SPARROW+CP3 and SPARROWTORISS

Adrian Balint
Universität Ulm, Germany

Norbert Manthey
Knowledge Representation and Reasoning,
TU Dresden, Germany

***Abstract*—SPARROW+CP3 and SPARROWTORISS are using as a first step the preprocessor CP3 to simplify the formula in a way that is beneficial for SLS solvers. SPARROW+CP3 then uses the solver SPARROW to solve the simplified problem. SPARROWTORISS is first trying to solve the problem with Sparrow, limiting its execution to $5 \cdot 10^8$ flips and then passes the assignment found to the CDCL solver RISS3G, which uses this information for initialization and then tries to solve the problem. The solver RISS3G combines the improved Minisat-style solving engine of GLUCOSE 2.2 with a state-of-the-art preprocessor COPROCESSOR and adds further modifications to the search process. The SLS solver SPARROW is an improved version of SPARROW 2011**

I. INTRODUCTION

SLS solvers showed remarkable performance on the satisfiable crafted problems in the competitions from the last years. Motivated by this results we have analyzed in [1] the utility of different preprocessing techniques for the SLS solver SPARROW. The best found technique together with SPARROW represents the basis of our solver SPARROW+CP3.

As SPARROW is not able to prove the unsatisfiability of a problem we have decided to append a CDCL solver to SPARROW+CP3, namely RISS3G after limiting the execution of SPARROW to $5{\cdot}10^8$ flips. The CDCL solver RISS3G uses the MINISAT search engine [2], more specifically the extensions added in GLUCOSE 2.2 [3], [4]. Furthermore, RISS3G is equipped with the preprocessor COPROCESSOR(CP3) [5], that implements most of the recently published formula simplification techniques, ready to be used as inprocessing as well.

II. MAIN TECHNIQUES

SPARROW is a clause weighting SLS solvers that uses promising variables and probability distribution based selection heuristics. It is described in detail in [6]. Compared to the original version, the one submitted here is updating weights of unsatisfied clauses in every step where no promising variable can be found.

The built-in preprocessor CP3 has been ported from COPROCESSOR 2 and supports the following simplification techniques: Unit Propagation, Subsumption, Strengthening (also called self-subsuming resolution) – where for small clauses all subsuming resolvents can be produced, (Bounded) Variable Elimination (BVE) [7] combined with Blocked Clause Elimination (BCE) [8], (Bounded) Variable Addition (BVA) [9], Probing [10], Covered Clause Elimination [11], Hidden Tautology Elimination [12], Equivalent Literal Elimination [13], Unhiding (Unhide) [14], Add Binary Resolvents [15], at-most-one rewriting [16], [17], a 2SAT algorithm [18], and a walksat implementation [19]. The preprocessor furthermore supports parallel subsumption, strengthening and variable elimination, which is described in [20].

RISS3G uses GLUCOSE 2.2 as main search engine – the version used in SPARROWTORISS just replaces the internal preprocessor with CP3.

The combination of the SPARROW and RISS3G, called SPARROWTORISS, does not simply execute the two solvers after each other, but also forwards information from the SLS solver to the CDCL solver: when SPARROW terminates, it outputs its last full assignment in chronological order (i.e. the oldest variable first), which is used to initialize the phase saving of RISS3G, such that the first decisions of RISS3G follow this assignment. In a brief empirical evaluation this communication turned out to be useful. The solvers are also able to forward the information about the age of the variables in the SLS search. This data could be used to initialize the activities of the variables inside RISS3G. However, this feature is not enabled in the used configuration.

III. MAIN PARAMETERS

SPARROW is using the same parameters as SPARROW 2011.

The configuration of CP3 has been tuned for SPARROW in [1] on the SAT Challenge 2012 satisfiable hard combinatorial benchmarks.

The main parameters of RISS3G control how the formula simplification of CP3 is executed. The configuration of CP3 has been tuned for GLUCOSE 2.2 in [1] on the SAT Challenge 2012 application benchmark. The final setup of the preprocessor inside RISS3G uses the following techniques: UP, SUB+STR (producing all resolvents for ternary clauses), Unhide without *hidden literal elimination* [14] and 5 iterations, BVE without on the fly BCE and BVA with a small number of 120000 steps.

For SPARROWTORISS it can be chosen whether to forward the last assignment, or the activity information.

IV. IMPLEMENTATION DETAILS

SPARROW is implemented in C. The solver RISS3G is build on top of MINISAT 2.2 and GLUCOSE 2.2. Furthermore, we integrated COPROCESSOR into the system, allowing inprocessing techniques to be executed during search – however, this feature is not used in the competition. All solvers have been compiled with the GCC C++compiler as 64 bit binaries.

V. AVAILABILITY

The source code of RISS3G (including CP3) is available at tools.computational-logic.org for research purposes.

Acknowledgment

The authors would like to thank Armin Biere for many helpful discussions on formula simplification and the BWGrid [21] project for providing computational resources to tune CP3. This project was partially funded by the Deutsche Forschungsgemeinschaft (DFG) under the number SCHO 302/9-1. Finally, the authors would like to thank TU Dresden for providing the computational resources to develop, test and evaluate Riss3g.

References

[1] A. Balint and N. Manthey, "Boosting the Performance of SLS and CDCL Solvers by Preprocessor Tuning," in *Pragmatics of SAT*, 2013.

[2] N. Eén and N. Sörensson, "An extensible sat-solver," in *SAT*, ser. Lecture Notes in Computer Science, E. Giunchiglia and A. Tacchella, Eds., vol. 2919. Springer, 2003, pp. 502–518.

[3] G. Audemard and L. Simon, "Predicting learnt clauses quality in modern SAT solvers," in *Proc. 21st Int. Joint Conf. on Artifical Intelligence (IJCAI '09)*. Morgan Kaufmann, 2009, pp. 399–404.

[4] ——, "Refining restarts strategies for sat and unsat," in *Proceedings of the 18th international conference on Principles and Practice of Constraint Programming*, ser. CP'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 118–126. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33558-7_11

[5] N. Manthey, "Coprocessor 2.0: a flexible cnf simplifier," in *Proceedings of the 15th international conference on Theory and Applications of Satisfiability Testing*, ser. SAT'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 436–441. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-31612-8_34

[6] 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*, ser. SAT'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 10–15. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-14186-7_3

[7] N. Eén and A. Biere, "Effective preprocessing in sat through variable and clause elimination," in *Proceedings of the 8th international conference on Theory and Applications of Satisfiability Testing*, ser. SAT'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 61–75. [Online]. Available: http://dx.doi.org/10.1007/11499107_5

[8] M. Järvisalo, A. Biere, and M. Heule, "Blocked clause elimination," in *Proceedings of the 16th international conference on Tools and Algorithms for the Construction and Analysis of Systems*, ser. TACAS'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 129–144. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-12002-2_10

[9] N. Manthey, M. J. H. Heule, and A. Biere, "Automated reencoding of boolean formulas," in *Proceedings of Haifa Verification Conference 2012*, 2012.

[10] I. Lynce and J. Marques-Silva, "Probing-Based Preprocessing Techniques for Propositional Satisfiability," in *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, ser. ICTAI '03. IEEE Computer Society, 2003, pp. 105–110. [Online]. Available: http://portal.acm.org/citation.cfm?id=951951.952290

[11] M. Heule, M. Järvisalo, and A. Biere, "Covered clause elimination," *CoRR*, vol. abs/1011.5202, 2010.

[12] M. Heule, M. Järvisalo, and A. Biere, "Clause elimination procedures for cnf formulas," in *Proceedings of the 17th international conference on Logic for programming, artificial intelligence, and reasoning*, ser. LPAR'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 357–371. [Online]. Available: http://dl.acm.org/citation.cfm?id=1928380.1928406

[13] A. V. Gelder, "Toward leaner binary-clause reasoning in a satisfiability solver," *Ann. Math. Artif. Intell.*, vol. 43, no. 1, pp. 239–253, 2005.

[14] M. J. H. Heule, M. Järvisalo, and A. Biere, "Efficient cnf simplification based on binary implication graphs," in *Proceedings of the 14th international conference on Theory and application of satisfiability testing*, ser. SAT'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 201–215. [Online]. Available: http://dl.acm.org/citation.cfm?id=2023474.2023497

[15] W. Wei and B. Selman, "Accelerating random walks," in *Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming*, ser. CP '02. London, UK, UK: Springer-Verlag, 2002, pp. 216–232. [Online]. Available: http://dl.acm.org/citation.cfm?id=647489.727142

[16] N. Manthey and P. Steinke, "Quadratic Direct Encoding vs. Linear Order Encoding," in *First International Workshop on the Cross-Fertilization Between CSP and SAT(CSPSAT'11)*, 2011.

[17] M. N. V. Van Hau Nguyen and S. Hölldobler, "Application of hierarchical hybrid encoding to efficient translation of a csp to sat," Knowledge Representation and Reasoning Group, Technische Universität Dresden, 01062 Dresden, Germany, Tech. Rep., 2013.

[18] A. del Val, "On 2-sat and renamable horn," in *AAAI/IAAI*, H. A. Kautz and B. W. Porter, Eds. AAAI Press / The MIT Press, 2000, pp. 279–284.

[19] B. Selman, H. A. Kautz, and B. Cohen, "Noise strategies for improving local search," in *AAAI*, B. Hayes-Roth and R. E. Korf, Eds. AAAI Press / The MIT Press, 1994, pp. 337–343.

[20] K. Gebhardt and N. Manthey, "Parallel Variable Elimination on CNF Formulas," in *Pragmatics of SAT*, 2013.

[21] bwGRiD (http://www.bw grid.de/), "Member of the german d-grid initiative, funded by the ministry of education and research (bundesministerium für bildung und forschung) and the ministry for science, research and arts baden-wuerttemberg (ministerium für wissenschaft, forschung und kunst baden-württemberg)," Universities of Baden-Württemberg, Tech. Rep., 2007-2010.