International Workshop on Quantified Boolean Formulas,
September 2015, Austin, Texas, USA

Skolem functions computation for CEGAR based QBF solvers

Valeriy Balabanov[1], Jie-Hong Roland Jiang[2], and Christoph Scholl[3]

[1]Academia Sinica, [2]National Taiwan University, [3]University of Freiburg
{balabasik@gmail.com, jhjiang@ntu.edu.tw, scholl@informatik.uni-freiburg.de}

Abstract. In this work we propose an approach to extract Skolem-functions from CEGAR based QBF solvers (e.g., RareQS [4]) for true QBF formulas containing 2 or 3 quantification levels. We as well propose some optimizations to improve extracted certificates and perform detailed experimental evaluation.

1 Introduction

Recent QBF solvers evaluation verified the robustness of CEGAR based QBF solvers. On contrary to search-based approaches (e.g., DepQBF [5]), however, there exists no methodology to certify their answer with semantic winning strategies in a closed form (e.g., Skolem-functions for true QBFs, which are essential for many QBF applications). CEGAR based QBF solver RareQS [4], can produce partial winning moves for both existential and universal players at each turn of an abstraction-refinement game [4]. One straightforward use of this ability is that RareQS returns the winning assignment to outermost existential variables for true QBFs upon completion. In this work we describe how to construct full Skolem-functions models for QBFs, based on partial winning moves information emitted by RareQS. Currently our algorithm is limited to two and three level true QBFs (with an innermost quantification level to be existential), but preliminary analysis confirms the existence of an extension to arbitrary QBFs, based on the given approach and interpolation.

2 The main section

Preliminaries. We consider QBFs in PCNF through this work. All Boolean notations follow standard semantics. Given a 3QBF $\Phi = \exists\vec{w}\forall\vec{x}\exists\vec{y}.\phi(\vec{w},\vec{x},\vec{y})$, it is true (or valid) if and only if there exists a set of constant functions $S_{\vec{w}}$, and a set of functions $S_{\vec{y}}(\vec{x})$ (i.e., depending on $\vec{x}$), such that $\phi(S_{\vec{w}},\vec{x},S_{\vec{y}})$ is a tautology. $S_{\vec{w}}$ and $S_{\vec{y}}$ form the so-called Skolem-functions model, certifying the validity of Φ. There are other forms of certificates (e.g., Q-resolution). For more details on QBF certification please refer to [2].

Construction procedure. First consider a true 2QBF formula Φ. Assume that RareQS QBF solver (please refer to Algorithm 2 in [4]) needs three refinement loops to prove its validity. It means that three candidate solutions for the universal player were found, leading to existential counterexamples $\vec{\epsilon}_1$, $\vec{\epsilon}_2$ and $\vec{\epsilon}_3$. Let Φ_{cof} be the refined by three corresponding cofactors formula, as shown below.

$$\Phi = \forall\vec{x}\exists\vec{y}.\Phi(\vec{x},\vec{y}) \qquad \Phi_{cof} = \forall\vec{x}.\{\Phi(\vec{x},\vec{\epsilon}_1) \vee \Phi(\vec{x},\vec{\epsilon}_2) \vee \Phi(\vec{x},\vec{\epsilon}_3)\}$$

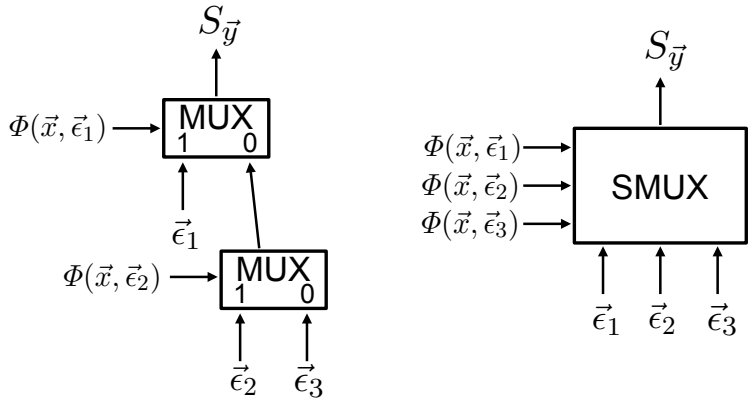

Fig. 1. Multiplexer construction [on the left], SMUX cell [on the right].

Now the search for a candidate solution fails, i.e., the universal player does not find a candidate solution which falsifies all cofactors generated so far. Consequently Φ_{cof} is true, which is determined by an unsatisfiable SAT call $\neg\Phi_{cof}$ (which is propositional as it has only existentially quantified variables $\vec{x}$).

Effectively, validity of Φ_{cof} says that an arbitrary assignment to $\vec{x}$ is included in $ON(\Phi(\vec{x},\vec{\epsilon}_1))$, $ON(\Phi(\vec{x},\vec{\epsilon}_2))$, or $ON(\Phi(\vec{x},\vec{\epsilon}_3))$. This information in fact is sufficient to get Skolem functions $S_{\vec{y}}(\vec{x})$ for any assignment $\vec{\alpha}$ to $\vec{x}$, by the following steps:

1. For all $\vec{\alpha} \in ON(\Phi(\vec{x},\vec{\epsilon}_1))$ we define $S_{\vec{y}}(\vec{\alpha}) = \vec{\epsilon}_1$.
2. For all $\vec{\alpha} \in ON(\Phi(\vec{x},\vec{\epsilon}_2)) \setminus ON(\Phi(\vec{x},\vec{\epsilon}_1))$ we define $S_{\vec{y}}(\vec{\alpha}) = \vec{\epsilon}_2$.
3. For all $\vec{\alpha} \in ON(\Phi(\vec{x},\vec{\epsilon}_3)) \setminus (ON(\Phi(\vec{x},\vec{\epsilon}_1)) \cup ON(\Phi(\vec{x},\vec{\epsilon}_2)))$ we define $S_{\vec{y}}(\vec{\alpha}) = \vec{\epsilon}_3$.

The above computation of the Skolem functions is visualized by a multiplexer construction as shown on the left of Fig. 1. We abbreviate the multiplexer construction by a cell "SMUX" which means that we have a series of multiplexers defining a prioritization in case that the sets $ON(\Phi(\vec{x},\vec{\epsilon}_i))$ overlap. SMUX cell is shown on the right of Fig. 1. By the following proposition we ensure the soundness of returned Skolem functions.

Proposition 1. *Constructed functions $S_{\vec{y}}(\vec{x})$ form a valid model for Φ.*

Proof. Φ_{cof} is true, therefore for every assignment $\vec{\alpha}$ to $\vec{x}$, some $\Phi(\vec{\alpha},\vec{\epsilon}_i))$ $(i \in [1..3])$ must be true. Our construction ensures that $S_{\vec{y}}(\vec{\alpha}) = \vec{\epsilon}_i$, i.e., that $\Phi(\vec{\alpha}, S_{\vec{y}}(\vec{\alpha}))$ evaluates to true. By definition, constructed $S_{\vec{y}}$ form a valid set of Skolem functions. □

Clearly, proposed construction procedure can be extended to true 2QBFs with an arbitrary number of refinement steps. Procedure below further extends approach for true 3QBFs. Suppose we are given a true 3QBF $\Phi = \exists\vec{w}\forall\vec{x}\exists\vec{y}.\Phi(\vec{w},\vec{x},\vec{y})$, and a winning move (assignment) $\vec{\beta}$ for $\vec{w}$ variables (which is returned upon completion of RareQS as a byproduct of solving process). 2QBF $\Phi = \forall\vec{x}\exists\vec{y}.\Phi(\vec{\beta},\vec{x},\vec{y})$ must be true, therefore Skolem function $S_{\vec{y}}$ can be extracted using previous 2QBF method. The complete Skolem model now consists of $\{S_{\vec{w}} = \vec{\beta}, S_{\vec{y}}\}$. Efficient implementation of this extension is slightly more elaborated, but conceptually it is as described above.

Optimizations. Below we propose three optimizations in order to minimize certificates returned by the Skolem functions construction procedure.

1. Please note that any order of switches in a SMUX cell leads to a valid set of Skolem functions. For the moment we allow an option to use cofactors either in forward, or backward order. Backward order is used as an empirical optimization.

2. We observed that cofactors often share identical clauses, therefore we implemented a hashing procedure that detects and substitutes the repeated clauses.

3. Although each next counterexample returned by CEGAR QBF solving approach covers at least one new (unblocked so far) universal move candidate, in practice it happens that older cofactors are fully covered by newer ones. The problem of identifying the redundant cofactors can be done using the so-called group minimal unsatisfiability subset extraction (group MUS, or GMUS). GMUS framework allows to partition CNF into groups (cofactors in our case), and return the minimal subset of them, which is still unsatisfiable (which is clearly a requirement for our extracted Skolem functions to be sound). More information on GMUS extraction can be found at [6].

3 Implementation and Experiments

We patched RareQS solver to emit countermoves associated with the CEGAR QBF solving process. Proposed in Section 2 algorithm was implemented into a tool CegarSkolem[1]. In order to test the unsat core optimization from Section 2 we used Haifa-HLMUC group MUS extractor [6]. Experimental setup consisted of QBFs from "QBFLIB" and "Applications" tracks taken from QBF Gallery 2014 [1]. We used DepQBF QBF solver [5] and ResQu [2] to compare CegarSkolem against existing Q-resolution based model computation in search-based QBF-solvers framework. All the experiments were performed on a Linux machine with Xeon 2.3 GHz CPU and 32Gb of RAM. All the tools were limited to 4Gb memory limit and 900 seconds time limit. An additional limitation of 1 Gb was imposed on Q-resolution proofs produced by DepQBF and moves information emitted by RareQS.

29 and 137 true 3QBFs either solved by DepQBF or RareQS, were chosen for experiments from "QBFLIB" and "Applications" tracks, respectively. Table 1 shows the solving and certification time statistics (rightmost "#cert" and "#mincert" columns stand for CegarSkolem w/o and w/ optimization heuristics, respectively). Note that as certification requires additional effort, both DepQBF and RareQs were not able to solve some of the instances that they could solve w/o certification. As we could see in general RareQS solved more instances, but DepQBF has much smaller runtime-per-instance. On contrary, even with optimizations, CegarSkolem constructed Skolem functions much quicker than ResQu, which is explained by large overhead in the size of Q-refutations produced by DepQBF, in comparison to (relatively small) number of existential counterexamples emitted by RareQS.

[1] CegarSkolem and other tools used in the experiments could be found here: `https://www.dropbox.com/s/qp18ovdzgyo3zuu/cegar_skolem_tools.zip?dl=0`

Table 1. Solving and certification statistics.

	DepQBF+ResQu				RareQS+CegarSkolem					
	#solved	time, s	#cert	time, s	#solved	time, s	#cert	time, s	#mincert	time, s
QBFLIB [29]	15	424.1	14	943.2	19	2710.6	19	49.5	19	67.5
Applications [137]	94	457.9	86	5162.8	102	4351.6	97	533.8	97	589.0

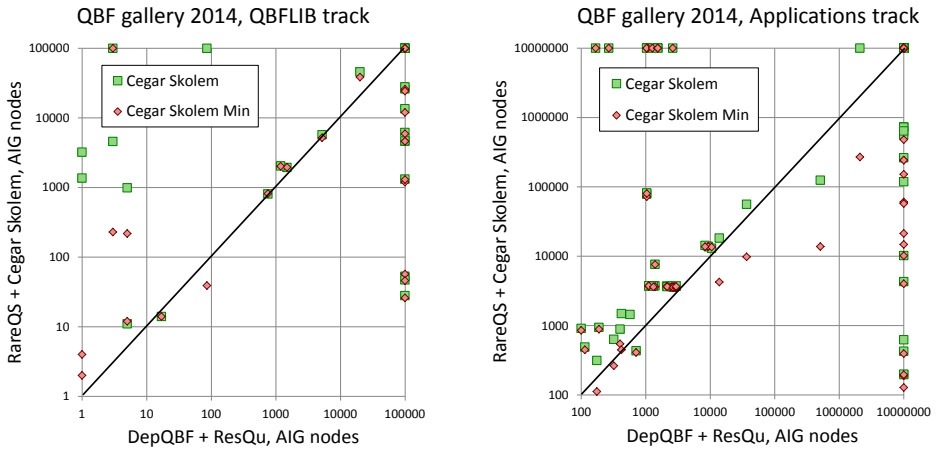

Fig. 2. Comparison of Skolem functions AIG sizes.

Fig. 2 compares certificates quality in terms of numbers of and-inverter-graphs (AIG) nodes (after minor AIG synthesis in tool ABC [3]). X-axis in figures corresponds to certificates produced by ResQu, while Y-axis corresponds to those by CegarSkolem and CegarSkolemMin. We do not provide a detailed statistics on the impact of various optimizations we have in CegarSkolemMin, but as one can see from Table 1 the computational overhead they introduce is small anyway. The certificate sizes, on the other hand, are reduced much in some cases.

Another observation to make is that certificates for DepQBF and RareQS are quite scattered across the figures. This means that for some benchmarks there exist simple Skolem-functions found by ResQu but not found by Cegar-Skolem and vice-versa. This phenomenon shall be investigated in the future work.

4 Conclusions and Future Work

In this work we proposed an algorithm for extraction of Skolem functions from true QBFs with 2 and 3 quantification levels. Experimental evaluation proves the robustness of the approach, and shows improvement over search-based certification/extraction procedures. Our main future goal is to extend the presented in this work extraction procedure for arbitrary (true, false, arbitrary number of levels) QBFs. Preliminary analysis confirms the existence of such an extension, based on the given approach and interpolation. Certificate minimization is another direction to pursue in the future.

References

1. QBF Gallery 2014. `http://qbf.satisfiability.org/gallery/`.
2. V. Balabanov and J.-H. R. Jiang. Unified QBF Certification and Its Applications. *Formal Methods in System Design*, 41:45–65, 2012.
3. Berkeley Logic Synthesis and Verification Group. ABC: A System for Sequential Synthesis and Verification. `http://http://www.eecs.berkeley.edu/~alanmi/abc/`.
4. M. Janota, W. Klieber, J. Marques-Silva, and E. Clarke. Solving QBF with Counterexample Guided Refinement. In *International conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 7317 of *LNCS*, pages 114–128. Springer, 2012.
5. F. Lonsing and A. Biere. DepQBF: A Dependency-Aware QBF Solver (System Description). *Journal on Satisfiability, Boolean Modeling and Computation*, 7:71–76, 2010.
6. V. Ryvchin and O. Strichman. Faster extraction of high-level minimal unsatisfiable cores. In *Theory and Applications of Satisfiability Testing - SAT 2011 - 14th International Conference, SAT 2011, Ann Arbor, MI, USA, June 19-22, 2011. Proceedings*, pages 174–187, 2011.