

Relating Size and Width in Variants of Q-Resolution

Judith Clymo^a, Olaf Beyersdorff^{b,a,1}

^a*School of Computing, University of Leeds, United Kingdom*

^b*Institute of Computer Science, Friedrich Schiller University Jena, Germany*

Abstract

In their influential paper ‘Short proofs are narrow – resolution made simple’ [3], Ben-Sasson and Wigderson introduced a crucial tool for proving lower bounds on the lengths of proofs in the resolution calculus. Over a decade later their technique for showing lower bounds on the size of proofs, by examining the width of all possible proofs, remains one of the most effective lower bound techniques in propositional proof complexity.

We continue the investigation begun in [6] into the application of this technique to proof systems for quantified Boolean formulas. We demonstrate a relationship between the size of proofs in level-ordered Q-Resolution and the width of proofs in Q-Resolution. In general, however, the picture is not positive, and for most stronger systems based on Q-Resolution, the size-width relation of [3] fails, answering an open question from [6].

1. Introduction

Proof complexity aims to understand the strength and limitations of various systems of logic. In particular, we seek upper and lower bounds on the size of proofs, and to develop general methods for finding such bounds. Resolution is a refutational system for propositional logic, with close connections to modern SAT solvers [7]. An important tool for proving lower bounds on the length of Resolution refutations was introduced in [3]. Ben-Sasson and Wigderson showed that whenever a short resolution refutation exists, a narrow refutation (i.e., of small width) can be constructed from it; so conversely if every refutation of some family of formulas must contain a clause of large width, then no small refutation can exist. In this context, width refers to the maximum number of literals in any clause in the proof.

The authors of [6] began the study of possible relationships between size, width and space of refutations in the context of resolution-based proof systems for quantified Boolean formulas (QBF). Understanding which lower bound techniques are effective for QBF is of great importance (cf. [4, 5]); however, the findings of [6] show that size-width relations in the spirit of [3] fail in Q-Resolution, both tree-like and DAG-like. This was shown by presenting a specific class of formulas with short proofs, but requiring large width (even when just counting existential variables).

¹Supported by grant no. 60842 from the John Templeton Foundation.

This investigation is continued here by considering three additional QBF proof systems: level-ordered Q-Resolution, universal Q-Resolution (QU-Res), and long-distance Q-Resolution (LDQ-Res). While QU-Res is a natural counterpart to propositional Resolution, level-ordered and long-distance Q-Resolution are motivated by their connections to QBF solving [10, 12, 14]. In particular, tree-like level-ordered Q-Resolution corresponds to the basic QDPLL algorithm, which lifts the classic DPLL algorithm to QBF and underlies a number of solving approaches.

Implicit assumptions underlying Ben-Sasson and Wigderson’s argument break down in the context of Q-Resolution due to the restrictions imposed by the quantifier prefix (cf. [6]). We show that the original argument of [3] can be lifted to QBF in level-ordered Q-Resolution and relate the proof size in that system to the width of Q-Resolution refutations. This holds only for the tree-like systems and cannot be lifted to DAG-like level-ordered Q-Resolution. In contrast, we lift the negative results of [6] to the stronger systems of QU-Res and LDQ-Res, thus answering a question of [6].

2. Preliminaries

2.1. Quantified Boolean formulas

Quantified Boolean logic is an extension of propositional logic in which variables may be universally as well as existentially quantified. We consider quantified Boolean formulas (QBFs) in closed prenex conjunctive normal form, denoted $\Phi = \mathcal{Q}\phi$. In the quantifier prefix, $\mathcal{Q} = \mathcal{Q}_1 X_1 \dots \mathcal{Q}_m X_m$, the X_i are disjoint sets of variables, and $\mathcal{Q}_i \in \{\forall, \exists\}$. The matrix ϕ is a formula in conjunctive normal form over the variables in $\bigcup_{i=1}^m X_i$. A variable $x \in X_i$ is at quantification level i , written $\text{lv}(x) = i$. We say that x is existentially quantified if $\mathcal{Q}_i = \exists$, universally quantified otherwise. If C is a clause in ϕ then $\text{var}(C)$ is the set of variables appearing in C .

For $a \in \{0, 1\}$, $\Phi[x/a]$ is the result of setting $x = a$ throughout ϕ and removing x from \mathcal{Q} , so $\Phi[x/1]$ removes all clauses from ϕ that contain x as a positive literal, and removes $\neg x$ from the clauses that contain the negative literal.

Semantically, $\forall u \Phi = \Phi[u/0] \wedge \Phi[u/1]$ and $\exists x \Phi = \Phi[x/0] \vee \Phi[x/1]$.

2.2. Refutations of false QBFs

A refutation π of QBF Φ derives the empty clause by application of the derivation rules. Each line of π is a clause that either appears in the input formula Φ , or is the result of applying a derivation rule to one or two clauses that already appear at an earlier line of π . The final line is the empty clause. A refutation induces a directed acyclic graph (DAG) with each internal node of the DAG associated with a clause, and edges directed from the parent(s) to the child of a single proof step. If the induced graph is a tree then the refutation is tree-like. In this case, each derived clause is only used once.

The result of substituting a ($a \in \{0, 1\}$) for x in every clause in π is $\pi[x/a]$. This substitution may cause some clauses to be satisfied, in which case they cannot be used in future derivations. Therefore it cannot immediately be assumed that the result of this restriction is a valid proof.

2.3. Resolution proof systems

Search-based solving for SAT is based on the DPLL procedure [8], often augmented with clause learning. The Resolution proof system [15] is a refutational proof system acting on propositional formulas in conjunctive normal form. Refutations of false formulas generated in search-based SAT solvers can be understood as Resolution proofs. The proof system has a single inference rule deriving $C \vee D$ from $C \vee x$ and $D \vee \neg x$ where C and D are clauses and x (the pivot) is a variable, and for all variables $y \neq x$ that appear in C , the negation of y does not appear in D .

Resolution is extended to act on QBFs in prenex conjunctive normal form by the addition of the universal reduction rule, which derives C from $C \vee x$ when x is universally quantified and is at the highest (inner-most) quantifier level of all variables appearing in C . This proof system is known as QU-Resolution (QU-Res) [16]. In Q-Resolution [13], the pivot in the resolution rule is restricted to existentially quantified variables.

Since search-based SAT solvers have proved successful, it is natural that this approach has been extended to QBF, for example in the solver depQBF [14]. The DPLL algorithm [9, 8], on which most SAT solvers are based, accepts a propositional formula in conjunctive normal form and iteratively selects variables to assign a value in a depth-first search for a satisfying assignment. The algorithm can be lifted to QBF with the additional restriction that it must assign variables according to the order of the quantifier prefix, starting from the outermost block.

Just as DPLL corresponds to tree-like Resolution [7], so the decision tree generated by a QDPLL search on a false QBF corresponds to a proof in tree-like level-ordered Q-Resolution. Decisions on existentially quantified variables correspond to resolution steps. For universally quantified variables only one assignment needs to be considered in order to prove the input formula false, so these assignments correspond to universal reduction steps. It has been shown that even with clause learning and other heuristics, it is impossible for a search-based QBF solver to produce a refutation that is not level-ordered on some input formulas [12]. This link to QBF solvers is the reason for studying *level-ordered* Q-Resolution in particular.

Definition 1 ([11]). *Let π be a Q-Resolution proof. We say that π is level-ordered if and only if the following holds: Consider an arbitrary path from the root to some leaf and two resolution steps on that path on literals x_1 and x_2 , respectively, such that the resolution on x_1 is closer to the root. Then, $lv(x_1) \leq lv(x_2)$.*

Long-distance Resolution (LDQ-Res) [1] allows universal literals u and $\neg u$ to appear in the two clauses being resolved together provided that the resolution variable x is at a lower quantification level than u . The opposing literals are merged to form the special universal literal u^* . Formally,

$$\frac{C_1 \vee U_1 \vee x \quad C_2 \vee U_2 \vee \neg x}{C_1 \vee C_2 \vee U}$$

where x is existentially quantified. The clauses C_1 and C_2 must not contain any complementary literals or special universal literals. Further, U_1 and U_2 contain only universal literals appearing later in the quantifier prefix than the pivot x ,

every literal in U_1 is a special universal literal, or has its complement in U_2 (and vice versa). Then $U = \{u^* | u \in \text{var}(U_1)\}$. The literal u^* can be \forall -reduced in the same way as any other universal literal.

Note that while QU-Resolution and LDQ-Res are exponentially separated from Q-Resolution [16, 10], it is easy to see that they do not reduce proof size when the proof is required to be level-ordered.

2.4. Size and width

The size of a proof π is written $|\pi|$ and is the number of clauses in π (equivalently, the number of nodes in the associated tree or DAG). The size of deriving a clause C from Φ (in proof system P), denoted $S_P(\Phi \vdash C)$, is the minimum size of any P -proof of C from Φ . We drop the subscripts indicating the proof system under consideration if it is already clear from the context.

The width $w(C)$ of a clause C is the number of existential variables it contains. The width $w(\Phi)$ of a QBF Φ is the maximum width of a clause in Φ . Similarly, the width $w(\pi)$ of derivation π is the maximum width of any clause contained in π . The width of deriving a clause C from Φ (in proof system P), denoted $w_P(\Phi \vdash C)$, is the minimum width of any P -proof of C from Φ .

For tree-like propositional Resolution Ben-Sasson and Wigderson [3] showed that $w(\phi \vdash \perp) \leq w(\phi) + \lg(S(\phi \vdash \perp))$, and a similar relation holds for DAG-like Resolution, $w(\phi \vdash \perp) \leq w(\phi) + O(\sqrt{n \ln S(\phi \vdash \perp)})$ where n is the number of variables in ϕ .

3. Negative results

We revisit the counterexample for the size-width relation in tree-like Q-Resolution from [6].

Proposition 2 ([6]). *There is a family of false QBF sentences Φ_n over $O(n^2)$ variables, such that $w(\Phi_n) = 3$ and in tree-like Q-Resolution $S(\Phi_n \vdash \perp) = n^{O(1)}$, and $w(\Phi_n \vdash \perp) = \Omega(n)$.*

To prove this proposition, the following QBFs, introduced in [11], are used.

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

$$\bigwedge_{i,j=1}^n (x_{i,j} \vee z \vee a_i) \wedge \bigwedge_{i,j=1}^n (\neg x_{i,j} \vee \neg z \vee b_j) \quad (1)$$

$$\wedge \neg y_0 \wedge \bigwedge_{i=1}^n (y_{i-1} \vee \neg a_i \vee \neg y_i) \wedge y_n \quad (2)$$

$$\wedge \neg p_0 \wedge \bigwedge_{j=1}^n (p_{j-1} \vee \neg b_j \vee \neg p_j) \wedge p_n \quad (3)$$

There are $O(n^2)$ -size tree-like Q-Resolution refutations, given by the following procedure:

1. Collapse the clauses in (2) and (3) to $\bigvee_{i=1}^n \neg a_i$ and $\bigvee_{j=1}^n \neg b_j$ (this takes $O(n)$ steps).
2. For each j

- (a) Use the clauses $(x_{i,j} \vee z \vee a_i)$ with $\bigvee_{i=1}^n \neg a_i$ to derive $(x_{1,j} \vee \dots \vee x_{n,j} \vee z)$, then \forall -reduce z (requiring $O(n)$ steps for each j).
 - (b) The result of 2a, $(x_{1,j} \vee \dots \vee x_{n,j})$, and the clauses $(\neg x_{i,j} \vee \neg z \vee b_j)$ are used to derive $(b_j \vee \neg z)$ (again, $O(n)$ steps for each j).
3. The n clauses of the form $(b_j \vee \neg z)$ together with $\bigvee_{j=1}^n \neg b_j$ are used to derive $\neg z$ which is \forall -reduced to reach the empty clause (this takes $O(n)$ steps).

Now we show that any valid Q-Resolution refutation must be wide, by arguing that the clause immediately following the first \forall -red step in any refutation of Φ_n must have width $\Omega(n)$. We do not repeat the full argument here, the idea is that in order to perform \forall -red on z (w.l.o.g.), some a_i must be removed from a clause $(x_{i,j} \vee z \vee a_i)$. Doing so necessarily introduces another a literal negatively (perhaps via the intermediate introduction of a y literal), and to remove this introduces another (different) $x_{i,j}$, which must remain until after the \forall -red, as well as another positive a literal. This repeats and ensures that n different $x_{i,j}$ literals must collect in the clause before it is free of a_i and y_i literals and the \forall -red can be performed.

It was left open in [6] whether the size-width relation holds for extensions of Q-Resolution. We begin by demonstrating that simple modifications of this example, inspired by [2], show that the result also fails for the tree-like versions of QU-Res and LDQ-Res.

Proposition 3. *There is a family of false QBF sentences Φ'_n over $O(n^2)$ variables, such that $w(\Phi'_n) = 3$ and in tree-like QU-Resolution $S(\Phi'_n \vdash \perp) = n^{O(1)}$, and $w(\Phi'_n \vdash \perp) = \Omega(n)$.*

Proof. Modify Φ_n to Φ'_n by adding another universal variable z' at the same level as z . Replace (1) with $\bigwedge_{i,j=1}^n (x_{i,j} \vee z \vee z' \vee a_i) \wedge \bigwedge_{i,j=1}^n (\neg x_{i,j} \vee \neg z \vee \neg z' \vee b_j)$.

The size $O(n^2)$ refutation of Φ_n is trivially extended to a refutation of Φ'_n by performing a \forall -reduction step on z' immediately after any \forall -reduction step on z . It is simple to confirm that the proof remains valid. In particular, any resolution step that could be blocked by z' in Φ'_n would already have been blocked by z in Φ .

The duplication of the universal variable also ensures that universal resolution cannot result in narrower proofs compared to Q-Resolution. This is simply because we have ensured that every universal literal in the input formula has its complement only in clauses which also conflict on another variable. In addition, any derived clause must contain all of the universal variables from its parents unless it is derived by \forall -reduction or universal resolution. Therefore universal resolution is forbidden until some \forall -reduction step has occurred, until this point we may only use existential resolution. The argument sketched above therefore readily applies to show that the clause immediately following the first \forall -red step in any refutation of Φ_n must have width $\Omega(n)$. \square

The idea of duplicating universal variables can be applied to any formula to prevent universal resolution steps. In particular, using the family of formulas from [6] with short refutations and large width in dag-like Q-Resolution, we can construct similar formulas with short refutations and large width in dag-like QU-Resolution, thus also refuting the size-width relation for dag-like QU-Resolution. (Note that the formulas Φ'_n above cannot be used directly as they have a quadratic number of variables.)

Proposition 4. *There is a family of false QBF sentences Φ_n'' over $O(n^2)$ variables, such that $w(\Phi_n'') = 3$ and in tree-like LDQ-Resolution $S(\Phi_n'' \vdash \perp) = n^{O(1)}$, and $w(\Phi_n'' \vdash \perp) = \Omega(n)$.*

Proof. In this case Φ_n is modified by replacing lines (2) and (3) with $\neg y_0 \wedge \bigwedge_{i=1}^n (y_{i-1} \vee \neg a_i \vee \neg y_i \vee z) \wedge y_n$ and $\neg p_0 \wedge \bigwedge_{j=1}^n (p_{j-1} \vee \neg b_j \vee \neg p_j \vee \neg z) \wedge p_n$ respectively. This does not affect satisfiability since these clauses are only relevant under one or other of the assignments to z . The same $O(n^2)$ refutation of the original formula applies to give the size upper bound.

For the width lower bound note that long-distance steps are impossible in refuting this formula. The only long-distance steps that could be performed are on some $x_{i,j}$ variable prior to any \forall -reduction in the clauses involved. Then the parent clauses contain some a , b , y , or p literal, as well as the special universal z^* . In order for the derived clause to form part of the refutation, a_i (without loss of generality) would need to be removed via resolution at some later point. This is now impossible. The only input clause that contains $\neg a_i$ also contains z , and any derived clause containing $\neg a_i$ must also contain z because \forall -reduction is blocked by $\neg a_i$. None of these clauses can be resolved with a clause containing z^* , so no long-distance resolution step can apply before at least one \forall -red step, until then we are restricted to standard resolution steps, and so again the clause following the first \forall -red must have width $\Omega(n)$. \square

4. Relating size and width between tree-like and tree-like level-ordered Q-Resolution

Ben-Sasson and Wigderson [3] show that it is possible to construct a narrow (tree-like) Resolution proof from a short Resolution proof. We will show that the same argument can be applied to a level-ordered Q-Resolution refutation, but that the constructed proof may not remain level-ordered. First we examine the reason that the proof must be level-ordered.

Suppose we have a Resolution refutation of some propositional formula ϕ . The final step in the proof resolves x and $\neg x$. So we also have a derivation from ϕ to x (and also $\neg x$), that is, ϕ implies x ($\neg x$). A crucial part of Ben-Sasson and Wigderson's argument in [3] rests on the observation that this derivation of x can be easily transformed into a refutation of $\phi[x/0]$, by simply applying the assignment $x = 0$ to every clause in the derivation.

This does not hold in general in Q-Resolution, even in the tree-like case. Indeed, it is possible to have a Q-Resolution derivation from Φ to u , a universal variable not at the outermost level in the prefix, where $\Phi[u/0]$ is not even false. The restricted derivation only remains valid if no clause has been satisfied by the assignment. Therefore, if we have a derivation of x , and so wish to restrict by $x = 0$, we must know that $\neg x$ does not appear in any clause in the derivation. We show that this property holds for tree-like level-ordered Q-Resolution proofs.

Definition 5 (Regularity). *A refutation π of Φ is regular if along any path from root to leaf no two nodes are associated with the same variable.*

Lemma 6. *A tree-like level-ordered Q-resolution proof can be transformed into a tree-like level-ordered regular Q-resolution proof without increasing its size.*

Proof. Suppose two \forall -reduction steps that both remove u , and the path between them. By the assumption that the proof is level-ordered no resolution step appears along this path, therefore each clause is strictly contained in its predecessor and the second \forall -reduction step must act on a clause that does not contain u , creating a contradiction.

Suppose that there are two resolution steps on the same branch that both have pivot x and, without loss of generality, assume all intermediate steps have a different pivot. First $(A \vee x)$ and $(B \vee \neg x)$ are resolved to give $(A \vee B)$. Since $(A \vee B)$ cannot contain x or $\neg x$, then x must be reintroduced to this branch by some other resolution step. The second resolution step on x resolves clause $(F \vee \neg x)$ with $(E \vee x)$ from our branch.

We seek to construct a new, smaller proof. Consider the branch from the first resolution step on x to the root of the tree. We have clauses $C_1 \dots C_n \dots C_m$ with $C_1 = A \vee B$, $C_n = E \vee F$ and $C_m = \perp$. Along the branch each clause C_k will be replaced by a new clause C'_k .

Our aim is to remove the duplicate resolution on x , so we begin by replacing $C_1 = A \vee B$ with $C'_1 = A \vee x$. If C_k is the result of \forall -reduction removing u from C_i then $C'_k = C'_i \setminus \{u\}$. If C_k is the result of resolving C_i from this branch and side clause D_i on pivot y and $y \notin C'_i$ let $C'_k = C'_i$. The branch deriving D_i can be removed. If $y \in C'_i$, then $C'_k = C'_i \cup D_i \setminus \{y\}$. By assumption D_i does not contain $\neg x$ while $i < n$ so deriving C'_k from C'_i and D_i is a valid resolution step provided that $C'_i \setminus x \subseteq C_i$ for $i < n$ and $C'_i \subseteq C_i$ for $i \geq n$, which we show now by induction.

For the base case, $C'_1 \setminus x = A \subseteq A \vee B = C_1$. Assume $C'_i \setminus x \subseteq C_i$ and consider how C_k is derived.

1. If C_k is the result of \forall -reduction on u then $C_k = C_i \setminus \{u\}$ and $C'_k = C'_i \setminus \{u\}$ so $C'_k \setminus \{x\} = C'_i \setminus \{u, x\} \subseteq C_i \setminus \{u\} = C_k$.
2. If C_k is the result of resolving C_i and D_i on y then $C_k = C_i \cup D_i \setminus \{y\}$.
 - (a) When $y \notin C'_i$ then $C'_k = C'_i$ so $C'_k \setminus \{x\} = C'_i \setminus \{x\} = C'_i \setminus \{x, y\} \subseteq C_i \setminus \{y\} \subseteq C_k$.
 - (b) When $y \in C'_i$ then $C'_k = C'_i \cup D_i \setminus \{y\}$ so $C'_k \setminus \{x\} = C'_i \cup D_i \setminus \{x, y\} \subseteq C_i \cup D_i \setminus \{y\} = C_k$.

If $C'_i \subseteq C_i$ we can by an identical argument infer that $C'_k \subseteq C_k$ for $k > i$. Therefore $C' \setminus x$ subsumes C and from the point at which x was reintroduced to the branch in the original refutation C' subsumes C . The resolution steps are all valid, and since because the original proof was level-ordered no \forall -reduction step is blocked by x .

Any consecutive nodes with identical associated clauses can be merged, and we have removed at least one resolution step to construct a new refutation no larger than the original, which remains tree-like and level-ordered. \square

From now on we will assume that a level-ordered Q-Resolution refutation is also regular, and make the following simple observation.

Lemma 7. *Let π be a regular refutation of Φ . If the final step of π is resolution on x then $\pi[x/0]$ is a refutation of $\Phi[x/0]$ and $\pi[x/1]$ is a refutation of $\Phi[x/1]$. If the final step of π is \forall -reduction on x then either $\pi[x/0]$ is a refutation of $\Phi[x/0]$ or $\pi[x/1]$ is a refutation of $\Phi[x/1]$.*

Lemma 8. *Let Φ be a QBF, C a clause, and x an existentially quantified variable in the outermost quantifier block of Φ . In Q-Resolution, if $w(\Phi[x/0] \vdash C) \leq k$ then $w(\Phi \vdash C \vee x) \leq k + 1$.*

Proof. Let π be a width k Q-Resolution derivation of C from $\Phi[x/0]$. Add x into every clause of π . Initial clauses may be obtained by weakening. If C is the result of resolving A and B then $C \vee x$ can be derived from $A \vee x$ and $B \vee x$, which is valid since neither clause can contain $\neg x$. If C is the result of a universal reduction from A then $C \vee x$ is the result of universal reduction from $A \vee x$, which is valid since x is outermost. The width of the derivation is increased by 1. \square

Similarly, if $w(\Phi[x/1] \vdash C) \leq k$ then $w(\Phi \vdash C \vee \neg x) \leq k + 1$. Also in the following, all $[x/0]$ may be swapped for $[x/1]$ and vice versa.

Lemma 9. *Let Φ be a QBF and x an existentially quantified variable in the outermost quantifier block of Φ . In Q-Resolution, if $w(\Phi[x/0] \vdash \perp) \leq k - 1$ and $w(\Phi[x/1] \vdash \perp) \leq k$ then $w(\Phi \vdash \perp) \leq \max\{k, w(\Phi)\}$. For x universally quantified in the outermost quantifier block, if $w(\Phi[x/0] \vdash \perp) \leq k$ then $w(\Phi \vdash \perp) \leq k$.*

Proof. If x is existentially quantified then since $w(\Phi[x/0] \vdash \perp) \leq k - 1$ we have also that $w(\Phi \vdash x) \leq k$ by Lemma 8. Resolve x with every clause in Φ containing $\neg x$, the resulting collection of clauses are exactly those in the matrix of $\Phi[x/1]$, and from these we can derive \perp in width k . The total width of the derivation is $\max\{k, w(\Phi)\}$.

If x is universally quantified then $\{x\}$ can be derived in width k and \forall -reduction derives \perp . The width of the derivation is k since universally quantified variables do not contribute to the width of a clause. \square

We can now state the relation between width in Q-Resolution and size in level-ordered Q-Resolution.

Theorem 10. $w_Q(\Phi \vdash \perp) \leq w(\Phi) + \lceil \lg S_L(\Phi \vdash \perp) \rceil$, where Q is Q-Resolution and L is level-ordered tree-like Q-Resolution.

Proof. We begin with a level-ordered refutation π of Φ . Let $b = \lceil \lg S_L(\Phi \vdash \perp) \rceil$ so that $S_L(\Phi \vdash \perp) \leq 2^b$. If $b = 0$ then the empty clause is in Φ , so both $w(\Phi)$ and $w_Q(\Phi \vdash \perp)$ are 0 and we are done.

Otherwise the last step of the proof may be a universal reduction $\frac{x}{\perp}$ or a resolution step $\frac{x \quad \neg x}{\perp}$. The pivot x must belong to the outermost quantifier block that appears in the proof, since the proof is level ordered.

In the case of universal reduction, consider π_x , the derivation of x . Then $\pi_x[x/0]$ is a level-ordered refutation of $\Phi[x/0]$ of size S_x . By induction on the number of variables in Φ we have that $w_Q(\Phi[x/0] \vdash \perp) \leq w(\Phi[x/0]) + \lceil \lg S_x \rceil$. By Lemma 9, $\Phi \vdash \perp$ has the same width as the restricted proof, and $w(\Phi[x/0]) = w(\Phi)$, so the result follows.

In the case of resolution being the last step, consider π_x and $\pi_{\neg x}$, the level-ordered derivations of x and $\neg x$, of sizes S_x and $S_{\neg x}$, respectively. We have that $\pi_x[x/0]$ is a level-ordered refutation of $\Phi[x/0]$ of size S_x and $S_L(\Phi \vdash \perp) = S_x + S_{\neg x} + 1$. Without loss of generality, $S_x \leq 2^{b-1}$ so by induction on b , there is a (possibly not level-ordered) proof with $w_Q(\Phi[x/0] \vdash \perp) \leq w(\Phi[x/0]) + b - 1 \leq$

$w(\Phi) + b - 1$, and by induction on the number of variables in Φ , $w_Q(\Phi[x/1] \vdash \perp) \leq w(\Phi[x/1]) + b \leq w(\Phi) + b$. Since x is outermost, by Lemma 9 we can use these two refutations to construct a refutation of Φ with width at most $w(\Phi) + b$ and the result follows. \square

In the proof of Theorem 10, we begin with a small level-ordered proof and construct another proof from it which has small width. However, during the construction, the proof loses the level-ordered property. In general it is not possible to construct a level-ordered proof with small width, as demonstrated by the following counter-example

$$\Phi = \exists x_1 \dots x_n \forall z \exists a_1 \dots a_n, y_0 \dots y_n \\ (\neg y_0) \wedge (y_n) \wedge \bigwedge_{i \in [n]} (\neg x_i) \wedge (z \vee a_i) \wedge (y_{i-1} \vee \neg a_i \vee x_i \vee \neg y_i).$$

All clauses are needed to refute Φ . Any level-ordered proof must carry out all resolution steps on y_i variables before resolving on x_i variables, and it is simple to verify that doing so must result in a clause that contains all x_i variables. There is a short tree-like level-ordered refutation which collapses $(y_{i-1} \vee \neg a_i \vee x_i \vee \neg y_i)$ together to $(\neg a_1 \vee \dots \vee \neg a_n \vee x_1 \vee \dots \vee x_n)$, then resolves this with all $(z \vee a_i)$, removes z and finally refutes $\bigwedge_{i \in [n]} (\neg x_i) \wedge (x_1 \vee \dots \vee x_n)$, all of which takes linear size.

Additionally, Theorem 10 cannot be lifted to DAG-like level-ordered Q-Resolution since for the counter-example given in [6] for DAG-like Q-Resolution, the short proof discussed in [6] is level-ordered and directly applies here. A crucial part of the argument in the propositional case is to carefully select the next variable to use in restricting the refutation, but it is not possible in general to ensure that this variable belongs to a particular level of the prefix.

5. Conclusion

We have demonstrated that the result of [3] can be lifted to relate two variants of Q-Resolution, highlighting an interesting relationship between level-ordered and non level-ordered proofs in Q-Resolution. Level-ordered Q-Resolution is important since it corresponds to the QDPLL algorithm that underlies some modern QBF solving algorithms, so a mechanism to lower bound the size of proofs is useful in understanding the strength of search-based QBF solvers. Removing either the restriction that the proof must be level ordered, or the restriction that it must be tree like, is sufficient to lose the desired behaviour. We have also answered the open question from [6] regarding extensions of Q-Resolution, by demonstrating how the counterexamples may be lifted to these stronger calculi.

- [1] Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. *Formal Methods in System Design*, 41(1):45–65, 2012.
- [2] 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.

- [3] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.
- [4] Olaf Beyersdorff, Ilario Bonacina, and Leroy Chew. Lower bounds: From circuits to QBF proof systems. In *Proc. ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 249–260. ACM, 2016.
- [5] Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Feasible interpolation for QBF resolution calculi. *Logical Methods in Computer Science*, 13, 2017.
- [6] Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Are short proofs narrow? QBF resolution is not so simple. *ACM Transactions on Computational Logic*, 19, 2018.
- [7] Samuel R. Buss. Towards NP-P via proof complexity and search. *Ann. Pure Appl. Logic*, 163(7):906–917, 2012.
- [8] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, July 1962.
- [9] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, July 1960.
- [10] 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.
- [11] Mikoláš Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theor. Comput. Sci.*, 577:25–42, 2015.
- [12] Mikoláš Janota. On Q-Resolution and CDCL QBF solving. In *Proc. Theory and Applications of Satisfiability Testing (SAT)*, pages 402–418, 2016.
- [13] Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified Boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995.
- [14] Florian Lonsing and Uwe Egly. DepQBF 6.0: A search-based QBF solver beyond traditional QCDCL. In *Proc. International Conference on Automated Deduction (CADE)*, pages 371–384, 2017.
- [15] John Alan Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965.
- [16] Allen Van Gelder. Contributions to the theory of practical quantified Boolean formula solving. In *Proc. Principles and Practice of Constraint Programming (CP)*, pages 647–663, 2012.