Automated Theorem Proving for Metamath*

Mario Carneiro[1], Chad Brown[2], and Josef Urban[2]

[1] Carnegie Mellon University, Pittsburgh, PA, USA
[2] Czech Technical University in Prague

Introduction Metamath [20] is a formal system developed by Norman Megill in 1990. Its largest database, `set.mm`[1], has 40338 theorems in ZFC set theory, including a diverse range of topics including analysis, topology, graph theory, number theory, Hilbert spaces, and it continues to grow steadily due to its small but active community. In the space of theorem prover languages, it is one of the simplest, by design. Metamath is one of the last formal proof systems with a large mathematical library that have not yet been translated to today's automated theorem provers (ATPs) [23]. Such translations between ITPs and ATPs are one of the main parts of *hammer systems* [4], which have become popular in the recent years, especially in the Isabelle community [21, 22, 3, 18, 8]. Hammer systems today exist also for the Coq [7, 9], HOL [10, 15], and Mizar [27, 16, 14] proof assistants. The goal of this work is to provide the first such ATP translation for Metamath, and to do the first evaluation of the potential of state-of-the-art ATP systems on the translated Metamath library. This also results in a new large mathematical benchmark for ATP systems. We also build other components of the first full Metamath hammer here, such as proof reconstruction and premise selection [1].

Translations The name "Metamath" comes from "metavariable mathematics," because the core concept is the pervasive use of metavariables over an object logic. This ability for a Metamath theorem to encode multiple α-equivalence classes of FOL theorems is known in the Metamath community as "bundling," and it poses a problem for translation to plain FOL or HOL. We use a translation of Metamath to HOL via Metamath Zero (MM0) [5]. The MM0 toolchain[2] implements a translation from Metamath to MM0 that addresses exactly the bundling issue. MM0 requires that all theorems are fully unbundled, i.e., split into separate theorems for each of the α-equivalence classes. The HOL translation is then used as an input to several versions of translation to the higher-order TPTP (TH0) format [11]. The three versions (denoted as *v1*, *v2* and *v3*) differ by providing more targeted translations of common constructors in v2 and v3. In v2, 10 constructors (such as true, false, implication, conjunction, equivalence, negation, etc.) are translated using their intended HOL meaning, and in v3 we handle 11 more constructors. For example, $\mathsf{wi}\ \varphi\ \psi$ (where wi is Metamath's implication) translates in v2 (and v3) as $\varphi' \to \psi'$ (where φ' is the v2 translation of φ and ψ' is the v2 translation of ψ). Here is how the conjecture of $\mathsf{rp\_simp2}$,[3] looks in the three translations:

- v1: $\forall\,(\varphi\ \psi\ \xi : o).\ \ \mathsf{wi}\ (\mathsf{w3a}\ \varphi\ \psi\ \xi)\ \psi$
- v2: $\forall\,(\varphi\ \psi\ \xi : o).\ \ \mathsf{w3a}\ \varphi\ \psi\ \xi\ \to\ \psi$
- v3: $\forall\,(\varphi\ \psi\ \xi : o).\ \ \varphi\ \wedge\ \psi\ \wedge\ \xi\ \to\ \psi$

We also translated Metamath into first-order theorem proving problems by interpreting propositions and terms as classes. This allows us to use the first-order Prover9 [19] system to obtain IVY proof objects which we then use to reconstruct the proofs in Metamath. Each of the v1, v2, v3 and FOL translations produce a benchmark of 40556 TPTP problems, each in the bushy (premise-minimized) and chainy (hammering) setting.

*The full paper was recently accepted to ITP 2023.
[1] https://github.com/metamath/set.mm
[2] https://github.com/digama0/mm0
[3] https://us.metamath.org/mpeuni/rp-simp2.html

Evaluation For the TH0 evaluation we use three top higher-order ATPs: (i) E prover version 2.6 [25, 24], run both in its default portfolio (auto-schedule) mode and with several strategies developed previously by strategy invention systems [26, 13] targeting ITP libraries [16, 12]. (ii) Vampire version 5980 [17, 2], using mainly default (casc2020) higher-order portfolio. We have also briefly tested some Vampire strategies in a standalone mode, (iii) Zipperposition version CASC20 [6, 28], using its default CASC'20 portfolio. We have also tried several other Zipperposition settings. For most of the experiments we have used a time limit of 60s, later trying also lower (10s) and higher (up to 1200s) times in several cases. Of the 40556 TH0 bushy problems, the ATPs can in total solve 27436, i.e., **67.65%**. The highest performance is achieved by Zipperposition which in 280s solves 62.68% (25420) of the v3 problems, and 61.53% (24959) of the v2 problems. Vampire solves 58.01% (23555) of the v3 problems in 280s and 45.57% (18482) of the v2 problems in 60s, which increases to 52.08% (21123) of the v3 problems in 60s, and to 56.65% (22976) v3 problems in 120s. E prover outperforms Vampire on v2 in 60s (21001 solved by E vs 18482 by Vampire), and even more so in 10s (20352 vs 17160). Table 1 shows the top-4 greedy cover using high time limits on unsolved problems only. We also evaluate the first

System	mode	version	time (s)	added	sum
Z	portfolio	v3	280	25420	25420
V	portfolio	v3	600	960	26380
V	portfolio	v3	1200	415	26795
E	portfolio	v3	600	279	27074

Table 1: The top 4 TH0 methods in the greedy sequence. Note that we use different (and also high) time limits and that the high-time runs are only done on previously unsolved problems.

order translation, by running Vampire, E and Prover9 on these problems. Vampire proves 15938 of them, while E and Prover9 solve 15136 and 14693 respectively. The Vampire performance can be compared to its 60s performance on the v2 higher-order problems (18482). This likely again demonstrates the efficiency of the v2 and v3 higher-order translations, because practically none of the standard logical connectives are mapped in a shallow way to their first-order logical counterparts in this first-order translation. On the chainy problems, Vampire solves 8509 v3 problems in 60s using SInE, and 12043 using k-NN selection with $k = 120$. A combination of 7 k-NN selections solves altogether 14787 problems (in general in 7 minutes).

Reconstruction, Hammer and Examples We use the FOL encoding together with Prover9 and its detailed IVY proof objects to reconstruct the ATP proofs in Metamath. k-NN followed by Vampire are used to select and minimize the premises for Prover9. The IVY proof steps (such as `instantiate, resolve`, etc.) are interpreted as Metamath proof steps, resulting in complete Metamath proof objects. The resulting mm-hammer tool is publicly available from our GitHub repository[4]. It can replay in Metamath all 15k proofs that Prover9 found. A number of examples of ATP proofs are available on our web page.[5] This includes E's proof `xmulneg1`[6] which has 127 steps in Metamath and takes 18131 given clause loops in 30 seconds to E.[7] It proves for extended reals that a product with a negative is the negative of the product. E also proves the `matinv` theorem in 12 seconds and 13052 given clause loops, which takes a 73-step proof in Metamath.[8] This states that the inverse of a matrix is its adjunct multiplied with the inverse of the determinant of the matrix if the determinant is a unit of the ring.

[4] https://github.com/digama0/mm-hammer
[5] http://grid01.ciirc.cvut.cz/~mptp/mm_prf/
[6] https://us.metamath.org/mpeuni/xmulneg1.html
[7] http://grid01.ciirc.cvut.cz/~mptp/mm_prf/mmset12407_xmulneg1.p
[8] https://us.metamath.org/mpeuni/matinv.html

References

[1] Jesse Alama, Tom Heskes, Daniel Kühlwein, Evgeni Tsivtsivadze, and Josef Urban. Premise selection for mathematics by corpus analysis and kernel methods. *J. Autom. Reasoning*, 52(2):191–213, 2014.

[2] Ahmed Bhayat and Giles Reger. A combinator-based superposition calculus for higher-order logic. In *IJCAR (1)*, volume 12166 of *Lecture Notes in Computer Science*, pages 278–296. Springer, 2020.

[3] Jasmin Christian Blanchette, Sascha Böhme, and Lawrence C. Paulson. Extending Sledgehammer with SMT solvers. In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *CADE*, volume 6803 of *LNCS*, pages 116–130. Springer, 2011.

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

[5] Mario Carneiro. Metamath zero: Designing a theorem prover prover. In *Intelligent Computer Mathematics: 13th International Conference, CICM 2020, Bertinoro, Italy, July 2631, 2020, Proceedings*, page 7188, Berlin, Heidelberg, 2020. Springer-Verlag.

[6] Simon Cruanes. *Extending Superposition with Integer Arithmetic, Structural Induction, and Beyond*. Theses, École polytechnique, September 2015.

[7] Lukasz Czajka and Cezary Kaliszyk. Hammer for coq: Automation for dependent type theory. *J. Autom. Reason.*, 61(1-4):423–453, 2018.

[8] Martin Desharnais, Petar Vukmirovi, Jasmin Blanchette, and Makarius Wenzel. Seventeen provers under the hammer, 2022. https://matryoshka-project.github.io/pubs/seventeen.pdf.

[9] Burak Ekici, Alain Mebsout, Cesare Tinelli, Chantal Keller, Guy Katz, Andrew Reynolds, and Clark W. Barrett. Smtcoq: A plug-in for integrating SMT solvers into coq. In Rupak Majumdar and Viktor Kuncak, editors, *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part II*, volume 10427 of *Lecture Notes in Computer Science*, pages 126–133. Springer, 2017.

[10] Thibault Gauthier and Cezary Kaliszyk. Premise selection and external provers for HOL4. In *Certified Programs and Proofs (CPP'15)*, LNCS. Springer, 2015. http://dx.doi.org/10.1145/2676724.2693173.

[11] Allen Van Gelder and Geoff Sutcliffe. Extending the TPTP language to higher-order logic with automated parser generation. In Ulrich Furbach and Natarajan Shankar, editors, *IJCAR*, volume 4130 of *LNCS*, pages 156–161. Springer, 2006.

[12] Thomas C. Hales, John Harrison, Sean McLaughlin, Tobias Nipkow, Steven Obua, and Roland Zumkeller. A revision of the proof of the Kepler conjecture. *Discrete & Computational Geometry*, 44(1):1–34, 2010.

[13] Jan Jakubův and Josef Urban. BliStrTune: hierarchical invention of theorem proving strategies. In Yves Bertot and Viktor Vafeiadis, editors, *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs, CPP 2017, Paris, France, January 16-17, 2017*, pages 43–52. ACM, 2017.

[14] Jan Jakubuv, Karel Chvalovský, Zarathustra Amadeus Goertzel, Cezary Kaliszyk, Mirek Olsák, Bartosz Piotrowski, Stephan Schulz, Martin Suda, and Josef Urban. Mizar 60 for mizar 50. *CoRR*, abs/2303.06686, 2023.

[15] Cezary Kaliszyk and Josef Urban. HOL(y)Hammer: Online ATP service for HOL Light. *Mathematics in Computer Science*, 9(1):5–22, 2015.

[16] Cezary Kaliszyk and Josef Urban. MizAR 40 for Mizar 40. *Journal of Automated Reasoning*, 55(3):245–256, 2015.

[17] 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.

[18] Daniel Kühlwein, Jasmin Christian Blanchette, Cezary Kaliszyk, and Josef Urban. MaSh: Machine learning for Sledgehammer. In Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie,

editors, *ITP 2013*, volume 7998 of *LNCS*, pages 35–50. Springer, 2013.

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

[20] Norman D. Megill and David A. Wheeler. *Metamath: A Computer Language for Mathematical Proofs*. Lulu Press, Morrisville, North Carolina, 2019. http://us.metamath.org/downloads/metamath.pdf.

[21] Jia Meng and Lawrence C. Paulson. Translating higher-order clauses to first-order clauses. *J. Autom. Reasoning*, 40(1):35–60, 2008.

[22] Lawrence C. Paulson and Jasmin C. Blanchette. Three years of experience with Sledgehammer, a practical link between automated and interactive theorem provers. In Geoff Sutcliffe, Stephan Schulz, and Eugenia Ternovska, editors, *Workshop on the Implementation of Logics (IWIL)*, volume 2 of *EPiC*, pages 1–11. EasyChair, 2010. Invited talk.

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

[24] 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.

[25] Stephan Schulz, Simon Cruanes, and Petar Vukmirovic. Faster, higher, stronger: E 2.3. In Pascal Fontaine, editor, *Automated Deduction - CADE 27 - 27th International Conference on Automated Deduction, Natal, Brazil, August 27-30, 2019, Proceedings*, volume 11716 of *Lecture Notes in Computer Science*, pages 495–507. Springer, 2019.

[26] 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.

[27] Josef Urban, Piotr Rudnicki, and Geoff Sutcliffe. ATP and presentation service for Mizar formalizations. *J. Autom. Reasoning*, 50:229–241, 2013.

[28] Petar Vukmirovic, Alexander Bentkamp, Jasmin Blanchette, Simon Cruanes, Visa Nummelin, and Sophie Tourret. Making higher-order superposition work. In *CADE*, volume 12699 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 2021.