Short Proofs in QBF Expansion

Judith Clymo

University of Leeds

Quantified Boolean formulas (QBFs) generalise propositional logic by adding Boolean quantification. While not more expressive than propositional formulas, QBFs allow more succinct encodings of many practical problems. From a complexity point of view, they capture all problems from PSPACE.

Following the enormous success of SAT solving there has been increased attention on QBF solving in the past two decades. However, lifting the success of SAT to QBF presents significant additional challenges stemming from quantification, and solvers use quite different techniques to do this.

We consider two popular paradigms for solving QBF in prenex conjunctive normal form (PCNF). In the first, QBF solvers use *Conflict Driven Clause Learning (CDCL)* techniques from SAT solving, together with a 'reduction' rule to deal with universally quantified literals. In the second method, QBF solvers use quantifier expansion to remove the universally quantified literals in order to then use resolution on the formulas.

These two paradigms can be modelled by different QBF proof systems. Modern SAT solvers correspond to the Resolution proof system; similarly QBF solvers correspond to different QBF Resolution calculi. CDCL-style QBF (QCDCL) systems correspond to the QBF resolution system Q-resolution (Q-Res) [3], although algorithms implementing QCDCL, such as DepQBF, typically also implement additional reasoning techniques. In contrast, expansion solving builds on expansion QBF proof systems, with ∀Exp+Res as their base system [2]. In fact, ∀Exp+Res was originally developed to model RAReQS [2].

The proof systems ∀Exp+Res and Q-Res are known to be incomparable, i.e. there are families of QBFs that have polynomial-size refutations in one system, but require exponential size refutations in the other [2, 1]. As such we would not expect either QCDCL or expansion-based algorithms to be consistently stronger than the other, but would instead anticipate that solvers implementing the two systems would complement each other.

We examine the relationship between Q-Res and ∀Exp+Res under the natural and practically important restriction to families of QBFs with *bounded quantifier complexity*, which express exactly all problems from the polynomial hierarchy and thus cover most application scenarios. In this case, we show that (dag-like) Q-Res is p-simulated by ∀Exp+Res. The simulation increases in complexity as the number of quantification levels grows, and indeed there is an exponential separation between the two systems on a family of QBFs with an unbounded number of quantification levels [2]. The opposite simulation does not hold, there are families of QBFs with only three quantifier blocks that have short ∀-Red proofs but require exponential proofs in Q-Res (for one or two quantifier blocks the two systems are equivalent). For practitioners, our result points towards a potential advantage of expansion solving techniques over QCDCL solving and offers a partial explanation for the observation that "the performance of solvers based on different solving paradigms substantially varies on classes of PCNFs defined by their numbers of alternations" [4].

The result is shown via a direct construction, we create a $\forall$Exp+Res proof from a Q-Res proof. A natural way to transform a clause from a Q-Res refutation into a clause in a $\forall$Exp+Res refutation is to define some complete assignment α to the universal variables of the input formula that does not satisfy the clause. All universal literals are removed (since they are falsified under α), and each existential literal x is replaced by an annotated literal $x^{[\alpha]}$ ($[\alpha]$ indicates the restriction of α to those variables that appear before the annotated literal in the quantifier prefix). The difficulty is to ensure that the resolution steps in the attempted $\forall$Exp+Res proof are valid. In particular, in every resolution step the pivot literals must have the same annotation. It may be impossible to find suitable annotations for each clause in the given Q-Res proof that respect this restriction.

We can manage this problem by duplicating clauses (and their entire derivation) whenever multiple incompatible annotations would be required in a $\forall$-Red proof. Of course, such duplication of clauses may result in an exponential increase in the proof size if there are many different possible annotations to consider. However, if attention is restricted to QBFs that have $O(1)$ quantifier blocks, then we can carefully manage the process of duplicating clauses to support all the required annotations without an exponential increase in the proof size. We show that

Theorem 1. *Let Φ be a QBF with k quantification blocks and π a Q-Res proof of Φ. Then there is an $\forall$Exp+Res proof of Φ of size at most $|\pi|^{1+k/2}$.*

Since the cost of transforming a dag-like Q-Res proof into a $\forall$Exp+Res proof by this construction depends on the number of quantifier alternations it provides theoretical support for the observation that QCDCL solvers are more competitive on formulas with longer quantifier prefixes. For tree-like proofs our construction does not increase the size of the proof for any number of quantifier blocks, so it can also be used to show the simulation of tree-like Q-Res by tree-like $\forall$Exp+Res (similar to the technique in [2]).

References

1. Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. Proof complexity of resolution-based QBF calculi. In *Proc. Symposium on Theoretical Aspects of Computer Science*, pages 76–89. LIPIcs series, 2015.
2. Mikoláš Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theor. Comput. Sci.*, 577:25–42, 2015.
3. Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified Boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995.
4. Florian Lonsing and Uwe Egly. Evaluating QBF solvers: Quantifier alternations matter. *CoRR*, abs/1701.06612, 2017.