Characterising tree-like Frege Proofs for QBF

Olaf Beyersdorff[1a], Luke Hinde[b]

[a]Institute of Computer Science, Friedrich Schiller University Jena, Germany
[b]School of Computing, University of Leeds, United Kingdom

Abstract

We examine the tree-like versions of QBF Frege and extended Frege systems. While in the propositional setting, tree-like and dag-like Frege systems are equivalent, we show that this is not the case for QBF Frege, where tree-like systems are exponentially weaker. This applies to the version of QBF Frege where the universal reduction rule substitutes universal variables by 0/1 constants.

To show lower bounds for tree-like QBF Frege we devise a general technique that provides lower bounds for all tree-like QBF systems of the form $\mathsf{P}+\forall\mathsf{red}$, where P is a propositional system. The lower bound is based on the semantic measure of *strategy size* corresponding to the size of countermodels for false QBFs.

We also obtain a full characterisation of hardness for tree-like QBF Frege. Lower bounds for this system either arise from a lower bound to propositional Frege, from a circuit lower bound, or from a lower bound to strategy size.

Keywords: proof complexity, QBF, Frege systems, lower bounds

1. Introduction

The primary goal of proof complexity is to show upper and lower bounds on the sizes of proofs of tautologies in different proof systems, and thus to be able to compare the relative strengths of these proof systems. The close ties between several of these proof systems and modern SAT and QBF solvers [1], such as the connection between Resolution and CDCL-based solvers,

[1]Corresponding author. Postal address: Friedrich Schiller University of Jena, Institute of Computer Science, Ernst-Abbe-Platz 2, 07743 Jena, Germany. Email: `olaf.beyersdorff@uni-jena.de`

ensure that such results can be leveraged to provide a better understanding of solving techniques.

Of particular interest to proof complexity are new techniques for showing lower bounds on the proofs of formulas, such as the relations between size and width for propositional resolution [2], and in the case of quantified Boolean formulas (QBFs), lifting circuit lower bounds via strategy extraction [3, 4], and the Size-Cost-Capacity theorem [5]. Such techniques not only provide the clear benefit of proving several lower bounds on proof systems, but also suggest families of formulas which ought to be hard instances for solvers, and therefore provide suitable benchmarks for the testing and improvement of such solvers.

The focal point of this paper are Frege systems. In the propositional setting these are very strong proof systems [6], based on axiom schemes and rules such as modus ponens. While Frege systems operate with Boolean formulas as lines, the extended Frege system EF works with Boolean circuits [7]. Showing lower bounds on Frege or even extended Frege systems constitutes a major open problem in proof complexity [8].

A common method for extending a propositional proof system P, for the SAT problem, to a QBF proof system is the addition of the universal reduction rule $\forall\mathsf{red}$, resulting in the QBF system $\mathsf{P}+\forall\mathsf{red}$ [9, 10, 4]. This construction also gives rise to QBF Frege and extended Frege systems [4, 11]. The $\forall\mathsf{red}$ rule allows the substitution of universal variables under certain restrictions, either by constants 0/1, or by some suitably expressed Boolean function. The $\forall\mathsf{red}$ rule is generally used in the form allowing only substitution by constants, since in many of the most commonly studied proof systems, such as $\mathsf{QU\text{-}Res}$ or dag-like $\mathsf{Frege}+\forall\mathsf{red}$, the two versions are equivalent [10, 11], and 0/1 substitution models solving techniques such as QDPLL and QCDCL [12].

The round-based strategy extraction algorithm defined in [13] has been used to construct lower bounds for $\mathsf{P}+\forall\mathsf{red}$ proof systems based on the *cost* of a formula [5], a semantic measure which counts how many responses are needed in one block of universal quantifiers. Here we consider *strategy size*, a more general notion than cost which looks at the responses across all universal blocks. Strategy size was first introduced in [14] where it was shown to provide lower bounds to the expansion QBF system $\forall\mathsf{Exp}+\mathsf{Res}$ [15]. However, strategy size is not sufficient to obtain lower bounds in QBF systems of the form $\mathsf{P}+\forall\mathsf{red}$.

In this paper we combine strategy size with a careful analysis of the round-based strategy extraction algorithm of [13] in order to lower bound the number of paths in a proof from the root to an axiom. In particular, this gives an immediate lower bound on any *tree-like* P+∀red proof system.

Having proved this lower bound technique, we obtain a characterisation of tree-like Frege+∀red and EF+∀red lower bounds. In [11] a dichotomy is shown for Frege+∀red (and respectively EF+∀red): hardness either arises from a circuit lower bound for NC^1 (resp. P/poly) or a propositional lower bound for Frege (resp. EF). Hence EF+∀red combines the hardest problems for circuit and proof complexity.

Here we extend this dichotomy to a characterisation of tree-like Frege+∀red and EF+∀red lower bounds. This characterisation demonstrates that all lower bounds on tree-like Frege+∀red and tree-like EF+∀red which do not arise from a lower bound on the corresponding dag-like system are a result of a lower bound on strategy size.

This result provides a trichotomy for hardness in tree-like Frege+∀red and EF+∀red and also exactly identifies those formulas which provide separations between the tree-like and dag-like versions. This is quite in contrast to the propositional scenario, where it is known that tree-like and dag-like Frege are equivalent (and similarly for EF) [16]. The separations between tree-like Frege+∀red and dag-like Frege+∀red crucially rely on the fact that the universal reduction rule only allows to substitute constants 0/1 for universal variables. If substitution by arbitrary formulas (or circuits in case of EF+∀red) is allowed, then again the equivalence of the tree-like and dag-like systems hold [11]. Furthermore, these versions are equivalent to the dag-like Frege+∀red systems with 0/1 reduction considered here [11]. Hence there are essentially two different versions of Frege+∀red: the tree-like 0/1-reduction version and the dag-like version (with either 0/1 or formula reduction).

2. Preliminaries

For a set of variables X, we use the notation $\langle X \rangle$ to refer to the set of Boolean assignments from X to $\{0, 1\}$. For clarity, for an assignment α on variables $x_1, \ldots, x_n$, we denote by α^i the restriction of α to the variables $x_1, \ldots, x_i$.

In the context of proof systems considered here, a *line* with variables in X is associated with a function $\langle X \rangle \to \{0, 1\}$. The set of variables which appear in a line L is denoted by $\text{vars}(L) \subseteq X$. Lines are often expressed as a Boolean circuit from a specified circuit class, but can also be in other forms such as a linear inequality or a polynomial equality. The restriction of a line L by a partial assignment α to a subset of the variables of L is denoted by $L[\alpha]$.

2.1. Quantified Boolean Formulas

A *quantified Boolean formula (QBF)*, often denoted $\Phi = \Pi \cdot \phi$, consists of a quantifier prefix $\Pi = \exists x_1 \forall u_1 \exists x_2 \forall u_2 \dots \forall u_n \exists x_{n+1}$, with quantifiers ranging over $\{0, 1\}$, and a propositional matrix $\phi = \phi(x_1, u_1, \dots, x_n, u_n, x_{n+1})$ containing only variables quantified in Π. The matrix ϕ is often expressed in conjunctive normal form (CNF); in the present work we assume all QBFs to be such *QCNFs*. It will be convenient to refer to the sets of existential variables $X = \{x_1, \dots, x_{n+1}\}$ and universal variables $U = \{u_1, \dots, u_n\}$, and their subsets X_i (resp. U_i) restricted to $\{x_1, \dots, x_i\}$ (resp. $\{u_1, \dots, u_i\}$). For any line L, we define the *level* $\text{lev}(L)$ of the line to be the least i such that $\text{vars}(L) \subseteq X_i \cup U_{i-1}$.

The semantics of a QBF Φ can be understood by expanding the quantifiers in the prefix, i.e. by repeatedly applying the equivalences $\exists x \; \Phi \equiv \Phi[x/0] \vee \Phi[x/1]$ and $\forall x \; \Phi \equiv \Phi[x/0] \wedge \Phi[x/1]$.

Alternatively, the semantics of QBFs can be conveniently described as a two player game between an existential player and a universal player. At the ith round of the game, the existential player assigns a value to x_i, and then the universal player assigns a value to u_i. The game concludes after the existential player has assigned a value to x_{n+1}. The existential player wins the game if the matrix ϕ evaluates to true under the assignment constructed, whereas the universal player wins the game if ϕ evaluates to false. A QBF Φ is false (true) if and only if the universal (existential) player has a winning strategy for the game played on Φ.

We can describe a strategy for the universal player for Φ formally as a function $S : \langle X \rangle \to \langle U \rangle$ such that for any $\alpha, \gamma \in \langle X \rangle$, and $1 \leq i \leq n$, if $\alpha^i = \gamma^i$, then $S(\alpha)^i = S(\gamma)^i$, i.e. a strategy's response on u_i depends only on the existential variables to the left of u_i. A strategy S for the universal player for Φ is *winning* if $\phi[\alpha \cup S(\alpha)] = \bot$ for any $\alpha \in \langle X \rangle$.

2.2. QBF proof systems

Informally, a proof system for a language $\mathcal{L}$ is a definition of what is considered to be a proof that $\Phi \in \mathcal{L}$ [6]. The key features of a proof system are that it is *sound* – only formulas in $\mathcal{L}$ have proofs, *complete* – all formulas in

$\mathcal{L}$ have proofs, and that there is an algorithm, with running time polynomial in $|\pi|$, to check whether π is a proof that $\Phi \in \mathcal{L}$.

In the present work, we consider refutational proof systems for the languages SAT and TQBF, of satisfiable CNFs and true QBFs respectively. As such, we use the terms proof and refutation interchangeably. A *line-based* proof system P defines what axioms may be introduced given a formula Φ, and a sound set of rules for deducing new lines from preceding ones.

A proof of Φ in P consists of a sequence of lines $L_1, \ldots, L_m$, with $L_m = \bot$, concluding that Φ is unsatisfiable (SAT) or false (TQBF). Each line L_i is either an axiom introducible from Φ, or is derived using an inference rule of P with antecedents $L_{i_1}, \ldots, L_{i_k}$, for some $i_1, \ldots, i_k < i$. Since π is an ordered sequence of lines, we write $L_1 <_\pi L_2$ if L_1 appears before L_2 in the sequence.

We can also consider such a proof π as a directed acyclic graph (dag) with edges from each L_{i_j} to L_i for each L_i derived using an inference rule as above. We say that L_i precedes L_j in π, and write $L_i \prec_\pi L_j$ if there is a path from L_i to L_j in this dag. It is clear that $\prec_\pi$ is a restriction of the order $<_\pi$ induced by the order in which the lines appear in π.

In a *tree-like* proof system, each line can be used as an antecedent at most once; the line must be rederived each time it is used as the antecedent in a deduction. As a result, the corresponding dag must be a tree. We refer to proof systems without this restriction as *dag-like*.

The most widely-studied line-based proof system for the SAT problem is Resolution, the tree-like version of which corresponds to the DPLL algorithm for SAT solving [17, 18]. A Resolution refutation of ϕ is a deduction of the empty clause, representing $\bot$, from the clauses of ϕ using only the resolution rule: $\dfrac{C \vee x \qquad D \vee \neg x}{C \vee D}$.

Many variants of Resolution and other line-based propositional proof systems have been studied. In particular, rather than using clauses, the Frege and Extended Frege proof systems operate using any Boolean formula (respectively circuit) and any sound and complete set of deduction rules [6, 7]. More generally, $\mathcal{C}$-Frege systems use lines which are circuits from the circuit class $\mathcal{C}$ with a suitable sound and complete set of deduction rules for circuits in $\mathcal{C}$. The strength of these systems is such that the tree-like versions of Frege and Extended Frege are equivalent to the dag-like versions [16]; this also holds for some weaker circuits classes such as AC^0 and TC^0.

There have been several paradigms proposed to extend propositional calculi to proof systems for QBFs. Perhaps the most prominent of these is the introduction of the ∀-reduction rule to the set of deduction rules [9, 4]. Given a line-based propositional proof system P and a QCNF $\Phi = \Pi \cdot \phi$, P+$\forall$$\mathsf{red}$ allows the same axioms (from ϕ) and deduction rules as P, but also allows the deduction of $C[u/b]$ from C for some $b \in \{0, 1\}$ whenever a universal variable u is right of all other variables in C with respect to the quantifier prefix Π. Given a few very natural restrictions on the proof system P, which all proof systems above satisfy, the proof system P+$\forall$$\mathsf{red}$ is sound and complete [4, 5].

Any lower bound for a propositional proof system P immediately extends to a lower bound for P+$\forall$$\mathsf{red}$ by quantifying all variables existentially. As observed in [19, 20], these bounds do not provide any information about the interaction of the proof system with the quantification of the variables. In the case of P+$\forall$$\mathsf{red}$, *genuine* QBF lower bounds can be identified by providing a lower bound on the total size of the ∀-reduction steps. A formal model for 'genuine' QBF lower bounds is developed in [20].

2.3. Restricting proofs

Finally, we provide a precise definition of restricting a proof by an assignment. A proof π of a QBF Φ in a proof system P+$\forall$$\mathsf{red}$ can be restricted by any assignment to a subset of the existential variables. If the leftmost variable in Φ is universal, π can be restricted by an assignment to this variable which witnesses that Φ is false. In both cases, the restricted proof $\pi[\alpha]$ will then be a proof of $\Phi[\alpha]$.

To construct $\pi[\alpha]$, let L_α be the first line in π which restricts to $\bot$ under this assignment. We remove from π any lines after L_α, and restrict every line by α. Finally, we remove any lines which now evaluate to $\top$, and iteratively remove any sinks which are not $L_\alpha[\alpha] = \bot$, so that no lines which are not directly used to derive $\bot$ are contained in $\pi[\alpha]$. This step of removing superfluous lines need not be included in the definition of a restriction, as such lines are permitted in a proof. We include it here as it greatly simplifies the structure of the restricted proofs.

3. Lower bounds on paths in P+∀red proofs

Suppose π is a P+$\forall$$\mathsf{red}$ refutation of a QBF Φ. Since P+$\forall$$\mathsf{red}$ is sound, Φ is false, and so there is a winning strategy for the universal player in the two-player game on Φ. In [13], a strategy extraction algorithm based on the restriction of refutations was developed, which we now describe.

Definition 1 (Strategy extraction algorithm [13]). Let π be a $\mathsf{P}+\forall\mathsf{red}$ refutation of a false QBF Φ, and let $\alpha \in \langle X \rangle$ be an assignment to the existential variables of Φ. The universal player's response β is constructed round by round. Let $\pi_1^\alpha = \pi[\alpha^1]$, and construct the response at round i as follows:

- Define L_i^α to be the final line in π_i^α. If L_i^α is derived by a $\forall$-reduction substituting u_i/b, define $\beta(u_i) = b$, else define $\beta(u_i) = 0$ when L_i^α is derived by a propositonal rule or by a $\forall$ reduction on $u_{i'}$ for $i' > i$.
- Restrict π_i^α by $\beta^i \cup \alpha^{i+1}$ to give $\pi_{i+1}^\alpha = \pi_i^\alpha[\alpha^{i+1} \cup \beta^i]$.

After n rounds, this constructs a complete universal response $\beta \in \langle U \rangle$, and the response at round i was computed using only the assignment α^i.

Observe that this strategy extraction algorithm not only defines a response for each existential assignment α, but also constructs a sequence of lines L_i^α from which the universal response on u_i is extracted. We use the notation L_i^α to refer to the line in π which becomes the final line in π_i^α under restriction by $\alpha \cup S_\pi(\alpha)$. We primarily concern ourselves with which lines are present in the restricted proofs π_i^α, so for a line $L \in \pi$, we write $L \in \pi_i^\alpha$ whenever $L[\alpha^i \cup S_\pi(\alpha)^{i-1}] \in \pi_i^\alpha$.

Since the response for u_i is determined by the deduction rule used to derive L_i^α, assignments with different responses must result in different sequences of lines of π.

Lemma 2. *Let π be a $\mathsf{P}+\forall\mathsf{red}$ refutation of Φ. If the assignments $\alpha, \gamma \in \langle X \rangle$ result in different responses under the strategy extraction algorithm, then there is some k such that $L_k^\alpha \neq L_k^\gamma$.*

Proof. Let $\beta_\alpha, \beta_\gamma \in \langle U \rangle$ be the responses to α and γ respectively. Without loss of generality, since $\beta_\alpha \neq \beta_\gamma$, let k be such that $\beta_\alpha(u_k) = 1$ and $\beta_\gamma(u_k) = 0$. Therefore, L_k^α is derived in π by a $\forall$-reduction step substituting $u_k/1$, whereas L_k^γ is derived by a $\forall$-reduction step using $u_k/0$, or by a propositional deduction rule. In either case, it is clear that $L_k^\alpha \neq L_k^\gamma$. □

We emphasise that the lines L_k^α and L_k^γ in Lemma 2 are distinct *as lines of* π. For example, a $\mathsf{Frege}+\forall\mathsf{red}$ refutation, particularly a tree-like refutation, may derive multiple copies of the same formula. Since these copies are considered distinct lines of π, L_k^α and L_k^γ may therefore still be identical as formulas, despite being distinct as lines of π.

Some definitions of the round-based strategy extraction algorithm use the restricted proof $\pi[\alpha^{i+1} \cup \beta^i]$ at the ith round, rather than using $\pi_i^\alpha[\alpha^{i+1} \cup \beta^i]$. Both result in a winning universal strategy, however since $\pi[\alpha^{i+1} \cup \beta^i]$ and $\pi_i^\alpha[\alpha^{i+1} \cup \beta^i]$ are not necessarily identical, they may result in different winning strategies.

Using restrictions of π_i^α rather than π ensures the following useful property of the lines L_i^α: for any assignment $\alpha \in \langle X \rangle$, and any $i < j$, either $L_i^\alpha = L_j^\alpha$, or $L_i^\alpha \succ_\pi L_j^\alpha$. We can use the strategy extraction algorithm to extend this sequence to a path through π corresponding to the run of the strategy extraction algorithm on π and α.

Definition 3. 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 $L \in \pi_i^\alpha$.

Several such paths may exist; to ensure the uniqueness of p_α, define p_α to be the first such path in the lexicographic ordering induced by $<_\pi$. However the properties above are the only ones we shall use in this work, so any suitable path could be chosen.

Proposition 4. *Let π be a* P+$\forall$red *refutation of a false QBF Φ. For any assignments $\alpha, \gamma \in \langle X \rangle$ which produce distinct responses using the strategy extraction algorithm on π, $p_\alpha \neq p_\gamma$.*

Proof. Define $L_0^\alpha = L_0^\gamma = \bot$ to be the final line of π. By Lemma 2, there is some $1 \leq k \leq n$ such that $L_k^\alpha \neq L_k^\gamma$; pick the least such k, so that $L_{k-1}^\alpha = L_{k-1}^\gamma = L_{k-1}$.

If L_k^α and L_k^γ are incomparable in the partial order $\prec_\pi$, then no path can contain both L_k^α and L_k^γ and the paths p_α and p_γ are distinct. 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 have $L \in \pi_k^\gamma$. To show $p_\alpha \neq p_\gamma$, it therefore suffices to show that $L_k^\alpha \notin \pi_k^\gamma$ and hence $L_k^\alpha \notin p_\gamma$, since $L_k^\alpha \in p_\alpha$.

It is clear that if $L_k^\alpha \notin \pi_{k-1}^\gamma$, then $L_k^\alpha \notin \pi_k^\gamma$ and we are done, so assume $L_k^\alpha \in \pi_{k-1}^\gamma$. By the definition of L_k^α, $\mathrm{lev}(L_k^\alpha) \leq k$, so the assignment $\gamma^k \cup \beta^{k-1}$ is a total assignment to the variables of L_k^α. It cannot be the case that $L_k^\alpha[\gamma^k \cup \beta^{k-1}] = \bot$, as this contradicts the choice of L_k^γ as the first line in π_{k-1}^γ which restricts to $\bot$ under this assignment, hence $L_k^\alpha[\gamma^k \cup \beta^{k-1}] = \top$.

As tautologies are removed from the restricted proof $\pi^{\gamma}_{k-1}[\gamma^k \cup \beta^{k-1}] = \pi^{\gamma}_k$, $L^{\alpha}_k \notin \pi^{\gamma}_k$ and $p_\alpha \neq p_\gamma$. □

Given that assignments resulting in different responses from the strategy extraction algorithm give rise to distinct paths in the proof, it is natural to define a measure counting the number of distinct responses required in a strategy. We can then use Proposition 4 to gain some understanding of the structure of P+∀red proofs of QBFs requiring a large number of responses.

Definition 5 ([14]). For any QBF Φ, the *strategy size* $\rho(\Phi)$ is the minimal size of the range of a winning strategy for Φ:

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

Corollary 6. *For any QBF Φ and P+∀red proof π of Φ, the strategy extraction algorithm constructs at least $\rho(\Phi)$ distinct paths through π.*

This lower bound on the number of paths demonstrates the importance of reusing lines in the derivation, as this allows multiple distinct paths through the same line. In the case of tree-like P+∀red proofs, where lines cannot be reused, the lower bound on paths immediately gives a lower bound for proof size based only on the relatively simple measure of strategy size, and independent of the base propositional proof system.

Theorem 7. *For any QBF Φ, if π is a tree-like P+∀red proof of Φ, then $|\pi| \geq \rho(\Phi)$.*

Proof. Since π is a tree-like proof, there is a unique path from each axiom to the final line of the proof. By Corollary 6, there are at least $\rho(\Phi)$ paths through π, so π contains at least $\rho(\Phi)$ axioms. □

To show a lower bound on tree-like P+∀red proofs, it therefore suffices to show a lower bound on $\rho(\Phi_n)$ for some family of QBFs Φ_n. There are already several examples of such QBF families in the literature, such as the formulas defined by Kleine Büning et al. [9] or the equality formulas defined in [5]. The formulas we choose to exemplify such a lower bound were defined in [21], where it was noted that these formulas have short QU-Res proofs (QU-Res coincides with Res +∀red). These formulas therefore not only provide a lower bound for tree-like Frege+∀red and EF+∀red, but also a separation between tree-like EF+∀red and dag-like QU-Res.

Corollary 8. *If* π *is a tree-like* Frege+∀red *or* EF+∀red *proof of*

$$\Phi_n := \exists x_1 \forall u_1 \exists t_1 t_2 \dots \exists x_n \forall u_n \exists t_{2n-1} t_{2n}.$$
$$\bigwedge_{i=1}^{n} [(\neg x_i \vee t_{2i-1}) \wedge (\neg u_i \vee t_{2i-1}) \wedge (x_i \vee t_{2i}) \wedge (u_i \vee t_{2i})] \wedge \bigvee_{j=1}^{2n} \neg t_j$$

then $|\pi| \geq 2^n$.

Proof. The only winning universal strategy is to play $u_i = \neg x_i$. This forces the existential player to set both t_{2i-1} and t_{2i} positively, ultimately falsifying the large clause at the final round. Given this unique winning strategy, $\rho(\Phi_n) = 2^n$, and the lower bound follows by Theorem 7. □

Given the equivalences previously shown between various tree-like and dag-like versions of Frege+∀red and EF+∀red, this lower bound may at first seem surprising. In [11], it was shown that tree-like and dag-like Frege+∀red are equivalent when the ∀-reduction rule can substitute in any suitable Boolean formula (instead of just constants 0/1 as defined here). Additionally, [11] shows that the *dag-like* Frege+∀red systems allowing reduction by $\{0, 1\}$ and allowing reduction by any Boolean formula are equivalent. The same two equivalences hold for EF+∀red in place of Frege+∀red, where for EF+∀red we allow substitutions by Boolean circuits.

However, both of these equivalences rely on the fact that the other restriction is not present. Restricting proofs to be tree-like *and* only allowing ∀-reduction on $\{0, 1\}$ results in a substantially weaker system, as shown by the lower bound in Corollary 8. As Frege+∀red p-simulates QU-Res, we can conclude that tree-like Frege+∀red is exponentially weaker than dag-like Frege+∀red (both in the version with 0/1-reduction), whereas the latter is equivalent to tree-like Frege+∀red and dag-like Frege+∀red where universal reduction substitutes Boolean formulas.

4. Characterising tree-like Frege+∀red and EF+∀red lower bounds

In [11], a characterisation of superpolynomial lower bounds for Frege+∀red and EF+∀red was established. By giving a normal form for proofs in these proof systems, into which any proof can be efficiently transformed, it was shown that any lower bounds on (dag-like) Frege+∀red or EF+∀red proofs are a result of lower bounds on propositional proofs, or circuit complexity lower bounds.

The lower bound and consequent separation shown in Corollary 8 demonstrates that this characterisation does not hold for tree-like Frege+∀red or tree-like EF+∀red. However, with a variation of the normal form, we can extend this characterisation to these tree-like systems, with any lower bounds not characterised by propositional or circuit complexity lower bounds being the result of a strategy size lower bound. Similarly to the characterisation of [11], our characterisation also holds for $\mathcal{C}$-Frege+∀red for circuit classes such as AC^0 and TC^0 with the circuit lower bounds for the corresponding circuit class $\mathcal{C}$, but for clarity we refer only to Frege+∀red and EF+∀red throughout this section.

It is known that circuits computing winning strategies for the universal player can be constructed in polynomial time from a Frege+∀red or EF+∀red refutation π [4]. However, the construction of these circuits uses a different algorithm, possibly resulting in a different winning strategy from that constructed by the round-based algorithm. To give a normal form for tree-like proofs, we begin by extending this strategy extraction result to show that in the case of a tree-like Frege+∀red or EF+∀red proof, we can ensure that these circuits compute the winning strategy produced by the strategy extraction algorithm as given in Definition 1.

Lemma 9. *Let π be a tree-like* Frege+∀red *(resp. tree-like* EF+∀red*) refutation. There are formulas (resp. circuits) C_i with inputs $\{x_1, u_1, \dots, x_i\}$ of size $O(|\pi|^2)$ computing the strategy for u_i extracted from π by the strategy extraction algorithm in Definition 1.*

Proof. A *decision list* for a Boolean function is a sequence of lines of the form 'if C then b, else ... ' with the circuits C in some class $\mathcal{C}$ and $b \in \{0, 1\}$. Given a decision list for u_i with circuits in NC^1 (or $\mathsf{P/poly}$), there is a formula (or circuit) computing the same function of size polynomial in that of the decision list (for details, see [4]). We therefore reduce the problem to finding a suitable decision list for u_i.

Furthermore, for a tree-like proof π, the construction of π_i^α depends only on the lines selected at each round by the strategy extraction algorithm, and is independent of the precise assignment α. That is, for each line L and each $i \geq \text{lev}(L)$, we can construct a proof π_i^L such that for any $\alpha \in \langle X \rangle$ where $L_i^\alpha = L$, $\pi_i^\alpha = \pi_i^L$.

The proof π_i^L contains all lines L' such that $L' \preceq_\pi L$, $\text{lev}(L') > i$ and for any L'' with $L' \preceq_\pi L'' \preceq_\pi L$, $\text{lev}(L'') > i$. This is because for any line L' with

$\text{lev}(L') \leq i$ which has L as a descendant, it must be the case that $L'[\alpha] = \top$ for any assignment α which chooses L at round i.

We can now construct decision lists for each variable $u_i \in U$. Let $L \in \pi$ be a line such that $\text{lev}(L) \leq i$, i.e. all variables in L are assigned by the ith round. We construct a conjunction C_i^L of lines of π and their negations, all with level at most i, which is a sufficient and necessary condition to ensure that the strategy extraction algorithm selects L in the ith round.

Define $C_0^\perp = \top$. 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; specifically, this is the first descendant of L with level $i-1$. Having chosen M at round $i-1$, the restricted proof is therefore π_{i-1}^M, and so to choose L at round i the algorithm must verify that L evaluates to $\perp$, and also verify that no lines in π_{i-1}^M preceding L evaluate to $\perp$. The set of lines the algorithm considers at this round is therefore $\mathcal{L}_i^L = \{L' \in \pi_{i-1}^M : L' <_\pi L, \text{lev}(L') = i\}$, resulting in the conjunction

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

For each line $L \in \pi$ with $\text{lev}(L) \leq i$, we can add to the decision list for u_i the line

$$\text{if } C_i^L \text{ then } b_L$$

where b_L is the value assigned to u_i by the strategy extraction algorithm if L is the line at the root of π_i^α. It is clear that this decision list computes the same strategy as given by the algorithm. Furthermore, each line of π can only appear in one polarity in the conjunction C_i^L, and the number of lines in the decision list for u_i is at most the number of lines in π. The size of the decision list, and therefore the size of the circuit constructed from it, is $O(|\pi|^2)$. $\square$

Having shown the existence of small circuits computing this strategy, we can now use them to define the normal form for tree-like Frege+$\forall$red and EF+$\forall$red proofs which gives our characterisation.

This normal form is based on the normal form used in [11] to provide a characterisation for dag-like Frege+$\forall$red and EF+$\forall$red proofs. We begin in the same way, using the fact that the C_i form a winning strategy for the universal variables to derive the line $\bigvee_{i=1}^n (u_i \not\leftrightarrow C_i)$. However, instead of deriving it only once, we derive a copy of the line for each response β given by the winning strategy described by the C_i. The normal form proof proceeds by reducing each u_j in turn according to the corresponding response for that line, and then combining lines whose responses first differ on u_j to derive a copy of $\bigvee_{i=1}^{j-1}(u_i \not\leftrightarrow C_i)$ for each response to the variables $u_1, \ldots, u_{j-1}$, ultimately deriving the empty disjunction after reducing u_1.

Definition 10 formalises this form of proof; in Lemma 11 we show how to efficiently transform any tree-like Frege+∀red or EF+∀red proof into such a normal form.

Definition 10 (Normal form for proofs). Let $\Phi = \Pi \cdot \phi$ be a QBF, and let the circuits C_i compute a winning universal strategy for the variables u_i. Define $S : \langle X \rangle \to \langle U \rangle$ to be the strategy computed by the C_i, with $\mathrm{rng}(S) = \{\beta_1, \ldots, \beta_s\}$. 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 $\phi \models \bigvee_{i=1}^{n}(u_i \not\leftrightarrow C_i)$.

The proof 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 universal literals falsified by β_j. Each line is now ∀-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 a propositional inference from a single line if β_j is the unique extension of β_j^{n-1} in $\mathrm{rng}(S)$, or by combining the lines corresponding to the two extensions of β_j^{n-1} otherwise. By repeating this process for each universal variable from u_n to u_1, we eventually derive $\bot$.

Given a proof π, Lemma 9 produced circuits of size $|\pi|^{O(1)}$ which compute a strategy S with $|\mathrm{rng}(S)| \leq |\pi|$. By using these circuits as the circuits C_i in the normal form, we are able to construct from π a proof in this normal form with only a polynomial increase in size.

Lemma 11. *Given a QBF $\Phi = \Pi \cdot \phi$, and a tree-like* Frege+∀red *(respectively tree-like* EF+∀red*) proof π of Φ , there is a tree-like* Frege+∀red *(respectively tree-like* EF+∀red*) proof of Φ of the form in Definition 10 with size $|\pi|^{O(1)}$.*

Proof. Let the circuits C_i be those constructed in Lemma 9. These circuits have size $|\pi|^{O(1)}$, and by applying Proposition 4 it is clear that the corresponding strategy S satisfies $|\mathrm{rng}(S)| = |\pi|^{O(1)}$. As dag-like and tree-like propositional Frege systems are equivalent [16], it suffices to show that each of the propositional inferences described in Definition 10 has a dag-like proof of size $|\pi|^{O(1)}$.

To first derive $\bigvee_{i=1}^{n}(u_i \not\leftrightarrow C_i) \vee \neg\beta$ for some $\beta \in \mathrm{rng}(S)$, we construct from π a proof that $\phi \wedge \beta \wedge \bigwedge_{i=1}^{n}(u_i \leftrightarrow C_i) \to \bot$ by deriving for each line

$L \in \pi$, the line $\neg C^L = \neg C^L_j$ where C^L_j is as in (1) and $j = \text{lev}(L)$. For the final line $\bot$, $\neg C^\bot_0 = \bot$ so this is indeed a derivation of $\bot$.

To begin, note that if we have derived $\neg C^M$ for each $M <_\pi L$, then it suffices to derive (a subclause of) $\neg C^L_i$ for any $i \geq \text{lev}(L)$, since $\neg C^L_i = \neg C^L \vee \bigvee_k \neg M_k$ for those lines $M_k \prec_\pi L$ checked by the algorithm between choosing L at round $\text{lev}(L)$ and round i. Each C^{M_k} contains C^L, so each instance of $\neg M_k$ can be 'resolved' away in turn using $\neg C^{M_k}$ to obtain $\neg C^L$. As $k \leq |\pi|$, this requires size $|\pi|^{O(1)}$.

First, suppose L is introduced as an axiom in π, i.e. L is a clause in ϕ. For any line L, the disjunction $\neg C^L$ contains L so it is clear that there is a derivation of C^L from the axiom L, and hence from the clauses of Φ, which has size $O(|C^L|)$.

If L is derived from L' by a $\forall$-reduction on u_i which agrees with β, then C^L_{i+1} is identical to $C^{L'} = C^{L'}_{i+1}$ with $\neg L'$ replaced by L'. Using β, it is straightforward to derive L from L' and thus deduce from $\neg C^{L'}$ a stronger clause than $\neg C^L_{i+1}$.

If L is derived by a $\forall$-reduction on u_i which does not agree with β, then there is a derivation of $\neg C^L_i$ from $\beta \wedge (u_i \leftrightarrow C_i)$ and the already derived lines $\neg C^M$ for $M <_\pi L$. Since C_i is a decision list, if each C^M is false but C^L_i is true, it requires $O(|C_i|)$ lines to evaluate the decision list and conclude that $C_i \not\leftrightarrow \beta(u_i)$, from which a contradiction can easily be derived using $\beta \wedge (u_i \leftrightarrow C_i)$.

Lastly, suppose L is derived by a propositional rule from L_1 and L_2. Without loss of generality, we can assume that $L_1 <_\pi L_2$ and that $\text{lev}(L_1) \leq \text{lev}(L_2) = l$. 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}. It is clear that $\text{lev}(L) \leq \text{lev}(L_2)$. Furthermore, C^L_l contains all conjuncts in C^{L_2} except $\neg L_2$, as we assume without loss of generality that L is the next line derived after L_2. Since $\neg C^L_l$ contains L as a disjunct, we can derive a subclause of $\neg C^L_l$ in size linear in $|C^{L_2}|$ by using L_1 in $\neg C^{L_1}$ and L_2 in $\neg C^{L_2}$ to derive L.

Having derived $\bigvee_{i=1}^{n}(u_i \not\leftrightarrow C_i) \vee \neg\beta$ for each response β, we now turn to the deduction of $\bot$ from these axioms. Since $(u_i \leftrightarrow \beta(u_i)) \wedge (u_i \leftrightarrow C_i)$ is equivalent to $C_i \leftrightarrow \beta(u_i)$, constructing $\bigvee_{i=1}^{j-1}(u_i \not\leftrightarrow C_i) \vee \neg\beta^{j-1}$ from the corresponding lines for two different extensions of β^{j-1} on u_j requires only proving that $C_j \wedge \neg C_j \models \bot$, which has a proof of size $O(|C_j|)$.

In the case where there is a unique extension of β^{j-1}, it is sufficient to prove $\bigwedge_{i=1}^{j}(C_i \leftrightarrow \beta(u_i))$ from $\bigwedge_{i=1}^{j-1}(C_i \leftrightarrow \beta(u_i))$. Construct for each i in turn the disjunction of the C^L_i which would result in the response β^i.

This can be constructed in size $|C_i|^{O(1)}$ at each stage. For each C_{j-1}^L in the final disjunction, there is a linear-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. $\square$

To show superpolynomial lower bounds on the size of tree-like $\mathsf{Frege}+\forall\mathsf{red}$ proofs, it is therefore sufficient to show such lower bounds on proofs of the form in Definition 10. We use this to provide a characterisation of such lower bounds, similar to that shown for $\mathsf{Frege}+\forall\mathsf{red}$ in [11].

Theorem 12. *Each of the following is sufficient to give a superpolynomial lower bound on tree-like* $\mathsf{Frege}+\forall\mathsf{red}$ *(resp. tree-like* $\mathsf{EF}+\forall\mathsf{red}$*) proofs:*

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

Moreover, any superpolynomial lower bound on tree-like $\mathsf{Frege}+\forall\mathsf{red}$ *(resp. tree-like* $\mathsf{EF}+\forall\mathsf{red}$*) is due to one of the above lower bounds.*

Proof. We just argue for $\mathsf{Frege}+\forall\mathsf{red}$; the $\mathsf{EF}+\forall\mathsf{red}$ case is analogous. First we show that each of items 1 to 3 is sufficient for a superpolynomial lower bound.

For item 1, it is clear that a Frege lower bound for propositional formulas ϕ_n implies a $\mathsf{Frege}+\forall\mathsf{red}$ lower bound for the existentially quantified version of ϕ_n.

For item 2, if Φ_n is a sequence of QBFs with a superpolynomial lower bound on $\rho(\Phi_n)$, then this is also a lower bound on the size of a tree-like $\mathsf{Frege}+\forall\mathsf{red}$ proof of Φ_n by Theorem 7.

To see that item 3 is sufficient, let Φ_n be a sequence of QBFs such that $\rho(\Phi_n)$ is small, but there are no polynomial-size circuits in NC^1 computing a universal winning strategy with small range. By Lemma 9, we can extract from a tree-like $\mathsf{Frege}+\forall\mathsf{red}$ proof π circuits of size $|\pi|^{O(1)}$ which compute a winning strategy S with $|\mathrm{rng}(S)| \leq |\pi|$. This provides a superpolynomial lower bound on $|\pi|$.

To argue that each lower bound for $\mathsf{Frege}+\forall\mathsf{red}$ arises from items 1 to 3, assume that Φ_n is a sequence of QBFs hard for $\mathsf{Frege}+\forall\mathsf{red}$, but for which neither item 2 nor item 3 holds. Then there exist circuits C_i of size polynomial in n computing a strategy S such that $|\text{rng}(S)|$ is polynomial in n. Use these circuits to construct a proof π of the form given in Definition 10. Since $|C_i|$ and $|\text{rng}(S)|$ are polynomial, any lower bound on $|\pi|$ is due to a propositional lower bound on one of the propositional subderivations in π. □

Note that (1) and (3) are almost identical to the characterisation of lower bounds on dag-like Frege+∀red from [11]. Indeed, if a tree-like Frege+∀red lower bound falls only under (3), the formulas can be easily modified to force the universal player's response to belong to rng(S) for some winning strategy S with a polynomial-size range. The circuit lower bound in (3) then translates into a lower bound on any circuit computing a winning strategy, giving a lower bound for dag-like Frege+∀red.

The key consequence of Theorem 12 therefore is this: not only does strategy size provide a simple method for producing tree-like Frege+∀red lower bounds, it is the *only* way to show such lower bounds which does not entail showing a lower bound for dag-like Frege+∀red.

5. Conclusion

By examining more closely the running of the round-based strategy extraction algorithm, we have shown how the structure of the proof relates to the trace of the algorithm. The simple structure of tree-like proofs then gives a simple lower bound on the size of these proofs. Moreover, extending a previous normal form for Frege+∀red and EF+∀red proofs to the tree-like version, we see that lower bounds of this form are the only lower bounds for tree-like Frege+∀red and EF+∀red which do not provide lower bounds for the corresponding dag-like systems.

On a broader scale, our results highlight an important distinction between two different approaches to the ∀-reduction rule. In many of the most studied proof systems, such as tree-like or dag-like QU-Res, and dag-like Frege+∀red, restricting ∀-reductions to 0-1 substitutions rather than any suitable formula defines an equivalent system. However, this does not hold for tree-like Frege+∀red, and for tree-like versions of several proof systems with comparatively expressive lines. In such proof systems, the choice of ∀-reduction rule must be considered more carefully. This separation may also limit the effectiveness of practical implementations corresponding to proof systems with a 0-1 ∀-reduction rule, including many modern QBF solvers.

It has been observed that in the case of dag-like systems, the Frege+∀red characterisation of [11] does not hold for weaker systems such as QU-Res [20].

However, the only known examples which do not fit this characterisation have large strategy size. It is thus a natural question whether the characterisation in Theorem 12 extends to weaker tree-like $\mathsf{P}+\forall\mathsf{red}$ systems.

Acknowledgments

Research was supported by grants from the John Templeton Foundation (grant no. 60842) and the Carl Zeiss Foundation.

References

[1] S. R. Buss, Towards NP-P via proof complexity and search, Ann. Pure Appl. Logic 163 (7) (2012) 906–917.

[2] E. Ben-Sasson, A. Wigderson, Short proofs are narrow - resolution made simple, Journal of the ACM 48 (2) (2001) 149–169.

[3] O. Beyersdorff, L. Chew, M. Janota, Proof complexity of resolution-based QBF calculi, in: Proc. Symposium on Theoretical Aspects of Computer Science (STACS'15), LIPIcs, 2015, pp. 76–89.

[4] O. Beyersdorff, I. Bonacina, L. Chew, Lower bounds: From circuits to QBF proof systems, in: Proc. ACM Conference on Innovations in Theoretical Computer Science (ITCS'16), ACM, 2016, pp. 249–260.

[5] O. Beyersdorff, J. Blinkhorn, L. Hinde, Size, cost, and capacity: A semantic technique for hard random QBFs, Logical Methods in Computer Science 15 (1) (2019) 13:1–39.

[6] S. A. Cook, R. A. Reckhow, The relative efficiency of propositional proof systems, The Journal of Symbolic Logic 44 (1) (1979) 36–50.

[7] E. Jeřábek, Weak pigeonhole principle, and randomized computation, Ph.D. thesis, Faculty of Mathematics and Physics, Charles University, Prague (2005).

[8] P. Beame, T. Pitassi, Propositional proof complexity: Past, present, and future, in: G. Paun, G. Rozenberg, A. Salomaa (Eds.), Current Trends in Theoretical Computer Science: Entering the 21st Century, World Scientific Publishing, 2001, pp. 42–70.

[9] H. Kleine Büning, M. Karpinski, A. Flögel, Resolution for quantified Boolean formulas, Information and Computation 117 (1) (1995) 12–18.

[10] A. Van Gelder, Contributions to the theory of practical quantified Boolean formula solving, in: Proc. Principles and Practice of Constraint Programming (CP'12), 2012, pp. 647–663.

[11] O. Beyersdorff, J. Pich, Understanding Gentzen and Frege systems for QBF, in: Proc. ACM/IEEE Symposium on Logic in Computer Science (LICS), 2016.

[12] E. Giunchiglia, M. Narizzano, A. Tacchella, Clause/term resolution and learning in the evaluation of quantified boolean formulas, J. Artif. Intell. Res. 26 (2006) 371–416.

[13] A. Goultiaeva, A. V. Gelder, 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.

[14] O. Beyersdorff, J. Blinkhorn, Genuine lower bounds for QBF expansion, in: 35th Symposium on Theoretical Aspects of Computer Science (STACS), 2018, pp. 12:1–12:15.

[15] O. Beyersdorff, L. Chew, M. Janota, On unification of QBF resolution-based calculi, in: MFCS, II, 2014, pp. 81–93.

[16] J. Krajíček, Bounded Arithmetic, Propositional Logic, and Complexity Theory, Vol. 60 of Encyclopedia of Mathematics and Its Applications, Cambridge University Press, Cambridge, 1995.

[17] M. Davis, G. Logemann, D. W. Loveland, A machine program for theorem-proving, Commun. ACM 5 (7) (1962) 394–397.

[18] M. Davis, H. Putnam, A computing procedure for quantification theory, Journal of the ACM 7 (1960) 210–215.

[19] H. Chen, Proof complexity modulo the polynomial hierarchy: Understanding alternation as a source of hardness, ACM TOCT 9 (3) (2017) 15:1–15:20.

[20] O. Beyersdorff, L. Hinde, J. Pich, Reasons for hardness in QBF proof systems, in: Conference on Foundations of Software Technology and Theoretical Computer Science, 2017, pp. 14:1–14:15.

[21] M. Janota, J. Marques-Silva, Expansion-based QBF solving versus Q-resolution, Theor. Comput. Sci. 577 (2015) 25–42.