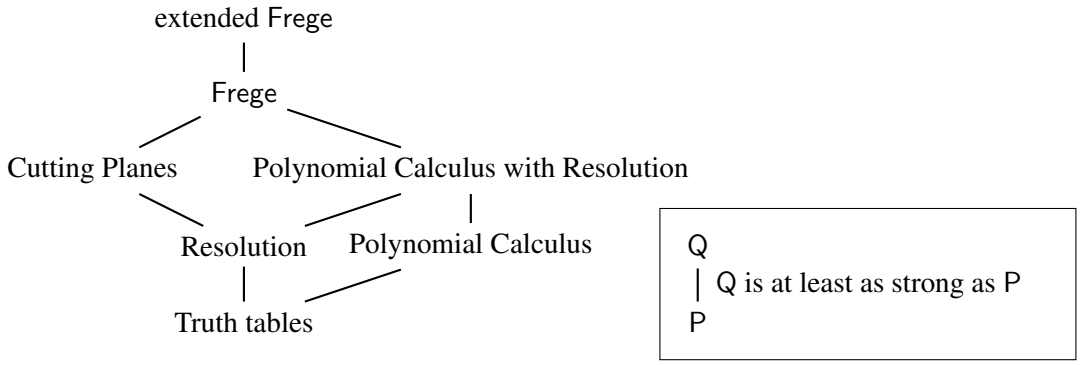

Fig. 1. A simulation diagram showing the relative power of some propositional proof systems

The **PSPACE**-completeness of the language of true QBFs (TQBF) gives practical QBF solving much wider applications beyond propositional satisfiability, in fields as varied as formal verification [14] and conformant planning [100]. The remarkable success of SAT solving and the greater applicability of QBFs have spurred a large drive to develop both the practical and theoretical aspects of QBF solving. Several QBF solvers, such as DepQBF [82], are based on the conflict-driven clause learning algorithm common in SAT solvers, but introduce new techniques to handle universal quantification. The solver RAReQS [69] adopts an alternative approach of counterexample guided abstraction and refinement (CEGAR), making use of a semantic interpretation of a QBF as a two player game. These techniques are combined to an extent in GhostQ [73]. Many solvers, such as CAQE [97] and DepQBF [82], allow for the extraction of certificates verifying the truth or falsity of a QBF; such certificates broaden the applications of such solvers yet further [106].

QBF proof complexity In a similar manner to the propositional case, one goal of QBF proof complexity is to study proof systems which can be related to runs of QBF solvers in order to show the strengths and weaknesses of different solvers. The differing approaches taken by solvers to handling universal quantification are mirrored in definitions of different QBF proof systems. The universal reduction ($\forall$red) rule allows the assignment of a value to a universal variable whenever the universal variable is quantified after all other variables in the formula. By adding the $\forall$red rule to Resolution, the proof systems Q-Res and QU-Res [72, 109] provide insight into the CDCL-based QBF solvers. On the other hand, expansion-based solving, such as that used in RAReQS, is better modelled by expanding universal variables, using the equivalence $\exists x \forall y \exists z \cdot \phi(x, y, z) \equiv \exists x \exists z \exists z' \cdot \phi(x, 0, z) \wedge \phi(x, 1, z')$, before using propositional resolution, giving the proof system $\forall$Exp+Res [70]. Extensions of these systems have also been defined, such as LD-Q-Res and LQU$^+$-Res [8,9,110], which allow tautologous clauses, in Q-Res and QU-Res respectively, under limited circumstances to mirror certain techniques used in practice, or IR-calc and IRM-calc [20], combining the expansion of $\forall$Exp+Res with Q-Res and LD-Q-Res respectively. The relative powers of these systems are shown in Figure 2, with a line indicating that a system is strictly stronger than the one below it.

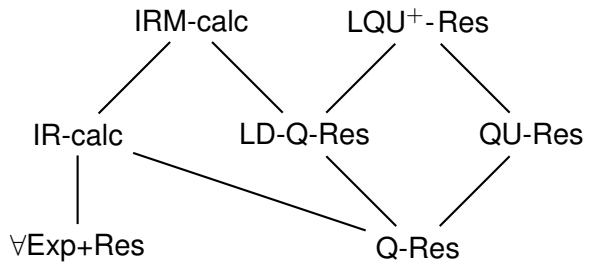

Fig. 2. A simulation diagram for Resolution-based QBF proof systems

Many of the propositional proof systems stronger than Resolution have also been extended to the realm of QBFs, generally by adapting the ∀red rule, producing systems such as CP+∀red and Frege +∀red, which are QBF proof systems obtained by adding a universal reduction rule to the propositional proof systems Cutting Planes and Frege respectively. The relative strength of these proof systems of the form P+∀red corresponds to that of the propositional systems in Figure 1, where Resolution +∀red is the proof system QU-Res.

The motivation for studying these stronger QBF proof systems is twofold. Lower bounds on some of these more powerful QBF proof systems, such as Frege +∀red, have particularly tight connections to circuit complexity [30]. Furthermore, analogously with the propositional case, sufficiently strong QBF upper or lower bounds would resolve the longstanding **NP** vs **PSPACE** question.

Despite the number of QBF resolution systems, lower bounds are obtainable for all of them simply by quantifying a known propositional Resolution bound with existential quantifiers. However, such lower bounds do not say much about the relative strength of different approaches to handling universally quantified variables. Some other propositional techniques have been lifted to provide lower bounds in QBF proof systems, such as feasible interpolation [22] and Prover-Delayer games [25]. However not all propositional techniques can be lifted, as evidenced by the failure of the size-width relations [23].

Beyond QBF adaptations of propositional techniques for QU-Res and other relatively weak QBF proof systems, and a few carefully constructed specific formulas [70, 72], the most successful technique for proving QBF proof complexity lower bounds has been that of *strategy extraction*. This technique relies on the interpretation of a QBF as a two player game, between existential and universal players. Any false QBF has a winning strategy for the universal player. Two different methods have been used to construct this strategy from a proof, one based on restricting proofs by partial assignments [53, 62], and one constructing Boolean circuits from proofs [19]. Both approaches can be extended to QBF proof systems beyond those based on Resolution, and both rely on strategies for universally quantified variables, and so have naturally been considered as techniques which provide 'genuine' QBF lower bounds. However, no satisfactory formal definition of what constitutes a 'genuine' QBF lower bound has been given, and the range of techniques for showing such bounds is still limited when compared with propositional proof systems.

1.2 Contributions

We begin by covering the necessary preliminaries, as well as surveying some of the key results in the literature. Chapter 2 covers the relevant background in propositional logic, SAT solving and propositional proof complexity. Chapter 3 deals similarly with quantified Boolean formulas, including QBF solving and QBF proof complexity, including definitions of several QBF proof systems used throughout this thesis. We now briefly describe the main contributions of this thesis.

Hardness from quantification Every SAT problem can be represented as a QBF by existentially quantifying all variables. All QBF proof systems therefore implicitly also define a propositional proof system, and lower bounds for this propositional proof system provide lower bounds for the QBF proof system, even if the propositional formula is obscured by some universal variables. Such propositional lower bounds are somewhat unsatisfactory in the realm of QBFs, as they give no insight into the effect of universal variables on the complexity of the formula.

A potential approach to resolve this problem is to construct a proof system in which there are no superpolynomial lower bounds arising solely due to propositional reasons. Just such a system, relaxing QU-Res, was proposed in [35] by employing oracles for some fixed level of the polynomial hierarchy when introducing axioms. This immediately provides short proofs of any propositional formula. A superpolynomial lower bound for relaxing QU-Res is then defined as any set of QBFs requiring proofs of superpolynomial size for any fixed level of the polynomial hierarchy.

While it does eliminate lower bounds which are entirely existentially quantified, relaxing QU-Res is not sufficient for distinguishing lower bounds based only on propositional hardness. In order to provide an example of a propositional lower bound for relaxing QU-Res, we give a way of combining false QBFs such that we have very precise control over the size of refutations in QU-Res, based only on the size of refutations of the original formulas. Using this, we construct a family of QBFs combining the pigeonhole principle formulas PHP_n, known to be hard for Resolution [65], with the QBFs KBKF_n of Kleine Büning et al. [72] where the falsity relies on many alternations of quantifiers, but which have short proofs in QU-Res. Despite the hardness of these QBFs in QU-Res being purely a propositional phenomenon, arising ony due to the hardness of the pigeonhole principle, we prove an exponential lower bound for proofs of these QBFs in relaxing QU-Res. This lower bound arises because the oracle access permitted when deriving axioms is unable to solve the pigeonhole principle effectively, since it is obscured behind the quantifer alternations of the KBKF_n formulas.

Theorem 4.11. *The QBF $\Phi_n = \mathrm{PHP}_n \oplus \mathrm{KBKF}_n$ requires relaxing QU-Res proofs of size $2^{\Omega(n)}$.*

The lower bound of Theorem 4.11, and the necessary definitions and constructions preceding it, are the focus of Chapter 4.

Not only does relaxing QU-Res still admit superpolynomial lower bounds based only on Resolution lower bounds, as defined it also has short proofs of any QBF with a bounded number of quantifier alternations, including some where it is clear that the hardness for QU-Res is a

genuinely QBF phenomenon, rather than a propositional one. To alleviate this issue, in Chapter 5 we define Σ_k^p-QU-Res and analogous systems Σ_k^p-P+$\forall$red, as an alternative method for incorporating Σ_k^p-oracle access into a proof system. Inspired by relaxing QU-Res, this new system simplifies the deduction rules while allowing the use of Σ_k^p-oracles elsewhere in the proof.

By construction, Σ_1^p-QU-Res has short proofs of propositional formulas. The restriction to using Σ_1^p-oracles allows Σ_1^p-QU-Res to distinguish between propositional lower bounds and those based on the universal quantification in the prefix. We exemplify this first with a short proof of the lower bound shown for relaxing QU-Res. We then also observe that lower bounds for Σ_1^p-P+$\forall$red are in fact bounds on the size of $\forall$-reduction steps in a proof, which is what we would expect if the hardness is derived from the alternation of quantifiers rather than for propositional reasons.

Having shown that Σ_1^p-QU-Res is able to characterise lower bounds for QU-Res which do not arise due to propositional hardness, we then consider Σ_k^p-QU-Res for larger k in order to understand the precise effect of the quantifier prefix on proof size. After limiting the proof systems necessary to consider to only Σ_k^p-QU-Res for odd k, we provide a separation between Σ_k^p-QU-Res and Σ_{k+2}^p-QU-Res for any odd k. Indeed, we show that there are formulas with a Σ_{k+2}^b-prefix for which QU-Res requires superpolynomial-size proofs, even with access to an oracle for Σ_k^p. We complete our consideration of Σ_k^p-QU-Res with a lower bound for Σ_k^p-QU-Res for any fixed k. The formulas which give this bound are the modified versions of the KBKF_n formulas, which provide a lower bound for QU-Res [9]. By this we see that not only does the falsity of these formulas rely on the unbounded number of quantifier alternations, but that this is also the source of the lower bound in QU-Res.

Characterising Frege +$\forall$red lower bounds Having described a means of identifying QBF lower bounds which arise due to the presence of universal quantifiers, we turn our attention to understanding the underlying reasons behind such results.

With this in mind, we first present a refinement of a dichotomy for Frege +$\forall$red proof systems shown in [30] in Chapter 6. By constructing a normal form for Frege +$\forall$red proofs using circuits witnessing the universal variables, the dichotomy of [30] proved that a large Frege +$\forall$red lower bound implies a correspondingly large lower bound on the propositional Frege proof system, or in circuit complexity. We observe that by reframing this in terms of Σ_1^p-Frege +$\forall$red, this result suffices to show that formulas which are hard for Σ_1^p-Frege +$\forall$red are precisely those which are hard for Frege +$\forall$red for non-propositional reasons. However, this characterisation of lower bounds only applies to relatively strong proof systems, whose lines are from circuit classes admitting certain closure properties. Weaker systems, such as QU-Res and others based on Resolution, do not fall into this category and so may admit lower bounds which do not fall into either of the above categories.

Nonetheless, we observe that the normal form used for Frege +$\forall$red can still be applied with some minor modifications, since extension variables required in these weaker proof systems can be encoded into the witnessing circuits. Through this modified normal form we obtain a characterisation

of lower bounds for any proof system P which admits strategy extraction – lower bounds which are neither propositional or circuit complexity lower bounds arise due to a lower bound in P on constructing witnessing circuits from the QBF. This characterisation applies to a wide variety of proof systems, including the widely used QU-Res. We exemplify our characterisation by providing examples for each type of lower bound from existing formulas known to be hard for QU-Res.

A lower bound technique for tree-like systems The equivalence of several different definitions of the Frege +∀red proof system was also observed in [30]. In particular, restricting to a tree-like proof, or restricting ∀-reductions to only substitute 0/1 values result in an equivalent system. However, applying both of these restrictions at once does not result in an equivalent proof system, a distinction we prove in Chapter 7 using a round-based strategy extraction argument. The key observation is that if universal reduction is restricted to 0/1 values, then each response in the strategy constructed by this algorithm corresponds to a path through the proof from root to axiom. Moreover, different responses arise from different paths, giving a lower bound for tree-like proofs even in strong systems such as Frege +∀red and eFrege +∀red based only on the number of responses required by a winning strategy, $\rho(\Phi)$.

Theorem 7.6. *For any QBF Φ, if π is a tree-like* P+∀red *refutation of Φ, then $|\pi| \geq \rho(\Phi)$.*

Such lower bounds via strategy size give the rather surprising result that, when ∀-reductions are required to substitute 0/1 values, tree-like Frege +∀red cannot simulate even relatively weak dag-like systems such as QU-Res. This is in stark contrast to the situation in propositional proof systems, where tree-like Frege is equivalent to dag-like Frege, and is substantially more powerful than Resolution.

This new lower bound technique for tree-like P+∀red proof systems shows that the dichotomy of [30] also does not hold in these tree-like systems. However, lower bounds via strategy size are the *only* method of showing lower bounds for tree-like Frege +∀red systems without also proving lower bounds for the corresponding dag-like Frege +∀red system. We show this by another modification of the normal form for Frege +∀red proofs. This normal form has a branch for each response, and so a strategy with few responses constructs few branches, resulting in a small proof.

A new lower bound technique Beyond propositional lower bounds, there are relatively few techniques for proving lower bounds for QBF proof systems. The strategy extraction technique of [19] provides a method for translating circuit complexity lower bounds into proof complexity lower bounds. As we have seen, for strong enough systems, this is sufficient to prove all non-propositional lower bounds. In weaker systems, it would be desirable to have a wider selection of tools available to prove bounds that cannot be shown by circuit complexity. Such tools may also provide some insight into those lower bounds which do not fall into the propositional or circuit complexity categories. Some propositional techniques have been adapted to provide genuinely QBF bounds, rather than propositional bounds, such as feasible interpolation [22] or, in the case of

tree-like systems, Prover-Delayer games [25]; lower bounds have also been provided for specific formulas by ad hoc methods [9, 70, 72].

Above, we introduced a new lower bound technique for tree-like proof systems based on the complexity of winning strategies. Chapter 8 introduces a similar technique for P+$\forall$red and Σ_1^p-P+$\forall$red proof systems in general, both tree-like and dag-like. We first define the *cost* of a QBF as a measure of the number of responses a winning strategy requires on a block of universal variables, and the *capacity* of a proof as the number of different responses a line of the proof system can give. We then combine these measures to give a simple lower bound on the size of proofs.

Theorem 8.7 (Size-Cost-Capacity). *Suppose π is a Σ_1^p-P+$\forall$red refutation of a false QBF Φ. Then*

$$|\pi| \geq \frac{\mathrm{cost}(\Phi)}{\mathrm{capacity}(\pi)}.$$

In order to provide lower bounds via cost and capacity, it is necessary to provide an upper bound on the capacity of proofs in the relevant proof system. It has been observed that all QU-Res proofs have capacity 1 [18]. We show the same bound on the capacity of proofs in the QBF version of the Cutting Planes proof system. In the case of Polynomial Calculus, it is not possible to give a constant upper bound, but we see that small proofs have small capacity. These capacity upper bounds allow us to obtain proof size lower bounds on these systems based only on the cost of formulas. We demonstrate an application of this by giving a simple proof that refutations of the well known formulas KBKFd_n require exponential size in these proof systems.

We conclude our study of the capacity of proof systems with examples of proof systems with large capacity. It is clear from the characterisation of Frege +$\forall$red lower bounds that there are Frege +$\forall$red proofs with high capacity. We also give an example of an algebraic proof system with large capacity. To do so, we give a QBF version of the Ideal Proof System (IPS) [64]. Since IPS is a static proof system, we first define an equivalent line-based proof system we call line-IPS (L-IPS). We can then add a universal reduction rule to L-IPS to give a QBF proof system IPS+$\forall$red. Demonstrating short proofs of QBFs with a large cost suffices to show that IPS+$\forall$red proofs can have large capacity.

IPS+$\forall$red p-simulates Frege +$\forall$red, so proving lower bounds for this system would represent a major breakthrough in proof complexity. While lower bounds via cost and capacity are not possible, we show that strategy extraction is still possible in IPS+$\forall$red, with the resulting strategy being represented as an arithmetic circuit. Given recent progress in lower bounds on some restricted arithmetic circuits [56, 57], this represents a promising direction for strong proof complexity lower bounds.

Random QBFs Thus far, cost lower bounds have only been shown for a couple of specific families of QBFs. In order to provide a large collection of QBFs which have large cost, we give in Chapter 9 a method for combining simple false QBFs to provide a false QBF which has large cost. These product formulas are constructed similarly to the combination which provided the lower bound for

relaxing QU-Res. However, altering the order in which the variables are quantified results in a substantially more difficult formula to refute. To produce a QBF with large cost, we only require that the QBFs in the product have non-constant winning strategies, and that there is a suitably small set of existential assignments which witness to this fact.

Indeed, these requirements are sufficiently weak that we can use these products to randomly generate QBFs which are false and have large cost with high probability. By considering a modification of randomly generated (1,2)-QCNFs [36, 45], and combining this with results on the falsity of random 2-SAT, we construct small random QBFs without constant winning strategies. From these, we define the randomly generated QBF $Q(n, m, c)$, which for suitable parameters m and c, is false and has high cost with high probability, giving a lower bound.

Theorem 9.16. *Let $1 < c < 2$ be a constant, and let $m \leq (1 - \epsilon)\log_2(n)$ for some constant $\epsilon > 0$. With high probability, the randomly generated QBF $Q(n, m, c)$ is false, and any QU-Res, CP+$\forall$red or PCR+$\forall$red refutation of $Q(n, m, c)$ requires size $2^{\Omega(n^\epsilon)}$.*

For propositional formulas, Resolution lower bounds for random 3-CNFs are well known [12, 38, 55, 68]. However, Theorem 9.16 represents the first proof size lower bound on randomly generated QBFs. The model used is more complex than that of a random 3-CNF, but this is necessitated by the presence of universally quantified variables. The QBFs $Q(n, m, c)$ are therefore guaranteed to have a specific number of universal and existential variables, to ensure they are neither trivially false by containing a clause with only universal variables, nor false for purely propositional reasons.

Finally, we offer some concluding thoughts and possible directions for future work in Chapter 10.

Some results in this thesis have appeared in previous publications. Chapters 4, 5 and 6 contain work from [29]. Chapter 7 contains work from [28]. Chapters 8 and 9 contain work from [17, 18] ([18] is the journal version of [17]).

Chapter 2

Background on Proof Complexity

QBFs are extensions of propositional logic, and as a result many QBF solving techniques and proof systems are based on those for propositional logic. In order to study QBF proof complexity, therefore, we must first understand the proof complexity of propositional logic. Much of this framework can then be lifted to QBFs.

This chapter contains an overview of propositional proof complexity and together with Chapter 3 defines much of the notation we use; a full list of the notation used is given in the preamble to this thesis. In Section 2.1, we give an introduction to propositional logic and Boolean circuits. Section 2.3 looks at the SAT problem, including solving techniques and its importance as an **NP**-complete problem. Finally, in Section 2.4, we discuss proof complexity, and give some examples of propositional proof systems and lower bounds.

2.1 Propositional logic

Propositional formulas The language of propositional logic contains the two constants 0 and 1, representing falsity and truth respectively, and a countably infinite set of variables V. An *assignment* to a set of variables $W \subseteq V$ is a function from W to $\{0,1\}$. We denote the set of all possible assignments to the variables of W by $\langle W \rangle = \{\alpha \mid \alpha : W \to \{0,1\}\}$.

In order to define the set of propositional formulas, it remains only to define a set of connectives. The set of connectives we use to define propositional logic consists of $\neg$, $\wedge$ and $\vee$, representing negation, conjunction and disjunction respectively. Although defining conjunction and disjunction as binary operators is sufficient, they are both associative and commutative, and so we extend $\wedge$ and $\vee$ for any arity $k \geq 0$. In the special case of arity 0, the empty conjunction is true, and the empty disjunction is false. Negation is defined only as a unary connective. The values of these connectives on different Boolean inputs can be seen in Figure 3.

We can now inductively define the set of propositional formulas.

Definition 2.1. *The constants 0 and 1 are propositional formulas, and v is a propositional formula for every variable $v \in V$. If p and q are propositional formulas, then so are $\neg p$, $p \wedge q$ and $p \vee q$.*

We define the function $\mathrm{var}(p)$, which maps a formula p to the set of variables which appear in p. Given propositional formulas p and q, and a variable x, we define $p[x/q]$ to be the formula obtained by replacing each instance of x in p with the formula q. For an assignment α, we define $p[\alpha]$, the restriction of p by α, to be the propositional formula obtained from p by performing the substitution $x/\alpha(x)$ for each $x \in \mathsf{dom}(\alpha)$. We generally assume that formulas restricted by assignments are also simplified by replacing $0 \wedge x$ by 0, $1 \wedge x$ by x and analogously for other connectives.

There are several other common Boolean connectives. The binary connective $\rightarrow$, representing implication, is sufficient to represent all Boolean functions as propositional formulas. The other connective we define is exclusive-or, denoted $\oplus$. Like $\wedge$ and $\vee$, this can be naturally extended to a k-ary connective for any k. The truth values of both $\rightarrow$ and $\oplus$ are given in Figure 3. Unless specified we only use connectives from $\{\neg, \wedge, \vee\}$; however, we use $p \rightarrow q$ as a shorthand for $(\neg p) \vee q$, and $p \leftrightarrow q$ for $(p \rightarrow q) \wedge (q \rightarrow p)$.

a	b	$\neg a$	$a \wedge b$	$a \vee b$	$a \rightarrow b$	$a \oplus b$
0	0	1	0	0	1	0
0	1	1	0	1	1	1
1	0	0	0	1	0	1
1	1	0	1	1	1	0

Fig. 3. The truth table for Boolean connectives

Boolean Circuits The inductive definition of a propositional formula in Definition 2.1 naturally gives rise to a rooted tree representing the formula. Such a tree has each leaf labelled with a variable from V, or a constant from $\{0, 1\}$, and all other nodes labelled with a connective defined on the number of inputs to that node. Such a tree defines a formula in the obvious way.

We can generalise this representation of formulas from a rooted tree to a directed acyclic graph (dag), in which a node can be an input to more than one subsequent node. We use the term *circuit* to refer to such a representation, and use *formula* to refer to circuits in which the graph is a tree.

Definition 2.2. *A* Boolean circuit *(or simply, circuit) is a directed acyclic graph with a unique sink, in which each source (node of in-degree* 0*) is labelled by a variable or a constant in* $\{0, 1\}$ *and each node of in-degree* $d \geq 1$ *is labelled by a connective which is defined on* d *variables.*

We also introduce two important properties of circuits: size and depth. The *size* of a Boolean circuit is the number of nodes in the dag. The *depth* of a Boolean circuit is the number of edges in the longest path from a source to the sink, so, e.g., constants and variables have depth 0. Allowing $\wedge$ and $\vee$ to take any number of inputs ensures that any Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ can be represented by a circuit of depth d for any $d \geq 3$.

Conjunctive Normal Form We now highlight some common constructions of propositional formulas which are of particular importance. A *literal* is a formula which is either a single variable x or the negation of a variable; we sometimes use $\bar{x}$ instead of $\neg x$. For convenience and clarity, we may refer to assigning a value to a literal, rather than a variable; the assignment $\bar{x}/0$ is equivalent to $x/1$ and vice versa.

A *clause* is a disjunction of literals, often represented as a set of literals. A formula is said to be in *conjunctive normal form (CNF)* if it is a conjunction of clauses. For any propositional formula, there is an equivalent formula in CNF. Observe that the size of a CNF equivalent to a formula p may be exponentially larger than the size of p. However, when we concern ourselves only with satisfiability of the CNF, we can construct an equisatisfiable CNF by introducing an additional variable for each node of the formula (known as Tseitin variables) and adding clauses requiring these variables to be equivalent to the value computed at that node. Adding a unit clause requiring the value computed at the root node of the formula to be true gives our equisatisfiable CNF.

Analogously to a clause, a *term* is a conjunction of literals; a formula is in disjunctive normal form (DNF) if it is a disjunction of terms. Similarly to CNF, for any propositional formula p, it is possible to find a DNF which is equivalent to p.

2.2 Circuit Complexity

Circuit complexity classes As noted above, while CNFs and DNFs allow us to construct circuits with a relatively simple structure expressing any Boolean function, such circuits may have size exponential in the number of variables. Complexity theory concerns itself with what problems, or functions, are efficiently computable, given a computational model and some measure of efficiency. One possible model is computation using Boolean circuits, often a class of circuits with a particular structure; this is the primary focus of circuit complexity. We conclude our overview of Boolean circuits by giving definitions of several prominent such circuit classes.

The first circuit class we define is the class of functions for which there exist polynomial-size circuits, denoted $\mathbf{P/poly}$. A function $f : \{0,1\}^* \to \{0,1\}$ is in $\mathbf{P/poly}$ if, for each $n \in \mathbb{N}$, there is a circuit on n variables which has size $n^{O(1)}$ and computes the function on any input of length n.

For $i \in \mathbb{N}$, the class $\mathbf{NC}^i$ contains all functions for which there are Boolean circuits with in-degree at most 2 which have size $n^{O(1)}$ and depth $O(\log^i n)$. Of particular interest is the class $\mathbf{NC}^1$, with depth $O(\log n)$, as this class contains all circuits which can be represented as polynomial-size formulas.

The classes $\mathbf{AC}^i$ are defined in a similar way to $\mathbf{NC}^i$, but circuits may contain $\wedge$ and $\vee$ nodes of any arity. Circuits in $\mathbf{AC}^0$, the class of constant depth circuits, are therefore able to depend on all input variables (unlike $\mathbf{NC}^0$), and are relatively well studied. Exponential lower bounds have been shown on the size of any constant depth circuit computing the parity of n bits [1, 59, 66]. Since the parity function is known to be in $\mathbf{NC}^1$, this gives the separation $\mathbf{AC}^0 \subsetneq \mathbf{NC}^1$.

Allowing $\mathsf{mod}\, p$ gates, which check whether the sum of the inputs is equal to $1 \mod p$, or threshold gates, checking whether the number of 1's in the input is above a certain threshold, in $\mathbf{AC}^i$ circuits results in the classes $\mathbf{AC}^i[p]$ and $\mathbf{TC}^i$ respectively.

It is important to note that even though the circuits witnessing that a function is in one of these circuit classes have polynomial-size, these witnessing circuits need not be computable in polynomial-time. If we require the nth circuit to be computable in polynomial-time, the circuit class is said to be *uniform*, otherwise, the class is *non-uniform*.

Arithmetic Circuits Beyond Boolean circuits, given a field $\mathbb{F}$ we analogously define an *arithmetic circuit* to have input gates labelled by variables, or by constants from $\mathbb{F}$, and internal gates labelled as either addition or multiplication gates. Such a circuit computes a polynomial over $\mathbb{F}$ in the natural way. The objects computed by arithmetic circuits are formal polynomials, rather than functions. For example, over the field $\mathbb{Z}_p$, the polynomials x^p and x are considered as distinct polynomials, despite computing the same function from $\mathbb{Z}_p$ to $\mathbb{Z}_p$.

An arithmetic formula is an arithmetic circuit in which the underlying dag is a tree. A particular case of arithmetic formulas is that of a *sparse polynomial*, computing a polynomial as a sum of monomials. Formally, a sparse polynomial consists of an addition gate, which is the sink, the inputs to which are a layer of multiplication gates, each of which have as input only constants or variables.

2.3 SAT and complexity

The Satisfiability problem Given a propositional formula, it is natural to ask when the formula is true and when it is false. The *satisfiability* (SAT) problem is the problem of deciding whether a formula ϕ, given as a CNF, has a satisfying assignment, i.e. an assignment to the variables of ϕ such that ϕ evaluates to true. Such decision problems are usually expressed in terms of establishing whether a string is a member of a given language, and so we define SAT to be the language consisting of CNFs which have at least one satisfying assignment.

The complexity class $\mathbf{NP}$ can be described as the class of languages for which membership can be witnessed by a polynomial-size witness, and given such a witness, membership in the language can be verified in polynomial time [5]. In the case of SAT, it is easy to see that a satisfying assignment provides this witness, as the evaluation of a CNF can be performed in polynomial time, and so SAT is in $\mathbf{NP}$.

In fact, the Cook-Levin theorem [42] shows that the SAT problem is $\mathbf{NP}$-hard, and therefore $\mathbf{NP}$-complete. The practical consequence of the Cook-Levin theorem is that for any problem in $\mathbf{NP}$, there is an efficient (polynomial-time) reduction of the problem to a SAT instance. This provides strong motivation for the development of algorithms and solvers for SAT. Given any problem in $\mathbf{NP}$, such as the travelling salesman problem or the graph colouring problem, both of which have several practical applications, we can reduce an instance of this problem to a SAT instance and solve it using the most efficient SAT solver.

We also introduce the complexity class $\mathbf{P}$, the class of languages for which there is a polynomial time deterministic algorithm to check membership of the language. No polynomial-time deterministic algorithm for SAT is known. The existence of such an algorithm would place SAT in $\mathbf{P}$, and thus show that $\mathbf{P} = \mathbf{NP}$, resolving a major open problem in complexity theory.

SAT solving Despite the apparent difficulty posed by the $\mathbf{NP}$-completeness of SAT, there has been much work on the design and implementation of algorithms for SAT. State-of-the-art SAT solvers regularly solve instances containing millions of variables, and are regularly used in applications such as planning problems [71] and formal verification through bounded model checking [32].

The DPLL algorithm [46, 47] lies at the heart of many modern SAT solvers. This algorithm is given a CNF in the form of a set of clauses, which are themselves sets of literals. The basic idea of this algorithm is to search all possible assignments by branching on each of the possible assignments for each variable until it either falsifies all literals in a clause, in which case it backtracks to the last branching point, or finds a satisfying assignment, in which case it returns 1. If all branches have been checked and no satisfying assignment has been found, the algorithm returns 0.

The DPLL algorithm (Algorithm 1) enhances this depth-first search procedure with two rules to simplify the CNF at each stage before branching on a variable. The first, *pure literal elimination*, checks for a literal l such that $\neg l$ does not appear in any clause – such a literal l is called a *pure literal*. If so, this literal can be assigned to true. The second, *unit propagation*, checks for a clause with only one unfalsified literal (a unit clause), and assigns this literal to true. When backtracking after finding a falsifying assignment, DPLL backtracks to before the last branching variable, prior to any unit propagation leading to the falsification of a clause.

Modern SAT solvers employ a variety of techniques to improve the efficiency of the DPLL algorithm. The choice of branching variable is of great importance in such solvers. Several heuristics have been developed to improve this choice, such as VSIDS and its variations [33, 88], used in solvers such as Chaff and MiniSat [51].

When the DPLL algorithm backtracks on finding a falsified clause, it establishes that this branch does not lead to a satisfying assignment, but learns nothing about other branches. Conflict driven clause learning (CDCL) [86, 87] constructs an implication graph throughout the algorithm, containing information on which of the current variables were assigned by an application of unit propagation, and which previous assignments to variables resulted in the relevant clause having only a single literal remaining. Upon reaching a conflict (i.e. containing the unit clauses x and $\neg x$ for some variable x), CDCL then finds a cut in this implication graph separating the choices made by the algorithm from the derived assignments x and $\neg x$. The negation of the assignments immediately prior to the cut is then added to the list of clauses in ϕ, since this partial assignment leads to a conflict.

Learning clauses in this way can greatly improve the efficiency of solvers by ensuring that they do not arrive at the same conflict by a different partial assignment, thus narrowing the search space. However, as memory is also a limitation on practical SAT solvers, care must be taken to only learn

Algorithm 1 The DPLL algorithm

```
function DPLL(φ)
    if φ contains a pure literal l then
        φ' ← φ[l/1]
        return DPLL(φ')
    else if φ contains a unit clause {l} then
        φ' ← φ[l/1]
        return DPLL(φ')
    else if φ = ∅ then
        return 1
    else if ∅ ∈ φ then
        return 0
    else
        Select a variable x in φ
        φ_0 ← φ[x/0]
        φ_1 ← φ[x/1]
        if DPLL(φ_0) = 1 then
            return 1
        else if DPLL(φ_1) = 1 then
            return 1
        else
            return 0
```

Note that $\phi[l/0] = \{C \setminus \{l\} \mid C \in \phi, \neg l \notin C\}$ and analogously for $\phi[l/1]$.

the 'most useful' clauses. Nonetheless, this CDCL technique forms the basis of many of the most successful modern SAT solvers such as Glucose [7] and MapleCOMSPS [81].

2.4 Proof systems and proof complexity

Given a CNF $\phi \in \mathrm{SAT}$, a satisfying assignment in $\langle \mathrm{var}(\phi) \rangle$ serves as a witness proving that $\phi \in \mathrm{SAT}$, as we can efficiently, in the size of the witness, compute the truth value of ϕ under this assignment and verify that it does indeed satisfy ϕ. More generally, for any class known to be in $\mathbf{NP}$, such witnesses exist by definition of $\mathbf{NP}$. The problem UNSAT is the complement of SAT, consisting of all unsatisfiable CNFs. This problem is $\mathbf{coNP}$-complete, as it is the complement of an $\mathbf{NP}$-complete problem. As $\mathbf{coNP}$ is not known to be in $\mathbf{NP}$, it is not known whether there exist polynomial-size witnesses that a formula is unsatisfiable.

One possible witness would be a computation of the truth value of ϕ under every possible assignment, such as a truth table. This can clearly be checked in polynomial-time (in the size of the truth table), but the size of a truth table is always exponential in the number of variables. The SAT solving techniques described in Section 2.3 demonstrate that it should be possible to reduce this search space, and reduce the size of the witness. In this vein, a refutational propositional proof system defines for each ϕ a set of witnesses that $\phi \notin \mathrm{SAT}$. More generally, a proof system for $\mathcal{L}$ is one answer to the question: *What is a suitable witness that $\phi \in \mathcal{L}$?*

Proof systems We begin with a formal definition of a proof system for any language $\mathcal{L}$. While the proof systems we use will be described in a more intuitive fashion, they can all be expressed in this formal sense using a suitable encoding of proofs as bit strings.

Definition 2.3 (Cook and Reckhow [43]). *A* proof system *for a language $\mathcal{L} \subseteq \{0,1\}^*$ is a polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ such that $\mathsf{rng}(f) = \mathcal{L}$.*

It is generally convenient when describing proof systems to use larger alphabets than $\{0,1\}$, for example when describing propositional formulas. Since all such alphabets Γ we use are finite, there is a simple translation between Γ^* and a subset of $\{0,1\}^*$, and so we do not give formal definitions of these alphabets or their encodings. By way of a simple example, we use this definition to describe the proof system in which a complete truth table is the only suitable witness that a CNF is unsatisfiable.

Example 2.4. There are polynomial-time computable encodings of both CNFs and truth tables as bit strings. Let f be the function such that $f(\pi) = \phi$ if π encodes the complete truth table for the CNF ϕ, and the final column of the truth table is identically 0, and $f(\pi) = \bot$ otherwise. Then f is a proof system for UNSAT.

For a proof system f, a string π is said to be an f-proof of $l \in \mathcal{L}$ if $f(\pi) = l$. Usually, the proof system will be clear from the context, and we simply refer to π as a proof of l.

When constructing a proof system, the condition that $\mathsf{rng}(f) = \mathcal{L}$ ensures that the proof system is both *sound* and *complete*. A proof system f is sound if $\mathsf{rng}(f) \subseteq \mathcal{L}$, i.e. there is no proof of any string which is not in $\mathcal{L}$. Conversely, f is complete if $\mathcal{L} \subseteq \mathsf{rng}(f)$, i.e. every string in $\mathcal{L}$ has a proof.

Proof complexity Given a proof system, the primary goal of proof complexity is to answer the question of how large proofs in a given proof system need to be. Other measures for proof systems have been considered, such as the memory space required by a proof system [2], but in this thesis we focus our attention solely on the size of proofs.

As we are interested in the size of the smallest proof of a given formula, we define $s_f(\phi) = \min\{|\pi| \mid f(\pi) = \phi\}$ for a proof system f for $\mathcal{L}$, and $\phi \in \mathcal{L}$. We generally assume that the formulas ϕ are *minimally unsatisfiable*, meaning that removing any clause from ϕ results in a satisfiable CNF, and so every clause is required for a refutation. A proof system is *polynomially-bounded* if there is some polynomial $p(n)$ such that for any $\phi \in \mathcal{L}$ with $|\phi| \leq n$, $s_f(\phi) \leq p(n)$. The existence of a polynomially-bounded proof system for UNSAT is equivalent to $\mathbf{NP} = \mathbf{coNP}$ [43]. Showing that no such proof system exists would therefore suffice to show that $\mathbf{NP} \neq \mathbf{coNP}$ and hence $\mathbf{P} \neq \mathbf{NP}$.

In order to show that an individual proof system is not polynomially-bounded, we give superpolynomial lower bounds on these proof systems. Such a lower bound consists of a sequence of formulas $\{\phi_n \mid n \in \mathbb{N}\}$ such that for any polynomial $p(x)$, there is some n such that $s_f(\phi_n) \geq p(|\phi_n|)$. In general, since $|\phi_n|$ is polynomial in n, a superpolynomial lower bound is a family of formulas such that $s_f(\phi_n) = n^{\omega(1)}$.

In addition to proving lower bounds, we also seek to compare the sizes of proofs in different proof systems. A proof system f is said to *simulate* a proof system g if there is a polynomial p such that for any $\pi \in \{0,1\}^*$, there is a $\pi' \in \{0,1\}^*$ with $|\pi'| \leq p(|\pi|)$ and $f(\pi') = g(\pi)$. That is, for every g-proof of a formula ϕ, there is an f-proof of ϕ which is at most polynomially larger. If such a π' is computable in polynomial-time from π, then we say that f *p-simulates* g. If two proof systems (p-)simulate each other, they are said to be *(p-)equivalent*, and if neither simulates the other, the proof systems are *incomparable*.

Line-based proof systems A propositional proof system is a proof system for the language UNSAT, or equivalently for the language TAUT of tautologous DNFs by showing the negation is in UNSAT. We introduce two such systems here; other propositional proof systems will be introduced where they are used. In common with almost all proof systems considered in this thesis, these are *line-based* proof systems.

A line-based proof system consists of a set of axiom rules, defining what lines can be introduced as axioms given a formula ϕ, and a set of deduction rules, defining what lines can be deduced from previous lines. A *derivation* is then a sequence of lines $L_1, \ldots, L_m$ such that each L_i is either introduced by an axiom rule, or deduced from $L_1, \ldots, L_{i-1}$ by a deduction rule. A proof that $\phi \in \text{UNSAT}$ is therefore a derivation of the empty clause $\bot$ or some other trivial falsity. A proof deriving $\bot$ is sometimes referred to as a refutation. Since all proof systems we consider prove the falsity of a formula, we use the terms 'proof' and 'refutation' interchangeably.

The axiom and deduction rules are chosen so that the proof system is sound and complete, i.e. so that $\phi \wedge L_1 \wedge \cdots \wedge L_{i-1} \models L_i$, and consequently $\phi \models L_i$ for each i, and so that there is a derivation of $\bot$ for any false CNF ϕ. In order to ensure that this is a proof system in the formal sense of Definition 2.3, it is only necessary to ensure that there is a polynomial-time computable algorithm to check whether a line L_i was derived by a deduction rule of the proof system. We can then construct a function which returns ϕ if and only if each line was correctly derived and the last line is $\bot$, and returns the trivial unsatisfiable formula $\bot$ otherwise.

We can also view such a proof as a directed acyclic graph with vertices $\{L_1, \ldots, L_m\}$ and an edge from L_j to L_i for any $j \leq i$ where L_j is used as a premise in the deduction rule used to deduce L_i. If there is a path from L_j to L_i in this dag, i.e. L_j is used in the derivation of L_i, we denote this by $L_j \prec L_i$. If we require this dag to be a tree, i.e. if a line is used in multiple deductions, it must be derived multiple times, we refer to the resulting proof system as *tree-like*; otherwise the proof is *dag-like*.

Resolution The first proof system we describe is the *Resolution* proof system for the language UNSAT of unsatisfiable CNFs [34, 101]. This relatively simple proof system works with lines which are clauses. It can introduce as an axiom any clause from the CNF ϕ and has only one deduction rule: the resolution rule (see Figure 4). The variable appearing in opposite polarities which is removed

via the application of the resolution rule is referred to as the *pivot* (x in the example in Figure 4). A Resolution proof that ϕ is unsatisfiable is a derivation of the empty clause $\bot$ from the clauses of ϕ.

Axiom:	$\dfrac{}{C}$	C is a clause in the CNF
Resolution:	$\dfrac{C \vee x \qquad D \vee \neg x}{C \vee D}$	C and D are clauses, x is a variable

Fig. 4. The derivation rules of Resolution [34, 101]

Example 2.5. As an example of a (dag-like) Resolution proof, Figure 5 is one possible refutation of the unsatisfiable CNF $\phi = (x \vee \neg y \vee \neg z) \wedge (\neg y \vee z) \wedge (\neg x \vee \neg z) \wedge (\neg x \vee y \vee z) \wedge (x \vee y)$. Observe that the proof in Figure 5 is not the only Resolution refutation of ϕ, and indeed is not even the shortest possible Resolution proof.

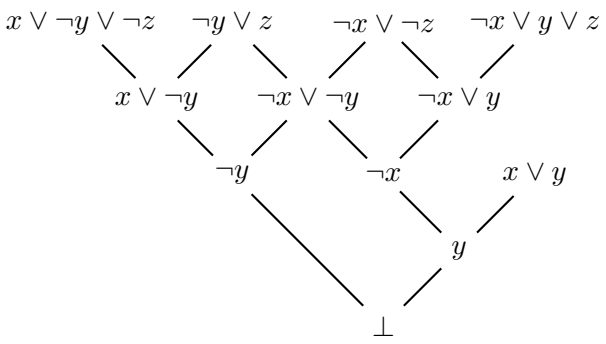

Fig. 5. An example Resolution proof

Resolution is often augmented with the *weakening rule*, which allows the derivation of the clause $C \vee l$ from C, for some literal l. Resolution with weakening is no more powerful than Reslution without weakening, since the shortest refutation of any formula contains no instances of the weakening rule. However, with the addition of weakening, Resolution becomes *implicationally complete*: there is now a derivation of C from ϕ for any clause C such that $\phi \models C$. As this is a convenient property for a proof system, and does not alter the lengths of refutations, we allow weakening in Resolution, and its analogues in other proof systems.

Resolution is perhaps the most well studied propositional proof system, due to the tight connections between Resolution and SAT solving, particularly the DPLL algorithm and CDCL used by many SAT solvers. Given the trace of the DPLL algorithm on a formula ϕ, one can construct a tree-like Resolution refutation of at most the size of the branching tree by mapping each node to a

clause which is falsified by the assignment at that node. Branching on two different assignments to a variable then corresponds to a resolution step on that variable, unit propagation is modelled by resolution with a clause of ϕ and pure literal detection requires no further steps. The clause at the root must be $\bot$ as the only clause false under the empty assignment. The learning of clauses in CDCL necessitates using the full power of dag-like Resolution, since learnt clauses can be subsequently be falsified several times by subsequent assignments, but traces of a CDCL algorithm can still be modelled as a Resolution proof [10]. Conversely, given a suitable decision strategy and clause-learning strategy, modern CDCL-based SAT solvers can run as efficiently as any Resolution proof [6, 92]. Finding the optimal such strategies is not known to be possible efficiently.

The first superpolynomial lower bounds for Resolution were shown by Haken [65] for the CNFs PHP_n (Definition 2.6), a family of formulas based on the pigeonhole principle. This lower bound on PHP_n was later improved to an exponential lower bound in [93].

Definition 2.6. *The nth pigeonhole principle formula, denoted PHP_n is the unsatisfiable CNF*

$$\bigwedge_{i=1}^{n+1} (x_{i,1} \vee \cdots \vee x_{i,n}) \wedge \bigwedge_{j=1}^{n} \bigwedge_{1 \leq i < k \leq n+1} (\neg x_{ij} \vee \neg x_{kj})$$

where the variable $x_{i,j}$ is interpreted as 'the ith pigeon is assigned to the jth hole.'

Theorem 2.7 ([65, 93]). *The pigeonhole principle formulas PHP_n require Resolution refutations of exponential size.*

Frege systems Instead of working with clauses, as in Resolution, we can extend the lines of a proof system to work with any Boolean circuit from a circuit class $\mathcal{C}$ to define a $\mathcal{C}$-Frege proof system.

Various $\mathcal{C}$-Frege systems can be defined, using any sound and complete set of axioms and derivation rules. However, for a suitable and fixed circuit class $\mathcal{C}$, all $\mathcal{C}$-Frege systems are known to be p-equivalent [99], so we do not distinguish between such variations, and instead consider $\mathcal{C}$-Frege as a single proof system. A definition of $\mathcal{C}$-Frege with suitable axioms and a single derivation rule, modus ponens, is given in Figure 6.

Resolution could potentially be viewed as a $\mathcal{C}$-Frege system for the very restrictive class of circuits consisting only of clauses. A stronger system is bounded-depth Frege, or $\mathbf{AC}^0$-Frege. This is the strongest propositional Frege system for which superpolynomial lower bounds are known, using the formulas PHP_n [79].

We refer to the $\mathbf{NC}^1$-Frege proof system simply as Frege. In this proof system, the lines can be any Boolean formula. This is a relatively powerful proof system, and no superpolynomial lower bounds are known for this proof system; in fact, the best known lower bounds are quadratic [75].

A potentially more powerful system is $\mathbf{P/poly}$-Frege, which we call extended Frege or eFrege. An alternative definition of eFrege is as a proof system working with formulas, as in Frege, but with the addition of extension variables [108]. Extension variables are variables equivalent to a Boolean formula, introduced by the axiom $x_i \leftrightarrow C$ for a variable x not already in the proof, and

Clause Axiom:	$\dfrac{}{C}$	C is a clause of the CNF
Axioms:	$\dfrac{}{(\neg A \to \neg B) \to (B \to A)} \quad \dfrac{}{A \to (B \to A)} \quad \dfrac{}{(A \to (B \to C)) \to ((A \to B) \to (A \to C))}$	for any $A, B, C \in \mathcal{C}$
Modus Ponens:	$\dfrac{A \qquad A \to B}{B}$	for any $(A \to B) \in \mathcal{C}$

Fig. 6. A possible set of derivation rules for $\mathcal{C}$-Frege [83]

a formula $C \in \mathbf{NC}^1$. Using extension variables allows us to represent any circuit in $\mathbf{P/poly}$ using a Boolean formula by introducing extension variables equivalent to subcircuits which are used multiple times. The power of these extension variables is such that the addition of extension variables to the relatively weak Resolution proof system, which we denote *extended Resolution*, results in a proof system equivalent to eFrege [43, 78].

The strength of these Frege systems are such that tree-like $\mathcal{C}$-Frege and dag-like $\mathcal{C}$-Frege are equivalent for large enough circuit classes [75]. In particular, this is the case for Frege and eFrege, as well as for the circuit classes $\mathbf{AC}^0$ and $\mathbf{TC}^0$.

This is by no means a complete list of propositional proof systems, however for most of this thesis the proof systems used will be based on Resolution or $\mathcal{C}$-Frege systems. Figure 7 contains the various simulations of these systems; superpolynomial lower bounds are known for all proof systems below the dashed line, and all p-simulations below this line are also separations. For proof systems above the line, no separations are known, since such a separation would require a superpolynomial lower bound on these proof systems. Also included in Figure 7 are several proof systems based on algebraic reasoning, namely Cutting Planes, Polynomial Calculus and its extension PCR,and IPS, which will be introduced in Chapter 8. Techniques for lifting these propositional proof systems to the language of QBF are discussed in Chapter 3.

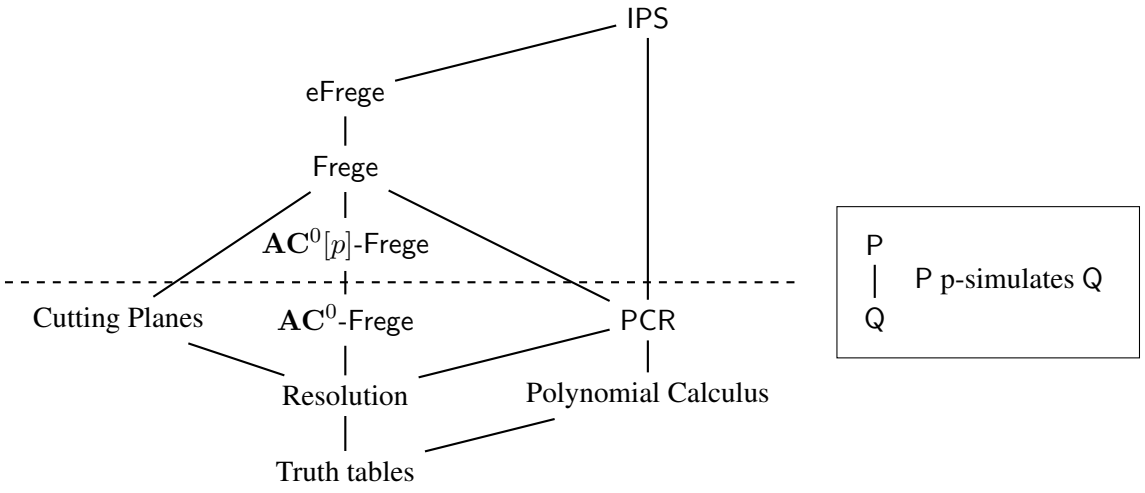

Fig. 7. A simulation diagram for propositional proof systems used in this thesis

Chapter 3

Background on Quantified Boolean Formulas

Quantified Boolean formulas introduce universal quantifiers, complementing the implicit existential quantification of a SAT problem. This allows for much more expressive formulas, modelling problems from fields such as ontological reasoning [74], conformant planning [100] and formal verification [14], as well as more light-hearted problems, including games such as Tic-Tac-Toe and Connect-4 [50, 60]. However, to solve QBFs, we must introduce some new techniques, as propositional reasoning alone is not sufficient.

In this chapter, we provide an overview of QBFs and QBF proof complexity. The definition of QBFs, and different approaches to their semantics, are discussed in Section 3.1, as well as the complexity of deciding QBFs and some restrictions of QBFs. Section 3.2 contains a description of how SAT solving can be extended to QBF solving, and in Section 3.3 we detail the most prominent approaches to constructing QBF proof systems from propositional proof systems, and the relations between them.

3.1 Quantified Boolean Formulas

Quantified Boolean formulas extend propositional logic with the addition of existential and universal quantifiers, $\exists$ and $\forall$ respectively, ranging over the values $\{0, 1\}$. Semantically, we can interpret $\exists x \cdot \phi$ as $\phi[x/0] \vee \phi[x/1]$ and similarly $\forall x \cdot \phi$ is equivalent to $\phi[x/0] \wedge \phi[x/1]$. A variable that is in the scope of a quantifier is *bound*, otherwise it is *free*.

A *prenex* QBF is a QBF Φ of the form $\Phi = \Pi \cdot \phi$, where Π is a *quantifier prefix*, i.e. a sequence of quantified variables, such as $\forall x \exists y \forall z$, and ϕ, the *matrix*, is a propositional formula, i.e. containing no quantifiers. Any QBF can be transformed into a prenex QBF by renaming variables so that each of the bound variables is distinct from all other variables, and then moving all quantifiers in front of the propositional formula, changing quantifiers as needed whenever a quantifier is moved out of the scope of a negation.

Example 3.1. The QBF $\exists x(x \vee y) \wedge \forall x \exists z(x \vee (y \wedge z))$ is not a prenex QBF. We can transform it into a prenex QBF by moving all quantifiers to the front, and renaming one copy of x. The QBF $\exists x \forall x' \exists z \cdot (x \vee y) \wedge (x' \vee (y \wedge z))$ is therefore an equivalent prenex QBF. The quantifier prefix is $\exists x \forall x' \exists z$ and the matrix is $(x \vee y) \wedge (x' \vee (y \wedge z))$.

A QBF is *closed* if it has no free variables, i.e. all variables are quantified. Since it has no free variables, expanding the quantifiers using the equivalences $\exists x \cdot \phi \equiv \phi[x/0] \vee \phi[x/1]$ and $\forall x \cdot \phi \equiv \phi[x/0] \wedge \phi[x/1]$ results in a propositional formula with no free variables, which can be evaluated to either true or false.

Similarly to the case for SAT, we can require that the propositional formula in a prenex QBF is a CNF. If a closed, prenex QBF has a CNF matrix, then it is in *prenex normal form*, denoted PCNF. Using extension variables, which must be quantified existentially and after the variables they depend on, we can transform any closed QBF into a PCNF with a polynomial increase in size. We therefore assume that all QBFs we consider are PCNFs, and use the terms interchangeably.

Example 3.1 (continued). The prenex QBF $\exists x \forall x' \exists z \cdot (x \vee y) \wedge (x' \vee (y \wedge z))$ is not closed, since y is a free variable. If y is also quantified in the prefix, e.g. as $\forall y$, then the resulting QBF $\exists x \forall x' \exists z \forall y \cdot (x \vee y) \wedge (x' \vee (y \wedge z))$ is a closed prenex QBF. This can be transformed into a PCNF by expanding the propositional matrix into an equivalent CNF, giving the PCNF

$$\exists x \forall x' \exists z \forall y \cdot (x \vee y) \wedge (x' \vee y) \wedge (x' \vee z)$$

It is clear that the order in which variables are quantified in the quantifier prefix of a PCNF is important: the QBF $\forall y \exists x \cdot (x \leftrightarrow y)$ is true, while $\exists x \forall y \cdot (x \leftrightarrow y)$ is false. However if variables are quantified consecutively with the same quantifier, changing their relative order does not affect the truth of the PCNF. With this in mind, we introduce the notion of the *level* of a variable in a QBF prefix. A PCNF is of the form $\Phi = \mathcal{Q}_1 X_1 \mathcal{Q}_2 X_2 \ldots \mathcal{Q}_n X_n \cdot \phi$, where $\mathcal{Q}_i \in \{\exists, \forall\}$ for $1 \leq i \leq n$ are quantifiers with $\mathcal{Q}_i \neq \mathcal{Q}_{i+1}$, and $X_1, \ldots, X_n$ are pairwise disjoint sets of variables. We refer to each individual $\mathcal{Q}_i X_i$ as a *quantifier block*, or simply a *block*. For any variable $x \in X_i$, we say that the level of x is i, and write $\mathrm{lv}(x) = i$. We can also extend this notation to literals l by defining $\mathrm{lv}(\bar{x}) = i$. We say that x is left (resp. right) of y whenever $\mathrm{lv}(x) < \mathrm{lv}(y)$ (resp. $\mathrm{lv}(x) > \mathrm{lv}(y)$). The sets $\mathcal{X}$ and $\mathcal{U}$ consist of all existential and universal variables respectively, i.e. $\mathcal{X} = \bigcup_{\{i \mid \mathcal{Q}_i = \exists\}} X_i$ and $\mathcal{U} = \bigcup_{\{i \mid \mathcal{Q}_i = \forall\}} X_i$.

Game semantics The semantics of QBFs can be understood by expanding a (closed prenex) QBF into a propositional formula. Since all variables are quantified, all variables will be expanded and the resulting formula will have no variables, evaluating to either true or false.

However, a perhaps simpler and certainly more useful semantic interpretation of a QBF is as a game between a universal player and an existential player. Given a closed prenex QBF $\Phi = \mathcal{Q}_1 X_1 \mathcal{Q}_2 X_2 \ldots \mathcal{Q}_n X_n \cdot \phi$, the players assign variables in the order they appear in the quantifier prefix, with variables of X_i being assigned by the player corresponding to $\mathcal{Q}_i$. The universal

player wins this game if ϕ evaluates to 0 under the total assignment constructed during the game, whereas the existential player wins if ϕ evaluates to 1. A QBF is false if and only if the universal player can guarantee a win in this game; for any false QBF, it is therefore possible to construct a winning strategy for the universal player. Formally, a *strategy* for a variable $u \in X_i$ is a function $S_u : \langle X_1 \cup \cdots \cup X_{i-1} \rangle \to \{0, 1\}$. A strategy for the universal player therefore consists of a strategy S_u for every universally quantified variable u in Φ. A winning universal strategy $S : \langle \mathcal{X} \rangle \to \langle \mathcal{U} \rangle$ for a QBF $\Phi = \Pi \cdot \phi$ is then one such that $\phi[\alpha \cup S(\alpha)] \equiv \bot$ for every $\alpha \in \langle \mathcal{X} \rangle$.

Considering a QBF as such a game allows for natural encodings of many problems, including many two player games, such as Connect-4 [60] and some problems in Chess [4]. It also suggests the use of winning strategies as a certificate for the truth or falsity of a QBF; partial strategies are provided as such a certificate by some QBF solvers [97].

Complexity The language TQBF is the language of true closed QBFs given as PCNFs. The addition of quantifiers suggests that deciding this language may be substantially harder than the $\mathbf{NP}$-complete SAT problem. Indeed, it is known that TQBF, and the complement FQBF, the language of false PCNFs, are complete for the class $\mathbf{PSPACE}$ [91], the class of problems solvable in polynomial space and a potentially much larger complexity class than $\mathbf{NP}$. Nonetheless, the questions of $\mathbf{NP}$ vs $\mathbf{PSPACE}$, and even $\mathbf{P}$ vs $\mathbf{PSPACE}$, are currently open.

The number of alternations of quantifiers also plays a key role in the complexity of TQBF. Observe that if we have a PCNF in which every variable is quantified existentially, this is an instance of a SAT problem. We can extend this limitation on the number of quantifier blocks in the prefix to allow a constant number of blocks for constants larger than 1.

For a constant k, a Σ_k^b quantifier prefix is a prefix of the form $\exists X_1 \forall X_2 \ldots \mathcal{Q}_k X_k$, i.e. a prefix containing k quantifier blocks with the first block quantified existentially. Analogously, a Π_k^b prefix has k blocks with the first block being universally quantified. We can then define the complexity classes Σ_k^p and Π_k^p to be those problems polynomial-time reducible to a QBF with a Σ_k^b and a Π_k^b prefix respectively. Observe in particular that $\Sigma_1^\mathsf{p} = \mathbf{NP}$. The *polynomial hierarchy* ($\mathbf{PH}$), the union of the classes Σ_k^p and Π_k^p for all $k \in \mathbb{N}$, is the class of all problems polynomial-time reducible to a QBF with some constant number of quantifier blocks.

3.2 QBF solving

Given a QBF Φ in prenex normal form, there is a relatively straightforward extension of the DPLL algorithm to the algorithm QDPLL (Algorithm 2), which determines whether the QBF is true or false. In order to handle the addition of quantifiers, two modifications are made. First, when picking a variable to branch on, this variable must be chosen from the leftmost quantifier block, as the assignment of later variables may depend on the value of those in previous blocks. Second, when a variable x has been picked to branch on, if x is existentially quantified, the algorithm must verify that $\Phi[x/0]$ or $\Phi[x/1]$ are true, as in DPLL. However, if x is universally quantified, the QDPLL algorithm must check that both $\Phi[x/0]$ and $\Phi[x/1]$ evaluate to true.

Algorithm 2 The QDPLL algorithm

Input: a closed QBF $\mathcal{Q}_1 X_1 \dots \mathcal{Q}_k X_k \cdot \phi$ in prenex normal form

function $\mathsf{QDPLL}(\mathcal{Q}_1 X_1 \dots \mathcal{Q}_k X_k, \phi)$
  $\phi \leftarrow \mathsf{Simplify}(\mathcal{Q}_1 X_1 \dots \mathcal{Q}_k X_k, \phi)$
  if $\phi = \varnothing$ **then**
    return 1
  else if $\varnothing \in \phi$ **then**
    return 0
  else
    Select a variable x in X_1
    $\phi_0 \leftarrow \phi[x/0]$
    $\phi_1 \leftarrow \phi[x/1]$
    if $\mathcal{Q}_1 = \forall$ **then**
      return $\mathsf{QDPLL}(\mathcal{Q}_1 X_1 \setminus \{x\} \dots \mathcal{Q}_k X_k, \phi_0) \wedge \mathsf{QDPLL}(\mathcal{Q}_1 X_1 \setminus \{x\} \dots \mathcal{Q}_k X_k, \phi_1)$
    else
      return $\mathsf{QDPLL}(\mathcal{Q}_1 X_1 \setminus \{x\} \dots \mathcal{Q}_k X_k, \phi_0) \vee \mathsf{QDPLL}(\mathcal{Q}_1 X_1 \setminus \{x\} \dots \mathcal{Q}_k X_k, \phi_1)$

If $X_1 \setminus \{x\} = \varnothing$, then this block is removed and all other blocks are renumbered.

As in DPLL, various simple reasoning techniques can be used to simplify the formula at each stage; these are represented in Algorithm 2 by the function Simplify. Pure literal elimination is still possible, however if a pure literal is quantified universally, then the algorithm assigns the pure literal to be false. Unit propagation can be performed using any clause containing only a single existential literal which is quantified leftmost among all literals in that clause. The function Simplify performs both of these, as well as other simple procedures such as evaluating the matrix to false if it contains a clause with only universal literals.

Much like with DPLL, this can be extended to allow clause learning in a similar way to the propositional case, resulting in the QCDCL algorithm, which forms a basis for several QBF solvers [82, 110]. Many techniques similar to those used in SAT solving are used to optimise the choice of branching variables and clauses learnt. A tool unique to the QBF solving case, however, is the calculation of dependency schemes. While the order of variables in the quantifier prefix gives a natural order in which to assign variables in QCDCL, not every variable will depend on the all variables to its left. Solvers such as DepQBF [82, 105] recalculate these dependency schemes regularly, and are thereby sometimes able to soundly assign variables much earlier than might otherwise be possible when a variable is found to be independent of all remaining variables. Doing so can lead to a substantially faster runtime on certain formulas.

Rather than QCDCL, some solvers use an alternative algorithm based on the expansion of quantifiers. While a full expansion of all the variables would result in an exponential increase in the size of the formula, the *counterexample guided abstraction refinement* (CEGAR) [39, 69] algorithm attempts to avoid this issue by expanding variables beginning from the rightmost, and constructing partial winning strategies on the blocks of variables working from right to left. However, in the worst case a full expansion of the variables is still needed.

3.3 QBF proof complexity

Soundness in QBF proofs Before considering QBF proof systems, it is prudent to consider the soundness of such systems given the added complexity of quantifiers. In some sense, the definition of a sound QBF proof system is clear – it can only prove true QBFs, or equivalently can only refute false QBFs. However, if we wish to consider line-based QBF proof systems, we must define what it means for the derivation of a particular line to be sound, rather than a proof as a whole.

All line-based QBF proof systems we consider have lines consisting of propositional statements, where the axioms that can be introduced are defined by the matrix of the QBF, and where the quantifier prefix is implicitly identical to that of the QBF. Line-based proof systems are refutational, and so in a propositional proof system, $\frac{C_1 \quad \dots \quad C_n}{D}$ is sound if any assignment satisfying $C_1 \wedge \dots \wedge C_n$ also satisfies D. For QBFs, the situation is a little more complex due to the quantification of variables, however soundness can still be expressed relatively concisely: given a quantifier prefix Π, the derivation $\frac{C_1 \quad \dots \quad C_n}{D}$ is sound if any existential winning strategy on $C_1 \wedge \dots \wedge C_n$ (with respect to Π) also satisfies D. With this definition of soundness in hand, we are now in a position to consider line-based QBF proof systems.

Universal reduction Since an existentially quantified QBF is simply a SAT problem, all QBF proof systems must restrict to a propositional proof system on such instances. In fact, most commonly studied QBF proof systems extend a line based propositional proof system by adding a method for handling universally quantified variables.

Perhaps the most commonly studied such method is the addition of the $\forall$-reduction rule. The $\forall$-reduction rule allows the deduction of $L[u/b]$ from L whenever u is a universal variable in the rightmost block appearing in L, and $b \in \{0, 1\}$. This rule is easily seen to be sound: if in the two player game, the existential player must ensure L is satisfied, then the existential player must also ensure that $L[u/b]$ is satisfied, else when the universal player comes to choose a value for u, she can play $u \leftarrow b$ and falsify L.

The $\forall$-reduction rule was first introduced in the system Q-Res [72], which allows only two deduction rules: resolution on existentially quantified pivots, and the $\forall$-reduction rule. The deduction of tautologies is also forbidden; this is not essential but allows for a simpler representation of the $\forall$-reduction rule. A Q-Res derivation of the empty clause from a QBF Φ is therefore a proof that Φ is false. Similar to the correspondence between Resolution and solvers based on DPLL or CDCL, the trace of a QDPLL solver can be modelled by a proof in Q-Res.

In [109], Q-Res was extended to QU-Res, which allows resolution on any pivot variable, regardless of its quantifier. QU-Res clearly p-simulates Q-Res, and in fact there is an exponential separation. The derivation rules of QU-Res are given in Figure 8; the rules of Q-Res are identical, with the added restriction that the variable x in the Resolution rule must be existentially quantified.

Axiom:	$\dfrac{}{C}$	C is a non-tautological clause in the CNF
Resolution:	$\dfrac{C \vee x \qquad D \vee \neg x}{C \vee D}$	If $l \in C$ then $\neg l \notin D$ for any literal l
$\forall$-reduction:	$\dfrac{C \vee u}{C}$	u a universal literal, $\mathrm{lv}(u) \geq \mathrm{lv}(l)$ for all literals $l \in C$

Fig. 8. The derivation rules of QU-Res [72, 109]

Example 3.2. Figure 9 gives an example Q-Res proof (and therefore also a QU-Res proof) of the QBF $\exists x \forall u \exists y \cdot (x \vee u \vee y) \wedge (x \vee u \vee \neg y) \wedge (\neg x \vee \neg u \vee y) \wedge (\neg x \vee \neg u \vee \neg y)$, representing resolution steps by solid lines, and $\forall$-reduction steps by dotted lines.

No $\forall$-reduction steps can be performed initially, since all clauses contain a literal on y, which is quantified to the right of u. Having resolved on y, a $\forall$-reduction is necessary before any further resolution steps can take place, as tautologies are forbidden in Q-Res.

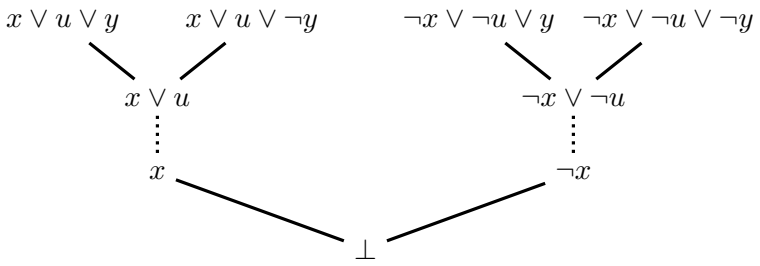

Fig. 9. An example Q-Res proof

On existentially quantified formulas, both Q-Res and QU-Res are equivalent to Resolution, and so the existentially quantified pigeonhole principle PHP_n provides an exponential lower bound, as do any other lower bounds on Resolution. However, there are also Q-Res and QU-Res lower bounds for QBFs making use of universal quantifiers. The most prominent of these are the QBFs KBKF_n defined by Kleine Büning, Karpinski and Flögel in [72].

Definition 3.3 (Kleine Büning et al. [72]). *Define the QBF* KBKF_n *as*

$$\mathrm{KBKF}_n := \exists y_0 (\exists y_1 y_1' \forall u_1) \ldots (\exists y_n y_n' \forall u_n) \exists y_{n+1} y_{n+2} \ldots y_{2n} \cdot \bigwedge_{i=0}^{2n} C_i \wedge C_i'$$

where the clauses C_i and C'_i are defined as

$$\begin{aligned}
C_0 &= \{\neg y_0\} & C'_0 &= \{y_0, \neg y_1, \neg y'_1\} & \\
C_k &= \{y_k, \neg u_k, \neg y_{k+1}, \neg y'_{k+1}\} & C'_k &= \{y'_k, u_k, \neg y_{k+1}, \neg y'_{k+1}\} & \text{for } k \in [n-1] \\
C_n &= \{y_n, \neg u_n, \neg y_{n+1}, \ldots, \neg y_{2n}\} & C'_n &= \{y'_n, u_n, \neg y_{n+1}, \ldots, \neg y_{2n}\} & \\
C_{n+k} &= \{\neg u_k, y_{n+k}\} & C'_{n+k} &= \{u_k, y_{n+k}\} & \text{for } k \in [n]
\end{aligned}$$

The QBFs KBKFd_n *are defined similarly, but replace each instance of* $\forall u_i$ *in the prefix with* $\forall u_i \forall v_i$*, and add the matching literal on* v_i *to every clause containing* u_i*, so e.g.* $C_{n+k} = \{\neg u_k, \neg v_k, y_{n+k}\}$.

The intuition behind the KBKF_n formulas is an induction principle. The initial clauses C_0 and C'_0 require that y_1 or y'_1 are false, and the final clauses C_{n+k} and C'_{n+k} require both y_n and y'_n to be true. However if the existential player sets either y_k or y'_k to false, then the universal player can ensure that either y_{k+1} or y'_{k+1} must also be false. The value of u_k required depends on which of y_k, y'_k is false, and these 2^n possible sequences of moves provide an exponential lower bound on the size of refutations of KBKF_n.

Theorem 3.4 ([9,72]). *The QBFs* KBKF_n *require exponential size proofs in* Q-Res*, and* KBKFd_n *require exponential size proofs in* QU-Res*.*

The $\forall$-reduction rule can also be used to lift other line-based propositional proof systems to sound and complete QBF proof systems, such as all $\mathcal{C}$-Frege systems [19]. Rather than removing literals from clauses, in general the $\forall$-reduction rule allows for restricting lines by a partial assignment to variables from the rightmost block if that block is universal (Figure 10). For a propositional proof system P, we denote the corresponding QBF proof system resulting from adding the $\forall$-reduction rule by P+$\forall$red. Notice that QU-Res coincides with Resolution +$\forall$red.

$\forall$-reduction:	$\dfrac{L}{L[\alpha]}$	$\mathsf{dom}(\alpha) \subseteq X_i$ where $i = \mathrm{lv}(L)$ and $\mathcal{Q}_i = \forall$

Fig. 10. The $\forall$-reduction rule

The $\forall$-reduction rule can even be used to extend algebraic proof systems such as CP and PCR (see Chapter 8) to the QBF proof systems CP+$\forall$red and PCR+$\forall$red respectively [24]. In these proof systems, which work over rings and fields larger than $\mathbb{Z}_2$, such as $\mathbb{Z}$ or $\mathbb{Q}$, the $\forall$-reduction rule requires the assignments to the universal variables to be Boolean, rather than any value in the field, to ensure soundness as a QBF proof system.

Strategy extraction Attempts have been made to lift propositional lower bound techniques to QBF proof systems, with mixed success. Feasible interpolation [76, 95] lifts circuit complexity lower bounds to proof complexity lower bounds by efficiently deriving interpolating circuits $C(\boldsymbol{b})$ from a refutation of $\phi(\boldsymbol{a}, \boldsymbol{b}) \wedge \psi(\boldsymbol{b}, \boldsymbol{c})$. These circuits $C(\boldsymbol{b})$ determine which of $\phi(\boldsymbol{a}, \boldsymbol{b})$ and $\psi(\boldsymbol{b}, \boldsymbol{c})$ is unsatisfiable for a given assignment $\boldsymbol{b}$. This technique has been successfully lifted to QBF proof systems [22]. Prover-Delayer games construct a two-player game on a formula, in which the number of points the Delayer can score provides a lower bound on proof size. Such games have been successfully employed to prove tree-like Resolution lower bounds [26, 27], and an adaptation of this method in [25] was used to show lower bounds on tree-like Q-Res and QU-Res. However the size-width relations for Resolution [13], in which a lower bound for Resolution proofs follows if the proof must contain a clause with many literals, hold only in certain very weak Q-Res systems, and are known to fail in general Q-Res and QU-Res [23, 41].

Conversely, there exist methods for proving QBF proof complexity lower bounds that have no analogues in propositional proof complexity. The most prominent of these is *strategy extraction*. Given a P+$\forall$red refutation π of a false QBF Φ, [19] exhibited a method for constructing Boolean circuits of size polynomial in $|\pi|$ such that these circuits compute a universal strategy which is winning on Φ. For a given proof system P, these circuits will be in a circuit class $\mathcal{C}_{\mathsf{P}}$ depending on P; in particular, in the case of $\mathcal{C}$-Frege +$\forall$red, the corresponding circuit class is $\mathcal{C}$ for all circuit classes introduced in Chapter 2. For Resolution, this strategy extraction is in $\mathbf{AC}^0_3$, circuits of depth 3. To find a lower bound for P+$\forall$red, it suffices to find a function f which requires large circuits in $\mathcal{C}_{\mathsf{P}}$, and to construct a QBF for which any winning universal strategy must play according to f.

Theorem 3.5 (Beyersdorff, Bonacina and Chew [19]). *Let π be a $\mathcal{C}$-Frege +$\forall$red refutation of a QBF Φ. There are circuits $C_i \in \mathcal{C}$ such that $|C_i| = |\pi|^{O(1)}$ and the C_i compute a winning universal strategy by setting $u_i = C_i(\alpha)$ for each universal variable u_i.*

The computation of this strategy begins by constructing for each universal variable u_i, a decision list. This list is constructed by taking each line of π in turn, and adding the line

$$\text{if } \neg L \text{ then } u_i \leftarrow b, \text{ else ...}$$

to the decision list if L is derived as $L'[u_i/b]$ for some previous line L', and concluding with the line $u_i \leftarrow 0$ to ensure that the decision list assigns a value to u_i. The construction of this decision list ensures that for any assignment α to the variables left of u_i, the restricted proof $\pi[\alpha, u_i/C_i(\alpha)]$ is a refutation of $\Phi[\alpha, u_i/C_i(\alpha)]$, and so the strategy is a winning strategy.

It only remains to construct a Boolean circuit from a decision list containing lines of the form 'if $\neg L_j$ then $u_i \leftarrow b_j$, else...' This is achieved by the circuit $\bigvee_{\{j:b_j=1\}}(\neg L_j \wedge \bigwedge_{k<j} L_k)$.

Using this technique, circuit lower bounds for $\mathbf{AC}^0$ and $\mathbf{AC}^0[p]$ have been lifted to proof size lower bounds in $\mathbf{AC}^0$-Frege +$\forall$red and $\mathbf{AC}^0[p]$-Frege +$\forall$red. The lower bound for $\mathbf{AC}^0[p]$-Frege +$\forall$red is particularly noteworthy, as no superpolynomial lower bounds are currently known for the corresponding propositional proof system $\mathbf{AC}^0[p]$-Frege.

An alternative approach to strategy extraction is to obtain the response $S_u(\alpha)$ directly from the proof by restricting the proof by α, initially introduced in [62]. This approach, which is detailed in Chapter 7, can also be leveraged to construct proof size lower bounds, as described in Chapters 7 and 8.

QU-Res, and more generally proof systems of the form P+∀red, are the main focus of this thesis. Other approaches to extending propositional proof systems, particularly Resolution, to QBFs have been considered. We describe some of these here. While we do not work with these systems, instead focussing on QU-Res and proof systems of the form P+∀red, several of the results we show have analogues in these systems, which we occasionally highlight.

Long distance Resolution To formally model certain aspects of QDPLL and QCDCL solving, particularly some unit propagations, it is useful to introduce an extension to Q-Res and QU-Res known as *long distance Q-Resolution* [8, 110]. Long distance Q-Resolution (LD-Q-Res) allows resolution steps which would introduce tautologies on a universal variable u whenever the pivot variable x satisfies $\mathrm{lv}(x) < \mathrm{lv}(u)$ (see Figure 11). Rather than include both literals, we merge them into the literal u^*. The literal u^* cannot appear in a clause alongside any other literal on u, but can be further merged with other literals on u, including other copies of u^*, in resolution steps on pivots left of u, and can be removed in a $\forall$-reduction step if it is the rightmost literal in a clause. If both clauses in a resolution step contain u^*, then we also require that the pivot be left of u.

Axiom:	$\dfrac{}{C}$	C a clause in the CNF
$\forall$-reduction:	$\dfrac{C \vee u}{C}$ $\quad$ $\dfrac{C \vee u^*}{C}$	$\mathrm{lv}(l) < \mathrm{lv}(u)$ for any literal $l \in C$
Resolution:	$\dfrac{C \vee U_1 \vee x \quad D \vee U_2 \vee \neg x}{C \vee D \vee U}$	$U = \{u^* \mid u \in \mathrm{var}(U_1)\}$

- If $l_1 \in C$ and $l_2 \in D$ with $\mathrm{var}(l_1) = \mathrm{var}(l_2) = v$ then $l_1 = l_2$ and $l_1 \neq v^*$
- $\mathrm{var}(U_1) = \mathrm{var}(U_2)$, and if $u \in \mathrm{var}(U_1)$ then u is universal and $\mathrm{lv}(x) < \mathrm{lv}(u)$
- If $l_1 \in U_1$ and $l_2 \in U_2$ with $\mathrm{var}(l_1) = \mathrm{var}(l_2) = u$ then $l_1 = u^*$ or $l_1 \neq l_2$

Fig. 11. The derivation rules of LD-Q-Res [8, 110]

The addition of long distance Resolution to Q-Res results in an exponentially stronger proof system, with polynomial size proofs of KBKF_n, however exponential lower bounds are still known for modified versions of the formulas KBKF_n. The proof system LQU$^+$-Res [9] extends LD-Q-Res by allowing both standard resolution and long distance resolution on universal pivots as well as

existential pivots, and thus also p-simulates QU-Res. Nonetheless, superpolynomial lower bounds are also known for LQU$^+$-Res [21, 52].

Expansion and instantiation Universal expansion is an alternative approach to QBF proof systems which does not use universal reduction. First introduced in the proof system ∀Exp+Res [70] to model CEGAR solving, axioms of universal expansion systems take an assignment τ to the universal variables such that τ falsifies all universal literals in the clause, and annotate each existential literal l in the clause with the restriction of τ to the variables left of l. Literals with different annotations are treated as different variables. As all literals in the introduced clauses are existential, the only derivation rule required is the resolution rule. These rules are given in Figure 12.

Axiom: $$\frac{}{\{l^{\tau_l} \mid l \in C, l \text{ existential}\}} \quad \begin{array}{l} C \text{ a clause in the CNF} \\ \tau \in \langle \mathcal{U} \rangle \text{ with } C[\tau] \neq \top \\ \tau_l = \{u/\tau(u) \mid u \in \mathcal{U}, \mathrm{lv}(u) < \mathrm{lv}(l)\} \end{array}$$

Resolution: $$\frac{C \vee x^\tau \qquad D \vee \neg x^\tau}{C \vee D}$$

Fig. 12. The derivation rules of ∀Exp+Res [70]

Although ∀Exp+Res does p-simulate tree-like Q-Res, it is incomarable with dag-like Q-Res; there are QBFs with polynomial size proofs in ∀Exp+Res which require exponential size proofs in Q-Res, and vice versa [21, 70]. Indeed, ∀Exp+Res is even incomparable with LQU$^+$-Res, demonstrating how the distinct approaches of QCDCL and CEGAR solving have both strengths and weaknesses relative to each other.

The system IR-calc [20] combines the power of these systems by only annotating literals with the negations of those universal literals which appear in that clause when introducing an axiom. IR-calc also allows instantiation by an assignment σ to a subset of the universal variables, which extends each annotation by $u/\sigma(u)$ for each $u \in \mathsf{dom}(\sigma)$ which does not already appear in the annotation. The annotation on a literal l still only contains variables left of l, and variables with different annotations are still considered distinct propositional variables.

This generalisation allows IR-calc to p-simulate Q-Res and ∀Exp+Res. By introducing annotations of the form $u/*$, analogous to the literals u^* in LD-Q-Res, we obtain the IRM-calc proof system, which further p-simulates LD-Q-Res. The full picture of simulations and separations for these Resolution-based QBF proof systems is given in Figure 13. Simulation of proof systems is transitive, and for any pair of proof systems in Figure 13 for which no simulation is shown, these proof systems are known to be incomparable.

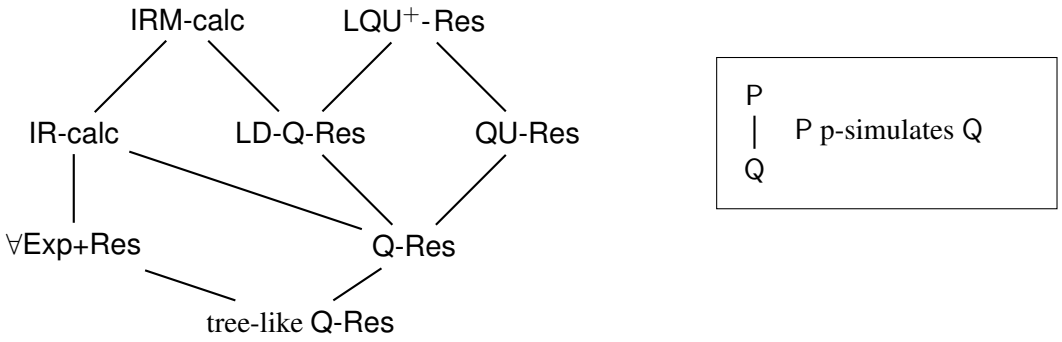

Fig. 13. A simulation diagram for Resolution-based QBF proof systems

Chapter 4

Relaxing QU-Res

Any QBF proof system restricts to a propositional proof system on existentially quantified formulas. Indeed, most QBF proof systems build on a propositional proof system by introducing a method of handling universal quantification, and hence a lower bound for the propositional proof system immediately translates into a lower bound for the QBF proof system. Furthermore, it is clear that even beyond purely propositional formulas, there are lower bounds containing universally quantified variables for which the hardness is a propositional phenomenon. Such lower bounds are somewhat unsatisfactory – they do not relate to the quantification of variables, which is the distinguishing feature of QBF proof complexity rather than propositional proof complexity – and so it is natural to ask whether we can construct a proof system or other technique to distinguish such lower bounds.

Relaxing QU-Res [35] has been proposed as just such a system. Using oracles for Σ_k^{p} for some constant k, relaxing QU-Res allows the introduction of axioms not given as clauses of a CNF matrix in such a way that any propositional formula can be solved instantly. It was therefore argued that relaxing QU-Res, when given access to Σ_1^{p}-oracle, is able to make the distinction between propositional hardness and hardness based on quantifier alternation. A lower bound for relaxing QU-Res based on quantified Boolean circuits was subsequently given, from which it can be inferred that this lower bound is dependent on quantifier alternation.

In this chapter, we present a lower bound for relaxing QU-Res which arises only due to a propositional lower bound on QU-Res. These formulas are based on both the pigeonhole principle formulas, a propositional QU-Res lower bound, and the KBKF_n formulas, which are easy for QU-Res but where the falsity relies on a large number of quantifier blocks. We provide a novel method for combining two QBFs in such a way that we can simply add lower bounds for the base formulas to provide a lower bound for the new formula. This combination of QBFs has the effect of 'hiding' the hard propositional part behind this large number of quantifier alternations to prevent relaxing QU-Res from using a Σ_k^{p}-oracle to solve the pigeonhole principle efficiently.

We first give an overview of relaxing QU-Res in Section 4.1. In Section 4.2 we present our method of combining QBFs, and then prove our lower bound based on propositional hardness.

4.1 The issue of propositional lower bounds for QBF proof systems

In this section, we introduce the system relaxing QU-Res, proposed by Chen in [35] in response to the issue of propositional lower bounds in QBF proof systems. We also define some necessary related notions dealing with the alternation of quantifiers in the prefix of QBFs.

The majority of QBF proof systems which have been defined build on a propositional proof system, such as the construction of QU-Res from Resolution, or more generally through the use of universal reduction or universal expansion. Proof systems defined in this way reduce to this original propositional proof system on Σ_1^b-formulas, which are equivalent to instances of the satisfiability problem. Indeed, for any QBF proof system P, there is a corresponding propositional proof system obtained by restricting P to Σ_1^b-formulas, and any lower bound for this propositional proof system is also a lower bound for the QBF proof system P.

Such propositional lower bounds are not limited to Σ_1^b-formulas. A simple example such as the conjunction of the pigeonhole principle with some true QBF shows that the presence of universal variables alone is not sufficient to ensure they contribute to the hardness of a formula. However, QBF lower bounds which arise only due to a lower bound on a propositional proof system are somewhat unsatisfactory. They belong in the realm of propositional proof complexity and do not enhance our understanding of how the proof system handles the universal quantification of variables, a feature which is fundamental to the increased complexity of QBFs. It is therefore desirable to be able to easily identify such propositional lower bounds, even when the formula itself is not purely propositional, and to construct systems in which all lower bounds must be 'genuine' QBF lower bounds, relying on the alternation of quantifiers in the prefix.

Relaxing QU-Res The issue of propositional lower bounds for QBF proof systems has been raised several times previously, in particular by Chen in [35]. This motivates the definition of a set of new proof systems called *relaxing QU-Res*. The goal of relaxing QU-Res is first and foremost to construct a system in which all lower bounds are 'genuine' QBF lower bounds, in the sense that they are not derived from propositional hardness. This system is then extended to attempt to capture the extent to which the alternation of quantifiers influences the lower bound.

Before giving a full definition of relaxing QU-Res, we must first introduce the notion of a relaxation. Relaxations allow for a reduction in the number of quantifier alternations in the prefix of a QBF in such a way that no true QBF becomes false under this transformation.

Definition 4.1 (Relaxation). *Let $\Pi = \mathcal{Q}_1 x_1 \dots \mathcal{Q}_n x_n$ be a quantifier prefix, with each $\mathcal{Q}_i \in \{\exists, \forall\}$. For a permutation $\pi : [n] \to [n]$, the prefix $\Pi' = \mathcal{Q}_{\pi(1)} x_{\pi(1)} \dots, \mathcal{Q}_{\pi(n)} x_{\pi(n)}$ is a* relaxation *of Π if for all $i < j$ with $\mathcal{Q}_i = \forall$ and $\mathcal{Q}_j = \exists$, we also have $\pi(i) < \pi(j)$. That is, no universal variable moves to the right of an existential variable.*

We refer to a relaxation which results in a Π_k^b-prefix as a Π_k^b-relaxation. The key property of relaxations we are interested in is that true QBFs remain true under any relaxation. This is

straightforward to verify, and so if any relaxation of a QBF is false, we can therefore conclude that the original QBF is also false.

Lemma 4.2 (folklore). *Let $\Phi = \Pi \cdot \phi$ be a QBF. If there is a relaxation Π' of Π such that $\Pi' \cdot \phi$ is false, then Φ is false.*

Proof. Since $\Phi' = \Pi' \cdot \phi$ is false, there is a winning strategy σ for the universal player on Φ'. For each universal variable u, all existential variables which are left of u in Π' must have been left of u in Π. Therefore σ is also a strategy for Φ, and σ is winning for the universal player. □

In the context of relaxing QU-Res, we also need to introduce a slightly altered definition of the restriction of a QBF by a partial assignment. The primary difference is that here we define a restriction such that it also alters the prefix of the QBF. This construction is only used in Sections 4.1 and 4.2, in which we discuss relaxing QU-Res; elsewhere, when we restrict a QBF $\Phi = \Pi \cdot \phi$ by α, we define $\Phi[\alpha] := \Pi \cdot \phi[\alpha]$ as usual.

To define a restriction for the purposes of relaxing QU-Res, let $\Phi = \Pi \cdot \phi$ be a prenex QBF, where $\Pi = \mathcal{Q}_1 X_1 \dots \mathcal{Q}_n X_n$, and let α be a partial assignment to the variables of ϕ. We define the restriction of Φ by α to be $\Phi[\alpha] = \Pi[\alpha] \cdot \phi[\alpha]$, where $\phi[\alpha]$ is the usual restriction of a propositional formula by a partial assignment[1]. Define the prefix $\Pi[\alpha] = \mathcal{Q}'_1 X'_1 \dots \mathcal{Q}'_n X'_n$, where $X'_i = X_i \setminus \mathsf{dom}(\alpha)$ and $\mathcal{Q}'_i = \exists$ for any $i < \max\{\mathrm{lv}(x) \mid x \in \mathsf{dom}(\alpha)\}$, otherwise $\mathcal{Q}'_i = \mathcal{Q}_i$. Described in a more intuitive fashion, $\Pi[\alpha]$ removes from Π any variables which are in $\mathsf{dom}(\alpha)$, and switches any universal quantifiers which are strictly left of a variable in $\mathsf{dom}(\alpha)$ to existential quantifiers.

Observe that there is a natural correspondence between clauses and partial assignments. We denote by $\mathrm{clause}(\alpha)$ the largest clause falsified by the partial assignment α, and conversely denote the smallest partial assignment falsifying a clause C by $\mathrm{assign}(C)$. It is then possible to use these restrictions of QBFs to verify that certain clauses are entailed by a QBF Φ.

Proposition 4.3 (Chen [35]). *Let $\Phi = \Pi \cdot \phi$ be a QBF and α be a partial assignment to the variables of Φ. If $\Phi[\alpha]$ is false, then any winning existential strategy for Φ satisfies* $\mathrm{clause}(\alpha)$*, i.e. $\Phi \models \Pi \cdot \mathrm{clause}(\alpha)$.*

For any such assignment α, $\mathrm{clause}(\alpha)$ can therefore be thought of as an 'axiom' of Φ, in the sense that if the existential player can win the game on Φ, then they can win the game on $\Pi \cdot \phi \wedge \mathrm{clause}(\alpha)$, and so the two formulas are equivalent. Determining the truth value of the QBF $\Phi[\alpha]$ would in general require access to a $\mathbf{PSPACE}$-oracle. However by using Π^b_k-relaxations of $\Phi[\alpha]$, which do not translate a true QBF to a false QBF, we can limit this oracle to a fixed level of the polynomial hierarchy. This results in the axiom set

$$H(\Phi, \Pi^b_k) = \{\mathrm{clause}(\alpha) \mid \text{ there is a false } \Pi^b_k\text{-relaxation of } \Phi[\alpha]\}.$$

[1] Generally, $\Phi[\alpha]$ is defined only by restricting ϕ to $\phi[\alpha]$ and removing the variables in $\mathsf{dom}(\alpha)$ from Π.

Notice that any clause C of ϕ is in $H(\Phi, \Pi^b_k)$ for any k, since $C[\text{assign}(C)] = \bot$, and hence the empty clause is in $\phi[\text{assign}(C)]$. As a result, $\Phi[\text{assign}(C)]$ is false under any relaxation. We can see from Proposition 4.3 and Lemma 4.2 that introducing any axiom from $H(\Phi, \Pi^b_k)$ is sound. If a relaxation of $\Phi[\alpha]$ is false, then $\Phi[\alpha]$ is false and so Proposition 4.3 ensures introducing $\text{clause}(\alpha)$ as an axioms is sound.

Whether a given clause is in $H(\Phi, \Pi^b_k)$ can be determined by a $\Sigma^{\mathsf{P}}_{k+1}$-oracle. A proof system which can introduce any axiom in $H(\Phi, \Pi^b_k)$ for some fixed value of k is therefore sound. Allowing the use of any deduction rule of QU-Res ensures that it is also complete, since each clause of ϕ is in $H(\Phi, \Pi^b_k)$, and so any QU-Res proof is also a relaxing QU-Res proof.

Definition 4.4 (Chen [35]). *A* relaxing QU-Res *refutation of a QBF Φ is a QU-Res derivation of the empty clause $\bot$ from the axioms of $H(\Phi, \Pi^b_k)$ for some constant k.*

One consequence of allowing the introduction of axioms from $H(\Phi, \Pi^b_k)$ is that relaxing QU-Res can be used as a proof system even if the matrix of the QBF is not a CNF, but any Boolean circuit, as the $\Sigma^{\mathsf{P}}_{k+1}$-oracle can still verify whether there is a false Π^b_k-relaxation of $\Phi[\alpha]$ when $\phi[\alpha]$ is a Boolean circuit.

It is important to observe that unless $\Sigma^{\mathsf{P}}_{k+1} = \mathbf{P}$, relaxing QU-Res with the axiom set $H(\Phi, \Pi^b_k)$ is not a proof system in the formal sense of Definition 2.3, as there is no polynomial-time algorithm to check the membership of a given clause in $H(\Phi, \Pi^b_k)$. For convenience, we nonetheless refer to relaxing QU-Res as a proof system.

The question of proving lower bounds in relaxing QU-Res must also be carefully considered. For any given QBF Φ, the empty clause $\bot \in H(\Phi, \Pi^b_k)$ for sufficiently large k, which would give a constant size proof in relaxing QU-Res. It is therefore more sensible to consider relaxing QU-Res to be a collection of proof systems, one for each value of k. A family of QBFs Φ_n is then said to require relaxing QU-Res proofs of size $\Omega(f(n))$ if for any fixed value of k, relaxing QU-Res refutations of Φ_n from the axiom set $H(\Phi_n, \Pi^b_k)$ require size $\Omega(f(n))$. The first such superpolynomial lower bound for relaxing QU-Res was shown by Chen, also in [35].

Theorem 4.5 (Chen [35]). *Let $\phi_n(x_1, y_1, \ldots, x_n, y_n)$ be a Boolean circuit which is true if and only if $\sum_{i=1}^{n}(x_i + y_i) \not\equiv n \mod 3$. Then $\Phi_n = \exists x_1 \forall y_1 \ldots \exists x_n \forall y_n \cdot \phi_n$ is false for each n, and Φ_n requires proofs of size $2^{\Omega(n)}$ in relaxing QU-Res.*

Proof (Sketch). For large enough n, any Π^b_k-relaxation of Φ_n is true, since it requires at least two y_j to be to the left of x_j. The existential player then has a winning strategy by forcing $\sum_{i=1}^{n-1}(x_i + y_i) + x_n \equiv n + 1 \mod 3$.

Fix an oracle $\Sigma^{\mathsf{P}}_{k+1}$ and therefore an axiom set $H(\Phi_n, \Pi^b_k)$. Consider the restriction of Φ_n by any assignment α to at most $n - 2k$ variables. If there is more than one 'gap' in the assignment, then the formulas $\Phi_n[\alpha]$ is true by following a similar strategy to the case for a relaxation of Φ_n. The assignment α must therefore assign variables on the left of the quantifier prefix, and the resulting formula $\Phi_n[\alpha]$ is equivalent to Φ_m for some $m \geq 2k$, for which there is not false Π^b_k-relaxation.

Any clause derived by a relaxing QU-Res refutation therefore contains more than $n - 2k$ variables, and so a relaxing QU-Res refutation must contain $2^{\Omega(n)}$ axioms. □

This lower bound demonstrates that despite the apparent strength of relaxing QU-Res, being able to use oracles for level of the polynomial hierarchy, it is still possible to show lower bounds in this proof system. However, the lower bound is defined with the matrix in the form of a Boolean circuit, rather than as a CNF. Indeed, as we now show, it is not possible to construct a polynomial-size CNF which is equivalent to this circuit.

Lemma 4.6. *Any CNF $\psi_n(x_1, y_1, \ldots, x_n, y_n)$ equivalent to $\sum_{i=1}^{n}(x_i + y_i) \not\equiv n \mod 3$ must contain $2^{\Omega(n)}$ clauses.*

Proof. Let ψ_n be such a CNF on the $2n$ variables $x_1, y_1, \ldots, x_n, y_n$. For any assignment to any subset of $2n - 1$ variables, there is an assignment to the final variable such that ψ_n is satisfied, since if the sum of these $2n - 1$ variables is m, it cannot be the case that both $m \equiv n \mod 3$ and $m + 1 \equiv n \mod 3$. If there were a clause C in ψ_n containing fewer than $2n$ literals, then $\text{assign}(C)$ can be extended to an assignment to $2n - 1$ variables which falsifies ψ_n, a contradiction. Any clause in ψ_n must therefore contain literals on all $2n$ variables, i.e. there is a bijection between clauses in ψ_n and assignments falsifying ψ_n.

It remains only to show that there are $2^{\Omega(n)}$ assignments falsifying ψ_n. Let α be any assignment to the variables $x_1, y_1, \ldots, x_{n-1}, y_{n-1}$. By ensuring $x_n + y_n \equiv n - \sum_{i=1}^{n-1}(x_i + y_i) \mod 3$, which is always possible, we can extend α to an assignment falsifying ψ_n. Since there are 2^{2n-2} such partial assignments α, we can construct at least 2^{2n-2} distinct assignments falsifying ψ_n, and so ψ_n contains $2^{\Omega(n)}$ clauses. □

Given that the relaxing QU-Res proof system works with lines that are clauses, it is unconventional to provide such a lower bound consisting of circuits without a polynomial-size representation as a CNF. Indeed, comparing the proof of the lower bound of Theorem 4.5 with the proof of Lemma 4.6 suggests that the relaxing QU-Res lower bound primarily arises due to the lack of an efficient CNF representation of the formulas Φ_n. It would consequently be desirable to find a superpolynomial lower bound for relaxing QU-Res which can be expressed using a QBF with a polynomial-size CNF matrix.

4.2 A propositional lower bound for relaxing QU-Res

We now give an example of a family of QBFs which have a polynomial-size CNF matrix and which require exponential-size proofs in relaxing QU-Res. The formulas are constructed as a combination of hard propositional formulas and easy QBFs. Furthermore, this construction allows us to show that the lower bound is based entirely on a propositional lower bound for Resolution, demonstrating that relaxing QU-Res is not an adequate formalism to distinguish propositional lower bounds from 'genuine' QBF lower bounds which arise from quantifier alternation or otherwise.

Combining false QBFs We begin by defining a method for combining two false QBFs to construct a new false QBF. As it allows a lot of control over the properties of this new QBF, this method may be of independent interest for creating new families of hard QBFs.

Definition 4.7. *Let $\Phi = \Lambda(\boldsymbol{x}) \cdot \bigwedge_{i=1}^{n} C_i(\boldsymbol{x})$ and $\Psi = \Pi(\boldsymbol{z}) \cdot \bigwedge_{j=1}^{m} D_j(\boldsymbol{z})$ be QBFs consisting of quantifier prefixes Λ and Π over disjoint sets of variables $\boldsymbol{x}$ and $\boldsymbol{z}$ respectively, and with clauses C_i and D_j over $\boldsymbol{x}$ and $\boldsymbol{z}$ respectively. Let $\boldsymbol{z}_i$ be a fresh set of variables for each $1 \leq i \leq n$, and define*

$$\Phi \oplus \Psi := \Lambda(\boldsymbol{x})\Pi(\boldsymbol{z}_1) \dots \Pi(\boldsymbol{z}_n) \cdot \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{m} (C_i(\boldsymbol{x}) \vee D_j(\boldsymbol{z}_i)).$$

The intuition behind this construction is that each clause C_i in the matrix of Φ is replaced by $C_i \vee \Psi$, with the variables of the copies of Ψ mutually disjoint, and quantified after all variables of Φ. It is then relatively straightforward to verify that if both Φ and Ψ are false, then the combined QBF $\Phi \oplus \Psi$ is false.

Lemma 4.8. *The QBF $\Phi \oplus \Psi$ is false if and only if both Φ and Ψ are false.*

Proof. If Φ and Ψ are both false, then the universal player has winning strategies σ_Φ and σ_Ψ on Φ and Ψ respectively. Playing the universal variables in $\boldsymbol{x}$ according to σ_Φ will falsify some clause C_i of Φ. The universal player can then play the variables of $\boldsymbol{z}_i$ according to σ_Ψ, which will falsify some clause $D_j(\boldsymbol{z}_i)$. The clause $C_i \vee D_j(\boldsymbol{z}_i)$ is therefore falsified in the matrix of $\Phi \oplus \Psi$.

If Φ is true, then the existential player has a winning strategy for Φ, and can use this winning strategy to satisfy every clause C_i, and hence every clause of $\Phi \oplus \Psi$. Similarly, if Ψ is true, then the existential player can play according to the winning strategy for Ψ on every set of variables $\boldsymbol{z}_i$, satisfying the clause $D_j(\boldsymbol{z}_i)$ for every $1 \leq j \leq m$ and $1 \leq i \leq n$. □

The significant feature of combining QBFs in this way is that the size of proofs of the newly constructed QBF $\Phi \oplus \Psi$ can be fairly tightly bounded in terms of the size of proofs of Φ and Ψ. Not only does this allow the construction of QBFs which require proofs of a precise size, but also ensures that the reason for any proof size lower bounds for $\Phi \oplus \Psi$ can be easily understood in terms of corresponding lower bounds for Φ and Ψ.

Lemma 4.9. *Let P be a QBF proof system closed under restrictions to existential variables, and let $\Phi = \Lambda \cdot \bigwedge_{i=1}^{n} C_i$ and $\Psi = \Pi \cdot \bigwedge_{j=1}^{m} D_j$ be minimally unsatisfiable QBFs. Then*

$$\max(s_{\mathsf{P}}(\Phi), s_{\mathsf{P}}(\Psi)) \leq s_{\mathsf{P}}(\Phi \oplus \Psi) \leq O\left(s_{\mathsf{P}}(\Phi) + n \cdot s_{\mathsf{P}}(\Psi)\right).$$

Moreover, if P is QU-Res, then

$$s_{\mathsf{P}}(\Phi \oplus \Psi) = \Theta\left(s_{\mathsf{P}}(\Phi) + n \cdot s_{\mathsf{P}}(\Psi)\right).$$

Proof. Let π be a refutation of $\Phi \oplus \Psi$ and let σ be a winning strategy for the universal player on Φ. Restrict π by some assignment α to the existential variables of Φ, and the corresponding

response of σ to the universal variables. The resulting proof is at most as large as π, and is also a sound refutation, and so contains a refutation of some copy of Ψ. Restricting by some assignment to the existential variables of all the copies of Ψ, we restrict the proof such that it amounts to a refutation of Φ, possibly with some additional universal variables from Ψ quantified rightmost, and so $\max(s_{\mathsf{P}}(\Phi), s_{\mathsf{P}}(\Psi)) \leq s_{\mathsf{P}}(\Phi \oplus \Psi)$.

Since $\Phi \oplus \Psi$ can be refuted by first deriving each clause C_i from $\bigwedge_{j=1}^{m}(C_i(\boldsymbol{x}) \vee D_j(\boldsymbol{z}_i))$, which can be done in $O(s_{\mathsf{P}}(\Psi))$ for each clause, and then refuting $\bigwedge_{i=1}^{n} C_i(\boldsymbol{x})$ with size $s_{\mathsf{P}}(\Phi)$, we can find a refutation of $\Phi \oplus \Psi$ of size $O\left(s_{\mathsf{P}}(\Phi) + n \cdot s_{\mathsf{P}}(\Psi)\right)$.

As Φ is minimally unsatisfiable, we can find for each clause C_i some assignment α_i to the existential variables of Φ such that C_i is the only clause of Φ falsified by $\alpha_i \cup \sigma(\alpha_i)$. Restricting π by this assignment results in a refutation of $\Psi(\boldsymbol{z}_i)$, and so we can restrict π to a refutation of Φ, or to a refutation of $\Psi(\boldsymbol{z}_i)$ for any $1 \leq i \leq n$. In QU-Res, each Resolution step or $\forall$-reduction step is performed on only one variable, and so will remain as a Resolution or $\forall$-reduction step in at most one of the restrictions to proofs of $\Psi(\boldsymbol{z}_i)$ or of Φ, being replaced by a trivial or weakening step in all others. The size of a QU-Res proof of $\Phi \oplus \Psi$ is therefore at least $\Omega\left(s_{\mathsf{QU\text{-}Res}}(\Phi) + n \cdot s_{\mathsf{QU\text{-}Res}}(\Psi)\right)$. By the upper bound above, in the case of QU-Res we have $s_{\mathsf{P}}(\Phi \oplus \Psi) = \Theta\left(s_{\mathsf{P}}(\Phi) + n \cdot s_{\mathsf{P}}(\Psi)\right)$. □

A relaxing QU-Res lower bound With this technique for combining false QBFs, and having established precisely the size of proof required for QBFs constructed in this way, we are now in a position to define the QBFs which provide our lower bound on relaxing QU-Res. They are built from the pigeonhole principle formulas, which are known to be hard for Resolution, and the formulas KBKF_n (Definition 3.3). An exponential lower bound for QU-Res proofs of $\mathrm{PHP}_n \oplus \mathrm{KBKF}_n$ follows immediately from Lemma 4.9 and the lower bound for the pigeonhole principle (Theorem 2.7).

Corollary 4.10. *The QBFs* $\mathrm{PHP}_n \oplus \mathrm{KBKF}_n$ *require QU-Res proofs of size* $2^{\Omega(n)}$.

Since it is known that there are polynomial-size refutations of KBKF_n in QU-Res [109], Lemma 4.9 also makes clear that the reason for this lower bound on QU-Res proofs is solely due to the propositional lower bound for PHP_n. We would therefore expect that any proof system distinguishing propositional lower bounds from genuine QBF lower bounds would have short proofs of $\mathrm{PHP}_n \oplus \mathrm{KBKF}_n$. However, this is not the case in relaxing QU-Res, where we can show an exponential lower bound on the size of proofs for any fixed k.

Theorem 4.11. *The QBF* $\Phi_n = \mathrm{PHP}_n \oplus \mathrm{KBKF}_n$ *requires relaxing QU-Res proofs of size* $2^{\Omega(n)}$.

The proof of Theorem 4.11 is the focus of the remainder of this chapter. The proof essentially observes that combining the QBFs in this way prevents the pigeonhole principle from being solved immediately by the Σ_k^{P}-oracle by including a linear number of quantifier alternations to the right of these variables. Moreover, the linear number of quantifier blocks in KBKF_n are all essential to the

falsity of the formula, in the sense that relaxing the quantifers in KBKF_n in any way results in a true formula.

Lemma 4.12. *Any relaxation of the quantifier prefix of* KBKF_n *to a* Π^b_k *prefix results in a true QBF, for any* $k < n$.

Proof. In any Π^b_k-relaxation of the quantifier prefix of KBKF_n, if $k < n$ then there is some t such that either u_t is quantified existentially, or u_t is quantified to the left of y_t and y'_t. In either case, we can construct a winning strategy for the existential player with the new quantifier prefix.

First, suppose that some u_t is now existentially quantified. A winning strategy for the existential player is to play $y_i = 0$, $y'_i = 1$ for each $i \leq t$, and to play $y_j = y'_j = 1$ for each $t < j \leq n$. Finally, playing $y_{n+i} = 1$ for each i then satisfies every clause apart from $\{y_t, \neg u_t, \neg y_{t+1}, \neg y'_{t+1}\}$, which can be satisfied by playing $u_t = 0$.

Now suppose that there is some u_t which is universally quantified and to the left of y_t, y'_t. The winning strategy for the existential player is identical to the strategy above, except on the variables y_t and y'_t. After these restrictions, the restricted clauses not yet satisfied are $\{u_{t-1}, \neg y_t, \neg y'_t\}$, $\{y_t, \neg u_t\}$ and $\{y'_t, u_t\}$.

When assigning the variables y_t and y'_t, the existential player may observe the value of u_t, as it is quantified further left. The existential player can thus set $y_t = u_t$ and $y'_t = \neg u_t$. It is clear that this assignment will satisfy the three remaining clauses, and so completes a winning existential strategy on the relaxation of KBKF_n. □

We now introduce some notation to allow us to more easily talk about the structure of proofs of Φ_n. We use the terms X-variables and Z-variables to refer to any variables in $\boldsymbol{x}$ and in $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m$ respectively. Given a clause C in the variables of Φ_n, we define C^X to be the restriction of C to the literals on X-variables, and similarly C^Z to be the restriction to literals on Z-variables. We refer to these as X-clauses and Z-clauses respectively, and observe that $C = C^X \vee C^Z$. We extend this to restrictions of proofs, denoting by $\pi^X = \{C^X \mid C \in \pi\}$. We maintain the same partial order on these clauses, representing the structure of π, but do *not* assume that π^X is a sound proof.

To prove Theorem 4.11, we show that if the $\Sigma^{\mathsf{p}}_{k+1}$-oracle deriving axioms 'proves' a large part of the pigeonhole principle when deriving an axiom, it can only do so under a large restriction on the existential variables of the copies of KBKF. Thus a relaxing QU-Res proof of Φ_n must contain either a large part of a proof of the pigeonhole principle, or a large number of different restrictions on the copies of KBKF.

To this end, we first show that, for any clause A derived as an axiom by relaxing QU-Res, if A^X requires at least c clauses from PHP_n to prove, then it also contains at least c existentially quantified Z-variables (Lemma 4.13). We then establish an upper bound on the size of a Resolution proof of an X-clause derived from c axioms of PHP_n which depends only on c (Lemma 4.14). Using this, we conclude that any relaxing QU-Res axiom where the corresponding X-clause requires proofs of size 2^m must contain $\Omega(m)$ Z-variables (Corollary 4.15).

Lastly, we show that given any relaxing QU-Res proof, for any assignment to the existential Z-variables, the set of axioms agreeing with this assignment constructs an unsatisfiable set of X-axioms (Lemma 4.16). Using these results, we conclude that the relaxing QU-Res proof must contain either a refutation of X-axioms of size $2^{\Omega(n)}$, or $2^{\Omega(n)}$ axioms corresponding to different assignments to Z-variables.

We begin by showing that for any clause $A \in H(\Phi_n, \Pi^b_k)$ with $k < n$, A^Z must contain an existential variable from $\boldsymbol{z}_i$ for each clause C_i needed to derive A^X. This limits the use of the $\Sigma^{\mathsf{p}}_{k+1}$-oracle in deriving X-clauses, as in order to derive an X-clause from a large number of pigeonhole principle axioms, we must make a correspondingly large restriction to the Z-variables.

Lemma 4.13. *Suppose that the clause $A = A^X \vee A^Z$ is derived as an axiom of Φ_n by relaxing QU-Res using a $\Sigma^{\mathsf{p}}_{k+1}$-oracle, i.e. $A \in H(\Phi_n, \Pi^b_k)$ for some $k < n$. Let $\boldsymbol{z}_{i_1}, \ldots, \boldsymbol{z}_{i_l}$ be such that each existential variable in A^Z is in some $\boldsymbol{z}_{i_j}$. Then $C_{i_1} \wedge \cdots \wedge C_{i_l} \models A^X$ for the corresponding pigeonhole principle axioms $C_{i_1}, \ldots, C_{i_l}$.*

Proof. Suppose that $C_{i_1} \wedge \cdots \wedge C_{i_l} \not\models A^X$, that is, there is some assignment to the X-variables which falsifies A^X but satisfies each C_{i_j}. We show that under this assumption, any Π^b_k-relaxation of $\Phi[\mathrm{assign}(A)]$ is true, and thus any such clause A cannot be an axiom of $H(\Phi_n, \Pi^b_k)$.

Let $\alpha = \mathrm{assign}(A)$ be the unique assignment to the variables of A which falsifies A. In particular, the only X-variables assigned by α are those in A^X, and so α can be extended to an assignment α' which falsifies A^X (and A^Z) but satisfies each C_{i_j}. Since α' extends α only by assignments to X-variables, which are existentially quantified and remain so in any relaxation, it suffices to construct a winning strategy for the existential player on any Π^b_k-relaxation of $\Phi_n[\alpha']$. This can be extended to an existential winning strategy on a Π^b_k-relaxation of $\Phi_n[\alpha]$, by playing the variables in $\mathsf{dom}(\alpha') \setminus \mathsf{dom}(\alpha)$ according to α', and following the winning strategy on the remaining Π^b_k-relaxation of $\Phi_n[\alpha']$.

Given a Π^b_k-relaxation of Φ_n, with quantifier prefix $\boldsymbol{\mathcal{Q}}'$, we show by induction that for each t, we can construct a strategy σ_t which extends the assignment α' and is a winning existential strategy for $\boldsymbol{\mathcal{Q}}' \cdot \bigwedge_{i=1}^{t} \bigwedge_{j=1}^{m} (C_i(\boldsymbol{x}) \vee D_j(\boldsymbol{z}_i))$. The final strategy σ_n is then a winning strategy for $\Phi_n[\alpha']$.

Define $\sigma_0 := \alpha'$. This clearly satisfies the empty conjunction. For each $t \leq n$, we extend the strategy σ_{t-1} which is winning for $\bigwedge_{i=1}^{t-1} \bigwedge_{j=1}^{m} (C_i(\boldsymbol{x}) \vee D_j(\boldsymbol{z}_i))$ to obtain σ_t. It therefore suffices to find a strategy for the unassigned existential variables in $\boldsymbol{z}_t$ which satisfies $\bigwedge_{j=1}^{m} (C_t(\boldsymbol{x}) \vee D_j(\boldsymbol{z}_t))$. We divide into two possible cases:

- Suppose $t = i_j$ for some $1 \leq j \leq l$. Then α', and hence σ_{t-1}, already satisfies $C_t(\boldsymbol{x})$. Therefore σ_{t-1} satisfies $C_t(\boldsymbol{x}) \vee D(\boldsymbol{z}_i)$ for any D, and we can define σ_t to be any extension of σ_{t-1} to the existential variables of $\boldsymbol{z}_t$ in $\boldsymbol{\mathcal{Q}}'$.
- Suppose $t \neq i_j$ for any $1 \leq j \leq l$. By the definition of the i_j, A^Z does not contain any existential variables in $\boldsymbol{z}_t$, so α', and hence by construction σ_{t-1}, are not defined on any variables in $\boldsymbol{z}_t$ which were originally existentially quantified. $\boldsymbol{\mathcal{Q}}' \cdot \bigwedge_{j=1}^{m} D_j(\boldsymbol{z}_t)$ is therefore a Π^b_k-relaxation of

KBKF_n, possibly restricted by assignments to some universal variables. By Lemma 4.12 this QBF is true and we can find a winning strategy τ_t.
The strategies σ_{t-1} and τ_t are defined on disjoint sets of variables. Extend σ_{t-1} by τ_t to give σ_t, which is a winning strategy on $\mathcal{Q}' \cdot \bigwedge_{i=1}^{t} \bigwedge_{j=1}^{m} (C_i(\boldsymbol{x}) \vee D_j(\boldsymbol{z}_i))$.

The final strategy σ_n is therefore a winning strategy for the existential variables of the Π_k^b-relaxation of Φ_n, and σ_n extends the assignment α'. This suffices to show that the relaxation of $\Phi_n[\alpha']$ is true. Since α' extends α, the smallest assignment falsifying A, with assignments to the existential X-variables only, the strategy detailed here can be extended to a winning existential strategy for any Π_k^b-relaxation of $\Phi[\alpha]$ by assigning any variables in $\mathsf{dom}(\alpha') \setminus \mathsf{dom}(\alpha)$ according to α', and so any Π_k^b-relaxation of $\Phi[\alpha]$ is true. This does not satisfy the axiom derivation rules of relaxing QU-Res, and so $A = \mathrm{clause}(\alpha)$ cannot be derived as an axiom. By contraposition, if A is derived as an axiom of relaxing QU-Res, then it must be the case that $C_{i_1} \wedge \cdots \wedge C_{i_l} \models A^X$. □

To make use of this lemma, we now look at the clauses derivable from a given set of axioms of PHP_n. In particular, we show an upper bound on the length of a Resolution derivation of an X-clause derived from a fixed number of pigeonhole principle axioms.

Lemma 4.14. *Let C be an X-clause such that $C_1 \wedge \cdots \wedge C_t \models C$ for some axioms $C_1, \ldots, C_t$ of PHP_n. There is a Resolution proof of C from PHP_n of size at most 18^t.*

Proof. We show that without any instances of the weakening rule, which we can assume occurs only once as the final step if it is needed for the derivation of C, there are at most 18^t clauses that can be derived by Resolution from t axioms of PHP_n. Since a Resolution proof of C need only contain a subset of these clauses, any Resolution proof of C has size at most 18^t. This upper bound is far from tight, but is sufficient for the proof of Theorem 4.11.

All negative literals in PHP_n are in clauses of length 2. Given t clauses, there are therefore at most $2t$ variables x_i which appear in both positive and negative literals in the clauses $C_1, \ldots, C_t$. For each clause C_i, there is a subclause Y_i consisting of the literals of C_i whose negation does not appear in any other clause. Any clause derived by Resolution from $C_1, \ldots, C_t$ contains a subset of the clauses Y_i, and may contain each variable x_j as a positive literal, a negative literal or not at all. Thus the total number of clauses derivable in Resolution from $C_1, \ldots, C_t$ is at most $2^t \cdot 3^{2t} = 18^t$. Any Resolution derivation of C from $C_1, \ldots, C_t$ therefore has size at most 18^t. □

The combination of these two results produces the key observation required for the proof of the lower bound for relaxing QU-Res: if an X-clause derived as part of an axiom requires a large (exponential-size) derivation from PHP_n, then it must be derived under a large (linear-size) restriction on the Z-variables.

Corollary 4.15. *Let A be an axiom in $H(\Phi_n, \Pi_k^b)$ for some $k < n$, and let $s(A^X)$ be the size of the smallest Resolution derivation of A^X from PHP_n. The Z-clause A^Z contains at least $\frac{1}{\log 18} \log s(A^X)$ existential Z-variables.*

Proof. Suppose A^Z contains t existential Z-variables. By Lemma 4.13, we can find t clauses $C_{i_1}, \ldots, C_{i_t}$ such that $C_{i_1} \wedge \cdots \wedge C_{i_t} \models A^X$, and hence there is a Resolution derivation of A^X from PHP_n with size at most 18^t (Lemma 4.14). We conclude that $s(A^X) \leq 18^t$, and hence $t \geq \frac{1}{\log 18} \log s(A^X)$. □

Corollary 4.15 ensures that all X-axioms in π were derived under a partial Z-assignment, and the stronger the X-axiom, the larger the corresponding Z-assignment. The purpose of the following lemma is to observe that for any Z-assignment, π contains a refutation of the X-axioms which were derived under restrictions of that assignment.

Lemma 4.16. *Given a relaxing QU-Res proof π of Φ_n and an assignment α to the existential Z-variables of Φ_n, $\pi|_\alpha^X$ contains a Resolution refutation of $\{C^X \mid C \text{ is an axiom of } \pi \text{ and } C[\alpha] \not\equiv \top\}$, i.e. the X-axioms corresponding to axioms in π which agree with α.*

Proof. Consider $\pi|_\alpha$, the result of restricting π to those clauses which agree with α. This is a sound QU-Res refutation from the (restricted) relaxing QU-Res axioms, as QU-Res is closed under existential restrictions. We show by backwards induction on the structure of π that $\pi|_\alpha^X$ contains a Resolution refutation from the X-axioms.

- The empty clause is the root of a Resolution proof on the X-variables, and clearly agrees with α.
- Suppose a clause C is derived by a $\forall$-red step on a Z-variable u. Then clearly $C \vee u$ agrees with α if C agrees with α, since α does not assign u. Also $C^X = (C \vee u)^X$, so this is a sound step in a Resolution refutation.
- Suppose C agrees with α and C is derived from C_1 and C_2 by resolving on an X-variable x. Then $C_1^Z, C_2^Z \subseteq C^Z$, and so both C_1 and C_2 agree with α since C does so. Observe also that C^X is derived from C_1^X and C_2^X by a single Resolution step on x.
- Suppose C agrees with α and C is derived from C_1 and C_2 by resolving on a Z-variable z. Then at least one of C_1 and C_2 must agree with α, depending on the value of $\alpha(z)$. As $C_1^X, C_2^X \subseteq C^X$, we can derive C^X by a weakening step from whichever agrees with the Z-assignment. If z is universally quantified, then α agrees with both C_1 and C_2, so derive C^X by weakening from C_1^X.
- If C agrees with α and C is derived as an axiom, then C^X is an X-axiom of the form required.

This completes our induction, proving that the X-clauses of the clauses in π which agree with α contain a Resolution refutation of the X-axioms agreeing with α. □

We can now combine this with the previous results to prove Theorem 4.11 by observing that any X-axiom which proves a large proportion of PHP_n can only appear in a small number of proofs π_α^X.

Proof (of Theorem 4.11). Fix some constant k, and suppose that for any $n > k$, the length of the shortest relaxing QU-Res proof of Φ_n with axioms from $H(\Phi_n, \Pi_k^b)$ is $f(n)$. Let π be such a proof

with $|\pi| = f(n)$. Given an assignment α to the existential Z-variables, $\pi|_\alpha^X$ is a sound Resolution refutation of the X-axioms (Lemma 4.16), and has at most $f(n)$ axioms. Since any Resolution refutation of PHP_n requires proofs of size at least 2^{tn} for some constant t, there is some X-axiom B in $\pi|_\alpha^X$ which requires a Resolution derivation of size at least $\frac{2^{tn}-f(n)}{f(n)} = \frac{2^{tn}}{f(n)} - 1$. The X-axiom B in $\pi|_\alpha^X$ is a restriction of some axiom A in π. Since $A^X = B$, by Corollary 4.15, there is a constant c such that A contains at least $c(tn - \log f(n)) =: g(n)$ literals on existential Z-variables, all of which are falsified by α.

For every assignment α to the existential Z-variables, we can find such an axiom containing at least $g(n)$ existential Z-variables, all falsified by α. As each of these axioms can be falsified by at most a $2^{-g(n)}$ proportion of the possible assignments α, π must contain at least $2^{g(n)}$ axioms. A proof cannot contain more axioms than its length, so we conclude that $2^{g(n)} \leq f(n)$, i.e. $2^{ctn} \leq f(n)2^{c \log f(n)} = f(n)^{c+1}$ and so $f(n) = 2^{\Omega(n)}$ for any choice of the constant k. Thus relaxing QU-Res proofs of Φ_n require size $2^{\Omega(n)}$. □

This lower bound strengthens previous lower bounds for relaxing QU-Res by providing an exponential lower bound which has polynomial-size when represented as a prenex normal form QBF. Moreover, $\mathrm{PHP}_n \oplus \mathrm{KBKF}_n$ requires relaxing QU-Res proofs of size $2^{\Omega(n)}$ despite the corresponding lower bound for QU-Res arising purely as a result of the propositional lower bound on PHP_n, demonstrating that relaxing QU-Res does not adequately distinguish propositional lower bounds for QU-Res. In Chapter 5, we exhibit an alternative proof system which does achieve this distinction.

Chapter 5

Identifying lower bounds due to quantifier alternation

In Chapter 4, we considered relaxing QU-Res, and provided a lower bound based on propositional hardness to demonstrate that it was unable to distinguish such lower bounds from 'genuine' QBF lower bounds. We now introduce an alternative proof system, Σ_k^p-QU-Res, which also makes use of Σ_k^p-oracles. By using these oracles in a more natural way, Σ_k^p-QU-Res provides such a distinction, ensuring that no lower bounds are based on a propositional Resolution lower bound.

After defining the system Σ_k^p-QU-Res, and showing that it simulates relaxing QU-Res, we first focus on the simplest system Σ_1^p-QU-Res. The SAT problem of the satisfiability of propositional formulas is $\mathbf{NP}$-complete, and so it is natural to expect that access to an $\mathbf{NP}$-oracle should be sufficient to remove any lower bounds based on propositional hardness. We verify that this is indeed the case, and that Σ_1^p-QU-Res does indeed characterise 'genuine' QBF lower bounds. In doing so, we also find that lower bound techniques such as strategy extraction, which provide 'genuine' QBF lower bounds, still apply to Σ_1^p-QU-Res.

Expanding our focus to Σ_k^p-QU-Res in general, we show separations between the Σ_{2k+1}^p-QU-Res proof systems by providing lower bounds for Σ_k^p-QU-Res with a Σ_{k+2}^b-prefix. In particular, we demonstrate a technique for lifting lower bounds for Σ_k^p-QU-Res to Σ_{k+2}^p-QU-Res, while only adding two quantifier blocks to the prefix, allowing us to lift lower bounds for Σ_1^p-QU-Res to provide these lower bounds for any k.

Even though more powerful oracles result in a strictly more powerful proof system, we cannot always achieve short proofs for a family of QBFs simply by increasing k. We conclude this chapter by showing that the formulas KBKFd_n require large proofs in Σ_k^p-QU-Res for *any* fixed value of k. The proof observes that the QU-Res lower bounds for KBKFd_n shown in [9, 72] are a result of any QU-Res proof containing a complete binary tree of linear depth. We then show that since every alternation of quantifiers is necessary for the falsity of KBKFd_n, Σ_k^p-QU-Res cannot omit more than k consecutive levels from any branch of this tree.

Section 5.1 contains the definition of Σ_k^{p}-QU-Res and how it relates to relaxing QU-Res. We then focus on Σ_1^{p}-QU-Res and the difference between propositional and 'genuine' lower bounds in Section 5.2. In Section 5.3, we observe that it is sufficient to consider only Σ_k^{p}-QU-Res for odd k, and then show separations between any two such systems. The lower bound for KBKFd_n for all Σ_k^{p}-QU-Res proof systems is proved in Section 5.4.

5.1 A proof system characterising hardness due to quantifier alternation

In Chapter 4, we saw that relaxing QU-Res lower bounds do not characterise 'genuine' QBF lower bounds in the sense we would expect, as we constructed a lower bound for this system based on a propositional lower bound. We now define a family of proof systems, also making use of oracles for levels of the polynomial hierarchy, for which lower bounds do make this distinction. As we would expect, $\mathrm{PHP}_n \oplus \mathrm{KBKF}_n$ have polynomial-size proofs in these systems.

Definition 5.1. *A Σ_k^{p}-QU-Res proof of a QBF $\Phi = \Pi \cdot \phi$ is a derivation of the empty clause using any of the deduction rules of QU-Res (Figure 8), and the* Σ_k^{p}-derivation rule *(Figure 14).*

Σ_k^{p}-derivation:	$\dfrac{C_1 \quad \dots \quad C_l}{D}$	There is a Σ_k^b-relaxation Π' of Π such that $\Pi' \cdot \bigwedge_{i=1}^{l} C_i \models \Pi' \cdot D$

Fig. 14. The Σ_k^{p}-derivation rule

The definition of a Σ_k^b-relaxation we use in the Σ_k^{p}-derivation rule is as defined previously in Definition 4.1, but we further allow a relaxation to replace any universal quantifier with an existential quantifier. Any existential winning strategy for a QBF Φ will remain a winning strategy under such a replacement, by further assigning the formerly universal variable in some constant way. It follows that Lemma 4.2, that the relaxation of a true QBF is true, holds under this alternative definition. The benefit of this more general notion of relaxation will be apparent later, when we show that it eliminates the need for an analogously defined Π_k^{p}-QU-Res (Lemma 5.10).

As in the case of relaxing QU-Res, Σ_k^{p}-QU-Res is not a proof system in the formal sense of Definition 2.3, as the polynomial-time proof checking algorithm requires a Σ_k^{p}-oracle. Indeed, the construction of a proof system with polynomial-size proofs for any problem in SAT without the use of such an oracle would suffice to show that $\mathbf{NP} = \mathbf{coNP}$, a major open problem in computational complexity. Nonetheless, since they are both sound and complete, we refer to Σ_k^{p}-QU-Res systems as proof systems.

Theorem 5.2. *For any $k \in \mathbb{N}$, Σ_k^{p}-QU-Res is a sound and complete QBF proof system.*

Proof. Any QU-Res proof is also a Σ_k^p-QU-Res proof for any value of k. Since QU-Res is complete, so is Σ_k^p-QU-Res.

To show soundness, we show that we can replace any instance of a Σ_k^p-derivation with a QU-Res derivation. Doing so for all Σ_k^p-derivations constructs a (potentially much larger) QU-Res proof from a Σ_k^p-QU-Res proof. Since QU-Res is sound, this suffices to show that Σ_k^p-QU-Res is sound.

Any instance of the resolution rule in QU-Res is sound under any quantifier prefix. Given a relaxation Π' of a quantifier prefix Π, any universal variable in Π' is also universally quantified in Π. Moreover, any variable to the right of u in Π is also right of u in Π', so any sound $\forall$-reduction step on u under Π' is also sound under Π. With the weakening rule, QU-Res is implicationally complete, and so we can replace a Σ_k^p-derivation of D from $C_1, \ldots, C_l$ with a QU-Res derivation of D from $C_1, \ldots, C_l$ which is sound under some Σ_k^b-relaxation Π' of Π. This QU-Res derivation is also sound with the prefix Π, completing the proof. □

Notice that the proof of Theorem 5.2 does not depend on the propositional deduction rules available to QU-Res, only that QU-Res is implicationally complete. It is therefore straightforward to generalise Σ_k^p-QU-Res to define Σ_k^p-P+$\forall$red for any implicationally complete QBF proof system P+$\forall$red. We focus here on Σ_k^p-QU-Res, as QU-Res is one of the best studied QBF proof systems in which several lower bounds are known. Nevertheless, many of the following results have analogues in Σ_k^p-P+$\forall$red.

First, we prove that Σ_{k+1}^p-QU-Res p-simulates relaxing QU-Res when relaxing QU-Res introduces axioms from $H(\Phi, \Pi_k^b)$.

Theorem 5.3. *Σ_{k+1}^p-QU-Res p-simulates relaxing QU-Res with axiom set $H(\Phi, \Pi_k^b)$.*

Proof. Suppose π is a relaxing QU-Res refutation of the QBF $\Phi = \Pi \cdot \phi$ from the axioms of $H(\Phi, \Pi_k^b)$. Apart from the introduction of axioms, every line in π is derived from previous lines by a rule of QU-Res, and so the same deduction can be performed in Σ_{k+1}^p-QU-Res. It therefore suffices to show that any axiom of $H(\Phi, \Pi_k^b)$ can be derived from the clauses of ϕ by a Σ_{k+1}^p-derivation.

Let $C \in H(\Phi, \Pi_k^b)$ be introduced as an axiom in π. By the definition of $H(\Phi, \Pi_k^b)$, there is some Π_k^b-relaxation Π' under which $\phi[\mathrm{assign}(C)]$ is false. In the context of relaxing QU-Res, the prefix Π' is defined as a relaxation of an altered prefix Π_C. However, Π_C is obtained from Π only by switching universal quantifiers to existential ones, a process which is permitted in a relaxation. Π' is therefore also a relaxation of Π.

We know that $\Pi' \cdot \phi[\mathrm{assign}(C)] \models \bot$. Let Π'' be the prefix defined by existentially quantifying each variable in $\mathrm{var}(C)$ to the left of Π'. The prefix Π'' is a Σ_{k+1}^b-prefix, and is also a relaxation of Π, since a relaxation may always quantify variables existentially and move existential variables to the left. Doing so with the variables of $\mathrm{var}(C)$, then relaxing the remaining variables as in Π', constructs Π'' as a relaxation of Π. The QBF $\Pi'' \cdot \phi \wedge \neg C$ is false, since any winning existential strategy would first be required to play according to $\mathrm{assign}(C)$ to satisfy $\neg C$, and then the QBF

reduces to $\Pi'.\phi[\text{assign}(C)]$, which is false. We conclude that $\Pi''.\phi \wedge \neg C \models \bot$, and so $\Pi''.\phi \models \Pi''.C$. This is precisely what is required for C to be derived from ϕ by a single $\Sigma^{\mathsf{p}}_{k+1}$-derivation. □

These systems are directly comparable, as both require the use of a $\Sigma^{\mathsf{p}}_{k+1}$-oracle to check the correctness of proofs. Furthermore, we believe the Σ^{p}_{k}-derivation rule to be a simpler way to incorporate such an oracle into a line-based proof system, and hence that Σ^{p}_{k}-QU-Res is a more natural proof system to distinguish propositional lower bounds from those based on the alternation of quantifiers.

Having shown the soundness and completeness of Σ^{p}_{k}-QU-Res, and that it p-simulates relaxing QU-Res, we are now in a position to use Σ^{p}_{k}-QU-Res to characterise lower bounds resulting from the alternation of quantifiers rather than a propositional lower bound. We observe that any propositional lower bound, such as PHP_n for Resolution and QU-Res, has constant size proofs in Σ^{p}_{1}-QU-Res, as the empty clause can immediately be derived from the axioms. This naturally leads us to use Σ^{p}_{1}-QU-Res, and more generally Σ^{p}_{1}-P for a QBF proof system P, to characterise 'genuine' QBF lower bounds.

Definition 5.4. *We say that the QBFs Φ_n are* hard due to quantifier alternation *if Φ_n require superpolynomial-size proofs in Σ^{p}_{1}-QU-Res.*

A family of QBFs Φ_n is said to have alternation hardness Σ^{p}_{k} *if there are polynomial-size proofs of Φ_n in Σ^{p}_{k}-QU-Res, but any $\Sigma^{\mathsf{p}}_{k-1}$-QU-Res proofs of Φ_n require superpolynomial-size.*

Beyond the theoretical understanding of 'genuine' QBF lower bounds, Σ^{p}_{1}-QU-Res is also of practical interest. Recent success in SAT solving has led to some QBF solvers embedding a SAT solver as a black box [69, 102]. Proof systems of the form Σ^{p}_{1}-P model this technique, and may provide insights into the power and limitations of such QBF solvers.

5.2 Understanding Σ^{p}_{1}-QU-Res

We shall consider Σ^{p}_{k}-QU-Res for more general $k > 1$ in subsequent sections, but we first focus on the question of the size of proofs in Σ^{p}_{1}-QU-Res and their correspondence with 'genuine' QBF lower bounds.

As with relaxing QU-Res, any false family of propositional formulas, including those such as PHP_n which are hard for Resolution and QU-Res, have constant size refutations in Σ^{p}_{1}-QU-Res, as the empty clause can be derived immediately by a Σ^{p}_{1}-derivation. Propositional lower bounds for QU-Res need not have a Σ^{b}_{1}-prefix though, such as in the case of $\text{PHP}_n \oplus \text{KBKF}_n$. In contrast to relaxing QU-Res, these formulas also have short proofs in Σ^{p}_{1}-QU-Res.

Theorem 5.5. *There is a Σ^{p}_{1}-QU-Res refutation of $\Phi_n = \text{PHP}_n \oplus \text{KBKF}_n$ containing $O(n^3)$ lines.*

Proof. Let $\Phi_n = \Pi.\bigwedge_{i,j} C_i \vee D_j(\boldsymbol{z}_i)$, where C_i are the clauses of PHP_n over the variables $\boldsymbol{x}$, and D_j are the clauses of KBKF_n.

There is a QU-Res refutation of KBKF_n consisting of $O(n)$ lines [109]. Using this, we can construct a QU-Res derivation of C_i from $\bigwedge_j C_i \vee D_j(\boldsymbol{z}_i)$, since the variables $\boldsymbol{x}$ are quantified left of $\boldsymbol{z}_i$. There are $O(n^2)$ clauses in PHP_n, and so repeating this derivation for each clause C_i results in a derivation of $\bigwedge_i C_i$ containing $O(n^3)$ lines.

The CNF $\bigwedge_i C_i$ is unsatisfiable, as it consists of all clauses of PHP_n, and all variables in $\bigwedge_i C_i$ are existentially quantified. We can therefore derive the empty clause from $\bigwedge_i C_i$ in a single Σ_1^{p}-derivation step, under the unique Σ_1^{p}-relaxation of Π. □

As a consequence of this upper bound and Theorem 5.3, we see that not only does Σ_k^{p}-QU-Res p-simulate the corresponding relaxing QU-Res system, but is in fact exponentially separated from it. Moreover, these formulas demonstrate that relaxing QU-Res, despite oracle access to any fixed level of the polynomial hierarchy, cannot simulate even Σ_1^{p}-QU-Res.

A Σ_1^{p}-QU-Res lower bound We have seen that lower bounds for QU-Res based on propositional hardness do not give lower bounds for Σ_1^{p}-QU-Res. However, we can provide a lower bound on Σ_1^{p}-QU-Res from known QU-Res lower bounds. To do so, we prove that the strategy extraction technique for P+∀red proof systems [19] can also be applied to Σ_1^{p}-QU-Res, and indeed to Σ_1^{p}-P+∀red proof systems in general. Lower bounds for QU-Res obtained through this technique are therefore also genuine QBF lower bounds.

Lemma 5.6. *Σ_1^{p}-QU-Res admits strategy extraction by depth-3 Boolean circuits.*

Proof. QU-Res is known to have strategy extraction by depth-3 Boolean circuits [19]. We extend this result to Σ_1^{p}-QU-Res by showing that Σ_1^{p}-derivations do not contain any information on the strategy for the universal player.

Given any Σ_k^{p}-QU-Res proof, using the inferential completeness of QU-Res, we can replace each Σ_k^{p}-derivation with a QU-Res derivation consistent with the Σ_k^b-relaxation. This expansion of a Σ_k^{p}-QU-Res proof to a QU-Res proof is described in detail in the proof of Theorem 5.2 to demonstrate the soundness of Σ_k^{p}-QU-Res.

In the case of Σ_1^b, the relaxation of the prefix treats all variables as existential. A QU-Res proof constructed in this way, while potentially much larger than the Σ_1^{p}-QU-Res proof, does not contain any additional ∀-reduction steps that were not in the Σ_1^{p}-QU-Res proof. Strategy extraction for QU-Res, as defined in [19], constructs a depth-3 Boolean circuit which is polynomial in the number of ∀-reduction steps in the proof, and uses only these steps in the construction. Given any Σ_1^{p}-QU-Res proof, the same strategy extraction algorithm will therefore still produce a winning strategy for the universal variables as a depth-3 Boolean circuit with size polynomial in the length of the proof. □

The key feature of strategy extraction that allows us to extend it from QU-Res to Σ_1^{p}-QU-Res is that lower bounds arising from strategy extraction provide not only lower bounds on the size of a QU-Res proof, but also lower bounds on the number of ∀-reduction steps in the proof. The proof

of Lemma 5.6 observes that given a Σ_1^p-QU-Res proof, we can construct a QU-Res proof with the same set of $\forall$-reductions. This gives a simple and elegant sufficient condition for 'genuine' QBF lower bounds for P+$\forall$red systems: a lower bound on the total size of the $\forall$-reduction steps.

This condition allows us to immediately transfer known strategy extraction lower bounds from QU-Res to Σ_1^p-QU-Res. For this purpose, we use the QPARITY_n formulas from [21].

Definition 5.7 (Beyersdorff et al. [21]). *Define the QBF* $\text{QPARITY}_n = \exists x_1 \dots x_n \forall z \exists t_2 \dots t_n \cdot \phi$*, where the CNF* ϕ *is equivalent to* $(t_2 \leftrightarrow x_1 \oplus x_2) \wedge \bigwedge_{i=3}^{n} (t_i \leftrightarrow t_{i-1} \oplus x_i) \wedge (z \not\leftrightarrow t_i)$.

The clauses in QPARITY_n force the existential player to play $t_j = \bigoplus_{i=1}^{j} x_i$ for each $2 \leq j \leq n$, and hence $t_n = \bigoplus_{i=1}^{n} x_i$. The unique winning move for the universal player is to play z according to the parity of the x_i. However QU-Res has strategy extraction in $\mathbf{AC}_3^0$, the class of depth-3 circuits, whereas it is known that the parity function is not in $\mathbf{AC}^0$ [1, 59, 66]. In fact, any constant-depth circuits computing parity require exponential size, and provide an exponential lower bound on Σ_1^p-QU-Res and even Σ_1^p-$\mathbf{AC}^0$-Frege +$\forall$red.

Theorem 5.8. *The parity formulas* QPARITY_n *require proofs of size* $2^{\Omega(n)}$ *in* Σ_1^p*-QU-Res.*

Observe that the prefix for QPARITY_n is a Σ_3^b-prefix, and so the empty clause is a member of the axiom set $H(\text{QPARITY}_n, \Pi_4^b)$ for relaxing QU-Res. This provides a separation of relaxing QU-Res from Σ_1^p-QU-Res. When combined with the converse separation given by $\text{PHP}_n \oplus \text{KBKF}_n$, we see that relaxing QU-Res and Σ_1^p-QU-Res are incomparable.

The Σ_3^b-prefix of QPARITY_n also ensures that there are constant size proofs of QPARITY_n in Σ_k^p-QU-Res for any $k \geq 3$. We further explore the relationships between different Σ_k^p-QU-Res systems in Section 5.3.

To conclude our discussion of Σ_1^p-QU-Res, we show that the lower bound on the size of $\forall$-reduction steps is necessary as well as sufficient for Σ_1^p-QU-Res lower bounds. This natural characterisation of Σ_1^p-QU-Res lower bounds demonstrates the effectiveness of this model for identifying 'genuine' QU-Res lower bounds. This will subsequently be emphasised further by Theorem 6.3, which shows that in stronger P+$\forall$red systems, this model does indeed precisely encapsulate lower bounds arising from quantifier alternation.

Theorem 5.9. *A superpolynomial lower bound on the total size of the* $\forall$*-reduction steps in a QU-Res refutation is necessary and sufficient for a superpolynomial lower bound on the size of a* Σ_1^p*-QU-Res refutation.*

Proof. Given any QU-Res proof, we can construct a Σ_1^p-QU-Res proof containing precisely the same $\forall$-reduction steps, and no further lines except axioms and the final line $\bot$, by replacing propositional subderivations by a Σ_1^p-derivation. Since the size of the axioms and of $\bot$ is polynomial, a lower bound on the total size of $\forall$-reduction steps in QU-Res is a necessary condition for a lower bound on Σ_1^p-QU-Res.

Given the smallest Σ_1^p-QU-Res refutation π, the total size of the $\forall$-reductions is at most $|\pi|$. By 'expanding' the Σ_1^p-derivations as in the proof of Theorem 5.2, we can construct a QU-Res

refutation containing precisely the same $\forall$-reduction steps. A superpolynomial lower bound on the total size of the $\forall$-reduction steps in QU-Res is thus sufficient to prove the same lower bound the size of a Σ^{P}_1-QU-Res refutation. □

In the case of QU-Res, since the lines in each $\forall$-reduction are at most linear in size, a superpolynomial lower bound on the size of $\forall$-reduction steps is equivalent to a lower bound on the number of $\forall$-reduction steps; this is not necessarily the case in some stronger proof systems such as Frege +$\forall$red. The equivalence given in Theorem 5.9 allows certain QU-Res lower bounds to be lifted to Σ^{P}_1-QU-Res; KBKFd_n is an example of such a lower bound, since the QU-Res lower bound proved in [9, 72] lower bounds the number of $\forall$-reduction steps. All subsequent lower bounds proved in this thesis will also be of this form, being lower bounds on the number or total size of the $\forall$-reduction steps. We therefore use both P+$\forall$red and Σ^{P}_1-P+$\forall$red in subsequent results, working with whichever is most convenient.

5.3 Separating Σ^{P}_k-QU-Res and $\Sigma^{\mathsf{P}}_{k+2}$-QU-Res

Having studied Σ^{P}_1-QU-Res in detail, we now turn out attention to Σ^{P}_k-QU-Res in general, in order to better understand the effect of quantifier alternation on proof size. We begin by observing that in order to determine the precise alternation hardness of a formula, we need only look at the proof systems $\Sigma^{\mathsf{P}}_{2k+1}$-QU-Res, as all others can be polynomially simulated by a proof system of this form at some lower level of the polynomial hierarchy.

Lemma 5.10. *Let Φ_n be a family of QBFs in n variables. If Φ_n has refutations of size $s(n)$ in Π^{P}_m-QU-Res or Σ^{P}_{2k}-QU-Res, then it has proofs of size $s(n) + n^2$ in $\Sigma^{\mathsf{P}}_{m-1}$-QU-Res or $\Sigma^{\mathsf{P}}_{2k-1}$-QU-Res respectively. In particular, if Φ_n has alternation hardness $\mathcal{C}$, then $\mathcal{C} = \Sigma^{\mathsf{P}}_{2k+1}$ for some $k \in \mathbb{N}$.*

Proof. We first show that we can construct a $\Sigma^{\mathsf{P}}_{m-1}$-QU-Res refutation of Φ_n with size $s(n) + n$ from a Π^{P}_m-QU-Res refutation with size $s(n)$. Without loss of generality, we assume that any Π^{P}_m-derivations in a proof derive the strongest possible clause under a given relaxation, in the sense that if a Π^{P}_m-derivation derives the clause C under a relaxation $\mathit{\Pi}$, then no subclause of C can be derived under the same relaxation.

Consider the leftmost block of universal variables in a $\mathit{\Pi}^b_m$-relaxation. If we replace the universal quantifiers in this block with existential quantifiers, we obtain a $\mathit{\Sigma}^b_{m-1}$-prefix, which is also a relaxation of the original prefix of Φ_n. We show that we can replace any Π^{P}_m-derivation in a proof by a $\Sigma^{\mathsf{P}}_{m-1}$-derivation, replacing the relaxation in this way.

If the Π^{P}_m-derivation does not derive the empty clause, then all clauses derivable under the $\mathit{\Pi}^b_m$-relaxation $\mathit{\Pi}$ contain a variable quantified existentially in $\mathit{\Pi}$. If a clause containing only universal variables were derivable under $\mathit{\Pi}$, then the empty clause could also be derived under $\mathit{\Pi}$, since a clause containing only universal variables in $\mathit{\Pi}$ can immediately be used to derive the empty clause under $\mathit{\Pi}$. Expanding the Π^{P}_m-derivation to a QU-Res derivation consistent with $\mathit{\Pi}$, this QU-Res

derivation therefore contains no $\forall$-reductions on the universal variables in the leftmost block. As a result, the same QU-Res derivation could derive the same clause under the Σ^b_{m-1}-relaxation Π', obtained from Π by also quantifying the leftmost block existentially.

If the Π^{p}_m-derivation does derive the empty clause, then under the Σ^b_{m-1}-relaxation, it is possible to derive a clause D containing only variables which were in the leftmost block of Π. Since all variables in the leftmost block of Π are universally quantified in the prefix of Φ_n, we replace the Π^{p}_m-derivation by a $\Sigma^{\mathsf{p}}_{m-1}$-derivation of D, followed by at most n $\forall$-reduction steps to derive $\bot$.

We now consider the case of a Σ^{p}_{2k}-QU-Res refutation of Φ_n. In any Σ^b_{2k}-relaxation, the rightmost block of variables is universally quantified. From the definition of a relaxation, we see that all variables in this block are also universally quantified in the rightmost block of the prefix of Φ_n. Such variables can be immediately removed from any axioms of Φ_n by a $\forall$-reduction. We can then follow the Σ^{p}_{2k}-QU-Res proof, removing from each clause any variables quantified rightmost in Φ_n. In each Σ^{p}_{2k}-derivation, we replace the Σ^b_{2k}-relaxation with a Σ^b_{2k-1}-relaxation by quantifying the rightmost block existentially.

The only levels of the polynomial hierarchy where we cannot use these reductions to construct a polynomially larger $\mathcal{C}$-QU-Res proof at a lower level are $\Sigma^{\mathsf{p}}_{2k+1}$ and Π^{p}_1. Observe that Π^{p}_1-QU-Res is equivalent to QU-Res, since any prefix with an existential variable cannot be relaxaed to a Π^b_1-prefix. If Φ_n has alternation hardness precisely $\mathcal{C}$, then we conclude $\mathcal{C} = \Sigma^{\mathsf{p}}_{2k+1}$ for some $k \in \mathbb{N}$. □

The effect of this result is to show that trailing universal literals at the beginning or end of a prefix do not contribute to any complexity arising out of quantifier alternation. In the case of universal variables rightmost in a prefix, it is clear that this should be the case, as we can assume that QU-Res performs any $\forall$-reductions possible at each step. Such variables can be removed directly from axioms and need not appear again. While P+$\forall$red in general cannot be assumed to perform $\forall$-reductions as soon as possible, this assumption can be made for any clauses of the matrix introduced as axioms in P+$\forall$red.

In the case of universal variables leftmost in a prefix, the QBF need only be disproved for one assignment to these variables, and so a refutation of Φ under this assignment suffices to refute Φ. The proof of Lemma 5.10 essentially formalises these arguments to show we need only use $\Sigma^{\mathsf{p}}_{2k+1}$-derivations in proof systems of the form Σ^{p}_m-P+$\forall$red.

Our first application of this is to immediately determine the alternation hardness of the QPARITY$_n$ formulas, which we know are hard for Σ^{p}_1-QU-Res.

Corollary 5.11. *The QBFs* QPARITY$_n$ *have alternation hardness* Σ^{p}_3.

Proof. Since QPARITY$_n$ has a Σ^b_3-prefix, there are short Σ^{p}_3-QU-Res refutations of QPARITY$_n$. By Lemma 5.10, the alternation hardness of QPARITY$_n$ is therefore either Σ^{p}_1 or Σ^{p}_3. Theorem 5.8 states that there are no polynomial size Σ^{p}_1-QU-Res proofs of QPARITY$_n$, so QPARITY$_n$ has alternation hardness Σ^{p}_3. □

It is clear that Σ^{p}_m-QU-Res p-simulates Σ^{p}_k-QU-Res for any $m > k$, since a Σ^{p}_k-derivation is also a sound Σ^{p}_m-derivation. If the proof systems Σ^{p}_k-QU-Res provide information on the role of quantifier alternation in lower bounds, it is reasonable to expect that these systems are not all equivalent, but rather provide a hierarchy of systems with increasing strength, i.e. that $\Sigma^{\mathsf{p}}_{k+2}$-QU-Res can be separated from Σ^{p}_k-QU-Res. The QPARITY$_n$ formulas provide such a separation between Σ^{p}_3-QU-Res and Σ^{p}_1-QU-Res.

To show such a separation for arbitrary k, we first need lower bounds for Σ^{p}_k-QU-Res. We have so far shown superpolynomial lower bounds only for Σ^{p}_1-QU-Res. To give lower bounds for Σ^{p}_k-QU-Res for $k > 1$, we give a technique for lifting lower bounds for Σ^{p}_k-QU-Res to $\Sigma^{\mathsf{p}}_{k+2}$-QU-Res. Using this technique, we can therefore construct superpolynomial lower bounds for Σ^{p}_k-QU-Res for any fixed value of k.

Theorem 5.12. *Let $k \geq 1$ be fixed, and let $\Phi_n = \Pi \cdot \bigwedge_{i=1}^n C_i$ be a minimally false QBF requiring superpolynomial-size refutations in Σ^{p}_k-QU-Res. For variables $\boldsymbol{a}, \boldsymbol{b}$ not appearing in Π, the QBFs*

$$\Phi'_n := \exists \boldsymbol{a} \forall \boldsymbol{b} \Pi \cdot \bigwedge_{i=1}^{n} ((a_i \vee b_i \vee C_i) \wedge (\neg a_i \vee \neg b_i \vee C_i))$$

are false and require superpolynomial-size refutations in $\Sigma^{\mathsf{p}}_{k+2}$-QU-Res.

To prove Theorem 5.12, we use restrictions of Σ^{p}_k-QU-Res proofs. Restrictions of QU-Res proofs are relatively straightforward, but it is prudent to observe that Σ^{p}_k-QU-Res proofs can be restricted in the same way.

Lemma 5.13. *Let π be a Σ^{p}_k-QU-Res refutation of a QBF $\Phi = \Pi \cdot \phi$. If α is a partial assignment to the variables of Φ, such that all variables in $\mathsf{dom}(\alpha)$ are either existential, or universal and quantified leftmost in Π, then $\pi[\alpha] = \{C[\alpha] \mid C \in \pi\}$ is a sound Σ^{p}_k-QU-Res refutation of $\Phi[\alpha]$.*

Proof. We proceed line by line in π. If a clause $C \in \pi$ is derived by a rule of QU-Res, then the derivation step deriving $C[\alpha]$ in $\pi[\alpha]$ is sound, since restrictions of QU-Res are sound.

If $C \in \pi$ is derived by a Σ^{p}_k-derivation from $C_1, \ldots, C_l$, then there is a QU-Res derivation π_C of C from $C_1, \ldots, C_l$ under some Σ^b_k-relaxation Π' of Π. Any variable in $\mathsf{dom}(\alpha)$ which is existential in Π is also existential in Π'. Any variable in $\mathsf{dom}(\alpha)$ which is universal in Π is either existential in Π', or is universally quantified leftmost, as relaxations can only move universal variables to the left. The restricted derivation $\pi_C[\alpha]$ is therefore a sound QU-Res derivation of $C[\alpha]$ from $C_1[\alpha], \ldots, C_l[\alpha]$ under the prefix $\Pi'[\alpha]$, a Σ^b_k-relaxation of $\Pi[\alpha]$. In particular $\Pi'[\alpha] \cdot C_1[\alpha] \wedge \cdots \wedge C_l[\alpha] \models \Pi'[\alpha] \cdot C_1[\alpha] \wedge \cdots \wedge C_l[\alpha] \wedge C[\alpha]$ and so the restricted Σ^{p}_k-derivation is also sound. □

We can now use these restrictions of $\Sigma^{\mathsf{p}}_{k+2}$-QU-Res proofs to prove Theorem 5.12.

Proof (of Theorem 5.12). To show Φ'_n is false, we construct a universal winning strategy by playing $b_i = a_i$ for every variable b_i. The resulting QBF is then $\Pi \cdot \bigwedge_i C_i = \Phi_n$. Since Φ_n is false, there is a universal winning strategy for Φ_n, which completes the universal winning strategy for Φ'_n.

For the lower bounds, let π be the smallest $\Sigma^{\mathsf{p}}_{k+2}$-QU-Res refutation of Φ'_n. We consider restricting π by an assignment α to the variables $\boldsymbol{a}$. We aim to show that for the restricted proof $\pi|_\alpha$, either we can find a clause in $\pi|_\alpha$ containing a large number of literals on $\boldsymbol{b}$ which 'agree' with α, or we can construct a Σ^{p}_k-QU-Res refutation of Φ_n with size $|\pi|^{O(1)}$. If the latter holds for any α, a superpolynomial bound follows immediately from the bound on Σ^{p}_k-QU-Res proofs of Φ_n. If not, a clause containing many literals on $\boldsymbol{b}$ can only agree with a small proportion of assignments to $\boldsymbol{b}$, and so π must contain a large number of such clauses.

Observe that if α is an assignment to the variables of $\boldsymbol{a}$, the restricted QBF $\Phi'_n[\alpha]$ contains precisely one of the clauses $b_i \vee C_i$ or $\neg b_i \vee C_i$. In particular, each variable of $\boldsymbol{b}$ appears in only one polarity in the clauses of $\Phi'_n[\alpha]$. As such, if $\pi|_\alpha$ is the restriction of π by α, we assume that the literals on $\boldsymbol{b}$ appear only in this polarity in $\pi|_\alpha$. If at any point the opposing literal is introduced into a clause of $\pi|_\alpha$, the proof remains sound if this literal is removed from the clause, as it is introduced as a weakening step, or in a $\Sigma^{\mathsf{p}}_{k+2}$-derivation in which the literal no longer appears in any antecedents.

The last step in any QU-Res refutation of $\Phi'_n[\alpha]$ consists of a $\forall$-reduction on all the variables $\boldsymbol{b}$, since these variables cannot be removed from a clause in any other way. The last step in $\pi|_\alpha$, a $\Sigma^{\mathsf{p}}_{k+2}$-QU-Res proof, is therefore either such a $\forall$-reduction, or a $\Sigma^{\mathsf{p}}_{k+2}$-derivation in which the corresponding Σ^b_{k+2}-relaxation quantified the variables of $\boldsymbol{b}$ universally. Such a relaxation corresponds to a Σ^b_k-relaxation on the variables of Φ_n, or a Π^b_{k+1}-relaxation, which by Lemma 5.10, we can replace with a Σ^b_k-relaxation.

For a proof π', define $g(\pi')$ to be the largest number of literals on $\boldsymbol{b}$ in any of the clauses immediately antecedent to the empty clause. We then define

$$f(n) = \min\{g(\pi|_\alpha) \mid \alpha \text{ is an assignment to } \boldsymbol{a}\}$$

Suppose that $f(n) = O(\log n)$, and let α be such that $g(\pi|_\alpha) = f(n)$. If the last step of $\pi|_\alpha$ were a $\forall$-reduction, then we have $f(n) = n$, so for sufficiently large n, the final step of $\pi|_\alpha$ is a $\Sigma^{\mathsf{p}}_{k+2}$-derivation. Let $D_1, \ldots, D_{m(n)}$ be the clauses of $\pi|_\alpha$ from which the empty clause is derived in this final $\Sigma^{\mathsf{p}}_{k+2}$-derivation.

Each clause D_j contains at most $f(n)$ literals on $\boldsymbol{b}$. As these literals cannot be removed in a QU-Res derivation before the final $\forall$-reduction step, each D_j can be derived from at most $f(n)$ clauses of $\Phi'_n[\alpha]$. We can therefore construct a QU-Res derivation π_j of D_j from $\Phi'_n[\alpha]$ with size $2^{O(f(n))} = 2^{O(\log n)}$. If β is the universal player's winning response to α, then $\pi_j[\beta]$ is a polynomial-size derivation of $D_j[\alpha, \beta]$ from $\Phi'_n[\alpha, \beta] = \Phi_n$.

The Σ^b_{k+2}-relaxation corresponding to the final $\Sigma^{\mathsf{p}}_{k+2}$-derivation in π must quantify the variables of $\boldsymbol{b}$ universally and consequently to the left of any existential variables in Π. On the variables of Π, this therefore restricts to a Σ^b_k-relaxation (or equivalently a Π^b_{k+1}-relaxation, cf. Lemma 5.10). We can then construct a Σ^{p}_k-QU-Res refutation of Φ_n, polynomial in n and $m(n)$, by first deriving each $D_j[\alpha, \beta]$ using $\pi_j[\beta]$, followed by a Σ^{p}_k-derivation deriving the empty clause. Since all such proofs are superpolynomial in n, we conclude that $m(n) \leq |\pi|_\alpha| \leq |\pi|$, is superpolynomial in n.

So now assume $f(n) = \omega(\log n)$. For any assignment α to the variables of $\boldsymbol{a}$, we can find a clause in π which contains $f(n)$ literals on $\boldsymbol{b}$ matching the universal response to α. From this, we conclude that $|\pi| \geq 2^{f(n)}$, and since $f(n) = \omega(\log n)$, we have $|\pi| \geq n^{\omega(1)}$ for any $\Sigma_{k+2}^{\mathsf{p}}$-QU-Res refutation π of Φ'_n. □

We can now combine the lifting result of Theorem 5.12 with the lower bound for Σ_1^{p}-QU-Res in Theorem 5.8 to construct lower bounds for Σ_k^{p}-QU-Res. Moreover, the formulas which give a lower bound for Σ_k^{p}-QU-Res have a Σ_{k+2}^b-prefix, allowing us to precisely determine their alternation hardness.

Theorem 5.14. *Define $\Phi_n^3 = \text{QPARITY}_n$, and for $\Phi_n^k = \Pi \cdot \bigwedge_i C_i$, define*

$$\Phi_n^{k+2} := \exists \boldsymbol{a} \forall \boldsymbol{b} \Pi \cdot \bigwedge_i ((a_i \vee b_i \vee C_i) \wedge (\neg a_i \vee \neg b_i \vee C_i))$$

where $\boldsymbol{a}, \boldsymbol{b}$ are fresh variables not in Π. For any odd $k \geq 3$, the QBFs Φ_n^k have alternation hardness Σ_k^{p}, i.e. any $\Sigma_{k-2}^{\mathsf{p}}$-QU-Res refutation of Φ_n^k requires superpolynomial size, but there are polynomial size Σ_k^{p}-QU-Res refutations of Φ_n^k.

Proof. The prefix of $\Phi_n^3 = \text{QPARITY}_n$ is a Σ_3^b-prefix, and so for each odd k, the prefix of Φ_n^k is a Σ_k^b-prefix. Σ_k^{p}-QU-Res can therefore refute Φ_n^k in a single Σ_k^{p}-derivation step. It therefore remains only to show a superpolynomial lower bound on $\Sigma_{k-2}^{\mathsf{p}}$-QU-Res proofs of Φ_n^k. We proceed by induction on k.

In the case of Φ_n^3, the alternation hardness is Σ_3^{p} (Corollary 5.11), with an exponential lower bound on Σ_1^{p}-QU-Res proofs shown in Theorem 5.8. Now assume for any odd k that Φ_n^k requires superpolynomial-size proofs in $\Sigma_{k-2}^{\mathsf{p}}$-QU-Res. Observe that Φ_n^{k+2} is minimally false, since Φ_n^k is minimally false, and hence we require both $a_i \vee b_i \vee C_i$ and $\neg a_i \vee \neg b_i \vee C_i$ for every clause C_i of Φ_n^k. It is then a simple application of Theorem 5.12 to show that any Σ_k^{p}-QU-Res proof of Φ_n^{k+2} has superpolynomial-size. □

The QBFs Φ_n^k demonstrate that the Σ_k^{p}-QU-Res proof systems form a family of increasingly powerful proof systems, but that we can nonetheless separate any one of these systems from the systems below it. This leads us to conclude that, at least in the case of QU-Res, the number of alternations of quantifiers plays an important role in determining the size of proofs.

5.4 A lower bound for all Σ_k^{p}-QU-Res systems

So far, every family of QBFs we have considered in the context of Σ_k^{p}-QU-Res has had quantifier prefixes in Σ_k^b for some constant value of k. As a result, all such QBFs have had short proofs in some Σ_k^{p}-QU-Res system for a sufficiently large value of k. In this section, we show that the KBKFd_n formulas, which have unbounded quantifier alternation, require large proofs in all Σ_k^{p}-QU-Res proof systems.

In the proof systems Σ_k^{p}-QU-Res, as with QU-Res, we can assume that any $\forall$-reductions are performed as early as possible in a proof, i.e. trailing universal literals in a clause are immediately removed by $\forall$-reduction to produce a strictly stronger clause. In the context of KBKFd_n, and other QBFs formed by 'doubling' universal variables, any Σ_k^b-relaxation can be assumed to quantify each pair of universal literals in the same block. No resolution steps are possible using either u_i or v_i as a pivot until one is $\forall$-reduced. If u_i and v_i appear in the same block of universal variables, when one variable is removed by $\forall$-reduction, the second can also be $\forall$-reduced. Consqeuently, restricting to only Σ_k^b-relaxations in which both variables are in the same block does not restrict which clauses can be derived by a Σ_k^{p}-derivation.

As a result, in this section we refer only to universal variables u_i, despite working with the QBFs KBKFd_n. The presence of literals on v_i is implicit wherever there is such a literal on u_i, and this prevents resolution on the pivots u_i or v_i. Similarly, assignments to variables u_i are assumed to also make the same assignment to the corresponding variables v_i.

Before we prove our lower bound for KBKFd_n, we first show the following lemma, which ensures that the first time u_i is removed from a clause in Σ_k^{p}-QU-Res, it occurs in a $\forall$-reduction or in a Σ_k^{p}-derivation where u_i is quantified to the right of y_i and y'_i.

Lemma 5.15. *Suppose the non-tautologous clause C is derived by QU-Res from KBKFd_n, and C contains a literal on u_i. If the QU-Res derivation does not contain a $\forall$-reduction step on u_i, then C contains a literal on y_j or y'_j for some $i \leq j \leq 2n$.*

Proof. Since no $\forall$-reduction step has taken place on u_i, no resolution steps on u_i are possible in the derivation of C. We assume without loss of generality that the literal on u_i is a positive literal; the case for $\neg u_i$ is similar.

Suppose first that the literal u_i is introduced by the axiom $u_i \vee y_{n+i}$. In this axiom, all existential variables are quantified to the right of u_i. No resolution steps on universal pivots are possible, and the only axiom not conflicting with u_i which contains existential variables to the left and to the right of u_i is the axiom $y'_i \vee u_i \vee \neg y_{i+1} \vee \neg y'_{i+1}$. Unless this axiom appears in the derivation of C, then we are done, as no variables left of u_i can appear, or C is a tautology.

So now assume u_i is introduced by $y'_i \vee u_i \vee \neg y_{i+1} \vee \neg y'_{i+1}$, and consider a clause derived from it. All axioms containing the literal $\neg y'_i$ also contain the literal $\neg y_i$. The only axiom containing y_i also contains $\neg u_i$. Since the literal $\neg u_i$ also cannot be removed from this clause by $\forall$-reduction or resolution, it must remain in the final clause C. Any clause derived from $y'_i \vee u_i \vee \neg y_{i+1} \vee \neg y'_{i+1}$ which does not contain a literal on y_i or y'_i is therefore a tautology.

In the case $i = n$, the initial axiom introducing u_n is $y'_n \vee u_n \vee \neg y_{n+1} \vee \cdots \vee \neg y_{n+n}$, however a similar argument applies to show that removing literals on y_n and y'_n requires introducing the literal $\neg u_n$ and hence constructing a tautology. □

We can now use this result to prove our lower bound for KBKFd_n. In effect, we have shown that each variable u_i needs to be quantified right of the corresponding y_i, y'_i variables. This ensures that it is not possible to 'skip' large parts of a QU-Res proof of KBKFd_n with a Σ_k^{p}-derivation.

Theorem 5.16. *The QBFs* KBKFd_n *require proofs of size* $2^{\Omega(n)}$ *in* Σ_k^{p}*-QU-Res for any constant* k*.*

Proof. Throughout this proof, we refer only to universal variables u_i. As discussed above, we assume that in any relaxation, the variables u_i and v_i are quantified identically and in the same block. Recall that this ensures there are no resolution steps possible on universal variables in a Σ_k^{p}-QU-Res or QU-Res refutation of KBKFd_n.

QU-Res (with weakening) is implicationally complete, and so from any Σ_k^{p}-QU-Res proof we can construct a QU-Res proof by replacing each Σ_k^{p}-derivation with an appropriate series of QU-Res steps such that the $\forall$-reduction steps replacing a given Σ_k^{p}-derivation are consistent with some Σ_k^b-relaxation of the quantifier prefix, as in the proof of Theorem 5.2. We show a lower bound on the size of a Σ_k^{p}-QU-Res refutation of KBKFd_n by examining the QU-Res proof we obtain in this way. As all universal variables in KBKFd_n appear with another universal variable of the same polarity, at no point can there be a resolution step on universal variables. Thus once a clause contains a universal variable, the only way it can be removed from descendants of this clause is by $\forall$-reduction.

As observed by [72], before a $\forall$-reduction step on any clause is possible, the clause must contain a literal on all universal variables. Furthermore, all 2^n possible sets of literals on all universal variables are necessary for the QU-Res refutation. In fact, a further consequence of this observation is that for any $\forall$-reduction step on u_i not preceded by another $\forall$-reduction on u_i, the clause also contains literals on $u_1, u_2, \ldots, u_{i-1}$.

Observe from Lemma 5.15 that if a $\forall$-reduction on u_i is not preceded by any other $\forall$-reduction on u_i, then the clause must contain some existential variable y_j or y'_j for some $i \leq j \leq 2n$. However, if the step is a $\forall$-reduction, this existential variable is either y_i or y'_i. As a result we conclude that, for each $1 \leq i \leq n$, and for each assignment β to the variables $u_1, \ldots, u_{i-1}$, a QU-Res proof of KBKFd_n contains a $\forall$-reduction on u_i containing literals on $u_1, \ldots, u_{i-1}$ agreeing with β.

Now suppose that π is a Σ_k^{p}-QU-Res proof of KBKFd_n for some fixed value of $k < n$. Let α be one of the 2^n possible assignments to the universal variables of KBKFd_n which the universal player may be required to play. We show that there is some clause in π which contains at least $n - k$ literals on universal variables and agrees with α.

Let π' be a QU-Res proof obtained by expanding the Σ_k^{p}-derivations of π. As observed above, given the assignment α, there is some clause $C_{n-k} \in \pi'$ which is derived by a $\forall$-reduction step on u_{n-k}, such that C_{n-k} is not preceded by any $\forall$-reduction steps on $u_1, \ldots, u_{n-k}$, and the universal literals in C_{n-k} agree with α. In particular, C_{n-k} contains literals on all universal variables left of u_{n-k}.

We look now at the derivation of C_{n-k} in π'. In this derivation, there must be some clause C_{n-k+1} derived by a $\forall$-reduction on u_{n-k+1} with no preceding such $\forall$-reduction. We construct clauses $C_{n-k+2}, \ldots, C_n$ similarly, choosing the first $\forall$-reduction on u_i in the derivation of C_{i-1}. Consider now the path through π' from C_n to C_{n-k} through each C_i. Since C_n contains literals on

all universal variables, and the universal literals of C_{n-k} agree with α, all clauses on this path must contain literals on $u_1, \ldots, u_{n-k-1}$ agreeing with α.

We show that at least one clause in this path must also be in π. If this were not the case, then $C_n, \ldots, C_{n-k}$ are all in the expansion of a single Σ_k^{p}-derivation. By the choice of C_i, each C_i contains a literal on y_i or y_i' by Lemma 5.15. The derivation of C_i by a $\forall$-reduction is therefore only possible if the corresponding Σ_k^b-relaxation quantifies u_i universally and to the right of y_i, y_i'. However if this were the case for each $n - k \leq i \leq n$, the relaxation would require at least $2k$ alternations of quantifiers, since each u_i must be left of y_{i+1}, y_{i+1}' by the definition of relaxation. Thus there is some clause D on the path from C_n to C_{n-k} such that $D \in \pi$ and D contains literals on $u_1, \ldots, u_{n-k-1}$ agreeing with α.

There are 2^{n-k-1} possible assignments to $u_1, \ldots, u_{n-k-1}$ that α could define, and for each there is a clause in π which contains literals on all of these variables agreeing with α. The size of any Σ_k^{p}-QU-Res proof is therefore at least $2^{\Omega(n)}$. □

This lower bound for any Σ_k^{p}-QU-Res proof system gives us a relatively complete understanding of the family of Σ_k^{p}-QU-Res proof systems. It is clear that increasing the value of k results in a proof system which is at least as strong, and the separation of Theorem 5.14 demonstrates that it is indeed strictly stronger. On the other hand, despite providing a lower bound for Σ_k^{p}-QU-Res for any k, the KBKFd_n formulas have short proofs in even very restricted Frege systems, such as $\mathbf{AC}_3^0$-Frege +$\forall$red. As a result, we conclude that no Σ_k^{p}-QU-Res proof system is able to simulate even $\mathbf{AC}_3^0$-Frege +$\forall$red, implying that the lower bound for QU-Res proofs of KBKFd_n relies not only on the alternation of quantifiers, but on the structure of the lines of QU-Res. This is an idea we shall return to in Chapter 8.

Chapter 6

A refinement of formalised strategy extraction

Having shown that we can identify propositional lower bounds via Σ_1^{p}-P+$\forall\mathsf{red}$, we turn our focus to understanding lower bounds which do not arise from propositional reasons. A new technique for lifting circuit complexity lower bounds to QBF proof complexity lower bounds was shown by Beyersdorff et al. [19] (cf. Theorem 3.5). This technique consists of using a proof in a given proof system P to efficiently construct circuits in a class $\mathcal{C}$ computing a winning strategy for the universal player. A lower bound on P-proofs of Φ can therefore be shown if any function computing a winning universal strategy requires large circuits in $\mathcal{C}$.

By formalising this technique for constructing winning strategies in the form of Boolean circuits, Beyersdorff and Pich [30] gave a normal form into which Frege +$\forall\mathsf{red}$ and eFrege +$\forall\mathsf{red}$ proofs can be efficiently transformed. This normal form reduces Frege +$\forall\mathsf{red}$ and eFrege +$\forall\mathsf{red}$ proofs to two stages: constructing circuits representing winning strategies, and propositionally refuting the formula when the universal variables are witnessed by these circuits. Lower bounds on such proofs are therefore a lower bound on one of these stages, corresponding to either a circuit complexity or a propositional proof complexity lower bound.

We improve the formalisation of the strategy extraction by choosing the witnessing circuits more carefully. Including the correct extension variables in the circuit allows us to replace an eFrege proof with a tree-like Resolution proof. This further simplifies the search for short proofs of QBFs, and moreover allows the characterisation of [30] to be applied to any proof system simulating tree-like Resolution. However, in doing so, we observe that in these systems it is not necessarily the case that the witnessed formula can be derived efficiently from the matrix, introducing a third possible type of lower bound.

We exemplify this additional cause for lower bounds by a common adaptation of the well known KBKF_n formulas of Kleine Büning et al. [72], which have a very simple winning strategy resulting in a witnessed formula with a short Resolution proof. We therefore conclude that the lower bound for $\mathsf{QU\text{-}Res}$ proofs of these formulas must fall into this third category. Finally, we observe

that several other QU-Res lower bounds are known via propositional lower bounds and circuit complexity, placing them naturally into the other categories.

In Section 6.1 we describe the dichotomy observed by Beyersdorff and Pich [30] for lower bounds in strong $\mathcal{C}$-Frege +$\forall$red systems. Our refinement of this to a trichotomy for weaker systems is then given in Section 6.2, followed by examples of lower bounds in each category for the QU-Res proof system.

6.1 The dichotomy for $\mathcal{C}$-Frege +$\forall$red

We begin by describing a previous characterisation of lower bounds in the relatively strong proof systems of Frege +$\forall$red and eFrege +$\forall$red. By appropriately formalising strategy extraction, [30] showed that lower bounds in these systems arise either from a propositional lower bound, or from a circuit complexity lower bound.

A QBF proof system P is said to have the *strategy extraction property* if for any QBF Φ of the general form $\forall x_1 \exists y_1 \ldots \forall x_n \exists y_n \cdot \phi(x_1, \ldots, x_n, y_1, \ldots, y_n)$, where ϕ is the propositional matrix, and any P-proof π of Φ, there are circuits C_i of size $|\pi|^{O(1)}$ which witness the existential quantifiers in Φ, i.e. the propositional formula

$$\bigwedge_{i=1}^{n} (y_i \leftrightarrow C_i(x_1, \ldots, x_i, y_1, \ldots, y_{i-1})) \rightarrow \phi(x_1, \ldots, x_n, y_1, \ldots, y_n) \tag{1}$$

is a tautology. The strategy extraction is Q-*formalised* if, in addition, the propositional formulas in (1) have $|\pi|^{O(1)}$-size proofs in a propositional proof system Q.

The QBF proof systems Frege +$\forall$red and eFrege +$\forall$red are known to have the strategy extraction property by Theorem 3.5, with the circuits known to be in $\mathbf{NC}^1$ and $\mathbf{P/poly}$ respectively. In [30], the strategy extraction for these proof systems was formalised in Frege and eFrege respectively. We state this formally for eFrege +$\forall$red:

Theorem 6.1 (Beyersdorff and Pich [30]). *Given an* eFrege +$\forall$red *refutation π of a QBF $\Phi = \exists x_1 \forall y_1 \ldots \exists x_n \forall y_n \cdot \phi$, it is possible to construct in $|\pi|^{O(1)}$ time an* eFrege *proof of*

$$\bigwedge_{i=1}^{n} (y_i \leftrightarrow C_i(x_1, \ldots x_i, y_1, \ldots, y_{i-1})) \rightarrow \neg\phi$$

for some circuits C_i.

Notice that since Frege +$\forall$red and eFrege +$\forall$red are refutational proof systems, a refutation of Φ is in fact a proof of $\neg\Phi$. As a result, the witnessing circuits C_i correspond to circuits describing a winning strategy for the universal player in Φ.

A normal form for proofs in these proof systems was given in [30], into which any proof can be efficiently transformed. This formalisation was then used to show that lower bounds for these Frege +$\forall$red systems must be either a circuit lower bound on the size of the witnessing circuits C_i,

or a lower bound on the corresponding propositional proof system for a derivation of the witnessed formula.

Theorem 6.2 (Beyersdorff and Pich [30]).

1. *There is a superpolynomial lower bound on* Frege +∀red *if and only if there is a superpolynomial lower bound on* Frege*, or a superpolynomial lower bound for* $\mathbf{NC}^1$.
2. *There is a superpolynomial lower bound on* eFrege +∀red *if and only if there is a superpolynomial lower bound on* eFrege*, or a superpolynomial lower bound on* $\mathbf{P/poly}$.

This characterisation of Frege +∀red and eFrege +∀red lower bounds gives an almost complete understanding of the proof complexity of these systems. In particular, to prove a superpolynomial lower bound on Frege +∀red without proving such a lower bound on the propositional Frege system would require a major breakthrough in circuit complexity, namely superpolynomial lower bounds on $\mathbf{NC}^1$.

The proof system Σ^{p}_1-Frege +∀red removes the possibility of a lower bound based on propositional hardness for Frege. We can therefore reframe the classification of Theorem 6.2 as an equivalence between Σ^{p}_1-Frege +∀red lower bounds and $\mathbf{NC}^1$ lower bounds.

Theorem 6.3. *A family of QBFs Φ_n require superpolynomial-size proofs in* Σ^{p}_1-Frege +∀red *(respectively* Σ^{p}_1-eFrege +∀red*) if and only if this lower bound is due to a lower bound on* $\mathbf{NC}^1$ *circuits (respectively* $\mathbf{P/poly}$ *circuits).*

Proof. We prove the statement for Frege +∀red, the case for eFrege +∀red is similar. It is simple to verify an analogue of Lemma 5.6 stating that Σ^{p}_1-Frege +∀red admits strategy extraction by $\mathbf{NC}^1$ circuits. A lower bound on Frege +∀red proofs of Φ_n due to an $\mathbf{NC}^1$-circuit lower bound therefore immediately lifts to a Σ^{p}_1-Frege +∀red lower bound.

Conversely, suppose that there are polynomial-size $\mathbf{NC}^1$ circuits C_i computing a winning strategy for Φ_n. We follow the normal form for Frege +∀red proofs described in [30]. The witnessed formula $\bigwedge_i (u_i \leftrightarrow C_i) \wedge \phi_n$ is deduced from ϕ_n using a single Σ^{p}_1-derivation. There is then a short Frege +∀red refutation of this witnessed formula, producing a short Σ^{p}_1-Frege +∀red refutation of Φ_n. □

This suffices to show that, in the case of those $\mathcal{C}$-Frege +∀red systems for which the dichotomy of Theorem 6.2 holds, Σ^{p}_1-$\mathcal{C}$-Frege +∀red precisely distinguishes propositional lower bounds from QBF lower bounds.

6.2 A trichotomy for weaker systems

In this section, we extend the characterisation of Theorem 6.2 to QBF proof systems weaker than Frege +∀red and eFrege +∀red. To do so, we begin by observing that with a careful choice of witnessing circuits, strategy extraction for Frege +∀red and eFrege +∀red can be formalised in

the relatively weak system of tree-like Resolution. In particular, we allow the witnessing circuits to contain additional gates computing any extension variables required in the eFrege proof of the witnessed formula.

Theorem 6.4. *Let $\mathcal{C}$ be the circuit class $\mathbf{NC}^1$ or $\mathbf{P/poly}$. Given a $\mathcal{C}$-Frege+$\forall$-red refutation π of a QBF*

$$\exists x_1 \forall y_1 \ldots \exists x_n \forall y_n \cdot \phi(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

where ϕ is a quantifier free CNF matrix, we can construct in time $|\pi|^{O(1)}$ a tree-like Resolution refutation of the witnessed formula

$$\bigwedge_{i=1}^{n} (y_i \leftrightarrow C_i(x_1, \ldots, x_i, y_1, \ldots, y_{i-1})) \wedge \phi(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

for some circuits $C_i \in \mathcal{C}$.

As with the characterisation in Theorem 6.2, this result easily generalises to further 'natural' circuit classes $\mathcal{C}$ such as $\mathbf{AC}^0$ or $\mathbf{TC}^0$. For clarity, we focus here on the two most interesting cases of $\mathbf{NC}^1$ and $\mathbf{P/poly}$, which correspond to Frege and eFrege systems respectively.

Proof. By the formalised strategy extraction theorem for $\mathcal{C}$-Frege systems [30] (Theorem 6.1), there is a $\mathcal{C}$-Frege proof of the witnessed formula (1) which has size $|\pi|^{O(1)}$. Given the equivalence of eFrege and tree-like extended Resolution [43, 78], this means there is a tree-like Resolution refutation of

$$Ext \wedge \bigwedge_{i=1}^{n} (y_i \leftrightarrow C_i(x_1, \ldots, x_i, y_1, \ldots, y_{i-1})) \wedge \phi(x_1, \ldots, x_n, y_1, \ldots, y_n) \tag{2}$$

of size $|\pi|^{O(1)}$, where Ext is a set of extension axioms defining $\mathcal{C}$ formulas on the variables $x_1, \ldots, x_n, y_1, \ldots, y_n$.

With the exception of those depending on y_n, the extenstion axioms of Ext can be encoded into circuits C_i with each extension variable represented by a possibly redundant gate of a circuit C_i. In order to remove the dependence of any extension variables on y_n, we take two independent tree-like Resolution refutations of (2). In one, we replace all occurrences of y_n in clauses of Ext with 0; in the other, occurrences of y_n in Ext are substituted by 1. This results in two derivations from (2), both at most as large as the original, one concluding with $\{y_n\}$ and the other with $\{\neg y_n\}$. Resolving on these two clauses we obtain the needed tree-like Resolution derivation without extension variables depending on y_n. □

The formalisation of strategy extraction given in Theorem 6.1 allowed the search for short proofs in QBF proof systems with strategy extraction to be reduced to a search for the correct witnessing circuits C_i, followed by finding an eFrege proof of the resulting witnessed formula. By formalising the strategy extraction in tree-like Resolution rather than Frege or eFrege, we can replace this latter

step with a search for a tree-like Resolution proof of the witnessed formula. Tree-like Resolution is known to be quasi-automatisable [11], i.e. for any false CNF, it is possible to construct a tree-like Resolution refutation in quasipolynomial time in the size of the shortest such proof. The problem of finding proofs in QBF proof systems with strategy extraction formalisable in tree-like Resolution is therefore essentially reduced to finding the correct witnessing circuits C_i.

As described above, [30] showed that any super-polynomial lower bound on eFrege $+\forall$red is either a super-polynomial circuit lower bound or a super-polynomial lower bound on eFrege. We now generalise this phenomenon to other QBF proof systems.

Let P be a refutational QBF proof system operating on clauses of matrices of (prenex normal form) QBFs which contains a resolution rule that allows resolution on both existential and universal variables. We say that a set of clauses C defines a formula $C_i(\boldsymbol{x}) = z$ for a circuit C_i with input variables $\boldsymbol{x}$ and output variable z if z appears in a literal of some clause in C and for any assignment of the input variables there is exactly one assignment of the remaining variables in $\mathrm{var}(C)$ satisfying all clauses in C.

Whenever a QBF Φ as above is hard for a QBF proof system P it is for one of the following reasons:

1. the existential quantifiers in Φ cannot be witnessed by circuits C_i such that formulas $\bigwedge_i C_i(x_1, \ldots, x_i, y_1, \ldots, y_{i-1}) = y_i$ have $|\phi|^{O(1)}$-size derivations from $\neg\phi$ in P.
2. the existential quantifiers in Φ are witnessable as in 1. but the witnessed formula

$$\bigwedge_{i=1}^{n} (y_i \leftrightarrow C_i(x_1, \ldots, x_i, y_1, \ldots, y_{i-1})) \wedge \neg\phi(x_1, \ldots, x_n, y_1, \ldots, y_n)$$

   is hard for Resolution.

By considering the first case more carefully, this characterisation can be specified further.

Theorem 6.5. *Let* P *be a refutational QBF proof system as above admitting strategy extraction by* $\mathcal{C}$ *circuits. If* $\Phi_n = \forall x_1 \exists y_1 \ldots \forall x_n \exists y_n.\, \phi_n(x_1, \ldots, x_n, y_1, \ldots, y_n)$ *are QBFs with propositional CNF matrix* ϕ_n*, which do not have polynomial-size proofs in* P*, then one of the following holds:*

1. ***Circuit lower bound.*** *The existential variables in* Φ_n *are not witnessable by polynomial-size* $\mathcal{C}$ *circuits.*
2. ***Resolution lower bound.*** *Condition 1. does not hold, but for all polynomial-size* $\mathcal{C}$ *circuits witnessing* Φ_n*, the witnessed formulas require super-polynomial size Resolution refutations.*
3. ***Genuine QBF hardness.*** *There are circuits* $C_i \in \mathcal{C}$ *witnessing* Φ_n *so that the witnessed formulas have polynomial-size Resolution refutations, but for all such circuits* C_i *it is hard to derive* $\bigwedge_i C_i(x_1, \ldots, x_i, y_1, \ldots, y_{i-1}) = y_i$ *from* $\neg\phi_n$ *in* P*.*

Proof. If the existential variables in Φ_n are not witnessable by polynomial-size $\mathcal{C}$ circuits, we are done. We therefore assume that there are small circuits in $\mathcal{C}$ witnessing the existential variables.

Suppose further that there are some circuits $C_i \in \mathcal{C}$ such that the witnessed formula (1) has a polynomial-size Resolution refutation. If this is not the case, we are done as we are in case 2.

We can construct a refutation of $\neg\Phi_n$ in P by first deriving $\bigwedge_i C_i(x_1, \ldots, x_i, y_1, \ldots, y_{i-1}) \leftrightarrow y_i$ from $\neg\phi_n$, and then refuting $\bigwedge_i (C_i \leftrightarrow y_i) \wedge \neg\phi_n$. Since the refutation of $\bigwedge_i (C_i \leftrightarrow y_i) \wedge \neg\phi$ is assumed to have polynomial-size, but any refutation of $\neg\Phi_n$ requires superpolynomial-size, it must be the case that for the circuits $C_i \in \mathcal{C}$, the derivation of $\bigwedge_i (C_i \leftrightarrow y_i)$ from $\neg\phi_n$ requires superpolynomial size (case 3). □

This means that any QBF lower bound on P is either a circuit lower bound, a propositional proof complexity lower bound, or a 'genuine' QBF proof complexity lower bound in the following sense: some small circuits witnessing the existential quantifiers in the original formula cannot be derived efficiently by P, and for any small witnessing circuits which P *can* derive efficiently, the witnessed formula is hard for Resolution.

Theorem 6.2 demonstrates that the last possibility does not happen in the case of strong systems like eFrege +∀red. The situation is, however, more delicate with weaker systems such as QU-Res, where we can encounter 'genuine' QBF lower bounds. We give an example, in the form of the formulas of Kleine Büning et al. (Definition 3.3).

The QBFs KBKF_n are known to require refutations of size $2^{\Omega(n)}$ in Q-Res (Theorem 3.4), and this bound can be extended to QU-Res using the formulas KBKFd_n, which were obtained by adding new universal variables v_k, quantified at the same level as u_k, and adding the literal v_k or $\neg v_k$ to each clause containing u_k or $\neg u_k$, respectively [9]. By showing that neither of the other two cases applies in the case of KBKFd_n, we show that this lower bound falls into third category of 'genuine QBF hardness' from Theorem 6.5. As a result, we see that at least some of the lower bounds in this third category are due to quantifier alternation, as Σ_1^{P}-QU-Res also requires large proofs of KBKF_n.

Theorem 6.6. *The formulas* KBKFd_n *are hard for* QU-Res *due to genuine QBF hardness (case 3 in Theorem 6.5).*

Proof. It is clear that playing the variables u_k and v_k identical to y'_k is a winning strategy for the universal player, and so there are circuits C_i as described in Theorem 6.5 which are of constant size. The QU-Res lower bound on KBKFd_n is therefore not an instance of case 1 in Theorem 6.5.

Looking now at the witnessed formula $\bigwedge_{i=1}^{n}((u_i \leftrightarrow y'_i) \wedge (v_i \leftrightarrow y'_i)) \wedge \phi$, we show this can be refuted by a linear-size proof. By resolving on each u_i and v_i to replace these with the relevant literal on y'_i, we obtain the clauses $y'_i \vee y_{n+i}$ and $\neg y'_i \vee y_{n+i}$, and thus the unit clause y_{n+i}. Resolving each such clause with C'_n and C_n produces the clauses y'_n and $y_n \vee \neg y'_n$ from which we also derive y_n. For each $1 \leq i \leq n$, we resolve the unit clauses y_i and y'_i with the axioms C_{i-1} and C'_{i-1} to deduce y_{i-1} and y'_{i-1} and finally resolve y_1 and y'_1 with C'_0 and C_0, completing the refutation.

Since KBKFd_n is known to require exponential size proofs in QU-Res [9], by Theorem 6.5, it must satisfy one of the three conditions given. We have established that there are small witnessing

circuits, for which the witnessed formula is easy to refute, and so it must be the case that it is hard to derive the witnessing circuits. □

For completeness, we also observe that examples of superpolynomial QU-Res lower bounds shown using propositional lower bounds and using circuit lower bounds are already known. In the case of propositional lower bounds, any propositional formulas which require large proofs in Resolution, such as the pigeonhole principle formulas PHP_n, fall into this case, as there are no universal variables to witness. For an example of a circuit lower bound for QU-Res, we give the example of the $\mathrm{QPARITY}_n$ formulas (Definition 5.7), where the lower bound arises as a result of a lower bound on $\mathbf{AC}^0_3$ circuits.

In this chapter, we have given a characterisation of QBF proof complexity lower bounds which extends beyond the powerful Frege +∀red and eFrege +∀red system to all systems with strategy extraction. In Chapter 7, we look in particular at the tree-like Frege +∀red and eFrege +∀red proof systems. In this case, with ∀-reduction restricted to 0/1 substitutions, we are able to give an alternative characterisation of lower bounds arising from this third case.

Chapter 7

Tree-like P+∀red lower bounds via strategy size

In Chapter 6 we observed that in weaker systems than Frege +∀red and eFrege +∀red, it is possible for lower bounds to arise by means other than propositional lower bounds or circuit lower bounds. Here, we present a novel lower bound technique which provides just such lower bounds in the case of tree-like P+∀red systems. Moreover, in the case of tree-like Frege +∀red and eFrege +∀red systems, we show that this technique is sufficient to show any lower bounds which do not also provide a lower bound for dag-like systems via propositional hardness or due to a circuit complexity lower bound.

For these lower bounds, we make use of a round-based strategy extraction algorithm [62]. Rather than using the proof to construct circuits computing a universal winning strategy, this algorithm uses iterative restriction of the proof by the players' assignments to obtain responses for the universal player. We make use of this strategy extraction algorithm to lower bound the size of such proofs by the number of different responses the universal player may need in order to falsify the QBF, a measure we call *strategy size*.

For any given existential assignment, we identify from the round-based strategy extraction algorithm a sequence of lines in the proof defining the universal player's response. We observe that with a careful definition of the restriction of a proof, we can ensure that this sequence of lines is totally ordered under $\prec_\pi$. As a result, for each existential assignment we can find a path from the root of a proof to an axiom which corresponds to the universal response. In the tree-like case, we show that paths corresponding to distinct responses must be distinct, giving us a lower bound on the size of the proof. This lower bound demonstrates that these tree-like systems are incomparable with even weak dag-like systems such as Q-Res.

To completely categorise lower bounds for tree-like Frege +∀red and eFrege +∀red, we make use of a variant of the normal form used in Chapter 6. We first construct small circuits computing an identical winning strategy to the round-based strategy extraction algorithm. The previous normal form is then followed, but with a separate branch of the tree for each distinct response in the range

of the strategy, combining two branches when all universal variables on which the responses differ have been removed.

We present the lower bound for tree-like P+∀red proof systems by strategy size in Section 7.1. In Section 7.2, we show that this lower bound technique suffices to prove all tree-like Frege +∀red lower bounds which are not also Frege +∀red lower bounds.

7.1 A lower bound technique for tree-like proofs

For this chapter, we assume that each block of variables in a QBF contains only a single variable. That is, we assume that any QBFs are of the form

$$\exists x_1 \forall u_1 \exists x_2 \forall u_2 \dots \exists x_n \forall u_n \exists x_{n+1} \cdot \phi(x_1, u_1, \dots, u_n, x_{n+1})$$

for some CNF ϕ. This is an equivalent definition of a PCNF to that given in Chapter 3, but we give an explicit ordering on the order in which the variables are assigned within the blocks. It is clear to see that the ordering of variables within the blocks does not affect the truth of the QBF. For an assignment $\alpha \in \langle \mathcal{X} \rangle$, we denote by $\alpha|_i$ the restriction of α to the variables $x_1, \dots, x_i$, and define $\beta|_i$ similarly for $\beta \in \langle \mathcal{U} \rangle$.

We further use only the version of the ∀-reduction rule which allows substitution by the constants 0 and 1. While this is regularly used interchangeably with substitution by any suitable circuit, since the two are usually equivalent, the results of this chapter apply only to 0/1 ∀-reduction, serving to highlight an important difference between the two approaches to this deduction rule.

We also wish to consider a slightly more careful restriction of proofs. Previously, it has been sufficient to define the restriction of a proof π by an assignment α as $\pi[\alpha] = \{C[\alpha] \mid C \in \pi\}$. However, while $\pi[\alpha]$ is a sound proof for suitable α, it may contain several lines which are not necessary for a refutation of $\Phi[\alpha]$, such as instances of $\top$.

Given a proof π of a QBF Φ and an assignment α, we therefore let L^α be the first line in π which restricts to $\bot$ under α. We initially restrict to the set $\{L[\alpha] \mid L \preceq_\pi L^\alpha\}$ of lines used to derive $L^\alpha[\alpha] = \bot$. We then remove any lines now evaluating to $\top$ and iteratively remove any sinks which are not $L^\alpha[\alpha]$, i.e. lines with no direct descendants in the proof. The restricted proof is therefore a derivation of $\bot$ from $\Phi[\alpha]$ which contains no superfluous lines. While these lines need not be removed from a restricted proof, doing so greatly simplifies the structure of the restricted proof, and the arguments in this chapter.

Round-based strategy extraction In Section 3.3, the concept of strategy extraction was introduced, allowing the construction of circuits computing winning strategies for the universal player from refutations of false QBFs. An alternative approach to strategy extraction was presented in [62], using iterative restrictions of the proof to generate strategies, which we now describe. This approach was initially shown only for Q-Res, but was later extended to LD-Q-Res [53], and the proof extends

naturally to any P+∀red system if we assume a total order on the universal variables; we give this extension in a slightly more general case in Lemma 8.9.

Given a P+∀red refutation π of a false QBF Φ, and an assignment α to the existential variables of Φ, the universal player's response β is constructed round by round, as described in Algorithm 3.

Algorithm 3 The round-based strategy extraction algorithm

function stratex(π,α)
  $\pi_0^\alpha \leftarrow \pi$
  $\beta \leftarrow \varnothing$
  for $1 \leq i \leq n$ **do**
    $\pi_i^\alpha \leftarrow \pi_{i-1}^\alpha[\alpha|_i \cup \beta]$
    $L_i^\alpha \leftarrow \mathrm{root}(\pi_i^\alpha)$
    if L_i^α is derived by a ∀-reduction u_i/b **then**
      $\beta \leftarrow \beta \cup \{u_i/b\}$
    else
      $\beta \leftarrow \beta \cup \{u_i/0\}$
  return β

In brief, at the ith round, the algorithm updates the current proof by restricting by the assignment to the existential variable x_i and by the current universal response. It then takes the root of this restricted proof as the line L_i^α. If L_i^α is derived by a ∀-reduction on u_i, which is now the leftmost unassigned variable, then the universal player plays according to this ∀-reduction, otherwise the universal player can play arbitrarily – for concreteness, we assume this is always to play $u_i/0$.

We define π_i^α to be the proof constructed at the beginning of the ith round. Since at any stage, the proof is only restricted by an assignment to existential variables or to a universal variable that is leftmost, each proof π_i^α is a sound refutation of $\Phi[\alpha|_i \cup \beta|_{i-1}]$. It is therefore clear that after n rounds, π_n^α is a sound refutation of $\Phi[\alpha|_n \cup \beta]$, a purely existential formula, so β is indeed a winning response to α, and the response at round i is computed using only the assignment $\alpha|_i$. The strategy $S_\pi : \langle\mathcal{X}\rangle \to \langle\mathcal{U}\rangle$ defined by $S_\pi(\alpha) = \mathsf{stratex}(\pi, \alpha)$ is therefore a winning universal strategy for Φ for any P+∀red refutation π of Φ.

Since the strategy in Algorithm 3 is determined by the deduction rule used to derive the lines L_i^α, we are primarily interested in which lines of π remain in π_i^α, rather than in the precise restriction to these lines. As a result, and to simplify notation, for a line $L \in \pi$, we abbreviate the statement $L[\alpha|_i \cup S_\pi(\alpha)|_{i-1}] \in \pi_i^\alpha$ to $L \in \pi_i^\alpha$, since the relevant restriction is evident in the proof π_i^α.

We now observe that the round-based strategy extraction algorithm not only defines a universal response to each assignment $\alpha \in \langle\mathcal{X}\rangle$, but it also constructs a sequence of lines L_i^α used to determine the universal response on u_i. Moreover, since the response for u_i can be determined by looking at the deduction rule deriving L_i^α, assignments eliciting different universal responses from S_π must arise from different sequences of lines.

Lemma 7.1. *Let π be a* P+∀red *refutation of a QBF Φ. Given two assignments $\alpha, \gamma \in \langle \mathcal{X} \rangle$ such that $S_\pi(\alpha) \neq S_\pi(\gamma)$, there is some $1 \leq k \leq n$ such that $L_k^\alpha \neq L_k^\gamma$.*

Proof. Let $\beta_\alpha, \beta_\gamma \in \langle \mathcal{U} \rangle$ be the universal responses to α and γ respectively under S_π. Since $\beta_\alpha \neq \beta_\gamma$, there is some k such that, without loss of generality, $\beta_\alpha(u_k) = 1$ and $\beta_\gamma(u_k) = 0$. By the construction of S_π, it must be the case that L_k^α is derived by a $\forall$-reduction substituting $u_k/1$, whereas L_k^γ is derived by a $\forall$-reduction substituting $u_k/0$, or by a propositional deduction rule. In either case, it is clear that $L_k^\alpha \neq L_k^\gamma$ since each line in π is only derived by a single deduction rule. □

It is worth emphasising that as we have defined round-based strategy extraction, the proof π_i^α is a restriction of the proof π_{i-1}^α used at the previous round, rather than simply restricting π by $\alpha|_i \cup \beta|_{i-1}$. Both approaches construct winning universal strategies, however since $\pi[\alpha|_i \cup \beta|_{i-1}]$ and π_i^α are not necessarily identical, the strategies constructed may also be different.

Using restrictions of π_{i-1}^α rather than π at the ith round ensures the following useful property of the lines L_i^α: for any assignment $\alpha \in \langle \mathcal{X} \rangle$ and any $j < i$, either $L_j^\alpha = L_i^\alpha$, or $L_j^\alpha \prec L_i^\alpha$. We can therefore extend the sequence of lines L_i^α to a path through π corresponding to the run of the strategy extraction algorithm on π and α.

Definition 7.2. *Define $p_\alpha \subseteq \pi$ to be a path through π, i.e. a maximal totally ordered subset of π under $\prec_\pi$, such that $L_i^\alpha \in p_\alpha$ for each $1 \leq i \leq n$, and for any $L \in p_\alpha$, if $L \prec_\pi L_i^\alpha$ then $L \in \pi_i^\alpha$.*

This definition does not necessarily uniquely define the path p_α; we could ensure the uniqueness of p_α by requiring it to be the first such path lexicographically under the order in which lines appear in π. Since the only properties of p_α we require are those given in Definition 7.2, we do not specify which of the many suitable paths we choose. In particular, the following result that different universal responses correspond to different paths, is independent of the precise choice of p_α.

Lemma 7.3. *Let π be a* P+∀red *refutation of a QBF Φ. For any two assignments $\alpha, \gamma \in \langle \mathcal{X} \rangle$, if $S_\pi(\alpha) \neq S_\pi(\gamma)$ then $p_\alpha \neq p_\gamma$.*

Proof. Extend the sequences L_i^α and L_i^γ by letting $L_0^\alpha = L_0^\gamma = \bot$, the root of π. Pick the least k such that $L_k^\alpha \neq L_k^\gamma$, and so $L_{k-1}^\alpha = L_{k-1}^\gamma = L_{k-1}$. Such a k must exist by Lemma 7.1.

If L_k^α and L_k^γ are incomparable in $\prec_\pi$, then they cannot appear in the same path, since paths are totally ordered. As $L_k^\alpha \in p_\alpha$ and $L_k^\gamma \in p_\gamma$, we see that $p_\alpha \neq p_\gamma$. Therefore assume without loss of generality that $L_k^\alpha \prec_\pi L_k^\gamma$. Recall that for any line $L \in p_\gamma$ such that $L \prec_\pi L_k^\gamma$, we require $L \in \pi_k^\gamma$. To show $p_\alpha \neq p_\gamma$, it therefore suffices to show that $L_k^\alpha \notin \pi_k^\gamma$, and hence $L_k^\alpha \notin p_\gamma$.

The lines of π_k^γ are a restrictions of a subset of the lines of π_{k-1}^γ, so if $L_k^\alpha \notin \pi_{k-1}^\gamma$, we are done. Now assume that $L_k^\alpha \in \pi_{k-1}^\gamma$. By the definition of L_k^α, $\mathrm{lv}(L_k^\alpha) \leq 2k - 1$ and so $\gamma|_k \cup \beta|_{k-1}$ is a total assignment to the variables of L_k^α. If $L_k^\alpha[\gamma|_k \cup \beta|_{k-1}] = \bot$, this would contradict the choice of L_k^γ as the first line in π_{k-1}^γ which restricts to $\bot$ under this assignment. It must therefore be the case that $L_k^\alpha[\gamma|_k \cup \beta|_{k-1}] = \top$, and so by the definition of restriction, $L_k^\alpha \notin \pi_k^\gamma = \pi_{k-1}^\gamma[\gamma|_k \cup \beta|_{k-1}]$, as tautologies are removed from the restricted proof. □

We have seen that existential assignments resulting in different responses from the universal winning strategy correspond to different sequences of lines L_i^α, and consequently different paths through the proof. Given these results, it is natural to define a measure counting the number of distinct responses required by a winning strategy. We can then use Lemma 7.3 to shed some light on the structure of P+$\forall$red proofs of QBFs which require a large number of responses.

Definition 7.4. *Given a QBF Φ, the* strategy size $\rho(\Phi)$ *is the minimal size of the range of a winning strategy for Φ:*

$$\rho(\Phi) := \min\{|\,\mathsf{rng}(S)|\mid S \text{ is a winning strategy for } \Phi\}$$

As an immediate corollary of Lemma 7.3, we get a lower bound on the number of paths through a P+$\forall$red proof, since for any two responses, the corresponding paths are pairwise distinct.

Corollary 7.5. *Given a QBF Φ and a* P+$\forall$red *proof π of Φ, the round-based strategy extraction algorithm constructs at least $\rho(\Phi)$ distinct paths through π.*

This lower bound on the number of paths through a refutation demonstrates the power of dag-like proofs over tree-like proofs. In a dag-like proof, where lines can be reused in several subsequent deduction steps, this allows multiple paths from a given line to the root of the proof. However, in the case of tree-like P+$\forall$red proofs, lines cannot be reused in this way, resulting in a unique path from any given line to the root of the proof. This lower bound on paths therefore immediately gives a lower bound on tree-like P+$\forall$red proofs based only on the simple measure of strategy size, and independent of the underlying propositional proof system P.

Theorem 7.6. *For any QBF Φ, if π is a tree-like* P+$\forall$red *refutation of Φ, then $|\pi| \geq \rho(\Phi)$.*

Proof. Since π is a tree-like proof, each axiom introduced in π defines a unique path between the axiom to the root of the proof. By Corollary 7.5, there are at least $\rho(\Phi)$ distinct paths in π constructed by the strategy extraction algorithm. Since each path identifies a distinct axiom, there are at least $\rho(\Phi)$ axioms introduced in π. □

The question of lower bounding tree-like P+$\forall$red proofs is therefore reduced to finding a lower bound on $\rho(\Phi_n)$ for some family of QBFs Φ_n. Several examples of such QBFs have previously been defined, such as the formulas KBKF_n. The QBFs we choose to exemplify a lower bound on $\rho(\Phi_n)$ were defined by Janota and Marques-Silva in [70].

Definition 7.7 (Janota, Marques-Silva [70]). *The QBFs DEQ_n are defined as*

$$\begin{aligned}\mathrm{DEQ}_n :=& \exists x_1 \forall u_1 \exists t_1 t_2 \ldots \exists x_n \forall u_n \exists t_{2n-1} t_{2n} \cdot \\ & \bigwedge_{i=1}^{n} [(\neg x_i \vee t_{2i-1}) \wedge (\neg u_i \vee t_{2i-1}) \wedge (x_i \vee t_{2i}) \wedge (u_i \vee t_{2i})] \wedge \bigvee_{j=1}^{2n} \neg t_j\end{aligned}$$

The reason for choosing these QBFs is that it has been observed that there are short refutations of DEQ_n even in proof systems as weak as Q-Res.

Theorem 7.8 (Janota, Marques-Silva [70]). *The QBFs* DEQ_n *have polynomial-size refutations in Q-Res and QU-Res.*

Proof. Let T_i be the clause $\bigvee_{j=1}^{2i} \neg t_j$. Beginning with T_n, which is an axiom, we derive T_{i-1} from T_i until finally deriving $T_0 = \bot$.

Given T_i, we can resolve with the axioms $\neg x_i \vee t_{2i-1}$ and $u_i \vee t_{2i}$ to obtain $T_{i-1} \vee \neg x_i \vee u_i$. Similarly, resolving T_i with $\neg u_i \vee t_{2i-1}$ and $x_i \vee t_{2i}$ gives the clause $T_{i-1} \vee x_i \vee \neg u_i$. In both cases, we can $\forall$-reduce u_i. Resolving the resulting clauses on the pivot x_i deduces the clause T_{i-1}. □

Despite these short proofs, there is a unique universal winning strategy for DEQ_n, and this strategy requires 2^n distinct responses.

Theorem 7.9. *If* π *is a tree-like* Frege +∀red *or* eFrege +∀red *refutation of* DEQ_n*, then* $|\pi| \geq 2^n$.

Proof. If $u_i = x_i$ for any i, then the existential player can set either t_{2i-1} or t_{2i} to 0 while still satisying all clauses containing x_i or u_i. Playing all other t_j positively satisfies all clauses. The only winning universal strategy is therefore to play $u_i = 1 - x_i$, as this forces the existential player to set both t_{2i-1} and t_{2i} positively, ultimately falsifying the large clause in the final round. As this is the unique winning strategy, we see that $\rho(\Phi_n) = 2^n$, and the lower bound for $|\pi|$ follows by Theorem 7.6. □

This lower bound for tree-like Frege +∀red and eFrege +∀red, and the consequent separation between tree-like eFrege +∀red and dag-like Q-Res, is in stark contrast to previously observed equivalences between Frege and Frege +∀red systems. In the propositional case, it is known that tree-like and dag-like Frege systems are equivalent [75]. In [30], it was shown that tree-like and dag-like Frege +∀red are equivalent if the $\forall$-reduction rule is allowed to substitute a variable u by any suitable Boolean formula in variables left of u, rather than only by constants 0/1 as we have defined here. Moreover, [30] also observed that in *dag-like* Frege +∀red systems, allowing $\forall$-reduction by 0/1 is equivalent to allowing $\forall$-reduction by arbitrary formulas. The same equivalences apply in the case of eFrege +∀red, allowing substitutions by Boolean circuits.

However, both of these equivalences rely on the fact that the alternative restriction is not present. Consequently, restricting proofs to be both tree-like and only use $\forall$-reduction by the constants 0/1 results in a substantially weaker system, as shown by the lower bound of Theorem 7.9. Since Frege +∀red p-simulates QU-Res, we conclude that tree-like Frege +∀red is exponentially weaker than dag-like Frege +∀red (both with 0/1 $\forall$-reduction) and even incomparable with Q-Res.

Theorem 7.10. *Tree-like* Frege +∀red *(with 0/1* $\forall$*-reduction) and Q-Res are incomparable.*

Proof. To see that Q-Res does not simulate tree-like Frege +∀red, the pigeonhole principle formulas PHP_n have short proofs in tree-like Frege, and hence short proofs in tree-like Frege +∀red, but require exponential size proofs in Resolution and hence Q-Res.

For the opposite separation, DEQ_n have short Q-Res proofs but require exponential-size proofs in tree-like Frege +∀red. □

This exponential lower bound for tree-like Frege +∀red and eFrege +∀red is also significant as no such lower bound is known for the corresponding propositional proof systems of tree-like Frege and eFrege, which are known to be equivalent to their dag-like versions [75]. Such QBF proof complexity lower bounds in the absence of propositional lower bounds are relatively rare. Other examples, such as the lower bound for $\mathbf{AC}^0[p]$-Frege +∀red in [19], have been shown by lifting circuit complexity lower bounds using the strategy extraction of Theorem 3.5. In the case of Theorem 7.9, the unique winning strategy can be computed by small circuits, suggesting that this represents a new approach to producing QBF proof complexity lower bounds.

7.2 Characterising tree-like Frege +∀red lower bounds

As described in Section 6.1, [30] established a characterisation of superpolynomial lower bounds for Frege +∀red and eFrege +∀red. By constructing a normal form for proofs in these proof systems, it was shown that such lower bounds on (dag-like) Frege +∀red or eFrege +∀red proofs are a result of lower bounds on the corresponding propositional proof systems, or a circuit complexity lower bound.

It is evident from the lower bound for tree-like Frege +∀red and eFrege +∀red given in Theorem 7.9 that this characterisation does not hold for the tree-like versions of these systems. However, by varying the normal form from [30] slightly, we can extend this characterisation of lower bounds to the tree-like systems, with any lower bounds not arising from propositional lower bounds or circuit complexity lower bounds being a result of a lower bound on strategy size. Similarly to the results of Chapter 6, the results of this section will hold for $\mathcal{C}$-Frege +∀red for suitable circuit classes such as $\mathbf{AC}^0$ and $\mathbf{TC}^0$, but for clarity we use only Frege +∀red and eFrege +∀red here.

Recall that for a proof π and a line $L \in \pi$, we write $L \in \pi_i^\alpha$ when the relevant restriction of L is in π_i^α. In the case of tree-like proofs, the lines of π_i^α depend only on the line L_i^α, and are otherwise independent of the assignment α.

Lemma 7.11. *Let π be a tree-like refutation of a QBF Φ, and let $\alpha, \gamma \in \langle \mathcal{X} \rangle$ be two distinct existential assignments. For any $1 \leq i \leq n$, if $L_i^\alpha = L_i^\gamma = L_i$ then:*

(i) $L_j^\alpha = L_j^\gamma$ for each $j \leq i$.

(ii) for any line $L \in \pi$, $L \in \pi_i^\alpha$ if and only if $L \in \pi_i^\gamma$, i.e. π_i^α and π_i^γ contain the relevant restrictions of the same lines of π.

Proof. (i) We prove the first claim by showing that if $L_j^\alpha = L_j^\gamma$ then $L_{j-1}^\alpha = L_{j-1}^\gamma$. The result then follows for all $j \leq i$ by induction. In the cases $j = 0$ and $j = 1$, this is trivially true, since $\pi_0^\alpha = \pi_0^\gamma = \pi$ and $L_0^\alpha = L_0^\gamma = \bot$.

There is a unique path in π from $L_j^\alpha = L_j^\gamma$ to the root of the proof. Furthermore, since $L_j^\alpha \prec_\pi L_{j-1}^\alpha$ and $L_j^\gamma \prec_\pi L_{j-1}^\gamma$, both L_{j-1}^α and L_{j-1}^γ lie on this path. Towards a contradiction, assume $L_{j-1}^\alpha \neq L_{j-1}^\gamma$, and so without loss of generality assume $L_{j-1}^\alpha \prec_\pi L_{j-1}^\gamma$.

Since $L_j^\gamma = L_j^\alpha \preceq_\pi L_{j-1}^\alpha \prec_\pi L_{j-1}^\gamma$, the restricted proof π_{j-1}^γ must contain L_{j-1}^α, and hence π_{j-2}^γ contains L_{j-1}^α. The assignment $\gamma|_{j-1} \cup S_\pi(\gamma)|_{j-2}$ which falsifies L_{j-1}^γ is also a total assignment to L_{j-1}^α. If L_{j-1}^α evaluates to $\top$ under this assignment, then L_{j-1}^α and all its predecessors would not appear in π_{j-1}^γ, contradicting the choice of $L_j^\gamma = L_j^\alpha \preceq L_{j-1}^\alpha$. Instead, L_{j-1}^α evaluates to $\bot$, contradicting the choice of L_{j-1}^γ as the first instance of $\bot$ in the restriction of π_{j-2}^γ. We therefore conclude that $L_{j-1}^\alpha = L_{j-1}^\gamma$.

(ii) We proceed by induction on i. In the case $i = 0$, $\pi_0^\alpha = \pi_0^\gamma = \pi$, so the property holds.

If $L_i^\alpha = L_i^\gamma$, by (i) we have that $L_{i-1}^\alpha = L_{i-1}^\gamma$, so the restricted proofs π_{i-1}^α and π_{i-1}^γ contain restrictions of the same lines of π. Suppose now that there is some line L such that, without loss of generality, $L \in \pi_i^\alpha$ but $L \notin \pi_i^\gamma$. Since $L \in \pi_{i-1}^\alpha$, $L \in \pi_{i-1}^\gamma$ and so there must be some line L' such that $L \preceq_\pi L' \prec_\pi L_{i-1}^\gamma$ satisfying $L'[\gamma|_i \cup S_\pi(\gamma)|_{i-1}] = \top$, since this is the only way that L could fail to appear in π_i^γ. Moreover, we have $L \preceq_\pi L' \prec_\pi L_i^\gamma = L_i^\alpha$, else L_i^γ would have been removed from π_i^γ.

Since $\gamma|_i \cup S_\pi(\gamma)|_{i-1}$ is a total assignment to the variables of L', we have that $\alpha|_i \cup S_\pi(\alpha)|_{i-1}$ is also a total assignment to the variables of L'. However, since $L' \prec_\pi L_i^\alpha$, it must be the case that $L'[\alpha|_i \cup S_\pi(\alpha)|_{i-1}] = \top$, else this would contradict the choice of L_i^α. However, if $L'[\alpha|_i \cup S_\pi(\alpha)|_{i-1}] = \top$, then we cannot have $L \in \pi_i^\alpha$, since any path from L_i^α to L in π would pass through L'. This contradicts the choice of L, so no such L exists. It follows that for any line $L \in \pi$, $L \in \pi_i^\alpha$ if and only if $L \in \pi_i^\gamma$. □

The properties shown in Lemma 7.11 allow us to compute which lines of π are in the proofs π_j^α for any $j \leq i$ when given no information about an assignment α other than L_i^α. As a result, we denote by π_i^L the set of lines of π in the restricted proof whenever L is chosen at the ith round. Being independent of any assignment α, we give such a construction of π_i^L. It is prudent to observe that, formally, π_i^L must be considered as a subset of lines of π, since even if $L_i^\alpha = L_i^\gamma = L$, the lines in π_i^L will be restricted by different assignments in π_i^α and π_i^γ

Lemma 7.12. *Let π be a tree-like* P+∀red *refutation of a QBF Φ. For any $L \in \pi$ and $0 \leq i \leq n$,*

$$\pi_i^L = \{L' \preceq_\pi L \mid \mathrm{var}(M) \not\subseteq \{x_1, \ldots, x_i, u_1, \ldots, u_{i-1}\} \textit{ for all } L' \preceq_\pi M \prec_\pi L\}$$

Proof. Let α be some assignment such that $L_i^\alpha = L$. If $i = 0$, $\pi_0^\alpha = \pi$. For every line $L' \in \pi$, it is clear that $L' \preceq_\pi L_0^\alpha$ since L_0^α is the root of the proof. Moreover, every other line in $M \in \pi$ with $M \neq L_0^\alpha$ contains a variable, so $\mathrm{var}(M) \not\subseteq \varnothing$, and $\pi_i^L = \pi$.

We proceed by induction on i. Since π_i^α is a restriction of π_{i-1}^α, it is clear that $\pi_i^L \subseteq \pi_{i-1}^L$. Moreover, since all lines in π_i^α are predecessors of $L_i^\alpha = L$, we see that

$$\pi_i^L \subseteq \{L' \preceq_\pi L \mid \mathrm{var}(M) \not\subseteq \{x_1, \ldots, x_{i-1}, u_1, \ldots, u_{i-2}\} \text{ for all } L' \preceq_\pi M \prec_\pi L\}.$$

Aside from restricting only to the predecessors of L_i^α, the only other lines removed from π_{i-1}^α to give π_i^α are those lines which restrict to $\top$ under the assignment $\alpha|_i \cup S_\pi(\alpha)|_{i-1}$ and their

predecessors. Any such line M must satisfy $\mathrm{var}(M) \subseteq \{x_1, \ldots, x_i, u_1, \ldots, u_{i-1}\}$, and all lines $M \prec_\pi L$ satisfying this property must restrict to $\top$ under $\alpha|_i \cup S_\pi(\alpha)|_{i-1}$, else they would restrict to $\bot$, contradicting the choice of $L_i^\alpha = L$. The definition of π_i^L follows. □

The normal form for the dag-like Frege +∀red and eFrege +∀red systems used circuits computing universal winning strategies that had been constructed in polynomial time from a Frege +∀red or eFrege +∀red refutation. However, the algorithm used to construct these circuits is that given in [19], which need not construct circuits computing the same winning strategy as the round-based strategy extraction algorithm. To give a normal form for tree-like proofs, we therefore begin by extending this strategy extraction result to show that in the case of a tree-like Frege +∀red or eFrege +∀red proof, we can construct circuits computing the winning strategy defined by the round-based strategy extraction algorithm in Algorithm 3.

Lemma 7.13. *Let π be a tree-like* Frege +∀red *(resp. tree-like* eFrege +∀red*) refutation. There are formulas (resp. circuits) C_i with inputs $\{x_1, u_1, \ldots, x_i\}$ of size $O(|\pi|^2)$ computing the strategy for u_i extracted from π by the strategy extraction algorithm in Algorithm 3.*

Proof. As for the construction of strategies in Theorem 3.5, we aim to construct small decision lists for the variables u_i with circuits in $\mathbf{NC}^1$ or $\mathbf{P/poly}$ respectively. From such a decision list, it is straightforward to construct a circuit in $\mathbf{NC}^1$ or $\mathbf{P/poly}$ respectively computing the same function as the decision list, with size polynomial in that of the decision list.

For a line $L \in \pi$ such that $\mathrm{var}(L) \subseteq \{x_1, u_1, \ldots, x_i\}$, we inductively construct conjunctions C_i^L in which all conjuncts are lines of π with all variables left of u_i, or negations of such lines. We further ensure that C_i^L is a sufficient and necessary condition that the strategy extraction algorithm selects L in the ith round. We can then use these conjunctions in our decision list.

Define $C_0^\bot = \top$, since L_0^α is always the root of the proof for any assignment α. For $i > 0$, there is a unique line M which must be selected at round $i-1$ in order to select L in the ith round (Lemma 7.11 (i)). Specifically, the line M is the first of L or its descendants to have all variables left of u_{i-1}. To choose L at the ith round, the assignment must therefore satisfy C_{i-1}^M.

Having chosen M at round $i-1$, the proof is now restricted to the lines π_{i-1}^M. To choose L at round i, the algorithm must verify that L restricts to $\bot$ under the current assignment, and that no lines appearing before L in π_{i-1}^M restrict to $\bot$. Since π_{i-1}^M has already removed any lines whose variables are all left of u_{i-1}, and hence these were checked in C_{i-1}^M, it remains only to check those lines of π_{i-1}^M which appear before L and whose rightmost variable is either u_{i-1} or x_i and verify they evaluate to true. The set of lines the algorithm considers at this round is therefore $\mathcal{L}_i^L = \{L' \in \pi_{i-1}^M \mid L' <_\pi L, 2i-2 \leq \mathrm{lv}(L) \leq 2i-1\}$, which can be computed efficiently. This gives us the conjunction

$$C_i^L = C_{i-1}^M \wedge \bigwedge_{L' \in \mathcal{L}_i^L} L' \wedge \neg L. \tag{3}$$

For each line $L \in \pi$ with variables left of u_i, we can add to the decision list for u_i the line

$$\text{if } C_i^L \text{ then } b_L, \text{ else ...}$$

where b_L is the value assigned to u_i by the round-based strategy extraction algorithm if L is the line at the root of π_i^α. By our construction of C_i^L, it is clear that this decision list computes the same strategy for u_i as that computed in Algorithm 3.

To verify that the size of the decision list is $O(|\pi|^2)$, observe first that the decision list for u_i contains at most one line for each line in π, so the number of lines in the decision list is at most $|\pi|$. The conjunction C_{i-1}^M is a conjunction only of lines with level at most $2i-3$, and to construct C_i^L, only lines with level $2i-2$ or $2i-1$ are added. Each line therefore appears at most once in the conjunction C_i^L, and hence $|C_i^L| = O(|\pi|)$. The size of the resulting decision list, and therefore the size of the circuit constructed from it, is $O(|\pi|^2)$. □

Having demonstrated the existence of small circuits computing our desired winning strategy, we can now use these circuits to define the normal form for tree-like Frege +∀red and eFrege +∀red proofs which we use to give our characterisation. This normal form is based on that used in [30] and in Chapter 6 to characterise proofs in dag-like Frege +∀red and eFrege +∀red, and other weaker systems.

We begin in the same way, using the fact that the circuits C_i compute a winning strategy for the universal variables to derive the line $\bigvee_{i=1}^n (u_i \not\leftrightarrow C_i)$. However, rather than deriving it only once, we derive a copy of this line for each response β in the range of the strategy computed by the circuits C_i. The normal form proceeds similarly to before, reducing each u_j according to the response associated with that line. We can then combine lines whose responses first differ on u_j in order to derive a copy of $\bigvee_{i=1}^{j-1}(u_i \not\leftrightarrow C_i)$ for each response to the variables $u_1, \ldots, u_{j-1}$, finally deriving the empty disjunction after reducing u_1.

We formally define this normal form in Definition 7.14, and Theorem 7.15 proves that we can efficiently transform any tree-like Frege +∀red or eFrege +∀red proof into a proof in this normal form.

Definition 7.14. *Let $\Phi = \Pi \cdot \phi$ be a QBF, and let the circuits C_i compute a winning universal strategy for the universal variables u_i. Define $S : \langle \mathcal{X} \rangle \to \langle \mathcal{U} \rangle$ to be the strategy computed by the C_i, with $\mathsf{rng}(S) = \{\beta_1, \ldots, \beta_s\}$. Since the C_i form a winning strategy for Φ, it is clear that $\bigwedge_{i=1}^n (u_i \leftrightarrow C_i) \models \neg\phi$ and so $\phi \models \bigvee_{i=1}^n (u_i \not\leftrightarrow C_i)$.*

A proof in the normal form begins by deriving (propositionally) $\bigvee_{i=1}^n (u_i \not\leftrightarrow C_i) \vee \neg\beta_j$ for each $1 \leq j \leq s$, where $\neg\beta_j$ is the disjunction of those literals falsified by β_j. Each such line is then $\forall$-reduced by the substitution $u_n/\beta_j(u_n)$. The lines $\bigvee_{i=1}^{n-1}(u_i \not\leftrightarrow C_i) \vee \neg\beta_j|_{n-1}$ can then be constructed either by propositional inference from a single line if β_j is the unique extension of $\beta_j|_{n-1}$ in $\mathsf{rng}(S)$, or by combining lines corresponding to the two extensions of $\beta_j|_{n-1}$ otherwise. Repeating this process for each of the universal variables from u_n to u_1 results in a derivation of $\bot$.

Given a tree-like proof π, Lemma 7.13 provides suitable circuits of size $|\pi|^{O(1)}$ which compute a strategy S with $|\operatorname{rng}(S)| \leq |\pi|$. Using these circuits C_i in the normal form, we can now transform any tree-like proof into this normal form with only a polynomial increase in size.

Theorem 7.15. *Let $\Phi = \Pi \cdot \phi$ be a QBF, and π be a tree-like* Frege +∀red *(respectively tree-like* eFrege +∀red*) refutation of Φ. There is a tree-like* Frege +∀red *(respectively tree-like* eFrege +∀red*) refutation of Φ in the form of Definition 7.14 with size $|\pi|^{O(1)}$.*

Proof. For the circuits C_i, we use the circuits constructed from π in Lemma 7.13. By their construction, these circuits have size $|\pi|^{O(1)}$, and Lemma 7.3 ensures that the corresponding strategy S computed by the C_i satisfies $|\operatorname{rng}(S)| \leq |\pi|$. Since dag-like and tree-like propositional Frege systems are equivalent [75], it remains only to show that each of the propositional inferences described in Definition 7.14 can be done using a dag-like Frege derivation of size $|\pi|^{O(1)}$.

To first derive $\bigvee_{i=1}^{n}(u_i \not\leftrightarrow C_i) \vee \neg\beta$ for some $\beta \in \operatorname{rng}(S)$, we construct from π a proof that $\phi \wedge \beta \wedge \bigwedge_{i=1}^{n}(u_i \leftrightarrow C_i) \to \bot$, which can be straightforwardly transformed into the derivation we require. To do so, we derive for each line $L \in \pi$ the line $\neg C^L$, where $C^L = \neg C_j^L$, and j is the least such j for which C_j^L is defined, i.e. j is minimal such that $\operatorname{var}(L) \subseteq \{x_1, u_1, \ldots, x_j\}$. Since the $C_0^{\bot} = \top$, we observe that $\neg C_0^{\bot} = \bot$, so this is indeed a derivation of $\bot$.

We first note that if we have derived $\neg C^M$ for each $M <_\pi L$, then it is sufficient to derive (a subclause of) $\neg C_i^L$ for any i such that C_i^L is defined. The disjunction $\neg C_i^L = \neg C^L \vee \bigvee_k \neg M_k$ where the lines $M_k \prec_\pi L$ are those lines checked by the algorithm between choosing L at the first round j in which it can be chosen, as must be the case by Lemma 7.11, and choosing L at round i. Each line $M_k \in \pi_j^L$, and so by the definition of π_j^L given in Lemma 7.12, in order to choose M_k at any given round, L must previously have been chosen at the jth round. In particular, this means that each C^{M_k} is of the form $C^L \wedge M_1 \wedge \cdots \wedge M_{k-1} \wedge \neg M_k$, and so each instance of $\neg M_k$ in C_i^L can be 'resolved' away in turn using $\neg C^{M_k}$ to obtain $\neg C^L$. Clearly $k < |\pi|$ and hence this requires size $|\pi|^{O(1)}$.

We can now derive each $\neg C^L$ by induction on the order lines appear in the proof.

- Suppose L is derived in π as an axiom. In this case, L appears as a clause in ϕ and can be introduced as an axiom. The definition of C_j^L in (3) ensures that L appears as a disjunct in $\neg C_j^L$, so $\neg C^L$ can be derived by weakening from L.
- Suppose L is derived from L' by a ∀-reduction on u_i which agrees with β, i.e. the ∀-reduction $u_i/\beta(u_i)$. Choosing either L or L' at round $i+1$ requires choosing L at round i. As a result, if C_{i+1}^L is of the form $C \wedge L' \wedge \neg L$ for some conjunction of lines C, then $C_{i+1}^{L'} = C^{L'} = C \wedge \neg L'$. Using β, there is a short derivation of $L = L'[u_i/\beta(u_i)]$ from L', and so from $\neg C^{L'} = \neg C \vee L'$ we can derive $\neg C \vee L$, which is a subclause of $\neg C_{i+1}^L$, in linear time.
- Suppose L is derived from L' by a ∀-reduction on u_i which does not agree with β. We can then derive a contradiction from $\beta \wedge (u_i \leftrightarrow C_i)$, C_i^L and the previously derived lines $\neg C^M$ for $M <_\pi L$. The circuit C_i is constructed from a decision list, so if each C^M is false but C_i^L is true, it requires $O(|C_i|)$ lines to evaluate the decision list line by line and conclude that

$C_i \not\leftrightarrow \beta(u_i)$. From this, a contradiction can easily be derived using $\beta \wedge (u_i \leftrightarrow C_i)$. This can be transformed into a short derivation of $\neg C_i^L$ from $\beta \wedge (u_i \leftrightarrow C_i)$ and the lines $\neg C^M$.

– Suppose L is derived by a propositional rule from L_1 and L_2, with $\mathrm{var}(L_1) \subseteq \{x_1, u_1, \ldots, x_l\}$ and the rightmost variable of L_2 being u_{l-1} or x_l, so $C^{L_2} = C_l^{L_2}$. Without loss of generality, we assume that the ordering of lines in π has been chosen such that $L_1 <_\pi L_2$ and that L is the next line derived after L_2 – the lines of π can always be ordered in this way in a tree-like proof. Choose the conjunction C such that C^{L_2} is of the form $C \wedge \neg L_2$.

  Until choosing L_1, the paths chosen for L_1 and L_2 are identical, so apart from $\neg L_1$, all conjuncts in C^{L_1} appear in C^{L_2}, that is, $C^{L_1} = D \wedge \neg L_1$ where all conjuncts in D appear in C. It is clear that $\mathrm{var}(L) \subseteq \{x_1, u_1, \ldots, x_l\}$ and so by the choice of ordering on lines of π, $C_l^L = C \wedge L_2 \wedge \neg L$. We can therefore derive from $\neg C^{L_1} = \neg D \vee L_1$ and $\neg C^{L_2} = \neg C \vee L_2$ the line $\neg C \vee L$, a subclause of $\neg C_l^L$ by a single propositional step.

Having now derived the line $\bigvee_{i=1}^n (u_i \not\leftrightarrow C_i) \vee \neg\beta$ for each response $\beta \in \mathsf{rng}(S)$, we now consider the deduction of $\bot$ from these axioms. The lines $(u_i \leftrightarrow \beta(u_i)) \wedge (u_i \leftrightarrow C_i)$ is equivalent to $C_i \leftrightarrow \beta(u_i)$. If $\beta|_{j-1}$ has two possible extensions on u_j in $\mathsf{rng}(S)$, a derivation of $\bigvee_{i=1}^{j-1}(u_i \not\leftrightarrow C_i) \vee \neg\beta|_{j-1}$ from the corresponding lines for the two extensions requires only proving that $C_j \wedge \neg C_j \models \bot$, for which there is a Frege proof of size $O(|C_j|)$.

In the case where there is a unique extension of $\beta|_{j-1}$ to $\beta|_j$ in $\mathsf{rng}(S)$, it suffices to derive $(C_j \leftrightarrow \beta(u_j))$ from $\bigwedge_{i=1}^{j-1}(C_i \leftrightarrow \beta(u_i))$. Each conjunction C_i^L defines not only a response on u_i, but a response on the variables $u_1, \ldots, u_i$, as C_i^L specifies which lines were picked at each round. We therefore construct for each i in turn the disjunction of those C_i^L which correspond to the response $\beta|_i$. This can be achieved in size $|C_i|^{O(1)}$ at each stage, deriving from each C_{i-1}^M the disjunction of those C_i^L for which choosing L at the ith round agrees with β and requires choosing M at round $i-1$.

For each C_{j-1}^L in the final disjunction, there is a $|C_j|^{O(1)}$-size proof that $C_{j-1}^L \models (C_j \leftrightarrow \beta(u_j))$, by comparing C_{j-1}^L with each line in the decision list for u_j and showing that each line in the decision list which would return $\neg\beta(u_j)$ is falsified by C_{j-1}^L.

For each $1 \leq j \leq n$, we can therefore derive the line $\bigvee_{i=1}^j (u_i \not\leftrightarrow C_i) \vee \beta|_j$ for all $\beta \in \mathsf{rng}(S)$, concluding with the empty conjunction. □

The question of showing superpolynomial lower bounds on the size of tree-like Frege +∀red proofs is therefore equivalent to showing such lower bounds on proofs in the normal form of Definition 7.14. We use this to provide a characterisation of such lower bounds similar to that shown for dag-like Frege +∀red and eFrege +∀red in [30] (Theorem 6.2).

Theorem 7.16. *Each of the following is sufficient to give a superpolynomial lower bound on tree-like* Frege +∀red *(resp. tree-like* eFrege +∀red*) proofs:*

1. *a propositional lower bound on* Frege *(resp.* eFrege*);*
2. *a lower bound on strategy size;*

3. *a lower bound on* $\mathbf{NC}^1$ *(resp.* $\mathbf{P/poly}$*) circuits computing* S *for any winning strategy* S *with polynomial-size range.*

Moreover, any superpolynomial lower bound on tree-like Frege +∀red *(resp. tree-like* eFrege +∀red*) is due to one of the above lower bounds.*

Proof. We focus on the case for Frege +∀red; the eFrege +∀red case is analogous. First, we observe that each of items 1 to 3 is sufficient to give a superpolynomial lower bound.

For item 1, a lower bound for (tree-like) Frege proofs of propositional formulas ϕ_n implies a tree-like Frege +∀red lower bound for the existentially quantified version of ϕ_n.

For item 2, if Φ_n is a sequence of QBFs for which there is a superpolynomial lower bound on $\rho(\Phi_n)$, then Theorem 7.6 gives the same lower bound on the size of a tree-like Frege +∀red proof of Φ_n.

For item 3, let Φ_n be a sequence of QBFs such that $\rho(\Phi_n)$ is small, but there are no polynomial-size circuits in $\mathbf{NC}^1$ computing a universal winning strategy with small range. By Lemma 7.13, we can extract from any tree-like Frege +∀red proof π circuits of size $|\pi|^{O(1)}$ which compute a winning strategy S with $|\,\mathsf{rng}(S)| \leq |\pi|$. This is sufficient to provide a superpolynomial lower bound on $|\pi|$.

To argue that any lower bound for tree-like Frege +∀red arises from at least one of the reasons above, assume that Φ_n is a sequence of QBFs which are hard for tree-like Frege +∀red, but for which neither item 2 nor item 3 holds. Since neither of these hold, there exist circuits C_i with size polynomial in n computing a strategy S such that $|\,\mathsf{rng}(S)|$ is also polynomial in n. We can use these circuits to construct a tree-like Frege +∀red proof π of the normal form given in Definition 7.14. Since $|C_i|$ and $|\,\mathsf{rng}(S)|$ are polynomial, any lower bound on $|\pi|$ is due to a propositional lower bound on one of the propositional subderivations in π. □

Observe that (1) and (3) from Theorem 7.16 are nearly identical to the characterisation of lower bounds on dag-like Frege +∀red and eFrege +∀red in Theorem 6.2. If a tree-like Frege +∀red lower bound falls only under (3), it is straightforward to modify the formulas to force the universal player's response to belong to $\mathsf{rng}(S)$ for some universal winning strategy S with a polynomial-size range. The circuit lower bound in (3) can then be translated into a lower bound on *any* circuits computing a universal winning strategy, giving a lower bound for dag-like Frege +∀red as well.

The key consequence of Theorem 7.16 therefore is as follows: not only does strategy size provide a simple method for producing tree-like Frege +∀red lower bounds, it is also the *only* way to show such lower bounds which does not also entail showing a lower bound for dag-like Frege +∀red, a major open problem in QBF proof complexity.

Chapter 8

Size, Cost and Capacity: a lower bound technique for $\mathsf{P}{+}\forall\mathsf{red}$ systems

We have seen that we can obtain lower bounds for $\mathsf{P}{+}\forall\mathsf{red}$ QBF proof systems via lower bounds on the propositional proof system P, or via circuit lower bounds using a strategy extraction algorithm. However, some lower bounds shown for Σ_1^{p}-$\mathsf{QU\text{-}Res}$, such as the formulas KBKFd_n, do not fall into either of these categories. In Chapter 7, we introduced an alternative round-based strategy extraction method from [53, 62], consisting of restricting the proof by existential assignments and reading off a universal response from the $\forall$-reduction steps. This algorithm can be extended to work with blocks of variables rather than individual variables. We use this strategy extraction algorithm to give a new technique for proving $\mathsf{P}{+}\forall\mathsf{red}$ lower bounds, distinct from circuit lower bounds or propositional lower bounds.

The two key notions needed are that of the cost of a QBF and the capacity of a proof. The cost of a QBF is the number of different responses the universal player requires on a block of universal variables in any winning universal strategy. Conversely, the capacity of a proof is the largest number of universal responses that can be extracted from a single line of the proof by the round-based strategy extraction algorithm. By formalising these notions, it is clear that if a proof of a QBF has small capacity, but the QBF has large cost, then the proof must be large in order for a suitably large number of responses to be extracted.

By showing an upper bound on the capacity of any proofs in the proof systems of $\mathsf{QU\text{-}Res}$, and QBF versions of the propositional systems of Cutting Planes and Polynomial Calculus, we can therefore show lower bounds for these proof systems simply by proving a lower bound on the cost of a QBF. To exemplify such lower bounds via large cost, we use the equality formulas of [18]. We further show the applications of cost and capacity by showing lower bounds for the formulas KBKFd_n using this method, despite these formulas themselves not requiring large cost.

Unfortunately, not all proof systems admit lower bounds via large cost alone. Some proof systems, such as $\mathsf{Frege}+\forall\mathsf{red}$, can have small proofs with large capacity. We also provide an example of such an algebraic proof system, in the form of a QBF version of the Ideal Proof

System (IPS) [64], which p-simulates Frege +∀red. To do this, it is first necessary to present IPS as a line-based proof system. We can then provide short proofs of the equality formulas in this proof system. Finally, we show that this proof system nonetheless admits strategy extraction by algebraic circuits. Limiting these circuits to the relatively restrictive non-commutative formulas still allows a simulation of Frege +∀red, and so provides a potential algebraic approach to Frege +∀red lower bounds.

We first introduce the measures of cost and capacity, and prove the Size-Cost-Capacity theorem, in Section 8.1. Sections 8.2 and 8.3 introduce the Cutting Planes and Polynomial Calculus proof systems respectively, and prove capacity upper bounds for the QBF versions of these systems. In Section 8.4 we use Size-Cost-Capacity to provide a new proof of lower bounds for KBKFd_n. Lastly, we introduce the algebraic system IPS in Section 8.5, the QBF version of which can have proofs with large capacity.

8.1 The Size-Cost-Capacity theorem

We consider QBFs of the general form $\exists X_1 \forall U_1 \dots \exists X_n \forall U_n \exists X_{n+1} \cdot \phi$, where each X_i and U_i are pairwise disjoint sets of variables; X_1 or X_{n+1} may be empty. It is clear that all PCNFs are of this form, as we can combine any adjacent blocks with the same quantifier. Similarly to Chapter 7, we denote by $\alpha|_i$ the restriction of an assignment $\alpha \in \langle \mathcal{X} \rangle$ to the domain $\bigcup_{j=1}^{i} X_j$, and analogously define $\beta|_i$ for $\beta \in \langle \mathcal{U} \rangle$.[2]

In this chapter, we concern ourselves in particular with the behaviour of winning strategies on an individual block. Given a universal strategy $S : \langle \mathcal{X} \rangle \to \langle \mathcal{U} \rangle$, define $S_i : \langle X_1, \dots, X_i \rangle \to \langle U_i \rangle$ to be the projection of the strategy S to the single universal block U_i.

In order to leverage the round-based strategy extraction to provide lower bounds on the size of dag-like proofs, we wish to compare the complexity of winning strategies for the QBF with the complexity of a strategy which can be extracted from an individual line of a proof. We therefore define two terms measuring these properties of a QBF and a proof respectively.

Cost The first measure we define is *cost*. The cost of a false QBF is a measure of the number of responses needed on a single block in order to construct a universal winning strategy.

Definition 8.1 (cost). *If S is a winning universal strategy for a false QBF Φ, define* $\mathrm{cost}(S) = \max\{|\,\mathsf{rng}(S_i)\,| \mid i \in [n]\}$. *We can then define the* cost *of Φ as*

$$\mathrm{cost}(\Phi) = \min\{\mathrm{cost}(S) \mid S \text{ is a winning universal strategy for } \Phi\}.$$

Cost bears similarities to strategy size (Definition 7.4) in that both count the minimum number of responses in a universal winning strategy. The important distinction between cost and strategy size is that cost depends on the number of universal responses to a single block, whereas strategy

[2] Recall that $\langle \mathcal{X} \rangle$ is the set of all Boolean assignments to the variables of $\mathcal{X}$, i.e. all functions $\mathcal{X} \to \{0, 1\}$.

size looks at responses across all blocks. This difference is made clear in the QBFs KBKF_n. Since the only winning strategy for KBKF_n is to play $u_i = y'_i$, KBKF_n has strategy size 2^n. On the other hand, each universal block in KBKF_n consists of a single universal variable, for which there are only two possible assignments. Both assignments are required, so $\mathrm{cost}(\mathrm{KBKF}_n) = 2$.

It is clear that for any QBF Φ, $\mathrm{cost}(\Phi) \leq \rho(\Phi)$. Nevertheless, it is relatively straightforward to find QBFs with large cost. By way of example, we give the equality formulas from [18].

Definition 8.2 (Beyersdorff, Blinkhorn, Hinde [18]). *The equality formulas are the QBFs*

$$\mathrm{EQ}(n) := \exists x_1 \dots x_n \forall u_1 \dots u_n \exists t_1 \dots t_n \cdot \left(\bigwedge_{i=1}^{n} (x_i \vee u_i \vee \neg t_i) \wedge (\neg x_i \vee \neg u_i \vee \neg t_i) \right) \wedge \bigvee_{i=1}^{n} t_i.$$

The equality formulas are evidently false, as the universal player can win by playing $u_i = x_i$ for each $i \in [n]$, ensuring the restricted matrix contains $\neg t_i$ for each $i \in [n]$. Indeed, this is the only possible winning strategy for the universal player, since playing any $u_j = \neg x_j$ allows the existential player to win by playing $t_j = 1$ and $t_i = 0$ for $i \neq j$. The only winning strategy for $\mathrm{EQ}(n)$ has exponential cost, and hence so does the QBF $\mathrm{EQ}(n)$.

Lemma 8.3 (Beyersdorff, Blinkhorn, Hinde [18]). *The QBFs* $\mathrm{EQ}(n)$ *have cost* 2^n.

Capacity The other measure we define is the *capacity* of a proof. In [18], capacity is defined by introducing the notion of a response map for a line, and considering the range of these response maps. The definition we give here via response sets is equivalent, and is more convenient for the proof we give of Theorem 8.7.

Definition 8.4. *Let L be a line with the rightmost variables in L belonging to U for some universal block U, and let $X = \mathrm{var}(L) \setminus U$. A* response set *for L is a set $\mathcal{R} \subseteq \langle U \rangle$ such that for any $\alpha \in \langle X \rangle$, either $L[\alpha]$ is a tautology or there is some $\beta \in \mathcal{R}$ such that $L[\alpha \cup \beta] = \bot$.*

Note that the responses in a response set $\mathcal{R}$ assign all variables of the block U, including those not in $\mathrm{var}(L)$. This does not affect the size of a response set, as we can assume that all responses in such a response set assign variables in $U \setminus \mathrm{var}(L)$ to 0. Requiring $\mathcal{R}$ to be a subset of $\langle U \rangle$, rather than a subset of $\langle U \cap \mathrm{var}(L) \rangle$, ensures that a response set represents the universal response on the entire block rather than just on the variables of L.

Roughly speaking, the capacity of a line L is the minimum number of different responses in $\langle U \rangle$ that the universal player requires in order to be able to falsify L whenever possible. We can define this formally as the size of the smallest response set for L. The capacity of a proof is then the maximum of the capacity of its lines.

Definition 8.5 (capacity). *Let L be a line. If the rightmost variables in L are existentially quantified, define* $\mathrm{capacity}(L) = 1$. *If the rightmost variables in L are universal, then* $\mathrm{capacity}(L) = \min\{|\mathcal{R}| \mid \mathcal{R}$ *is a response set for* $L\}$. *For a* $\mathsf{P}{+}\forall\mathsf{red}$ *proof π, define*

$$\mathrm{capacity}(\pi) = \max\{\mathrm{capacity}(L) \mid L \in \pi\}.$$

Notice that capacity is a property of an individual proof, rather than a proof system. Despite this, it is still possible to show bounds on the capacity of proofs in certain proof systems. As an example, we consider QU-Res. Given a clause C, the only response needed by the universal player in a response set is one falsifying all literals in the rightmost block of C. All QU-Res proofs therefore have capacity 1.

Lemma 8.6 (Beyersdorff, Blinkhorn, Hinde [18]). *For any QU-Res refutation π of any false QBF,* $\mathrm{capacity}(\pi) = 1$.

The Size-Cost-Capacity theorem We now have everything required to give our lower bound on the size of P+∀red proofs. Since the lower bound is in fact a lower bound on the total size of the ∀-reduction steps, we work with Σ_1^p-P+∀red, as this both simplifies the proof, and demonstrates that lower bounds proved in this way are 'genuine' QBF lower bounds. Clearly all such lower bounds are also lower bounds for P+∀red.

Theorem 8.7 (Size-Cost-Capacity). *Suppose π is a Σ_1^p-P+∀red refutation of a false QBF Φ. Then*

$$|\pi| \geq \frac{\mathrm{cost}(\Phi)}{\mathrm{capacity}(\pi)}.$$

To prove the Size-Cost-Capacity theorem, we first prove two lemmas. The first of these is to show that given a response set for a line L, the ∀-reductions of L by these responses allow us to infer propositionally anything we could infer from L. This allows us to restrict ∀-reductions in Σ_1^p-P+∀red refutations to only responses from corresponding response sets without a large increase in proof size.

Lemma 8.8. *Let L be a line in a Σ_1^p-P+∀red refutation π such that the rightmost variables in L are universally quantified in some block U, and let $\mathcal{R}_L = \{\beta_1, \ldots, \beta_k\} \subseteq \langle U \rangle$ be a response set for L. Any line $L' \in \pi$ with L as a parent can be derived propositionally from $L[\beta_1] \wedge \cdots \wedge L[\beta_k]$ and the other parents of L'.*

Proof. First suppose that L' is derived from L in π by a ∀-reduction step, so $L' = L[\beta']$ for some partial assignment β to the variables of U. We require that $L[\beta_1] \wedge \cdots \wedge L[\beta_k] \models L'$. For any assignment $\alpha \in \langle \mathcal{X} \rangle$ satisfying $L[\beta_1] \wedge \cdots \wedge L[\beta_k]$, $L[\alpha \cup \beta_j] = \top$ for all $i \in [k]$. Since $\mathcal{R}_L$ is a response set for L, this can only be the case if $L[\alpha]$ is a tautology. Since $L[\alpha]$ is a tautology, certainly $L[\alpha \cup \beta'] = L'[\alpha]$ is true.

In the case where L' is derived propositionally from $L \wedge F$ for some conjunction of lines F, the argument is similar, as $L[\beta_1] \wedge \cdots \wedge L[\beta_k]$ is only satisfied by assignments α such that $L[\alpha]$ is a tautology. □

Lemma 8.8 shows that we can limit ∀-reduction to assignments in a response set while maintaining the completeness of Σ_1^p-P+∀red. If response sets are small, we can do so while only increasing the size of Σ_1^p-P+∀red proofs by a small factor. The advantage of ensuring that response sets assign

all variables in a block is that proofs in which all $\forall$-reductions are assignments to an entire block admit a round-based strategy extraction algorithm (Algorithm 4).

This strategy extraction algorithm is identical to $\mathsf{stratex}(\pi, \alpha)$ (Algorithm 3) in almost all respects. The difference is that where $\mathsf{stratex}(\pi, \alpha)$ assigned a single existential variable and a single universal variable at each round, instead an entire block of existential or universal variables are assigned at each round. To construct the universal response, the algorithm's response on U_i is γ if the line L_i^α is derived by a $\forall$-reduction U_i/γ, otherwise it is some constant response from $\langle U_i \rangle$, e.g. the all-zero assignment.

Algorithm 4 The round-based strategy extraction algorithm

function blockstratex(π,α)
  $\pi_0^\alpha \leftarrow \pi$
  $\beta \leftarrow \varnothing$
  for $1 \leq i \leq n$ **do**
    $\pi_i^\alpha \leftarrow \pi_{i-1}^\alpha[\alpha|_{X_i} \cup \beta]$
    $L_i^\alpha \leftarrow \mathrm{root}(\pi_i^\alpha)$
    if L_i^α is derived by a $\forall$-reduction $\gamma \in \langle U_i \rangle$ **then**
      $\beta \leftarrow \beta \cup \gamma$
    else
      $\beta \leftarrow \beta \cup \{u/0 \mid u \in U_i\}$
  return β

Lemma 8.9. *Suppose π is a Σ_1^p-P+$\forall$red refutation of some false QBF Φ such that every $\forall$-reduction step is a $\forall$-reduction by some assignment $\alpha \in \langle U_i \rangle$ for some universal block U_i, i.e. any $\forall$-reduction step is a $\forall$-reduction by a total assignment to a block. Then the strategy $S_\pi : \langle \mathcal{X} \rangle \to \langle \mathcal{U} \rangle$ defined by $S_\pi(\alpha) = \mathsf{blockstratex}(\pi, \alpha)$ is a winning strategy for Φ.*

This is only a minor extension to the soundness of the original round-based strategy extraction algorithm which considers each variable individually. The proof presented here is similar to that in [62] and [53], but here we also verify that the same algorithm can be applied to Σ_1^p-P+$\forall$red refutations, rather than Q-Res and LD-Q-Res.

Proof. In a Σ_1^p-P+$\forall$red refutation, we assume all lines are derived either by a $\forall$-reduction step or by a Σ_1^p-derivation. It is clear from the construction of S_π that S_π is a universal strategy, in that the response on U_i depends only on the variables in $X_1, \ldots, X_i$. We need only show that S_π is a winning strategy. We show this by induction on the number of blocks of variables in Φ, by proving that π_i^α is a Σ_1^p-P+$\forall$red refutation of $\Phi_i = \Phi[\alpha|_{X_i} \cup S_\pi(\alpha)|_{U_{i-1}}]$ for all $i \in [n]$.

In the case $i = 0$, $\pi_0^\alpha = \pi$ is a Σ_1^p-P+$\forall$red refutation of $\Phi_0 = \Phi$. Now assume that π_{i-1}^α is a Σ_1^p-P+$\forall$red refutation of Φ_{i-1}.

The leftmost block in Φ_{i-1} is U_{i-1}. Since π_{i-1}^α is the restriction of some larger Σ_1^p-P+$\forall$red proof, the root of π_{i-1}^α is the first instance of $\bot$ in π_{i-1}^α. In particular, other than deriving $\bot$, there

can be no instances of $\forall$-reduction on U_{i-1} in π^α_{i-1}. The result of any $\forall$-reduction on U_{i-1} contains no variables in U_{i-1} or to the right of U_{i-1}. Since the root of π^α_{i-1} is the only instance of $\bot$, and any instances of $\top$ are removed upon restricting, no such lines exist.

If the root is derived by a $\forall$-reduction step assigning $\gamma \in \langle U_{i-1}\rangle$, $\pi^\alpha_{i-1}[\gamma]$ is a sound Σ^p_1-P+$\forall$red refutation of $\Phi_{i-1}[\gamma]$ since the line preceding the root restricts to $\bot$, and there are no other $\forall$-reduction steps on U_{i-1} in π. If the root is derived by a Σ^p_1-P+$\forall$red deduction step, then π^α_{i-1} contains no $\forall$-reduction steps on U_{i-1}, so $\pi^\alpha_{i-1}[\gamma]$ is clearly a sound refutation of $\Phi_{i-1}[\gamma]$. Since $S_\pi(\alpha)|_{U_{i-2}} \cup \gamma = S_\pi(\alpha)|_{U_{i-1}}$ by definition of S_π, we have $\pi^\alpha_{i-1}[S_\pi(\alpha)|_{U_{i-1}}]$ is a refutation of $\Phi_{i-1}[S_\pi(\alpha)|_{U_{i-1}}]$.

Now, observe that since both Σ^p_1-derivation steps and $\forall$-reduction steps remain sound under a restriction to any existential variables, $\pi^\alpha_{i-1}[S_\pi(\alpha)|_{U_{i-1}} \cup \alpha|_{X_i}] = \pi^\alpha_i$ is a Σ^p_1-P+$\forall$red refutation of $\Phi_{i-1}[S_\pi(\alpha)|_{U_{i-1}} \cup \alpha|_{X_i}] = \Phi_i$. If other propositional deduction rules were used other than Σ^p_1-derivation, the proof would be sound in the sense that each line would still follow semantically from its parents, but depending on the proof system P, this may not be an instance of a deduction rule of P. Nevertheless, such a deduction can be performed by a Σ^p_1-derivation, so π^α_i can be considered as a Σ^p_1-P+$\forall$red refutation of Φ_i.

Repeating this process until Φ_k contains no universal variables, results in a sound Σ^p_1-P+$\forall$red refutation π^α_k of the Σ^b_1-formula Φ_k. Since π^α_k is sound, Φ_k is false and the empty strategy is a winning strategy for the universal player. If $\Phi_k = \Phi[\alpha|_{X_k} \cup S_\pi(\alpha)]$ is an unsatisfiable Σ^b_1-formula, then certainly $\Phi[\alpha \cup S_\pi(\alpha)] = \bot$, and so S_π is a winning universal strategy. □

We now use these two results to prove the Size-Cost-Capacity theorem (Theorem 8.7).

Proof (of Theorem 8.7). Given a Σ^p_1-P+$\forall$red refutation π of Φ, we construct a new Σ^p_1-P+$\forall$red refutation π' such that the number of $\forall$-reductions in π' is at most $\mathrm{capacity}(\pi) \cdot |\pi|$. Since all $\forall$-reductions in π' will reduce by an assignment to an entire block, we can then construct a strategy S using Algorithm 4. The lower bound then follows since $\mathrm{cost}(S) \geq \mathrm{cost}(\Phi)$.

Let π be a Σ^p_1-P+$\forall$red refutation of Φ. For each line $L \in \pi$ in which the rightmost variables are universal, pick some response set $\mathcal{R}_L$ for L such that $|\mathcal{R}_L| = \mathrm{capacity}(L)$. In particular, for each suitable line L, $|\mathcal{R}_L| \leq \mathrm{capacity}(\pi)$.

We now construct the proof π' by considering each line in π in order. For each line $L \in \pi$, if the rightmost variables in L are existentially quantified, or if $\mathrm{var}(L) = \varnothing$, we do nothing. Otherwise, the line L has a response set $\mathcal{R}_L = \{\beta_1, \ldots, \beta_k\}$. We include in π' the lines $L[\beta_1], \ldots, L[\beta_k]$, each derived by $\forall$-reduction from L. For any line of π which is derived from L by a deduction rule of Σ^p_1-P+$\forall$red, this deduction step is replaced in π' by a Σ^p_1-derivation, replacing L in the set of antecedents with some subset of $L[\beta_1], \ldots, L[\beta_k]$. This is always a sound Σ^p_1-derivation step by Lemma 8.8. We refer to the line L and the additional lines $L[\beta_1], \ldots, L[\beta_k]$ as the *response tree for* L.

For each line $L \in \pi$, the response tree for L has $|\mathcal{R}_L| \leq \mathrm{capacity}(\pi)$ leaves. Observe that all $\forall$-reduction steps in π' occur as a leaf of the response tree for L for some $L \in \pi$, and hence the

number of $\forall$-reduction steps in π' is at most $\mathrm{capacity}(\pi) \cdot |\pi|$. Moreover, any $\forall$-reduction in π' is a reduction by an assignment to an entire block. For such a proof, we see from Lemma 8.9 that we can apply the round-based strategy extraction algorithm described in Algorithm 4 to obtain a winning universal strategy for Φ.

Let S be the winning universal strategy constructed by this round-based strategy extraction algorithm on π', i.e. $S(\alpha) = \mathsf{blockstratex}(\pi', \alpha)$. Observe that for each line $L \in \pi'$ and for each universal block U_i, there is a unique response returned if L is chosen at the ith round. If L is derived by a $\forall$-reduction on U_i from some line $L' \in \pi$, then L is of the form $L'[\beta]$ for some $\beta \in \langle U_i \rangle$ and the response returned is β. Otherwise, the response returned is some fixed constant response, which can be chosen to be identical to some other response appearing in a $\forall$-reduction step on U_i.

It is therefore clear that for any block U_i, $|\,\mathsf{rng}(S_i)|$ is at most the number of distinct $\forall$-reductions in π', so $|\,\mathsf{rng}(S_i)| \leq \mathrm{capacity}(\pi) \cdot |\pi|$. However, since S is a universal winning strategy for Φ, there is some i such that $|\,\mathsf{rng}(S_i)| \geq \mathrm{cost}(\Phi)$ and hence $\mathrm{cost}(\Phi) \leq \mathrm{capacity}(\pi) \cdot |\pi|$. We conclude that $|\pi| \geq \frac{\mathrm{cost}(\Phi)}{\mathrm{capacity}(\pi)}$. □

Since the Size-Cost-Capacity theorem provides lower bounds for Σ_1^{p}-P+$\forall$red, it is clear that any lower bounds arising this way are 'genuine' QBF lower bounds, and do not arise due to propositional hardness. We can see this explicitly, and prove the lower bound for P+$\forall$red directly, by observing that in the construction of π' in our proof, it would be sufficient to replace the line L by its response tree if and only if the line L is $\forall$-reduced in the proof. The number of $\forall$-reduction steps is then increased by only a factor of $\mathrm{capacity}(\pi)$ and the lower bound on $|\pi'|$ is in fact a lower bound on the number of $\forall$-reduction steps in π'.

Given the capacity upper bound for QU-Res (Lemma 8.6), the Size-Cost-Capacity theorem gives a simple lower bound for Σ_1^{p}-QU-Res refutations in terms of cost.

Theorem 8.10 (Beyersdorff, Blinkhorn, Hinde [18]). *Let π be a Σ_1^{p}-QU-Res refutation of a QBF Φ. Then $|\pi| \geq \mathrm{cost}(\Phi)$.*

An immediate application of this lower bound is to provide an exponential lower bound on the size of Σ_1^{p}-QU-Res refutations of the equality formulas.

Corollary 8.11 (Beyersdorff, Blinkhorn, Hinde [18]). *Any Σ_1^{p}-QU-Res refutation of $\mathrm{EQ}(n)$ requires size at least 2^n.*

Not all P+$\forall$red proof systems have small capacity bounds. By way of example, it has been shown that there are polynomial-size Frege +$\forall$red refutations of the equality formulas.

Theorem 8.12 (Beyersdorff, Blinkhorn, Hinde [18]). *There are polynomial-size* Frege +$\forall$red *refutations of the equality formulas $\mathrm{EQ}(n)$.*

As a consequence of this, [18] observed that polynomial-size Frege +$\forall$red proofs can have large capacity. If π_n is a polynomial-size refutation of $\mathrm{EQ}(n)$, then by Theorem 8.7, we have that

$\text{capacity}(\pi_n) \geq \frac{\text{cost}(\text{EQ}(n))}{|\pi_n|} = \frac{2^n}{n^{O(1)}} = 2^{\Omega(n)}$. Such capacity lower bounds can be be shown for other systems such as $\mathbf{AC}^0_3$-Frege +∀red, where the equality formulas also have short proofs.

Nonetheless, the Size-Cost-Capacity theorem (Theorem 8.7) provides a general method for proving lower bounds for several P+∀red QBF proof systems. If we can show an upper bound on the capacity of proofs in such a proof system, Size-Cost-Capacity gives a lower bound on the size of proofs of a QBF Φ in this system based only on $\text{cost}(\Phi)$. We now introduce several previously studied propositional proof systems based on algebraic reasoning, and their QBF extensions, and analyse the capacities of these proof systems.

8.2 Cutting Planes

The first such proof system we consider is Cutting Planes (CP). Inspired by integer linear programming, Cutting Planes works with linear inequalities as lines. The literals x and $\neg x$ are translated to the linear sums x and $1 - x$ respectively. The requirement that each clause is satisfied is then translated to the requirement that the sum of the literals in each clause is at least 1.

Definition 8.13. *A Cutting Planes (CP) derivation [44] contains lines consisting of linear inequalities $a_1x_1 + \cdots + a_nx_n \geq A$, where $x_1, \ldots, x_n$ are variables, and $a_1, \ldots, a_n, A \in \mathbb{Z}$. The derivation rules of CP are shown in Figure 15, and a CP refutation is a CP derivation of the trivial falsity $0 \geq 1$.*

Clause Axiom:	$\dfrac{}{\sum_{l \in C} R(l) \geq 1}$	C a clause, $R(l) = x$ if $l = x$, $R(l) = 1 - x$ if $l = \neg x$
Boolean Axioms:	$\dfrac{}{x \geq 0}$ $\dfrac{}{-x \geq -1}$	for any variable x
Linear Combination:	$\dfrac{\sum_i a_i x_i \geq A \qquad \sum_i b_i x_i \geq B}{\sum_i (\alpha a_i + \beta b_i) x_i \geq \alpha A + \beta B}$	for any $\alpha, \beta \in \mathbb{N}$
Division:	$\dfrac{\sum_i c a_i x_i \geq A}{\sum_i \alpha a_i x_i \geq \lceil \frac{A}{c} \rceil}$	for any non-zero $c \in \mathbb{N}$

Fig. 15. The derivation rules of Cutting Planes [44]

Cutting Planes can also be defined with a variation of the division rule in which all coefficients a_i are replaced by $\lceil \frac{a_i}{c} \rceil$, or in which any linear inequality can appear as a line, but all such definitions are equivalent. In particular, we may sometimes refer to a linear inequality not of this form in a line, but we assume it is translated into the form of a CP line in the proof.

Example 8.14. Figure 16 contains one possible CP refutation of the CNF $(x \vee y) \wedge (\neg x \vee y \vee z) \wedge (\neg y \vee z) \wedge (\neg z)$. All constants α and β in addition steps are 1 unless labelled otherwise.

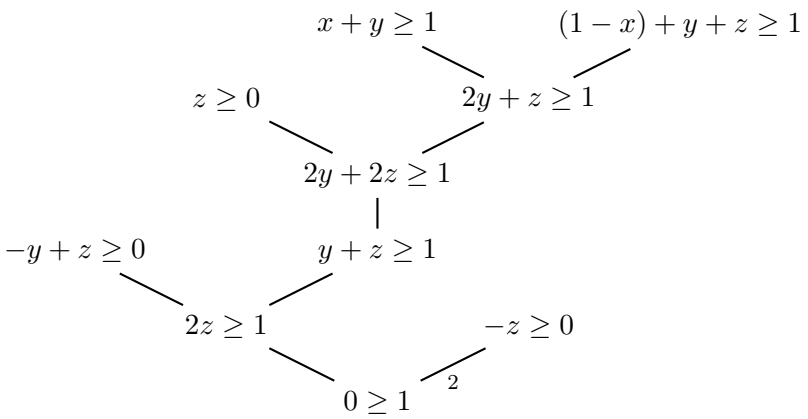

Fig. 16. An example CP proof

Considering the translation used in axiom introduction as the representation of a clause by an inequality, it is easy to see that CP p-simulates Resolution. A resolution step corresponds to a linear combination step adding the two inequalities representing the two clauses. Addition of Boolean axioms and a division step are then all that is required to deduce the inequality representing the resulting clause. The derivation of $y + z \geq 1$ from $x + y \geq 1$ and $(1 - x) + y + z \geq 1$ in Figure 16 is an example of this, in this case resolving $x \vee y$ and $\neg x \vee y \vee z$ to give $y \vee z$.

It is fairly straightforward to define the QBF proof system CP+$\forall$red by adding the $\forall$-reduction rule to CP [24]. Although the lines of CP are integer linear inequalities, in order to ensure the soundness of CP+$\forall$red as a QBF proof system, $\forall$-reduction steps are limited to substituting universal variables by either 0 or 1.

Despite CP+$\forall$red being a strictly stronger proof system than QU-Res, since CP has short proofs of PHP_n, any CP+$\forall$red proof still has unit capacity.

Lemma 8.15. *For any* CP+$\forall$red *or* Σ_1^{p}-CP+$\forall$red *derivation* π, $\mathrm{capacity}(\pi) = 1$.

Proof. We need only show that for any line $L \in \pi$ with rightmost block U, there is a response set $\mathcal{R}_L$ for L of size 1. Let $X = \mathrm{var}(L) \setminus U$. Then L is of the form $\sum_{x \in X} a_x x + \sum_{u \in U} b_u u \geq c$ where $a_x, b_u, c \in \mathbb{Z}$ for all $x \in X, u \in U$. For any assignment $\alpha \in \langle X \rangle$, the restricted line $L[\alpha]$ is therefore $\sum_{u \in U} b_u u \geq c'$ for some constant $c' \in \mathbb{Z}$.

Define the assignment $\beta_L \in \langle U \rangle$ by setting $\beta(u) = 0$ if $b_u \geq 0$ and $\beta(u) = 1$ otherwise. Assigning the variables of U according to β_L minimises the value of $\sum_{u \in U} b_u u$. If any assignment falsifies the inequality $\sum_{u \in U} b_u u \geq c'$, it must be the case that β_L does so, and so $\mathcal{R}_L = \{\beta_L\}$ is a response set for L. For every line $L \in \pi$, $|\mathcal{R}_L| = 1$ and hence $\mathrm{capacity}(L) = 1$. We conclude that $\mathrm{capacity}(\pi) = 1$. □

As in the case of QU-Res, this immediately gives lower bounds for Σ_1^{p}-CP+∀red using only cost lower bounds.

Theorem 8.16. *If π is a Σ_1^{p}-CP+∀red refutation of a QBF Φ then $|\pi| \geq \mathrm{cost}(\Phi)$.*

Proof. By Lemma 8.15, $\mathrm{capacity}(\pi) = 1$; applying Theorem 8.7 immediately gives the bound $|\pi| \geq \mathrm{cost}(\Phi)$. □

Hence, even in the stronger proof system of CP+∀red, we still have a straightforward proof that the equality formulas require refutations of size 2^n simply by looking at the cost of the formulas.

Corollary 8.17. *If π is a* CP+∀red *refutation of the equality formulas $\mathrm{EQ}(n)$, then $|\pi| \geq 2^n$.*

8.3 Polynomial Calculus

Rather than using linear inequalities, Polynomial Calculus (PC) [40] has lines consisting of polynomial equations over a fixed field $\mathbb{F}$. As in Cutting Planes, PC translates the literal $\neg x$ as $1 - x$. The clauses of a CNF can then be treated as the assertion that the product of their literals is 0; observe that in this context we consider 0 to be 'true' and 1 to be 'false'. A proof of unsatisfiability is then a derivation of $1 = 0$ by linear combinations and multiplication by variables.

The lines in PC proofs are represented as sparse polynomials, and so the size of a PC proof can be measured by the total number of monomials in the polynomial equations. However, the translation of $\neg x$ as $1 - x$ results in even a single clause $\bigvee_{i=1}^{n} \neg x_i$ requiring an exponential number of monomials. To alleviate this issue, Polynomial Calculus with Resolution (PCR) [2] introduces a new variable $\bar{x}$ as the translation of $\neg x$, with the additional requirement that $\bar{x} = 1 - x$ as an axiom. Using these additional variables, each clause can be represented as a single monomial. We give the definition only for the stronger system PCR; PC is defined similarly with any references to $\bar{x}$ replaced by $1 - x$.

Definition 8.18. *A Polynomial Calculus with Resolution derivation [2] contains lines of the form $q(\boldsymbol{x}, \bar{\boldsymbol{x}}) = 0$ where $q(\boldsymbol{x}, \bar{\boldsymbol{x}})$ is a polynomial in the variables $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\bar{\boldsymbol{x}} = (\bar{x}_1, \ldots, \bar{x}_n)$ over a fixed field $\mathbb{F}$. The derivation rules of* PCR *are given in Figure 17, and a* PCR *refutation is a derivation of the line $1 = 0$.*

In general, the linear combination and multiplication rules of PC and PCR are not sufficient to allow these proof systems to be implicationally complete. However, in the instances we are interested in, in which we can introduce the Boolean axioms forcing variables to take 0/1 values, we do have implicational completeness [54].

Example 8.19. Figure 18 contains one possible PCR refutation of the CNF $(x \vee y) \wedge (\neg x \vee y \vee z) \wedge (\neg y \vee z) \wedge (\neg z)$. A similar PC refutation could be constructed by replacing instances of $\bar{x}$ with $(1 - x)$.

Clause Axiom:	$\dfrac{}{\prod_{l \in C} V(l) = 0}$	C a clause, $V(x) = x$, $V(\neg x) = \bar{x}$
Boolean Axiom:	$\dfrac{}{y^2 - y = 0}$ $\dfrac{}{x + \bar{x} - 1 = 0}$	$y = x$ or $y = \bar{x}$, for any variable x
Linear Combination:	$\dfrac{p(\boldsymbol{x}) = 0 \qquad q(\boldsymbol{x}) = 0}{\alpha p(\boldsymbol{x}) + \beta q(\boldsymbol{x}) = 0}$	for any $\alpha, \beta \in \mathbb{F}$
Multiplication:	$\dfrac{p(\boldsymbol{x}) = 0}{y \cdot p(\boldsymbol{x}) = 0}$	$y = x$ or $y = \bar{x}$ for some variable x

Fig. 17. The derivation rules of Polynomial Calculus with Resolution [2, 40]

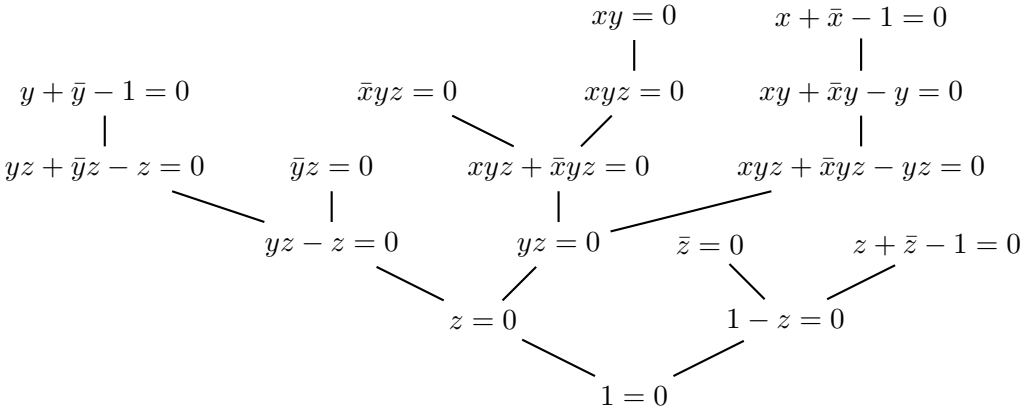

Fig. 18. An example PCR proof

Similarly to CP, the resolution rule can be simulated by the use of linear combination, with some use of other axioms to 'tidy up.' The derivation of $yz = 0$ from $xy = 0$ and $\bar{x}yz = 0$ in Figure 18 is an example of this, also representing derivation of $y \vee z$ from $x \vee y$ and $\neg x \vee y \vee z$.

As was the case with CP, PC and PCR can be extended to QBF proof systems PC+∀red and PCR+∀red by including a ∀-reduction rule allowed the substitutions $u/0$ or $u/1$ for some rightmost universal variable u. In the case of PCR+∀red, the ∀-reduction u/b also performs the substitution $\bar{u}/(1 - b)$ in order to preserve the soundness of this system.

In contrast to the situation in QU-Res and CP+∀red, not all proofs in PCR+∀red have unit capacity. For a simple example, consider the line $x(1 - u) + (1 - x)u = 0$. This polynomial equation is clearly satisfied if and only if $x = u$. In order to falsify this line in a PCR+∀red proof, it is therefore necessary for the response to be $u = 1 - x$. The only possible response set for this line therefore contains both assignments to u, and if a PCR+∀red proof π contains such a line, then $\text{capacity}(\pi) \geq 2$.

Indeed, given a QBF Φ with a universal block U_i containing n variables, it is possible to construct a polynomial equation such that all 2^n responses in $\langle U_i \rangle$ are required in a response set. However, such a line would require an exponential number of monomials. Since PCR+∀red proofs are measured by the number of monomials, any proof containing such a line would therefore be of exponential size. To provide a suitable capacity upper bound for PCR+∀red, it suffices to upper bound the size of a response set by the number of monomials in that line.

Theorem 8.20. *Let L be a line of a* PCR+∀red *proof such that L has rightmost universal block U and L contains M monomials. Then* $\mathrm{capacity}(L) \leq M$.

Proof. To show that $\mathrm{capacity}(L) \leq M$, it suffices to find a response set $\mathcal{R}_L$ for L such that $|\mathcal{R}_L| \leq M$.

Define $X = \mathrm{var}(L) \setminus U$. We can now write L as

$$\sum_{j=1}^{N} f_j v_j = 0$$

where each f_j is a polynomial (*not* necessarily a single monomial) in the variables of X and each v_j are distinct monomials in the variables U. We denote by $f_j[\alpha]$ and $v_j[\beta]$ the values (in $\mathbb{F}$) obtained by evaluating f_j, respectively v_j, according to the assignments $\alpha \in \langle X \rangle$, respectively $\beta \in \langle U \rangle$. Since v_j is simply a product of variables in U, we have $v_j[\beta] \in \{0, 1\}$ for any $\beta \in \langle U \rangle$.

Observe that since L contains at most M monomials, $N \leq M$. We now construct a response set $\mathcal{R}_L$ such that $|\mathcal{R}_L| \leq N$.

Such a response set must contain a response falsifying $L[\alpha]$ for all $\alpha \in \langle X \rangle$ such that $L[\alpha]$ is falsifiable. To begin, enumerate the assignments in $\langle X \rangle$ as $\langle X \rangle = \{\alpha_1, \ldots, \alpha_m\}$. We then construct a sequence of sets $\mathcal{R}_L^i$ such that each $\mathcal{R}_L^i$ is a 'partial response set' for L, in the sense that $\mathcal{R}_L^i$ contains responses falsifying $L[\alpha_k]$ for each $k \leq i$ such that $L[\alpha_k]$ is falsifiable. We further ensure that $|\mathcal{R}_L^i| \leq N$ for each $0 \leq i \leq m$, so defining $\mathcal{R}_L = \mathcal{R}_L^m$ gives the response set we require.

Our construction of the $\mathcal{R}_L^i$ is inductive, and in particular $\mathcal{R}_L^i \subseteq \mathcal{R}_L^{i+1}$, so the construction of $\mathcal{R}_L$ amounts to going through each assignment $\alpha \in \langle X \rangle$ in turn, and ensuring $\mathcal{R}_L$ contains a suitable response. Since we are aiming to minimise $|\mathcal{R}_L|$, we do not add a response if $\mathcal{R}_L^{i-1}$ already contains a response falsifying $L[\alpha_i]$, or if $L[\alpha_i]$ is a tautology. If we have not already chosen a suitable response, then such a response is added to $\mathcal{R}_L$. To ensure our upper bound on $|\mathcal{R}_L|$, we show that evaluating the monomials v_j by α constructs an injection from $\mathcal{R}_L$ into a linearly independent subset of $\mathbb{F}^N$.

If L is a tautology, then for any $\alpha \in \langle X \rangle$, $L[\alpha]$ cannot be falsified, and so $\mathcal{R}_L = \varnothing$ is a response set for L. We therefore assume without loss of generality that L is not a tautology, and in particular that $\langle X \rangle$ is enumerated such that $L[\alpha_1]$ is falsifiable.

For any $\beta \in \langle U \rangle$, denote by $\boldsymbol{v}[\beta]$ the vector $(v_1[\beta], \ldots, v_N[\beta]) \in \{0, 1\}^N$. We now inductively define the sets $\mathcal{R}_L^i \supseteq \mathcal{R}_L^{i-1}$ such that they are the partial response sets we require. Furthermore, we

show by induction on i that the set $V_L^i = \{\boldsymbol{v}[\beta] \mid \beta \in \mathcal{R}_L^i\}$ is linearly independent as a subset of $\mathbb{F}^N$ and that $|V_L^i| = |\mathcal{R}_L^i|$. Since $V_L^i \subseteq \{0,1\}^N$, this provides the necessary upper bound on $|\mathcal{R}_L^i|$.

We begin by defining $\mathcal{R}_L^0 = \varnothing$. It is clear that $V_L^0 = \varnothing$, so $|V_L^0| = |\mathcal{R}_L^0| = 0$ and that V_L^0 is linearly independent. We can also define $\mathcal{R}_L^1 = \{\beta_1\}$ for some $\beta_1 \in \langle U \rangle$ such that $L[\alpha_1 \cup \beta_1] = \bot$. It is clear that we also have $|V_L^1| = 1$, and since $\boldsymbol{v}[\beta_1] \neq \mathbf{0}$, V_L^i is linearly independent.

Suppose we now have defined $\mathcal{R}_L^{i-1}$ for some $2 \leq i \leq m$ such that $|\mathcal{R}_L^{i-1}| = |V_L^{i-1}|$ and V_L^{i-1} is linearly independent. We define $\mathcal{R}_L^i$ as follows:

- Suppose $L[\alpha_i]$ is a tautology, i.e. $\sum_{j=1}^N f_j[\alpha_i] v_j[\beta] = 0$ for all $\beta \in \langle U \rangle$. In this case, it is not necessary for $\mathcal{R}_L^i$ to contain a response falsifying $L[\alpha_i]$, so we define $\mathcal{R}_L^i = \mathcal{R}_L^{i-1}$. We also have $V_L^i = V_L^{i-1}$, so it is clear by induction that $|\mathcal{R}_L^i| = |V_L^i|$ and that V_L^i is linearly independent, since this is the case for $\mathcal{R}_L^{i-1}$ and V_L^{i-1}.
- Else, suppose $L[\alpha_i \cup \beta] = \bot$ for some $\beta \in \mathcal{R}_L^{i-1}$. We can then define $\mathcal{R}_L^i = \mathcal{R}_L^{i-1}$. As previously, $\mathcal{R}_L^i = \mathcal{R}_L^{i-1}$, and so $|\mathcal{R}_L^i| = |V_L^i|$ and V_L^i is linearly independent.
- Else, it is the case that $L[\alpha_i]$ is not a tautology, but $L[\alpha_i \cup \beta] = \top$ for all $\beta \in \mathcal{R}_L^{i-1}$. Pick $\beta' \in \langle U \rangle$ such that $L[\alpha_i \cup \beta'] = \bot$ and define $\mathcal{R}_L^i = \mathcal{R}_L^{i-1} \cup \{\beta'\}$. It is clear this is a suitable partial response set. We must show that $|\mathcal{R}_L^i| = |V_L^i|$ and that V_L^i is linearly independent.
  Enumerate $\mathcal{R}_L^{i-1}$ as $\mathcal{R}_L^{i-1} = \{\beta_1, \ldots, \beta_k\}$ for some $k \geq 1$. It is clear that $\boldsymbol{v}[\beta'] \neq \boldsymbol{v}[\beta_l]$ for any $1 \leq l \leq k$, else we would have $\sum_{j=1}^N f_j[\alpha_i] v_j[\beta_l] = \sum_{j=1}^N f_j[\alpha_i] v_j[\beta'] \neq 0$, and so $L[\alpha_i \cup \beta_l] = \bot$. We therefore have $|V_L^i| = |V_L^{i-1}| + 1 = |\mathcal{R}_L^{i-1}| + 1 = |\mathcal{R}_L^i|$.
  For the linear independence of V_L^i, we assume towards a contradiction that there is some linear dependence relation on V_L^i. Since V_L^{i-1} is linearly independent, $\boldsymbol{v}[\beta'] \neq \mathbf{0}$ must have a non-zero coefficient in any such linear combination. We can therefore find constant $c_1, \ldots, c_k \in \mathbb{F}$ such that $\sum_{t=1}^k c_t \boldsymbol{v}[\beta_t] = \boldsymbol{v}[\beta']$. We can use these same constants to construct a linear combination of the $\sum_{j=1}^N f_j[\alpha_i] v_j[\beta_t]$, by assumption all equal to zero, summing to $\sum_{j=1}^N f_j[\alpha_i] v_j[\beta']$, which by choice of β' is non-zero.

$$0 = \sum_{t=1}^k c_t \sum_{j=1}^N f_j[\alpha_i] v_j[\beta_t] = \sum_{j=1}^N f_j[\alpha_i] \sum_{t=1}^k c_t v_j[\beta_t] = \sum_{j=1}^N f_j[\alpha_i] v_j[\beta'] \neq 0$$

  From this contradiction, we conclude that no such constants c_t exist, and hence that V_L^i is a linearly independent set.

By construction, it is clear that $\mathcal{R}_L = \mathcal{R}_L^m$ is a response set for L. The set V_L^m is a linearly independent subset of $\mathbb{F}^N$ and so $|V_L^m| \leq N$. Since $|\mathcal{R}_L^m| = |V_L^m|$, we conclude that $\mathcal{R}_L$ is a response set for L with $|\mathcal{R}_L| \leq N$, so $\mathrm{capacity}(L) \leq N \leq M$. □

The effect of this bound on the capacity of lines is to show that proofs with large capacity must contain a line with a large number of monomials, which would require the proof itself to be large. This provides a lower bound for PCR+∀red proofs of a QBF Φ based only on $\mathrm{cost}(\Phi)$, since small proofs also have small capacity.

Theorem 8.21. *Let π be a* PCR+∀red *or* Σ_1^p-PCR+∀red *refutation of a QBF Φ. Then* $|\pi| \geq \sqrt{\text{cost}(\Phi)}$.

Proof. The size of π is measured by the number of monomials in π, so any line of π contains at most $|\pi|$ monomials, and hence $\text{capacity}(\pi) \leq |\pi|$. Applying the Size-Cost-Capacity theorem (Theorem 8.7), we have that $|\pi| \geq \frac{\text{cost}(\Phi)}{|\pi|}$, i.e. $|\pi| \geq \sqrt{\text{cost}(\Phi)}$. □

As with QU-Res and CP+∀red, this gives us an exponential lower bound on the size of PCR+∀red refutations of the equality formulas.

Corollary 8.22. *If π is a* PCR+∀red *refutation of the equality formulas* $\text{EQ}(n)$*, then* $|\pi| \geq 2^{\Omega(n)}$.

8.4 Lower bounds for KBKFd_n via Size-Cost-Capacity

The lower bounds we have shown for the proof systems above require large cost, rather than large strategy size. While this is evidently a stronger requirement for a lower bound, it is equivalent in the case of QBFs with bounded alternation. That is, for a family of QBFs in which the prefixes are all Σ_k^b-prefixes for some constant k, a superpolynomial lower bound on strategy size is sufficient to provide a superpolynomial lower bound on proof size.

Lemma 8.23. *Let Φ_n be a family of false QBFs such that each Φ_n has a Σ_k^b-prefix. Each Φ_n requires proofs of size at least $\rho(\Phi_n)^{\frac{1}{k}}$ in* QU-Res*,* CP+∀red *and* PCR+∀red.

Proof. We use the cost-based lower bounds we have already shown for these proof systems. We need only show that $\text{cost}(\Phi_n) \geq \rho(n)^{\frac{2}{k}}$.

Since Φ_n has a Σ_k^b-prefix, the prefix contains at most $\frac{k}{2}$ universal blocks, $U_1, \ldots, U_{\frac{k}{2}}$. Given any winning universal strategy $S : \langle \mathcal{X} \rangle \to \langle \mathcal{U} \rangle$, we see that

$$\rho(\Phi_n) \leq \mathsf{rng}(S) \leq \prod_{i=1}^{\frac{k}{2}} \mathsf{rng}(S_i) \leq \left(\max_i(\mathsf{rng}(S_i)) \right)^{\frac{k}{2}}$$

and in particular, $\max_i(\mathsf{rng}(S_i)) \geq \rho(\Phi_n)^{\frac{2}{k}}$. By definition, $\text{cost}(\Phi_n)$ is the minimum of $\max_i(\mathsf{rng}(S_i))$ over all winning strategies S, and so $\text{cost}(\Phi_n) \geq \rho(\Phi_n)^{\frac{2}{k}}$. □

In [16], it was observed that any family of formulas separating the instantiation calculus IR-calc from ∀Exp+Res must have an unbounded quantifier prefix. Lemma 8.23 provides a similar result for QU-Res: any lower bound for ∀Exp+Res which is not simply a Resolution lower bound on the expanded formula must arise through a lower bound on strategy size. Any family of QBFs providing a genuinely QBF separation between QU-Res and ∀Exp+Res must therefore have unbounded quantifier alternation in the prefix. Given that separations with bounded quantifer alternation between ∀Exp+Res and QU-Res are known, such as the QPARITYformulas [21], this is a striking comparison between the two approaches to QBF Resolution systems, and suggests that on

instances with a small number of quantifier blocks, expansion-based solving may have significantly more potential than CDCL-based solving.

In the case of QBFs with an unbounded quantifier prefix, strategy size will not suffice, and so we must focus on cost. As previously mentioned, the QBFs KBKF_n and KBKFd_n have large strategy size, yet their cost is constant due to the unbounded number of quantifier blocks in their prefixes. As a result, we cannot directly apply the Size-Cost-Capacity theorem to give lower bounds for these formulas. However, by rearranging the quantifier prefix to construct an even weaker QBF than KBKFd_n, we can then apply Size-Cost-Capacity using the capacity upper bounds we have shown to provide lower bounds for these weaker formulas. These lower bounds immediately give lower bounds on proofs of KBKF_n and KBKFd_n.

Recall from Definition 3.3 that KBKFd_n is defined as

$$\mathrm{KBKFd}_n := \exists y_0(\exists y_1 y_1' \forall u_1 v_1) \dots (\exists y_n y_n' \forall u_n v_n) \exists y_{n+1} y_{n+2} \dots y_{2n} \cdot \bigwedge_{i=0}^{2n} C_i \wedge C_i'.$$

We define the QBF κ_n by quantifying the doubling variables v_i together in the final universal block, rather than adjacent to their corresponding variables u_i.

$$\kappa_n := \exists y_0(\exists y_1 y_1' \forall u_1) \dots (\exists y_n y_n' \forall u_n)(\forall v_1 \dots v_n) \exists y_{n+1} y_{n+2} \dots y_{2n} \cdot \bigwedge_{i=0}^{2n} C_i \wedge C_i'.$$

It is clear that the universal winning strategy for KBKFd_n, of playing $u_i = v_i = y_i'$, is also a winning strategy for κ_n, since each v_i remains to the right of y_i and y_i' and hence κ_n is false.

Since the only difference between KBKF_n and κ_n is that universal variables in the prefix have been moved further right, any $\forall$-reduction steps in a P+$\forall\mathsf{red}$ or Σ_1^p-P+$\forall\mathsf{red}$ refutation of KBKF_n will also be sound under the prefix of κ_n.

Lemma 8.24. *Suppose $\Phi = \Pi \cdot \phi$ is a false QBF, and Π' is a relaxation of the quantifier prefix Π. If π is a P+$\forall\mathsf{red}$ or Σ_k^p-P+$\forall\mathsf{red}$ refutation of $\Phi' = \Pi' \cdot \phi$, then there is a P+$\forall\mathsf{red}$ or Σ_k^p-P+$\forall\mathsf{red}$ refutation of Φ of size $|\pi|^{O(1)}$.*

Proof. To refute Φ, we follow the deduction steps in π. Any propositional step is clearly sound under any prefix. In the case of a $\forall$-reduction, if the variable u is assigned in a $\forall$-reduction on a line L in π, then u is right of any existential variables in L under Π'. Since Π' is a relaxation of Π, u must be quantified universally in Π, and any existential variable left of u in Π' is either left of u in Π, or is quantified universally in Π. Since all existential variables in L are left of u in Π, u can be $\forall$-reduced from L under the prefix Π. This completes the proof for P+$\forall\mathsf{red}$.

In the case of Σ_k^p-P+$\forall\mathsf{red}$, it suffices to observe that since Π' is a relaxation of Π, any relaxation of Π' is also a relaxation of Π. Any Σ_k^p-derivation steps in π can therefore be performed as a single Σ_k^p-derivation step in a refutation of Φ. □

As a consequence, proving a superpolynomial lower bound on P+∀red refutations of κ_n is sufficient to show such a lower bound on refutations of KBKFd_n. We do this using Size-Cost-Capacity, by showing that κ_n has exponential cost.

Theorem 8.25. *The QBFs κ_n have cost 2^n.*

Proof. Since κ_n is false, some universal winning strategy exists. Let S be such a universal winning strategy for κ_n. We consider the response of S to the 2^n distinct existential assignments in

$$A = \{\alpha \in \langle\{y_1, y'_1, \ldots, y_n, y'_n\}\rangle \mid \alpha(y_k) \neq \alpha(y'_k) \text{ for all } k \in [n]\}.$$

For any assignment $\alpha \in A$, S must respond by setting $u_i = y'_i$ for each $i \in [n]$. If this were not the case, let u_k be the first universal variable such that $u_k = y_k$. For all $i < k$, either y_i or y'_i is false, and so C_i and C'_i are satisfied. The clauses C_k and C'_k are satisfied since $u_k = y_k$, and so the existential player can win by subsequently assigning $y_j = y'_j = 1$ for all $j > k$, as each clause C_j, C'_j contains one of the literals y_j or y'_j.

In order to show that $\mathrm{cost}(S) = 2^n$, we show that, for any $\alpha \in A$, the only winning response on the variables v_i is to play $v_i = u_i = y'_i$. Since the v_i appear in a single block in the prefix of κ_n, this requires that $\mathrm{cost}(S) = |A| = 2^n$. We demonstrate this in the specific case where $\alpha(y_i) = 1, \alpha(y'_i) = 0$ for all $i \in [n]$ – all other assignments in A are similar.

Restricting $\bigwedge_{i=0}^{2n} C_i \wedge C'_i$ by α, and by the only possible winning universal response β on the variables u_i, where $\beta(u_i) = 0$ for all $i \in [n]$, the restricted clauses remaining in the matrix are

$$\begin{aligned} C'_n|_{\alpha\cup\beta} &= \{v_n, \neg y_{n+1}, \ldots, \neg y_{n+n}\} \\ C'_{n+t}|_{\alpha\cup\beta} &= \{v_t, y_{n+t}\} \qquad\qquad \text{for each } 1 \leq t \leq n. \end{aligned}$$

If $S_n(\alpha)$ sets any $v_k = 1$, then this conjunction of clauses can be satisfied by setting $y_{n+k} = 0$ and $y_{n+t} = 1$ for all $t \neq k$. The unique response on the v_k for $S_n(\alpha)$ is therefore to set $v_i = u_i = y'_i$ for all $i \in [n]$. In the case of any other $\alpha \in A$, the restricted matrix will be similar with the polarity of the literals on v_i flipped as appropriate. We conclude that $|\,\mathsf{rng}(S_n)| = 2^n$ and hence $\mathrm{cost}(\kappa_n) = 2^n$. □

We can therefore immediately obtain the following hardness result, which was known for QU-Res [9], but also lifts to CP+∀red and PCR+∀red.

Theorem 8.26. *Any QU-Res, CP+∀red or PCR+∀red refutation of KBKFd_n requires size $2^{\Omega(n)}$.*

Proof. The cost lower bound of Theorem 8.25 and the Size-Cost-Capacity theorem (Theorem 8.7) immediately give a lower bound of $2^{\Omega(n)}$ on QU-Res, CP+∀red or PCR+∀red refutations of κ_n. In each of these systems, using Lemma 8.24 we can construct from a refutation of KBKF_n a refutation of κ_n with at most a polynomial increase in size. Any such refutation of KBKF_n must therefore have size at least $2^{\Omega(n)}$. □

8.5 Ideal Proof System

While Size-Cost-Capacity is able to prove lower bounds for a variety of proof systems, some proof systems are able to express lines with sufficiently high capacity that we cannot obtain superpolynomial lower bounds through high cost. The short Frege +∀red proofs of the equality formulas (Theorem 8.12) demonstrate that, despite their large cost, Frege +∀red proofs can have high capacity. We now give an example of an algebraic proof system which also has high capacity. This system is an extension to QBF of the recently introduced Ideal Proof System (IPS) [64], which works with arithmetic circuits, rather than sparse polynomials as in PCR.

Definition 8.27 (Grochow and Pitassi [64]). *Fix a field $\mathbb{F}$. Given a system of polynomial equations $f_j(\boldsymbol{x}) = 0$ for $j \in [m]$, an* IPS *proof that the system is unsatisfiable over the algebraic closure of $\mathbb{F}$ is an arithmetic circuit $C(x_1, \ldots, x_n, y_1, \ldots, y_m)$ satisfying*

(i) $C(x_1, \ldots, x_n, 0, \ldots, 0) = 0$
(ii) $C(x_1, \ldots, x_n, f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) = 1$.

In conditions (i) and (ii) on an IPS proof, we require the polynomials to be equal as formal polynomials. It is not sufficient that the polynomial always evaluates to 0 or 1 – the polynomials must contain precisely the same monomials. Checking equality as formal polynomials is the polynomial identity testing problem (PIT), and two instances of this are required to verify that a circuit C is an IPS proof. PIT is not known to be solvable in deterministic polynomial time, and so IPS is not a proof system in the formal sense of Definition 2.3. However, there are known randomised polynomial time algorithms for PIT [103, 111], and derandomised polynomial time algorithms are expected to exist.

As with PC and PCR, IPS can be used as a refutational proof system for propositional formulas. Given a propositional formula C, the equivalent polynomial $t(C)$ is defined inductively as

- $t(x) = x$ for any variable x,
- $t(\neg A) = 1 - t(A)$ for any propositional formula A,
- $t(A \vee B) = t(A) \cdot t(B)$ for propositional formulas A, B,
- $t(A \wedge B) = t(\neg(\neg A \vee \neg B))$ for propositional formulas A, B.

As a propositional proof system, an IPS proof of the unsatisfiability of $\bigwedge_i C_i$ is an IPS refutation of the axioms $t(C_i)$ for each formula C_i, alongside the Boolean axioms $x^2 - x$ for any variable x to ensure only Boolean solutions are permitted.

As with PC and PCR, true is represented as 0, and false as 1. Observe that since we are considering polynomials as arithmetic circuits rather than sums of monomials, there is no need to replace $1 - x$ with a new variable, as in PCR.

A line-based IPS proof system To extend IPS to a QBF proof system, we wish to add the ∀-reduction rule into this system. In order to do so, we must first consider IPS as a line-based proof

system, rather than the static system we have used so far. We therefore define the system *line*-IPS (L-IPS) to achieve this.

Definition 8.28. *Fix a field $\mathbb{F}$. A line-IPS refutation contains lines of the form $p = q$ where p and q are arithmetic circuits computing polynomials in $\mathbb{F}[\boldsymbol{x}]$. The derivation rules of* L-IPS *are given in Figure 19. A* L-IPS *refutation is a derivation of the line $1 = 0$.*

Polynomial Axiom:	$\dfrac{}{f_j(\boldsymbol{x}) = 0}$	$f_j(\boldsymbol{x})$ any axiom polynomial
Input Axiom:	$\dfrac{}{x = x}$ $\quad$ $\dfrac{}{1 = 1}$	for any variable x
Addition:	$\dfrac{p_1 = q_1 \qquad p_2 = q_2}{\alpha p_1 + \beta p_2 = \alpha q_1 + \beta q_2}$	for any $\alpha, \beta \in \mathbb{F}$
Multiplication:	$\dfrac{p_1 = q_1 \qquad p_2 = q_2}{p_1 \cdot p_2 = q_1 \cdot q_2}$	
Rewrite:	$\dfrac{p = q}{p' = q'}$	if $p \equiv p'$ and $q \equiv q'$ as formal polynomials

Fig. 19. The derivation rules of line-IPS

It is clear that there is a correspondence between IPS and L-IPS proofs, by mapping addition and multiplication gates in an IPS proof to addition and multiplication steps in a L-IPS proof, and vice versa. We formalise this in the following lemma, proving the equivalence of these two formulations of IPS.

Lemma 8.29. IPS *and* L-IPS *are p-equivalent.*

Proof. Given a gate g in an arithmetic circuit C, define C_g to be the arithmetic circuit defined by restricting C to a circuit with root at g.

Given an IPS refutation $C(\boldsymbol{x}, y_1, \ldots, y_m)$ of a set of polynomials $f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})$, construct a L-IPS refutation by deducing for each gate g in C the line $C_g(\boldsymbol{x}, f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) = C_g(\boldsymbol{x}, 0, \ldots, 0)$ as follows:

- Suppose g is an input gate. If C_g computes a variable x or a variable y_i, then the L-IPS refutation can deduce the necessary equation by introducing the axiom $x = x$ or $f_i(\boldsymbol{x}) = 0$ respectively. If C_g computes a constant α, then $\alpha = \alpha$ can be derived using the axiom $1 = 1$ and an addition step.
- If g is an addition or a multiplication gate, with inputs g_1 and g_2, then either $C_g = \alpha C_{g_1} + \beta C_{g_2}$ or $C_g = C_{g_1} \cdot C_{g_2}$. The line $C_g(\boldsymbol{x}, f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) = C_g(\boldsymbol{x}, 0, \ldots, 0)$ can be deduced from

the corresponding lines for C_{g_1} and C_{g_2} by an addition or multiplication step, followed by a rewriting step if necessary.

Since the final line computes $C(\boldsymbol{x}, f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) = C(\boldsymbol{x}, 0, \ldots, 0)$, which must be equivalent to $1 = 0$, this is a sound L-IPS refutation. Each line of the L-IPS proof has size at most $2|C| \cdot \max_i |f_i|$, and the number of lines is linear in $|C|$, and hence the constructed proof has size polynomial in $|C|$.

Conversely, given a L-IPS proof π, construct a circuit C_π by assigning to each line L in π a gate g_L. If the line L is derived as an axiom $x = x$, $f_i(\boldsymbol{x}) = 0$ or $1 = 1$, let g_L be the input gate x, y_i or 1 respectively. If L is derived using an addition or multiplication step from L_1 and L_2, then g_L is an addition or multiplication gate respectively, with inputs g_{L_1} and g_{L_2}. If L is a rewriting step from L', identify the gates $g_L = g_{L'}$.

By the construction of C_π, for any line L of the form $p = q$, $C_{g_L}(\boldsymbol{x}, f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})) \equiv p$ and $C_{g_L}(\boldsymbol{x}, 0, \ldots, 0) \equiv q$. Since the final line of π is $1 = 0$, this ensures that the circuit C_π is indeed an IPS refutation. Moreover, the number of gates in C_π is at most the number of lines in π, and so certainly $|C_\pi| \leq |\pi|$. □

These p-simulations also naturally extend to show an equivalence between formula-IPS, in which the circuit C is a formula, and tree-like L-IPS. In the case of tree-like L-IPS, we require the polynomials in each line to be represented as arithmetic formulas, and therefore can limit use of the rewriting step to a single instance at the root of the proof. Viewing IPS as a line based proof system, rather than the static proof system presented in [64], allows us to define the proof system IPS+∀red, by adding the ∀-reduction rule to L-IPS.

Definition 8.30. *Fix a field $\mathbb{F}$. For a false QBF $\Phi = \Pi \cdot \bigwedge_{i=1}^{n} C_i$, let the set of polynomials $F = \{p(C_i) \mid i \in [n]\} \cup \{x^2 - x \mid x \in \mathrm{var}(\Phi)\}$. An* IPS+∀red *refutation of Φ is a deduction of $1 = 0$ from F using the rules of* L-IPS*, or the ∀-reduction rule*

$$\frac{p = q}{p[u/b] = q[u/b]}$$

where p and q contain no variables to the right of u in Π, $b \in \{0, 1\}$ and $p[u/b]$ is the circuit constructed by replacing all instances of the input gate u by the constant b in p.

It is clear that IPS+∀red also requires the use of PIT to check a proof in polynomial time, since the rewriting rule is still allowable. Observe that if a polynomial p computed by a circuit C contains no monomials with a variable to the right of u, then there is a circuit C' also computing p with $|C'| = |C|$, which contains no input gates with variables to the right of u, by replacing all such input gates with constants. The ∀-reduction rule can therefore be applied whenever the *polynomials* contain no instances of variables right of u, even if the circuits computing them do, by preceding the ∀-reduction with a rewriting step. Similarly to other P+∀red proof systems, it is sound to allow the ∀-reduction rule to substitute any arithmetic circuit b in the variables left of u, provided $b[\alpha] \in \{0, 1\}$ for any $\alpha \in \{0, 1\}^n$.

It is also clear that IPS+∀red is sound, as we can show inductively that each equation derived must hold given the axioms $p(C_i) = 0$ and $x^2 - x = 0$, and the quantifier prefix Π. To prove the completeness of IPS+∀red, it suffices to show that IPS+∀red simulates some other complete QBF proof system. It is straightforward to see that IPS+∀red p-simulates QU-Res, translating clauses and simulating resolution steps similarly to PCR+∀red, albeit with the variable $\bar{x}$ replaced by $1 - x$. Since IPS+∀red works with arithmetic circuits, this causes only a linear-size increase in size compared to PCR+∀red.

However, IPS+∀red is strictly stronger than both QU-Res and PCR+∀red. We show this, and also show that IPS+∀red admits large capacity proofs, by exhibiting short proofs of the equality formulas $\mathrm{EQ}(n)$.

Theorem 8.31. *There are polynomial-size* IPS+∀red *refutations of the equality formulas* $\mathrm{EQ}(n)$.

Proof. It is clear that for any polynomial p, there is a derivation of $p = p$ of size polynomial in the size of p. Given a line $q = 0$, there is therefore a polynomial-size derivation of $p \cdot q = 0$.

For each $k \in [n]$, we have the axioms $(1-x_k)(1-u_k)(1-t_k) = 0$ and $x_k u_k(1-t_k) = 0$, from which we can deduce $(1 - x_k - u_k + 2x_k u_k)(1 - t_k) = 0$ by an addition step and a rewriting step. Using the axioms $x_k^2 - x_k = 0$ and $u_k^2 - u_k = 0$, we can therefore deduce $(1-(x_k-u_k)^2)(1-t_k) = 0$ in constant size for each $k \in [n]$, and hence the line $(1 - (x_k - u_k)^2)(1 - t_k) \cdot \prod_{i=1}^{k-1} t_i = 0$ is derivable in polynomial size.

Given the line $\prod_{i=1}^{k} t_i \prod_{i=k+1}^{n}(1 - (x_i - u_i)^2) = 0$ for any $k \in [n]$, there is a polynomial size derivation of $\prod_{i=1}^{k-1} t_i \prod_{i=k}^{n}(1 - (x_i - u_i)^2) = 0$ by adding $(1 - (x_k - u_k)^2)(1 - t_k) \cdot \prod_{i=1}^{k-1} t_i = 0$. In the case $k = n$, the line $\prod_{i=1}^{n} t_i$ is an axiom as it is the translation of the clause $\bigvee_{i=1}^{n} t_i$. There is therefore a polynomial size derivation of the $k = 0$ case $\prod_{i=1}^{n}(1 - (x_i - u_i)^2) = 0$.

The ∀-reductions $u_n/0$ and $u_n/1$ give the lines $(1 - x_n^2)\prod_{i=1}^{n-1}(1 - (x_i - u_i)^2) = 0$ and $(2x_n - x_n^2)\prod_{i=1}^{n-1}(1 - (x_i - u_i)^2) = 0$. By adding these lines together, and using the axiom $x_n^2 - x_n = 0$, we can efficiently derive $\prod_{i=1}^{n-1}(1 - (x_i - u_i)^2) = 0$. A sequence of similar derivations gives $\prod_{i=1}^{k}(1 - (x_i - u_i)^2) = 0$ for each $n \geq k \geq 0$. In the case $k = 0$, we have derived $1 = 0$ and we are done. □

This short IPS+∀red refutation of $\mathrm{EQ}(n)$ demonstrates that IPS+∀red proofs can have large capacity. Indeed, we can see this directly by observing that the refutation described above contains the line $\prod_{i=1}^{n}(1 - (x_i - u_i)^2) = 0$. The only falsifying assignment to the variables u_i is to play $u_i = x_i$ for each $i \in [n]$, and so the capacity of this line is 2^n.

Simulating Frege systems That IPS+∀red proofs can have large capacity is to be expected, as when IPS was introduced as an algebraic proof system, it was shown that IPS p-simulates eFrege. The equivalence of IPS and L-IPS (Lemma 8.29) show that L-IPS also p-simulates eFrege.

A perhaps more interesting IPS-based proof system is non-commutative formula-IPS, in which multiplication of variables is not assumed to be commutative and the circuit is a tree. The proof of the p-equivalence of IPS and L-IPS in Lemma 8.29 naturally extends to show that non-commutative

formula-IPS is p-equivalent to non-commutative tree-like L-IPS. Since polynomial identity testing for non-commutative formulas is possible in deterministic polynomial time [98], non-commutative IPS and non-commutative tree-like L-IPS are therefore checkable in polynomial time, and hence are proof systems in the formal sense of Cook and Reckhow.

Furthermore, while no superpolynomial lower bounds are known for the size of general arithmetic formulas, exponential lower bounds have been shown for non-commutative formulas computing certain polynomials, specifically the determinant and permanent polynomials [89]. While these lower bounds are not themselves sufficient to prove non-commutative formula-IPS lower bounds, as we require a lower bound on *all* polynomials which constitute a proof, they nonetheless represent significant progress towards such lower bounds.

Lower bounds on non-commutative formula-IPS would nonetheless be desirable, as a p-simulation of Frege by non-commutative formula-IPS was proved in [80]. In order to ensure the completeness of non-commutative IPS, the equations $xy - yx = 0$ are included as axioms for any variables x, y; without these, there would be no refutation of $xy - yx + 1 = 0$, for example.

Theorem 8.32 (Li, Tzameret and Wang [80]). *Let $\mathbb{F}$ be either $\mathbb{Q}$ or $\mathbb{Z}_q$ for some prime q. Then non-commutative formula-*IPS *over $\mathbb{F}$ p-simulates* Frege.

The proof of the simulation given in [80] almost immediately extends to the QBF case. In this case, we allow $\forall$-reduction by any formula, rather than only 0/1, to ensure the simulation of dag-like Frege +$\forall$red, since the lower bound of Theorem 7.6 also applies to tree-like IPS+$\forall$red systems with 0/1 $\forall$-reduction.

Corollary 8.33. *Let $\mathbb{F}$ be either $\mathbb{Q}$ or $\mathbb{Z}_q$ for some prime q. Then non-commutative formula-*IPS+$\forall$red *over $\mathbb{F}$ p-simulates* Frege +$\forall$red.

Proof. We allow $\forall$-reduction by any suitable formula, rather than only 0 or 1, and therefore may assume that all Frege and Frege +$\forall$red derivations are tree-like, since they are equivalent to their dag-like versions [30, 75], albeit with a possible slight increase in depth. The proof of Theorem 8.32 shows that if there is a tree-like Frege derivation of the line C from $C_1, \ldots, C_m$, then there is a non-commutative formula p such that $p(\boldsymbol{x}, t(C_1), \ldots, t(C_m)) = t(C)$, and $p(\boldsymbol{x}, 0, \ldots, 0) = 0$. Observing the simulation of non-commutative formula-IPS by non-commutative tree-like L-IPS, we see that this immediately provides a polynomial-size non-commutative tree-like L-IPS derivation of $t(C) = 0$ from the lines $t(C_1) = 0, \ldots, t(C_m) = 0$.

We can therefore construct a non-commutative tree-like L-IPS proof by deriving the line $t(C) = 0$ for each line C in a Frege +$\forall$red proof. We need only show that if C is derived by a $\forall$-reduction step, i.e. $C = C'[u/B]$ for some suitable formulas C and B, then $t(C) = 0$ can be derived from $t(C') = 0$ efficiently. Since C and B are non-commutative formulas, it is easy to verify that $t(C')[u/t(B)] = t(C'[u/B])$, and hence this is a simple $\forall$-reduction step in non-commutative tree-like L-IPS. □

Strategy extraction We conclude our discussion of IPS+∀red by considering strategy extraction techniques and consequent lower bounds. We first observe that in the case of tree-like IPS+∀red, the lower bound of Theorem 7.6 via strategy size still applies when ∀-reduction is limited to only the constants 0/1.

Corollary 8.34. *The equality formulas* $\mathrm{EQ}(n)$ *require proofs of size* $2^{\Omega(n)}$ *in tree-like* IPS+∀red *with 0/1* ∀*-reduction.*

Despite the high capacity of dag-like IPS+∀red proof systems preventing us from obtaining lower bounds for IPS+∀red systems via cost lower bounds, these systems still admit a strategy extraction algorithm via decision lists, similar to that of Theorem 3.5. Since IPS+∀red works with arithmetic circuits, the extracted strategy is also represented as an arithmetic circuit.

Theorem 8.35. *Let* $\mathbb{F}$ *be a field of characteristic* $q > 0$*. Given a QBF* Φ *and an* IPS+∀red *refutation* π *of* Φ*, there exist polynomials over* $\mathbb{F}$ *computing a winning strategy for the universal variables, and these polynomials have circuits of size* $|\pi|^{O(1)}$*. If* π *is a tree-like* IPS+∀red *refutation, then the polynomials have formulas of size polynomial in* $|\pi|, q$*.*

Proof. We follow the model of strategy extraction for P+∀red proof systems introduced in [19], by first constructing for each variable a decision list which computes a winning strategy. We then show that we can give a polynomial-size arithmetic circuit which computes the same function as this decision list.

Recall that in the case of $\mathcal{C}$-Frege +∀red systems, the decision lists were constructed by considering each line of the proof in order, and for each ∀-reduction $\dfrac{C}{C[u/b]}$, the line

$$\text{if } \neg C[u/b] \text{ then } u \leftarrow b, \text{ else...}$$

was added to the end of the decision list.

Given an IPS+∀red proof, we construct decision lists D_u for each universal variable u in the same way. Let $L_1, \ldots, L_m$ be the sequence of IPS+∀red lines in π. For each $1 \leq i \leq m$, if L_i is derived by a ∀-reduction step on a universal variable u, i.e. L_i is the line $p[u/b](\boldsymbol{x}) = q[u/b](\boldsymbol{x})$ for some polynomial $b(\boldsymbol{x})$, then add the line

$$\text{if } p[u/b](\boldsymbol{x}) - q[u/b](\boldsymbol{x}) \neq 0 \text{ then } u \leftarrow b(\boldsymbol{x}), \text{ else...}$$

to the decision list D_u. It is clear from this construction that $|D_u| = O(|\pi|)$ for each universal variable u. The proof that the decision lists D_u compute a winning universal strategy is identical to the proof of the correctness of this algorithm in [19].

It remains to show that we can construct an arithmetic circuit computing a polynomial which is equal to the function defined by the decision list. For a universal variable u, suppose that the decision list D_u has m lines, and let $p_i(\boldsymbol{x})$ and $b_i(\boldsymbol{x})$ be arithmetic circuits such that the ith line of

the decision list for u is

$$\text{if } p_i(\boldsymbol{x}) \neq 0 \text{ then } u \leftarrow b_i(\boldsymbol{x}), \text{ else...}$$

Define the polynomial

$$Q(\boldsymbol{x}) = \sum_{i=1}^{m} \left(p_i(\boldsymbol{x})^{q-1} \cdot b_i(\boldsymbol{x}) \cdot \prod_{j=1}^{i-1} \left(1 - p_j(\boldsymbol{x})^{q-1}\right) \right)$$

Since $\mathbb{F}$ has characteristic q, $p_i(\boldsymbol{x})^{q-1} = 1$ whenever $p_i(\boldsymbol{x}) \neq 0$. For each $1 \leq i \leq m$, the summand is either equal to $b_i(\boldsymbol{x})$ or 0, and in particular will only return $b_i(\boldsymbol{x})$ if $p_i(\boldsymbol{x}) \neq 0$ and for all $j < i$, $p_j(\boldsymbol{x}) = 0$. As a result, the value of $Q(\boldsymbol{x})$ is identical to the value of D_u for any assignment to the variables $\boldsymbol{x}$.

To construct a circuit computing $Q(\boldsymbol{x})$, each polynomial $p_i(\boldsymbol{x})^{q-1}$ and each $b_i(\boldsymbol{x})$ need only be computed once. These can all be computed using $O(|D_u| \log q)$ gates, since the polynomials $p_i(\boldsymbol{x})$ and $b_i(\boldsymbol{x})$ appear in D_u. Clearly $m < |D_u|$, so we require $O(|D_u|)$ additional gates to compute the products and the sum, hence there is an arithmetic circuit computing $Q(\boldsymbol{x})$ with size $O(|D_u| \log q) = O(|\pi| \log q)$. For any fixed field $\mathbb{F}$, and hence fixed q, this gives a circuit of size $O(|\pi|)$.

In the case of tree-like IPS+∀red, the polynomials $p_i(\boldsymbol{x})$ and $b_i(\boldsymbol{x})$ are arithmetic formulas. To compute the ith term in the sum, the polynomial $p_j(\boldsymbol{x})$ must be computed at most $q-1$ times for each $j < i$. The total size of the formulas computing the p_j is less than $|D_u|$, and so each summand requires a formula of size $O(|D_u|q)$. Since there are $m < |D_u|$ summands, there is a formula computing $Q(\boldsymbol{x})$ of size $O(|D_u|^2 q) = O(|\pi|^2 q)$. □

Strategy extraction of this form has been used to transfer lower bounds from Boolean circuit complexity to QBF proof complexity, particularly in the case of $\mathcal{C}$-Frege +∀red proof systems such as $\mathbf{AC}^0[p]$-Frege +∀red. Theorem 8.35 suggests a similar approach for proving lower bounds for IPS+∀red via lower bounds on arithmetic circuits or formulas. It is important to note that the lower bounds required in this case are *functional* lower bounds [56, 63], i.e. lower bounds on the size of circuits describing any polynomial which computes a winning universal strategy on the domain $\langle \mathcal{X} \rangle$, rather than a lower bound on a specific polynomial, as is often considered in arithmetic circuit complexity.

In the case of non-commutative tree-like IPS+∀red, the lines of the proof consist of non-commutative formulas, and so it is clear that the strategy computed in Theorem 8.35 is also a non-commutative formula. Observe that in the simulation of Frege +∀red by non-commutative tree-like IPS+∀red, ∀-reduction steps only occur on lines which are of the form $t(C) = 0$ for some Boolean formula C. In particular, each of the polynomials $p_i(\boldsymbol{x})$ computed in the decision list D_u evaluates to 0 or 1 under every Boolean assignment to the variables $\boldsymbol{x}$. This allows us to remove the restriction on the characteristic of the fields, with the formula computing the winning strategy now

expressible as

$$Q(\boldsymbol{x}) = \sum_{i=1}^{m} \left(p_i(\boldsymbol{x}) \cdot b_i(\boldsymbol{x}) \cdot \prod_{j=1}^{i-1} (1 - p_j(\boldsymbol{x})) \right)$$

regardless of the characteristic of the field $\mathbb{F}$.

This provides a potential route to proving lower bounds for Frege +∀red via functional lower bounds on non-commutative formulas, in keeping with the close connections between proof complexity and arithmetic circuit complexity [94]. For any sequence of Boolean functions $(f_n) \in \mathbf{P/poly}$, we can construct false QBFs Φ_n such that the only winning universal strategy is to play according to f_n [19]. A superpolynomial functional lower bound on (non-commutative) arithmetic formulas computing polynomials p_n over $\mathbb{Q}$ or $\mathbb{Z}_q$ such that $p_n(\alpha) = f_n(\alpha)$ for any $\alpha \in \{0,1\}^n$ then immediately provides a superpolynomial lower bound on Frege +∀red proofs of Φ_n.

No such functional lower bounds are currently known. However, functional lower bounds have been shown for certain restricted classes of arithmetic circuits, such as for low depth homogeneous circuits, both over finite fields [63] and over any field [56]. Functional circuit lower bounds have even previously been used to show proof complexity lower bounds in substantially more restricted versions of IPS [57]. Lower bounds have also been shown for the computation of specific polynomials using non-commutative formulas [89] and even the stronger model of read-k oblivious algebraic branching programs [3]. We therefore suggest that this represents a promising approach for obtaining superpolynomial lower bounds for Frege +∀red.

Chapter 9

Lower bounds on Randomly Generated QBFs

Chapter 8 introduced the notion of the cost of a QBF, and demonstrated that cost is an effective method for proving lower bounds on the size of refutations in several different QBF proof systems. However, cost lower bounds were only shown for a few specific formulas. While the results of Section 8.4 provide an example of a more general application of cost, it is desirable to show that many formulas with high cost exist, and to be able to construct such formulas easily.

To construct formulas with large cost, we introduce product formulas, of which the equality formulas $\mathrm{EQ}(n)$ are an example, as a method of combining false QBFs. These formulas are similar to the formulas $\Phi \oplus \Psi$ (Definition 4.7) introduced in Chapter 4, in that the clauses of the product formula $\Phi \otimes \Psi$ consist of disjunctions of a clause from Φ and a clause from Ψ. However, by choosing to quantify the variables from Ψ before those of Φ and interleaving the quantifier prefixes of the copies of Ψ, we obtain a large cost, since any winning strategy must be winning for all copies of Ψ simultaneously. If the QBFs Ψ have small sets of assignments witnessing that Ψ has non-constant cost, we can leverage this to give a lower bound on $\mathrm{cost}(\Phi \otimes \Psi)$. The cost lower bound of the equality formulas $\mathrm{EQ}(n)$ can be deduced in this way, but the technique allows for the construction of many such families of QBFs.

To demonstrate that this method of obtaining QBFs with large cost is able to generate a large number of such QBFs, we show that it can be applied when picking the formulas Ψ at random from a family of QBFs. This family is based on the family of random (1,2)-QCNFs, a model of random QBFs about which some results are already known [36, 45]. In this way, we can generate a random QBF, with the product formula structure, which with probability $1 - o(1)$ requires large proofs in QU-Res and other proof systems for which cost provides a lower bound. Lower bounds on Resolution proofs of random propositional 3-CNFs are known [38], but to the best of our knowledge, this is the first proof complexity lower bound on random QBFs which does not arise due to these propositional lower bounds.

In Section 9.1, we introduce the product formulas and show lower bounds on their cost. Section 9.2 defines the random QBFs, and shows the proof complexity lower bound via Size-Cost-Capacity and the cost lower bound from Section 9.1.

9.1 Cost lower bounds for product formulas

We begin by defining the product formulas $\Phi \otimes (\Psi_i)$. As with the formulas $\Phi \oplus \Psi$ (Definition 4.7), we essentially replace each clause C_i of Φ with $C_i \vee \Psi_i$. In the case of product formulas, we also wish to allow using different formulas Ψ_i for different clauses C_i of Φ. The primary difference between $\Phi \otimes (\Psi_i)$ and $\Phi \oplus \Psi$, however, is that the variables of Ψ_i are quantified *left* of those of Φ.

Definition 9.1. *Let $\Phi = \exists X_1 \forall X_2 \exists X_3 \dots \exists X_k \cdot \bigwedge_{i=1}^{m} C_i$ be a QBF, and for each $1 \leq i \leq m$, let $\Psi_i = \exists Y_1^i \forall Y_2^i \exists Y_3^i \dots \exists Y_k^i \cdot \bigwedge_{j=1}^{n_i} D_j^i$ be QBFs, where the sets X_i and Y_j^i are (potentially empty) pairwise disjoint sets of variables.*

Define the product formula $\Phi \otimes (\Psi_i)$ to be

$$\exists Y_1^1 \dots Y_1^m \forall Y_2^1 \dots Y_2^m \dots \exists Y_k^1 \dots Y_k^m \exists X_1 \forall X_2 \dots \exists X_k \cdot \bigwedge_{i=1}^{m} \bigwedge_{j=1}^{n_i} (C_i \vee D_j^i).$$

In the case that each Ψ_i is a variable disjoint copy of a single QBF Ψ, we denote this product as $\Phi \otimes \Psi$.

We have already seen an example of such a product formula, namely the equality formulas, which are the product of two very simple QBFs.

Example 9.2. Let $\Phi_n = \exists t_1 \dots t_n \cdot \bigwedge_{i=1}^{n} (\neg t_i) \wedge \bigvee_{i=1}^{n} t_i$. For each $1 \leq i \leq n$, define $\Psi_i = \exists x_i \forall u_i \cdot (x_i \vee u_i) \wedge (\neg x_i \vee \neg u_i)$, and let Ψ_{n+1} be the trivially false QBF. The product formula $\Phi_n \otimes (\Psi_i)$ is precisely the nth equality formula $\mathrm{EQ}(n)$ (Definition 8.2).

The only difference between the construction of $\Phi \otimes \Psi$ and $\Phi \oplus \Psi$ is the order in which the variables of Φ and the Ψ_i are quantified. However, as in the case of $\Phi \oplus \Psi$, the variables of the Ψ_i are pairwise disjoint and distinct from those of Φ, so it is straightforward to see that the product of false QBFs is a false QBF.

Lemma 9.3. *If Φ is a false QBF, and for each QBF Ψ_i, Ψ_i is false, then $\Phi \otimes (\Psi_i)$ is false.*

Proof. We show that the universal player has a winning strategy on $\Phi \otimes (\Psi_i)$ by playing according to winning strategies for Φ and for each Ψ_i.

Since each QBF Ψ_i is false, there exist winning universal strategies σ_i for each Ψ_i. If the universal player plays the universal variables in each Y_j^i according to σ_i for all $1 \leq i \leq m$, then for each $1 \leq i \leq m$, some clause D_j^i of Ψ_i will be falsified. After the kth round, when all variables in the Ψ_i have been assigned, the restricted QBF is therefore $\exists X_1 \forall X_2 \dots \exists X_k \cdot \bigwedge_{i=1}^{m} C_i$, since each clause D_j^i now evaluates to either $\top$ or $\bot$, and for each i, at least one such clause evaluates to $\bot$.

This restricted QBF is Φ, and so the universal player can win by playing according to a winning strategy for Φ. □

The choice to quantify the variables of the QBFs Ψ_i before, rather than after, the variables of Φ has a large effect on the size of proofs of product formulas. In the case of $\Phi \oplus \Psi$, we saw that the size of P+$\forall$red refutations is polynomial in the size of refutations of Φ and Ψ (Lemma 4.9). Considering the equality formulas as an instance of a product formula, it is clear that each of the individual components has short refutations in QU-Res, yet QU-Res refutations of $\mathrm{EQ}(n)$ require exponential size (Corollary 8.11).

As noted in Chapter 8, the lower bound for the equality formulas arises as a result of a lower bound on cost. We give a general method for constructing QBFs with large cost using product formulas. These products can be of relatively simple formulas, the only requirement is that we can find existential assignments for which the universal player must play distinct responses.

Theorem 9.4. *Suppose that $\Phi = \Pi \cdot \bigwedge_{i=1}^{n} C_i$ is a minimally unsatisfiable false QBF and that $\Psi = \exists X_1 \forall U_1 \ldots \exists X_k \forall U_k \cdot \psi$ is a false QBF. If there exist $\alpha, \beta \in \langle \mathcal{X} \rangle$ and a block U_j such that for any winning universal strategy $S : \langle \mathcal{X} \rangle \to \langle \mathcal{U} \rangle$ for Ψ, $S(\alpha)|_{U_j} \neq S(\beta)|_{U_j}$, then $\mathrm{cost}(\Phi \otimes \Psi) \geq 2^n$.*

Proof. Let α_i and β_i be the existential assignments to the variables of Ψ_i, the ith copy of Ψ in $\Phi \otimes \Psi$. We consider the 2^n different existential assignments in $A = \{\bigcup_{i=1}^{n} \gamma_i \mid \gamma_i \in \{\alpha_i, \beta_i\}\}$, constructed by picking either α_i or β_i on the existential variables of Ψ_i.

Let S be any universal winning strategy on $\Phi \otimes \Psi$, and let S_j be the restriction of S to $\bigcup_{i=1}^{n} U_j^i$. We claim that for two distinct assignments $\sigma, \tau \in A$, $S_j(\sigma) \neq S_j(\tau)$. This suffices to show that $|\,\mathsf{rng}(S_j)| \geq 2^n$ and so $\mathrm{cost}(\Phi \otimes \Psi) \geq 2^n$.

Suppose that there are $\sigma \neq \tau$ such that $S_j(\sigma) = S_j(\tau)$. Since $\sigma \neq \tau$, there is some $i' \in [n]$ such that without loss of generality, $\sigma|_{X^{i'}} = \alpha_{i'}$ and $\tau|_{X^{i'}} = \beta_{i'}$, i.e. σ and τ correspond to different choices of α or β on the variables of $\Psi_{i'}$. By the choice of $\alpha_{i'}$ and $\beta_{i'}$, the response $S_j(\sigma)|_{U_j^{i'}} = S_j(\tau)|_{U_j^{i'}}$ cannot be a winning universal response on $\Psi_{i'}$ to both $\alpha_{i'}$ and $\beta_{i'}$, else we could construct a winning universal strategy for Ψ which returns the same response on U_j to both α and β.

Without loss of generality, assume that $S_j(\sigma)|_{U_j^{i'}}$ is not a winning response to $\alpha_{i'}$ on $\Psi_{i'}$. The existential player can therefore win against the strategy S by playing according to σ on all variables not in $\Psi_{i'}$. On the variables of $\Psi_{i'}$, the existential player plays according to σ on variables left of $U_j^{i'}$, and then play a winning strategy on $\Psi_{i'}$ thereafter. As a result, after all variables in the Ψ_i are assigned, the resulting matrix is $\bigwedge_{i \neq i'} C_i$. Since Φ is minimally false, the existential player has a winning strategy on this matrix. This contradicts the choice of S as a winning universal strategy, and hence $S_j(\sigma) \neq S_j(\tau)$ whenever $\sigma \neq \tau$. □

Perhaps the simplest example of an application of Theorem 9.4 is to prove the cost lower bound on the equality formulas (Lemma 8.3). In the case of $\mathrm{EQ}(n)$, $\Psi = \exists x \forall u \cdot (x \vee u) \wedge (\neg x \vee \neg u)$. The assignments $\alpha(x) = 0$ and $\beta(x) = 1$ then require differing responses on u, and so Theorem 9.4 gives $\mathrm{cost}(\mathrm{EQ}(n)) \geq 2^n$.

In the proof of Theorem 9.4, we do not explicitly require that all the QBFs Ψ_i are distinct copies of a single QBF Ψ. It would suffice to find α and β requiring different responses in the jth universal block of each individual Ψ_i for some fixed j. Indeed, with some suitable interleaving of the quantifier prefixes of the Ψ_i, even the requirement of a fixed j could be relaxed. However, we present the lower bound in this form since this provides a natural way to combine two false QBFs to construct a false QBF with high cost.

Observe that the condition of having two assignments in $\langle \mathcal{X} \rangle$ to which no winning universal strategy responds in the same way is a stronger condition than requiring that $\mathrm{cost}(\Psi) \geq 2$. For an example, consider the false QBF $\Psi = \exists x_1 x_2 \forall u_1 u_2 \cdot (x_1 \leftrightarrow u_1) \wedge (x_2 \leftrightarrow u_2)$. It is clear that $\mathrm{cost}(\Psi) = 2$, since any given universal response will not be winning on the matching existential assignment. However, for any two assignments $\alpha, \beta \in \langle \{x_1, x_2\} \rangle$, the response $u_1 = \neg\alpha(x_1), u_2 = \neg\beta(x_2)$ is a winning response to both α and β.

Since the requirement in Theorem 9.4 to find existential assignments with no response in common is more restrictive than simply having non-constant winning strategies, we now prove a more general result. Instead of requiring assignments with no winning response in common, we consider the size of a set of assignments W witnessing the size of $\mathrm{cost}(\Psi)$, i.e. any winning universal strategy must have at least $\mathrm{cost}(\Psi)$ different responses on the assignments in W. If the size of such a set is sufficiently small in comparison to $\mathrm{cost}(\Psi)$, we can show that the product formulas have large cost.

Theorem 9.5. *Let $\Phi = \Pi \cdot \bigwedge_{i=1}^{n} C_i$ be a minimally unsatisfiable QBF. For each $i \in [n]$, let $\Psi_i = \exists X_i \forall U_i \cdot \psi_i$ be a false QBF, and let $W_i \subseteq \langle X_i \rangle$ be such that for any winning universal strategy S for Ψ_i, $|S(W_i)| \geq c_i$. Then*

$$\mathrm{cost}(\Phi \otimes (\Psi_i)) \geq \prod_{i=1}^{n} \left(1 + \frac{c_i - 1}{|W_i| - (c_i - 1)}\right).$$

The statement of Theorem 9.5 is only for products with QBFs with a Σ_2^b-prefix. We present it in this form for simplicity, and because this is sufficient for our later cost lower bound on products of random formulas (Theorem 9.15). The proof lifts relatively straightforwardly to products with Ψ_i of the form $\exists X_i \forall U_i \Pi_i \cdot \psi_i$, where witnessing sets $W_i \subseteq \langle X_i \rangle$ requiring at least c_i different responses in $\langle U_i \rangle$ still provide a cost lower bound of $\prod_{i=1}^{n} \left(1 + \frac{c_i - 1}{|W_i| - (c_i - 1)}\right)$.

If the block requiring a large number of responses is not the leftmost universal block, a similar result could be obtained with a more careful consideration of winning strategies for $\Phi \otimes (\Psi_i)$. Theorem 9.4 can be viewed as a special case of this more general form of Theorem 9.5, in which each of the Ψ_i is identical with $c_i = 2$. The witnessing set $W = \{\alpha, \beta\}$ ensures that $|W_i| = 2$ for each $i \in [n]$, and the cost lower bound of 2^n follows.

To prove Theorem 9.5, we need the following lemma, which shows that given a witnessing set W for which at least k different responses are required, for any response $\beta \in \langle U \rangle$ we can find $k - 1$ assignments in W for which β is not a winning response.

Lemma 9.6. *Let $\Psi = \exists X \forall U \cdot \psi$ be a false QBF with $\mathrm{cost}(\Psi) \geq k$. Let $W \subseteq \langle X \rangle$ be such that for any universal winning strategy $S : \langle X \rangle \to \langle U \rangle$ for Ψ, $|\{S(\alpha) \mid \alpha \in W\}| \geq k$. For any $\beta \in \langle U \rangle$, define $W_\beta := \{\alpha \in W \mid \psi[\alpha][\beta] = \top\}$ to be the set of assignments in W for which β is not a winning response. Then $|W_\beta| \geq k - 1$ for all $\beta \in \langle U \rangle$.*

Proof. Since Ψ is false, let $S : \langle X \rangle \to \langle U \rangle$ be a winning universal strategy. Define the strategy $S' : \langle X \rangle \to \langle U \rangle$ by

$$S'(\alpha) = \begin{cases} \beta & \text{if } \alpha \in W \backslash W_\beta \\ S(\alpha) & \text{otherwise} \end{cases}$$

The strategy S' is a winning universal strategy, since $S(\alpha)$ is a winning universal strategy, and for any $\alpha \in W \backslash W_\beta$, $\psi[\alpha][\beta] = \bot$ by the definition of W_β. However,

$$\{S'(\alpha) \mid \alpha \in W\} = \{S'(\alpha) \mid \alpha \in W_\beta\} \cup \{\beta\}$$

and so $|\{S'(\alpha) \mid \alpha \in W\}| \leq |W_\beta| + 1$. Since S' is a winning universal strategy, we have $|\{S'(\alpha) \mid \alpha \in W\}| \geq k$ and hence $|W_\beta| \geq k - 1$. □

We can then use this lower bound on $|W_\beta|$ to prove Theorem 9.5. For any assignment $\beta_i \in \langle U_i \rangle$, we have seen that we can find $c_i - 1$ assignments in W_i for which β_i is not winning. We use this to show that for any universal response β on the first block of $\Phi \otimes (\Psi_i)$, we can construct many existential assignments from the W_i for which β is not a winning response. Any winning universal strategy therefore requires many responses to cover all possible existential assignments.

Proof (of Theorem 9.5). Let $X = \bigcup_{i=1}^n X_i$ and $U = \bigcup_{i=1}^n U_i$ be the first existential and universal blocks of $\Phi \otimes (\Psi_i)$ respectively. Define $A = \{\bigcup_{i \in [n]} \alpha_i \mid \alpha_i \in W_i\} \subseteq \langle X \rangle$ to be the set of assignments to the leftmost block of $\Phi \otimes (\Psi_i)$ constructed by picking one assignment from each of the witnessing sets W_i. Observe that $|A| = \prod_{i=1}^n |W_i|$.

Fix any winning universal strategy S for $\Phi \otimes (\Psi_i)$, and let $S_U : \langle X \rangle \to \langle U \rangle$ be its restriction to the first universal block. Let $B = S_U(A) \subseteq \mathsf{rng}(S_U)$ be the responses on U given by S to the assignments in A. Showing a lower bound on $|B|$ therefore provides a lower bound on $\mathrm{cost}(S)$, and hence on $\mathrm{cost}(\Phi \otimes (\Psi_i))$.

For each $\beta \in \langle U \rangle$, let $A_\beta := \{\alpha \in A \mid \Phi \otimes (\Psi_i)[\alpha][\beta] \equiv \bot\} \subseteq A$ be the assignments in A for which β is a winning response to α. Since Φ is minimally unsatisfiable, in order for an assignment $\beta \in \langle U \rangle$ to be a winning response to $\alpha \in \langle X \rangle$, we must have that $\beta|_{U_i}$ is a winning response to $\alpha|_{X_i}$ on Ψ_i for every $i \in [n]$. For each $\alpha \in A$, $\alpha|_{X_i} \in W_i$, and by Lemma 9.6, there are at most $|W_i| - (c_i - 1)$ assignments in W_i for which $\beta|_{U_i}$ is a winning response. We therefore conclude that $|A_\beta| \leq \prod_{i=1}^n (|W_i| - (c_i - 1))$ for every $\beta \in \langle U \rangle$.

However, since S is a winning universal strategy, $(\Phi \otimes (\Psi_i))[\alpha][S_u(\alpha)]$ is false, and hence $\alpha \in A_{S_U(\alpha)}$ for every $\alpha \in A$. Moreover, for each $\alpha \in A$, $S_u(\alpha) \in B$, and so $A = \bigcup_{\alpha \in A} A_{S_U(\alpha)} = \bigcup_{\beta \in B} A_\beta$, hence $|A| \leq \sum_{\beta \in B} |A_\beta|$. Applying the bounds for the sizes of A and A_β we have

established, we conclude

$$\prod_{i=1}^{n} |W_i| \leq |B| \cdot \prod_{i=1}^{n} (|W_i| - (c_i - 1))$$

and hence

$$|B| \leq \prod_{i=1}^{n} \frac{|W_i|}{|W_i| - (c_i - 1)} = \prod_{i=1}^{n} \left(1 + \frac{c_i - 1}{|W_i| - (c_i - 1)}\right).$$

The cost lower bound follows immediately as this lower bound on $|B|$ does not depend on the winning strategy S. □

Theorem 9.5 gives a simple way to construct QBFs with high cost. By way of example, consider the equality formulas (Definition 8.2) which are of the form $\Phi \otimes (\Psi_i)$, where $\Psi_i = \exists x_i \forall u_i \cdot (x_i \vee u_i) \wedge (\neg x_i \vee \neg u_i)$. By taking $W_i = \langle \{x_i\} \rangle$, we have $|W_i| = c_i = 2$ for all $i \in [n]$ and hence $\mathrm{cost}(\mathrm{EQ}(n)) \geq (1 + \frac{2-1}{2-(2-1)})^n = 2^n$.

To ensure high cost, we need only find a sequence of QBFs Ψ_n with sets W_n of assignments to the first existential block witnessing that $\mathrm{cost}(\Psi_n) \geq c_n > 1$ such that $|W_n|$ does not grow too fast compared with c_n. In particular, for an exponential lower bound on cost, a witnessing set W_n such that $\frac{|W_n|}{c_n} = O(n^{1-\epsilon})$ will suffice.

Corollary 9.7. *Let $\Phi_n := \Pi \cdot \bigwedge_{i=1}^{n} C_i$ be a sequence of minimally unsatisfiable QBFs. Let $\Psi_n = \exists X_n \forall U_n \cdot \psi_n$ be a sequence of false QBFs, with a set $W_n \subseteq \langle X_n \rangle$ such that for any winning universal strategy S for Ψ_n, $|S(W_n)| \geq c_n \geq 2$. If $\frac{|W_n|}{c_n} = O(n^{1-\epsilon})$ for some $\epsilon > 0$, then* $\mathrm{cost}(\Phi_n \otimes \Psi_n) = 2^{\Omega(n^\epsilon)}$.

Proof. We apply Theorem 9.5 to obtain

$$\mathrm{cost}(\Phi_n \otimes \Psi_n) = \left(1 + \frac{c_n - 1}{|W_n| - (c_n - 1)}\right)^n = \left(1 + \frac{1}{\frac{|W_n|}{c_n - 1} - 1}\right)^n.$$

Since $c_n \geq 2$, $\frac{|W_n|}{c_n - 1} \leq 2\frac{|W_n|}{c_n} = O(n^{1-\epsilon})$ and hence for sufficiently large n, $\frac{|W_n|}{c_n - 1} \leq kn^{1-\epsilon}$ for some constant k. Substituting this into the bound above, we have

$$\mathrm{cost}(\Phi_n \otimes \Psi_n) \geq \left(1 + \frac{1}{kn^{1-\epsilon} - 1}\right)^n \geq 2^{\frac{n}{kn^{1-\epsilon}-1}} = 2^{\Omega(n^\epsilon)}$$

since $(1 + \frac{1}{m})^m \geq 2$ for all $m \geq 1$. □

Corollary 9.7 allows the construction of QBFs requiring exponential-size proofs in QU-Res, CP+$\forall$red and PCR+$\forall$red from any choice of two false QBFs, providing these QBFs satisfy relatively weak conditions. Moreover, such lower bounds can be given even when Φ and Ψ themselves have short proofs in these proof systems, such as in the case of the equality formulas.

9.2 Lower bounds for products of random formulas

We have shown that we can obtain cost lower bounds via product formulas. The conditions required to achieve these bounds are relatively unrestrictive. Indeed, we now show that selecting the formulas Ψ_i uniformly at random from a suitable class of QBFs will suffice to give a cost lower bound via Theorem 9.5. This represents the first proof size lower bound on a randomly generated QBF.

We begin by defining the class of random formulas we shall consider.

Definition 9.8. *For each $1 \leq i \leq n$, let $C_i^1, \ldots, C_i^{cn}$ be distinct clauses picked uniformly at random from the set of clauses containing 1 literal from the set $X_i = \{x_i^1, \ldots, x_i^m\}$ and 2 literals from $Y_i = \{y_i^1, \ldots, y_i^n\}$. Define the randomly generated QBF $Q(n, m, c)$ as*

$$Q(n,m,c) := \exists Y_1 \ldots Y_n \forall X_1 \ldots X_n \exists t_1 \ldots t_n \cdot \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{cn} (\neg t_i \vee C_i^j) \wedge \bigvee_{i=1}^{n} t_i.$$

Specifying that clauses contain a given number of literals from different sets may seem unusual, especially when compared with random k-SAT instances, where clauses are picked from the set of clauses containing any k literals. However, it is widely used in the study of random QBFs [36, 45]. Indeed, such a specification is necessary, since if any clause in the matrix of a QBF contains only literals on universal variables, then this QBF is immediately false, and all P+$\forall$red proof systems have a constant size refutation using only this clause and a sequence of $\forall$-reduction steps. Specifying that all clauses must contain a given number of literals from different sets of variables avoids this issue by ensuring that every clause contains existential variables. It is natural that we also expect clauses in a QBF to contain universal variables – it would be unsatisfying to have a random QBF which is false because it contains some unsatisfiable propositional instance.

The randomly generated formula $Q(n, m, c)$ builds on this idea by constructing formulas Ψ_i by choosing clauses uniformly at random from all clauses containing one universal variable and two existential variables, in a similar way to that in which random 3-SAT chooses clauses at random from all clauses containing 3 literals. We can then view $Q(n, m, c)$ as a product formula by defining $\Phi = \exists t_1 \ldots t_n \cdot \bigwedge_{i=1}^{n} \neg t_i \wedge \bigvee_{i=1}^{n} t_i$, and $\Psi_i = \exists Y_i \forall X_i \cdot \bigwedge_{j=1}^{cn} C_i^j$; $Q(n, m, c)$ is then equal to the product formula $\Phi \otimes (\Psi_i)$. Since Φ is fixed and contains no universal variables, we focus our attention on the randomly generated Ψ_i. In order for $Q(n, m, c)$ to be false, it must be the case that all Ψ_i are false. We therefore aim to show that for suitable values of m and c, with high probability all the Ψ_i are false, and furthermore a linear number of the Ψ_i have $\mathrm{cost}(\Psi_i) \geq 2$.

In order to satisfy the matrix of the formula Ψ_i, the existential player must play an assignment which satisfies a literal in each clause C_i^j. If not, the universal player could win by falsifying the universal literal in any such clause where the existential literals are both falsified. Determining the truth of Ψ_i is therefore reduced to the 2-SAT problem defined by the existential parts of the clauses C_i^j. We can then use the following result on the satisfiability of random 2-SAT formulas, shown independently by Chvátal and Reed [37], Goerdt [61] and de la Vega [48], to obtain the falsity of

the Ψ_i. We state it here with a tighter probability lower bound of $1 - o(n^{-1})$ proved by de la Vega in [49], which is necessary for Lemma 9.10.

Theorem 9.9 (de la Vega [49]). *Let Φ be a random 2-SAT formula on n propositional variables containing cn clauses selected uniformly at random with repetition. If $c > 1$ then Φ is unsatisfiable with probability $1 - o\left(n^{-1}\right)$.*

The following lemma is equivalent to the statement that, with the same bound on c, $Q(n, m, c)$ is false with probability $1 - o(1)$. This is a fairly immediate consequence of Theorem 9.9; we need only check that the clauses of Ψ_i contain sufficiently many different existential clauses. The possibility of repeating an existential clause many times with different universal variables makes this non-trivial, but this is still relatively easy to verify.

Lemma 9.10. *If $c > 1$, then with probability $1 - o(1)$, Ψ_i is false for every $1 \leq i \leq n$.*

Proof. For Ψ_i to be false, it is sufficient for the 2-SAT problem constructed by taking only the existential parts of the clause to be unsatisfiable, since the universal player can always falsify the universal literal on any unsatisfied existential clause. In order to use Theorem 9.9, we must show there is some constant $k > 1$ such that, for each $i \in [n]$, the clauses C_i^j contain at least kn distinct existential clauses with high probability.

For each i, there are $4\binom{n}{2}$ possible existential clauses, and $2m$ possible universal literals. The total number of possible clauses C_i^j is therefore $4mn(n-1)$.

Let k be some fixed constant such that $1 < k < c$. To determine the probability of $C_i^1, \ldots, C_i^{cn}$ containing at least kn distinct clauses in the existential variables, we consider making cn random choices of clause from the $4mn(n-1)$ possible such clauses. If fewer than kn distinct existential clauses have been chosen so far, the probability of a randomly chosen clause having existential part distinct from all previous chosen clauses is

$$\frac{4mn(n-1) - 2mkn}{4mn(n-1)} = 1 - \frac{k}{2(n-1)}.$$

We define the selection of a clause to be successful if it either selects a clause with existential part distinct from that of the previous clauses, of if kn distinct existential clauses have already been selected. The probability of any selection being successful is therefore at least $1 - \frac{k}{2(n-1)}$. We make cn selections, and require the probability that at least kn are successful.

Letting Z be the number of successes, Z is a sum of cn Bernoulli random variables with $p = 1 - \frac{k}{2(n-1)}$. Using Hoeffding's inequality [67], we obtain

$$P(Z \leq kn) \leq \exp\left(-2\frac{\left(cn - \frac{cnk}{2(n-1)} - kn\right)^2}{cn}\right) = \exp\left(-\frac{2(c-k)^2}{c}n + O(1)\right)$$

and so $P(Z > kn) = 1 - e^{-\Omega(n)} = 1 - o\left(n^{-1}\right)$.

The probability that Ψ_i is false is at least the probability of it containing at least kn distinct existential clauses and the first kn distinct such clauses being unsatisfiable. Since the clauses $C_i^1, \ldots, C_i^{cn}$ were chosen uniformly at random, each set of kn existential clauses is equally likely to be chosen, so the probability these clauses are unsatisfiable is $1 - o\left(n^{-1}\right)$, by Theorem 9.9. The probability of Ψ_i being false is therefore $P(Z > kn) \cdot \left(1 - o\left(n^{-1}\right)\right) = 1 - o\left(n^{-1}\right)$.

Finally, the clauses in each Ψ_i are independently chosen, so the probability of all Ψ_i being false is $\left(1 - o\left(n^{-1}\right)\right)^n = 1 - o(1)$. □

It remains to show that $\mathrm{cost}(Q(n, m, c))$ is large. As with falsity, we first look at the cost of Ψ_i, and observe that, for $m \leq \log_2(n)$ and $1 < c < 2$, $\mathrm{cost}(\Psi_i) \geq 2$ with probability $1 - o(1)$. Winning responses for $Q(n, m, c)$ are simultaneous winning responses for each of the Ψ_i. Since many of the Ψ_i require multiple distinct responses, it is reasonable to expect that the number of different responses required to be able to falsify all of them is large. With a careful choice of the parameters m and c, we can indeed ensure that $Q(n, m, c)$ has large cost with high probability.

To prove $\mathrm{cost}(\Psi_i) \geq 2$, it is sufficient to show that $\mathrm{cost}(\Psi_i) \neq 1$, i.e. that any universal winning strategy $S : \langle Y_i \rangle \to \langle X_i \rangle$ for Ψ_i is not constant. If there is such a constant winning strategy, say $S(\alpha) = \beta$ for all $\alpha \in \langle Y_i \rangle$, then the response is independent of α and so β also constitutes a winning strategy for $\Psi_i' = \forall X_i \exists Y_i \cdot \bigwedge_{j=1}^{cn} C_i^j$. Showing that Ψ_i' is true is therefore sufficient to show $\mathrm{cost}(\Psi_i) \geq 2$. With this modified prefix, Ψ_i' is of the form of a $(1, 2)$-QCNF, a previously studied model for generating random QBFs.

Definition 9.11 (Chen and Interian [36]). *A $(1, 2)$-QCNF is a QBF of the form $\forall X \exists Y \cdot \phi(X, Y)$ where $X = \{x_1, \ldots, x_m\}$, $Y = \{y_1, \ldots, y_n\}$ and $\phi(X, Y)$ is a 3-CNF formula in which each clause contains one universal literal and two existential literals.*

If such a winning strategy for Ψ_i' exists, then Ψ_i' is false. However, for $c < 2$, Ψ_i' is known to be true with high probability in the case that $m \leq \log_2(n)$.

Theorem 9.12 (Creignou et al. [45]). *Let $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$ be disjoint sets of variables, and let $\Phi = \forall X \exists Y \cdot \phi(X, Y)$ be a $(1, 2)$-QCNF in which $\phi(X, Y)$ contains cn clauses picked uniformly at random from the set of all suitable clauses. If $m \leq \log_2(n)$ and if $c < 2$, then Φ is true with probability $1 - o(1)$.*

Given the lower bound $c > 1$ of Lemma 9.10, and the upper bound $c < 2$ of Theorem 9.12, we pick $1 < c < 2$ to lie between these bounds. With this choice of c, we can combine these results to observe that not only are the Ψ_i all false with high probability, but Ψ_i also requires non-constant winning strategies with high probability.

Lemma 9.13. *Let $\Psi_i = \exists Y_i \forall X_i \cdot \psi_i$ be as above. If $1 < c < 2$ and $m \leq \log_2(n)$, then with probability $1 - o(1)$, Ψ_i is false and $\mathrm{cost}(\Psi_i) \geq 2$.*

Proof. From the proof of Lemma 9.10, we observe that since $c > 1$, Ψ_i is false with probability $1 - o\left(n^{-1}\right)$ and hence $\mathrm{cost}(\Psi_i) \geq 1$.

Suppose now that $\mathrm{cost}(\Psi_i) = 1$. There is therefore some $\beta \in \langle X_i \rangle$ such that β is a winning response for any $\alpha \in \langle Y_i \rangle$. That is, for any $\alpha \in \langle Y_i \rangle$, $\psi_i[\alpha][\beta] = \bot$. We can use β as a winning strategy for $\Psi_i' = \forall X_i \exists Y_i \cdot \psi_i$, defining a winning strategy as $S'(\varnothing) = \beta$. Since $\psi_i[\beta][\alpha] = \bot$ for all $\alpha \in \langle Y_i \rangle$, S' is a winning strategy for the universal player on Ψ', and hence Ψ' is false. However, since $c < 2$, by Theorem 9.12, Ψ' is false with probability $o(1)$ and so such a $\beta \in \langle X_i \rangle$ exists with probability $o(1)$.

The probability that Ψ_i is false and $\mathrm{cost}(\Psi_i) \geq 2$ is therefore $1 - o\left(n^{-1}\right) - o(1) = 1 - o(1)$. □

We use Lemma 9.13 to show that with high probability, a linear number of the Ψ_i require non-constant winning strategies.

Theorem 9.14. *For each $i \in [n]$, let Ψ_i be as defined above. Let $m \leq \log_2(n)$ and c, l be constants such that $1 < c < 2$ and $l < 1$. With high probability at least ln of the Ψ_i have $\mathrm{cost}(\Psi_i) \geq 2$.*

Proof. Since $c > 1$, we have from Lemma 9.10 that with high probability Ψ_i is false for every $i \in [n]$, so $\mathrm{cost}(\Psi_i)$ is defined for every $1 \leq i \leq n$. For any given $i \in [n]$, the probability that $\mathrm{cost}(\Psi_i) \geq 2$ is $1 - o(1)$, by Lemma 9.13.

Again using Hoeffding's bound on the sum of independent Bernoulli random variables, the probability that fewer than ln of the Ψ_i satisfy $\mathrm{cost}(\Psi_i) \geq 2$ is

$$\exp\left(-2(1 - l - o(1))^2 n\right)$$

which for sufficiently large n is upper bounded by $\exp\left(-2(1 - l')^2 n\right)$ for some constant $l' < 1$. Thus with probability $1 - o(1)$, $\mathrm{cost}(\Psi_i) \geq 2$ for at least ln of the Ψ_i. □

In the context of $Q(n, m, c)$, which is equivalent to $\bigvee_{i=1}^{n} \Psi_i$, Theorem 9.14 shows that for suitable values of m and c, not only are all the Ψ_i false with high probability, and hence also $Q(n, m, c)$ is false with high probability, but also a linear proportion of the Ψ_i have $\mathrm{cost}(\Psi_i) \geq 2$. With a slightly more careful choice of m, these properties will suffice to show a cost lower bound on $Q(n, m, c)$.

Unfortunately, as noted previously, we cannot obtain $\mathrm{cost}(Q(n, m, c))$ simply by multiplying the values of $\mathrm{cost}(\Psi_i)$ for each i, as the universal response on X_i may now depend on the assignment to Y_j for some $j \neq i$. Instead, we use the property that if $\mathrm{cost}(\Psi_i) \geq 2$ then for any response $\beta_i \in \langle X_i \rangle$, there is some assignment in $\langle Y_i \rangle$ for which β_i is not a winning response for the universal player in Ψ_i. Using these, we can construct for any universal response β on $Q(n, m, c)$, a large set of assignments in $\langle Y_1, \ldots, Y_n \rangle$ for which β is not a winning response.

Theorem 9.15. *Let $1 < c < 2$ be a constant, and let $m \leq (1 - \epsilon)\log_2(n)$ for some constant $\epsilon > 0$. With probability $1 - o(1)$, $Q(n, m, c)$ is false and $\mathrm{cost}(Q(n, m, c)) = 2^{\Omega(n^\epsilon)}$.*

Proof. The randomly generated QBF $Q(n, m, c)$ can be considered as a product formula $\Phi \otimes (\Psi_i)$, where $\Phi = \exists t_1 \ldots t_n \cdot \bigwedge_{i=1}^{n} \neg t_i \wedge \bigvee_{i=1}^{n} t_i$ and for each $i \in [n]$, $\Psi_i = \exists Y_i \forall X_i \cdot \bigwedge_{j=1}^{cn} C_i^j$ with

$Y_i = \{y_1, \ldots, y_n\}$, $X_i = \{x_1, \ldots, x_m\}$ and the clauses C_i^j chosen uniformly at random from the set of all clauses containing two literals on variables in Y_i and one literal on a variable of X_i. Clearly Φ is false, and by Lemma 9.10, all the Ψ_i are false with probability $1 - o(1)$, and hence the product formula $\Phi \otimes (\Psi_i) = Q(n, m, c)$ is false with probability $1 - o(1)$ (Lemma 9.3).

It remains to show that $\text{cost}(Q(n, m, c)) \geq 2^{\Omega(n^\epsilon)}$. By Theorem 9.14, for any constant $0 < l < 1$, at least ln of the Ψ_i satisfy $\text{cost}(\Psi_i) \geq 2$. We show that this suffices for the cost lower bound on $Q(n, m, c)$. Without loss of generality, assume that $\Psi_1, \ldots, \Psi_{\lceil ln \rceil}$ have $\text{cost}(\Psi_i) \geq 2$, and that $\text{cost}(\Psi_i) = 1$ for $\lceil ln \rceil < i \leq n$.

We aim to apply Theorem 9.5, with $c_i = 2$ for each $1 \leq i \leq \lceil ln \rceil$ and $c_i = 1$ for $\lceil ln \rceil < i \leq n$, to provide our cost lower bound. To do so, we must construct suitable witnessing sets $W_i \subseteq \langle Y_i \rangle$ witnessing that $\text{cost}(\Psi_i) \geq c_i$. For $\lceil ln \rceil < i \leq n$, this is straightforward, since we can let $W_i = \{\alpha\}$ for any $\alpha \in \langle X \rangle$. Any winning strategy S_i for Ψ_i will return a response to α, and so $|S_i(\{\alpha\})| = 1 = c_i$.

If $1 \leq i \leq \lceil ln \rceil$, list the elements of $\langle X_i \rangle$ as $\langle X_i \rangle = \{\beta_1^i, \ldots, \beta_N^i\}$, where $N = 2^m \leq n^{1-\epsilon}$. For each β_j^i, we can find some assignment $\alpha_j^i \in \langle Y_i \rangle$ such that β_j^i is not a winning response to α_j^i for the universal player on Ψ_i. If this were not the case, there would be a constant winning strategy for the universal player on Ψ_i by playing β_j^i. We therefore define $W_i = \{\alpha_j^i \mid j \in [N]\}$. Given a winning universal strategy S_i for Ψ_i, observe that $|S_i(W_i)| \geq 2$, as for any $\beta \in \langle X_i \rangle$, W_i contains an assignment for which β is not a winning response, and so we cannot have $S(W_i) = \{\beta\}$ for any $\beta \in \langle X_i \rangle$.

We can now use Theorem 9.5. For each $1 \leq i \leq \lceil ln \rceil$, we have $c_i = 2$ with $|W_i| \leq N$, and hence $(1 + \frac{c_i - 1}{|W_i| - (c_i - 1)}) \geq (1 + \frac{1}{N})$. For $\lceil ln \rceil < i \leq n$, $c_i = |W_i| = 1$ and $(1 + \frac{c_i - 1}{|W_i| - (c_i - 1)}) = 1$. We therefore have

$$\text{cost}(\Phi \otimes \{\Psi_i\}) \geq \left(1 + \frac{1}{N}\right)^{\lceil ln \rceil} \geq \left(1 + \frac{1}{n^{(1-\epsilon)}}\right)^{ln} = \left(\left(1 + \frac{1}{n^{(1-\epsilon)}}\right)^{n^{(1-\epsilon)}}\right)^{ln^\epsilon} = 2^{\Omega(n^\epsilon)}$$

since $N \geq n^{(1-\epsilon)}$ and for large n, $(1 + \frac{1}{n})^n \geq 2$.

We have shown that if all the Ψ_i are false, and if at least ln of the Ψ_i have no constant winning universal strategies, then $\Phi \otimes (\Psi_i) = Q(n, m, c)$ is false and $\text{cost}(Q(n, m, c)) \geq 2^{\Omega(n^\epsilon)}$. Both of these conditions hold with probability $1 - o(1)$ by Lemma 9.10 and Theorem 9.14, which completes the proof. □

Theorem 9.15 proves that for appropriate values of m and c, the QBF $Q(n, m, c)$ have large cost with high probability. We can then apply the Size-Cost-Capacity theorem, and the capacity upper bounds from Chapter 8 to give proof size lower bounds on $Q(n, m, c)$.

Theorem 9.16. *Let $1 < c < 2$ be a constant, and let $m \leq (1 - \epsilon)\log_2(n)$ for some constant $\epsilon > 0$. With high probability, the randomly generated QBF $Q(n, m, c)$ is false, and any* QU-Res, CP+$\forall$red *or* PCR+$\forall$red *refutation of $Q(n, m, c)$ requires size $2^{\Omega(n^\epsilon)}$.*

Proof. From Theorem 9.15, with high probability $Q(n, m, c)$ is false with $\text{cost}(Q(n, m, c)) \geq 2^{\Omega(n^\epsilon)}$. Theorems 8.10 and 8.16 provide that cost is a lower bound for QU-Res and CP+$\forall$red proofs respectively. By Theorem 8.21, we have that the size of a PCR+$\forall$red proof of $Q(n, m, c)$ is at least $\sqrt{\text{cost}(Q(n, m, c))} \geq 2^{\frac{1}{2}\Omega(n^\epsilon)} = 2^{\Omega(n^\epsilon)}$. □

The underlying structure of the formulas means that the QBF $Q(n, m, c) = \Phi \otimes (\Psi_i)$ may not seem as generic as the model of random 3-CNFs, but nonetheless, to the best of our knowledge this is the first instance of a proof complexity lower bound on a randomly generated class of QBFs. As noted, some additional structure is necessary in any random QBF to ensure that no clause contains purely universal literals. The structure of Φ need not be fixed. Our choice of Φ in $Q(n, m, c)$ was only for the sake of simplicity. The key feature required of Φ for the results of this section is that any refutation requires $\Omega(n)$ clauses.

An analogous lower bound to that in Theorem 9.16 could therefore be applied to a similar class of random formulas of the form $\Phi \otimes (\Psi_i)$ in which the formula Φ is a existentially quantified random 3-CNF. A random 3-CNF with n variables and kn clauses is false with high probability for sufficiently large values of k [58], yet still requires a linear number of clauses in any refutation. However, with high probability such random 3-CNFs require large refutations in Resolution [38], Cutting Planes [55, 68], and in Polynomial Calculus [12], and so while we can show a 'genuine' QU-Res or CP+$\forall$red lower bound on such formulas in this way, in the sense that the formulas would require large proofs in Σ_1^p-QU-Res, Σ_1^p-CP+$\forall$red and Σ_1^p-PCR+$\forall$red via the cost lower bounds above, QU-Res, CP+$\forall$red and PCR+$\forall$red lower bounds also follow immediately from the propositional lower bounds.

Suitably chosen random 2-CNFs are also false with high probability (Theorem 9.9), and all false 2-CNFs have short Resolution refutations. However, the proof of Theorem 9.9 in [49] also shows that, with high probability, such a random 2-CNF contains an unsatisfiable set of clauses of size $O(\log n)$. This is not sufficient to provide a lower bound on $\text{cost}(\Phi \otimes (\Psi_i))$, as a winning strategy for the universal player need only be able to falsify $O(\log n)$ of the Ψ_i.

To show the proof size lower bound for the random formulas, we gave a lower bound on their cost and applied Theorem 8.7. It is therefore natural to ask whether we can find short proofs of the QBFs $Q(n, m, c)$ in stronger proof systems, such as Frege +$\forall$red, for which we do not have capacity upper bounds. This is indeed the case.

Theorem 9.17. *Whenever the QBF $Q(n, m, c)$ is false, there is a polynomial-size* Frege +$\forall$red *refutation of $Q(n, m, c)$.*

Proof. Consider $Q(n, m, c)$ to be the product formula $\Phi \otimes (\Psi_i)$ as previously, where $\Phi = \exists t_1 \dots t_n \cdot \bigwedge_{i=1}^n \neg t_i \wedge \bigvee_{i=1}^n t_i$ and $\Psi_i = \exists Y_i \forall X_i \cdot \psi_i$, with $\psi_i = \bigwedge_{j=1}^{cn} C_i^j$.

From the clauses $\neg t_i \vee C_i^j$, there is a polynomial-size Frege derivation of $\neg t_i \vee \psi_i$, and hence a polynomial-size Frege derivation of $\bigvee_{i=1}^n \psi_i$ from the clauses of $Q(n, m, c)$. Since the variables of each of the ψ_i are disjoint, it therefore suffices to show that for each $i \in [n]$ there is a short

refutation of Ψ_i, since we can then use this refutation to derive $\bigvee_{i=1}^{k-1} \psi_i$ from $\bigvee_{i=1}^{k} \psi_i$, ultimately deriving $\bot$.

Let $\beta_0, \beta_1 \in \langle X_i \rangle$ be the identically 0 and identically 1 assignments respectively to the variables of X_i. Using two $\forall$-reductions, we can obtain $\psi_i[\beta_0]$ and $\psi_i[\beta_1]$, and hence derive $\psi_i[\beta_0] \wedge \psi_i[\beta_1]$. By the construction of the clauses C_i^j, this is a 2-CNF. Moreover, a clause of ψ_i is satisfied by precisely one of β_0 and β_1, and hence $\psi_i[\beta_0] \wedge \psi_i[\beta_1]$ contains the existential parts of every clause of ψ_i. As discussed above, since Ψ_i is false, this is an unsatisfiable 2-CNF. All false 2-CNFs have linear-size refutations in Frege (indeed, even in Resolution); this completes a polynomial-size refutation of Ψ_i. □

That there is always a short Frege +$\forall$red, and indeed even $\mathbf{AC}^0$-Frege +$\forall$red, refutation of $Q(n, m, c)$ serves to emphasise that lower bounds via cost fall into the third case of the trichotomy given in Theorem 6.5. Moreover, the existence of a large number of formulas with large cost represents a significant challenge for QBF solvers, many of which correspond to proof systems admitting lower bounds via cost. While some solvers incorporate elements of long distance resolution [110], our results suggest that working with clauses or linear inequalities rather than more expressive formulas is a major limitation of current solvers.

Chapter 10

Conclusion

In this thesis, we have made contributions to QBF proof complexity in two related areas. The first is an increased understanding of the causes for lower bounds in QBF proof systems, in particular distinguishing those arising for propositional reasons. The second is in introducing two related new lower bound techniques based on round-based strategy extraction.

Understanding lower bounds Dissatisfaction at the presence of propositional lower bounds for QBFs has previously been raised [35]. Some previous progress had been made in identifying such lower bounds, either through a new proof system, or by characterising any non-propositional lower bounds [30]. We have provided a strengthening for both these approaches.

We first described the relaxing QU-Res proof system, which was designed to remove propositional hardness as a cause of lower bounds. By introducing a new method of combining false QBFs which allows precise control over proof size and the reasons for lower bounds, we showed that relaxing QU-Res does not eliminate all propositional lower bounds.

Having observed that relaxing QU-Res does not achieve this distinction, we introduced an alternative proof system for identifying propositional lower bounds in Chapter 5. This takes the form of a proof system with access to an oracle for Σ^{p}_1. Rather than only allowing use of an oracle when deriving axioms, Σ^{p}_1-QU-Res can use the Σ^{p}_1-oracle throughout the proof, but is limited to only Σ^{p}_1-oracles, rather than the higher levels of the polynomial hierarchy permitted in relaxing QU-Res. This allows for more effective modelling of the approaches of QBF solvers which call a SAT solver as a black box, as well as identifying families of formulas which are useful for testing solvers' handling of quantified variables.

The Σ^{p}_1-QU-Res proof system also extends to Σ^{p}_k-QU-Res for larger k, allowing us to study the effect of the alternation of quantifiers on the complexity of proofs. We have given a thorough analysis of the effect of such oracles on the QU-Res proof system. We showed a hierarchy of separations between QU-Res proof systems with access to different levels of the polynomial hierarchy, but that there also exist families of formulas which cannot be efficiently solved in QU-Res with any fixed such oracle. These results emphasise the importance of the dependency schemes used by

solvers such as DepQBF, and associated proof systems [82, 105]. Such systems offer the prospect of overcoming some of the hardness produced by the alternation of quantifiers in the prefix by considering more carefully the dependence of variables on each other beyond the information given by the quantifier prefix.

The second approach to identifying propositional lower bounds in QBF is to understand all possible reasons for hardness, including propositional hardness, and therefore be able to classify QBFs accordingly. In Chapter 6, we refined the characterisation of lower bounds for $\mathcal{C}$-Frege +$\forall$red systems to give a characterisation for any system with strategy extraction by circuits. Since most QBF proof systems admit such strategy extraction, this allows for a substantially greater understanding of QBF lower bounds. In particular, our results extend the classification of lower bounds to proof systems such as QU-Res, which more closely relate to algorithms used in QBF solvers. In contrast to circuit lower bounds, the additional cause of hardness which arises in weaker systems such as QU-Res is harder to encapsulate in a simple idea. However, in the case of tree-like proof systems we completely described lower bounds in this category in Chapter 7.

Lower bound techniques Having identified propositional lower bounds for QBF proof systems via Σ_1^p-P+$\forall$red, we provided two related techniques of a distinctly QBF flavour for proving lower bounds. Both of these techniques are based on the size of a winning strategy for the universal player, specifically the number of different ways the universal player may need to play on the universal variables.

The first of these, presented in Chapter 7, is a lower bound for treelike versions of even very strong QBF proof systems including Frege +$\forall$red and eFrege +$\forall$red. This lower bound is based on the absolute number of responses required by a universal winning strategy. The simplicity of this lower bound ensures that it can be applied to a wide variety of formulas, and demonstrates the importance of clause learning in QCDCL based solvers. Moreover, the lower bounds proved via this technique highlight the distinction between the two different implementations of the $\forall$-reduction rule. This is the first instance in which a difference in power between these approaches has been observed in a P+$\forall$red proof system.

The second lower bound technique we presented seeks to apply a similar lower bound from strategy size to dag-like proof systems. In order to achieve this, we measured the number of responses in a single block of universal variables, rather than across the entire QBF. We used this measure to show lower bounds on various proof systems, including CP+$\forall$red and PCR+$\forall$red. This allowed us to show lower bounds for several systems simultaneously simply by showing a lower bound on the cost of formulas. We also observed that this technique cannot be applied to stronger proof systems such as Frege +$\forall$red or IPS+$\forall$red, suggesting that if solving techniques corresponding to proof systems such as CP+$\forall$red or PCR+$\forall$red can be adapted to work with stronger circuits, this could lead to significant improvements in solving.

Chapter 9 applied this technique based on cost and capacity to give a method for constructing QBFs which are hard for these weaker proof systems. Such a method of constructing has obvious

applications in generating test sets for solving competitions and solver development. The number of hard formulas that can be generated in this way is large, as we showed by presenting a model for randomly generating QBFs which are lower bounds for these proof systems with high probability.

Future work We have shown that in systems weaker than $\mathcal{C}$-Frege +$\forall$red systems, lower bounds fall into three categories rather than two. It would be desirable to be able to give a simpler characterisation of the additional causes of lower bounds in systems such as QU-Res. We provided a lower bound technique which produces lower bounds in this third category, namely Size-Cost-Capacity, but further work is needed to determine if this suffices to prove all such lower bounds.

Superpolynomial lower bounds for stronger P+$\forall$red proof systems cannot easily be shown via cost and capacity since these systems allow proofs with high capacity. Recent work by Beyersdorff and Blinkhorn [16] has shown that a technique similar to cost and capacity, employing an adaptation of cost, can be implemented in the expansion-based system IR-calc. In the case of long-distance calculi such as LD-Q-Res, neither a dichotomy of the form of Theorem 6.2, nor a lower bound technique based on the number of universal responses, is known. The establishment of either would contribute greatly to the understanding the strength of long-distance QBF resolution systems.

References

1. M. AJTAI, *Σ^1_1-formulae on finite structures*, Annals of Pure and Applied Logic, 24 (1983), pp. 1–48.
2. M. ALEKHNOVICH, E. BEN-SASSON, A. A. RAZBOROV, AND A. WIGDERSON, *Space complexity in propositional calculus*, SIAM J. Comput., 31 (2002), pp. 1184–1211.
3. M. ANDERSON, M. A. FORBES, R. SAPTHARISHI, A. SHPILKA, AND B. L. VOLK, *Identity testing and lower bounds for read-*k *oblivious algebraic branching programs*, ACM Trans. Comput. Theory, 10 (2018), pp. 3:1–3:30.
4. C. ANSÓTEGUI, C. P. GOMES, AND B. SELMAN, *The Achilles' heel of QBF*, in National Conference on Artificial Intelligence (AAAI), 2005, pp. 275–281.
5. S. ARORA AND B. BARAK, *Computational Complexity - A Modern Approach*, Cambridge University Press, 2009.
6. A. ATSERIAS, J. K. FICHTE, AND M. THURLEY, *Clause-learning algorithms with many restarts and bounded-width resolution*, J. Artif. Intell. Res., 40 (2011), pp. 353–373.
7. G. AUDEMARD AND L. SIMON, *Predicting learnt clauses quality in modern SAT solvers*, in International Joint Conference on Artificial Intelligence (IJCAI), 2009, pp. 399–404.
8. V. BALABANOV AND J. R. JIANG, *Unified QBF certification and its applications*, Formal Methods in System Design, 41 (2012), pp. 45–65.
9. V. BALABANOV, M. WIDL, AND J. R. JIANG, *QBF resolution systems and their proof complexities*, in Theory and Applications of Satisfiability Testing (SAT), 2014, pp. 154–169.
10. P. BEAME, H. A. KAUTZ, AND A. SABHARWAL, *Towards understanding and harnessing the potential of clause learning*, J. Artif. Intell. Res., 22 (2004), pp. 319–351.
11. P. BEAME AND T. PITASSI, *Simplified and improved resolution lower bounds*, in Foundations of Computer Science (FOCS), 1996, pp. 274–282.
12. E. BEN-SASSON AND R. IMPAGLIAZZO, *Random CNF's are hard for the polynomial calculus*, Computational Complexity, 19 (2010), pp. 501–519.
13. E. BEN-SASSON AND A. WIGDERSON, *Short proofs are narrow - resolution made simple*, J. ACM, 48 (2001), pp. 149–169.
14. M. BENEDETTI AND H. MANGASSARIAN, *QBF-based formal verification: Experience and perspectives*, JSAT, 5 (2008), pp. 133–191.
15. D. L. BERRE AND A. PARRAIN, *The sat4j library, release 2.2*, JSAT, 7 (2010), pp. 59–6.
16. O. BEYERSDORFF AND J. BLINKHORN, *Genuine lower bounds for QBF expansion*, in Symposium on Theoretical Aspects of Computer Science (STACS), 2018, pp. 12:1–12:15.
17. O. BEYERSDORFF, J. BLINKHORN, AND L. HINDE, *Size, cost and capacity: A semantic technique for hard random QBFs*, in Innovations in Theoretical Computer Science (ITCS), 2018, pp. 9:1–9:18.
18. ———, *Size, cost, and capacity: A semantic technique for hard random QBFs*, Logical Methods in Computer Science, 15 (2019).
19. O. BEYERSDORFF, I. BONACINA, AND L. CHEW, *Lower bounds: From circuits to QBF proof systems*, in Innovations in Theoretical Computer Science (ITCS), 2016, pp. 249–260.
20. O. BEYERSDORFF, L. CHEW, AND M. JANOTA, *On unification of QBF resolution-based calculi*, in Mathematical Foundations of Computer Science (MFCS), 2014, pp. 81–93.
21. ———, *Proof complexity of resolution-based QBF calculi*, in Symposium on Theoretical Aspects of Computer Science (STACS), 2015, pp. 76–89.
22. O. BEYERSDORFF, L. CHEW, M. MAHAJAN, AND A. SHUKLA, *Feasible interpolation for QBF resolution calculi*, Logical Methods in Computer Science, 13 (2017).
23. ———, *Are short proofs narrow? QBF resolution is* not *so simple*, ACM Trans. Comput. Logic, 19 (2018), pp. 1:1–1:26.
24. ———, *Understanding cutting planes for QBFs*, Information and Computation, 262 (2018), pp. 141–161.
25. O. BEYERSDORFF, L. CHEW, AND K. SREENIVASAIAH, *A game characterisation of tree-like Q-resolution size*, J. Comput. Syst. Sci., 104 (2019), pp. 82–101.

26. O. BEYERSDORFF, N. GALESI, AND M. LAURIA, *A lower bound for the pigeonhole principle in tree-like resolution by asymmetric prover-delayer games*, Information Processing Letters, 110 (2010), pp. 1074–1077.
27. ———, *A characterization of tree-like resolution size*, Information Processing Letters, 113 (2013), pp. 666–671.
28. O. BEYERSDORFF AND L. HINDE, *Characterising tree-like Frege proofs for QBF*, Information and Computation, - (2019). to appear.
29. O. BEYERSDORFF, L. HINDE, AND J. PICH, *Reasons for hardness in QBF proof systems*, in Foundations of Software Technology and Theoretical Computer Science (FSTTCS), 2017.
30. O. BEYERSDORFF AND J. PICH, *Understanding Gentzen and Frege systems for QBF*, in Logic in Computer Science (LICS), 2016, pp. 146–155.
31. A. BIERE, *Lingeling, plingeling, picosat and precosat at sat race 2010*, tech. rep., FMV Reports Series, Institute for Formal Models and Verification, Johannes Kepler University, Linz, Austria, 2010.
32. A. BIERE, A. CIMATTI, E. M. CLARKE, AND Y. ZHU, *Symbolic model checking without BDDs*, in Tools and Algorithms for Construction and Analysis of Systems (TACAS), 1999, pp. 193–207.
33. A. BIERE AND A. FRÖHLICH, *Evaluating CDCL variable scoring schemes*, in Theory and Applications of Satisfiability Testing (SAT), 2015, pp. 405–422.
34. A. BLAKE, *Canonical expressions in boolean algebra*, PhD thesis, University of Chicago, 1937.
35. H. CHEN, *Proof complexity modulo the polynomial hierarchy: Understanding alternation as a source of hardness*, ACM Trans. Comput. Theory, 9 (2017), pp. 15:1–15:20.
36. H. CHEN AND Y. INTERIAN, *A model for generating random quantified Boolean formulas*, in International Joint Conference on Artificial Intelligence (IJCAI), 2005, pp. 66–71.
37. V. CHVÁTAL AND B. A. REED, *Mick gets some (the odds are on his side)*, in Foundations of Computer Science (FOCS), 1992, pp. 620–627.
38. V. CHVÁTAL AND E. SZEMERÉDI, *Many hard examples for resolution*, J. ACM, 35 (1988), pp. 759–768.
39. E. M. CLARKE, O. GRUMBERG, S. JHA, Y. LU, AND H. VEITH, *Counterexample-guided abstraction refinement for symbolic model checking*, J. ACM, 50 (2003), pp. 752–794.
40. M. CLEGG, J. EDMONDS, AND R. IMPAGLIAZZO, *Using the Groebner basis algorithm to find proofs of unsatisfiability*, in ACM Symposium on the Theory of Computing (STOC), 1996, pp. 174–183.
41. J. CLYMO AND O. BEYERSDORFF, *Relating size and width in variants of Q-resolution*, Information Processing Letters, 138 (2018), pp. 1–6.
42. S. A. COOK, *The complexity of theorem-proving procedures*, in ACM Symposium on the Theory of Computing (STOC), 1971, pp. 151–158.
43. S. A. COOK AND R. A. RECKHOW, *The relative efficiency of propositional proof systems*, Journal of Symbolic Logic, 44 (1979), pp. 36–50.
44. W. J. COOK, C. R. COULLARD, AND G. TURÁN, *On the complexity of cutting-plane proofs*, Discrete Applied Mathematics, 18 (1987), pp. 25–38.
45. N. CREIGNOU, H. DAUDÉ, U. EGLY, AND R. ROSSIGNOL, *Exact location of the phase transition for random (1, 2)-QSAT*, RAIRO - Theor. Inf. and Applic., 49 (2015), pp. 23–45.
46. M. DAVIS, G. LOGEMANN, AND D. W. LOVELAND, *A machine program for theorem-proving*, Commun. ACM, 5 (1962), pp. 394–397.
47. M. DAVIS AND H. PUTNAM, *A computing procedure for quantification theory*, J. ACM, 7 (1960), pp. 201–215.
48. W. F. DE LA VEGA, *On random 2-SAT (revised version)*. preprint, 1998.
49. ———, *Random 2-SAT: results and problems*, Theoretical Computer Science, 265 (2001), pp. 131–146.
50. DIPTARAMA, R. YOSHINAKA, AND A. SHINOHARA, *QBF encoding of generalized Tic-Tac-Toe*, in Workshop on Quantified Boolean Formulas (QBF), at Theory and Applications of Satisfiability Testing (SAT), 2016, pp. 14–26.
51. N. EÉN AND N. SÖRENSSON, *An extensible SAT-solver*, in Theory and Applications of Satisfiability Testing (SAT), 2003, pp. 502–518.
52. U. EGLY, *On stronger calculi for QBFs*, in Theory and Applications of Satisfiability Testing (SAT), 2016, pp. 419–434.

53. U. EGLY, F. LONSING, AND M. WIDL, *Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving*, in Logic for Programming, Artificial Intelligence, and Reasoning (LPAR), 2013, pp. 291–308.
54. Y. FILMUS, M. LAURIA, M. MIKSA, J. NORDSTRÖM, AND M. VINYALS, *Towards an understanding of polynomial calculus: New separations and lower bounds - (extended abstract)*, in International Colloquium on Automata, Languages, and Programming (ICALP), 2013, pp. 437–448.
55. N. FLEMING, D. PANKRATOV, T. PITASSI, AND R. ROBERE, *Random Θ(log n)-CNFs are hard for cutting planes*, in Foundations of Computer Science (FOCS), 2017, pp. 109–120.
56. M. A. FORBES, M. KUMAR, AND R. SAPTHARISHI, *Functional lower bounds for arithmetic circuits and connections to Boolean circuit complexity*, in Conference on Computational Complexity (CCC), 2016, pp. 33:1–33:19.
57. M. A. FORBES, A. SHPILKA, I. TZAMERET, AND A. WIGDERSON, *Proof complexity lower bounds from algebraic circuit complexity*, in Conference on Computational Complexity (CCC), 2016, pp. 32:1–32:17.
58. J. FRANCO AND M. C. PAULL, *Probabilistic analysis of the Davis Putnam procedure for solving the satisfiability problem*, Discrete Applied Mathematics, 5 (1983), pp. 77–87.
59. M. L. FURST, J. B. SAXE, AND M. SIPSER, *Parity, circuits, and the polynomial-time hierarchy*, Mathematical Systems Theory, 17 (1984), pp. 13–27.
60. I. P. GENT AND A. G. D. ROWLEY, *Encoding connect-4 using quantified Boolean formulae*, International Workshop on Modelling and Reformulating CSPs, at Principles and Practice of Constraint Programming (CP), (2003), pp. 78–93.
61. A. GOERDT, *A threshold for unsatisfiability*, J. Comput. Syst. Sci., 53 (1996), pp. 469–486.
62. A. GOULTIAEVA, A. V. GELDER, AND F. BACCHUS, *A uniform approach for generating proofs and strategies for both true and false QBF formulas*, in International Joint Conference on Artificial Intelligence (IJCAI), 2011, pp. 546–553.
63. D. GRIGORIEV AND A. A. RAZBOROV, *Exponential lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields*, Appl. Algebra Eng. Commun. Comput., 10 (2000), pp. 465–487.
64. J. A. GROCHOW AND T. PITASSI, *Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system*, J. ACM, 65 (2018), pp. 37:1–37:59.
65. A. HAKEN, *The intractability of resolution*, Theoretical Computer Science, 39 (1985), pp. 297–308.
66. J. HÅSTAD, *Almost optimal lower bounds for small depth circuits*, in ACM Symposium on the Theory of Computing (STOC), 1986, pp. 6–20.
67. W. HOEFFDING, *Probability inequalities for sums of bounded random variables*, Journal of the American Statistical Association, 58 (1963), pp. 13–30.
68. P. HRUBES AND P. PUDLÁK, *Random formulas, monotone circuits, and interpolation*, in Foundations of Computer Science (FOCS), 2017, pp. 121–131.
69. M. JANOTA, W. KLIEBER, J. MARQUES-SILVA, AND E. M. CLARKE, *Solving QBF with counterexample guided refinement*, Artificial Intelligence, 234 (2016), pp. 1–25.
70. M. JANOTA AND J. MARQUES-SILVA, *Expansion-based QBF solving versus Q-resolution*, Theoretical Computer Science, 577 (2015), pp. 25–42.
71. H. A. KAUTZ AND B. SELMAN, *Planning as satisfiability*, in European Conference on Artificial Intelligence (ECAI), 1992, pp. 359–363.
72. H. KLEINE BÜNING, M. KARPINSKI, AND A. FLÖGEL, *Resolution for quantified Boolean formulas*, Information and Computation, 117 (1995), pp. 12–18.
73. W. KLIEBER, S. SAPRA, S. GAO, AND E. M. CLARKE, *A non-prenex, non-clausal QBF solver with game-state learning*, in Theory and Applications of Satisfiability Testing (SAT), 2010, pp. 128–142.
74. R. KONTCHAKOV, F. WOLTER, AND M. ZAKHARYASCHEV, *Logic-based ontology comparison and module extraction, with an application to DL-Lite*, Artificial Intelligence, 174 (2010), pp. 1093–1141.
75. J. KRAJÍČEK, *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, vol. 60 of Encyclopedia of Mathematics and Its Applications, Cambridge University Press, Cambridge, 1995.

76. ———, *Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic*, Journal of Symbolic Logic, 62 (1997), pp. 457–486.
77. ———, *Proof Complexity*, vol. 170 of Encyclopedia of Mathematics and Its Applications, Cambridge University Press, Cambridge, 2019.
78. J. KRAJÍČEK AND P. PUDLÁK, *Some consequences of cryptographical conjectures for S_2^1 and EF*, Information and Computation, 140 (1998), pp. 82–94.
79. J. KRAJÍČEK, P. PUDLÁK, AND A. R. WOODS, *An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle*, Random Structures and Algorithms, 7 (1995), pp. 15–40.
80. F. LI, I. TZAMERET, AND Z. WANG, *Characterizing propositional proofs as noncommutative formulas*, SIAM J. Comput., 47 (2018), pp. 1424–1462.
81. J. H. LIANG, C. OH, V. GANESH, K. CZARNECKI, AND P. POUPART, *MapleCOMSPS, MapleCOMSPS LRB, MapleCOMSPS CHB*, in SAT Competition, 2016, p. 52.
82. F. LONSING AND A. BIERE, *DepQBF: A dependency-aware QBF solver*, JSAT, 7 (2010), pp. 71–76.
83. J. ŁUKASIEWICZ, *Elements of Mathematical Logic*, Macmillan, New York, 1963.
84. M. LUO, C. LI, F. XIAO, F. MANYÀ, AND Z. LÜ, *An effective learnt clause minimization approach for CDCL SAT solvers*, in International Joint Conference on Artificial Intelligence (IJCAI), 2017, pp. 703–711.
85. S. MALIK AND L. ZHANG, *Boolean satisfiability from theoretical hardness to practical success*, Commun. ACM, 52 (2009), pp. 76–82.
86. J. P. MARQUES SILVA, I. LYNCE, AND S. MALIK, *Conflict-driven clause learning SAT solvers*, in Handbook of Satisfiability, 2009, pp. 131–153.
87. J. P. MARQUES SILVA AND K. A. SAKALLAH, *GRASP: A search algorithm for propositional satisfiability*, IEEE Transactions on Computers, 48 (1999), pp. 506–521.
88. M. W. MOSKEWICZ, C. F. MADIGAN, Y. ZHAO, L. ZHANG, AND S. MALIK, *Chaff: Engineering an efficient SAT solver*, in Design Automation Conference (DAC), 2001, pp. 530–535.
89. N. NISAN, *Lower bounds for non-commutative computation*, in ACM Symposium on the Theory of Computing (STOC), 1991, pp. 410–418.
90. J. NORDSTRÖM, *On the interplay between proof complexity and SAT solving*, ACM SIGLOG News, 2 (2015), pp. 19–44.
91. C. H. PAPADIMITRIOU, *Computational complexity*, Addison-Wesley, 1994.
92. K. PIPATSRISAWAT AND A. DARWICHE, *On the power of clause-learning SAT solvers as resolution engines*, Artificial Intelligence, 175 (2011), pp. 512–525.
93. T. PITASSI, P. BEAME, AND R. IMPAGLIAZZO, *Exponential lower bounds for the pigeonhole principle*, Computational Complexity, 3 (1993), pp. 97–140.
94. T. PITASSI AND I. TZAMERET, *Algebraic proof complexity: progress, frontiers and challenges*, ACM SIGLOG News, 3 (2016), pp. 21–43.
95. P. PUDLÁK, *Lower bounds for resolution and cutting plane proofs and monotone computations*, Journal of Symbolic Logic, 62 (1997), pp. 981–998.
96. P. PUDLÁK AND R. IMPAGLIAZZO, *A lower bound for DLL algorithms for* k*-SAT*, in Symposium on Discrete Algorithms (SODA), 2000, pp. 128–136.
97. M. N. RABE AND L. TENTRUP, *CAQE: A certifying QBF solver*, in Formal Methods in Computer-Aided Design (FMCAD), 2015, pp. 136–143.
98. R. RAZ AND A. SHPILKA, *Deterministic polynomial identity testing in non-commutative models*, Computational Complexity, 14 (2005), pp. 1–19.
99. R. A. RECKHOW, *On the lengths of proofs in the propositional calculus*, PhD thesis, University of Toronto, 1976.
100. J. RINTANEN, *Asymptotically optimal encodings of conformant planning in QBF*, in AAAI Conference on Artificial Intelligence, 2007, pp. 1045–1050.
101. J. A. ROBINSON, *A machine-oriented logic based on the resolution principle*, J. ACM, 12 (1965), pp. 23–41.
102. H. SAMULOWITZ AND F. BACCHUS, *Using SAT in QBF*, in Principles and Practice of Constraint Programming (CP), 2005, pp. 578–592.

103. J. T. SCHWARTZ, *Fast probabilistic algorithms for verification of polynomial identities*, J. ACM, 27 (1980), pp. 701–717.
104. N. SEGERLIND, *The complexity of propositional proofs*, Bulletin of Symbolic Logic, 13 (2007), pp. 417–481.
105. F. SLIVOVSKY AND S. SZEIDER, *Quantifier reordering for QBF*, J. Autom. Reasoning, 56 (2016), pp. 459–477.
106. S. STABER AND R. BLOEM, *Fault localization and correction with QBF*, in Theory and Applications of Satisfiability Testing (SAT), 2007, pp. 355–368.
107. L. J. STOCKMEYER AND A. R. MEYER, *Word problems requiring exponential time: Preliminary report*, in ACM Symposium on the Theory of Computing (STOC), 1973, pp. 1–9.
108. G. S. TSEITIN, *On the complexity of derivation in propositional calculus*, Zapiski Nauchnykh Seminarov POMI, 8 (1968), pp. 234–259.
109. A. VAN GELDER, *Contributions to the theory of practical quantified Boolean formula solving*, in Principles and Practice of Constraint Programming (CP), 2012, pp. 647–663.
110. L. ZHANG AND S. MALIK, *Conflict driven learning in a quantified Boolean satisfiability solver*, in International Conference on Computer-aided Design (ICCAD), 2002, pp. 442–449.
111. R. ZIPPEL, *Probabilistic algorithms for sparse polynomials*, in Symbolic and Algebraic Computation, EUROSAM, 1979, pp. 216–226.

Index