Lower Bound Techniques for QBF Proof Systems

Meena Mahajan
The Institute of Mathematical Sciences, HBNI, Chennai, India
meena@imsc.res.in

Abstract

How do we prove that a false QBF is indeed false? How big a proof is needed? The special case when all quantifiers are existential is the well-studied setting of propositional proof complexity. Expectedly, universal quantifiers change the game significantly. Several proof systems have been designed in the last couple of decades to handle QBFs. Lower bound paradigms from propositional proof complexity cannot always be extended - in most cases feasible interpolation and consequent transfer of circuit lower bounds works, but obtaining lower bounds on size by providing lower bounds on width fails dramatically. A new paradigm with no analogue in the propositional world has emerged in the form of strategy extraction, allowing for transfer of circuit lower bounds, as well as obtaining independent genuine QBF lower bounds based on a semantic cost measure.

This talk will provide a broad overview of some of these developments.

2012 ACM Subject Classification Theory of computation → Proof complexity, Theory of computation → Complexity theory and logic

Keywords and phrases Proof Complexity, Quantified Boolean formulas, Resolution, Lower Bound Techniques

Digital Object Identifier 10.4230/LIPIcs.STACS.2018.2

Category Invited Talk

Funding This work was partially supported by the EU Marie Curie IRSES grant CORCON.

Acknowledgements I thank Olaf Beyersdorff, from whom I learnt all I know about QBF proof complexity, and Anil Shukla, who helped immensely by learning along with me.

1 Introduction

Despite the NP-completeness of SAT, SAT solvers have proven to be highly successful in tackling humongous instances of satisfiability arising in practical applications. This has spurred more ambitious programs to develop practical solvers for more complex and expressive formulas. In the last couple of decades, several solvers have been developed to decide the truth or falsity of Quantified Boolean Formulas QBFs, a PSPACE-complete problem. As in the case of SAT, underlying the solvers are proof systems – formal systems where the truth or falsity of a QBF is established through a sequence of easily checkable steps. A natural measure of efficiency is the number of steps in such a proof, since it corresponds to the length of the run of the solver. Understanding the limitations of a solver is thus intimately connected to understanding the limitations of a proof system; hence the quest for explicit lower bounds in proof systems.

2 Proof systems for QBF

What does a typical proof system for QBFs look like? One could start with the standard propositional proof systems, where one proves that a formula is not satisfiable (that is, a QBF

35th Symposium on Theoretical Aspects of Computer Science (STACS 2018).
Editors: Rolf Niedermeier and Brigitte Vallée; Article No. 2; pp. 2:1–2:8
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

with only existentially quantified variables is false), and strengthen it to handle universally quantified variables. An obvious starting point is the well-studied resolution system Res, that works with unsatisfiable formulas in conjunctive normal form CNF. If a set $\mathcal{C}$ of clauses is simultaneously satisfiable by an assignment $\tilde{a}$, and if $\mathcal{C}$ contains clauses $A \vee x$, $B \vee \neg x$, then the set $\mathcal{C}' = \mathcal{C} \cup \{A \vee B\}$ is also satisfied by $\tilde{a}$. Using this resolution rule, in which we call x the pivot variable and $A \vee B$ the resolvent, we repeatedly enlarge the set of clauses until it contains the empty clause $\Box$; this set of clauses is patently unsatisfiable. This allows us to conclude that the original set of clauses must also have been unsatisfiable.

To strengthen this system to handle quantifications, it is useful to consider the evaluation game played on QBFs. We assume that the QBF is in prenex CNF: all variables are quantified first, and then there is a matrix of clauses. There are two players. The existential player owns the existentially quantified variables, the universal player owns the rest. The players step through the quantifier prefix, and as each variable is encountered, the player who owns it declares a value for it. The existential player wins a run of the game if the constructed assignment satisfies all clauses in the matrix; otherwise the universal player wins. A QBF is true if and only if the existential player has a winning strategy that allows him to win no matter how the universal player plays; it is false if and only if the universal player has a winning strategy. Thus, a strategy for the universal player, and a proof that it is a winning strategy, is a proof that the QBF is false.

We can now consider three different approaches to augmenting Res or other propositional proof systems to handle QBFs; more specifically, to handle universally quantified variables.

Eliminate-by-expansion

Remove the universal variables altogether! Use the semantics $\forall u F(u) \equiv F(0) \wedge F(1)$, but to avoid explosion of formula size, do the expansion on-the-fly, so to say. Since, in a run of the QBF game, values of existential variables can depend on those of the preceding universal variables, we make appropriate copies of existential variables, annotated with assignments to preceding universal variables. When using a clause from the matrix in the proof, the universal variables in the clause must be set to false. Other universal variables need not be set at this stage. That is, the annotations can be complete or partial. Now use standard resolution, keeping in mind that a single existential variable with two different annotations must be treated as two different variables.

The systems $\forall$Exp+Res ($\forall$Expansion + Resolution), IR (Instantiation + Resolution) are based on this idea; see [22] and [8]. The solvers CAQE [27], CEGAR [18], Ghost-Q [24], RAReQS [21] use such expansion-based ideas.

Eliminate-via-∀-reduction

Consider an intermediate stage during a run of the game on a QBF. If the partial assignment constructed so far results in a clause getting simplified to one with only un-assigned universal variables, then the universal player can win by simply falsifying this clause. This gives rise to the $\forall$-reduction rule that can be added on to any line-based propositional system. The simplest such system, and indeed, one of the earliest formal proof systems for QBFs, is the system Q-Res, see [23], where resolution can be performed on existential pivots, tautologies must be discarded, and a clause A can be inferred from a clause $A \vee u$ where u is a universal literal and all variables in A appear left of u in the prefix. A natural generalisation is QU-Res, where the pivot for resolution can also be universal, see [30]. A more informative name for QU-Res is perhaps Res + $\forall$Red (Resolution + $\forall$Reduction). Many DPLL-based solvers using conflict-driven-clause-learning, eg Evaluate, [15, 16], QuBe, [20] are based on such systems. In a similar vein, one could add a ∀ Reduction rule to Frege proof systems or to the Polynomial Calculus system and obtain proof systems sound and complete for QBFs; see [7].

Merging complementary literals

When performing resolution, tautologies are removed because they contribute nothing to proofs of unsatisfiability. In the case of QBFs, however, what looks like a tautology may not really be one. If a resolution rule produces a clause containing u and $\neg u$ for some universal variable u, such a clause could be still be useful, because the two complementary literals come from different sub-derivations, and a winning strategy could use this information to decide how to set u. So instead of discarding the clause, we retain a version of it, replacing the literals $u, \neg u$ with a single merged literal u^*. This is referred to as long-distance resolution. To preserve soundness of the rule, some side-conditions are imposed on when such a resolution and merging is permissible.

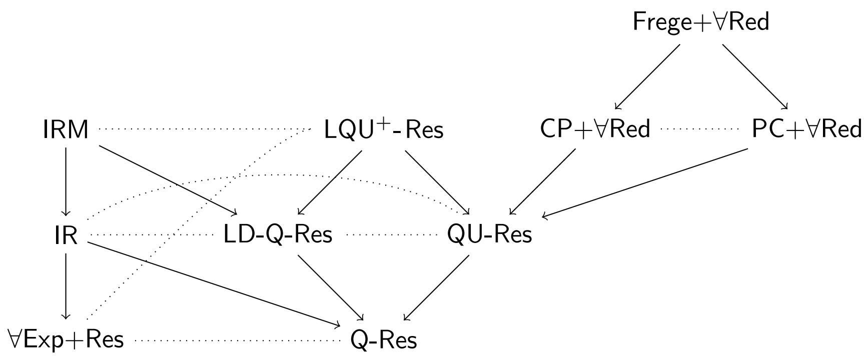
Augmenting Q-Res and QU-Res with long-distance resolution gives rise to the proof systems LD-Q-Res and LQU$^+$-Res; see [2, 3].

A similar situation can also arise in the expansion-based systems, where a variable appears in the resolvent with annotations that differ in their assignment to universal variables. Again, under appropriate side-conditions, a merger of the annotations is permissible. This gives rise to the proof system IRM (Instantiation, Resolution, Merge); see [8].

Many solvers use long-distance resolution very effectively; for instance, Quaffle [31].

The relative power of some QBF proof systems

The figure below shows some of the relationships between the QBF proof systems discussed. An arrow from A to B indicates that a proof in system B can be transformed (in polynomial time) to a proof in system A; that is, system A p-simulates system B. A dotted line indicates that neither p-simulates the other; they are incomparable.



3 Where the hardness comes from

3.1 Propositional hardness

An unsatisfiable CNF formula is a false QBF with only existential variables. A proof of its falsity in a QBF proof system is just a proof of unsatisfiability. If this is hard to demonstrate in the specialisation of the QBF proof system to purely existential formulas, then of course it is a hard QBF for the QBF proof system. Thus, for instance, the pigeonhole principle formula, asserting that $m+1$ pigeons can be placed in m holes without collision, is known to require exponentially long proofs in Res; trivially, it then requires exponentially long proofs in Q-Res and ∀Exp+Res. The Clique-color formulas, asserting that there is a k-colorable graph with a clique of size $k+1$, require exponentially long proofs in Cutting Planes; hence they require long proofs in CP+∀Red as well.

While propositional hardness is indeed a valid source of hardness for QBF proof systems, it does not tell us anything new about the ability (or lack thereof) of a QBF solver to handle universal variables. Thus in QBF proof complexity, such hardness is not particularly interesting.

3.2 Adapting techniques from propositional hardness

While pure propositional hardness may not be interesting, it is reasonable to expect that techniques used to establish such hardness could perhaps be adapted to prove non-trivial QBF hardness.

The size-width technique fails. The central lower-bound technique in the case of resolution is the relation between the size of proofs and their width (the maxmimum number of literals in any clause in the proof), due to [4]. The width of proofs also yields lower bounds on the space complexity; see [1]. Unfortunately, this technique fails completely in the case of the simplest extension, Q-Res, as shown in [10]; there are formulas with short proofs, derivable using very little space, but the width of any proof for these formulas must be large.

Feasible interpolation works. The technique of feasible interpolation exploits known (monotone) circuit lower bounds to obtain lower bounds for proofs of formulas of a specific type. It was used to show exponential lower bounds in the propositional proof systems Res and Cutting Planes, see [25, 26]. The technique can be adapted to similarly obtain lower bounds in QBF proof systems as well. The set-up is as follows: We start with a false QBF of the form

$$\varphi = \exists \vec{p}\ \mathcal{Q}\vec{q}\ \mathcal{Q}\vec{r}\ \cdot\ [A(\vec{p},\vec{q}) \wedge B(\vec{p},\vec{r})].$$

For every assignment $\vec{a}$ to the common variables $\vec{p}$, either $\mathcal{Q}\vec{q}\cdot A(\vec{a},\vec{q})$ or $\mathcal{Q}\vec{r}\cdot B(\vec{a},\vec{r})$ (or both) must be false. From a proof π that φ is false, we extract a circuit C in the $\vec{p}$ variables with the property that $C(\vec{a})=0$ implies $\mathcal{Q}\vec{q}\cdot A(\vec{a},\vec{q})$ is false and $C(\vec{a})=1$ implies $\mathcal{Q}\vec{r}\cdot B(\vec{a},\vec{r})$ is false. That is, C computes an interpolant. Furthermore, the size of C is polynomial in the size of π. Now, if interpolants for a formula are known to require large circuits, it follows that the formula cannot have small proofs. If the extracted circuit is monotone, then monotone circuit hardness gives a proof size lower bound.

In [11, 12] it is shown that all the Res-based QBF proof systems (Q-Res, QU-Res, LD-Q-Res, LQU^+-Res, ∀Exp+Res, IR, IRM), as well as the proof system CP+∀Red, admit monotone feasible interpolation. Hence the Clique-co-Clique formulas, asserting that a graph both has and does not have a large clique, are hard to prove false in these systems.

3.3 Winning strategies hard for decision lists

The universal player has a winning strategy in the evaluation game played on a false QBF. In general, a winning strategy may not be easy to compute. However, proofs in most proof systems reveal strategies. Let π be a proof of falsity in some QBF proof system of the form $P+\forall\text{Red}$, where P is some propositional proof system. By examining π, the player can figure out a way to compute a winning strategy. The computation focuses on the $\forall$ Reduction steps in the proof. Let $L_1, L_2, \ldots, L_m$ be the lines of the proof, with the last line being something obviously false (such as the empty clause $\Box$). Suppose it is the universal player's turn to play, and she has to choose a value for the variable u. All variables left of u in the prefix have already been assigned, and she knows the partial assignment $\vec{a}$. She now looks at the lines in the proof that are obtained by a $\forall$ reduction applied to the variable u. Let the indices of these lines be $(1 <) i_1 < \ldots < i_k (\leq m)$, and let each L_{i_r} be obtained from some L_{j_r}, $j_r < i_r$, by dropping u or $\neg u$. Note that since dropping u was permitted, each L_{i_r} is either true or false under $\vec{a}$; there are no variables still unassigned. She steps through the following decision list:

if	$L_{i_1}(\vec{a}) = 0$	**then**	set u to make $L_{j_1}(\vec{a}) = 0$
elseif	$L_{i_2}(\vec{a}) = 0$	**then**	set u to make $L_{j_2}(\vec{a}) = 0$
$\vdots$			
elseif	$L_{i_k}(\vec{a}) = 0$	**then**	set u to make $L_{j_k}(\vec{a}) = 0$
else			set $u = 0$.

This can be shown to be a winning strategy; see [9, 7]. Note that the kind of decisions to be made while computing it depend on the nature of the lines allowed in the proof system $P + \forall$Red.

Now, if the formula has the property that there is a unique winning strategy for some variable, and if this winning strategy function is known to require large decision lists of the type specified by the proof system, then it follows that the proof must have large size. What is more, it must be large not because of the steps from the propositional part P (although that too may be the case), but because it has many $\forall$Reduction steps. This may be the case even if there are very few universal variables. A simple example is the formula QParity: $\exists \vec{x}\ \forall z\ \exists \vec{t}\ A(\vec{x}, \vec{t}) \wedge (t_n \vee z) \wedge (\neg t_n \vee \neg z)$, where the clauses in A express the property that t_n computes the parity of $x_1, \ldots, x_n$. (for example, clauses equivalent to $t_2 = x_1 \oplus x_2$, $t_i = t_{i-1} \oplus x_i$ for $i > 2$). The only winning strategy for the universal player is to choose z to be the parity of $x_1, \ldots, x_n$. A Q-Res or QU-Res proof with S lines would give a decision list computing parity with at most S decision steps, where each decision involves evaluating a clause on an assignment. If S were polynomial in n, this would give a way of computing parity in AC^0, something we know is not possible. So this formula has no short proof in Q-Res or QU-Res ([9]). Similarly, a formula asserting that for some $\vec{x}, \vec{y}$, the inner product modulo 2 is both 0 and 1 is hard for CP+$\forall$Red ([12]) because the inner product function requires exponentially long decision lists of threshold functions ([29]).

See [9, 7, 12] for more examples of such hardness. In particular, it is noteworthy that this technique gives explicit lower bounds against proof systems $\mathrm{AC}^0[p]$-Frege+$\forall$Red for any prime p. In contrast, in the propositional world, the strongest lower bound holds for the system AC^0-Frege, while lower bounds for $\mathrm{AC}^0[p]$-Frege remains an outstanding open question.

The feasible interpolation technique for QBFs described earlier is essentially a special case of strategy extraction; see [11].

3.4 Winning strategies requiring varied responses

In [14], it is shown that hardness in Frege+$\forall$Red must stem from either propositional hardness in the system Frege or from a circuit lower bound. (Thus any hardness proof for Frege+$\forall$Red would constitute a major advance, either in proof complexity or in circuit complexity.) There are no other sources of hardness. This is not the case for weaker systems. Consider the formula QEQUALITY:

$$\exists x_1 \cdots x_n \forall u_1 \cdots u_n \exists t_1 \cdots t_n \left(\bigwedge_{i=1}^{n} (x_i \vee u_i \vee t_i) \wedge (\neg x_i \vee \neg u_i \vee t_i) \right) \wedge \left(\bigvee_{i=1}^{n} \neg t_i \right)$$

It has a unique winning strategy that is extremely simple to compute: just let $u_i = x_i$ for each $i \in [n]$. Thus the decision list approach cannot show that this formula is hard in any system. However, it turns out to require exponentially long proofs in QU-Res. The reason is as follows: the winning strategy has a range of size 2^n; all responses are required. However, the algorithm that extracts a winning strategy from a proof builds up responses line-by-line, with each line contributing a limited amount to the range. In particular, in QU-Res, each line is a clause and contributes a single response. It follows that there must be many lines.

Generalising this idea needs some work and yields the very elegant size-cost-capacity theorem, see [6], applicable to proof systems of the form $P + \forall\text{Red}$. The cost of a formula is the number of distinct responses required in any winning strategy. A caveat: it is important here that we only count responses to universal variables in a single quantification block (which block doesn't matter). The capacity of a proof is an upper bound on the number of responses that a single line in a proof can contribute. The theorem says that the cost of a formula is bounded above by the size of a proof times the capacity of the proof.

In the case of QU-Res and CP+∀Red, the capacity of any proof is simply 1, and thus the formula cost is itself a proof size lower bound. In the case of PC+∀Red, the capacity of a proof is no more than its size, and so proof size is at least square root of the formula cost.

The size-cost-capacity theorem has bene used to show families of random QBFs hard in QU-Res, CP+∀Red and PC+∀Red.

Paralleling the size-cost-capacity theorem, while dealing with expansion-based systems, strategy size and weight provide proof size lower bounds; these results are established by counting annotations, and are reported in [5].

A prominent example of hard QBFs are the KBKF formulas introduced in [23]. These are known to require large proofs in Q-Res and IR [23, 9], but have short proofs in QU-Res [30] and in LD-Q-Res [19]. The proofs of hardness are length and cumbresome, and essentially use an ad hoc combinatorial argument. The size-cost-capacity technique provides an alternative, and, arguably, more insightful, proof that the KBKF formulas require exponentially long proofs in Q-Res and that a doubled variant requires exponentially long proofs in QU-Res, CP+∀Red, PC+∀Red. Similarly, the strategy-size theorem provides a more insightful proof that the KBKF formulas require exponentially long proofs in IR.

4 Conclusion

QBF proof complexity is a relatively young field, but it has already thrown up very interesting insights and techniques. A good starting point to read about QBF proof complexity are the doctoral dissertations of Leroy Chew [17] and Anil Shukla [28], although there have already been quite a few advances after that, eg [13, 6, 5]. In this short article, I have not tried to be exhaustive, and I apologise to the readers who find their favourite papers missing.

References

1 Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74(3):323–334, 2008. URL: `https://doi.org/10.1016/j.jcss.2007.06.025`, `doi:10.1016/j.jcss.2007.06.025`.

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

3 Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In *Proceeding of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 154–169, 2014.

4 Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.

5 Olaf Beyersdorff and Joshua Blinkhorn. Genuine lower bounds for QBF expansion. In *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science (STACS)*, page to appear, 2018.

6 Olaf Beyersdorff, Joshua Blinkhorn, and Luke Hinde. Size, cost, and capacity: A semantic technique for hard random QBFs. In *Proceedings of the ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, page to appear, 2018.

7 Olaf Beyersdorff, Ilario Bonacina, and Leroy Chew. Lower bounds: From circuits to QBF proof systems. In *Proceedings of the ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 249–260. ACM, 2016.

8 Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. On unification of QBF resolution-based calculi. In *Proceedings of 39th International Symposium on Mathematical Foundations of Computer Science MFCS, Proceedings Part II, LNCS 8635*, pages 81–93, 2014.

9 Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. Proof complexity of resolution-based QBF calculi. In *Proceedings of the 32nd International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 76–89. LIPIcs, 2015.

10 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(1):1:1–1:26, Dec 2017. (preliminary version in STACS 2016).

11 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Feasible Interpolation for QBF Resolution Calculi. *Logical Methods in Computer Science*, Volume 13, Issue 2, 2017. (preliminary version in ICALP 2015).

12 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Understanding cutting planes for QBFs. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:37, 2017. (preliminary version in FSTTCS 2016).

13 Olaf Beyersdorff, Luke Hinde, and Ján Pich. Reasons for hardness in QBF proof systems. In *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2016)*, pages 14:1–14:14, 2017.

14 Olaf Beyersdorff and Ján Pich. Understanding Gentzen and Frege systems for QBF. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*, pages 146–155, 2016.

15 Marco Cadoli, Andrea Giovanardi, and Marco Schaerf. An algorithm to evaluate Quantified Boolean Formulae. In *Proceedings of the 15th National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference (AAAI)*, pages 262–267, 1998.

16 Marco Cadoli, Marco Schaerf, Andrea Giovanardi, and Massimo Giovanardi. An algorithm to evaluate Quantified Boolean Formulae and its experimental evaluation. *Journal of Automated Reasoning*, 28(2):101–142, 2002.

17 Leroy Chew. *QBF Proof Complexity*. PhD thesis, University of Leeds, UK, 2017.

18 Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement for symbolic model checking. *Journal of the ACM*, 50(5):752–794, 2003.

19 Uwe Egly, Florian Lonsing, and Magdalena Widl. Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In *19th International Conference* *on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, pages 291–308. Springer Berlin Heidelberg, 2013.

20 Enrico Giunchiglia, Massimo Narizzano, and Armando Tacchella. QuBE++: An efficient QBF solver. In *Formal Methods in Computer-Aided Design*, pages 201–213. Springer Berlin Heidelberg, 2004.

21 Mikolás Janota, William Klieber, João Marques-Silva, and Edmund M. Clarke. Solving QBF with counterexample guided refinement. In *Proceeding of the 15th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 114–128, 2012.

22 Mikolás Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theoretical Computer Science*, 577:25–42, 2015.

23 Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified Boolean formulas. *Information and Computation*, 117(1):12–18, 1995.

24 William Klieber, Samir Sapra, Sicun Gao, and Edmund M. Clarke. A non-prenex, non-clausal QBF solver with game-state learning. In *13th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 128–142, 2010.

25 Jan Krajíček. Interpolation theorems, lower bounds for proof systems and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997.

26 Pavel Pudlák. Lower bounds for resolution and cutting planes proofs and monotone computations. *The Journal of Symbolic Logic*, 62(3):981–998, 1997.

27 Markus N. Rabe and Leander Tentrup. CAQE: A certifying QBF solver. In *Formal Methods in Computer-Aided Design, (FMCAD)*, pages 136–143, 2015.

28 Anil Shukla. *On Proof Complexity for Quantified Boolean Formulas.* PhD thesis, The Institute of Mathematical Sciences, HBNI, India, 2017.

29 György Turán and Farrokh Vatan. Linear decision lists and partitioning algorithms for the construction of neural networks. In *Foundations of Computational Mathematics*. Springer, 1997.

30 Allen Van Gelder. Contributions to the theory of practical quantified Boolean formula solving. In *Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming (CP)*, pages 647–663, 2012.

31 Lintao Zhang and Sharad Malik. Conflict driven learning in a quantified Boolean satisfiability solver. In *International Conference on Computer Aided Design*, pages 442–449, 2002.