Towards Certified MaxSAT Solving

Dieter Vandesande[0000−0002−8150−3202] and Bart Bogaerts[0000−0003−3460−4251]

Vrije Universiteit Brussel, Belgium, Artificial Intelligence Lab
{dieter.vandesande,bart.bogaerts}@vub.be

Maximum Satisfiability (MaxSAT) [31, 4] is a combinatorial optimization paradigm where a so-called *solver* searches for a solution to a propositional formula while simultaneously optimizing a linear objective function. Extensive research in the field has led to highly efficient solving algorithms. This has been witnessed year after year by an increase in performance of the participating solvers in the annual MaxSAT evaluation [19]. Unfortunately, the constant race for more efficient combinatorial solving algorithms comes at a price: the development of complex software often leads to the introduction of bugs and erroneous reasoning, which may even result in incorrect outcomes [14, 13]. One way to mitigate this incorrectness is by designing *certifying algorithms* [1, 35]. A certifying algorithm does not only produce an answer to its input, but also produces an easily verifiable *proof*, called a *certificate*, of the correctness of its answer. An independent and simpler *proof checker* then verifies if the provided certificate indeed proves the correctness of the solver's answer for the given input.

In *Satisfiability Checking* (SAT) [9], proof logging is now common practice. For over a decade, producing certificates has been a mandatory requirement for solvers in the main-track of the yearly SAT competition. For SAT, several practical proof systems have been developed [24, 44, 15, 16, 8]. Unfortunately, for other formalisms, such as MaxSAT, proof logging is often still out of reach. Even for MaxSAT solvers that make use of repeated calls to SAT solvers, proof systems for SAT are not usable since they have no mechanism to reason about the optimality of a solution, which is crucial for MaxSAT. Additionally, in contrast to SAT (where all state-of-the-art solvers implement variations of the CDCL algorithm), MaxSAT algorithms exhibit much more diversity [31, 4]. While proof systems tailored towards MaxSAT have been proposed [29, 25, 11, 12, 30, 36, 39, 33, 34, 32], none of them can capture this wide variety of MaxSAT solvers.

Recently, VeriPB [18, 23, 21, 20, 10] was introduced as a proof system that reasons over 0–1 integer linear inequalities, also known as *pseudo-Boolean constraints*, and has been utilized for certifying a broad spectrum of solving techniques, including graph solving [21, 20], advanced SAT solving [23, 10] and constraint programming [18, 22]. In this thesis [40], we propose to use the VeriPB proof system as a general-purpose proof system for certifying MaxSAT solvers.

We demonstrate that VeriPB can be be used to certify two main classes of MaxSAT algorithms. More specifically, we add proof logging to the *model-improving* solver QMaxSAT [28] as well as the *core-guided* solver CGSS [27].

CGSS is a state-of-the-art MaxSAT solver implementing a so-called core-guided search algorithm. This makes that, to the best of our knowledge, our certifying version of CGSS is the first state-of-the-art MaxSAT solver with proof logging capabilities. Moreover, CGSS is an extension to RC2 [26], which is arguably the most-commonly used MaxSAT solver in practice. Our implementation of the certified version of CGSS also demonstrates how to certify modern heuristics, such as hardening [2], the intrinsic-at-most-ones technique [26], weight-aware core extraction [7], and structure sharing [27].

The other solver we equip with proof logging capabilities, QMaxSAT, implements the (simpler) model-improving search paradigm. The difficulty in proof logging QMaxSAT lies in certifying different ways of encoding pseudo-Boolean constraints as propositional formulas. This step is required since QMaxSAT works by repeatedly calling a SAT oracle, asking for a solution that is better than the previously found one. The constraint "the next solution should be better" is naturally expressed as a pseudo-Boolean formula; however, SAT solvers do not understand this. We show how to certify the totalizer encoding [5], binary adder [43], modulo-based totalizer [38] and cardinality networks [17, 3].

Our experimental evaluation suggests that the overhead induced by proof logging is limited in most cases. Where proof logging induces overhead, preliminary tests suggest that there is still room for improvement in the technical implementation. The same holds for the time necessary to validate the proofs: while in many cases the proof is validated within a reasonable time period, VeriPB is currently under active development to increase its performance, especially on the reverse unit propagation rule, which seemed to be one of the biggest sources of performance issues in the proof checker [42, 37].

Interestingly, our experiments also exemplified the importance of proof logging by revealing two bugs in CGSS and RC2. These bugs never resulted in any wrong results on any of the benchmarks used. However, thanks to proof logging, we were able to detect that the reasoning happening was actually faulty. After a careful analysis, we created an example where this incorrect reasoning leads to an incorrect answer. This showcases the value of proof logging as a testing and debugging methodology. Of course, proof logging can never be used to get guarantees of correctness of the solver; when a proof is accepted by the checker, the only guarantee we get is that the answer is correct and is obtained using correct reasoning. This thesis contributes to obtain such guarantees for answers obtained by using MaxSAT solvers. Specifically, by combining the work on core-guided and model-improving search, this thesis shows that out of the four major families of MaxSAT algorithms, two (namely core-guided and model-improving search) are certifiable using the VeriPB proof system. The other two major families (implicit hitting-set search and branch-and-bound) remain for future work.

Acknowledgements Computational resources and services were provided by the VSC (Flemish Supercomputer Center), funded by FWO and the Flemish Government. This thesis was developed in collaboration with Wolf De Wulf, Jeremias Berg, Jakob Nordström and Andy Oertel; shorter versions of two chapters have been published [41, 6]. This work has benefited from Dagstuhl Seminar 22411 "Theory and Practice of SAT and Combinatorial Solving."

References

1. Alkassar, E., Böhme, S., Mehlhorn, K., Rizkallah, C., Schweitzer, P.: An introduction to certifying algorithms. it Inf. Technol. **53**(6), 287–293 (2011)
2. Andres, B., Kaufmann, B., Matheis, O., Schaub, T.: Unsatisfiability-based optimization in clasp. In: ICLP (Technical Communications). pp. 211–221 (2012)
3. Asín, R., Nieuwenhuis, R., Oliveras, A., Rodríguez-Carbonell, E.: Cardinality networks and their applications. In: SAT. LNCS, vol. 5584, pp. 167–180. Springer (2009)
4. Bacchus, F., Järvisalo, M., Martins, R.: Maximum satisfiabiliy. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability - Second Edition, Frontiers in Artificial Intelligence and Applications, vol. 336, pp. 929–991. IOS Press (2021). https://doi.org/10.3233/FAIA201008
5. Bailleux, O., Boufkhad, Y.: Efficient CNF encoding of boolean cardinality constraints. In: CP. LNCS, vol. 2833, pp. 108–122. Springer (2003)
6. Berg, J., Bogaerts, B., Nordström, J., Oertel, A., Vandesande, D.: Certified core-guided maxsat solving. In: CADE. LNCS, vol. 14132, pp. 1–22. Springer (2023)
7. Berg, J., Järvisalo, M.: Weight-aware core extraction in sat-based maxsat solving. In: CP. LNCS, vol. 10416, pp. 652–670. Springer (2017)
8. Biere, A.: Tracecheck. `http://fmv.jku.at/tracecheck/` (2006)
9. Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.): Handbook of Satisfiability - Second Edition, Frontiers in Artificial Intelligence and Applications, vol. 336. IOS Press (2021)
10. Bogaerts, B., Gocht, S., McCreesh, C., Nordström, J.: Certified symmetry and dominance breaking for combinatorial optimisation. In: Proceedings of AAAI (2022)
11. Bonet, M.L., Levy, J., Manyà, F.: A complete calculus for MaxSAT. In: SAT. Lecture Notes in Computer Science, vol. 4121, pp. 240–251. Springer (2006)
12. Bonet, M.L., Levy, J., Manyà, F.: Resolution for MaxSAT. Artif. Intell. **171**(8-9), 606–618 (2007). https://doi.org/10.1016/j.artint.2007.03.001, `https://doi.org/10.1016/j.artint.2007.03.001`
13. Brummayer, R., Biere, A.: Fuzzing and delta-debugging smt solvers. In: Proceedings of SMT. p. 1–5 (2009)
14. Brummayer, R., Lonsing, F., Biere, A.: Automated testing and debugging of SAT and QBF solvers. In: Proceedings of SAT. pp. 44–57 (2010)
15. Cruz-Filipe, L., Heule, M.J.H., Jr., W.A.H., Kaufmann, M., Schneider-Kamp, P.: Efficient certified RAT verification. In: Proceedings of CADE. pp. 220–236 (2017)
16. Cruz-Filipe, L., Marques-Silva, J., Schneider-Kamp, P.: Efficient certified resolution proof checking. In: TACAS (1). LNCS, vol. 10205, pp. 118–135 (2017)
17. Eén, N., Sörensson, N.: Translating pseudo-Boolean constraints into SAT. JSAT **2**(1-4), 1–26 (2006)
18. Elffers, J., Gocht, S., McCreesh, C., Nordström, J.: Justifying all differences using pseudo-boolean reasoning. In: Proceedings of AAAI. pp. 1486–1494 (2020)
19. Evaluations, M.: `https://maxsat-evaluations.github.io/`
20. Gocht, S., McBride, R., McCreesh, C., Nordström, J., Prosser, P., Trimble, J.: Certifying solvers for clique and maximum common (connected) subgraph problems. In: CP. LNCS, vol. 12333, pp. 338–357. Springer (2020)
21. Gocht, S., McCreesh, C., Nordström, J.: Subgraph isomorphism meets cutting planes: Solving with certified solutions. In: Proceedings of IJCAI. pp. 1134–1140 (2020)

22. Gocht, S., McCreesh, C., Nordström, J.: An auditable constraint programming solver. In: CP. LIPIcs, vol. 235, pp. 25:1–25:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022)
23. Gocht, S., Nordström, J.: Certifying parity reasoning efficiently using pseudo-boolean proofs. In: Proceedings of AAAI. pp. 3768–3777 (2021)
24. Goldberg, E.I., Novikov, Y.: Verification of proofs of unsatisfiability for CNF formulas. In: Proceedings of DATE. pp. 10886–10891 (2003)
25. Heras, F., Larrosa, J.: New inference rules for efficient MaxSAT solving. In: AAAI. pp. 68–73. AAAI Press (2006)
26. Ignatiev, A., Morgado, A., Marques-Silva, J.: RC2: an efficient maxsat solver. J. Satisf. Boolean Model. Comput. **11**(1), 53–64 (2019)
27. Ihalainen, H., Berg, J., Järvisalo, M.: Refined core relaxation for core-guided maxsat solving. In: Proceedings of CP. pp. 28:1–28:19 (2021)
28. Koshimura, M., Zhang, T., Fujita, H., Hasegawa, R.: Qmaxsat: A partial MaxSAT solver. J. Satisf. Boolean Model. Comput. **8**(1/2), 95–100 (2012)
29. Larrosa, J., Heras, F.: Resolution in MaxSAT and its relation to local consistency in weighted csps. In: IJCAI. pp. 193–198. Professional Book Center (2005)
30. Larrosa, J., Nieuwenhuis, R., Oliveras, A., Rodríguez-Carbonell, E.: A framework for certified boolean branch-and-bound optimization. J. Autom. Reason. **46**(1), 81–102 (2011)
31. Li, C.M., Manyà, F.: Maxsat, hard and soft constraints. In: Biere et al. [9], pp. 903–927. https://doi.org/10.3233/FAIA201007, `https://doi.org/10.3233/FAIA201007`
32. Li, C.M., Manyà, F.: Inference in maxsat and minsat. In: The Logic of Software. A Tasting Menu of Formal Methods. Lecture Notes in Computer Science, vol. 13360, pp. 350–369. Springer (2022)
33. Li, C.M., Manyà, F., Soler, J.R.: A clause tableau calculus for maxsat. In: IJCAI. pp. 766–772. IJCAI/AAAI Press (2016)
34. Li, S., Coll, J., Habet, D., Li, C., Manyà, F.: A tableau calculus for maxsat based on resolution. In: CCIA. Frontiers in Artificial Intelligence and Applications, vol. 356, pp. 35–44. IOS Press (2022)
35. McConnell, R.M., Mehlhorn, K., Näher, S., Schweitzer, P.: Certifying algorithms. Comput. Sci. Rev. **5**(2), 119–161 (2011)
36. Morgado, A., Marques-Silva, J.: On validating boolean optimizers. In: ICTAI. pp. 924–926. IEEE Computer Society (2011)
37. Oertel, A.: Personal communication (August 2023)
38. Ogawa, T., Liu, Y., Hasegawa, R., Koshimura, M., Fujita, H.: Modulo based CNF encoding of cardinality constraints and its application to maxsat solvers. In: ICTAI. pp. 9–17. IEEE Computer Society (2013)
39. Py, M., Cherif, M.S., Habet, D.: A proof builder for MaxSAT. In: SAT. LNCS, vol. 12831, pp. 488–498. Springer (2021)
40. Vandesande, D.: Towards Certified MaxSAT Solving: Certified MaxSAT solving with SAT oracles and encodings of pseudo-Boolean constraints. Master's thesis, Vrije Universiteit Brussel (2023), `https://researchportal.vub.be/nl/studentTheses/towards-certified-maxsat-solving`
41. Vandesande, D., De Wulf, W., Bogaerts, B.: QMaxSATpb: A certified MaxSAT solver. In: Proceedings of LPNMR. pp. 429–442 (2022)
42. VeriPB: Verifier for pseudo-Boolean proofs. `https://gitlab.com/MIAOresearch/software/VeriPB`

43. Warners, J.P.: A linear-time transformation of linear inequalities into conjunctive normal form. Inf. Process. Lett. **68**(2), 63–69 (1998). https://doi.org/10.1016/S0020-0190(98)00144-6, `https://doi.org/10.1016/S0020-0190(98)00144-6`
44. Wetzler, N., Heule, M., Jr., W.A.H.: Drat-trim: Efficient checking and trimming using expressive clausal proofs. In: Proceedings of SAT. pp. 422–429 (2014)