An Achilles' Heel of Term-Resolution

Mikoláš Janota[1] and Joao Marques-Silva[2]

[1] IST/INESC-ID, Lisbon, Portugal
[2] LaSIGE, Faculty of Science, University of Lisbon, Portugal

Abstract. Term-resolution provides an elegant mechanism to prove that a quantified Boolean formula (QBF) is true. It is a dual to Q-resolution and is practically highly important as it enables certifying answers of DPLL-based QBF solvers. While term-resolution and Q-resolution are very similar, they are not completely symmetrical. In particular, Q-resolution operates on clauses and term-resolution operates on models of the matrix. This paper investigates the impact of this asymmetry. We will see that there is a large class of formulas (formulas with "big models") whose term-resolution proofs are exponential. As a possible remedy, the paper suggests to prove true QBFs by refuting their negation (*negate-refute*), rather than proving them by term-resolution. The paper shows that from the theoretical perspective this is indeed a favorable approach. In particular, negation-refutation p-simulates term-resolution and there is an exponential separation between the two calculi. These observations further our understanding of proof systems for QBFs and provide a strong theoretical underpinning for the effort towards non-CNF QBF solvers.

1 Introduction

Arguably, the interest of computer scientists in proof complexity begins with the seminal work of Cook and Reckhow who showed a relation between proof complexity and the question NP vs. co-NP [8]. This interest was further fueled by the practical success of programs for automated reasoning, such as *SMT solvers* or *SAT solvers*. Machine-verifiable proofs serve as *certificates* for such solvers. It is important that a solver can produce a certificate of its answer as the solver itself can contain bugs [24,6]. Moreover, proofs have turned out to be important artifacts for further computations, like invariant inference for example [18]. This paper follows this line of research, i.e. proof complexity and solver complication certification, with the focus on *quantified Boolean formula* (QBF). In particular, it focuses on QBFs whose propositional part is in *conjunctive normal form* (QCNF). QCNF is complete and widely popular input for QBF solvers due to its susceptibility to simple representation inside the solver.

A number of QCNF solvers take inspiration in the approach that turned out to be so successful for SAT; and that is *conflict driven clause learning* (CDCL) [21,20,17]. Since *propositional resolution* is the underlying proof principle used in SAT, an analogous proof system was developed for QCNF. In particular, *Q-resolution* [14] for false formulas, and *term-resolution* [10] for true formulas. It has been shown that CDCL-based QBF solvers [27] can be certified by these two proof systems [10]. Recently, several proof complexity analyses of Q-resolution were published. A separation result for

Q-resolution and a sequent calculus by Krajíček and Pudlák [16] is shown by Egly [9]; Van Gelder shows that enabling resolution on universal-variables in Q-resolution proofs gives an exponential advantage to Q-resolution [25]. The relation between Q-resolution and expansion-based systems was investigated by Beyersdorff et al. [5]. A number of variants of Q-resolution have been considered [3]. Proof systems for QBF were also investigated in the context of preprocessing [12].

This paper brings the focus to term-resolution. While term-resolution is an elegant system because it provides a dual to Q-resolution, the two types of resolution are not perfectly symmetric. This is because Q-resolution can operate on the given clauses but term-resolution operates on the satisfying assignments of those clauses. This paper shows that this difference exposes an Achilles' heel of term-resolution.

The first result of this paper is that it shows that term-resolution proofs are large for QCNFs whose propositional part have models with a large number of universal literals. More precisely, if each model has at least k universal literals, any term-resolution proof has at least 2^k nodes. Subsequently, the paper investigates an alternative route to term-resolution and that is *refuting the negation* of the formula. The paper shows that any term-resolution proof can be translated to a negation-refutation in polynomial time. On a particular formula Ψ we show an exponential separation between negation-refutation and term-resolution, i.e. all term-resolution proofs of Ψ are exponential but there is a Q-resolution proof of $\neg\Psi$ that is polynomial.

These results have direct practical implications for QBF solving because term-resolution enables certifying DPLL-based QBF solvers. Consequently, a formula whose term-resolution proofs are exponential, will require exponential time to *solve*. These theoretical results further substantiate an observation already made in the QBF community and that is that QBF with propositional part in CNF are particularly harmful for solving [2,26].

2 Preliminaries

A *literal* is a Boolean variable or its negation. For a literal l, we write $\bar{l}$ to denote the literal *complementary* to l, i.e. $\bar{x} = \neg x$, $\overline{\neg x} = x$. A *clause* is a disjunction of finitely many literals. A formula in *conjunctive normal form* (CNF) is a conjunction of finitely many clauses. As common, whenever convenient, a clause is treated as a set of literals and a CNF formula as a set of sets of literals.

For a literal $l = x$ or $l = \bar{x}$, we write $\mathsf{var}(l)$ for x; for a clause C, $\mathsf{var}(C)$ denotes $\{\mathsf{var}(l) \mid l \in C\}$, and for a CNF ψ, $\mathsf{var}(\psi)$ denotes $\{l \mid l \in \mathsf{var}(C), C \in \psi\}$. For a set of variables X an *assignment* is a function from X to the constants 0 and 1. We say that the assignment is *complete* for X if the function is total.

A *term* is a conjunction of finitely many non-complementary literals. Whenever convenient, a term is treated as a set of literals. We say that a term T is a *model* of a CNF ϕ if for each $C \in \phi$ there is a literal l both in T and C, i.e. $C \cap T \neq \emptyset$.

There is an obvious relation between terms and assignments. A term uniquely determines a set of assignments that satisfy the term. If an assignment satisfies a model of ϕ, then it is a satisfying assignment of ϕ. Note that some definitions require a model

to be a complete assignment to the variables of ϕ. The aforementioned correspondence shows that there's no substantial difference between the definitions.

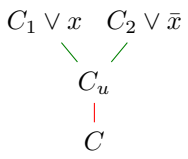
Quantified Boolean Formulas (QBFs) [13] extend propositional logic with quantifiers with the standard semantics that $\forall x.\Psi$ is satisfied by the same truth assignments as $\Psi[x/0] \wedge \Psi[x/1]$ and $\exists x.\Psi$ as $\Psi[x/0] \vee \Psi[x/1]$. Unless specified otherwise, QBFs are in *closed prenex* form with a CNF *matrix*, i.e. $\mathcal{Q}_1 X_1 \ldots \mathcal{Q}_k X_k.\ \phi$, where X_i are pairwise disjoint sets of variables; $\mathcal{Q}_i \in \{\exists, \forall\}$ and $\mathcal{Q}_i \neq \mathcal{Q}_{i+1}$. The formula ϕ is in CNF and is defined only on variables $X_1 \cup \ldots \cup X_k$. The propositional part ϕ is called the *matrix* and the rest the *prefix*. We write QCNF to talk about formulas in this form. If a variable x is in the set X_i, we say that x is at *level* i and write $\mathsf{lv}(x) = i$; we write $\mathsf{lv}(l)$ for $\mathsf{lv}(\mathsf{var}(l))$. A closed QBF is *false* (resp. *true*), iff it is semantically equivalent to the constant 0 (resp. 1).

If a variable is universally quantified, we say that the variable is universal. For a literal l and a universal variable x such that $\mathsf{var}(l) = x$, we say that l is universal. The notions of existential variable and literal are defined analogously.

2.1 Q-resolution

Q-resolution [14] is an extension of propositional resolution for showing that a QCNF is false. For a clause C, a universal literal $l \in C$ is *blocked* by an existential literal $k \in C$ iff $\mathsf{lv}(l) < \mathsf{lv}(k)$. $\forall$-*reduction* is the operation of removing from a clause C all universal literals that are *not* blocked by some literal. For two $\forall$-reduced clauses $x \vee C_1$ and $\bar{x} \vee C_2$, where x is an existential variable, a *Q-resolvent* [14] is obtained in two steps. (1) Compute $C_u = C_1 \cup C_2 \setminus \{x, \bar{x}\}$. If C_u contains complementary literals, the Q-resolvent is undefined. (2) $\forall$-reduce C_u. For a QCNF $\mathcal{P}.\phi$, a A *Q-resolution proof* of a clause C is a finite sequence of clauses $C_1, \ldots, C_n$ where $C_n = C$ and any C_i in the sequence is part of the given matrix ϕ or it is a Q-resolvent for some pair of the preceding clauses. A Q-resolution proof is called a *refutation* iff C is the empty clause, denoted $\bot$.

In this paper Q-resolution proofs are treated as connected directed acyclic graphs so that the each clause in the proof corresponds to some node labeled with that clause. We assume that the input clauses are already $\forall$-reduced. Q-resolution steps are depicted as on the right. Note that the $\forall$-reduction step is depicted separately.



2.2 Term-Resolution

Term-resolution is analogous to Q-resolution with the difference that it operates on terms and its purpose is to prove that a QCNF is true [10]. Since the calculus operates on QBFs with CNF matrices, it needs a mechanism to generate terms to operate on. This is done by a rule that enables using models of the given matrix in the proof.

Term-resolution, resolves on universal literals and reduces existential ones. For a term T an existential literal l is *blocked*, iff there is a universal $k \in T$ such that $\mathsf{lv}(l) < \mathsf{lv}(k)$. $\exists$-reduction removes from a term T all existential literals that are *not*

blocked by some universal literal. For two $\exists$-reduced terms $x \wedge T_1$ and $\bar{x} \wedge T_2$, a *term-resolvent* is defined as the $\exists$-reduction of the term $T_1 \wedge T_2$, if T_1 and T_2 do not contain complementary literals; it is undefined otherwise.

For a QCNF $P.\ \phi$ a *term-resolution proof* of the term T_m a is a finite sequence $T_1, \ldots, T_m$ of terms such that each term T_i is a model of ϕ or it was obtained from the previous terms by $\exists$-reduction or term-resolution. Such a proof *proves* $\mathcal{P}\ .\ \phi$ iff T_m is the empty term, denoted as $\top$. Those terms of the proof that are models of ϕ are said to be generated by a *model generation rule*. In the literature, terms are sometimes referred to as "cubes", especially in the context of DPLL QBF solvers that apply cube learning [27].

2.3 Proof complexity

A *proof system* P is relation $P(\Phi, \pi)$ that is computable in polynomial time such that a formula Φ is true iff there exists a proof π for which $P(\Phi, \pi)$. A proof system P_1 *p-simulates* a proof system P_2 iff any proof in P_2 of a formula Φ can be translated into a proof in P_1 of Φ in polynomial time (cf. [8,23]).

As is common, we will count the sizes of Q-resolution and term-resolution as the number of resolution steps and number of $\forall$ / $\exists$-reductions where each reduced literal is counted separately.

3 The Achilles' Heel

This section describes a large class of formulas that have exponential term-resolution proofs. Recall that a leaf of a term-resolution proof must be generated by the model-generation rule. We will show that due to this, a refutation proof can be "forced" to generate exponentially many leafs.

First we make a simple observation that for any assignment to universal variables, there must be a leaf-term in a term-resolution proof that "agrees" with that assignment. We say that a term T *agrees* with an assignment τ iff there is no literal l such that $\bar{l} \in T$ and $\tau(l) = 1$.

Lemma 1. *For any assignment τ to universal variables and a term-resolution proof π of some QCNF $P.\ \phi$ there is a leaf-term T of π that agrees with τ.*

Proof. We construct a path from the root to some leaf such that each node on that path agrees with τ. The root of π agrees with τ because it does not contain any literals. If a term T agrees with τ and T is obtained from T' by existential-reduction, then T' also agrees with τ since τ assigns only to universal variables. If T agrees with τ and is obtained from T_1 and T_2 by term-resolution on some variable y, it has to be that y is in one of the T_1, T_2 and $\bar{y}$ in the other. Hence, at least one of the terms agrees with τ. $\square$

Theorem 1. *Consider a QCNF $\Phi = P.\ \phi$. If all models of ϕ contain at least k universal literals, then any term-resolution proof of Φ has at least 2^k leafs.*

Proof. Let V_u be the set of universal variables of Φ. Since each leaf-term of any term-resolution proof has at least k universal literals, it can agree with at most $2^{|V_u|-k}$ different complete assignments to the universal variables. Lemma 1 gives that for any of the $2^{|V_u|}$ total assignments to V_u there must be a corresponding leaf-term. Averaging gives that π has at least $\frac{2^{|V|}}{2^{|V|-k}} = 2^k$ leafs. □

Theorem 1 gives us a powerful method of constructing formulas with large term-resolution proofs. It is sufficient to construct a true QCNF whose models have many universal literals. Let us construct one such formula. For a given parameter $N \in \mathbb{N}^+$ construct the following formula with $2N$ variables and $2N$ clauses.

$$\forall x_1, \exists y_1, \ldots, \forall x_N, \exists y_N . \bigwedge_{i \in 1..N} (\bar{x}_i \vee y_i) \wedge (x_i \vee \bar{y}_i) \tag{1}$$

Proposition 1. *Any term-resolution proof of* (1) *is exponential in* N.

Proof. Formula (1) is true as each of the existential variables y_i can be set to the same value as the variable x_i and thus satisfying the matrix.

Let ψ denote the matrix of (1). Each pair of clauses $\bar{x}_i \vee y_i$ and $\bar{y}_i \vee x_i$ must be satisfied by any model τ of ψ, which can be only done in two ways: the model contains the literals $\{y_i, x_i\}$ or it contains the literals $\{\bar{y}_i, \bar{x}_i\}$. Hence τ contains a literal for each x_i and for each y_i. Theorem 1 gives that at least 2^N models are needed in the leafs of any term-resolution proof. □

4 A Possible Remedy—Negation

This section suggests a possible remedy to the weakness exposed in the previous section. Instead of proving a formula true by term-resolution, we propose to refute its negation by Q-resolution.

To construct the negation of a formula, we follow the standard equalities $\neg\forall x. \Psi = \exists x. \neg\Psi$ and $\neg\exists x. \Psi = \forall x. \neg\Psi$. In order to bring the matrix back to conjunctive normal form, we add additional (*Tseitin*) variables [22]. We use the optimization by Plaisted-Greenbaum, which enables encoding variables' semantics only in one direction [19]. In particular, for each clause we introduce a fresh variable that is forced to true when that clause becomes true. Using these variables, we construct a clause that requires that at least one of the clauses is false.

It would be correct to insert these fresh variables at the end of the prefix (existentially quantified) but we will see that it is useful to insert them further towards the outer levels, if possible. It has been shown elsewhere that in fact can give an exponential speedup [4].

Definition 1. *The negation of a formula* $P. \phi$ *is denoted as* $\mathit{Neg}(P. \phi)$ *and constructed as follows. For each clause* C *introduce a fresh variable* n_C*. Construct the prefix of* $\mathit{Neg}(P. \phi)$ *from* P *inverting all the quantifiers in* P *and inserting each of the variables the variable* n_C *as existential after the variable with maximal level in* C*. Construct a matrix of* $\mathit{Neg}(P. \phi)$ *as the following set of clauses.*

$$\{\bar{l} \vee n_C \mid l \in C, C \in \phi\} \cup \left\{\bigvee\nolimits_{C \in \phi} \bar{n}_C\right\}$$

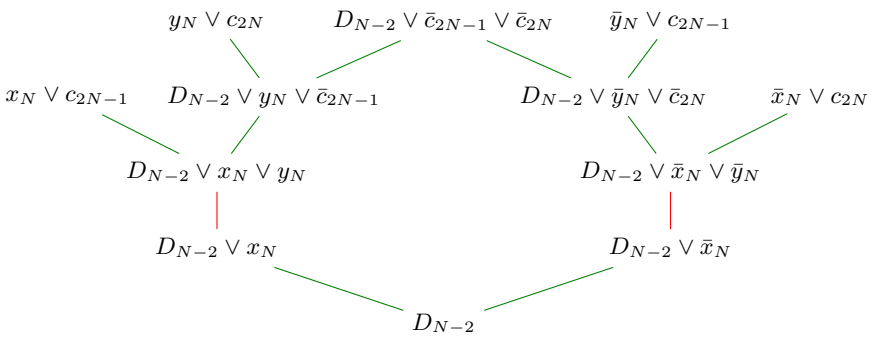


Fig. 1. Resolving away negation of (1) (where $D_{N-2} = \bar{c}_1 \vee \cdots \vee \bar{c}_{2N-3} \vee \bar{c}_{2N-2}$).

Example 1. The Neg $(\forall x \exists y \exists z.\, (\bar{x} \vee y) \wedge (x \vee z))$ is equal to $\exists x \forall y \exists c_1 \forall z \exists c_2.\, (x \vee c_1) \wedge (\bar{y} \vee c_1) \wedge (\bar{x} \vee c_2) \wedge (\bar{z} \vee c_2) \wedge (\bar{c}_1 \vee \bar{c}_2)$.

Clearly, Neg(Ψ) is false if and only if Ψ is true. We say that a QCNF Ψ is *negation-refuted* by a Q-resolution proof π iff π is a refutation of Neg(Ψ).

4.1 Negation-Refutation P-simulates Term-Resolution

The first question we should ask is whether for any term-resolution proof there is a polynomial-size negation-refutation proof. We show this is indeed the case.

Theorem 2. *Negation-refutation p-simulates term-resolution.*

Proof (sketch). Let π be a term-resolution proof of a QCNF $P.\, \phi$. Construct a Q-resolution of Neg$(P.\, \phi)$ as follows. Let M be a leaf of π. From the rules of term-resolution for each $C \in \phi$ there is a literal l s.t. $l \in C$ and $l \in M$. From the definition of Neg, the QCNF Neg$(P.\, \phi)$ contains the clause $\bar{n}_C \vee \bar{l}$ for such literal.

Starting with the clause $\bigvee_{C \in \phi} \bar{n}_C$, resolve each literal $\bar{n}_C$ with the clause $n_C \vee \bar{l}$, for each l s.t. $l \in C$ and $l \in M$. This results in the clause $\bigvee_{l \in M} \bar{l}$. Note that this clause does not contain contradictory literals because M must not contain contradictory literals.

Repeating this process for each leaf of π produces clauses that are negations of those leafs. Perform Q-resolutions steps and $\forall$-reductions as are done term-resolutions steps and $\exists$-reductions in π. This produces a Q-resolution proof where each derived clause is a negation of the corresponding term in π. Since π has the empty term in the root, the produced tree has the empty clause in the root. Resolutions needed to produce each of the leaf clauses requires at most $\min(|\pi|, |\,\mathsf{var}(\phi)|)$ steps thus the resulting Q-resolution is at most of size $(|\Phi| + |\pi|)^2$. □

4.2 Separation Between Term-Resolution and Negation-Refutation

The previous section shows that negation-refutation is at least as powerful as term-resolution. To show that the negation-refutation proof system is in fact stronger, we recall formula (1), whose term-resolution proofs are exponential, and show it has a negation-refutation proof of linear size.

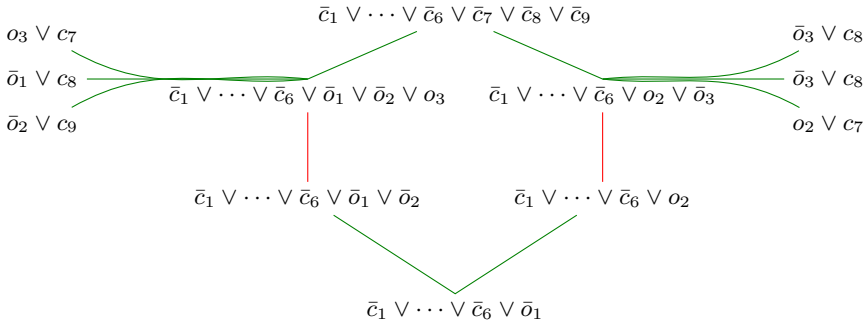


Fig. 2. Resolving definitions

Proposition 2. *Formula* (1) *has a linear negation-refutation proof.*

Proof (sketch). Negation of (1) introduces variables $c_1, \ldots, c_{2N}$ representing the respective clauses. In particular, the following clauses are constructed $x_i \vee c_{2i-1}$, $\bar{y}_i \vee c_{2i-1}$, $\bar{x}_i \vee c_{2i}$, $y_i \vee c_{2i}$ for $i \in 1..N$ and the clause $\bar{c}_1 \vee \cdots \vee \bar{c}_{2N}$. With the prefix $\exists x_1 \forall y_1 \exists c_1 c_2 \ldots \exists x_1 \forall y_1 \exists c_{2N-1} c_{2N}$.

We show how to resolve away the literals $\bar{c}_{2N-1}$ and $\bar{c}_{2N}$; the rest of the c_i literals is resolved in the same fashion. For conciseness we define D_{N-2} as $\bar{c}_1 \vee \cdots \vee \bar{c}_{2N-3} \vee \bar{c}_{2N-2}$. Figure 1 shows how $\bar{c}_{2N-1}$ and $\bar{c}_{2N}$ are replaced by y_N and x_N at which point y_N is universally reduced. Analogously, the literals are replaced with $\bar{x}_N$ and $\bar{y}_N$, which enables resolving x_N away.

Using this construction, each of the literals $\bar{c}_{2i-1}, \bar{c}_{2i}$ are resolved away in 7 resolution/reduction steps thus resulting in a resolution proof with $7N$ resolution/reduction steps in total. □

4.3 Variable Definitions

We observe that formula (1) is an example of a formula where an existential variable y is *defined*, i.e. the value of the variable is determined by values of some variables with a lower level (in the case of formula (1) the value of y_i is determined by the value of x_i). So the natural question to ask is whether any definition can be proven true by negation-refutation. We show that this is indeed the case but we will need *QU-resolution*—an extension of Q-resolution that enables resolving on *universal variables* [25].

We will demonstrate how negations of definitions can be refuted on the following representative example. Consider the prefix $\exists x_1 \forall x_2 \exists x_3 o_1 o_2 o_3$ and a matrix capturing the equalities $o_1 = \mathsf{NAND}(x_1, x_2)$, $o_2 = \mathsf{NAND}(x_2, x_3)$, and $o_3 = \mathsf{NAND}(o_1, o_2)$. These correspond to the following clauses (Tseitin variables that will be used for negating the clauses are indicated in parentheses).

(c_1) $\bar{x}_1 \vee \bar{x}_2 \vee \bar{o}_1$	(c_4) $\bar{x}_2 \vee \bar{x}_3 \vee \bar{o}_2$	(c_7) $\bar{o}_1 \vee \bar{o}_2 \vee \bar{o}_3$
(c_2) $x_2 \vee o_1$	(c_5) $x_2 \vee o_2$	(c_8) $o_1 \vee o_3$
(c_3) $x_2 \vee o_1$	(c_6) $x_2 \vee o_2$	(c_9) $o_1 \vee o_3$

After negating this formula, we obtain the following prefix.

$$\forall x_1 \exists x_2 \forall x_3 \forall o_1 \exists c_1 c_2 c_3 \forall o_2 \exists c_4 c_5 c_6 \forall o_3 \exists c_7 c_8 c_9$$

We omit the negated formula's matrix for succinctness. The Q-resolution proof proceeds in a similar fashion as the one for (1). Starting with the clause $\bar{c}_1 \vee \cdots \vee \bar{c}_9$, the $\bar{c}_i$ literals are resolved away, starting with the innermost ones.

Figure 2 shows a fragment of the proof, which resolves away the literals $\bar{c}_7, \ldots, \bar{c}_9$ (certain resolution steps are collapsed). Using the clauses determining the value of o_3, the proof generates the clauses $\bar{c}_1 \vee \cdots \vee \bar{o}_1 \vee \bar{o}_2$ and $\bar{c}_1 \vee \cdots \vee o_2$. Resolving these two clauses removes the variable o_2. Note that o_2 is universal, which is why we need QU-resolution. In order to resolve away o_1, the clause $\bar{c}_1 \vee \cdots \vee o_1$ is generated analogously. Leaving us with the clause $\bar{c}_1 \vee \cdots \vee \bar{c}_6$. Note that it was possible to $\forall$-reduce o_3 throughout the process because it is blocked only by the variables $c_7, \ldots, c_9$. In contrast, the variables o_1 and o_2 could *not* be $\forall$-reduced because they are blocked by the literals $\bar{c}_5, \ldots, \bar{c}_6$. The literals $\bar{c}_4, \ldots, \bar{c}_6$ and subsequently $\bar{c}_1, \ldots, \bar{c}_3$ are resolved in the same fashion.

An analogous proof can be carried out for any acyclic circuit of NAND gates. One picks a topological order of the gates and resolves them away as in the example above.

5 Summary, Conclusions, and Future Work

This paper investigates the strength of term-resolution: a well-established calculus for true quantified Boolean formulas. This paper exposes a significant vulnerability in the term-resolution calculus, which stems from the fact that the number of leafs of a term-resolution proof is not bound by the size of the formula in question. Instead, the model-generation rule enables generating new leafs of the proof from models of the matrix. The paper demonstrates that this lets us force the proof to generate exponentially many leafs by constructing QBF matrices with "many" universal literals.

This theoretical observation provides a further underpinning of the well-known observation that solving quantified Boolean formula with a CNF matrix can be sometimes particularly harmful [2,26]. Indeed, we demonstrate that even a very simple formula where each clause has only two literals leads to exponential term-resolution proofs.

At the practical level, in response to this issue, Zhang proposes to reason on a formula and on its negation at the same time [26]. This idea was realized with different flavors in various solvers [11,15,1]. The second part of this paper takes a similar avenue at the theoretical level. We compare the term-resolution calculus with the negation-refutation calculus, a calculus which refutes the formula's negation in order to show the formula true. The paper demonstrates that this proof system indeed has favorable theoretical properties, in particular it p-simulates term-resolution and there is an exponential separation between the two calculi.

This result is related to the well-known fact that enabling adding new variables in propositional resolution yields a more powerful proof system (extended resolution) [7]. Negation-refutation introduces new variables too. However, in extended resolution, the prover must come up with the variables' definitions. In negation-refutation, the definitions are determined by the clauses of the formula.

The last part of the paper touches upon some limitations of the negation-refutation calculus. If a variable's value is defined as a function of some other variables, through a Boolean circuit, we ask if it's possible to prove that it is always possible to come up with the right value for the variable being defined, i.e. complete the circuit. This is something that we would hope to be proven easily. We show that it is indeed possible to prove such definitions true linearly using negation-refutation but we show so with the use of QU-resolution—extension of Q-resolution that enables resolving on universal variables. This result is important from a theoretical perspective but raises further questions because existing QBF solvers use Q-resolution. It is the subject of future work to look for linear proofs for such formulas using only Q-resolution.

Acknowledgments. This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 and by FCT funding of LASIGE Research Unit, reference UID/CEC/00408/2013.

References

1. A. Goultiaeva, M. Seidl, A.B.: Bridging the gap between dual propagation and CNF-based QBF solving. In: In Proc. Intl. Conf. on Design, Automation & Test in Europe (DATE) (2013)
2. Ansótegui, C., Gomes, C.P., Selman, B.: The Achilles' heel of QBF. In: Veloso, M.M., Kambhampati, S. (eds.) AAAI. pp. 275–281. AAAI Press / The MIT Press (2005)
3. Balabanov, V., Widl, M., Jiang, J.H.R.: QBF resolution systems and their proof complexities. In: SAT. pp. 154–169 (2014)
4. Beyersdorf, O., Chew, L., Janota, M.: Extension variables in QBF resolution. In: Workshops at the Thirtieth AAAI Conference on Artificial Intelligence (2016)
5. Beyersdorff, O., Chew, L., Janota, M.: Proof complexity of resolution-based QBF calculi. In: Proc. Symposium on Theoretical Aspects of Computer Science (STACS). pp. 76–89. LIPIcs series (2015)
6. Brummayer, R., Lonsing, F., Biere, A.: Automated testing and debugging of SAT and QBF solvers. In: Strichman, O., Szeider, S. (eds.) SAT. pp. 44–57 (2010)
7. Cook, S.A.: A short proof of the pigeon hole principle using extended resolution. SIGACT News 8(4), 28–32 (Oct 1976)
8. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. J. Symb. Log. 44(1), 36–50 (1979)
9. Egly, U.: On sequent systems and resolution for QBFs. In: Cimatti, A., Sebastiani, R. (eds.) SAT. Lecture Notes in Computer Science, vol. 7317, pp. 100–113. Springer (2012)
10. Giunchiglia, E., Narizzano, M., Tacchella, A.: Clause/term resolution and learning in the evaluation of quantified Boolean formulas. Journal of Artificial Intelligence Research 26(1), 371–416 (2006)
11. Goultiaeva, A., Bacchus, F.: Exploiting QBF duality on a circuit representation. In: AAAI (2010)
12. Heule, M., Seidl, M., Biere, A.: A unified proof system for QBF preprocessing. In: Automated Reasoning – 7th International Joint Conference, IJCAR. vol. 8562, pp. 91–106. Springer (2014)
13. Kleine Büning, H., Bubeck, U.: Theory of quantified Boolean formulas. In: Handbook of Satisfiability. IOS Press (2009)
14. Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. Inf. Comput. 117(1) (1995)

15. Klieber, W., Sapra, S., Gao, S., Clarke, E.M.: A non-prenex, non-clausal QBF solver with game-state learning. In: SAT (2010)
16. Krajíček, J., Pudlák, P.: Quantified propositional calculi and fragments of bounded arithmetic. Mathematical Logic Quarterly 36(1), 29–46 (1990)
17. Lonsing, F.: Dependency Schemes and Search-Based QBF Solving: Theory and Practice. Ph.D. thesis, Johannes Kepler Universität (2012), http://www.kr.tuwien.ac.at/staff/lonsing/diss/
18. McMillan, K.L.: Interpolation and SAT-Based model checking. In: Jr., W.A.H., Somenzi, F. (eds.) CAV. Lecture Notes in Computer Science, vol. 2725, pp. 1–13. Springer (2003)
19. Plaisted, D.A., Greenbaum, S.: A structure-preserving clause form translation. J. Symb. Comput. 2(3), 293–304 (1986)
20. Silva, J.P.M., Lynce, I., Malik, S.: Conflict-driven clause learning sat solvers. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability, pp. 131–153. IOS Press (2009)
21. Silva, J.P.M., Sakallah, K.A.: Conflict analysis in search algorithms for satisfiability. In: ICTAI. pp. 467–469 (1996)
22. Tseitin, G.S.: On the complexity of derivations in the propositional calculus. Studies in Constructive Mathematics and Mathematical Logic (1968)
23. Urquhart, A.: The complexity of propositional proofs. Bulletin of the EATCS 64 (1998)
24. Van Gelder, A.: Decision procedures should be able to produce (easily) checkable proofs. In: Workshop on Constraints in Formal Verification (2002), (in conjunction with CP02)
25. Van Gelder, A.: Contributions to the theory of practical quantified Boolean formula solving. In: Milano, M. (ed.) CP. Lecture Notes in Computer Science, vol. 7514, pp. 647–663. Springer (2012)
26. Zhang, L.: Solving QBF by combining conjunctive and disjunctive normal forms. In: AAAI (2006)
27. Zhang, L., Malik, S.: Conflict driven learning in a quantified Boolean satisfiability solver. In: ICCAD (2002)