Solving Hard Mizar Problems with Instantiation and Strategy Invention *

Jan Jakubův[1,2], Mikoláš Janota[1], and Josef Urban[1]

[1] Czech Technical University in Prague, Prague, Czech Republic
jakubuv@gmail.com
[2] University of Innsbruck, Innsbruck, Austria

1 Introduction: Mizar, ATPs, Hammers

In this work, we prove over 3000 previously ATP-unproved Mizar/MPTP problems by using several ATP and AI methods. First, we start to experiment with the cvc5 SMT solver which uses several instantiation-based heuristics that differ from the superposition-based systems, that were previously applied to Mizar, and add many new solutions. Then we use automated strategy invention system Grackle to develop cvc5 strategies that largely improve cvc5's performance on the hard problems. In particular, the best invented strategy solves over 14% more problems than the best previously available cvc5 strategy. We also show that different clausification methods have a high impact on such instantiation-based methods, again producing many new solutions. In total, the methods raise the number of ATP-solved Mizar problems from 75% to above 80%. This is a new milestone over the Mizar large-theory benchmark and a large strengthening of the hammer methods for Mizar.

The Mizar Mathematical Library (MML) [1] is one of the earliest large libraries of formal mathematics, containing a wide selection of lemmas and theorems from various areas of mathematics. The MML and the Mizar system [26, 2, 15] has been used as a source of automated theorem proving (ATP) [31] problems for over 25 years, starting with the export of several Mizar articles done by the ILF system [10, 9]. Since 2003, the MPTP system [36, 37] has been used to export the MML in the DFG [16] and later TPTP [35] formats. In the earliest (2003) ATP experiments over the whole library, state-of-the-art ATPs could prove about 40% of these problems when their premises were limited to those used in the human-written Mizar proofs (the so called *bushy*[1], i.e., easier, mode).

Since 2013, a fixed version of the MML (1147) and MPTP consisting of 57880 problems has been used as a large benchmark for ATPs and related hammer [6] (large-theory) methods over Mizar [29, 21, 34, 30, 17, 8]. When using many ATP and premise-selection methods, 56.2% of the problems could be proved in [22]. This was recently raised to 75.5% [19], mainly by using the learning-guided E [32] (ENIGMA [20, 13]) and Vampire [25] (Deepire [33]) systems.

Both E and Vampire are mainly saturation-style superposition systems. In the recent years, instantiation-based systems and SMTs such as cvc5 [3], iProver [24] and Z3 [11] are however becoming competitive even for problems that do not contain explicit theories in the SMT sense [5, 12, 14]. The problems that they solve are often complementary to those solved by the superposition-based systems.

*Supported by the Czech MEYS under the ERC CZ project no. LL1902 *POSTMAN*, by the European Union under the project *ROBOPROX* (reg. no. CZ.02.01.01/00/22_008/0004590), Amazon Research Awards, EU ICT-48 2020 project no. 952215 *TAILOR*, ERC PoC grant no. 101156734 *FormalWeb3*, and by the Czech Science Foundation project no. 24-12759S.

[1] https://tptp.org/MPTPChallenge

2 Summary of the Involved Methods

We employ instantiation-based methods in cvc5 to solve automatically as many hard Mizar problems as possible. Our main result is that the set of ATP-provable MPTP problems has been increased by over 3,000, from 75.5% to 80.7%. All these problems are proved by the cvc5 system which we improve in several ways. First, we use the Grackle system [18] to automatically invent stronger strategies for MPTP. Our best strategy outperforms the previously best cvc5 strategy by 14% and our best 7-strategy portfolio solves 8.8% more problems than the corresponding CASC portfolio. We also combine strategy development with alternative clausification methods. This turns out to have a surprisingly high impact on the instantiation-based system, contributing many new solutions. Finally, we obtain further solutions by modifying the problems with premise selection. Ultimately, these methods together double the number of the previously ATP-unproved Mizar problems solved by cvc5 from 1,534 to 3,021. We show that the methods extend to previously unseen Mizar problems.

Grackle Strategy Invention. Grackle [18] is a system for the automated invention of a portfolio of solver strategies targeted to selected benchmark problems. A user provides a set of benchmark problems and Grackle can automatically discover a set of diverse solver strategies that maximize the number of solved benchmark problems. Grackle supports invention of good-performing strategies for several solvers, including ATP solvers E [32], Vampire [25], Lash [7], and an SMT solver Bitwuzla [27]. Support for additional solvers can be easily added by providing a parametrization of the solver strategy space, and by implementing a simple wrapper to launch the solver. In this paper, we extend Grackle to support an SMT solver cvc5 [3], and we evaluate its capabilities on a first-order translation of Mizar problems.

Different Clausification Methods. The Mizar problems are given as TPTP [35] problems in first-order logic (FOF). For cvc5 we translate them to the SMT2 language [4] in the theory of uninterpreted functions (UF). By default, cvc5 converts to clausal normal form (CNF) internally but since instantiation-based heuristics seem sensitive to problem reformulation, we also experiment with external clausification. This gives us syntactically different variants of the problems and we can test whether cvc5 benefits from such alternative ways of clausification. We use E as the external clausifier and we construct two more problem variants. The first variant is produced by using E's default clausification parameters. The second variant uses much more aggressive introduction of definitions for frequent subformulas, introducing a new definition if a subformula appears at least four times.

Effects of Premise Selection. Based on the success with problem reformulation, we perform additional experiments, this time with different premise selection methods developed in our prior work [19]. Namely, we evaluate Grackle and baseline strategies on the *bushy* variants of the problems, on the strongest GNN (graph neural network [28]) premise selection slices, and on LightGBM [23] premise selection slices. These variants were found complementary in our previous experiments [19].

Conclusions. In the end, we have solved **3,021** (21.3%) of the remaining 14,163 hard Mizar problems, raising the percentage of automatically proved Mizar problems from 75.5% to **80.7%**. This was mainly done by automatically inventing suitable instantiation-based strategies for the cvc5 solver, using our Grackle system. Further improvements were obtained by using alternative clausifications of the problems, and also alternative premise selections. Such problem transformations have a surprisingly large effect on the instantiation-based procedures and are likely to be explored further when creating strong portfolios for such systems.

References

[1] Grzegorz Bancerek, Czeslaw Bylinski, Adam Grabowski, Artur Kornilowicz, Roman Matuszewski, Adam Naumowicz, and Karol Pak. The role of the Mizar Mathematical Library for interactive proof development in Mizar. *J. Autom. Reason.*, 61(1-4):9–32, 2018.

[2] Grzegorz Bancerek, Czeslaw Bylinski, Adam Grabowski, Artur Kornilowicz, Roman Matuszewski, Adam Naumowicz, Karol Pak, and Josef Urban. Mizar: State-of-the-art and beyond. In Manfred Kerber, Jacques Carette, Cezary Kaliszyk, Florian Rabe, and Volker Sorge, editors, *Intelligent Computer Mathematics - International Conference, CICM 2015, Washington, DC, USA, July 13-17, 2015, Proceedings*, volume 9150 of *Lecture Notes in Computer Science*, pages 261–279. Springer, 2015.

[3] Haniel Barbosa, Clark W. Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Andrew Reynolds, Ying Sheng, Cesare Tinelli, and Yoni Zohar. cvc5: A versatile and industrial-strength SMT solver. In *TACAS (1)*, volume 13243. Springer, 2022.

[4] Clark Barrett, Aaron Stump, Cesare Tinelli, et al. The SMT-LIB standard: Version 2.0. In *Proceedings of the 8th international workshop on satisfiability modulo theories (Edinburgh, UK)*, volume 13, page 14, 2010.

[5] Jasmin Christian Blanchette, Sascha Böhme, and Lawrence C. Paulson. Extending sledgehammer with SMT solvers. *J. Autom. Reason.*, 51(1):109–128, 2013.

[6] Jasmin Christian Blanchette, Cezary Kaliszyk, Lawrence C. Paulson, and Josef Urban. Hammering towards QED. *J. Formalized Reasoning*, 9(1):101–148, 2016.

[7] Chad E. Brown and Cezary Kaliszyk. Lash 1.0 (system description). In *IJCAR*, volume 13385 of *Lecture Notes in Computer Science*, pages 350–358. Springer, 2022.

[8] Karel Chvalovský, Konstantin Korovin, Jelle Piepenbrock, and Josef Urban. Guiding an instantiation prover with graph neural networks. In *LPAR*, volume 94 of *EPiC Series in Computing*, pages 112–123. EasyChair, 2023.

[9] Ingo Dahn. Interpretation of a Mizar-like logic in first-order logic. In Ricardo Caferra and Gernot Salzer, editors, *FTP (LNCS Selection)*, volume 1761 of *LNCS*, pages 137–151. Springer, 1998.

[10] Ingo Dahn and Christoph Wernhard. First order proof problems extracted from an article in the MIZAR Mathematical Library. In Maria Paola Bonacina and Ulrich Furbach, editors, *Int. Workshop on First-Order Theorem Proving (FTP'97)*, RISC-Linz Report Series No. 97-50, pages 58–62. Johannes Kepler Universität, Linz (Austria), 1997.

[11] Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3: An Efficient SMT Solver. In C. R. Ramakrishnan and Jakob Rehof, editors, *TACAS*, volume 4963 of *LNCS*, pages 337–340. Springer, 2008.

[12] Martin Desharnais, Petar Vukmirovic, Jasmin Blanchette, and Makarius Wenzel. Seventeen provers under the hammer. In *ITP*, volume 237 of *LIPIcs*, pages 8:1–8:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[13] Zarathustra Amadeus Goertzel, Karel Chvalovský, Jan Jakubův, Miroslav Olsák, and Josef Urban. Fast and slow Enigmas and parental guidance. In *FroCoS*, volume 12941 of *Lecture Notes in Computer Science*, pages 173–191. Springer, 2021.

[14] Zarathustra Amadeus Goertzel, Jan Jakubův, Cezary Kaliszyk, Miroslav Olsák, Jelle Piepenbrock, and Josef Urban. The Isabelle ENIGMA. In *ITP*, volume 237 of *LIPIcs*, pages 16:1–16:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[15] Adam Grabowski, Artur Korniłowicz, and Adam Naumowicz. Mizar in a nutshell. *J. Formalized Reasoning*, 3(2):153–245, 2010.

[16] Reiner Hähnle, Manfred Kerber, and Christoph Weidenbach. Common syntax of the DFGSchwerpunktprogramm deduction. Technical Report TR 10/96, Fakultät für Informatik, Universität Karlsruhe, Karlsruhe, Germany, 1996.

[17] Edvard K. Holden and Konstantin Korovin. Graph sequence learning for premise selection. *CoRR*, abs/2303.15642, 2023.

[18] Jan Hůla, Jan Jakubův, Mikolás Janota, and Lukás Kubej. Targeted configuration of an SMT solver. In *CICM*, volume 13467 of *Lecture Notes in Computer Science*, pages 256–271. Springer, 2022.

[19] Jan Jakubův, Karel Chvalovský, Zarathustra Amadeus Goertzel, Cezary Kaliszyk, Mirek Olsák, Bartosz Piotrowski, Stephan Schulz, Martin Suda, and Josef Urban. MizAR 60 for Mizar 50. In *ITP*, volume 268 of *LIPIcs*, pages 19:1–19:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[20] Jan Jakubův and Josef Urban. ENIGMA: efficient learning-based inference guiding machine. In Herman Geuvers, Matthew England, Osman Hasan, Florian Rabe, and Olaf Teschke, editors, *Intelligent Computer Mathematics - 10th International Conference, CICM 2017, Edinburgh, UK, July 17-21, 2017, Proceedings*, volume 10383 of *Lecture Notes in Computer Science*, pages 292–302. Springer, 2017.

[21] Jan Jakubův and Josef Urban. Hammering Mizar by learning clause guidance. In John Harrison, John O'Leary, and Andrew Tolmach, editors, *10th International Conference on Interactive Theorem Proving, ITP 2019, September 9-12, 2019, Portland, OR, USA*, volume 141 of *LIPIcs*, pages 34:1–34:8. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[22] Cezary Kaliszyk and Josef Urban. MizAR 40 for Mizar 40. *J. Autom. Reasoning*, 55(3):245–256, 2015.

[23] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*, pages 3146–3154, 2017.

[24] Konstantin Korovin. iprover - an instantiation-based theorem prover for first-order logic (system description). In *IJCAR*, volume 5195 of *Lecture Notes in Computer Science*, pages 292–298. Springer, 2008.

[25] Laura Kovács and Andrei Voronkov. First-order theorem proving and Vampire. In Natasha Sharygina and Helmut Veith, editors, *CAV*, volume 8044 of *LNCS*, pages 1–35. Springer, 2013.

[26] Roman Matuszewski and Piotr Rudnicki. Mizar: the first 30 years. *Mechanized Mathematics and Its Applications*, 4:3–24, 2005.

[27] Aina Niemetz and Mathias Preiner. Bitwuzla at the SMT-COMP 2020. *CoRR*, abs/2006.01621, 2020.

[28] Miroslav Olsák, Cezary Kaliszyk, and Josef Urban. Property invariant embedding for automated reasoning. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 1395–1402. IOS Press, 2020.

[29] Michael Rawson and Giles Reger. A neurally-guided, parallel theorem prover. In *FroCos*, volume 11715 of *Lecture Notes in Computer Science*, pages 40–56. Springer, 2019.

[30] Michael Rawson and Giles Reger. lazyCoP: Lazy paramodulation meets neurally guided search. In *TABLEAUX*, volume 12842 of *Lecture Notes in Computer Science*, pages 187–199. Springer, 2021.

[31] John Alan Robinson and Andrei Voronkov, editors. *Handbook of Automated Reasoning (in 2 volumes)*. Elsevier and MIT Press, 2001.

[32] Stephan Schulz. System description: E 1.8. In Kenneth L. McMillan, Aart Middeldorp, and Andrei Voronkov, editors, *LPAR*, volume 8312 of *LNCS*, pages 735–743. Springer, 2013.

[33] Martin Suda. Improving ENIGMA-style clause selection while learning from history. In *CADE*, volume 12699 of *Lecture Notes in Computer Science*, pages 543–561. Springer, 2021.

[34] Martin Suda. Vampire with a brain is a good ITP hammer. In *FroCoS*, volume 12941 of *Lecture Notes in Computer Science*, pages 192–209. Springer, 2021.

[35] Geoff Sutcliffe, Christian B. Suttner, and Theodor Yemenis. The TPTP problem library. In *CADE*, volume 814 of *Lecture Notes in Computer Science*, pages 252–266. Springer, 1994.

[36] Josef Urban. MPTP – Motivation, Implementation, First Experiments. *J. Autom. Reasoning*, 33(3-4):319–339, 2004.

[37] Josef Urban. MPTP 0.2: Design, implementation, and initial experiments. *J. Autom. Reasoning*, 37(1-2):21–43, 2006.