SAT-Based Enumeration Of Solutions To The Yang-Baxter Equation

Daimy Van Caudenberg[1][0000−0002−7975−4838], Leandro Vendramin[2][0000−0003−0954−7785], Bart Bogaerts[1][0000−0003−3460−4251], and Samuele Pollaci[1][0000−0002−2914−787X]

[1] Department of Computer Science, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel
`Daimy.Stefanie.Van.Caudenberg,Bart.Bogaerts,Samuele.Pollaci@vub.be`
[2] Department of Mathematics and Data Science, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel
`Leandro.Vendramin@vub.be`

When mathematicians develop theories, it is important to have examples of the structures that are studied. These can be used to generate conjectures about the structures at hand, or as counterexamples to previously generated conjectures. In case the objects at hand are finite, combinatorial search techniques can be used to generate some of them, or even to enumerate all of them.

In this thesis, we construct a database containing a subset of finite solutions to the Yang-Baxter equation (YBE). This equation was introduced in the context of statistical and quantum mechanics [15, 4], however, applications in knot theory, quantum group theory, cryptography etc. [11, 12] are known as well. Even though the YBE is an active research area, enumerating all solutions is still an open problem. One well-researched subset of solutions is the set of combinatorial solutions [9]. However, without any additional assumptions, even the task of finding these combinatorial solutions becomes highly unpredictable. Therefore, the focus often lies on enumerating solutions that meet specific extra assumptions. Given the combinatorial nature of the problem, one can gain a richer understanding of the equation's behavior and its connections to various mathematical structures by incorporating group actions. Non-degenerate solutions for example, have deep connections with various topics in pure mathematics, especially group and ring theory. Hence, we limit our focus to the enumeration of finite, non-degenerate, involutive combinatorial solutions. We refer to [14] for a friendly introduction.

In this thesis, we have chosen to express the problem as a Boolean satisfiability (SAT) problem. Handling isomorphisms has a long history in SAT, with various tools being used to exploit so-called symmetries of the given propositional formula either before search (eg. [2, 8, 3]) or during the search (eg. [13, 5, 7, 10]). The SAT modulo symmetries (SMS) framework [10] stands out in this list by focusing on the enumeration of satisfying, non-isomorphic assignments. It was designed with use cases in mathematics in mind and in particular it was first used to enumerate graphs with certain interesting properties. The core idea underlying SMS is that we can *(1)* encode as a propositional formula what it means to be a suitable mathematical structure and *(2)* force a SAT solver, during search, to generate canonical representations of each of the classes of isomorphic solutions. The second point is achieved by designing a procedure that is aware of the isomorphisms of the problem at hand, known as the minimality check, which takes the state of the SAT solver (a partial interpretation) and checks whether the current assignment can still be extended to a complete assignment that represents a solution that is lexicographically minimal (among all solutions isomorphic to it). If not, it forces the solver to abort the current branch of the search tree by analyzing why this is no longer possible and learning a new clause that is then added to the solver's working formula. In general this minimality check is incomplete, but guaranteed to be complete when ran on complete assignments. This minimality check needs to be designed for each application, taking into account the (encoding of) the mathematical problem at hand as well as the structure of the set of isomorphisms.

We have implemented a minimality check that allows us to reason about partially constructed cycle sets, a mathematical structure equivalent to a specific subset of YBE solutions. We have experimentally verified this new enumeration tool against an existing approach introduced in [1]. Here, they represent the problem at hand as a higher-level constraint program and take the isomorphisms into account by adding extra constraints before the search starts, combined with a final isomorphism check. Using this technique, the authors were able to enumerate finite, non-degenerate, involutive combinatorial solutions over sets up to size 10. Our methods outperform the state-of-the-art by an order of magnitude and we have good hopes for pushing them further in order to enumerate all solutions of size 11.

In the future, the methods we used can be extended to the construction of other combinatorial structures similar to those considered here. This includes racks and quandles, which are used in topology to construct invariants of knots; arbitrary solutions (e.g. non-involutive or with relaxed degeneracy conditions); and objects that appear in algebraic logic, especially L-algebras.

One important question that might remain is why one should trust our implementation, except from the fact that up to size 10 our results coincide with what is known so far. In combinatorial optimization, proof logging, which is the idea that solvers should not just output a solution (or a set of solutions), but also a machine-checkable proof that this answer is indeed correct, is gaining popularity. The SAT solver underlying SMS supports some form of proof logging, only guaranteeing us that the SAT solver did not make any mistakes, and not providing any guarantees whatsoever on the encoding or on the correctness of the custom propagator. The most promising approach to achieve proof logging for SMS appears to be VeriPB, which was recently used to certify static symmetry breaking [6]. However, there are many challenges on the road ahead, providing true trustworthy proof logging for isomorphism-free generation appears to be a major challenge.

Acknowledgments. This work was partially supported by Fonds Wetenschappelijk Onderzoek – Vlaanderen (projects G070521N and G004124N) and by the project OZR3762 of Vrije Universiteit Brussel.

References

1. Akgün, Ö., Mereb, M., Vendramin, L.: Enumeration of set-theoretic solutions to the yang–baxter equation. Mathematics of Computation (jan 2022). https://doi.org/10.1090/mcom/3696, https://doi.org/10.1090/mcom/3696
2. Aloul, F.A., Sakallah, K.A., Markov, I.L.: Efficient symmetry breaking for Boolean satisfiability. IEEE Trans. Computers **55**(5), 549–558 (2006)
3. Anders, M., Schweitzer, P., Stieß, J.: Engineering a preprocessor for symmetry detection. In: SEA. LIPIcs, vol. 265, pp. 1:1–1:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2023)
4. Baxter, R.J.: Partition function of the eight-vertex lattice model. Annals of Physics **70**(1), 193–228 (1972). https://doi.org/https://doi.org/10.1016/0003-4916(72)90335-1, https://www.sciencedirect.com/science/article/pii/0003491672903351
5. Benhamou, B., Nabhani, T., Ostrowski, R., Saïdi, M.R.: Enhancing clause learning by symmetry in SAT solvers. In: ICTAI (1). pp. 329–335. IEEE Computer Society (2010)
6. Bogaerts, B., Gocht, S., McCreesh, C., Nordström, J.: Certified dominance and symmetry breaking for combinatorial optimisation. J. Artif. Intell. Res. **77**, 1539–1589 (2023)
7. Devriendt, J., Bogaerts, B., Bruynooghe, M.: Symmetric explanation learning: Effective dynamic symmetry handling for SAT. In: SAT. LNCS, vol. 10491, pp. 83–100. Springer (2017)
8. Devriendt, J., Bogaerts, B., Bruynooghe, M., Denecker, M.: Improved static symmetry breaking for SAT. In: SAT. LNCS, vol. 9710, pp. 104–122. Springer (2016)
9. Drinfeld, V.G.: On some unsolved problems in quantum group theory, p. 1–8. Springer Berlin Heidelberg (1992). https://doi.org/10.1007/bfb0101175, http://dx.doi.org/10.1007/BFb0101175
10. Kirchweger, M., Szeider, S.: SAT modulo symmetries for graph generation. In: Michel, L.D. (ed.) 27th International Conference on Principles and Practice of Constraint Programming, CP 2021, Montpellier, France (Virtual Conference), October 25-29, 2021. LIPIcs, vol. 210, pp. 34:1–34:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPICS.CP.2021.34, https://doi.org/10.4230/LIPIcs.CP.2021.34
11. Nichita, F.F.: Yang–baxter equations, computational methods and applications. Axioms **4**(4), 423–435 (2015). https://doi.org/10.3390/axioms4040423, https://www.mdpi.com/2075-1680/4/4/423
12. Perk, J.H., Au-Yang, H.: Yang–baxter equations. In: Françoise, J.P., Naber, G.L., Tsun, T.S. (eds.) Encyclopedia of Mathematical Physics, pp. 465–473. Academic Press, Oxford (2006). https://doi.org/https://doi.org/10.1016/B0-12-512666-2/00191-7, https://www.sciencedirect.com/science/article/pii/B0125126662001917
13. Sabharwal, A.: Symchaff: exploiting symmetry in a structure-aware satisfiability solver. Constraints An Int. J. **14**(4), 478–505 (2009)
14. Vendramin, L.: What is... a skew brace? Notices Amer. Math. Soc. **71**(1), 65–67 (2024)
15. Yang, C.N.: Some exact results for the many-body problem in one dimension with repulsive delta-function interaction. Phys. Rev. Lett. **19**, 1312–1315 (Dec 1967). https://doi.org/10.1103/PhysRevLett.19.1312, https://link.aps.org/doi/10.1103/PhysRevLett.19.1312