Capturing Formula Isomorphism with Structure-based Hashing

markus.iser@kit.edu
Institut für Theoretische Informatik, Algorithmik II

I. Introduction

SAT benchmark instances are used to compare and evaluate the performance of state-of-the-art SAT solvers, e.g., in international competitive events [1]. Most experiments in research on SAT solving are based on benchmark instances submitted to and compiled for the annual SAT competitions [8].

Attributes of benchmark instances are used for instance classification in order to solve the per-instance algorithm selection problem [19], [2], [6], [3] and have also been employed to reduce redundancy in experimentation [16].

II. Global Benchmark Database

In our project "Global Benchmark Database" (GBD), we collect attributes of SAT instances and develop tools to organize, distribute and query that data [9]. GBD Tools include the GBD command-line tool `gbd` and the GBD web services `gbd-server` [10]. For contributions, we maintain a public repository on Github [11]. Both applications are available in the Python Package Index (PyPI) [12].

We use the hash-based instance identifier GBD Hash to identify SAT benchmark instances which are given in DIMACS CNF [7]. Associating benchmark instance attributes with GBD Hash has the advantage that it becomes easy to exchange and aggregate meta-information about benchmark instances, such as their problem family, structural measures or algorithm runtimes.

III. Scrambled Formulas

The performance of CDCL SAT solvers largely depends on heuristics [17], [14], e.g., branching and forgetting heuristics [13], and the most successful heuristics implicitly exploit the structure of many instances which are generated in industrial practice [20].

The diversity of algorithms, heuristics and configuration parameters used in practical SAT solving is subject to manual or automatic configuration based on experimental data, and some approaches are adaptive based on automatic instance classification [6].

In order to avoid overfitting to a known set of benchmark instances, these are often scrambled in past SAT Competitions, e.g., by shuffling the ordering of variables or clauses. The effect of scrambling on the runtime of a specific SAT solver can be tremendous [5]. But permutations of SAT instances can also be used to analyze the stability of an algorithm's performance on specific types of formulas.

IV. The Task

In order to capture these equivalence classes, dedicated identifiers which are invariant to the shuffling can be associated with benchmark instances. For example, in order to generate an identifier which is invariant with respect to shuffling, one could sort the literals and clauses before hashing them. Solutions are less obvious if we allow flipping of literal polarities.

When it comes to variable renaming, structural properties become important. The problem to see if formulas are isomorphic with respect to renaming reduces to a graph isomorphism problem for hypergraphs, such that we are entering the realm of GI completeness [4].

The goal in this research project is the specification of a hash-based identifier for CNF formulas which is invariant to methods of formula scrambling, i.e., clause and literal reordering, flipping of literal polarities and renaming of variables [18], [15].

We expect a thorough evaluation of several possible hash-based identifiers of equivalence classes of CNF formulas for a well-founded specification of an identifier that could be integrated in GBD.

A. Theory Part

Find and analyze existing approaches for hypergraph and graph isomporphism. Find ways to adapt them for the task at hand, e.g., by using the bipartite representation of a hypergraph. Which of the approaches are most suitable and why?

B. Practice Part

Implement your approaches very efficiently such that they could be integrated in GBD. Write a formula perturbator and evaluate your approaches on real SAT instances.

References

[1] SAT Competition Website, http://www.satcompetition.org/
[2] Alfonso, E.M., Manthey, N.: New cnf features and formula classification. In: Fifth Pragmatics of SAT workshop, a workshop of the SAT 2014 conference, July 13, 2014, Vienna, Austria (2014)
[3] Ansótegui, C., Bonet, M.L., Giráldez-Cru, J., Levy, J.: Structure features for SAT instances classification. J. Applied Logic **23**, 27–39 (2017)
[4] Babai, L.: Graph isomorphism in quasipolynomial time. CoRR **abs/1512.03547** (2015), http://arxiv.org/abs/1512.03547
[5] Biere, A., Heule, M.: The effect of scrambling cnfs. In: Proceedings of Pragmatics of SAT 2018, Oxford, UK, July 7, 2018. EPiC Series in Computing, vol. 59, pp. 111–126. EasyChair (2018)

[6] Bischl, B., Kerschke, P., Kotthoff, L., Lindauer, M.T., Malitsky, Y., Fréchette, A., Hoos, H.H., Hutter, F., Leyton-Brown, K., Tierney, K., Vanschoren, J.: Aslib: A benchmark library for algorithm selection. Artif. Intell. **237**, 41–58 (2016)

[7] DIMACS: Satisfiability suggested format (1993), http://www.domagoj-babic.com/uploads/ResearchProjects/Spear/dimacs-cnf.pdf

[8] Heule, M.J.H., Järvisalo, M., Suda, M.: Proceedings of sat competition 2019; solver and benchmark descriptions (2019)

[9] Iser, M., Sinz, C.: A problem meta-data library for research in SAT. In: Proceedings of Pragmatics of SAT 2018, Oxford, UK, July 7, 2018. pp. 144–152 (2018)

[10] Iser, M., Sinz, C., Springer, L., Heil, M.: GBD server, https://gbd.iti.kit.edu

[11] Iser, M., Springer, L.: GBD, https://github.com/Udopia/gbd

[12] Iser, M., Springer, L.: Global Benchmark Database Tool, https://pypi.org/project/global-benchmark-database-tool/

[13] Jamali, S., Mitchell, D.: Centrality-based improvements to CDCL heuristics. In: Theory and Applications of Satisfiability Testing - SAT 2018 - 21st International Conference, SAT 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9-12, 2018, Proceedings. pp. 122–131 (2018)

[14] Katebi, H., Sakallah, K.A., Silva, J.P.M.: Empirical study of the anatomy of modern sat solvers. In: Theory and Applications of Satisfiability Testing - SAT 2011 - 14th International Conference, SAT 2011, Ann Arbor, MI, USA, June 19-22, 2011. Proceedings. pp. 343–356 (2011)

[15] Kundu, A., Bertino, E.: On hashing graphs. IACR Cryptology ePrint Archive **2012**, 352 (2012), http://eprint.iacr.org/2012/352

[16] Möhle, S., Manthey, N.: Better evaluations by analyzing benchmark structure. In: Seventh Pragmatics of SAT workshop, a workshop of the SAT 2016 conference, July 4th, 2016, Bordeaux, France (2016)

[17] Silva, J.P.M.: The impact of branching heuristics in propositional satisfiability algorithms. In: Progress in Artificial Intelligence, 9th Portuguese Conference on Artificial Intelligence, EPIA '99, Évora, Portugal, September 21-24, 1999, Proceedings. pp. 62–74 (1999)

[18] Wang, X., Huan, J., Smalter, A.M., Lushington, G.H.: Application of kernel functions for accurate similarity search in large chemical databases. In: 2009 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2009, Washington, DC, USA, November 1-4, 2009, Proceedings. pp. 356–361. IEEE Computer Society (2009)

[19] Xu, L., Hutter, F., Hoos, H.H., Leyton-Brown, K.: Satzilla: Portfolio-based algorithm selection for SAT. CoRR **abs/1111.2249** (2011), http://arxiv.org/abs/1111.2249

[20] Zulkoski, E., Martins, R., Wintersteiger, C.M., Liang, J.H., Czarnecki, K., Ganesh, V.: The effect of structural measures and merges on SAT solver performance. In: Principles and Practice of Constraint Programming - 24th International Conference, CP 2018, Lille, France, August 27-31, 2018, Proceedings. pp. 436–452 (2018)