Künstl Intell (2016) 30:339–342
DOI 10.1007/s13218-015-0406-8

Towards Next Generation Sequential and Parallel SAT Solvers

Norbert Manthey[1]

Published online: 14 October 2015

Keywords SAT solving · Parallel search · Formula simplification · Formula rewriting

1 Introduction

Satisfiability testing (SAT) is used to solve many academic and industrial problems from the complexity class $\mathcal{NP}$, for example hardware verification or scheduling [1]. The described dissertation [4] focuses on improving the SAT solving technology, such that tools that build on SAT solvers are improved automatically as well. The improvements focus on two major subjects: sequential SAT solving and parallel SAT solving. However, to allow also formal reasoning on the soundness of the presented SAT techniques, a theoretical foundation has been built as well. Hence, the thesis covers a broad range from theory and soundness proofs over abstract reduction systems and algorithm to parallel algorithms. New approaches and approaches from the literature have been implemented and have been evaluated. The probably most useful presented technique is bounded variable addition, which allows to automatically rewrite an existing CNF formula into a smaller formula by introducing auxiliary variables. The implemented solvers participated in international SAT competitions, and the first versions have been implemented from scratch. From the year 2012 on, the search engine was replaced by the MiniSAT solver. The parallel solver Pcasso showed a good performance in 2013 and 2014. The sequential SAT solver Riss in combination with the formula simplification tool Coprocessor won several first, second and third prices, including two Kurt-Gödel-Medals. These results show, among other contributions of the thesis, that the research summarized in the thesis improved the state of the art in modern SAT solving.

2 Sequential SAT Solving

This part of the thesis analyses sequential SAT solving techniques ranging from reencoding the formula over simplification methods to search algorithms and search extensions. Solving the SAT problem is to answer whether there exists a satisfying interpretation for a given propositional formula, where implemented systems accept only formulas in conjunctive normal form (CNF). The considered algorithms, which nowadays are based on the conflict driven clause learning (CDCL) algorithm, also show if a given formula is unsatisfiable.

As SAT solvers construct partial models, an extension to classical logic is proposed, that allows to work with partial interpretations. Next, the abstract reduction system Generic CDCL has been introduced [2] for a formal analysis of new solving methods. The system is based on a solver state, which is formed by a tuple of the current formula and the interpretation. Then, eight rules are specified that allow to modify the state. These rules include termination rules like SAT and UNSAT, as well as rules for decisions and inference for controlling search, unit propagation and other inferences, as well as formula transformations for formula simplification techniques. With this system a modern SAT solver can be modeled, and properties like soundness can be shown in the abstract reduction system. Based on this reduction system many proposed solving techniques from

✉ Norbert Manthey
norbert.manthey@tu-dresden.de

1 Knowledge Representation and Reasoning Group, Technische Universität Dresden, 01062 Dresden, Germany

the literature presented and a way how to model these techniques with Generic CDCL was shown. All techniques that can be modeled automatically inherit the properties of Generic CDCL, most importantly being sound. The techniques have been implemented into the SAT solver Riss and are evaluated.

Figure 1 presents the work flow for solving a problem with a SAT solver. Besides the steps preprocessing and search, modern systems furthermore interleave formula simplification with search, as during search additional knowledge is acquired that allows further simplifications. With the work of this thesis, another step is added as well: the formula can be rewritten by the formula simplification tool. The runtime distribution of the three parts in the SAT solving workflow shows that improvements in the search will lead to the most significant effect. Still, formula simplification techniques are crucial to the overall system, as they for example allow to reduce the formula size significantly.

Therefore, the formulas simplifier Coprocessor has been implemented [3]. Furthermore, the implemented formula simplification techniques from the literature are presented, and a way how to model them with Generic CDCL has been demonstrated. The implementation of some of these techniques has been adapted to work only on a subset of the variables of the formula, so that the simplifier can also be used for formula transformations [7], or for model counting or model enumeration of models for a subset of the variables of the formula. During the work of this thesis two formula simplification techniques have been proposed, namely bounded variable addition (BVA) [5] and covered literal elimination.[1] With BVA, many formulas that resulted from naive CNF encodings can be automatically reencoded into formulas that lead to an improved performance of the SAT solver, with an size reduction of up to factor 10 and a runtime improvement of up to factor 36 [5]. This especially includes the widely used *at-most-one* constraints over a set of variables, that is used to encode high level constraints into CNF, for example the domain of a constraint variable based on a set of Boolean variables M. The naive CNF formula has a quadratic size in M, more precisely $\frac{|M|(|M|-1)}{2}$ clauses are produced. After applying BVA to this encoding, the reduced formula contains only $3|M|-6$ clauses. As for application formulas the number of clauses usually has a higher influence on the performance of modern SAT solvers, BVA turned out to be very powerful.

Furthermore, the cardinality constraint recognition for the Fourier–Motzkin simplification technique has been improved. Without the detection, the Fourier–Motzkin method is ineffective on CNF formulas. With these additional techniques, Riss can solve more unsatisfiable formulas from the used benchmark than the state-of-the-art SAT solvers Glucose or Lingeling.

During introducing the search algorithms three extensions have been proposed: Local Look-Ahead, Local Probing and All Unique Implication Point Learning. All three techniques aim at finding implied unit clauses during search, either by using look-ahead with up to five decision literals, by performing probing with each learned clause, or by trying to learn multiple unit clauses from a single conflict. With all extensions enabled, Riss is improved over its default configuration—instead of 2510 the new configuration can solve 2521 instances of the benchmark with 3886 problems. The resulting configuration of Riss can solve as many formulas as Glucose, but more unsatisfiable formulas.

3 Parallel SAT Solving

Due to the increasing number of cores in CPUs, parallel solving approaches for the SAT problem have been investigated in this thesis. After giving a detailed overview on related work, the thesis discusses both low-level parallelization approaches and high-level approaches, and evaluated the proposed improvements.

3.1 Parallel Approaches: Ideas and Weaknesses

A contribution is that for each proposed techniques in the literature weaknesses have been pointed out and that ideas that led to improvements have been emphasized. Where for sequential SAT solvers clause learning and formula simplification is very important, parallel SAT solvers benefit especially from sharing the learned clauses. Hence, the compatibility of clause sharing and formula simplification in parallel SAT solvers also has been analyzed [6]. Sound combinations of simplification and sharing have been explained and a novel combination was proposed. This work received the best paper award of the SAT conference 2013.

3.2 Low-Level Parallel Search

Next, the thesis presents low-level parallelizations for SAT solving techniques. Parallel solving approaches can be divided into high-level parallelization and low-level parallelization. Although not presented in much detail, the parallelization of unit propagation has been investigated with the result that this low-level parallelization does not scale beyond two workers.

[1] Covered literal elimination has been invented independently also by Marijn Heule.

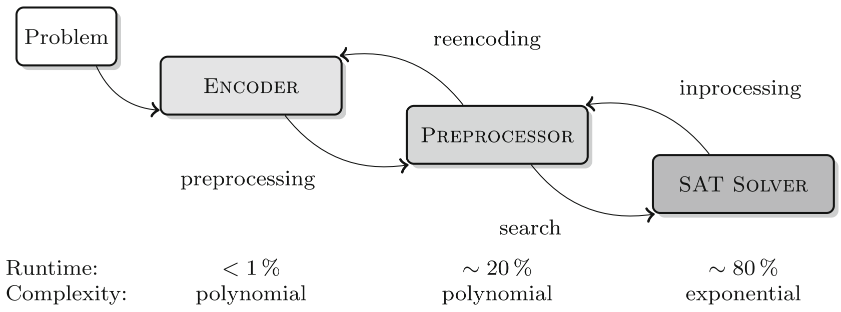

Fig. 1 Tool chain of modern SAT solver with a run time distribution

As a low level parallelization of the CDCL algorithm is not promising, we focussed on formula simplification techniques. A parallel algorithm for the most powerful formula simplification techniques subsumption, strengthening and variable elimination have been presented. The evaluation showed that especially on large formulas the simplification time improves for the parallel variants. The implementation of these algorithms is scalable on the hardware that has been used for the benchmarks and achieved almost linear speedups.

3.3 Scalable Parallel Search

High-level parallelizations can be divided into search space partitioning solvers and parallel portfolio solvers that run multiple configurations in parallel. Since portfolio solvers are known to scale with the number of configurations, the thesis presents the parallel and more scalable SAT solving approach iterative partitioning for the multi-core architecture. The multi-core implementation has been adapted from a grid implementation. Along the thesis, modifications and extensions are proposed, which increase the performance of the resulting parallel solver—from 771 instances in the benchmark, the basic version solves 636 instances and the final configuration solves 656 instances. The most important extensions are as follows. The first extension is clause sharing, more precisely clause sharing based on *partition tree level based clause tagging*. Here each learnt clause memorizes its dependencies, by storing the level of the partition tree it depends on, such that sound clause sharing is possible. Clauses are then only shared in the subtree they depend on. Furthermore, the search space partitioning was first based on a short sequential search of a SAT solver, and a procedure called *scattering*. The best combination is to partition the search space with the look-ahead procedure and the scattering approach. Finally, the partition tree level of the clauses can also be used to abort search in redundant search space partitions once some solver found an unsatisfiable search space partition. If a subtree is found to be unsatisfiable, all solvers in this subtree can be aborted.

The resulting parallel SAT solver Pcasso has been compared to state-of-the-art SAT solvers, and their scalability has been evaluated when moving from 8 cores to 16 cores. The number of solvable formulas from Pcasso is very competitive to other parallel SAT solvers like Plingeling or PeneLoPe. However, the iterative partitioning solving approach of Pcasso turns out to be the most scalable parallel SAT solving routine.

4 Conclusion

From the findings in the thesis a few conclusions can be drawn: since there is currently no known approach to parallelize the sequential SAT algorithm, high-level parallelizations are the only possible way to exploit modern parallel hardware. While the portfolio approach is a simple way to obtain a robust parallel solving procedure, search space partitioning results in a better scalability. An important fact is that improvements to the sequential algorithm can be easily added to the parallel procedure. Since parallelization is not believed to result in superlinear speedups in average, the sequential solving algorithm should remain in the research scope so that parallel SAT solvers are improved by developments on both the parallel and the sequential part. The thesis contributed to both fields, sequential and parallel SAT solving and presented algorithms that when being added to modern systems result in improved state-of-the-art SAT solvers.

Due to the missing understanding of the reason why modern CDCL SAT solvers are so powerful, this thesis contributes more to the engineering side of SAT solving by introducing an appropriate extension to classical propositional logic and an abstract reduction system. Then, extensions of sequential search are introduced and additional prototypical formula simplification techniques are presented, which already boost the reasoning power of current CDCL solvers. Finally, a scalable parallel solving approach is ported to the multi-core architecture and has been extended to increase its performance. All these novelties do not consider studying the reasoning power of the CDCL solvers, and neither try to improve the underlying reasoning systems. The presented simplification techniques bounded variable addition and the cardinality extraction for the Fourier–Motzkin method are only two steps into this direction.

From an applied point of view, the most interesting contribution of the thesis might be bounded variable addition, which rewrites inefficient CNF encodings with respect to formula size into more sophisticated encodings. Hence, naively encoded CNF formulas can be rewritten with the help of the simplifier Coprocessor.

To obtain a next generation SAT solver, the underlying reasoning system should become stronger than the currently used resolution system, which is only loosely linked with more powerful reasoning systems in a few systems. However, such a next generation SAT solver cannot compete with current state-of-the-art solvers if the engineering that leads to the implemented system is not performed well. Hence, both theory and engineering have to work closely together in future research to enter the next generation of SAT solving.

References

1. Biere A, Heule MJH, van Maaren H, Walsh T (eds) Handbook of satisfiability, Frontiers in artificial intelligence and applications, vol 185. IOS Press (2009)
2. Hölldobler S, Manthey N, Philipp T, Steinke P (2014) Generic CDCL—A formalization of modern propositional satisfiability solvers. In: Berre DL (ed) POS-14, EPiC Series, vol 27, pp 89–102. EasyChair
3. Manthey N (2012) Coprocessor 2.0—A flexible CNF simplifier. In: Cimatti A, Sebastiani R (eds) Theory and Applications of Satisfiability Testing—SAT 2012, Lecture Notes in Computer Science, vol 7317, pp 436–441. Springer Berlin Heidelberg (2012). doi:10.1007/978-3-642-31612-8_34
4. Manthey N (2014) Towards next generation sequential and parallel SAT solvers. Ph.D. thesis, Technische Universität Dresden. http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-158672
5. Manthey N, Heule MJH, Biere A (2013) Automated reencoding of Boolean formulas. In: Biere A, Nahir A, Vos T (eds) Hardware and software: verification and testing, Lecture Notes in Computer Science, vol 7857, pp 102–117. Springer Berlin Heidelberg (2013). doi:10.1007/978-3-642-39611-3_14
6. Manthey N, Philipp T, Wernhard C (2013) Soundness of inprocessing in clause sharing SAT solvers. In: Järvisalo M, Van Gelder A (eds) Theory and applications of satisfiability testing—SAT 2013, Lecture Notes in Computer Science, vol 7962, pp 22–39. Springer Berlin Heidelberg. doi:10.1007/978-3-642-39071-5_4
7. Wernhard C (2013) Computing with logic as operator elimination: The ToyElim system. In: Tompits H, Abreu S, Oetsch J, Pührer J, Seipel D, Umeda M, Wolf A (eds) Applications of declarative programming and knowledge management, Lecture Notes in Computer Science, vol 7773, pp 289–296. Springer Berlin Heidelberg (2013). doi:10.1007/978-3-642-41524-1_17

Norbert Manthey was born 4th December in 1986 in Räckelwitz. He conducted his studies of computer science at the Technische Universität Dresden from 2006 to 2010. In this period he also visited the NICTA in Sydney, Australia, for a project work. After his studies, he stayed at TU Dresden and did his doctorate in 2014 under the supervision of Prof. Steffen Hölldobler, who is the head of the Knowledge Representation and Reasoning group. In 2011 he visited Helsinki and Linz for research stays. The implemented software systems won several first prices as well as many other Top 3 prices, and hence these systems have also been used for collaboration with leading research experts of his field.