Models and Counter-Models of Quantified Boolean Formulas

Martina Seidl
Johannes Kepler University Linz, Austria

Abstract

Because of the duality of universal and existential quantification, quantified Boolean formulas (QBF), the extension of propositional logic with quantifiers over the Boolean variables, have not only solutions in terms of models for true formulas like in SAT. Also false QBFs have solutions in terms of counter-models. Both models and counter-models can be represented as certain binary trees or as sets of Boolean functions reflecting the dependencies among the variables of a formula. Such solutions encode the answers to application problems for which QBF solvers are employed like the plan for a planning problem or the error trace of a verification problem. Therefore, models and counter-models are at the core of theory and practice of QBF solving. In this invited talk, we survey approaches that deal with models and counter-models of QBFs and identify some open challenges.

2012 ACM Subject Classification Theory of computation → Automated reasoning

Keywords and phrases Quantified Boolean Formula, Solution Extraction, Solution Counting

Digital Object Identifier 10.4230/LIPIcs.SAT.2024.1

Category Invited Talk

Funding Parts of this work were supported by the LIT AI Lab founded by the State of Upper Austria.

1 Overview

The evaluation of a *quantified Boolean formula* (QBF) [8] is often seen as a two-player game between a universal and an existential player: given a QBF $\Phi = \forall X_1 \exists X_2 \ldots \exists X_n.\phi$ where X_i are disjoint sets of variables and ϕ is a propositional formula over these variables, the task is to decide if the QBF is true or false. The existential player aims at satisfying the formula by assigning values to the existentially quantified variables, while the universal player aims at falsifying ϕ by setting the universally quantified variables. The variables need to be assigned in the order as they occur in the prefix. If the formula is true under the chosen assignment, the existential player wins, otherwise the universal player wins. Overall, a QBF is true if and only if there is a winning strategy for the existential player and a QBF is false if and only if there is a winning strategy for the universal player. Nowadays, QBF solvers are applied for many applications [32], and in this context winning strategies play a crucial role. For example, in the context of formal synthesis, a winning strategy for the existential variables encodes the program that is synthesized from a given specification [9], or in the context of a planning problem, the winning strategy encodes the plan [31].

Winning strategies are also often called the *solutions* of a QBF, i.e., they are the *models of true QBFs* and, respectively, the *counter-models of false QBFs*. For a true QBF $\Phi = \forall X_1 \exists X_2 \ldots \exists X_n.\phi$, a model contains the information how to set the values of the existential variables X_{2i} based on the values of the universal variables X_{2j+1} with $0 \leq j < i$ such that ϕ evaluates to true under this assignment. Such a model can be represented either as a binary tree of a certain structure or as a set of Boolean functions, so-called *Skolem functions*. A Skolem function for an existential variable $x \in X_{2i}$ is a Boolean function over

27th International Conference on Theory and Applications of Satisfiability Testing (SAT 2024).
Editors: Supratik Chakraborty and Jie-Hong Roland Jiang; Article No. 1; pp. 1:1–1:7
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

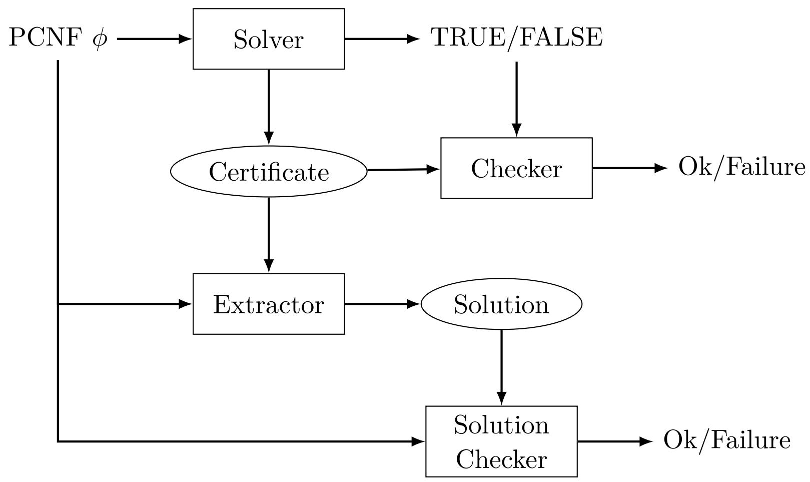

Figure 1 Workflow of solution extraction.

the universal variables X_{2j+1} with $0 \leq j < i$, i.e., the universal variables that precede x in the prefix. If all existential variables are replaced by their Skolem functions, the resulting propositional formula is valid.

Counter-models of false QBFs are defined dually. For a false QBF $\Phi = \forall X_1 \exists X_2 \ldots \exists X_n.\phi$, a counter-model contains the information how to set the values of the universal variables X_{2i+1} based on the values of the existential variables X_{2j} with $0 \leq j \leq i$ such that ϕ evaluates to false under this assignment. Such a counter-model can be represented either as a binary tree of a certain structure or as a set of Boolean functions, so-called *Herbrand functions*. A Herbrand function for an universal variable $x \in X_{2i+1}$ is a Boolean function over the existential variables X_{2j} with $0 \leq j \leq i$, i.e., the existential variables that precede x in the prefix. If all universal variables are replaced by their Herbrand functions, the resulting propositional formula is unsatisfiable.

In QBF research, much emphasis is set on the evaluation of QBFs, i.e., on deciding whether they are true or false. Less effort is set on obtaining models as counter-models despite they are very relevant for practical applications. This is underpinned by the fact that in recent QBF competitions [24, 28] there was no track involving solution extraction, although the benefits of being able to produce winning strategies are manifold. On the one hand models and counter-models can serve as a certificate confirming the correctness of a solving result with the aid of a SAT solver. On the other hand, they encode the solution to the application problem that was translated to a QBF. Furthermore, in the field of proof complexity, counter-models of false QBFs establish a strong tie between theory and practice of QBF solving. In this paper, we take a short tour through the works that explicitly deal with QBF models and counter-models and identify some perspectives for future work.

2 Solutions of Quantified Boolean Formulas

There exist some solvers that generate solutions at runtime. Almost 20 years back, the solver sKizzo was presented [5] which finds BDD-based models for true QBFs. Almost at the same time, the solvers Squolem and EBDDRES were presented that could also generate BDD-based solutions for true formulas [17]. At this time, it was not clear how to combine the generation of solution with clause/cube learning as successfully used in SAT. Later, the abstraction-based solvers CAQE and QuABs were developed that at least in some versions support the extraction of solutions [34]. The solver Cadet [29] for 2QBFs searches for Skolem functions by incrementally adding constraints until these constraints describe a model in terms of Skolem function or until it can be proven that it is not possible to construct a model, i.e., the formula is false. The QBF solver QFun employs machine learning techniques to find short Skolem and Herbrand functions [16].

For a very long time it was unclear how to generate solutions for solvers based on QCDCL, the QBF variant of conflict-driven clause learning [25] which is the major solving paradigm in SAT solving. Independently two approaches were presented that both rely on the fact that QCDCL is based on the Q-resolution calculus [19], the QBF version of resolution. A QCDCL solver like DepQBF [21] can directly emit clause Q-resolution proofs for false formulas and cube Q-resolution proofs for true formulas. For the extraction of models/counter-models, the applications of the QBF-specific existential reduction rule/universal reduction rule has to be taken into account. The approach of Goultiaeva and Van Gelder [12] interactively rewrites a Q-resolution proof as follows. Assume that the considered QBF is false and that the QBF solver produced a clausal Q-resolution refutation. If the outermost variables of the quantifier prefix are existential, then they can be assigned any value and the formula will still evaluate to false. Now some assignment to those variables has to be provided and is applied on the proof, i.e., the variables are assigned. The proof is then simplified resulting in a proof of the QBF under the respective assignment. Now it can be shown that when a universal variable of the new outermost block is eliminated by universal reduction, it occurs only in one polarity. This polarity determines the value of the universal variable. In this way, all values of the outermost universal variables can be directly read off from the proof. Again, the proof is simplified under the assignment resulting in a new proof. Next, the outermost existential variables assigned, resulting again in a proof from which the values of the next outermost universal variables can be read off. This procedure is repeated until all variables are assigned. The other approach by Balabanov and Jiang [1] traverses a Q-resolution proof in reverse topological order and builds Skolem functions from cubes on which existential reduction is applied in the case of true formulas and it builds Herbrand functions from clauses on which universal reduction is applied. In this way, Boolean functions are generated which are typically represented as And-Inverter-Graphs (AIGs). Figure 1 shows the workflow of solution generation after the actual solving: first, the QBF solver decides whether the QBF is true or false and produces a proof. This proof can be efficiently checked by an independent checker to validate the solving result. Furthermore, the proof is then analyzed and the Skolem/Herbrand functions are extracted. By replacing variables by their functions, a SAT solver can then be used to (1) check the correctness of the Skolem/Herbrand functions and (2) to confirm the solving result again. The complete tool chain is implemented in the QBFCert framework [26]. Solution extraction from proofs has been considered for other proof systems than the basic version of Q-resolution like QU-resolution and long-distance resolution [3, 2], for the expansion-based proof system $\forall Exp + Res$ and its extensions [7, 13] as well as for the QRAT proof system [15] mainly used for preprocessing. In [11], we presented an approach to combine the partial solutions obtained from preprocessing with solutions obtained from complete solvers. In practical QBF solving, the extraction of solution is also useful for debugging QBF encodings as suggested in [30].

When dealing with models and counter-models of QBFs as first-class objects, it also becomes relevant to ask for symmetries [18] leading to the distinction of syntactic and semantic symmetries as well as to ask for the overall number of solutions. First approaches to solution counting for true and false QBFs have been presented. First, only assignments to variables of the outermost quantifier block were considered [33, 4] using an enumerative approach, but later this work was lifted to variables at the second quantifier level [27]. Most recently, an efficient recursive approach has been introduced that computes the full model count of true QBFs [10].

The impact of extracting winning strategies is not only practically motivated, but it is also motivated by important results in the field of proof complexity (see [6] for a survey by Beyersdorff). Beyersdorff identifies strategy extraction as a distinctive feature of QBF proof systems for which no propositional analogue exists and states that most QBF lower-bound techniques employ strategy extraction.

3 Some Open Challenges

While there has been made considerable progress in many theoretical and practical aspects of QBF solving, there are still many open challenges that need to be addressed in the future. In the following, three of such challenges are shortly discussed.

Obtaining Solutions for True Formulas

Although solution extraction and generation should be dual for true and false formulas in theory, in practice there is a gap because of QBFs are usually provided in prenex conjunctive normal form (PCNF). The PCNF representation yields several advantages because of the easier implementation and because some techniques work only for clauses, but not for arbitrary formulas. At the same time, the PCNF representation also introduces a bias, because for true formulas having the formula in prenex disjunctive normal form (PDNF) would be preferable. Currently, the PDNF representation has to be constructed during the solving leading to large initial cubes or it involves an expensive transformation. To overcome this problem, it might be preferable to focus on the original structure of a formula and not to flatten it to an equivalent PCNF.

Solutions from Different Solving Paradigms

Over the last years, several different QBF solving paradigm have been shown to be orthogonal in their strength. Therefore, their integration is often beneficial either in terms of distributed portfolio solvers [14] or via well defined interfaces as suggested in [22, 20]. However, when combining multiple approaches, the extraction of solutions becomes challenging, because it is not clear how to define the interfaces at the solution level. We presented some work going in this direction in [11].

Solutions from Parallel and Distributed QBF Solving

Current work shows the potential of exploiting modern distributed and parallel hardware resources for QBF solving [23]. QBFs can be easily split into smaller subproblems that can be handled individually. In the case of SAT, also the solutions can be handled individually: if one of the subproblems is satisfiable, then its solution can be directly extended to the solution of the overall problem. If all subproblems are unsatisfiable, then there exists no solution. For QBFs, the situation is more complicated because the Skolem and Herbrand functions need to be assembled.

References

1 Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. *Formal Methods Syst. Des.*, 41(1):45–65, 2012. doi:10.1007/S10703-012-0152-6.

2 Valeriy Balabanov, Jie-Hong Roland Jiang, Mikolas Janota, and Magdalena Widl. Efficient extraction of QBF (counter)models from long-distance resolution proofs. In Blai Bonet and Sven Koenig, editors, *Proc. of the 29th AAAI Conference on Artificial Intelligence*, pages 3694–3701. AAAI Press, 2015.

3 Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In Carsten Sinz and Uwe Egly, editors, *Proc. of the 17th Int. Conf. on Theory and Applications of Satisfiability Testing (SAT 2014)*, volume 8561 of *Lecture Notes in Computer Science*, pages 154–169. Springer, 2014.

4 Bernd Becker, Rüdiger Ehlers, Matthew Lewis, and Paolo Marin. ALLQBF Solving by Computational Learning. In *Proc. of the 10th Int. Conf. on Automated Technology for Verification and Analysis (ATVA)*, volume 7561 of *LNCS*, pages 370–384. Springer, 2012.

5 Marco Benedetti. skizzo: A suite to evaluate and certify qbfs. In Robert Nieuwenhuis, editor, *Proc. of the 20th Int. Conf. on Automated Deduction (CADE-20)*, volume 3632 of *Lecture Notes in Computer Science*, pages 369–376. Springer, 2005.

6 Olaf Beyersdorff. *Proof Complexity of Quantified Boolean Logic – A Survey*, chapter Chapter 15, pages 397–440. World Scientific, 2023.

7 Olaf Beyersdorff, Leroy Chew, and Mikolas Janota. On unification of QBF resolution-based calculi. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Proc. of the 39th Int. Symposium on Mathematical Foundations of Computer Science (MFCS 2014)*, volume 8635 of *Lecture Notes in Computer Science*, pages 81–93. Springer, 2014.

8 Olaf Beyersdorff, Mikolás Janota, Florian Lonsing, and Martina Seidl. Quantified boolean formulas. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, pages 1177–1221. IOS Press, 2021.

9 Roderick Bloem, Robert Könighofer, and Martina Seidl. Sat-based synthesis methods for safety specs. In Kenneth L. McMillan and Xavier Rival, editors, *Proc. of the 15th Int. Conf. on Verification, Model Checking, and Abstract Interpretation (VMCAI 2014)*, volume 8318 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2014.

10 Florent Capelli, Jean-Marie Lagniez, Andreas Plank, and Martina Seidl. A top-down tree model counter for quantified boolean formulas. In *Accepted for IJCAI 2024*, 2024.

11 Katalin Fazekas, Marijn J. H. Heule, Martina Seidl, and Armin Biere. Skolem function continuation for quantified boolean formulas. In Sebastian Gabmeyer and Einar Broch Johnsen, editors, *Proc. of the 11th Int. Conf. on Tests and Proofs (TAP 2017)*, volume 10375 of *Lecture Notes in Computer Science*, pages 129–138. Springer, 2017.

12 Alexandra Goultiaeva, Allen Van Gelder, and Fahiem Bacchus. A uniform approach for generating proofs and strategies for both true and false QBF formulas. In Toby Walsh, editor, *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI 2011)*, pages 546–553. IJCAI/AAAI, 2011.

13 Vedad Hadzic, Roderick Bloem, Ankit Shukla, and Martina Seidl. Ferpmodels: A certification framework for expansion-based QBF solving. In Bruno Buchberger, Mircea Marin, Viorel Negru, and Daniela Zaharie, editors, *Proc. of the 24th Int. Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2022)*, pages 80–83. IEEE, 2022.

14 Maximilian Heisinger, Martina Seidl, and Armin Biere. Paraqooba: A fast and flexible framework for parallel and distributed QBF solving. In Sriram Sankaranarayanan and Natasha Sharygina, editors, *Proc. of the 29th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2023)*, volume 13993 of *Lecture Notes in Computer Science*, pages 426–447. Springer, 2023.

15 Marijn J. H. Heule, Martina Seidl, and Armin Biere. Solution validation and extraction for QBF preprocessing. *J. Autom. Reason.*, 58(1):97–125, 2017.

16 Mikolás Janota. Towards generalization in QBF solving via machine learning. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proc. of the 32nd AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pages 6607–6614. AAAI Press, 2018.

17 Toni Jussila, Armin Biere, Carsten Sinz, Daniel Kröning, and Christoph M. Wintersteiger. A first step towards a unified proof checker for QBF. In João Marques-Silva and Karem A. Sakallah, editors, *Proc. of the 10th Int. Conf. on Theory and Applications of Satisfiability Testing (SAT 2007)*, volume 4501 of *Lecture Notes in Computer Science*, pages 201–214. Springer, 2007.

18 Manuel Kauers and Martina Seidl. Symmetries of quantified boolean formulas. In Olaf Beyersdorff and Christoph M. Wintersteiger, editors, *Proc. of the 21st Int. Conf. on Theory and Applications of Satisfiability Testing (SAT 2018)*, volume 10929 of *Lecture Notes in Computer Science*, pages 199–216. Springer, 2018.

19 Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995.

20 Florian Lonsing, Fahiem Bacchus, Armin Biere, Uwe Egly, and Martina Seidl. Enhancing search-based QBF solving by dynamic blocked clause elimination. In Martin Davis, Ansgar Fehnker, Annabelle McIver, and Andrei Voronkov, editors, *Proc. of the 20th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-20)*, volume 9450 of *Lecture Notes in Computer Science*, pages 418–433. Springer, 2015.

21 Florian Lonsing and Uwe Egly. Depqbf 6.0: A search-based QBF solver beyond traditional QCDCL. In Leonardo de Moura, editor, *Proc. of the 26th Int. Conf. on Automated Deduction (CADE 26)*, volume 10395 of *Lecture Notes in Computer Science*, pages 371–384. Springer, 2017.

22 Florian Lonsing, Uwe Egly, and Martina Seidl. Q-resolution with generalized axioms. In Nadia Creignou and Daniel Le Berre, editors, *Proc. of the 19th Int. Conf. on Theory and Applications of Satisfiability Testing (SAT 2016)*, volume 9710 of *Lecture Notes in Computer Science*, pages 435–452. Springer, 2016.

23 Florian Lonsing and Martina Seidl. Parallel solving of quantified boolean formulas. In Youssef Hamadi and Lakhdar Sais, editors, *Handbook of Parallel Constraint Reasoning*, pages 101–139. Springer, 2018.

24 Florian Lonsing, Martina Seidl, and Allen Van Gelder. The QBF gallery: Behind the scenes. *Artif. Intell.*, 237:92–114, 2016.

25 João Marques-Silva, Inês Lynce, and Sharad Malik. Conflict-driven clause learning SAT solvers. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, pages 133–182. IOS Press, 2021.

26 Aina Niemetz, Mathias Preiner, Florian Lonsing, Martina Seidl, and Armin Biere. Resolution-based certificate extraction for QBF – (tool presentation). In Alessandro Cimatti and Roberto Sebastiani, editors, *Proc. of the 15th Int. Conf. on Theory and Applications of Satisfiability Testing (SAT 2012)*, volume 7317 of *Lecture Notes in Computer Science*, pages 430–435. Springer, 2012.

27 Andreas Plank, Sibylle Möhle, and Martina Seidl. Enumerative level-2 solution counting for quantified boolean formulas (short paper). In *Proc. of the 29th Int. Conf. on Principles and Practice of Constraint (CP)*, volume 280 of *LIPIcs*, pages 49:1–49:10. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

28 Luca Pulina and Martina Seidl. The 2016 and 2017 QBF solvers evaluations (QBFEVAL'16 and QBFEVAL'17). *Artif. Intell.*, 274:224–248, 2019.

29 Markus N. Rabe and Sanjit A. Seshia. Incremental determinization. In Nadia Creignou and Daniel Le Berre, editors, *Proc. of the 19th Int. Conf. on Theory and Applications of Satisfiability Testing (SAT 2016)*, volume 9710 of *Lecture Notes in Computer Science*, pages 375–392. Springer, 2016.

30 Irfansha Shaik, Maximilian Heisinger, Martina Seidl, and Jaco van de Pol. Validation of QBF encodings with winning strategies. In Meena Mahajan and Friedrich Slivovsky, editors, *Proc. of the 26th Int. Conf. on Theory and Applications of Satisfiability (SAT 2023)*, volume 271 of *LIPIcs*, pages 24:1–24:10. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

31 Irfansha Shaik and Jaco van de Pol. Classical planning as QBF without grounding. In Akshat Kumar, Sylvie Thiébaux, Pradeep Varakantham, and William Yeoh, editors, *Proc. of the 32nd Int. Conf. on Automated Planning and Scheduling (ICAPS 2022)*, pages 329–337. AAAI Press, 2022.

32 Ankit Shukla, Armin Biere, Luca Pulina, and Martina Seidl. A survey on applications of quantified boolean formulas. In *Proc. of the 31st IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI 2019)*, pages 78–84. IEEE, 2019.

33 Ankit Shukla, Sibylle Möhle, Manuel Kauers, and Martina Seidl. Outercount: A first-level solution-counter for quantified boolean formulas. In *Proc. of the 15th Int. Conf on Intelligent Computer Mathematics (CICM 2022)*, volume 13467 of *Lecture Notes in Computer Science*, pages 272–284. Springer, 2022.

34 Leander Tentrup. CAQE and quabs: Abstraction based QBF solvers. *J. Satisf. Boolean Model. Comput.*, 11(1):155–210, 2019.