A QSAT Benchmark Based on Vertex-Folkman Problems: Implementation and Experiments

David E. Narváez

November 20, 2019

Solving times with 3 different QSAT solvers that support the QDIMACS format, namely `DepQBF` [7], `RAReQS` [6], and `caqe` [8] are summarized in Table 1. This table also includes information about some of the flags we considered for each solver that seemed promising in the sense that they improved the solving time for certain instances: for `DepQBF` we tried traditional QCDCL and for `caqe` we tried disabling expansion refinement. One can notice from Table 1 that `RAReQS` seems to outperform the other solvers at any parameter setting, though at the time of this writing we have not investigated if this is always the case and what is the reason.

Table 2 shows the running times of the `QFUN` [5] (with and without strategy learning), `CQESTO` [4] and `QuAbS`[1] [2] QSAT solvers which take circuits in the QCIR-14 format as input.

We looked at the task of listing all solution of graph searches up to order 8 without isolators. The comparison is between `sat-to-sat`'s model enumeration feature [3], our own implementation of a simple AllQSAT model enumerator for the QDIMACS format based on `DepQBF`'s C API using a blocking-clause approach, and `QFun`'s top-level winning move enumeration feature. The results are shown in Table 3.

References

[1] Armin Biere. PicoSAT essentials. *JSAT*, 4(2-4):75–97, 2008.

[2] Jesko Hecking-Harbusch and Leander Tentrup. Solving QBF by abstraction. In Andrea Orlandini and Martin Zimmermann, editors, *GandALF*, volume 277 of *EPTCS*, pages 88–102, 2018.

[3] Tomi Janhunen, Shahab Tasharrofi, and Eugenia Ternovska. SAT-to-SAT: Declarative extension of SAT solvers with new propagators. In Dale Schuurmans and Michael P. Wellman, editors, *AAAI*, pages 978–984. AAAI Press, 2016.

[4] Mikolás Janota. Circuit-based search space pruning in QBF. In Olaf Beyersdorff and Christoph M. Wintersteiger, editors, *SAT*, volume 10929 of *Lecture Notes in Computer Science*, pages 187–198. Springer, 2018.

[5] Mikolás Janota. Towards generalization in QBF solving via machine learning. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI*, pages 6607–6614. AAAI Press, 2018.

[6] Mikolás Janota, William Klieber, João Marques-Silva, and Edmund M. Clarke. Solving QBF with counterexample guided refinement. *Artificial Intelligence*, 234:1–25, 2016.

[7] Florian Lonsing and Uwe Egly. DepQBF 6.0: A search-based QBF solver beyond traditional QCDCL. In Leonardo de Moura, editor, *CADE*, volume 10395 of *Lecture Notes in Computer Science*, pages 371–384. Springer, 2017.

[1] We use a version of `QuAbS` which uses the `picosat` [1] SAT solver as the internal solver.

Table 1: Solving times in milliseconds for small QDIMACS instances (all possible sequences yielding graph searches of order at most 10) using partial symmetry breaking. For `DepQBF`, the default settings and traditional QCDCL were used. For `caqe`, the defaults and no expansion refinement settings were used.

Sequence	m	Order $(m+a)$	`DepQBF`		`RAReQS`	`caqe`	
			Defaults	Trad. QCDCL		Defaults	No Exp. Ref.
2,2	3	5	20	5	10	62	43
2,2,2	4	6	14	13	9	70	71
2,3	4	7	99	261	12	168	154
2,2,2,2	5	7	322	68	28	112	100
3,3	5	8	580	461	18	258	270
2,2,3	5	8	3398	882	30	275	273
2,2,2,2,2	6	8	912	401	53	177	150
2,4	5	9	660	6984	47	1001	951
2,3,3	6	9	24512	5539	52	464	458
2,2,2,3	6	9	2765	2966	81	442	422
2,2,2,2,2,2	7	9	2623	1585	123	364	283
3,4	6	10	26142	428	115	2130	1845
3,3,3	7	10	13979	304	105	743	629
2,2,4	6	10	16474	8429	327	1717	1750
2,2,3,3	7	10	9412	1443	138	901	708
2,2,2,2,3	7	10	34179	6781	164	699	582
2,2,2,2,2,2,2	8	10	6038	8858	303	2016	1667

Table 2: Solving times in milliseconds for small QCIR-14 instances (all possible sequences yielding graph searches of order at most 10) using partial symmetry breaking. For `QFUN`, the default settings as well as disabling learning were used.

Sequence	m	Order $(m+a)$	`QFUN`		`CQESTO`	`QuAbS`
			Defaults	No Learning		
2,2	3	5	30	8	27	14
2,2,2	4	6	68	63	52	9
2,3	4	7	42	43	95	121
2,2,2,2	5	7	333	400	376	133
3,3	5	8	2049	437	2253	368
2,2,3	5	8	1282	1869	891	1626
2,2,2,2,2	6	8	4282	1319	1443	975
2,4	5	9	-	-	-	9791
2,3,3	6	9	-	26793	12852	2612
2,2,2,3	6	9	-	29581	14375	3181
2,2,2,2,2,2	7	9	49546	7972	10397	1228
3,4	6	10	-	-	-	-
3,3,3	7	10	-	-	-	-
2,2,4	6	10	-	-	-	-
2,2,3,3	7	10	-	-	-	32050
2,2,2,2,3	7	10	-	-	-	-
2,2,2,2,2,2,2	8	10	-	30559	34846	12411

Table 3: Running times for the task of enumerating all solutions of instances of $F^{m+a}(a_1, \ldots, a_k; m)$ (without symmetry breaking), taking the average over 5 runs with a timeout of 10 minutes.

Sequence	m	Order $(m+a)$	`sat-to-sat`	`DepQBF`'s C API	`QFun`
`2,2`	3	5	80	36	41
`2,2,2`	4	6	13	200	941
`2,3`	4	7	382	3433	7553
`2,2,2,2`	5	7	180	3765	20087
`3,3`	5	8	12859	-	-
`2,2,3`	5	8	20580	-	-
`2,2,2,2,2`	6	8	1911	52100	-

[8] Markus N. Rabe and Leander Tentrup. CAQE: A certifying QBF solver. In Roope Kaivola and Thomas Wahl, editors, *FMCAD*, pages 136–143. IEEE, 2015.