Stepping Stones in the TPTP World

Geoff Sutcliffe

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

Abstract

The first release of the TPTP problem library was made on Friday 12th November 1993. Since then the TPTP World (once gently referred to as the "TPTP Jungle") has evolved into a well established infrastructure that supports research, development, and deployment of ATP systems. There have been some key developments that helped make the TPTP World a success: the first TPTP problem library that was first released in 1993, the CADE ATP System Competition (CASC) that was conceived after CADE-12 in Nancy in 1994, the problem difficulty ratings that were added in 1997, the current TPTP language that was adopted in 2003, the SZS ontologies that were specified in 2004, the TSTP solution library that was built starting around 2005, the Specialist Problem Classes (SPCs) used to classify problems from 2010, the SystemOnTPTP service that was offered from 2011, and the StarExec service that started in 2013. This talk reviews these stepping stones in the development of the TPTP World.

1 Introduction

The TPTP World [11, 13] is a well established infrastructure that supports research, development, and deployment of ATP systems. Salient components of the TPTP World are the TPTP problem library [10], the TSTP solution library [11], the TPTP languages [18], the SZS ontologies [9], the Specialist Problem Classes (SPCs) and problem difficulty ratings [?], and the CADE ATP System Competition (CASC) [12]. StarExec [?] and SystemOnTPTP [?] provide computational support for the TPTP World. There are dependencies between these parts of the TPTP World, as shown in Figure 1, forming a series of "stepping stones" from key starting points through to happy users.
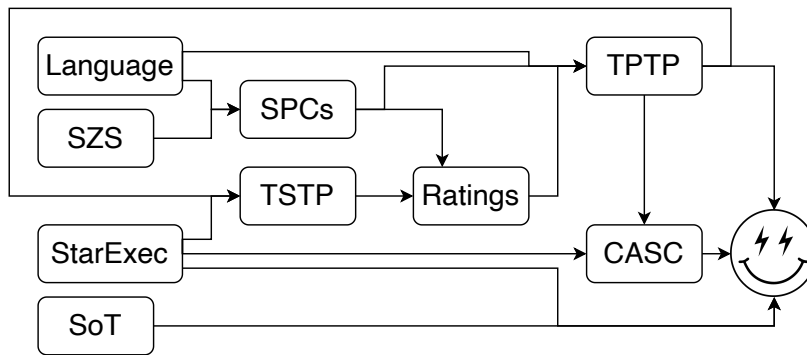

Figure 1: Dependencies between the Stepping Stones

Various parts of the TPTP World have been deployed in a range of applications, in both academia and industry. Since the first release of the TPTP problem library in 1993, many researchers have used the TPTP World as an appropriate and convenient basis for ATP system

research and development. Over the years the TPTP World has provided a platform upon which ATP users have presented their needs to ATP system developers, who have then adapted their ATP systems to the users' needs. The web page https://www.tptp.org provides access to all components.

This paper is organized as follows:

2   The TPTP Languages

The TPTP language [14] is one of the keys to the success of the TPTP World. The language is used for writing both problems and solutions, which enables convenient communication between systems. Originally the TPTP World supported only first-order clause normal form (CNF) [19]. Over the years full first-order form (FOF) [10], typed first-order form (TFF) [17, 4], typed extended first-order form (TXF) [16], typed higher-order form (THF) [15, 5], and non-classical forms (NTF) [1] [8] have been added. A general principle of the TPTP language is "we provide the syntax, you provide the semantics". As such, there is no a priori commitment to any semantics for the languages, although in almost all cases the intended logic and semantics are well known. All the typed forms include constructs for arithmetic. TF0 [17], the monomorphic subform of TFF, is used in this work (see Section **??**).

The top level building blocks of the TPTP language are *annotated formulae*. An annotated formula has the form:

language(*name*, *role*, *formula*, *source*, *useful_info*)

The *language*s supported are `cnf` (clause normal form), `fof` (first-order form), `tff` (typed first-order form), and `thf` (typed higher-order form). The *role*, e.g., `axiom`, `lemma`, `conjecture`, defines the use of the formula in an ATP system. In a *formula*, terms and atoms follow Prolog conventions – functions and predicates start with a lowercase letter or are 'single quoted', and variables start with an uppercase letter. The language also supports interpreted symbols, which either start with a `$`, e.g., the truth constants `$true` and `$false`, or are composed of non-alphabetic characters, e.g., integer/rational/real numbers such as 27, 43/92, -99.66. The logical connectives in the TPTP language are `!`, `?`, `~`, `|`, `&`, `=>`, `<=`, `<=>`, and `<~>`, for the mathematical connectives $\forall$, $\exists$, $\neg$, $\vee$, $\wedge$, $\Rightarrow$, $\Leftarrow$, $\Leftrightarrow$, and $\oplus$ respectively. Equality and inequality are expressed as the infix operators `=` and `!=`. The *source* and *useful_info* are optional.

3   Conclusion

This paper
Currently this work is being extended to

References

[1] A.R. Anderson and N.D. Belnap. *Entailment: The Logic of Relevance and Necessity, Vol. 1.* Princton University Press, 1975.

[2] L. Augusto. *Many-valued Logics: A Mathematical and Computational Introduction.* College Publications, 2017.

---

[1]There are many "non-classical logics", including multi-valued logics [2], paraconsistent logics [7], relevance logics [1], etc. In this work we are interested in those that admit Kripke interpretation [6], e.g., modal logics [3].

[3] P. Blackburn, J. van Benthem, and F. Wolther. *Handbook of Modal Logic.* Number 3 in Studies in Logic and Practical Reasoning. Elsevier Science, 2006.

[4] J. Blanchette and A. Paskevich. TFF1: The TPTP Typed First-order Form with Rank-1 Polymorphism. In M.P. Bonacina, editor, *Proceedings of the 24th International Conference on Automated Deduction*, number 7898 in Lecture Notes in Artificial Intelligence, pages 414–420. Springer-Verlag, 2013.

[5] C. Kaliszyk, G. Sutcliffe, and F. Rabe. TH1: The TPTP Typed Higher-Order Form with Rank-1 Polymorphism. In P. Fontaine, S. Schulz, and J. Urban, editors, *Proceedings of the 5th Workshop on Practical Aspects of Automated Reasoning*, number 1635 in CEUR Workshop Proceedings, pages 41–55, 2016.

[6] S. Kripke. Semantical Considerations on Modal Logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[7] G. Priest. Paraconsistent Logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 6, pages 287–393. Springer-Verlag, 2002.

[8] A. Steen, D. Fuenmayor, T. Gleißner, G. Sutcliffe, and C. Benzmüller. Automated Reasoning in Non-classical Logics in the TPTP World. In B. Konev, C. Schon, and A. Steen, editors, *Proceedings of the 8th Workshop on Practical Aspects of Automated Reasoning*, number 3201 in CEUR Workshop Proceedings, page Online, 2022.

[9] 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.

[10] G. Sutcliffe. The TPTP Problem Library and Associated Infrastructure. The FOF and CNF Parts, v3.5.0. *Journal of Automated Reasoning*, 43(4):337–362, 2009.

[11] 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.

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

[13] 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.

[14] G. Sutcliffe. The Logic Languages of the TPTP World. *Logic Journal of the IGPL*, 31(6):1153–1169, 2023.

[15] G. Sutcliffe and C. Benzmüller. Automated Reasoning in Higher-Order Logic using the TPTP THF Infrastructure. *Journal of Formalized Reasoning*, 3(1):1–27, 2010.

[16] 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.

[17] G. Sutcliffe, S. Schulz, K. Claessen, and P. Baumgartner. The TPTP Typed First-order Form with Arithmetic. In N. Bjørner and A. Voronkov, editors, *Proceedings of the 18th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, number 7180 in Lecture Notes in Artificial Intelligence, pages 406–419. Springer-Verlag, 2012.

[18] G. Sutcliffe, S. Schulz, K. Claessen, and A. Van Gelder. Using the TPTP Language for Writing Derivations and Finite Interpretations. In U. Furbach and N. Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning*, number 4130 in Lecture Notes in Artificial Intelligence, pages 67–81. Springer, 2006.

[19] G. Sutcliffe and C.B. Suttner. The TPTP Problem Library: CNF Release v1.2.1. *Journal of Automated Reasoning*, 21(2):177–203, 1998.