Appears in A. Balint, A. Belov, D. Diepold, S. Gerber, M. Järvisalo, and C. Sinz (eds.), *Proceedings of SAT Challenge 2012: Solver and Benchmark Descriptions*, volume B-2012-2 of Department of Computer Science Series of Publications B, University of Helsinki 2012. ISBN 978-952-10-8106-4

pfolioUZK: Solver Description

Andreas Wotzlaw*, Alexander van der Grinten*, Ewald Speckenmeyer*, Stefan Porschen†
*Institut für Informatik
Universität zu Köln, Pohligstr. 1, D-50969 Köln, Germany
Email: {vandergrinten,wotzlaw,esp}informatik.uni-koeln.de
†Fachgruppe Mathematik, FB4
HTW-Berlin, Treskowallee 8, D-10318 Berlin, Germany
Email: porschen@htw-berlin.de

Solver Description

pfolioUZK is a portfolio SAT solver based on the portfolio SAT solver ppfolio developed by Olivier Roussel [1]. It can be used either as a parallel portfolio SAT solver on multicore systems, or as a sequential portfolio SAT solver. Here, the number of cores that may be used by pfolioUZK can be selected on the command line with the parameter `-c <number of cores>`. Like ppfolio, it is a simple computer program that starts SAT solvers from the available portfolio in parallel, among others an instance of our new complete SAT solver satUZK [2].

Currently, to the portfolio belong the following SAT solvers:

- satUZKs: a version of the complete SAT solver satUZK developed by Alexander van der Grinten and Andreas Wotzlaw, see [2] for a detailed description,
- glucose 2.0: a complete SAT solver by Gilles Audemard and Laurant Simon [3],
- lingeling 587 and plingeling 587: two complete SAT solvers by Armin Biere [4],
- contrasat: a complete SAT solver by Allen van Gelder,
- march_hi 2009: a complete SAT solver by Marijn Heule and Hans Van Maaren,
- TNM 2009: an incomplete SAT solver by Wanxia Wei and Chu Min Li [5],
- MPhaseSAT_M: a complete SAT solver by Jingchao Chen [6], and
- sparrow2011: an incomplete SAT solver developed by Dave Tompkins using the sparrow algorithm of Adrian Balint and Andreas Fröhlich [7].

The solvers have been chosen on the basis of their performance on the SAT Competition 2011. The type and the number of solvers that are started depend on the number of allocated cores and on the uniformity of the input instance. A CNF formula is *uniform* if all its clauses have exactly the same length. In case the input instance is uniform we start parallel only march_hi 2009, TNM 2009, MPhaseSAT_M, and sparrow2011, when possible each on a separate core. For all other instances, we use the following predefined configurations:

- 1 core or `-c 1`: satUZK, lingeling 587, TNM 2009, and MPhaseSAT_M are started on the same core (this configuration constitutes a sequential version of pfolioUZK),
- 2 cores or `-c 2`: satUZK and TNM 2009 on the first core, and glucose 2.0 and MPhaseSAT_M on the second core,
- 4 cores or `-c 4`: satUZK, glucose 2.0, contrasat, and lingeling 587, all on separate cores,
- 8 cores or `-c 8`: satUZK, glucose 2.0, contrasat, and four instances of plingeling 587 are started for CNF formulas with up to 12 millions clauses, all on their own cores. For larger formulas, satUZK is not used due to memory limitations.

For the SAT Challenge 2012 in tracks "Parallel Solvers - Application SAT+UNSAT" and "Sequential Portfolio Solvers" we have submitted both precompiled (with gcc 4.4.3 and -O3) and statically linked binaries (32- and 64-bit) as well as all sources (C/C++ programs and shell scripts). We consider to make the source code available online.

References

[1] O. Roussel, "Description of ppfolio," SAT Competition 2011, 2011.
[2] A. van der Grinten, A. Wotzlaw, E. Speckenmeyer, and S. Porschen, "satUZK: Solver description," SAT Challenge 2012, 2012.
[3] G. Audemard and L. Simon, "Predicting learnt clauses quality in modern SAT solver," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, 2009, pp. 399–404.
[4] A. Biere, "Lingeling, plingeling, picoSAT and precoSAT at SAT Race 2010," Institute for Formal Models and Verification, Johannes Kepler University, Linz, Austria, FMV Reports Series 10/1, August 2010.
[5] W. Wei and C. M. Li, "Switching between two adaptive noise mechanisms in local search for SAT," SAT Competition 2009, 2009.
[6] J. Chen, "Phase selection heuristics for satisfiability solvers," *CoRR*, 2011. [Online]. Available: http://arxiv.org/abs/1106.1372
[7] A. Balint and A. Fröhlich, "Improving stochastic local search for SAT with a new probability distribution," in *Proceedings of the 13th International Conference on Theory and Applications of Satisfiability Testing (SAT'10)*, ser. Lecture Notes in Computer Science, vol. 6175, 2010, pp. 10–15.