Computable Short Proofs*

Marijn J.H. Heule

Department of Computer Science, The University of Texas at Austin

The success of satisfiability solving presents us with an interesting peculiarity: modern solvers can frequently handle gigantic formulas while failing miserably on supposedly easy problems. Their poor performance is typically caused by the weakness of their underlying proof system—resolution. To overcome this obstacle, we need solvers that are based on stronger proof systems. Unfortunately, existing strong proof systems—such as extended resolution [1] or Frege systems [2]—do not seem to lend themselves to mechanization.

We present a new proof system that not only generalizes strong existing proof systems but that is also well-suited for mechanization. The proof system is surprisingly strong, even without the introduction of new variables — a key feature of short proofs presented in the proof-complexity literature. Moreover, we introduce a new decision procedure that exploits the strengths of our new proof system and can therefore yield exponential speed-ups compared to state-of-the-art solvers based on resolution.

Our new proof system, called PR (short for Propagation Redundancy), is a *clausal proof system* and closely related to state-of-the-art SAT solving. Informally, a clausal proof system allows the addition of redundant clauses to a formula in conjunctive normal form. Here, a clause is considered redundant if its addition preserves satisfiability. If the repeated addition of clauses allows us finally to add the empty clause—which is, by definition, unsatisfiable—the unsatisfiability of the original formula has been established.

Since the redundancy of clauses is not efficiently decidable in general, clausal proof systems only allow the addition of a clause if it fulfills some efficiently decidable criterion that ensures redundancy. For instance, the popular DRAT proof system [3], which is the de-facto standard in practical SAT solving, only allows the addition of so-called *resolution asymmetric tautologies* [4]. Given a formula and a clause, it can be decided in polynomial time whether the clause is a resolution asymmetric tautology with respect to the formula and therefore the soundness of DRAT proofs can be checked efficiently. Several formally-verified checkers for DRAT proofs are available [5, 6].

We present a new notion of redundancy by introducing a characterization of clause redundancy based on a semantic implication relationship between formulas. By replacing the implication relation in this characterization with a restricted notion of implication that is computable in polynomial time, we then obtain powerful notion of redundancy that is still efficiently decidable. The PR proof system, which based on this notion of redundancy, turns out to be highly expressive, even without allowing the introduction of new variables. This is in

* Based on joint work with Benjamin Kiesl, Armin Biere, and Martina Seidl

contrast to resolution, which is considered relatively weak as long as the introduction of new variables via definitions—as in the stronger proof system of extended resolution—is not allowed. The introduction of new variables, however, has a major drawback—the search space of variables and clauses we could possibly add to a proof is clearly exponential. Finding useful clauses with new variables is therefore hard in practice and resulted only in limited success in the past [7, 8].

In order to capitalize on the strengths of the PR proof system in practice, we enhance conflict-driven clause learning (CDCL) [9]. To do so, we introduce *satisfaction-driven clause learning* (SDCL) [10], a SAT solving paradigm that extends CDCL as follows: If the usual unit propagation does not lead to a conflict, we do not immediately decide for a new variable assignment (as would be the case in CDCL). Instead, we first try to prune the search space of possible truth assignments by learning a so-called PR clause. We demonstrate the strength of SDCL by computing short PR proofs for the famous pigeon hole formulas without new variables.

At this point there exists only an unverified checker to validate PR proofs, written in C. In order to increase the trust in the correctness of PR proofs, we implemented a tool to convert PR proofs into DRAT proofs [11], which in turn can be validated using verified proof checkers. Thanks to various optimizations, the size increase during conversion is rather modest on available PR proofs, thereby making this a useful certification approach in practice.

References

1. Tseitin, G.S.: On the complexity of derivation in propositional calculus. In: Automation of Reasoning 2. Springer (1983) 466–483
2. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. The Journal of Symbolic Logic **44**(1) (1979) pp. 36–50
3. Wetzler, N.D., Heule, M.J.H., Hunt Jr., W.A.: DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In: Theory and Applications of Satisfiability Testing – SAT 2014, Cham, Springer International Publishing (2014) 422–429
4. Järvisalo, M., Heule, M.J.H., Biere, A.: Inprocessing rules. In Gramlich, B., Miller, D., Sattler, U., eds.: IJCAR. Volume 7364 of LNCS., Springer (2012) 355–370
5. Cruz-Filipe, L., Heule, M.J.H., Hunt Jr., W.A., Kaufmann, M., Schneider-Kamp, P.: Efficient certified rat verification. In: Automated Deduction – CADE 26, Cham, Springer International Publishing (2017) 220–236
6. Lammich, P.: Efficient verified (un)sat certificate checking. In: Automated Deduction – CADE 26, Cham, Springer International Publishing (2017) 237–254
7. Audemard, G., Katsirelos, G., Simon, L.: A restriction of extended resolution for clause learning sat solvers. In: Proc. of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010), AAAI Press (2010)
8. Manthey, N., Heule, M.J.H., Biere, A.: Automated reencoding of boolean formulas. In: Proceedings of Haifa Verification Conference 2012. (2012)
9. Marques-Silva, J.P., Sakallah, K.A.: GRASP – a new search algorithm for satisfiability. In: ICCAD '96: Proceedings of the 1996 IEEE/ACM international conference

on Computer-aided design, Washington, DC, USA, IEEE Computer Society (1996) 220–227
10. Heule, M.J.H., Kiesl, B., Seidl, M., Biere, A.: Pruning through satisfaction. In Strichman, O., Tzoref-Brill, R., eds.: Hardware and Software: Verification and Testing, Cham, Springer International Publishing (2017) 179–194
11. Heule, M.J.H., Biere, A.: What a difference a variable makes. In Beyer, D., Huisman, M., eds.: Tools and Algorithms for the Construction and Analysis of Systems, Cham, Springer International Publishing (2018) 75–92