Polynomial Calculus for Quantified Boolean Logic: Lower Bounds Through Circuits and Degree

Olaf Beyersdorff
Friedrich Schiller University Jena, Germany

Tim Hoffmann
Friedrich Schiller University Jena, Germany

Kaspar Kasche
Friedrich Schiller University Jena, Germany

Luc Nicolas Spachmann
Friedrich Schiller University Jena, Germany

Abstract

We initiate an in-depth proof-complexity analysis of polynomial calculus ($\mathcal{Q}$-PC) for Quantified Boolean Formulas (QBF). In the course of this we establish a tight proof-size characterisation of $\mathcal{Q}$-PC in terms of a suitable circuit model (polynomial decision lists). Using this correspondence we show a size-degree relation for $\mathcal{Q}$-PC, similar in spirit, yet different from the classic size-degree formula for propositional PC by Impagliazzo, Pudlák and Sgall (1999).

We use the circuit characterisation together with the size-degree relation to obtain various new lower bounds on proof size in $\mathcal{Q}$-PC. This leads to incomparability results for $\mathcal{Q}$-PC systems over different fields.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases proof complexity, QBF, polynomial calculus, circuits, lower bounds

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.27

Funding *Olaf Beyersdorff*: Carl-Zeiss Foundation and DFG grant BE 4209/3-1
Tim Hoffmann: Carl-Zeiss Foundation
Kaspar Kasche: Carl-Zeiss Foundation

1 Introduction

Proof complexity studies the problem to understand the minimal size of proofs of specific formulas in various formal proof systems. The field bears deep connections to computational complexity [28,40], logic – mainly through the correspondence to bounded arithmetic [8,27,40] – and has practical significance due to intricate relations to SAT solving [23]. In fact, proof complexity is the main theoretical framework to assess the strength and limitations of solvers.

While traditionally proof complexity concentrated on propositional logic, there has been intense work in the past two decades on proof complexity for further logics, most notably for *Quantified Boolean Formulas* (QBF) [9], but also for other non-classical logics such as modal and intuitionistic logics [19,36,46]. For QBF, one of the main drivers for the field has been significant advances in QBF solving [18,44]. As in the propositional case, QBF proof complexity provides the theoretical tools to model, assess and guide QBF solving [12,21,38].

In propositional proof complexity, various proof systems have been studied intensively, including resolution, Frege systems, algebraic and geometric systems [40]. While resolution has arguably received most attention – and underpins modern SAT solving in the form of CDCL [2,5,42] – algebraic proof complexity has enjoyed a boost of interest in the past decade with many strong results shown for Nullstellensatz, polynomial calculus (PC), sum of squares (SOS), and very strong systems such as the ideal proof system (cf. e.g. [26,29,31,32,34,35,43]).

Algebraic proof systems typically work with polynomials and the central system of polynomial calculus [25] is a refutational proof system demonstrating that a given set of polynomial equations does not admit a common solution.

Similarly, in QBF proof complexity there are many results on various QBF resolution systems [4, 9, 11, 14]. Yet, in stark contrast to the propositional case, information on algebraic proof systems for QBF is rather scarce. A version of polynomial calculus for QBF – called $\mathcal{Q}$-PC here – is straightforward to define [13] as there is a general framework how to lift a line-based propositional proof system P – fulfilling some modest closure properties – to a quantified system $\mathcal{Q}$-P by adding just one rule of universal reduction that allows to substitute a universal variable u from a formula F (under the condition that u is quantified rightmost in F) [13]. This system $\mathcal{Q}$-PC naturally works with polynomials as lines and provides a succinct way to prove the falsity of QBFs. Hence we view this algebraic system as a refutational system for QBFs. The existential and universal variables are therefore propositional and take 0/1 values in accordance with the QBF semantics, while intermediate values computed by the polynomials can be arbitrary field elements, making proofs more succinct.

So far, the only information on proof size in $\mathcal{Q}$-PC stems from the general semantic technique of cost through the size-cost-capacity theorem from [10] which allows to obtain lower bounds for QBF proof systems of bounded capacity (which applies to $\mathcal{Q}$-PC as well as to most QBF resolution systems). With the cost technique, QBFs become hard to prove whenever the universal player needs large winning strategies (measured as the number of different answers of the universal player in the game interpretation of QBFs) and these lower bounds simultaneously hold in all QBF systems to which this technique is applicable. Hence this method does not allow to separate QBF resolution from $\mathcal{Q}$-PC, for example.

One key motivation to study algebraic proof systems in the propositional case is their recently emerging connection to algebraic circuit complexity [31, 35, 43]. In general, a correspondence between progress for lower bounds for circuit and proof size has often been postulated (e.g. [6]), but formal connections for propositional proofs could not yet be established outside the algebraic domain. In fact, it could be argued that this correspondence perfectly works in the QBF setting: for QBF resolution – tightly corresponding to a version of decision lists [11] – and for QBF Frege systems where proof size is characterised by circuit size in Boolean circuits [13]. This is quite fruitful as it allows a direct transfer of known circuit lower bounds to proof complexity, e.g. from $\mathsf{AC}^0[p]$ to the corresponding system of $\mathcal{Q}$-$\mathsf{AC}^0[p]$-Frege [13, 47]. A similar transfer in the propositional case remains wide open.

Curiously, an analogous relation between algebraic circuits and algebraic QBF systems is missing, whereas exactly in this algebraic case, some connections are known propositionally [31, 35, 43], as mentioned above.

Our aim in this paper is to initiate a comprehensive analysis of the algebraic system $\mathcal{Q}$-PC. In the course of this investigation we achieve a circuit characterisation for $\mathcal{Q}$-PC. This leads to new lower bound techniques for proof size in $\mathcal{Q}$-PC, which we apply to show a number of new proof size lower bounds for this system.

Our contributions

A. Circuit characterisation for $\mathcal{Q}$-PC. Our first result is a tight circuit characterisation of $\mathcal{Q}$-PC proof size by circuit size in an appropriate circuit model. The circuit model in question is a generalisation of decision lists [45], which are lists of simple statements of the form: If *(condition on existential variables)* Then *(assignment to universal variables)*.

The decision lists – termed PDLs here for *p*olynomial *d*ecision *l*ists – have polynomial equations in existential variables as conditions and compute a complete assignment to the universal variables. Semantically, a PDL for a quantified set of polynomial equations Φ computes a countermodel for Φ in the two-player game semantics of QBFs.

We show that the minimal proof size for Φ (of bounded quantifier complexity) in $\mathcal{Q}$-PC is polynomially equivalent to the minimal size of a PDL for Φ. In fact, we show a more general result that applies to a whole class of QBF proof systems with bounded capacity [10] (and fulfilling some closure properties). The result is parameterised by the lines of the proof system, which in turn correspond to the conditions in the decision lists. This generalises a result for Q-Resolution [11] and lifts it to $\mathcal{Q}$-PC.

B. Size-degree relation for $\mathcal{Q}$-PC. Having the PDL characterisation in place, we can obtain a size-degree result, relating minimal proof size in $\mathcal{Q}$-PC to the minimal degree of polynomials in the refutation. This is similar in spirit to the size-degree method known for propositional PC [37], albeit the actual relation is different and includes the quantifier depth of the QBF. Technically, the result is shown via the degree-preserving transfer from $\mathcal{Q}$-PC to PDLs and back explained above, together with an additional size-degree relation that we show for PDLs. The technique is similar to a prior size-width result for Q-Resolution [11].

C. New lower bounds for $\mathcal{Q}$-PC. Having both the PDL characterisation and size-degree relation at hand opens the door to new lower bounds for degree and size in $\mathcal{Q}$-PC.

Specifically, we show that the parity and more generally the modulo k functions mod_n^k on n variables as well as the majority function maj_n all require high-degree PDLs over all subfields of $\mathbb{C}$. Using a general construction from [13, 14] we can turn any Boolean function f into a QBF $\mathcal{Q}$-f that has f as its only countermodel. Together with our results above this implies that the $\mathcal{Q}$-mod_n^k and $\mathcal{Q}$-maj_n QBFs require both linear degree and exponential monomial size in $\mathcal{Q}$-PC.

In addition to using the size-degree method to prove lower bounds for PDLs and hence for $\mathcal{Q}$-PC proofs, we show that for finite fields of characteristic p, PDLs can be efficiently transformed into $\mathsf{AC}^0[p]$ circuits. This allows to directly transfer circuit lower bounds of [47] into $\mathcal{Q}$-PC proof lower bounds. As a result, either if F and G are both finite fields of different characteristics, or if F is finite and G is a subfield of $\mathbb{C}$, then the systems $\mathcal{Q}$-PC over F and G are incomparable.

In fact, all our lower bounds are very strong as they apply to a succinct model of QBF proof systems were propositional sub-derivations – for PC comprised of additions and multiplications of polynomials – can be abbreviated as semantic entailment steps that are checked with an NP oracle [17]. This implies that all our lower bounds and incomparability results also hold in the traditional proof model with "unfolded" computations, but remain even valid in the mentioned stronger NP oracle model.

Due to space constraints, some proofs are omitted.

2 Preliminaries

We assume familiarity with basic notions from computational complexity, cf. [1], as well as from logic, cf. [39], and algebra, cf. [41], but define all specific concepts needed in this paper.

We consider propositional formulas φ built from constants $0, 1$, connectors $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and propositional variables. A literal is a variable v or its negation $\overline{v}$. A clause is a disjunction of literals, and a formula is in Conjunctive Normal Form (CNF) if it is a conjunction of

clauses. When V is a set of variables, a (partial) assignment to V is a (partial) function $\alpha : V \to \{0,1\}$. We write $\langle V \rangle$ for the set of all complete assignments to V, and $\mathrm{vars}(\varphi)$ or $\mathrm{vars}(\alpha)$ for the set of all variables occurring in φ or α. For $V' \subseteq V$, we denote by $\alpha|_{V'}$ the restriction of α to the variables in V'. We denote by $\varphi[\alpha]$ the formula φ where each variable $v \in \mathrm{vars}(\alpha)$ has been substituted by $\alpha(v)$, and by $\phi[v_1/\theta_1, \ldots, v_k/\theta_k]$ the formula φ where variables v_i have been substituted by formulas θ_i.

Circuit classes. We recall the definitions of standard circuit classes used in this paper. The class AC^0 contains all languages computable by polynomial-size circuits with gates $\neg$, $\vee$, $\wedge$ with bounded depth and unbounded fan-in. The class $\mathsf{AC}^0[p]$ additionally uses MOD_p gates determining whether the sum of the inputs is 0 modulo p. $\mathsf{P/poly}$ uses circuits of polynomial size but arbitrary depth. For an in-depth account on circuit complexity we refer to [48].

Proof systems. We refer to [28] for a formal definition of proof systems and only illuminate certain restrictions. We only consider *line-based proof systems* where proofs consist of a sequence of lines (in our case polynomials) that are derived with certain rules. A propositional *base system* is a line-based proof system with certain very natural restrictions, formally defined in [10]. All propositional proof systems discussed in this paper are base systems.

Polynomial Calculus. Polynomial Calculus (PC) [25] is an algebraic proof system showing that a set of polynomials does not have common roots. For variables $V = \{v_1, \ldots, v_n\}$ over a field F, its lines are polynomial equations $0 = \sum_{i=1}^{m} c_i \prod_{v \in V_i} v$ with $m \in \mathbb{N}, c_i \in F, V_i \subseteq V$. A PC *refutation* starts with a set of polynomials, derives linear combinations of previous polynomials, and ends with the contradiction $0 = 1$. The size of a polynomial is its number of monomials, and the size of a PC refutation is the sum of the sizes of its polynomials.

Here, we view PC as a propositional proof system, and allow only the values 0 and 1 for each variable. This is ensured by including the Boolean axioms $v^2 - v = 0$ for each $v \in V$. In order to represent literals and clauses compactly, we introduce *complementary variables* $\overline{v}$ that are required to have the value $1 - v$, and introduce axioms $v + \overline{v} - 1 = 0$.

Perhaps counterintuitively, a variable that is 0 in a polynomial corresponds to a propositional variable that is true, and 1 corresponds to false. We recall that a polynomial equation is true if its polynomial equals 0. This way, a monomial corresponds to a clause containing the same literals, and a CNF can be efficiently encoded as a set of monomial equations.

When p is a polynomial, v a variable, and $c \in \{0,1\}$, we denote by $p[v/c]$ the polynomial p where v has been replaced by c and $\overline{v}$ by $1 - c$.

Quantified Boolean Formulas. A (closed prenex) *Quantified Boolean Formula* (QBF) is a formula $Q\,\phi$ where ϕ is a propositional formula and Q is the quantifier prefix that quantifies all variables v in ϕ either existentially as $\exists v$ or universally as $\forall v$. We typically use x_i for existentially quantified variables and u_i for universally quantified variables. When a system includes complementary variables $\overline{v}$ as in PC, those do not occur in the quantifier prefix. Their values are determined by the corresponding variables v instead.

Whether a QBF is true or not can be defined recursively. The formula $\forall u\, Q\, \varphi$ is true if both $Q\, \varphi[u = 0]$ and $Q\, \varphi[u = 1]$ are true. The formula $\exists x\, Q\, \varphi$ is true if at least one of $Q\, \varphi[x = 0]$ and $Q\, \varphi[x = 1]$ is true. When a QBF proof system is derived from an algebraic system such as PC, it nonetheless has these Boolean semantics.

In a fully quantified prenex QBF, the quantifier prefix determines a total order of the variables. Given a variable v, we will sometimes refer to the variables preceding v in the prefix as variables *left* of v; analogously we speak of the variables *right* of v.

A QBF $Q_1x_1 \cdots Q_kx_k\,\phi$ can be seen as a game between two players: *universal* ($\forall$) and *existential* ($\exists$). In the i-th step of the game, the player Q_i assigns a value to the variable x_i. The existential player wins if ϕ evaluates to 1 under the assignment constructed in the game. The universal player wins if ϕ evaluates to 0. Given a universal variable u with index i, a *strategy for* u is a function from all variables of index $< i$ to $\{0,1\}$. A QBF is false if and only if there exists a *winning strategy* for the universal player, that is if the universal player has a strategy for all universal variables that wins any possible game [1, 33].

3 A general characterisation of $\mathcal{Q}$-P proof size by decision lists

We start by characterising proof size in $\mathcal{Q}$-PC by a suitable circuit model. In fact, we will show a more general result that applies to a class of QBF proof systems which are lifted from a propositional base system P to the QBF system $\mathcal{Q}$-P. This lifting is done by adding the $\forall$red rule to the rules of P. The $\forall$red rule allows to derive $l[\mu]$ from a line l and a propositional assignment μ to universal variables, as long as $\mathrm{vars}(\mu)$ occur after all existential variables in l in the quantifier prefix [13].

However, lower bounds for the resulting $P + \forall$red system are trivial to obtain from lower bounds for P by existentially quantifying all variables. We are not too interested in such bounds, but in "genuine" QBF lower bounds that arise from quantifier alternation (cf. [17, 24] for a longer discussion and details). In other words, we want to filter out any propositional hardness in a QBF by ignoring purely propositional sub-derivations. For this, we introduce the Sem rule for semantic steps. It can derive a line l from a line r if $r \models l$. In general this inference step cannot be checked efficiently, but needs an NP oracle call [17]. Using Sem steps also removes the need for any other propositional inference rules as these can be carried out by Sem. We call the resulting system $\mathcal{Q}$-P.

▶ **Definition 1** ($\mathcal{Q}$-P). *Let P be a propositional base system. The system $\mathcal{Q}$-P is a refutational proof system for QBF that has the same lines as P, and rules* $\forall$red *and* Sem.

The system we are most interested in is $\mathcal{Q}$-PC where the lines are polynomials. We briefly review the semantics of this system. As specified in Definition 1, its only rules are $\forall$red and Sem. Its lines are polynomial equations with coefficients from a field F. The $\forall$red rule can be applied to a polynomial p to obtain $p[u/0]$ (which is p with variable u set to 0 and $\overline{u}$ set to 1) or $p[u/1]$ (which is p with variable u set to 1 and $\overline{u}$ set to 0) as long as u is quantified universally right of all existential variables in p. This also means that u cannot be a complementary variable $\overline{v}$. The Sem rule allows to derive polynomial equations that semantically follow from previous equations, the Boolean axioms, and the $v + \overline{v} = 1$ axioms. Semantically, the variables can only take the values 0 or 1, and $\overline{v}$ must always take the value $1 - v$. The propositional rules of PC can be added, but are not strictly needed and do not shorten proofs in the presence of Sem.

We now define the circuit model that we will use for the characterisation of $\mathcal{Q}$-P proof size. The model is a variant of decision lists [10, 45].

▶ **Definition 2** (P-UDL). *Let P be a base system and X, U sets of variables. A* P-UDL *of length k is a sequence $L = (p_1, \mu_1), (p_2, \mu_2), \ldots, (p_k, \mu_k)$ where $(\overline{p_i})$ is a line of P,* $\mathrm{vars}(p_i) \subseteq X$, *$\mu_i \in \langle U \rangle$ for each $i \in [k]$ and $p_k = \top$. It computes a function $f_L : \langle X \rangle \to \langle U \rangle$ with $f_L(\alpha) = \mu_j$ for the smallest j such that $\alpha \models p_j$.*

Intuitively, a P-UDL checks conditions p_j, which are negations of lines of P using only existential variables and outputs a full assignment μ_j to the universal variables.

Again the main instantiation of this definition for us is to choose P as PC. To ease notation we abbreviate PC-UDL by PDL for *p*olynomial *d*ecision *l*ists. The lines of PDLs are then polynomials. As lines in PC are polynomials p with the implicit meaning of $p = 0$, conditions in a PDL check that the polynomial is *not* zero. Hence a line p in a PDL becomes active, if the p does not evaluate to 0 under the assignment.

We use P-UDLs to compute countermodels for QBFs. More formally, we say that a P-UDL L is *correct* for a QBF $\mathcal{Q}.\varphi$ if it has all of the following properties:

- All variables in X are existential variables of $\mathcal{Q}$.
- All universal variables of $\mathcal{Q}$ are in U.
- Let $\alpha, \beta \in \langle X \rangle, u \in U$. If $f_L(\alpha)$ and $f_L(\beta)$ disagree on u, then α and β disagree on a variable $x \in X$ that occurs before u in $\mathcal{Q}$.
- For every $\alpha \in \langle X \rangle$, $\alpha \wedge f_L(\alpha)$ falsifies φ.

For P-UDL $L = (p_1, \mu_1), \ldots, (p_k \mu_k)$, we define the *size* of L, $|L| = \sum_{i=1}^{k} size_P(\overline{p_i})$, where $size_P$ corresponds to the respective size measure in the base system P. Additionally, the length of L is defined as $\text{len}(L) = k$.

Our goal is to use P-UDLs to characterise the hardness of $\mathcal{Q}$-P proofs on QBFs of constant alternation depth, i.e. to show that there exists a short P-UDL for a QBF Φ if and only if there exists a short $\mathcal{Q}$-P proof of Φ.

From the definition of P-UDL, it is apparent that the size of P-UDL for a QBF Φ is always at least as big as the size of the smallest countermodel of Φ, since each line always assigns all universal variables. As such, P-UDL cannot possibly characterise proof size in $\mathcal{Q}$-P for all base systems P. In fact, we need three restrictions on $\mathcal{Q}$-P for the characterisation to work. Firstly, the negations of the lines of P must allow to succinctly represent assignments to variables. Secondly, the base system P must be *closed under disjunction*, i.e. for two arbitrary lines l and p of P, the disjunction $l \vee p$ is also a valid line in P with size $\mathcal{O}(|l| \cdot |p|)$.

Thirdly, we require $\mathcal{Q}$-P to have limited *capacity*. Capacity is a measure introduced in [10], which counts how many different answers the universal player needs at most to respond to one line in a $\mathcal{Q}$-P proof π. In particular, we require that the capacity of $\mathcal{Q}$-P is at most polynomial in the size of π for all proofs π. For the formal definition of capacity, we refer to [10]. We remark that $\mathcal{Q}$-PC and $\mathcal{Q}$-Res (as well as the QBF cutting planes system $\mathcal{Q}$-CP) all have bounded capacity [10].

We are now ready to characterise proof size in $\mathcal{Q}$-P by the size of P-UDLs.

▶ **Theorem 3.** *Let P be a line-based, propositional proof system, where the lines are closed under disjunction, such that $\mathcal{Q}$-P has at most polynomial capacity. Then for QBFs Φ with bounded quantifier alternation depth, we can efficiently transform a P-UDL for Φ into a $\mathcal{Q}$-P refutation for Φ and vice versa. In particular, the minimal size of a P-UDL for Φ is polynomially equivalent to the size of the minimal $\mathcal{Q}$-P refutation for Φ.*

Proof sketch. The proof outline follows [11]. *From $\mathcal{Q}$-P to P-UDL.* Let a QBF Φ of alternation depth d and a $\mathcal{Q}$-P refutation π of Φ be given. Using bounded capacity, we can show that w.l.o.g. π always uses universal reductions on entire blocks of universal variables. Employing the paradigm of strategy extraction [3, 11, 30], we can extract a set of P-decision lists computing a countermodel of Φ from π. Each of these decision lists computes the universal strategy for one of the universal blocks and has size at most $|\pi|$. We combine these decision lists to a single decision list, whose size is at most the product of the sizes of the original lists, using the *direct product* construction from [11]. The resulting decision list computes the correct function and is a P-UDL. Since there are d decision lists of size $|\pi|$, the computed P-UDL has size $\mathcal{O}(|\pi|^d)$.

From P-UDL to $\mathcal{Q}$-P. Let a QBF Φ of alternation depth d and a P-UDL L computing a countermodel of Φ be given. We define the *entailment sequence* $\mathcal{E}$ of L recursively in d.

- if $d = 1$, $\mathcal{E}(L) := \overline{\varepsilon_1} \vee \overline{\mu_1}, \ldots, \overline{\varepsilon_s} \vee \overline{\mu_s}$
- if $d \geq 2$, for each $i \in [s]$ define L_i as the list obtained from L by replacing the first $i-1$ existential terms by their X_1 components and setting all U_1 components to $\mu_i|_{U_1}$. $\mathcal{E}(L)$ is the sequence $\pi_1, \ldots, \pi_s$, where $\pi_i := (\overline{\varepsilon_i}|_{X_1} \vee \overline{\mu_i}|_{U_1}) \otimes \mathcal{E}(L_i[\alpha])$ and α is some arbitrary but fixed assignment on $X_1 \cup U_1$ satisfying $\varepsilon_i|_{X_1} \wedge \mu_i|_{U_1}$.

Here, the $\otimes$-operator between a line of P and the entailment sequence defines an elementwise disjunction between the line and each element in the entailment sequence, i.e. $\ell \otimes (c_1, c_2, \ldots, c_r) = (\ell \vee c_1, \ell \vee c_2, \ldots, \ell \vee c_r)$. For a line ℓ, we call $\mathrm{red}(\ell)$ the line obtained by using universal reduction on ℓ by some specific universal assignment. Since negations of lines of P can succinctly represent assignments, $\overline{\mu_i}$ can be represented in P. If $\mathcal{E}(L) = (c_1, \ldots, c_r)$, then $(c_1, \mathrm{red}(c_1), c_2, \mathrm{red}(c_2), \ldots, c_r, \mathrm{red}(c_r))$ is a $\mathcal{Q}$-P refutation of Φ. ◀

This reproves a characterisation of QBF resolution by decision lists from [11]. The main application for us is polynomial calculus (PC) for which this result is new. PC has a capacity of $\sqrt{n}$, where n is the size of the proof [10]. Additionally, PC is closed under disjunctions as the disjunction of two lines can be expressed as the product of the respective polynomials. The size of the disjunction is the product of the sizes of the original lines.

▶ **Corollary 4.** *For QBFs of bounded quantifier depth, the minimal size of $\mathcal{Q}$-PC proofs and the minimal PDL sizes are polynomially equivalent.*

4 Size-degree bounds for polynomial calculus in QBF

Using the connection between $\mathcal{Q}$-PC and PDLs from Corollary 4, we now aim to show lower bounds for $\mathcal{Q}$-PC by proving lower bounds for PDLs. The latter task will be simplified by a relation between PDL size and PDL degree, which is measured as the maximal degree of the polynomial conditions in the PDL. As these polynomials are just defined in existential variables, it makes sense to define the *existential degree* for PDLs and $\mathcal{Q}$-PC refutations. This is in line with an analogous definition of existential width for Q-Resolution [11, 15].

▶ **Definition 5** (Existential degree of a $\mathcal{Q}$-PC refutation). *Let $f = \exists X_1 \forall U_1 \cdots \exists X_{d+1}.\varphi$, $\pi = (\pi_1, \ldots, \pi_s)$ be a $\mathcal{Q}$-PC refutation of f and $X = \bigcup_{i=1}^{d} X_i$. The* existential degree $\deg_\exists$ *of f is defined as* $\deg_\exists(\pi) = \max_{i \in [s]} \deg(\pi_i|_X)$.

Analogously, the degree of a PDL L is defined as the maximal degree of all queries in L. We can show a size-degree relation for PDLs.

▶ **Theorem 6.** *Let f be a multi-output Boolean function with n input variables. If f is computed by a PDL of size s, it is also computed by a PDL of degree $O(\sqrt{n \log s})$.*

In essence, the proof of this theorem is the same as the original size-degree result for propositional polynomial calculus by Impagliazzo, Pudlák and Sgall [37] (cf. also [7, 22] for similar arguments). Equivalently, a function f that can only be computed by PDLs of degree at least k needs PDLs of size $\exp(\Omega(\frac{k^2}{n}))$.

We can use this size-degree relation for PDLs to obtain a size-degree relation on $\mathcal{Q}$-PC.

▶ **Theorem 7.** *Let f be a QBF with n variables of alternation depth d such that every $\mathcal{Q}$-PC proof has existential degree at least k. Then every $\mathcal{Q}$-PC proof has size at least* $\exp(\Omega(\frac{k^2}{d^3 n}))$.

Proof. Let $f = \exists X_1 \forall U_1 \cdots \exists X_{d+1}.\varphi$, $X = \bigcup_{i=1}^{d} X_i$ the set of existential variables except the last block and $|X| = v$. Additionally, let π be a shortest $\mathcal{Q}$-PC refutation of f. Using Corollary 4, π can be transformed into a PDL L of size at most $|\pi|^d$. With Theorem 6, L can then be transformed into a PDL M with degree at most $\mathcal{O}(\sqrt{dv \log|\pi|})$. Transforming M back into a $\mathcal{Q}$-PC proof with Corollary 4 results in a proof π' with degree at most $k = \mathcal{O}(d\sqrt{dv \log|\pi|})$. Solving this equation for $|\pi|$ yields the result. ◀

The proof is similar to Theorem 6.2 in [11]. In contrast to the size-degree relation from [37], Theorem 7 includes the quantifier depth of the QBF, but not the initial degree of the QBF.

In the rest of this section, we will explore specific lower bounds for the degree of PDLs. We will first show degree lower bounds for PDLs computing specific functions and then turn this into $\mathcal{Q}$-PC size lower bounds for related QBFs.

4.1 Parity and mod functions require PDLs of high degree

Let $\mathrm{par}_n(x_1, \ldots, x_n) = \bigoplus_{i=1}^{n} x_n$ be the parity function. Lower bounds for this function only hold for PDLs over certain fields; in fact, par_n is just the sum of input variables in fields of characteristic 2, and is therefore trivial for PDLs over those fields. However, it seems to be hard in fields of characteristic 0. We prove a lower bound for $\mathbb{C}$ and its subfields.

▶ **Proposition 8.** *A PDL with polynomials over a subfield of* $\mathbb{C}$ *computing* par_n *has degree at least* $\frac{n}{2}$.

Proof. We consider the first line of the PDL and assume without loss of generality that it has output 1.[1]

Let p be the first line's polynomial and d its degree. Let $X = \{x_1, \ldots, x_n\}$. We can assume that there is a $w \in \langle X \rangle$ with $p(w) \neq 0$ or we could omit the first line. However, to avoid giving any wrong answers, for every $a \in \langle X \rangle$ with $\mathrm{par}_n(a) = 0$, it must hold that $p(a) = 0$.

We compute the complex conjugate p^* by conjugating every coefficient. Because we only evaluate the polynomials on real numbers (specifically 0 and 1), we know that $p^*(a) = (p(a))^*$ for every $a \in \langle X \rangle$. We define $q := p \cdot p^*$ and note that $\deg(q) \leq 2d$ and for all $a \in \langle X \rangle$, $q(a) = p(a) \cdot p(a)^* \in \mathbb{R}^{\geq 0}$.

For a polynomial r, define the function

$$s(r) := \sum_{\substack{a \in \langle X \rangle \\ \mathrm{par}(a)=1}} r(a) - \sum_{\substack{a \in \langle X \rangle \\ \mathrm{par}(a)=0}} r(a)$$

which is linear with respect to r. If r is a monomial that does not contain the variable x,

$$\begin{aligned} s(r) &= \sum_{\substack{a \in \langle X \setminus \{x\} \rangle \\ b \in \langle \{x\} \rangle \\ \mathrm{par}(a,b)=1}} r(a,b) - \sum_{\substack{a \in \langle X \setminus \{x\} \rangle \\ b \in \langle \{x\} \rangle \\ \mathrm{par}(a,b)=0}} r(a,b) = \sum_{\substack{a \in \langle X \setminus \{x\} \rangle \\ b \in \langle \{x\} \rangle \\ \mathrm{par}(a,b)=1}} r(a) - \sum_{\substack{a \in \langle X \setminus \{x\} \rangle \\ b \in \langle \{x\} \rangle \\ \mathrm{par}(a,b)=0}} r(a) \\ &= \left(\sum_{\substack{a \in \langle X \setminus \{x\} \rangle \\ \mathrm{par}(a)=1}} r(a) + \sum_{\substack{a \in \langle X \setminus \{x\} \rangle \\ \mathrm{par}(a)=0}} r(a) \right) - \left(\sum_{\substack{a \in \langle X \setminus \{x\} \rangle \\ \mathrm{par}(a)=0}} r(a) + \sum_{\substack{a \in \langle X \setminus \{x\} \rangle \\ \mathrm{par}(a)=1}} r(a) \right) \\ &= 0. \end{aligned}$$

[1] If it has output 0, we can invert all outputs in the PDL and replace every occurence of x_1 with $1 - x_1$. This does not change its degree, and the resulting PDL computes $\neg \mathrm{par}_n(\overline{x_1}, x_2, \ldots, x_n) = \mathrm{par}_n(x_1, \ldots, x_n)$.

Because s is linear, $s(r) = 0$ also holds for any polynomial r of degree $< n$.

To obtain a value of $s(q)$, we use that

$$\sum_{\substack{a\in\langle X\rangle \\ \mathrm{par}(a)=1}} q(a) > 0 \quad \text{and} \quad \sum_{\substack{a\in\langle X\rangle \\ \mathrm{par}(a)=0}} q(a) = 0$$

(in the left equation, none of the summands are negative; one of them is $q(w) = |p(w)|^2 > 0$). Consequently, $s(q) > 0$ and $\deg(q) \geq n$, so using $\deg(q) \leq 2d$ we conclude that $d \geq \frac{n}{2}$. ◀

To turn this lower bound into a $\mathcal{Q}$-PC bound we need QBFs based on the parity function. For this we describe a general transformation originating from [13,14] to construct QBFs that are false, but force the universal player to use a unique strategy by computing a particular function f. We define the QBF $\mathcal{Q}$-f:

▶ **Definition 9** ($\mathcal{Q}$-f). *Let $f : \langle X\rangle \to \langle U\rangle$ be a function that is computed by a* P/poly *circuit C. Then $\mathcal{Q}$-$f := \exists X \forall U \exists T\, \varphi$ where φ is the Tseitin transformation of the circuit $U \neq C(X)$, and T is the corresponding set of auxiliary Tseitin variables.*

In our case above, f is par_n and C is a simple P/poly circuit for par_n. The existential player wins if and only if the circuit $U \neq C(X)$ yields true, after the assignments to X and U were chosen by the respective players. The universal player therefore has the unique winning strategy of playing $U = f(X)$.

From Proposition 8 together with the size-degree relation for PDLs (Theorem 6) and the efficient transformation into $\mathcal{Q}$-PC (Theorem 3) we obtain:

▶ **Corollary 10.** *$\mathcal{Q}$-PC refutations over a subfield of $\mathbb{C}$ of $\mathcal{Q}$-par_n require size $\exp(\Omega(n))$.*

We can generalise this to the modulo k functions. Let $\mathrm{mod}_n^k(x_1, \ldots, x_n) = 1$ if and only if $\sum_{i=1}^n x_i \equiv 0 \mod k$ (otherwise it is 0). With a similar, but somewhat more technical proof than for Proposition 8 we can show:

▶ **Proposition 11.** *A PDL with polynomials over a subfield of $\mathbb{C}$ computing mod_n^k has degree at least $\frac{1}{2}\left\lfloor \frac{n}{(k-1)} \right\rfloor$.*

Consequently, the $\mathcal{Q}$-mod_n^k QBFs are hard for $\mathcal{Q}$-PC over $\mathbb{C}$.

▶ **Corollary 12.** *$\mathcal{Q}$-PC refutations over a subfield of $\mathbb{C}$ of $\mathcal{Q}$-mod_n^k have size $\exp(\Omega(\frac{n}{k}))$.*

4.2 Majority PDLs have high degree

Next we want to show degree lower bounds for PDLs that compute the majority functions $\mathrm{maj}_n(x_1, \ldots, x_n)$, which evaluate to one, if and only if $\sum_{i=1}^n x_i \geq \frac{n}{2}$. In contrast to the par_n and mod_n^k functions, this lower bound does not depend on the underlying field. We start by proving a useful lemma about PDLs:

▶ **Lemma 13.** *Let $X = \{x_1, \ldots, x_n\}$, $\alpha \in \langle X\rangle$ a complete assignment and p a polynomial with variables in X that is not the constant 0. There is an assignment $\beta \in \langle X\rangle$ such that α and β only differ in at most $\deg(p)$ variables, and $p(\beta) \neq 0$.*

Proof. We start by taking p and constructing an equivalent polynomial q that contains no variables according to their polarity in α: when $\alpha(x) = 0$, we replace $\overline{x}$ by $1 - x$, and when $\alpha(x) = 1$, we replace x by $1 - \overline{x}$. Let m be one of the lowest-degree monomials in q, and β the assignment that satisfies every literal in m, and assigns every other variable

according to α. Note that α and β only differ in at most $\deg(m) \leq \deg(q) = \deg(p)$ variables. Because m has minimal degree, every other monomial in q contains a variable $v \notin \text{vars}(m)$, so $\beta(v) = \alpha(v)$. Because v only occurs in the polarity opposite of $\alpha(v)$, β does not satisfy that other monomial. Therefore, $0 \neq m(\beta) = q(\beta) = p(\beta)$. ◀

With this, we can show the lower bound fairly easily:

▶ **Proposition 14.** *A PDL computing* $\text{maj}_n(x_1, \ldots, x_n)$ *has degree at least* $\frac{n}{2}$.

Proof. Let L be such a PDL, d its degree, and p its first polynomial. We also define $X := \{x_1, \ldots, x_n\}$. If the first line of L has output 1, let α be the assignment that assigns all $x_i = 0$, and apply Lemma 13. Because $p(\beta) \neq 0$ and L is correct, we obtain $\text{maj}_n(\beta) = 1$. So β has to assign 1 to at least $\frac{n}{2}$ variables and thus differs from α in at least $\frac{n}{2}$ variables. Therefore, $\frac{n}{2} \leq \deg(p) \leq d$. If the first line has output 0 instead, let α be the assignment that assigns all $x_i = 1$. Lemma 13 yields an assignment β with $p(\beta) \neq 0$, so $\text{maj}_n(\beta) = 0$ and β has to assign 0 to at least $\frac{n}{2}$ variables. Therefore, $\frac{n}{2} \leq \deg(p) \leq d$. ◀

▶ **Corollary 15.** *$\mathcal{Q}$-PC refutations of $\mathcal{Q}$-maj_n have size* $\exp(\Omega(n))$.

4.3 Limits of the size-degree method

While the size-degree technique is useful for showing several lower bounds as demonstrated above, it does not capture all nuances of PDL size. We will illustrate this by constructing two functions, each having n^2 variables and requiring PDLs of degree n. For these values, Theorem 6 does not yield any useful lower bound. Indeed, one of the functions requires PDLs of size $\exp(\Omega(n))$, while for the other one, size $O(n)$ is sufficient.

▶ **Theorem 16.** *Given a function f, the minimal size of its PDLs is not solely determined by its number of variables and minimal PDL degree. In particular, there are families of functions f_n and g_n, each with n^2 input variables and minimal PDL degree n, such that f_n has PDLs of size $O(n)$ and g_n requires PDLs of size* $\exp(\Omega(n))$.

In order to prove this, we introduce examples for those functions g and f. We define the square-majority function as $\text{sqm}_n(x_1, \ldots, x_{n^2}) = 1$ if and only if $\sum_{i=1}^{n^2} x_i \geq n$, otherwise 0.

▶ **Lemma 17.** *The sqm_n function can be computed by PDLs of degree n, but not by PDLs of smaller degree. It requires PDLs of size* $\exp(\Omega(n))$.

The lower bound on the size of PDLs for sqm already cannot be shown by Theorem 6. This raises the question of whether our size-degree relation is simply too weak, and whether we could obtain a stronger one that can capture the complexity of the sqm function. It turns out that this is not the case: there are functions with the same degree and number of variables, but smaller decision lists. The complexity of the sqm function cannot be derived from its degree and number of variables alone.

To obtain such a function, let $X = \{x_i^j \mid i, j \in \{1, \ldots, n\}\}$ and $f_n(X) := \bigvee_{i=1}^{n} \bigwedge_{j=1}^{n} x_i^j$.

▶ **Lemma 18.** *The function f_n defined above can be computed by PDLs of degree n and size $O(n)$, but not by PDLs of smaller degree.*

Proof of Theorem 16. Let f_n be the function defined above, and $g_n = \text{sqm}_n$. The statement follows directly from Lemmas 17 and 18. ◀

5 Converting PDLs into Boolean circuits

We now establish a different lower bound method for $\mathcal{Q}$-PC that directly imports circuit lower bounds. For this we show a connection between PDLs over finite fields and $AC^0[p]$ circuits. First we convert PDLs over a finite field to PDLs over a prime-order finite field.

▶ **Lemma 19.** *Let p be a prime, $k, l, m \in \mathbb{N}^{\geq 1}$ and L a PDL of length l and size m over the finite field $\mathbb{F}_{p^k}$. Then L can be converted into a PDL of length $k \cdot l$ and size $k \cdot m$ over the finite field $\mathbb{F}_p$ that computes the same function.*

This lemma is based on the fact that $\mathbb{F}_{p^k}$ has the structure of a k-dimensional $\mathbb{F}_p$ vector space with respect to addition. An equation in $\mathbb{F}_{p^k}$ that does not use multiplication can therefore be split up into k equations in $\mathbb{F}_p$. The only multiplication that occurs in the polynomials of a PDL is between coefficients and the corresponding unit monomials, and those unit monomials can only be 1 or 0. We can use this to convert each line of a PDL over $\mathbb{F}_{p^k}$ into k lines over $\mathbb{F}_p$.

We will now efficiently convert PDLs over $\mathbb{F}_p$ for primes p into $AC^0[p]$ circuits.

▶ **Proposition 20.** *Let $f : \{0,1\}^n \to \{0,1\}^s$ be a function that is computed by a PDL of size m over a finite field $\mathbb{F}_p$. Then f can be computed by an $\mathsf{AC}^0[p]$ circuit of depth 6 with only $O(pm + s)$ AND or OR gates.*

Proof. We construct the circuit iteratively starting at the inputs and ending at the output. At each layer, we describe the semantics of the newly-added circuit gates.

- Start with v input gates for the variables.
- Add v NOT gates, each negating one of the variables. All literals are now represented in the circuit.
- Consider each monomial $c \cdot \prod_i t_i$ where $c \in \mathbb{F}_p$ and the t_i are literals. Add c identical AND gates with all the t_i as inputs. The sum of the outputs of these gates will be equivalent to the value of the monomial. Because each $c < p$, the total number of these gates is smaller than pm.
- For each line, add a MOD_p gate over all the AND gates of its monomials. The sum of its inputs will be equivalent to the sum of the values of its monomials, so its output indicates whether the polynomial equation holds in $\mathbb{F}_p$.
- For each line, add a NOT gate negating the MOD_p gate.
- For each line, add an AND gate that checks if all polynomial equations of previous lines hold, but the equation of the current line does not. Its output indicates whether this line's output value is active.
- Finally, for each output variable z add an OR gate connecting the AND gates of those lines whose output sets $z = 1$. Its output indicates whether the PDL sets $z = 1$, it is the output of the circuit for variable z.

This circuit has depth 6 and at most $O(pm + s)$ AND or OR gates. ◀

Using Lemma 19 we can extend this result to fields $\mathbb{F}_{p^k}$:

▶ **Corollary 21.** *Let $f : \{0,1\}^n \to \{0,1\}^s$ be a function that is computed by a PDL of size m over a finite field $\mathbb{F}_{p^k}$. Then f can be computed by an $\mathsf{AC}^0[p]$ circuit of depth 6 with only $O(pmk + s)$ AND or OR gates.*

This corollary allows us to lift lower bounds for $\mathsf{AC}^0[p]$ circuits to PDLs over finite fields. We cite such a circuit lower bound by Smolensky [47]. Let p be a prime and r not a power of p. An $\mathsf{AC}^0[p]$ circuit of depth k that computes mod_n^r requires $\exp(\Omega(n^{\frac{1}{2k}}))$ AND or OR gates. Using Corollary 21 we can apply this result to PDLs:

▶ **Corollary 22.** *Let p, r be distinct primes. A PDL over the finite field $\mathbb{F}_{p^k}$ that computes mod_n^r needs size $\exp(\Omega(\sqrt[12]{n} - \log(pk)))$.*

We also want to apply these results to another class of functions that we call balance. $\mathrm{balance}_{2n}(x_1, x_2, \ldots, x_{2n}) = 1$ if and only if $\sum x_i = n$, else it is zero.

We now show that PDLs for the balance function can be transformed into $AC^0[p]$ circuits for the function mod_n^q with arbitrary q.

▶ **Proposition 23.** *If there is a PDL of size m over $\mathbb{F}_p$ that computes $\mathrm{balance}_{2n}$, then mod_n^q can be computed by an $\mathsf{AC}^0[p]$ circuit of depth 8 that uses only $O(pnm)$ AND or OR gates.*

Proof. Let $R = \{q \cdot i \mid i \in \mathbb{Z}, 0 \leq q \cdot i \leq n\} = q\mathbb{Z} \cap [0; n]$, and note that $|R| = \left\lceil \frac{n+1}{q} \right\rceil$. When exactly k of the variables $x_1, \ldots, x_n$ are true, then $\mathrm{mod}_n^q(x_1 \ldots, x_n) = 1$ if and only if $k \in R$.

Using Proposition 20 we can obtain a circuit that computes $\mathrm{balance}_{2n}$ with $pm + l$ AND gates and one OR gate. We instantiate this circuit once for each $k \in R$, replacing the $2n$ inputs with n actual inputs $x_1, \ldots, x_n$, as well as k constants 0 and $n - k$ constants 1. The output of such an instance is 1 if and only if k of the actual inputs are 1. We use a single OR gate over all these outputs to obtain $\mathrm{mod}_n^q(x_1 \ldots, x_n)$. The total number of AND or OR gates is $|R|(pm + l) + 1 = \left\lceil \frac{n+1}{q} \right\rceil (pm + l) + 1 \leq (n + 1)(p + 1)m + 1$. ◀

▶ **Corollary 24.** *The balance formulas require exponential-sized PDLs over finite fields.*

We can now show that many of the systems $\mathcal{Q}$-PC over different fields are incomparable.

▶ **Theorem 25.** *Let F be a finite field, and G either a finite field with different characteristic, or a subfield of $\mathbb{C}$. Then the $\mathcal{Q}$-PC systems over F and G are incomparable.*

Proof. Because PC systems and PDLs are polynomially equivalent, we can use exponential separations between the respective PDLs to obtain the result. Let p and q be the characteristics of F and G, respectively. The function mod_n^p is trivial for PDLs over F and requires only linear size. However, it requires exponential-sized PDLs over G, either due to Corollary 12 (if G is a finite field) or due to Corollary 22 (if G is a subfield of $\mathbb{C}$).

If G is a finite field, the function mod_n^q requires exponential-size PDLs over F and only linear-size PDLs over G. If G is a subfield of $\mathbb{C}$, the $\mathrm{balance}_n$ functions can be computed in linear size by PDLs over G, but require exponential-size PDLs over F by Corollary 24. ◀

6 Conclusion

While we concentrated on $\mathcal{Q}$-PC in this paper, it is interesting to explore the wider consequences of our P-UDL characterisation in Theorem 3 for further proof systems $\mathcal{Q}$-P. Besides PC and Res, the most studied base systems are arguably *Cutting Planes* (CP) and the various $\mathcal{C}$-Frege systems, where $\mathcal{C}$ is some circuit class, e.g. $\mathsf{AC}^0, \mathsf{NC}^1$ or $\mathsf{P/poly}$.

While $\mathcal{Q}$-CP has polynomial capacity (the capacity is 1) [10], the lines are not closed under disjunction. Hence we cannot use Theorem 3 to obtain a CP-UDL characterisation. We leave open, whether the result itself does not hold or if it only requires a different proof.

For $\mathcal{Q}$-$\mathcal{C}$-Frege, we cannot use Theorem 3 either. Here the lines are closed under disjunction, but the capacity is exponential. However, it is known that the circuit class $\mathcal{C}$ itself, which is a strictly stronger model than $\mathcal{C}$-UDL, tightly characterises $\mathcal{Q}$-$\mathcal{C}$-Frege [20]. As such, the equivalence between $\mathcal{C}$-UDLs and $\mathcal{Q}$-$\mathcal{C}$-Frege fails.

Interestingly, if we consider treelike $\mathcal{Q}$-$\mathcal{C}$-Frege systems, the $\mathcal{C}$-UDL characterisation does hold, even though the capacity is still superpolynomial. Intuitively, this could be explained by the fact that limited capacity is only required for blockwise reductions, and such reductions

can (possibly) be obtained by combining reductions along a path in the proof-tree. We prove the result even without using Theorem 3, as a direct proof is straightforward to obtain using a previous characterisation of treelike $\mathcal{Q}$-$\mathcal{C}$-Frege from [16]. We just state the result.

▶ **Theorem 26.** *For a circuit class $\mathcal{C}$ and a QBF family, the minimal sizes of $\mathcal{Q}$-$\mathcal{C}$-$\mathsf{Frege}_{\mathrm{tree}}$ proofs and the minimal $\mathcal{C}$-UDL sizes are polynomially bounded by each other.*

Interestingly, this result even holds for QBFs with unbounded quantifier alternation. We leave open whether similar characterisations can be obtained for daglike systems $\mathcal{Q}$-P on formulas with unbounded alternation depth.

References

1 Sanjeev Arora and Boaz Barak. *Computational Complexity – A Modern Approach*. Cambridge University Press, 2009. URL: http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264.

2 Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. Clause-learning algorithms with many restarts and bounded-width resolution. *J. Artif. Intell. Res.*, 40:353–373, 2011. doi:10.1613/jair.3152.

3 Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. *Form. Methods Syst. Des.*, 41(1):45–65, 2012. doi:10.1007/s10703-012-0152-6.

4 Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In *Proc. Theory and Applications of Satisfiability Testing (SAT)*, pages 154–169, 2014. doi:10.1007/978-3-319-09284-3_12.

5 Paul Beame, Henry A. Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. *J. Artif. Intell. Res.*, 22:319–351, 2004. doi:10.1613/jair.1410.

6 Paul Beame and Toniann Pitassi. Propositional proof complexity: Past, present, and future. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science: Entering the 21st Century*, pages 42–70. World Scientific Publishing, 2001.

7 Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001. doi:10.1145/375827.375835.

8 Olaf Beyersdorff. On the correspondence between arithmetic theories and propositional proof systems – a survey. *Mathematical Logic Quarterly*, 55(2):116–137, 2009. doi:10.1002/malq.200710069.

9 Olaf Beyersdorff. Proof complexity of quantified Boolean logic – a survey. In Marco Benini, Olaf Beyersdorff, Michael Rathjen, and Peter Schuster, editors, *Mathematics for Computation (M4C)*, pages 353–391. World Scientific, Singapore, 2022.

10 Olaf Beyersdorff, Joshua Blinkhorn, and Luke Hinde. Size, cost, and capacity: A semantic technique for hard random QBFs. *Logical Methods in Computer Science*, 15(1), 2019. doi:10.23638/LMCS-15(1:13)2019.

11 Olaf Beyersdorff, Joshua Blinkhorn, Meena Mahajan, and Tomás Peitl. Hardness characterisations and size-width lower bounds for QBF resolution. *ACM Trans. Comput. Log.*, 24(2):10:1–10:30, 2023. doi:10.1145/3565286.

12 Olaf Beyersdorff and Benjamin Böhm. Understanding the relative strength of QBF CDCL solvers and QBF resolution. *Log. Methods Comput. Sci.*, 19(2), 2023. doi:10.46298/lmcs-19(2:2)2023.

13 Olaf Beyersdorff, Ilario Bonacina, Leroy Chew, and Jan Pich. Frege systems for quantified Boolean logic. *J. ACM*, 67(2):9:1–9:36, 2020. doi:10.1145/3381881.

14 Olaf Beyersdorff, Leroy Chew, and Mikolás Janota. New resolution-based QBF calculi and their proof complexity. *ACM Transactions on Computation Theory*, 11(4):26:1–26:42, 2019. doi:10.1145/3352155.

15 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Are short proofs narrow? QBF resolution is *not* so simple. *ACM Trans. Comput. Log.*, 19(1):1:1–1:26, 2018. doi:10.1145/3157053.

16 Olaf Beyersdorff and Luke Hinde. Characterising tree-like Frege proofs for QBF. *Inf. Comput.*, 268, 2019. doi:10.1016/j.ic.2019.05.002.

17 Olaf Beyersdorff, Luke Hinde, and Ján Pich. Reasons for hardness in QBF proof systems. *ACM Transactions on Computation Theory*, 12(2):10:1–10:27, 2020. doi:10.1145/3378665.

18 Olaf Beyersdorff, Mikolás Janota, Florian Lonsing, and Martina Seidl. Quantified Boolean formulas. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications, pages 1177–1221. IOS Press, 2021. doi:10.3233/FAIA201015.

19 Olaf Beyersdorff and Oliver Kutz. Proof complexity of non-classical logics. In N. Bezhanishvili and V. Goranko, editors, *Lectures on Logic and Computation - ESSLLI 2010 /ESSLLI 2011, Selected Lecture Notes*, pages 1–54. Springer-Verlag, Berlin Heidelberg, 2012. doi:10.1007/978-3-642-31485-8_1.

20 Olaf Beyersdorff and Ján Pich. Understanding Gentzen and Frege systems for QBF. In *Proc. ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2016. doi:10.1145/2933575.2933597.

21 Benjamin Böhm, Tomás Peitl, and Olaf Beyersdorff. QCDCL with cube learning or pure literal elimination - what is best? In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1781–1787. ijcai.org, 2022. doi:10.24963/ijcai.2022/248.

22 Nader H. Bshouty. A subexponential exact learning algorithm for DNF using equivalence queries. *Inf. Process. Lett.*, 59(1):37–39, 1996. doi:10.1016/0020-0190(96)00077-4.

23 Sam Buss and Jakob Nordström. Proof complexity and SAT solving. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications, pages 233–350. IOS Press, 2021. doi:10.3233/FAIA200990.

24 Hubie Chen. Proof complexity modulo the polynomial hierarchy: Understanding alternation as a source of hardness. *ACM Transactions on Computation Theory*, 9(3):15:1–15:20, 2017. doi:10.1145/3087534.

25 Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proc. 28th ACM Symposium on Theory of Computing*, pages 174–183, 1996. doi:10.1145/237814.237860.

26 Jonas Conneryd, Susanna F. de Rezende, Jakob Nordström, Shuo Pang, and Kilian Risse. Graph colouring is hard on average for polynomial calculus and Nullstellensatz. In *64th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1–11. IEEE, 2023. doi:10.1109/FOCS57990.2023.00007.

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

28 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979. doi:10.2307/2273702.

29 Susanna F. de Rezende, Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, and Dmitry Sokolov. Automating algebraic proof systems is NP-hard. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 209–222. ACM, 2021. doi:10.1145/3406325.3451080.

30 Uwe Egly, Florian Lonsing, and Magdalena Widl. Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In *Proc. Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, pages 291–308, 2013. doi:10.1007/978-3-642-45221-5_21.

31 Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. *Theory Comput.*, 17:1–88, 2021. URL: https://theoryofcomputing.org/articles/v017a010/.

32 Nicola Galesi, Joshua A. Grochow, Toniann Pitassi, and Adrian She. On the algebraic proof complexity of tensor isomorphism. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference (CCC)*, volume 264 of *LIPIcs*, pages 4:1–4:40. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.CCC.2023.4.

33 Alexandra Goultiaeva, Allen Van Gelder, and Fahiem Bacchus. A uniform approach for generating proofs and strategies for both true and false QBF formulas. In *IJCAI*, pages 546–553, 2011. doi:10.5591/978-1-57735-516-8/IJCAI11-099.

34 Nashlen Govindasamy, Tuomas Hakoniemi, and Iddo Tzameret. Simple hard instances for low-depth algebraic proofs. In *63rd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 188–199. IEEE, 2022. doi:10.1109/FOCS54457.2022.00025.

35 Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system. *J. ACM*, 65(6):37:1–37:59, 2018. doi:10.1145/3230742.

36 Pavel Hrubeš. On lengths of proofs in non-classical logics. *Ann. Pure Appl. Logic*, 157(2–3):194–205, 2009. doi:10.1016/j.apal.2008.09.013.

37 Russell Impagliazzo, Pavel Pudlák, and Jirí Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Comput. Complex.*, 8(2):127–144, 1999. doi:10.1007/s000370050024.

38 Mikolás Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theor. Comput. Sci.*, 577:25–42, 2015. doi:10.1016/j.tcs.2015.01.048.

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

40 Jan Krajíček. *Proof complexity*, volume 170 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 2019.

41 Serge Lang. *Algebra (3. ed.)*. Addison-Wesley, 1993.

42 Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artif. Intell.*, 175(2):512–525, 2011. doi:10.1016/j.artint.2010.10.002.

43 Toniann Pitassi and Iddo Tzameret. Algebraic proof complexity: progress, frontiers and challenges. *SIGLOG News*, 3(3):21–43, 2016. doi:10.1145/2984450.2984455.

44 Luca Pulina and Martina Seidl. The 2016 and 2017 QBF solvers evaluations (QBFEVAL'16 and QBFEVAL'17). *Artif. Intell.*, 274:224–248, 2019. doi:10.1016/j.artint.2019.04.002.

45 Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987. doi:10.1007/BF00058680.

46 Sarah Sigley and Olaf Beyersdorff. Proof complexity of modal resolution. *J. Autom. Reason.*, 66(1):1–41, 2022. doi:10.1007/s10817-021-09609-9.

47 R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proc. ACM Symposium on Theory of Computing (STOC)*, pages 77–82. ACM Press, 1987. doi:10.1145/28395.28404.

48 Heribert Vollmer. *Introduction to Circuit Complexity – A Uniform Approach*. Texts in Theoretical Computer Science. Springer Verlag, Berlin Heidelberg, 1999. doi:10.1007/978-3-662-03927-4.