The Expansion, Modernisation, and Future of the TPTP World

Geoff Sutcliffe

University of Miami, USA
geoff@cs.miami.edu

Abstract

The Thousands of Problems for Theorem Provers (TPTP) World is a rich infrastructure that supports the development, deployment, and application of Automated Theorem Proving (ATP) for classical logics. This paper describes proposed expansion and modernisation of some parts of the TPTP World: the TPTP problem library, the TSTP solution library, a new TDTP data library, new logics and languages, and improved user services. Hopefully this will attract suggestions, feedback, and offers of support to help achieve these goals.

1 Introduction

The Thousands of Problems for Theorem Provers (TPTP) World [48] is a rich infrastructure that supports the development, deployment, and application of Automated Theorem Proving (ATP) for classical logics. The TPTP World has some core components: the TPTP problem library of test problems for ATP systems [48]; the TSTP solution library of solutions to those problems [46]; the TPTP languages and SZS ontology of standards for writing ATP problems and solutions [45]; the SystemB4TPTP, SystemOnTPTP, and SystemOnTSTP online services for preparing problems, running ATP systems, and examining solutions [46]; and linked projects such as the CADE ATP System Competition (CASC) [47]. Since its first release in 1993 the ATP community has used the TPTP World (prima facie the TPTP problem library, but more deeply other components) as an appropriate and convenient infrastructure for ATP system development, evaluation, and application. Over the years the TPTP problem library has been used as a conduit for ATP users to provide samples of their problems to ATP system developers - this exposes the problems to the developers, who then improve their systems' performances on the problems, which completes a cycle to provide the users with more effective tools.

Recently I submitted a grant proposal to the National Science Foundation for "Expansion and Modernisation of the TPTP World", summarized as follows:

> The TPTP World will be expanded to add new features, logics, and services. The TPTP World will be modernised to take advantage of new technologies, to improve access to and use of the TPTP World. There will be three aspects to the enhancement of the infrastructure:

- Libraries. The existing TPTP and TSTP libraries will be expanded and modernised [this includes moving the TPTP World onto GitHub], and a new TDTP data library will be added to support research and development of machine learning in ATP.
- New Logics. Three new logics/logic families will be added to the TPTP World. These are the extended typed first-order logic, modal logics, and multi-valued logics.
- User Services. The various TPTP World tools that are used to manipulate and query the TPTP World data, and services that provide (online) access to the TPTP World, will be expanded and modernised.

> The TPTP World has evolved over the years to a sophisticated state, and it would be an unreasonable undertaking to build it again from scratch. The proposed expansion and modernisation will provide new richness to the TPTP World, which will not be equally available elsewhere. The enhancements are well justified, based on community input and demand. The enhancements respond well to emerging and important topics in the development and application of ATP, in areas that might not otherwise be addressed by automated reasoning, particularly areas that are emerging in the burgeoning AI industry (e.g., healthcare, privacy, natural sciences, etc.). The enhancements will be integrated into the established framework of the TPTP world, respecting the structures that already exist and have been adopted by the ATP community, while adding features that will attract new users.

Sadly the grant proposal was declined[1], the reasons for which are a topic for another presentation.

This paper provides some details of what was proposed, in order to inform the ATP and TPTP World user communities of my plans, and hopefully attract suggestions, feedback, and offers of support to help achieve these goals.

2 The TPTP Problem Library

The following expansions and modernisations are planned for the TPTP problem library.

• Problem Collection

The ongoing success and utility of the TPTP problem library depends on contributions of problems from the ATP community, who are regularly encouraged to make contributions of all types of problems, e.g., [49]. This will continue as part of the expansion of the TPTP World.

• Problem Statistics

Each problem in the TPTP problem library has a header section containing metadata for users, formatted as comments in four parts. The third part provides syntactic statistics of the problem, such as the numbers of formulae, quantified variables, type declarations, arithmetic terms, etc. The statistics are useful for selecting problems with certain characteristics. The statistics are mined from the problems' formulae, and this is not always straightforward. There are situations in which it is not clear what exactly should be counted (e.g., should the symbols on the left hand side of a definition be counted as primitives?). As such the statistics are

[1] I am very very grateful for the help and support I received writing that proposal, and for the feedback I received when it was declined – the future of the TPTP World is bright!

currently not completely precise, and will be overhauled. The choice of statistical measures will be reviewed and made consistent across the TPTP logics (as far as logically possible). Each statistical measure will be precisely defined, and the program that extracts the statistics from the formulae (based on the TPTP4X parser - see Section 4.1) will be updated.

- **(Co)Datatypes and Recursive Functions**

Freely generated algebraic data types and codatatypes are useful for representing finite and infinite data structures in computer science applications [31, 27]. They can be implemented efficiently and enjoy properties that can be exploited by ATP systems [8]. The main use of (co)datatypes in ATP is software verification, where describing properties of functional programs naturally uses (co)datatypes. Discussions with Simon Robillard, Martin Suda, Jasmin Blanchette, and Dmitriy Traytel, at the 9th International Conference on Automated Reasoning in Oxford, 2018 led to a preliminary proposal for adding (co)datatypes to the TPTP World. Most recently, Laura Kovacs has reemphasized the importance of adding (co)datatypes and recursive functions to the TPTP, to support research in software verification. (Co)datatypes are already present in the SMT language used in the SMT-LIB [4]; adding (co)datatypes to the TPTP World will allow SMT-LIB users to use the TPTP World in more ways, and conversely developers of ATP systems in the TPTP world will be motivated to add features for reasoning over (co)datatypes thus enabling them to attempt more SMT-LIB problems.

- **Real-time Axioms**

Almost all efforts in ATP assume a static set of axioms and a conjecture. In many real-world situations data is dynamic, e.g., stock prices, weather conditions, etc. In order to support development of ATP with dynamic real-time data, the TPTP problem library will be expanded to include a framework that provides streams of "real-time axioms" whose formulae change over time, so that an ATP system can resample the axioms periodically to get updated versions. For example, the axiom `price('APPL',date(16,03,2020),'USD',57.31)` would later be updated in real-time to `price('APPL',date(28,12,2020),'USD',132.69)`, and an ATP system might then (which it could not before) be able to prove `sell('APPL',date(28,12,2020))`. In addition to actual real-time retrieval of data, static files of time-stamped axioms will be captured, and the TPTP4X tool (see Section 4.1) will be modified to simulate real-time axioms according to the time stamps (this will be useful for repeatability of experiments). Initial work on this concept was done with Martin Suda in 2009-2010 [44, 52] in the SPASS ATP system [58].

- **Linking with StarExec**

StarExec [43] is a cross community logic solving service developed jointly at the University of Iowa and the University of Miami (funded by NSF CNS-CI grants, 1058925 and 1730419). StarExec facilitates the experimental evaluation of automated tools that are based on formal reasoning. There are instances of StarExec at the University of Iowa and the University of Miami, with the Miami instance being focussed on the TPTP World. StarExec has become a widely used infrastructure, hosting individual researcher's experiments through to community-wide competitions (e.g., CADE ATP System Competition (CASC) [47], which uses problems from the TPTP problem library). Facilities will be created to package subsets of the TPTP problem library (see Section 4.1) in a form suitable for upload to StarExec, and a tool will be written to automate the upload. This facility will be the first step in the automated update of the TSTP solution library.

• **Repository and Archiving**
Currently the TPTP problem library is stored on a server at the University of Miami. The TPTP problem library will be added to a Github organization called “TPTP World” (see Section 4.2).

2.1 The TDTP Data Library

Machine Learning (ML) is emerging as a game changer in ATP, and an increasing number of ATP systems that take advantage of ML are being developed [56, 17, 40, 35]. External tools that can use ML to help with the tuning of ATP systems have also been developed [25, 55]. Concrete examples of the ways ML can be used in ATP include: axiom selection, to select a small but adequate subset of the very many axioms available for proving a conjecture [24, 57, 35, 36]; search guidance, to guide the choice of what (inference) action the ATP system should take next, e.g., “given clause” selection in saturation-based ATP systems [18, 11, 17], and branch selection in tableau based systems [40]; direct logical reasoning [13, 42, 41]; and learning assisted reasoning [20, 2]. In order to provide support for the development, evaluation, and deployment of ATP systems that include ML components, the TPTP World will be extended to provide the large corpora of (ATP) data required for ML, and infrastructure to support access to and use of the data. This will be embodied in the new “Tons of Data for Theorem Provers” (TDTP) data library. There have already been successful efforts building corpora suitable for developing ML support for ATP, including [39, 54, 33, 10, 26, 19, 9]. The corpora resulting from some of these efforts are publicly available, and some use the TPTP languages and standards. However, to the best of my knowledge there have been only limited prior efforts aiming to combine the collection of ML data with infrastructure for accessing that data [3, 16]. Neither of those cited were built in the context of an established research infrastructure like the TPTP World, nor aimed at supporting the ATP research community.

An initial effort to build something like the TDTP data library was started in 2016 when Cezary Kaliszyk (Computational Logic Group, University of Innsbruck, Austria) visited Miami. The effort took into account the two somewhat orthogonal views of the data, viz., the TPTP view in which the problems and solutions would be structured in the same way as the TPTP problem library and the TSTP solution library, and the ML view in which the problems, solutions, and supporting metadata would be structured in ways that are useful for ML. The research notes from that visit provide a strong starting point for this work; a summary is provided in Appendix A. The TDTP data library will be seamlessly integrated with the existing TPTP problem library, TSTP solution library, and TPTP World infrastructure. An exciting possibility is to build a tool framework that allows a “plug-and-play” approach to combining ML techniques with ATP systems. This will require a decoupling of the two aspects, with a standard set of programmatic APIs that will allow each side to invoke capabilities of the other. Thus the existing structures of the TPTP World will be respected, while at the same time the needs of the ML approaches will be added. The TPTP World’s online services will be expanded to include facilities for experimenting with such combinations. The TDTP data library and associated infrastructure will be a major expansion of the TPTP World, requiring fundamental research and development.

3 New Logics and Languages

The TPTP problem library and TSTP solution library will be expanded to include problems and solutions in three new logics. These expansions are described in the following subsections.

3.1 The Extended Typed First-order Form

The TPTP World's Typed First-order Form (TFF) is a simple many-sorted logic, in which sorts are interpreted by non-empty, pairwise disjoint, domains. For ATP users, typing leads to much simpler encodings than using type predicates, and the requirement of well-typedness helps to correctly encode problems. A typed language is also necessary for correct encoding of problems with arithmetic. The existing TFF language includes features that require modernising: tuples, conditional expressions, and let expressions. In an independent development, Kotelnikov et al. introduced FOOL logic [22], which extends classical first-order logic with an interpreted boolean type, conditional expressions, and let expressions. FOOL can be straightforwardly extended with the polymorphic theory of tuples [23]. These features of TFF and FOOL can be used to concisely express problems coming from program analysis or translated from more expressive logics. The TPTP World will be expanded to include the eXtended Typed First-order form (TFX), modernising the TFF language features, and incorporating the features of FOOL [50]. This will include:

- Upgrading the TPTP4X parser to process TFX formulae (see Section 4.1).
- Collecting TFX problems for the TPTP problem library.
- Running ATP systems on the problems to collect solutions for the TSTP solution library.
- Defining additional syntactic statistics for the problems' and solutions' headers.

3.2 Non-classical Logics

Automated reasoning in non-classical logics is of increasing interest in artificial intelligence, computer science, mathematics, and philosophy. They are used for reasoning in fields such as law [37], privacy [28], and ethics [12]. The TPTP World will be expanded to include non-classical logics in Typed First-order Form (TFF) and Typed Higher-order Form (THF). A language for specifying the chosen logic and its semantics has already been developed [59][2], and ATP systems for these logics exist, e.g., [32, 15]. A tool for translating formulae from non-classical logics to the TPTP World's classical THF is available [14], which will make the problems accessible to classical ATP systems, and also provide a source of new THF problems for the TPTP problem library. This work will include:

- Upgrading the TPTP4X parser to process the non-classical logic semantic specifications and formulae (see Section 4.1).
- Collecting non-classical problems for the TPTP problem library.
- Running ATP systems on the problems to collect solutions for the TSTP solution library.
- Defining additional syntactic statistics for the problems' and solutions' headers.

3.3 Multi-valued Logics

Multi-valued logics are of interest when reasoning about scenarios in which the truth of a statement may be something less definite than the classical *true* or *false*. It is expected that these logics will, like modal and other non-classical logics, be useful for reasoning in non-STEM fields, e.g., the liveness of characters in the TV series "The Walking Dead" can be alive, not alive, but also both alive and not alive! In 1977 Nuel Belnap published two articles that addressed the needs for reasoning about such situations, summarised in [6], leading to the specification of the FDE logic [5]. FDE has four truth values *true*, *false*, *both*, and *neither*. Related logics include

[2] That proposal is being superceded by a more generic proposal that will support a large range of non-classical logics. Attend the 10th TPTP Tea Party to hear all about these exciting developments.

K3 [21], LP [38], Ł3 [29, 30], A3 [1, 53], and RM3 [34]. A tool for translating formulae from these multi-valued logics to the TPTP World's classical First-order Form (FOF) is available [51], which will make the problems accessible to classical ATP systems, and also provide a source of new FOF problems for the TPTP problem library. The TPTP World will be expanded to include multi-valued logics. This will include:

- Developing a language for specifying the chosen logic and its semantics (akin to the one developed for modal logics (see Section 3.2).
- Upgrading the TPTP4X parser to process the multi-valued logic semantic specifications, and multi-valued logic formulae (see Section 4.1).
- Extending the translation tool to read the logic and semantics specification, and reconfigure itself accordingly (currently this is somewhat manual).
- Collecting multi-valued logic problems for the TPTP problem library.
- Running ATP systems on the problems to collect solutions for the TSTP solution library.
- Defining additional syntactic statistics for the problems' and solutions' headers.

4 User Services

4.1 TPTP World Tools

The TPTP World includes a suite of service tools that are used to access, manipulate, examine, and document TPTP problems, solutions, and metadata. The tools have been written over the years in C and Perl. Two particular pieces of software that will be modernised are described here.

• The TPTP4X Parser
Many of the TPTP World tools are built around a parser library, written in C, called TPTP4X [46]. TPTP4X was originally written (in 2002!) to parse TPTP formulae in untyped first-order logic. With the addition of typed logics into the TPTP, TPTP4X has been "hacked" to accommodate these richer logics, but its data structures were not extended to explicitly track type information. TPTP4X will be upgraded to separately capture type information. This upgrade is already necessary, and will be important for its use with the new logics that will be added to the TPTP World (see Section 3).

• The TPTP2T Problem and Solution Selector
A little known tool in the TPTP World is the TPTP2T problem and solution selector[3]. TPTP2T allows the user to extract problems from the TPTP problem library, and their solutions from the TSTP solution library, according to specified problem and solution characteristics. TPTP2T offers a complex set of parameters for fine grained selection of problems and solutions. An option to export the selected problems in an archive format will be added, in order to support the linkage of the TPTP problem library with StarExec (see Section 2). TPTP2T will also be upgraded to select data from the TDTP data library.

4.2 Repository and Archiving

Currently the TPTP World is stored on a server (funded by NSF CCRI-CISE grant 1405674). A small fraction is also maintained in repositories in a Github organization called "TPTP

[3] http://www.tptp.org/cgi-bin/TPTP2T

World"[4]. This small fraction is used by a few specific research students and colleagues. The entirety of the TPTP World will be added to the Github organization, in a suite of repositories corresponding to the various data and software components. From a user perspective, this transition to repository-based archiving will provide comprehensive access to the TPTP World, including modernised web browsing of the TPTP, TSTP, and TDTP libraries.

- **Web Access to the TPTP World**

The TPTP World provides web browsing of the TPTP problem library[5] and the TSTP solution library[6]. These interfaces will be extended to the TDTP data library (subject to the following). With the migration of the hosting of the TPTP World to Github, modernised interfaces will be built using a tool such as Hugo or Jekyll. The TPTP2T tool (see Section 4.1) will be integrated to allow users to browse selected problems, solutions, and data.

5 Conclusion

This paper has described a proposed expansion and modernisation of some parts of the TPTP World: the TPTP problem library, the TSTP solution library, a new TDTP data library, new logics and languages, and improved user services. Hopefully this will attract suggestions, feedback, and offers of support to help achieve these goals. In particular, the long term future of the TPTP World needs to be discussed (this is the topic of another paper). More immediately, I will be asking the ARCADE audience questions such as:

- Can you donate problems for the TPTP problem library? What are appropriate sources for new problems?
- What statistics are useful as metadata in TPTP problems and TSTP solutions?
- What data should be in the TDTP data library, and how should it be structured?
- What non-classical logics would be of interest to you? Will multi-valued logics be useful?
- Who can help with advanced Github access, e.g., selecting problems with given characteristics?
- **When are you going to donate money to the TPTP World project?**

References

[1] A. Avron. Natural 3-Valued Logics: Characterization and Proof Theory. *Journal of Symbolic Logic*, 56(1):276–294, 1991.

[2] K. Bansal, S. Loos, M. Rabe, and C. Szegedy. Learning to Reason in Large Theories Without Imitation. arXiv:1905.10501, 2019.

[3] K. Bansal, S. Loos, C. Szegedy, and S. Wilcox. HOList: An Environment for Machine Learning of Higher-order Theorem Proving. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, pages 454–463, 2019.

[4] C. Barrett, A. Stump, and C. Tinelli. The SMT-LIB Standard: Version 2.0. In A. Gupta and D. Kroening, editors, *Proceedings of the 8th International Workshop on Satisfiability Modulo Theories*, 2010.

[5] N. Belnap. A Useful Four-Valued Logic. In J.M. Dunn and G. Epstein, editors, *Modern Uses of Multiple-Valued Logic*, pages 5–37. Springer Netherlands, 1977.

[4] https://github.com/TPTPWorld

[5] http://www.tptp.org/cgi-bin/SeeTPTP?Category=Problems

[6] http://www.tptp.org/cgi-bin/SeeTPTP?Category=Solutions

[6] N. Belnap. A Useful Four-valued Logic: How a Computer Should Think. In A.R. Anderson, N.D. Belnap, and J.M. Dunn, editors, *Entailment: The Logic of Relevance and Necessity, Volume II*, pages 506–541. Princeton UP, 1992.

[7] J. Blanchette, M. Haslbeck, D. Matichuk, and T. Nipkow. Mining the Archive of Formal Proofs. In M. Kerber, J. Carette, C. Kaliszyk, F. Rabe, and V. Sorge, editors, *Proceedings of the 8th Conference on Intelligent Computer Mathematics*, number 9150 in Lecture Notes in Computer Science, pages 3–17, 2015.

[8] J. Blanchette, N. Peltier, and S. Robillard. Superposition with Datatypes and Codatatypes. In D. Galmiche, S. Schulz, and R. Sebastiani, editors, *Proceedings of the 9th International Joint Conference on Automated Reasoning*, number 10900 in Lecture Notes in Computer Science, pages 370–387, 2018.

[9] C. Brown, T. Gauthier, C. Kaliszyk, G. Sutcliffe, and J. Urban. GRUNGE: A Grand Unified ATP Challenge. In P. Fontaine, editor, *Proceedings of the 27th International Conference on Automated Deduction*, number 11716 in Lecture Notes in Computer Science, pages 123–141. Springer-Verlag, 2019.

[10] V. Chaudri, B. Cheng, A. Overholtzer, J. Roschelle, A. Spaulding, P. Clark, M. Greaves, and D. Gunning. Inquire Biology: A Textbook that Answers Questions. *AI Magazine*, 34(3), 2013.

[11] K. Chvalovsky, J. Jakubuv, M. Suda, and J. Urban. ENIGMA-NG: Efficient Neural and Gradient-Boosted Inference Guidance for E. In P. Fontaine, editor, *Proceedings of the 27th International Conference on Automated Deduction*, number 11716 in Lecture Notes in Computer Science, pages 197–215. Springer-Verlag, 2019.

[12] L. Dennis, M. Fisher, M. Slavkovik, and M. Webster. Formal Verification of Ethical Choices in Autonomous Systems. *Robotics and Autonomous Systems*, 77:1–14, 2016.

[13] H. Dong, J. Mao, T. Lin, C. Wang, L. Li, and D. Zhou. Neural Logic Machines. In A. Rush, S. Levine, K. Livescu, and S. Mohamed, editors, *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[14] T. Gleissner and A. Steen. The MET: The Art of Flexible Reasoning with Modalities. In C. Benzmüller, F. Ricca, X. Parent, and D. Roman, editors, *Proceedings of the 2nd International Joint Conference on Rules and Reasoning*, number 11092 in Lecture Notes in Computer Science, pages 274–284, 2018.

[15] T. Gleissner, A. Steen, and C. Benzmüller. Theorem Provers for Every Normal Modal Logic. In T. Eiter and D. Sands, editors, *Proceedings of the 21st International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, number 46 in EPiC Series in Computing, pages 14–30. EasyChair Publications, 2017.

[16] D. Huang, P. Dhariwal, D. Song, and I. Sutskever. GamePad: A Learning Environment for Theorem Proving. In *Proceedings of the 7th International Conference on Learning Representations*, pages 454–463, 2019.

[17] J. Jakubuv, K. Chvalovský, M. Olsák, B. Piotrowski, M. Suda, and J. Urban. ENIGMA Anonymous: Symbol-Independent Inference Guiding Machine (System Description). In N. Peltier and V. Sofronie-Stokkermans, editors, *Proceedings of the 10th International Joint Conference on Automated Reasoning*, number 12167 in Lecture Notes in Artificial Intelligence, pages 448–463, 2020.

[18] J. Jakubuv and J. Urban. ENIGMA: Efficient Learning-based Inference Guiding Machine. In H. Geuvers, M. England, O. Hasan, F. Rabe, and O. Teschke, editors, *Proceedings of the 10th International Conference on Intelligent Computer Mathematics*, number 10383 in Lecture Notes in Artificial Intelligence, pages 292–302. Springer-Verlag, 2017.

[19] C. Kaliszyk, F. Chollet, and C. Szegedy. HolStep: A Machine Learning Dataset for Higher-order Logic Theorem Proving. In I. Murray, M. Ranzato, and O. Vinyals, editors, *Proceedings of the 6th International Conference on Learning Representations*, page To appear, 2018.

[20] C. Kaliszyk and J. Urban. Learning-assisted Automated Reasoning with Flyspeck. *Journal of Automated Reasoning*, 53(2):173–213, 2014.

[21] S. Kleene. *Introduction to Metamathematics*. North-Holland, 1952.

[22] E. Kotelnikov, L. Kovacs, and A. Voronkov. A First Class Boolean Sort in First-Order Theorem Proving and TPTP. In M. Kerber, J. Carette, C. Kaliszyk, F. Rabe, and V. Sorge, editors, *Proceedings of the International Conference on Intelligent Computer Mathematics*, number 9150 in Lecture Notes in Computer Science, pages 71–86. Springer-Verlag, 2015.

[23] E. Kotelnikov, L. Kovacs, and A. Voronkov. A FOOLish Encoding of the Next State Relations of Imperative Programs. In D. Galmiche, S. Schulz, and R. Sebastiani, editors, *Proceedings of the 9th International Joint Conference on Automated Reasoning*, number 10900 in Lecture Notes in Computer Science, pages 405–421, 2018.

[24] D. Külwein and J. Blanchette. A Survey of Axiom Selection as a Machine Learning Problem. In S. Geschke, editor, *Computability and Metamathematics : Festschrift Celebrating the 60th birthdays of Peter Koepke and Philip Welch*, pages 1–15. College Publications, 2014.

[25] D. Külwein and J. Urban. MaLeS: A Framework for Automatic Tuning of Automated Theorem Provers. *Journal of Automated Reasoning*, page To appear, 2015.

[26] R. Kumar, M. Myreen, M. Norrish, and S. Owens. CakeML: A Verified Implementation of ML. In P Sewell, editor, *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 179–191. ACM Press, 2014.

[27] X. Leroy. Formal Verification of a Realistic Compiler. *Communications of the ACM*, 52(7):107–115, 2009.

[28] T. Libal. Towards Automated GDPR Compliance Checking. In F. Heintz, editor, *Proceedings of TAILOR 2020 - Workshop on the Scientific Foundations of Trustworthy AI, Integrating Learning, Optimisation, and Reasoning*, Lecture Notes in Computer Science, page To appear, 2020.

[29] J. Łukasiewicz. On Three-Valued Logic. *Ruch Filozoficny*, 5:170–171, 1920.

[30] J. Łukasiewicz and A. Tarski. Investigations into the Sentential Calculus. *Comptes Rendus des seances de la Societe des Sciences et des Lettres de Varsovie*, 23:30–50, 1930.

[31] T. Nipkow. Teaching Algorithms and Data Structures with a Proof Assistant. In A. Hritcu and A. Popescu, editors, *Proceedings of 10th ACM SIGPLAN International Conference on Certified Programs and Proofs*, number 5195 in Lecture Notes in Artificial Intelligence, pages 18–33. Springer-Verlag, 2021.

[32] J. Otten. MleanCoP: A Connection Prover for First-Order Modal Logic. In S. Demri, D. Kapur, and C. Weidenbach, editors, *Proceedings of the 7th International Joint Conference on Automated Reasoning*, number 8562 in Lecture Notes in Artificial Intelligence, pages 269–276, 2014.

[33] A. Pease, G. Sutcliffe, N. Siegel, and S. Trac. Large Theory Reasoning with SUMO at CASC. *AI Communications*, 23(2-3):137–144, 2010.

[34] F.J. Pelletier, G. Sutcliffe, and A.P. Hazen. Automated Reasoning for the Dialetheic Logic RM3. In V. Rus and Z. Markov, editors, *Proceedings of the 30th International FLAIRS Conference*, pages 110–115, 2017.

[35] B. Piotrowski and J. Urban. ATPboost: Learning Premise Selection in Binary Setting with ATP Feedback. In D. Galmiche, S. Schulz, and R. Sebastiani, editors, *Proceedings of the 9th International Joint Conference on Automated Reasoning*, number 10900 in Lecture Notes in Computer Science, pages 566–574, 2018.

[36] B. Piotrowski and J. Urban. Stateful Premise Selection by Recurrent Neural Networks. In E. Alpert and L. Kovacs, editors, *Proceedings of the 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, number 73 in EPiC Series in Computing, pages 409–422, 2020.

[37] H. Prakken and G. Sartor. Law and Logic: A Review from an Argumentation Perspective. *Artificial Intelligence*, 227:214–245, 2015.

[38] G. Priest. The Logic of Paradox. *Journal of Philosophical Logic*, 8:219–241, 1979.

[39] D. Ramachandran, Reagan P., and K. Goolsbey. First-orderized ResearchCyc: Expressiveness and

Efficiency in a Common Sense Knowledge Base. In Shvaiko P., editor, *Proceedings of the Workshop on Contexts and Ontologies: Theory, Practice and Applications*, 2005.

[40] M. Rawson and G. Reger. lazyCoP 0.1. EasyChair Preprints 3926, 2020.

[41] R. Riegel, A. Gray, F. Luus, N. Khan, N. Makondo, I. Akhalwaya, H. Qian, R. Fagin, F. Barahona, U. Sharma, S. Ikbal, H. Karanam, S. Neelam, A. Likhyani, and S. Srivastava. Logical Neural Networks. arXiv:2006.13155, 2020.

[42] S. Shi, H. Chen, M. Zhang, and Y. Zhang. Neural Logic Networks. arXiv:1910.08629, 2019.

[43] A. Stump, G. Sutcliffe, and C. Tinelli. StarExec: a Cross-Community Infrastructure for Logic Solving. In S. Demri, D. Kapur, and C. Weidenbach, editors, *Proceedings of the 7th International Joint Conference on Automated Reasoning*, number 8562 in Lecture Notes in Artificial Intelligence, pages 367–373, 2014.

[44] M. Suda, G. Sutcliffe, P. Wischnewski, M. Lamotte-Schubert, and G. de Melo. External Sources of Axioms in Automated Theorem Proving. In B. Mertsching, editor, *Proceedings of the 32nd Annual Conference on Artificial Intelligence*, number 5803 in Lecture Notes in Artificial Intelligence, pages 281–288, 2009.

[45] G. Sutcliffe. The SZS Ontologies for Automated Reasoning Software. In G. Sutcliffe, P. Rudnicki, R. Schmidt, B. Konev, and S. Schulz, editors, *Proceedings of the LPAR Workshops: Knowledge Exchange: Automated Provers and Proof Assistants, and The 7th International Workshop on the Implementation of Logics*, number 418 in CEUR Workshop Proceedings, pages 38–49, 2008.

[46] G. Sutcliffe. The TPTP World - Infrastructure for Automated Reasoning. In E. Clarke and A. Voronkov, editors, *Proceedings of the 16th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, number 6355 in Lecture Notes in Artificial Intelligence, pages 1–12. Springer-Verlag, 2010.

[47] G. Sutcliffe. The CADE ATP System Competition - CASC. *AI Magazine*, 37(2):99–101, 2016.

[48] G. Sutcliffe. The TPTP Problem Library and Associated Infrastructure. From CNF to TH0, TPTP v6.4.0. *Journal of Automated Reasoning*, 59(4):483–502, 2017.

[49] G. Sutcliffe. The 10th IJCAR Automated Theorem Proving System Competition - CASC-J10. *AI Communications*, page To appear, 2021.

[50] G. Sutcliffe and E. Kotelnikov. TFX: The TPTP Extended Typed First-order Form. In B. Konev, J. Urban, and S. Schulz, editors, *Proceedings of the 6th Workshop on Practical Aspects of Automated Reasoning*, number 2162 in CEUR Workshop Proceedings, pages 72–87, 2018.

[51] G. Sutcliffe and J. Pelletier. JGXYZ - An ATP System for Gap and Glut Logics. In P. Fontaine, editor, *Proceedings of the 27th International Conference on Automated Deduction*, number 11716 in Lecture Notes in Computer Science, pages 526–537. Springer-Verlag, 2019.

[52] G. Sutcliffe, M. Suda, A. Teyssandier, N. Dellis, and G. de Melo. Progress Towards Effective Automated Reasoning with World Knowledge. In C. Murray and H. Guesgen, editors, *Proceedings of the 23rd International FLAIRS Conference*, pages 110–115. AAAI Press, 2010.

[53] A. Tedder. Axioms for Finite Collapse Models of Arithmetic. *Review of Symbolic Logic*, 8:529–539, 2015.

[54] J. Urban. MPTP 0.2: Design, Implementation, and Initial Experiments. *Journal of Automated Reasoning*, 37(1-2):21–43, 2006.

[55] J. Urban. BliStr: The Blind Strategymaker. In S. Autexier, editor, *Proceedings of the 1st Global Conference on Artificial Intelligence*, number 36 in EPiC Series in Computing, pages 312–319. EasyChair Publications, 2015.

[56] J. Urban, G. Sutcliffe, P. Pudlak, and J. Vyskocil. MaLARea SG1: Machine Learner for Automated Reasoning with Semantic Guidance. In P. Baumgartner, A. Armando, and G. Dowek, editors, *Proceedings of the 4th International Joint Conference on Automated Reasoning*, number 5195 in Lecture Notes in Artificial Intelligence, pages 441–456. Springer-Verlag, 2008.

[57] M. Wang, Y. Tang, J. Wang, and J. Deng. Premise Selection for Theorem Proving by Deep

Graph Embedding. In L. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Proceedings of the 30th International Annual Conference on Neural Information Processing Systems*, pages 2786–2796, 2017.

[58] C. Weidenbach, A. Fietzke, R. Kumar, M. Suda, P. Wischnewski, and D. Dimova. SPASS Version 3.5. In R. Schmidt, editor, *Proceedings of the 22nd International Conference on Automated Deduction*, number 5663 in Lecture Notes in Artificial Intelligence, pages 140–145. Springer-Verlag, 2009.

[59] M. Wisniewski, A. Steen, and C. Benzmüller. TPTP and Beyond: Representation of Quantified Non-Classical Logics. In C. Benzmüller and J. Otten, editors, *Proceedings of the 2nd International Workshop on Automated Reasoning in Quantified Non-Classical Logics*, number 1770 in CEUR Workshop Proceedings, pages 51–65, 2016.

A Possible Details of the TDTP Data library

The TDTP will be based around problem and solution corpora, e.g., the Mizar 2078 corpus [54], the cakeML corpus [26], the HOL4 corpus [9], the Archive of Formal Proofs [7]. Data suitable for theorem proving, machine learning (ML), and the application of each to the other (but mainly of ML to ATP), will be collected and curated, as follows.

In the TPTP view the problems and solutions in the TDTP will be structured as in the TPTP problem library and the TSTP solution library, and will be accessed in the same way. The top level of the TDTP directory hierarchy will include `Axioms`, `Problems`, and `Solutions` directories. Within each of the `Problems` and `Solutions` directories there will be a three letter acronym (TLA) directory for each corpus. Within each of the corpus directories there will be the problems to be solved (as in the TPTP problem library). The `Axioms` directory will contain axiom files that are included in problems (as in the TPTP problem library). Within each of the `Solutions` corpus directories there will be subdirectories named after the corpus' problems, each of which will contain contain solutions to the corresponding problem (as in the TSTP solution library). The directories in the `Problems` and `Solutions` directories, and the files in the `Axioms` directory, will be symlinked into the existing TPTP `Problems`, `Solutions`, and `Axioms` directories, i.e., they will be accessible using the existing TPTP infrastructure by non-ML based ATP. An important interplay between problems and solutions arises in the context of axiom selection, in that a problem might have very many axioms, and a solution to the problem provides a subset adequate for finding a solution. This provides the data needed for machine learning of axiom selection.

In the ML view the problems, solutions, and supporting AI data will be structured in ways that are useful for ML. The top level of the TDTP directory hierarchy will include `AIData` and `Corpus` directories. Within the `AIData` directory there will be domain directories. The domain directories will contain files useful for ML:

- `.features` files that contain features of the problems that can be used in the ML processes, e.g., the symbols used in each formula of a corpus.
- `.names` files that list all the formula names in a corpus. The lines in these files will provide links from formula names to (their occurrences in) problems in the corpus.
- `.order` files that contain orderings of the formulae, e.g., the order in which the conjecture formulae were naturally exported from their original form to a logical form.
- `.dependencies` files that list the names of axioms that can be used to prove each conjecture (in the problems) in the corpus. The dependencies files capture the subsets adequate for finding a solution, and include minimally the subsets corresponding to solutions in the

TDTP.

The `Corpus` directory will provide a corpus-centric view of the data. There will be a corpus directory for each corpus in the TDTP. Within each corpus directory there will be symlinks to its `Axioms` directory, `Problems/TLA` directory, `Solutions/TLA` directory, and `AIData/TLA` directory.