Prover9 Unleashed: Automated Configuration for Enhanced Proof Discovery*

Kristina Aleksandrova[1], Jan Jakubův[1,2], and Cezary Kaliszyk[1]

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

Introduction. While many of the state-of-art Automated Theorem Provers (ATP) like E and Vampire, were subject to extensive tuning of strategy schedules in the last decade, the classical ATP prover Prover9 has never been optimized in this direction. Both E and Vampire provide the user with an automatic mode to select good proof search strategies based on the properties of the input problem, while Prover9 provides by default only a relatively weak auto mode. Interestingly, Prover9 provides more varied means for proof control than its competitors. These means, however, must be manually investigated and that is possible only by experienced Prover9 users with a good understanding of how Prover9 works.

In this paper, we investigate the possibilities of automatic configuration of Prover9 for user-specified benchmark problems. We employ the automated strategy invention system Grackle to generate Prover9 strategies with both basic and advanced proof search options which require sophisticated strategy space features for Grackle. We test the strategy invention on AIM train/test problem collection and we show that Prover9 can outperform both E and Vampire on these problems. To test the generality of our approach we train and evaluate strategies also on TPTP problems, showing that Prover9 can achieve reasonable complementarity with other ATPs.

For many automated reasoning problems a combination of complementary strategies is significantly better than a single strategy. For this reason, many provers support configurable options that can be user specified or automatically tuned. This has been done for many provers [12, 4] and led to their good performance on various benchmarks [10]. Prover9, despite its popularity among mathematicians [5] is mostly configured manually.

We discuss the specification of the Prover9 options using the strategy invention system Grackle [4]. Apart from all the basic option [1] we include the various Prover9 specific advanced options that require adaptations to the system. We also specify multi-staged domains. Starting with a preliminary set of of basic strategies, the system derives a large number of new strategies for the AIM dataset and for subparts of TPTP and show that Prover9 can perform significantly better than the other provers on some of these datasets.

A considerable amount of effort has been dedicated to parameter tuning in state-of-the-art theorem provers (mainly unpublished, unfortunately), aiming to discover a universal proof search strategy or a portfolio of strategies. However, Grackle's primary objective differs slightly. Instead of seeking a generic strategy or portfolio that excels across all benchmarks or competition problems, Grackle endeavors to develop a set of strategies capable of solving as many problems as possible from a benchmark provided by the user. This approach proves beneficial in scenarios where users encounter problems distinct from the competition problems typically optimized for by state-of-the-art provers.

*Supported by the Czech MEYS under the ERC CZ project no. LL1902 *POSTMAN*, ERC PoC grant no. 101156734 *FormalWeb3*, and by the Czech Science Foundation project no. 24-12759S. We are grateful to Bob Veroff for valuable comments and discussions.

Grackle Portfolio Invention. Grackle[1] [4] is designed to automate the creation of a portfolio of solver strategies tailored to user-provided benchmark problems, aiming to maximize the problem-solving effectiveness. By inputting a set of benchmark problems, Grackle autonomously invents a diverse set of solver strategies to tackle as many of these problems as possible. It currently integrates various solvers, such as ATP solvers E [9], Vampire [6], Lash [3], and SMT solvers Bitwuzla [8] and cvc5 [2]. Adding support for additional solvers is straightforward: users just need to specify solver strategy parameters and implement a basic wrapper for solver execution. This paper introduces an extension of Grackle to accommodate the ATP solver Prover9 [7], and assesses its performance on various first-order benchmarks.

The strategy space is described by a set of available parameters, their potential values, and the default value for each parameter. Thus, a single strategy is represented by a set of parameter/value pairs. The responsibility of the solver wrapper lies in translating the strategy representation into the actual solver input. Typically, the parameters directly correspond with solver command line options, though advanced transformations are feasible. We describe several embeddings of advanced Prover9 options within a Grackle strategy space. Furthermore, we extend Grackle with *staged strategy invention*, where a vast strategy space is subdivided into multiple smaller spaces and tuned separately. This approach resembles hierarchical tuning in BliStrTune and EmpireTune.

Experiments. We evaluate Grackle strategy invention for Prover9 on the AIM benchmark used in the CASC 2016 ATP competition [11], consisting of 1020 training and 200 evaluation problems from a large theorem proving project in loop theory [5]. We use the training problem set for Grackle strategy invention, and we evaluate the invented strategies on the 200 evaluation problems, comparing Prover9 strategies with state-of-the-art portfolios E and Vampire, which are supposed to be universally well-performing. While Vampire solves 50, and E solves 36 of the evaluation problems, our Grackle-invented portfolio solves 91 problems within the same time limit. Notably, Prover9 solved all problems solved by the other solvers.

Next, we evaluate Grackle strategy invention on TPTP problems [10]. As an initial assessment, we launch selected Grackle strategies, as well as Vampire and E, to discover that while the overall performance of Prover9 cannot compare to that of the state-of-the-art solvers, Prover9 still provides valuable contributions. TPTP problems are divided into categories, and we discover that the most significant contribution is in the category NUM, which contains problems from Number Theory. Following the main idea behind Grackle, which aims to enhance performance in areas where one performs best, we conduct several Grackle runs on 1,094 NUM problems, using the same settings as for AIM problems. The Grackle-invented Prover9 portfolio even slightly outperforms Vampire, solving 618, while Vampire solves 611 and E solves 541 within the same time limit. Moreover, our strategies solve significantly different problems, yielding 153 problems unsolved by E or Vampire.

Conclusions. We have integrated support for Prover9 into the automated strategy invention system Grackle and assessed its capabilities across two distinct benchmark problem sets. The findings reveal that Prover9's performance can be significantly enhanced through our fully automated strategy invention process. By comparison, Prover9 in its default *auto* mode can tackle 41 AIM problems and 512 on TPTP/NUM, whereas our strategies solve 91 and 619 within the same timeframe. Surprisingly, Prover9 can even outperform state-of-the-art provers, at least on the problem domains explored in this work.

[1] https://github.com/ai4reason/grackle

References

[1] Kristina Aleksandrova. Strategy invention for Prover9. Univesity of Innsbruck Thesis, 2023.

[2] 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.

[3] 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.

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

[5] Michael K. Kinyon, Robert Veroff, and Petr Vojtechovský. Loops with abelian inner mapping groups: An application of automated deduction. In Maria Paola Bonacina and Mark E. Stickel, editors, *Automated Reasoning and Mathematics - Essays in Memory of William W. McCune*, volume 7788 of *LNCS*, pages 151–164. Springer, 2013.

[6] 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.

[7] William McCune. Prover9 and Mace4. http://www.cs.unm.edu/~mccune/prover9/, 2005–2010.

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

[9] 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.

[10] Geoff Sutcliffe. The TPTP world - infrastructure for automated reasoning. In Edmund M. Clarke and Andrei Voronkov, editors, *LPAR (Dakar)*, volume 6355 of *LNCS*, pages 1–12. Springer, 2010.

[11] Geoff Sutcliffe. The 6th IJCAR automated theorem proving system competition - CASC-J6. *AI Commun.*, 26(2):211–223, 2013.

[12] Josef Urban. BliStr: The Blind Strategymaker. In Georg Gottlob, Geoff Sutcliffe, and Andrei Voronkov, editors, *Global Conference on Artificial Intelligence, GCAI 2015, Tbilisi, Georgia, October 16-19, 2015*, volume 36 of *EPiC Series in Computing*, pages 312–319. EasyChair, 2015.