Old Challenges and New Solutions: a Comprehensive Assessment of State-of-the-Art QBF Solvers

P. Marin[2], M. Narizzano[1], L. Pulina[3], A. Tacchella[1], and E. Giunchiglia[1]

[1] DIBRIS, Università di Genova, Via Opera Pia, 13 – 16145 Genova – Italy
{giunchiglia,narizzano,tacchella}@unige.it

[2] Lehrstuhl für Rechnerarchitektur, Georges-Köhler-Allee 051 – 79110 Freiburg i.B. – Germany marin@informatik.uni-freiburg.de

[3] POLCOMING, Università di Sassari, Viale Mancini n. 5 – 07100 Sassari – Italy
lpulina@uniss.it

Since the first QBF Evaluation (QBFEVAL'03) [1], whose aim was to assess the relatively young state of the art in the QBF reasoning field, almost every year an evaluation event was organized. The purpose of that was to measure the progress in QBF reasoning techniques and encourage the submission of new problems which could be encoded in QBF. A report of the last QBFEVAL event in the series was published in [2]. After more than a decade of new solvers being developed and new challenging problems being proposed, we believe that QBFEVAL and, more recently, QBF Gallery [3] events offer a series of snapshots about QBF solving and related aspects, but somehow fail to provide a long-term picture about what has been achieved, which of the techniques proposed are still worth considering, and which problems are still relevant for current QBF solvers. In this work we gather numerous results which enable us to assess the contributions of complete off-the-shelf QBF solving tools to the state-of-the-art considering the whole course of QBFEVAL and QBF Gallery evaluations and competitions, and exposing the result in a historical perspective. We are then able to suggest potential research directions for solving older and actual challenging problems. In the following, we list the solvers and describe the problems we used in our experiments, then we present the results, and conclude with some final remarks.

To accomplish our task we considered some *legacy solvers* (S-LEGACY in the following), i.e., tools that were proposed in the literature, but are not considered in more recents comparative events and *new solvers* (S-NEW in the following), i.e., all the other tools that we consider and which are not legacy. In particular, out of 8 solvers considered, the legacy ones are AIGSOLVE [4], the only AIG-based QBF solver; AQME [5], a multi-engine tool whose back-end solvers were released in 2006; QUANTOR [6], QUBE [7], and SKIZZO [8], which implement key QBF solving techniques like resolution and expansion, DLL-search, and skolemization, respectively; lastly, STRUQS [9], which dynamically combines very different solution techniques. These tools are chosen among winners of at least one category in the past QBFEVAL events, conditioned to their maintenance ending before 2010. The set of new solvers is assembled by including the winners of the last QBF Gallery 2014: DEPQBF [10], GHOSTQ [11, 12] and RAREQS [11]. We did not consider HIQQER [13] because we could not find a version available for download. We define the "state-of-the-art" (SOTA) solver abstraction, i.e., the ideal system that always fares the best time among the systems in our portfolio. Likewise SOTA-LEGACY and SOTA-NEW abstract from legacy and new solvers, to score the global performace of solvers in S-LEGACY vs. those in S-NEW. As for problems, we consider three pools assembled for previous evaluations. In particular, we consider from [13] (*i*) QBF Gallery 2014 Track 1 (276 instances) and (*ii*) Track 2 (735 instances), and (*iii*) challenging formulas which are those classified as "Hard" (unsolved) and "Medium-Hard" (solved by one tool only) in the QBFEVAL evaluations from 2004 to 2010. These are then split by year. Overall, the testset is purposefully biased towards recently submitted instances, in order to (try to) assess legacy solvers on problems that are probably "unseen" to them, i.e., for which their developers did not have a chance to optimize the solver. On the other hand, group (*iii*) lets us assess whether the progress advertised by more recent evaluations is due to novel solving techniques, or to the fact that some hard problems were no longer evaluated. The tools were fed with the QBF formulas in their original format, i.e., we made no external preprocessing. Yet, some tools apply preprocessing techniques before the complete solving phase.

Considering the pool QBF Gallery Track 1, only 6 solvers out of 9 were able to solve at least 25% of the test set. By ranking the solvers according to the number of problems solved, the first is AIGSOLVE, which can solve about 42% of the test set, followed by QUBE and AQME, closely followed by GHOSTQ. Taking the above cited abstractions, SOTA was able to cope with about 73% of QBFG-T1. Its main contributors are AIGSOLVE, DEPQBF, and RAREQS.We also report that SOTA-LEGACY solves about 31% more problems than SOTA-NEW. Considering the pool QBF Gallery 2014 Track 2, which was partitioned into 6 families, AIGSOLVE, RAREQS, and QUANTORshow 3 times into the top-three ranking, AQME, DEPQBF, GHOSTQ, and STRUQS twice, and QUBE once. RAREQS and STRUQS can uniquely solve 16 problems, AIGSOLVE 12. Each of the remaining solvers less than 4. Lastly, we consider the challenging formulas from the past (6) QBFEVAL events: AIGSOLVE is always in the top-three and can solve uniquely 114 formulas, GHOSTQ and QUBE get the best rankings 5 times and can uniquely solve resp. 114 and 47 instances. Notice that RAREQS appears only once, but can solve uniquely 71 formulas. Overall, 2 out of 3 systems are always S-LEGACY, only in 2006 and 2010 the best solver is GHOSTQ. SOTA is able to solve 83% of the 2006 dataset, in other cases no more than 75%. With the notable exception of 2010, SOTA-LEGACY outperforms SOTA-NEW for each year. More details of this analysis are listed in the Appendix.

In the paper we have shown the results of a massive evaluation of QBF solvers and benchmarks from an historical perspective, and what emerges is that new systems are the main contributors of a SOTA solvers, yet comparing the SOTA-LEGACY and SOTA-NEW abstractions we have also shown that legacy systems still outperform the new ones in many problem categories. Therefore, we believe that it would be interesting to fuse legacy techniques into new systems in order to really push forward the state of the art in the QBF arena.

References

1. Berre, D.L., Simon, L., Tacchella, A.: Challenges in the QBF arena: the SAT'03 evaluation of QBF solvers. In: Sixth International Conference on Theory and Applications of Satisfiability Testing (SAT 2003). Volume 2919 of Lecture Notes in Computer Science., Springer Verlag (2003) 468–485
2. Peschiera, C., Pulina, L., Tacchella, A., Bubeck, U., Kullmann, O., Lynce, I.: The seventh qbf solvers evaluation (qbfeval10). In: Theory and Applications of Satisfiability Testing–SAT 2010, Springer Berlin Heidelberg (2010) 237–250
3. F. Lonsing, M. Seidl, A.V.G.: QBF gallery 2013 (2013) `http://www.kr.tuwien.ac.at/events/qbfgallery2013/`.
4. Pigorsch, F., Scholl, C.: An aig-based qbf-solver using sat for preprocessing. In: Design Automation Conference (DAC), 2010 47th ACM/IEEE, IEEE (2010) 170–175
5. Pulina, L., Tacchella, A.: A self-adaptive multi-engine solver for quantified Boolean formulas. Constraints **14**(1) (2009) 80–116
6. Biere, A.: Resolve and Expand. In: Seventh Intl. Conference on Theory and Applications of Satisfiability Testing (SAT'04). Volume 3542 of LNCS., Springer Verlag (2005) 59–70
7. Giunchiglia, E., Marin, P., Narizzano, M.: Qube7.0. JSAT **7**(2-3) (2010) 83–88
8. Benedetti, M.: Evaluating QBFs via Symbolic Skolemization. In: Eleventh International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR 2004). Volume 3452 of Lecture Notes in Computer Science., Springer Verlag (2004)
9. Pulina, L., Tacchella, A.: A structural approach to reasoning with quantified Boolean formulas. In: 21st International Joint Conference on Artificial Intelligence (IJCAI 2009). (2009) 596–602
10. Lonsing, F., Biere, A.: Depqbf: A dependency-aware QBF solver. JSAT **7**(2-3) (2010) 71–76
11. Janota, M., Klieber, W., Marques-Silva, J., Clarke, E.: Solving qbf with counterexample guided refinement. In: Theory and Applications of Satisfiability Testing–SAT 2012, Springer Berlin Heidelberg (2012) 114–128
12. Klieber, W., Sapra, S., Gao, S., Clarke, E.: A non-prenex, non-clausal qbf solver with game-state learning. In: Theory and Applications of Satisfiability Testing–SAT 2010. Springer (2010) 128–142
13. Jordan, C., Seidl, M.: The QBF Gallery 2014 (2014)

Appendix

This Appendix shows the detailed results of our empirical analysis. We ran all the experiments on a cluster of Intel Xeon E31245 PCs at 3.30 GHz equipped with 64 bit Ubuntu 12.04. Each solver was limited to 600s of CPU time and 4GB of memory.

Solver	Total		True		False		Unique	
	#	Time	#	Time	#	Time	#	Time
AIGSolve	116	5333.01	56	2177.45	60	3155.56	22	1458.26
QuBE	106	8764.73	53	3997.78	53	4766.95	8	1195.58
aqme	97	3287.20	39	1098.00	58	2189.20	–	–
ghostq	91	4814.73	48	2912.38	43	1902.17	4	158.97
depqbf	88	2388.32	39	1163.15	49	1225.17	5	454.77
rareqs	79	2588.64	32	1593.25	47	995.39	6	787.33
sKizzo	51	948.81	18	556.76	33	392.06	–	–
quantor	50	1498.37	28	911.72	22	586.65	2	161.67
StruQS	43	6092.64	31	4052.98	12	2039.66	1	16.53

Table 1. Performance of the involved solvers on qbfg-t1. The table consists of nine columns that for each solver reports its name (column "Solver"), the total number of instances solved and the cumulative time to solve them (columns "#" and "Time", group "Total"), the number of instances found satisfiable and the time to solve them (columns "#" and "Time", group "True"), the number of instances found unsatisfiable and the time to solve them (columns "#" and "Time", group "False"), and, finally, the number of instances uniquely solved and the time to solve them (columns "#" and "Time", group "Unique"); a "–" (dash) means that the solver did not solve any instance. Finally, the table is sorted in descending order, according to the number of instances solved, and, in case of a tie, in ascending order according to the cumulative time taken to solve them.

Family	Solver	Total		True		False		Unique	
		#	Time	#	Time	#	Time	#	Time
bomb (132)	AIGSolve	83	1003.23	40	165.20	43	838.03	–	–
	rareqs	83	1420.61	34	165.41	49	1255.19	6	1094.71
	quantor	82	923.25	53	217.92	29	705.33	–	–
	aqme	80	674.38	53	345.02	27	329.36	–	–
	depqbf	67	2410.16	40	1693.36	27	716.79	–	–
	sKizzo	57	609.41	31	2.39	26	607.03	–	–
	ghostq	56	532.47	29	42.66	27	489.81	–	–
	QuBE	47	1168.86	23	470.47	24	698.39	–	–
	StruQS	36	1051.46	19	813.58	17	237.88	–	–
complexity (104)	rareqs	75	1559.65	29	466.77	46	1092.88	15	1148.51
	depqbf	49	1553.73	22	1086.35	27	467.38	–	–
	ghostq	42	1791.86	11	499.21	27	467.38	–	–
	QuBE	39	1273.95	19	277.87	20	996.09	–	–
	aqme	33	528.28	15	188.76	18	339.52	–	–
	quantor	26	170.44	11	11.29	15	159.14	–	–
	StruQS	21	1855.53	13	1677.81	8	177.72	–	–
	AIGSolve	15	70.26	7	12.24	8	58.02	–	–
	sKizzo	9	316.60	4	315.82	5	0.78	–	–
dungeon (107)	quantor	104	525.30	18	54.81	86	470.48	–	–
	aqme	104	1121.43	18	86.11	86	1035.32	–	–
	AIGSolve	87	1220.22	17	417.12	70	803.10	–	–
	rareqs	57	1870.73	18	54.89	39	1815.85	–	–
	depqbf	44	535.22	18	300.44	26	234.77	–	–
	QuBE	34	1429.60	7	212.89	27	1216.71	–	–
	ghostq	7	385.11	4	4.62	3	380.49	–	–
	sKizzo	2	0.99	–	–	2	0.99	–	–
	StruQS	1	21.96	1	21.96	–	–	–	–
hardness (114)	StruQS	88	7826.42	1	372.74	87	7453.68	12	3033.19
	QuBE	76	1346.11	–	–	76	1346.11	2	328.75
	ghostq	51	2649.30	2	239.56	49	2409.74	1	224.22
	aqme	50	265.14	–	–	50	265.14	–	–
	rareqs	14	1431.05	–	–	14	1431.05	–	–
	AIGSolve	12	2038.84	–	–	12	2038.84	–	–
	depqbf	8	617.99	–	–	8	617.99	–	–
	quantor	–	–	–	–	–	–	–	–
	sKizzo	–	–	–	–	–	–	–	–
planning (147)	AIGSolve	147	2371.36	38	114.02	109	2257.34	10	861.70
	rareqs	137	1093.01	38	125.66	99	967.35	–	–
	quantor	131	6750.13	37	122.68	94	6627.44	–	–
	aqme	123	9263.25	37	464.97	86	8798.28	–	–
	sKizzo	74	71.57	34	24.02	40	47.55	–	–
	depqbf	57	5134.24	29	1876.90	28	3257.34	–	–
	QuBE	14	1270.35	12	743.61	2	526.74	–	–
	ghostq	11	2155.26	8	1420.71	3	734.55	–	–
	StruQS	4	1229.67	4	1229.67	–	–	–	–
testing (131)	aqme	71	2675.64	64	2339.68	7	335.95	1	3.00
	StruQS	65	1770.09	63	1488.09	2	282.00	4	236.18
	depqbf	57	692.38	46	672.96	11	19.42	2	359.15
	AIGSolve	51	4194.44	46	4163.65	5	30.79	2	11.88
	QuBE	41	765.08	31	734.85	10	30.23	1	1.24
	rareqs	34	428.00	22	317.04	12	110.95	1	0.53
	ghostq	32	269.13	29	66.10	3	203.03	–	–
	quantor	26	121.15	25	110.52	1	10.63	–	–
	sKizzo	1	0.02	1	0.02	–	–	–	–

Table 2. Performances of QBF solvers on qbfg-t2: The table is split in six horizontal parts, one for each family. The first column contains families names, as well as its total amount of instances. The rest of the table is organized as Table 1.

Year	Solver	Total		True		False		Unique	
		#	Time	#	Time	#	Time	#	Time
2004 (167)	AIGSolve	62	2658.96	46	1781.89	16	877.07	15	1401.38
	ghostq	51	541.62	36	370.74	15	170.88	18	194.05
	QuBE	28	4783.92	23	3566.83	5	1217.09	6	626.75
	sKizzo	23	749.18	16	691.76	7	57.42	–	–
	StruQS	14	1081.04	9	885.70	5	195.34	–	–
	quantor	13	916.32	11	905.49	2	10.83	–	–
	depqbf	2	192.64	–	–	2	192.64	–	–
	rareqs	1	0.24	–	–	1	0.24	1	0.24
2005 (168)	AIGSolve	43	3386.36	32	2119.95	11	1266.41	18	1861.10
	ghostq	27	217.17	17	46.44	10	170.43	13	190.25
	QuBE	19	2653.56	16	2365.20	3	288.37	4	359.28
	rareqs	8	5.57	–	–	8	5.57	3	0.54
	StruQS	8	921.27	6	687.03	2	234.24	–	–
	sKizzo	7	595.39	7	595.39	–	–	–	–
	quantor	5	144.64	4	137.29	1	7.34	–	–
	depqbf	1	243.79	–	–	1	243.79	–	–
2006 (103)	ghostq	80	1577.10	80	1577.10	–	–	5	15.33
	AIGSolve	71	608.62	63	432.35	8	176.27	6	40.29
	QuBE	61	1108.04	57	506.54	4	601.49	1	190.78
	StruQS	37	373.31	36	298.82	1	74.48	–	–
	rareqs	4	277.89	1	102.27	3	175.62	–	–
	depqbf	1	22.85	–	–	1	22.85	–	–
	quantor	–	–	–	–	–	–	–	–
	sKizzo	–	–	–	–	–	–	–	–
2007 (281)	QuBE	88	7239.68	13	2729.53	75	4510.16	13	643.41
	rareqs	83	2706.28	10	1905.49	73	800.79	31	1741.36
	AIGSolve	61	765.34	36	547.74	25	217.61	34	445.93
	depqbf	50	2902.71	6	378.32	44	2524.39	5	264.06
	ghostq	49	1699.99	41	360.23	8	1339.76	19	275.34
	quantor	14	1302.44	11	1225.27	3	77.17	6	517.17
	StruQS	11	2051.88	11	2051.88	–	–	–	–
	sKizzo	5	953.55	1	50.44	4	903.11	–	–
2008 (961)	AIGSolve	335	13128.70	237	8439.00	98	4689.70	135	8250.56
	ghostq	304	8299.71	257	5325.55	47	2974.16	49	1493.92
	QuBE	198	19753.10	71	13501.39	127	6251.71	21	2320.72
	rareqs	126	4220.58	16	2744.06	110	1476.52	36	2255.13
	depqbf	96	4626.18	7	495.77	89	4130.41	4	261.53
	sKizzo	57	3599.21	26	1664.18	31	1935.02	1	275.69
	StruQS	50	5458.74	47	4844.87	3	613.87	–	–
	quantor	19	1646.95	18	1621.94	1	25.02	15	1166.44
2010 (96)	ghostq	29	591.60	25	281.30	4	309.76	10	342.94
	AIGSolve	22	282.18	20	262.94	2	19.24	5	208.02
	sKizzo	8	564.39	8	564.39	–	–	–	–
	QuBE	4	210.40	1	19.08	3	191.32	2	50.80
	rareqs	2	17.41	–	–	2	17.41	1	0.50
	depqbf	1	22.85	–	–	1	22.85	–	–
	quantor	–	–	–	–	–	–	–	–
	StruQS	–	–	–	–	–	–	–	–

Table 3. Performances of QBF solvers on challenging instances: The table is split in six horizontal parts, one for each family. The first column contains the QBFEVAL-related year families names, as well as its total amount of instances. The rest of the table is organized as Table 1.

	2004	**2005**	**2006**	**2007**	**2008**	**2010**
SOTA	88	65	85	207	601	37
SOTA-NEW	53	34	75	148	393	**30**
SOTA-LEGACY	**69**	**49**	**77**	**150**	**492**	26

Table 4. Performance of state-of-the-art solvers on CHALLENGING formulas. The table is organized as follows. The first column reports considered SOTA(s), while the remaining columns denote the pool of challenging formulas. In cells is shown the total amount of solved instances by the related SOTA. In bold we denote the best performance between SOTA-NEW and SOTA-LEGACY.