

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 [13, 15] 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 [12], the TSTP solution library [13], the TPTP languages [20], the SZS ontologies [11], the Specialist Problem Classes (SPCs) and problem difficulty ratings [22], and the CADE ATP System Competition (CASC) [14]. StarExec [9] and SystemOnTPTP [10] 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, who contribute to TPTP problem library. There is another cycle of dependencies: from the TPTP problem library, to the TSTP solution library, to the problem difficulty ratings, and back to the TPTP problem library. This cycle means that building these components, releases of the TPTP problem library in particular, requires iteration until stability is reached.

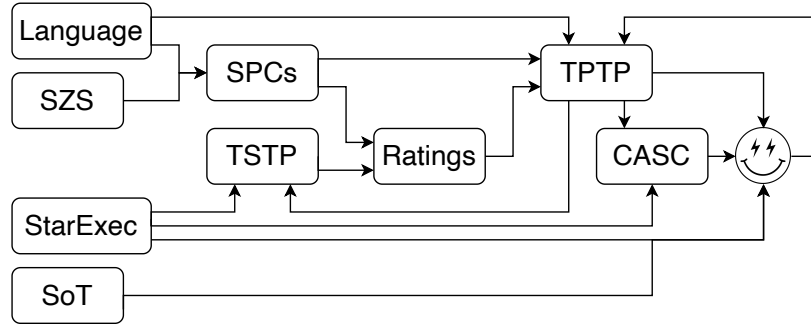


Figure 1: Dependencies between the Stepping Stones

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 [16] 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) [21]. Over the years full first-order form (FOF) [12], typed first-order form (TFF) [19, 4], typed extended first-order form (TXF) [18], typed higher-order form (THF) [17, 5], and non-classical forms (NTF) [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.

The formulae of problems solutions are built from *annotated formulae*, which have the form ...

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

The *languages* 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. Figure 2 shows an example with typed higher-order formulae.

3 The TPTP Problem Library

The development of the TPTP World started with the TPTP problem library in mid-1992, as a collaboration between Geoff Sutcliffe at the University of Western Australia (James Cook University from 1993), and Christian Suttner at the Technische Universität München. The TPTP problem library is managed in the manner of a software product, in the sense that fixed releases are made. Each release is identified by a release number in the form *vVersion.Edition.Patch*: the *Version* enumerates major new releases of the TPTP in which important new features have been added, the *Edition* is incremented each time new problems are added to the current version, and The *Patch* level is incremented each time errors, found in the current edition, are corrected. The first public release, TPTP v1.0.0, was made on 12th November 1993.

```

%-----
thf(beverage_decl,type,    beverage: $tType ).
thf(syrup_decl,type,      syrup: $tType ).
thf(coffee_type,type,     coffee: beverage ).
thf(mix_type,type,        mix: beverage > syrup > beverage ).
thf(heat_type,type,       heat: beverage > beverage ).
thf(heated_mix_type,type, heated_mix: beverage > syrup > beverage ).
thf(hot_type,type,        hot: beverage > $o ).

thf(heated_mix,axiom,
  ( heated_mix
    = ( ~ [B: beverage,S: syrup] : ( heat @ ( mix @ B @ S ) ) ) ).

thf(hot_mixture,axiom,
  ! [B: beverage,S: syrup] : ( hot @ ( heated_mix @ B @ S ) ) ).

thf(heated_coffee_mix,axiom,
  ! [S: syrup] : ( ( heated_mix @ coffee @ S ) = coffee ) ).

thf(hot_coffee,conjecture,
  ? [Mixture: syrup > beverage] :
    ~ ? [S: syrup] :
      ( ( ( Mixture @ S ) = coffee )
        & ( hot @ ( Mixture @ S ) ) ) ).
%-----

```

Figure 2: THF annotated formulae

The problems in the TPTP are classified into *domains* that reflect the natural hierarchy of scientific domains. Seven main fields are defined: logic, mathematics, computer science, science & engineering, social sciences, arts & humanities, and other. Each field is subdivided into domains, each identified by a three-letter mnemonic, e.g., the social science field has three domains: Social Choice Theory (SCT), Management (NGT), and Geography (GEG).

Table 1 lists the versions of the TPTP up to v9.0.0, with the new feature added, the number of problem domains, and the number of problems.¹ The number of domains indicates the semantic diversity of the TPTP problems, while the number of problems indicates the size of the TPTP problem library. The attentive reader might note that many releases have been made in July/August. This is because the CADE ATP System Competition (CASC - see Section 9), has an influence on the release cycle of the TPTP.

Release	Date	Changes	Domains	Problems
v1.0.0	12/11/93	First public release, only CNF [21]	23	2295
v2.0.0	05/06/97	FOF [12] and ratings (Section 7)	28	3277
v3.0.0	11/11/04	New TPTP language [20]	32	7267
v4.0.0	04/07/09	TH0 (monomorphic typed higher-order) [17]	41	16512
v5.0.0	16/09/10	TF0 (monomorphic typed first-order) [19]	45	18480
v6.0.0	21/09/13	TF1 (polymorphic typed first-order) [4]	48	20306
v7.0.0	24/07/17	TH1 (polymorphic typed higher-order) [5]	53	21851
v8.0.0	19/04/22	TXF (typed extended first-order) [18]	54	24785
v9.0.0	??/07/24	NTF (non-classical typed first-order) [8]	55	25598

Table 1: Overview of TPTP releases

¹The data for v9.0.0 is an estimate, because this paper was written before the release was finalised.

TPTP problem files present the logical formulae in a format that is both human and machine readable, and additionally provide useful information for users. The file names are built from the domain acronym, a 3 digit problem number, a separator that indicates the syntax (- for CNF, + for FOF, _ for TFF, ^ for THF), optional digits for the problem size, and a problem version number. Each file has three sections: a header, optional includes, and the formulae.

The header section contains information for users, formatted as comments in four parts: the first part identifies and describes the problem; the second part provides information about occurrences of the problem in the literature and elsewhere; the third part provides semantic and syntactic characteristics of the problem; the last part contains comments and bugfix information. The include section is optional, and if used contains `include` directives for axiom files, which in turn have the same three-part format as problem files. Their inclusion avoids the need for duplication of the formulae in commonly used axiomatizations. The formula section contains annotated formulae, as described in Section 2. Figure 3 shows an example header and `include` section. The header fields are self-explanatory, but of particular interest are the **Status** field that is explained in Section 5, the **Rating** field that is explained in Section 7, and the **SPC** field that is explained in Section 6.

```
%-----
% File      : DAT016_1 : TPTP v8.2.0. Bugfixed v5.1.0.
% Domain    : Data Structures
% Problem    : Some element is 53
% Version    : [PW06] axioms.
% English    : Show that some element of the array has the value 53.

% Refs      : [PW06] Prevosto & Waldmann (2006), SPASS+T
%            : [Wal10] Waldmann (2010), Email to Geoff Sutcliffe
% Source     : [Wal10]
% Names      : (40) [PW06]

% Status     : Theorem
% Rating     : 0.25 v8.2.0, 0.12 v7.5.0, 0.30 v7.4.0, 0.12 v7.3.0, etc.
% Syntax     : Number of formulae      : 6 ( 1 unt; 3 typ; 0 def)
%              Number of atoms         : 12 ( 5 equ)
%              Maximal formula atoms    : 4 ( 2 avg)
%              Number of connectives    : 4 ( 0 ~; 1 |; 1 &)
%              ( 0 <=>; 2 =>; 0 <; 0 <~>)
%              Maximal formula depth    : 6 ( 5 avg)
%              Maximal term depth       : 3 ( 1 avg)
%              Number of FOOLs          : 5 ( 5 fml; 0 var)
%              Number arithmetic        : 16 ( 2 atm; 2 fun; 5 num; 7 var)
%              Number of types          : 2 ( 1 usr; 1 ari)
%              Number of type conns     : 5 ( 2 >; 3 *; 0 +; 0 <>)
%              Number of predicates     : 3 ( 1 usr; 0 prp; 2-2 aty)
%              Number of functors       : 9 ( 2 usr; 5 con; 0-3 aty)
%              Number of variables      : 10 ( 9 !; 1 ?; 10 :)
% SPC        : TFO_THM_EQU_ARI

% Comments   : The array contains integers.
% Bugfixes   : v5.1.0 - Fixed conjecture
%-----
%----Includes axioms for arrays
include('Axioms/DAT001_0.ax').
%-----
```

Figure 3: Header of problem DAT016_1.

This section has naturally focused on the successful parts of the TPTP history. There have also been some failed developments and suboptimal (in retrospect) decisions ☹. For example,

in 2015 there was an attempt to develop a description logic form for the TPTP language. While some initial progress was made, it ground to a halt without support from the description logic community. A suboptimal design decision, rooted in the early days of the TPTP, is the naming scheme used for problem files. The naming scheme uses three digits to number the problems in each domain, thus setting a limit of 1000 problems, which failed to anticipate the numbers of problems that would be contributed to some of the problem domains. This has been overcome by creating multiple domain directories where necessary, but if it were to be done again, six or eight digit problem numbers shared across all domains would be an improvement.

4	The TSTP Solution Library
5	The SZS Ontologies
6	Specialist Problem Classes
7	Problem Difficulty Ratings
8	StarExec and SystemOnTPTP
9	The CADE ATP System Competition
10	TPTP World Users
11	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*. Princeton University Press, 1975.
- [2] L. Augusto. *Many-valued Logics: A Mathematical and Computational Introduction*. College Publications, 2017.
- [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. Guenther, editors, *Handbook of Philosophical Logic*, volume 6, pages 287–393. Springer-Verlag, 2002.
- [8] A. Steen, D. Fuenmayor, T. Gleißner, G. Sutcliffe, and C. Benz Müller. 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] 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.
- [10] G. Sutcliffe. SystemOnTPTP. In D. McAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction*, number 1831 in Lecture Notes in Artificial Intelligence, pages 406–410. Springer-Verlag, 2000.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] G. Sutcliffe. The CADE ATP System Competition - CASC. *AI Magazine*, 37(2):99–101, 2016.
- [15] 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.
- [16] G. Sutcliffe. The Logic Languages of the TPTP World. *Logic Journal of the IGPL*, 31(6):1153–1169, 2023.
- [17] G. Sutcliffe and C. Benz Müller. Automated Reasoning in Higher-Order Logic using the TPTP THF Infrastructure. *Journal of Formalized Reasoning*, 3(1):1–27, 2010.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] G. Sutcliffe and C.B. Suttner. The TPTP Problem Library: CNF Release v1.2.1. *Journal of Automated Reasoning*, 21(2):177–203, 1998.
- [22] G. Sutcliffe and C.B. Suttner. Evaluating General Purpose Automated Theorem Proving Systems. *Artificial Intelligence*, 131(1-2):39–54, 2001.