Different Proofs are Good Proofs

Geoff Sutcliffe
University of Miami, USA

Cynthia Chang
Rensselaer Polytechnic Institute, USA

Li Ding, Deborah McGuinness
Rensselaer Polytechnic Institute, USA

Paulo Pinheiro da Silva
University of Texas at El Paso, USA

Abstract

In order to compare the quality of proofs, it is necessary to measure artifacts of the proofs, and evaluate the measurements to determine differences between the proofs. This paper discounts the approach of *ranking* proofs by their measured proof artifacts, and takes the position that *different proofs are good proofs*. The position is based on proofs in the TSTP solution library, which are generated by Automated Theorem Proving (ATP) systems applied to first-order logic problems in the TPTP problem library.

1 Introduction

In order to compare the quality of proofs, it is necessary to measure artifacts of the proofs, and evaluate the measurements to determine differences between the proofs. For example, a commonly measured artifact is the number of inference steps in a proof, and a smaller number of steps might be considered better. It is difficult to take comparable measurements of artifacts in proofs that are expressed in different languages, are generated by different deduction processes, use different inference rules, etc., Thus, for meaningful measurement and evaluation, it is useful if the proofs are reasonably homogeneous, at least with respect to the artifacts being measured. Given such proofs, this paper discounts the approach of *ranking* proofs by their measured proof artifacts, and takes the position that *different proofs are good proofs*. Section 2 motivates and explains this position.

The position taken in this paper is based on proofs in the TSTP solution library [18]. The TSTP contains solutions for problems in the TPTP problem library [21], generated by Automated Theorem Proving (ATP) systems. The TPTP problems used for this work are all theorems expressed in the first-order form (FOF) fragment of the TPTP language [23], and consist of axioms and a conjecture to be proved. The TSTP proofs used for this work all come from ATP systems that convert the FOF problem formulae to an equisatisfiable clause normal form (CNF), and then apply sound inferences to derive a contradiction (a `$false` clause), i.e., the proofs are by CNF refutation. Thus the proofs consist of formulae in FOF and CNF.

For this work the TSTP proofs were converted [10] to the Proof Markup Language (PML) [9]. PML provides a slightly richer framework for representing proofs, including explicit structures for recording provenance information, and for representing alternative derivations of a conclusion. While neither of these capabilities are used directly in this work, the latter is important for subsequent work on combining proofs, as noted in Section 4. The process of parsing and translating from TPTP to PML has also been useful in exposing "flaws" in the TSTP proofs, in terms of syntax, structure (e.g., incomplete proof output), and semantics (e.g., misuse of SZS status values [20]). This has led to bugfixes in the ATP systems, and regeneration of proofs in the TSTP. A proof is of very low quality if it is not well-formed!

Deborah McGuinness, Aaron Stump, Geoff Sutcliffe, Cesare Tinelli (eds.); EMSQMS 2010, pp. 1-10

2 Difference is Quality

The building blocks of TSTP proofs are formulae, and inferences that infer a conclusion formula from parent formulae. In order to take comparable measurements of these proofs artifacts, it is necessary to recognize equivalent formulae and equivalent inference steps. The proofs are homogeneous with respect to their language (they are all written in the TPTP language), which makes it easy to compare formulae. For this work formula equivalence is determined syntactically, modulo variable and Skolem symbol renaming, and commutativity over disjunction in clauses. The proofs are also homogeneous with respect to their proof process (they are all proofs by CNF refutation), but they do differ in the inference rules used. In some cases the rules are different in principle, e.g., binary resolution is different from paramodulation. In cases where the same rule is used, almost all ATP systems name the rule differently in their proof outputs, thus making it difficult to recognize equivalent inferences based on the inference rule names. The differences in the inference rules and inference rule naming are bypassed in this work, by identifying an inference step with it's parent and conclusion formulae: if equivalent parents infer an equivalent conclusion, then an equivalent inference step has been performed.

In Section 2.2 it becomes clear that it is unreasonable to *rank* proofs according to the measured artifacts. Instead, it is possible to say only that proofs are *different*. Depending on the user's point of view, different proofs may be preferable. For examples: in mathematics, different proofs can demonstrate different levels of mathematical "elegance", or show that a theorem can be proved from different axiomatic bases; in formal methods, different proofs can be easier or harder to verify, or provide different insights into the structure or process being analyzed; in planning, different proofs can correspond to different sequences of actions that lead to the same goal state; in knowledge based reasoning, different proofs can be built from different knowledge bases, allowing independent agents with different knowledge to reach a common conclusion; in security analysis, different proofs can rely on different ground observations, thus increasing trust in the intelligence information that is generated. This observation motivates the position taken in this paper – *different proofs are good proofs*. A set of proofs is better if it contains proofs that are largely different from each other. Consequently, this provides a way to compare proofs and sets of proofs:

- The quality of a proof in a set of proofs is it's difference from the other proofs in the set.
- The quality of a set of proofs is (a slightly modified version of) its generalized clustering coefficient [8], based on the differences between the proofs.

2.1 TSTP Proof Artifacts

Three artifacts are naturally considered for comparing TSTP proofs: (1) the FOF axioms in the proofs (purely axioms, i.e., excluding the conjecture), (2) the inferred formulae in the proofs (including those inferred from the conjecture), and (3) the inference steps in the proofs. In all three cases, only those artifacts that are part of the proofs are considered, i.e., formulae that were (unnecessarily) inferred by the ATP system, and that are not part of the proof, are ignored. In the context of TSTP proofs it is natural to divide the inferred formulae into three subsets - (2a) the FOF formulae inferred in the FOF to CNF conversion, (2b) the CNF formulae inferred by the FOF to CNF conversion, and (2c) all the CNF formulae (a superset of (2b)). Similarly, the inference steps naturally divide into two subsets - (3a) the inference steps in the FOF to CNF conversion, and (3b) the CNF inference steps from CNF parents to CNF conclusions.

Counting formulae and inference steps in a proof does not produce data that can be meaningfully compared. Comparing the numbers of FOF axioms in proofs can be meaningless, because different axioms contain different information - using many axioms that each contain a small amount of information is incomparable with using fewer axioms that each contain a large amount of information. The in-principle differences between inference rules mean that it is not meaningful to Comparing the numbers of inferred formulae or the numbers of inference steps in proofs can be meaningless, because of the in-principle differences between inference rules - a proof that uses inferences rules that take "large" steps, e.g., hyperresolution, is likely to have less steps and less inferred formulae than one that uses rules that take "smaller" steps, e.g., binary resolution. As is explained in Section 2.2, it is easy to create proofs with less or more inferences, by combining inferences into larger steps, or splitting inferences into smaller steps. Thus, rather than counting these artifacts, is it necessary to *compare the sets* of these artifacts. Set comparisons can be made in terms of equivalence, subset/superset, intersection, and disjointness. For this work the difference between sets of artifacts is measured by the Jaccard distance [6] between the sets.

2.2 Comparing Sets of TSTP Proof Artifacts

A comparison of the sets of FOF axioms in proofs ((1) above) provides useful differentiation between proofs. A proof whose FOF axiom set is a subset of that in another proof is often considered better. However, in other domains, e.g., security analysis [3], a proof that has a superset of axioms might be considered better, because more information sources contribute to the proof. Either way, this provides a way to compare proofs:

- Proofs that have different FOF axiom sets, and one FOF axiom set is a subset of the other, are different. The greater the Jaccard distance between the FOF axiom sets, the greater the difference between the proofs.

Proofs that have different FOF axiom sets, but none of the sets is a subset of another, typically cannot be ranked according to the FOF axiom sets, for the same reason as not counting the FOF axioms - different axioms contain different information. However, the fact that proofs have different FOF axiom sets is useful information, because it shows that the theorem can be proved from different axiomatic bases. This provides a way to compare proofs:

- Proofs that have different FOF axiom sets, but neither FOF axiom set is a subset of the other, are different. The greater the Jaccard distance between the FOF axiom sets, the greater the difference between the proofs.

A comparison of the sets of inferred formulae in proofs ((2) above) provides some useful differentiation between the proofs. The sets FOF formulae inferred by the FOF to CNF conversion process ((2a) above) are of lesser interest, because the inference steps are simply satisfiability preserving - generally no inferences that combine information are performed. The sets of CNF formulae inferred by the FOF to CNF conversion ((2b) above) are of interest. These CNF formulae are called the *CNF leaves* of the proof. (Some ATP systems do not output details of their FOF to CNF conversion, and thus these CNF formulae appear to be the leaves of their proofs.) Smaller CNF leaf sets are typically easier to refute than larger CNF leaf sets [22, 7]. However, this is not always the case, because the availability of extra clauses, which can be redundant, can be helpful for finding a proof. A comparison of the CNF leaf sets is particularly interesting if the proofs have the same FOF axioms. In this situation different CNF leaf sets show that different FOF to CNF conversions have been performed, and an ATP system may benefit from using another's CNF leaf set.[1] These observations provide two ways to compare proofs:

- Proofs that have the same FOF axiom set and different CNF leaf sets, and one CNF leaf set is a subset of the other, are different. The greater the Jaccard distance between the CNF leaf sets, the greater the difference between the FOF to CNF conversions. This is particularly interesting if the proofs have the same FOF axiom set.
- Proofs that have the same FOF axiom set and different CNF leaf sets, but neither CNF leaf set is a subset of the other, are different. The greater the Jaccard distance between the CNF leaf sets, the greater the difference between the FOF to CNF conversions.

A comparison of the sets of all CNF formulae in proofs ((2c) above) is of interest only if the proofs have the same FOF axiom set – proofs that have different FOF axiom sets naturally have different CNF formulae sets. In contrast to the FOF axiom sets and CNF leaf sets, the fact that the CNF formula set in one proof is a subset of that in another proof is not interesting, because it's easy to make a proof with a subset of CNF formulae: Given a proof, a new proof with a singleton CNF formula set can be formed by removing all internal formulae of the original proof, and inferring the root `$false` clause from the FOF axioms by the inference rule "the ATP system". A less extreme variant of that idea is implemented in the AGInT system, which prunes out less interesting formulae in a derivation [11]. Conversely, it is almost always possible to transform a proof into one with a superset of CNF formulae, by expanding inferences into smaller steps [26]. A comparison of CNF formula sets is particularly interesting if the proofs have the same CNF leaf set, because it indicates that the CNF leaf clauses have been combined in different ways. This leads to a way to compare proofs:[2]

- Proofs that have the same FOF axiom set and different CNF formulae sets, but neither CNF formula set is a subset of the other, are different. The greater the Jaccard distance between the CNF formula sets, the greater the difference between the proofs. This is particularly interesting if the proofs have the same CNF leaf set.

A comparison of the sets of inference steps in proofs ((3) above) provides some useful differentiation between the proofs. As for the inferred formulae, the inferences steps in the FOF to CNF conversion ((3a) above) are of lesser interest. The sets of CNF inference steps from CNF parents to CNF conclusions ((3b) above) are of interest. Proofs that have different inferred formulae necessarily have different inference steps (the converse is not true because different inference steps can use the same formulae in different ways), which means that a comparison of CNF inference sets is particularly interesting if the proofs steps have the same CNF leaf sets. Also as for the sets of inferred formulae, a subset relationship is not interesting. These constraints provide a way to compare proofs:

- Proofs that have the same CNF leaf set and different CNF inference sets, but neither CNF inference set is a subset of the other, are different. The greater the Jaccard distance between the CNF inference sets, the greater the difference between the proofs. This is particularly interesting if the proofs have the same CNF leaf set.

[1] This is the approach taken in the Fampire ATP system, which uses the FOF to CNF conversion of the SPASS ATP system [25], and provides the resultant CNF formulae to the inference engine of the Vampire ATP system [12]. This combination has been highly effective [19].

[2] In this regard it is also important to note that proof quality in terms of inference step granularity must be associated with an intended use of the proof [14]. Data structures that support alternative proofs, and views of the proofs at different granularities, have been devised to support this [1].

2.3 Comparing Sets of TSTP Proofs

Given the various ways of measuring and comparing sets of proof artifacts, described in Section 2.2, it is possible to measure and compare sets of proofs. As described in Section 2, a set of proofs is better if it contains proofs that are largely different from each other. A simple way to measure this for a set of proofs is to determine the maximal difference between two proofs in the set. This data is provided in Table 1 of Section 3. A more comprehensive measure is based on the generalized clustering coefficient [8]. This measure views the elements of a set as vertices of a graph, and the weighted relationships between the elements as the weighted edges of the graph. For each pair of edges coincident on a vertex, the pair forms an *closed triplet* (of vertices) if the vertices at the the edges' other ends are linked by a *closing edge*, and and *open triplet* otherwise. The weight of a triplet is some function of the two edges' (the two edges that formed the triplet) weights, e.g., average, minimum, maximum, etc. The generalized clustering coefficient is then

$$C_w = \frac{\sum_{closed\ triplets} triplet\_weight}{\sum_{all\ triplets} triplet\_weight}$$

A high C_w indicates that the vertices of the graph, and correspondingly the elements of the set, are highly clustered, and vice versa.

For this work, the proofs of a set form the vertices of the graph, and the Jaccard similarity (1.0 minus the Jaccard distance) between proofs is used for edge weights. The weight of a triplet is computed as the average of the two edges' weights. In this application the graph is complete, and thus the standard C_w is 1.0. The modified C_{sw} coefficient takes into account the Jaccard similarity on the closing edge of a closed triplet, by using it as a scaling factor for the triplet's weight.

$$C_{sw} = \frac{\sum_{all\ triplets} triplet\_weight \times closing\_edge\_weight}{\sum_{all\ triplets} triplet\_weight}$$

For example, for three (somewhat similar) proofs A, B, and C, with Jaccard similarities A-B = 0.80, B-C = 0.60, and C-A = 0.40, the triplet formed from the edges A-B and B-C has weight 0.70, and scaled weight 0.28. Thus

$$C_{sw} = \frac{0.40 + 0.36 + 0.28}{0.50 + 0.60 + 0.70} = 0.58$$

For another example, for three (quite different) proofs A, B, and C, with Jaccard similarities A-B = 0.50, B-C = 0.30, and A-C = 0.10,

$$C_{sw} = \frac{0.10 + 0.09 + 0.04}{0.20 + 0.30 + 0.40} = 0.26$$

For a set of proofs that are all the same, so that all edge weights are 1.0, $C_{sw} = 1.0$. For a set of proofs that are all completely different, so that all edge weights are 0.0, $C_{sw} = 0.0$ (noting that is necessary to not do the math, to avoid a division by zero). For two proofs, C_{sw} is defined to be the Jaccard similarity between the two sets of artifacts. C_{sw} is undefined for single proofs. As difference is quality, a low C_{sw} value, indicating a highly unclustered set of proofs, is preferred. (The dual computation, using Jaccard distances, provide a converse measure, of non-clustering.)

3 Results

Tables 1 and 2 show data for TSTP proofs for a selection of TPTP problems. The proofs are from the ATP systems Ayane 1.1, EP 1.1[15], Metis 2.2 [5], SInE 0.4 [4], SNARK 20080805r018b

[17], SOS 2.0 [16], SPASS 3.5 [25], Vampire 11.0 [12], and VampireLT 10.0. The selection was motivated by the long-term goal mentioned in Section 4, of combining proofs. For combining to work the problems need to have common language and axioms, and there have to be some proofs to be combined. These problems all come from the MPTP contribution to the TPTP [24], and share common axioms from the Mizar Mathematical Library (MML) [13].

Table 1 shows maximal Jaccard distances between sets of proof artifacts. The columns Table 1 correspond to the Jaccard distances described in the bullet points of Section 2.2. The # columns show the numbers of different sets. The $\subset$ columns show the maximal Jaccard distances between pairs of sets that have a subset relationship. The $\not\subseteq$ columns show the maximal Jaccard distances between pairs of sets that do not have a subset relationship. The * superscript means that the data is for pairs of sets that have the same FOF axiom set, and a ** superscript means the data is for pairs of sets that have the same CNF leaf set. A dash indicates that there is no pair of sets that meets the criteria for that column, so there cannot be a maximal value. Table 2 shows the clustering coefficients for the sets of proof artifacts.

The results show (for this selection of problems) that different ATP systems regularly use different axiomatic bases for proving a given theorem, sometimes a strict subset of axioms, and sometimes a different set of axioms. In some cases the difference is quite marked, e.g., for `SEU140+2` EP and SInE use disjoint axiom sets. The CNF leaf sets produced by the FOF to CNF conversions of the ATP systems are also quite different, except when the same FOF axioms are used. Some CNF leaf sets are subsets of others, and larger differences are found between non-subsets, e.g., for `SEU142+1` EP and Metis have the same FOF axiom sets but their CNF leafs sets intersect only at the clauses inferred from the conjecture of the problem. This shows that they have very different FOF to CNF conversion processes. There are large differences between the CNF formula sets of the ATP systems' proofs, even when they start from the same CNF leaf set. This is not unexpected, because the different systems often use different sets of inference rules, and almost always use different heuristics that lead to the use of common inference rules in different orders. Finally, the different proofs almost always have different CNF inference sets. This is an amplification (and cause) of the differences in the CNF formula sets. However, there are some cases where there are similarities between the CNF formula sets, e.g., for `SEU140+1` the Jaccard distance between the sets of EP and SPASS is only 0.55, and for `SEU303+3` the Jaccard distance between the sets of EP and Metis is only 0.56.

From these statistics one is tempted to examine more closely those cases where there are large differences between the proofs. Interesting insights can be obtained. For example, problem `SEU140+2` is quite simple - if $A \subset B$, and $B \cap C = \emptyset$, then $A \cap C = \emptyset$. Three quite distinct proofs can be observed:

- EP finds the most elegant proof, directly using notions of set membership, subset, and disjointness.

- Metis and SInE use notions of set membership, subset, disjointness, empty sets, difference, and intersection.

- VampireLT uses notions of set membership, subset, disjointness, empty sets, difference, intersection, and union.

An examination of the original human proof in the MML [2] shows that the human's (Grzegorz Bancerek's) proof is similar to that of Metis and SInE. One might conclude that EP finds a better proof than the human.

Problem	FOF axioms			CNF leaves				CNFs			CNF Inf		
	#	$\subset$	$\not\subseteq$	#	$\subset$	$\subset^*$	$\not\subseteq^*$	#	$\not\subseteq^*$	$\not\subseteq^{**}$	#	$\not\subseteq$	$\not\subseteq^{**}$
NUM390+1	2	0.45	-	4	0.31	0.00	0.53	9	0.95	0.86	9	1.00	0.96
NUM401+1	3	-	0.53	4	-	-	-	4	-	-	4	0.98	-
NUM404+1	3	0.13	0.33	4	0.12	-	-	4	-	-	4	0.95	-
PUZ001+1	1	0.00	-	2	0.00	0.00	-	8	0.89	0.96	8	1.00	1.00
SET914+1	3	0.17	0.38	8	0.33	0.22	-	9	0.71	0.31	9	1.00	0.64
SET934+1	1	0.00	-	2	-	-	0.40	2	0.69	-	2	0.86	-
SEU020+1	1	0.00	-	2	0.00	0.00	0.46	6	0.73	0.68	6	1.00	1.00
SEU080+1	1	0.00	-	7	-	-	0.20	7	0.66	-	7	0.95	-
SEU085+1	1	0.00	-	2	0.00	0.00	1.00	4	0.33	0.33	4	1.00	1.00
SEU089+1	1	0.00	-	1	0.00	0.00	-	7	0.50	0.59	7	1.00	1.00
SEU090+1	1	0.00	-	2	0.27	0.00	-	4	0.38	0.50	3	1.00	1.00
SEU096+1	2	0.00	0.67	1	0.00	0.00	-	8	0.64	0.64	8	1.00	1.00
SEU118+1	4	0.29	0.57	6	0.27	0.00	-	9	0.57	0.57	9	1.00	1.00
SEU137+1	0	-	-	1	-	-	-	1	-	-	1	-	-
SEU137+2	1	0.00	-	8	-	-	0.40	8	0.94	-	8	1.00	-
SEU139+1	3	0.33	0.75	5	0.29	0.00	-	9	0.82	0.82	9	1.00	0.89
SEU139+2	5	0.33	0.86	7	0.00	0.00	-	11	0.33	0.57	11	1.00	1.00
SEU140+1	2	0.25	-	4	0.46	0.00	-	8	0.82	0.82	8	1.00	0.93
SEU140+2	4	0.85	0.92	6	0.13	0.00	-	8	0.20	0.47	8	1.00	1.00
SEU141+1	3	0.22	0.60	5	-	-	-	5	-	-	5	1.00	-
SEU141+2	4	-	0.92	7	-	-	-	7	-	-	7	1.00	-
SEU142+1	3	0.33	0.50	8	-	-	0.89	8	0.94	-	8	1.00	-
SEU142+2	2	0.67	-	3	-	-	-	3	-	-	3	1.00	-
SEU263+1	2	0.11	-	2	0.08	0.00	-	8	0.57	0.57	8	0.96	0.95
SEU295+1	1	0.00	-	3	0.00	0.00	1.00	5	0.33	0.33	5	1.00	1.00
SEU295+2	2	0.50	-	2	0.25	0.00	-	6	-	0.33	6	1.00	1.00
SEU295+3	1	0.00	-	3	0.00	0.00	1.00	5	0.33	0.33	5	1.00	1.00
SEU303+1	2	0.00	0.67	2	0.00	0.00	-	8	0.94	0.94	9	1.00	1.00
SEU303+2	1	0.00	-	2	0.14	0.00	-	4	0.31	0.42	4	1.00	1.00
SEU303+3	2	0.00	0.67	1	0.00	0.00	-	7	0.94	0.94	8	1.00	1.00
SEU353+1	1	0.00	-	7	-	-	0.25	7	0.70	-	7	1.00	-

Table 1: Maximal Jaccard distances between proofs

4 Conclusion

This paper has taken the position that *different proofs are good proofs*, based on observations made about proofs in the TSTP solution library. The specific measures presented in this paper are based on proofs of first-order logic problems, from CNF-refutation based ATP systems. However, the principle that different proofs are good proofs is generic, and can be applied to proofs in other logical systems. In order to do this, it is necessary to identify and measure proof artifacts that are naturally considered in those systems. This does not discount the possibility that in other logical systems it may also be possible to rank proofs, based on the measured artifacts.

When the position that different proofs are good proofs is adopted, the quality of a set of proofs can be measured by its clustering coefficient, with less clustering being higher quality.

Problem	FOF axioms #		CNF leaves #		CNFs #		CNF Inf #	
NUM390+1	6	0.86	9	0.67	9	0.30	9	0.04
NUM401+1	3	0.55	4	0.54	4	0.23	4	0.06
NUM404+1	3	0.76	4	0.67	4	0.43	4	0.14
PUZ001+1	5	1.00	9	0.92	9	0.35	9	0.09
SET914+1	5	0.81	9	0.62	9	0.33	9	0.04
SET934+1	2	1.00	2	0.60	2	0.31	2	0.14
SEU020+1	3	1.00	7	0.88	7	0.50	7	0.11
SEU080+1	3	1.00	7	0.80	7	0.51	7	0.20
SEU085+1	6	1.00	11	0.89	10	0.84	10	0.44
SEU089+1	4	1.00	8	1.00	8	0.64	8	0.09
SEU090+1	4	1.00	6	0.91	6	0.58	6	0.10
SEU096+1	4	0.46	8	1.00	8	0.59	8	0.05
SEU118+1	6	0.65	9	0.73	9	0.44	9	0.12
SEU137+1	0	-	1	-	1	-	1	-
SEU137+2	4	1.00	8	0.60	8	0.35	8	0.03
SEU139+1	5	0.69	10	0.52	10	0.34	10	0.09
SEU139+2	6	0.37	11	0.50	11	0.33	11	0.09
SEU140+1	4	0.82	8	0.80	8	0.43	8	0.16
SEU140+2	5	0.17	9	0.49	9	0.31	9	0.04
SEU141+1	3	0.54	5	0.31	5	0.17	5	0.03
SEU141+2	4	0.34	7	0.16	7	0.07	7	0.01
SEU142+1	4	0.70	8	0.23	8	0.12	8	0.01
SEU142+2	2	0.33	3	0.18	3	0.11	3	0.00
SEU263+1	5	0.96	9	0.97	9	0.58	9	0.17
SEU295+1	6	1.00	11	0.81	10	0.78	10	0.47
SEU295+2	3	0.62	6	0.50	5	0.53	5	0.17
SEU295+3	6	1.00	11	0.81	10	0.78	10	0.47
SEU303+1	6	0.81	11	0.95	11	0.59	11	0.14
SEU303+2	2	1.00	5	0.94	5	0.61	5	0.00
SEU303+3	5	0.55	9	1.00	9	0.61	9	0.18
SEU353+1	3	1.00	7	0.54	7	0.31	7	0.08

Table 2: Clustering coefficients for proofs

Given a set of proofs, the quality of a new proof can be measured by its difference from the given set, either by considering its difference from the proofs in the set, or by adding the new proof to the set and noting the change in the clustering coefficient.

High quality (i.e., highly differing) sets of proofs are strong candidates for creating new proofs by *proof combination*, which is the topic of current research. By combining very different proofs it is possible to construct new proofs that are again different from those being combined.

Different proofs provide users with choices, which is a good thing.

References

[1] S. Autexier, C. Benzmüller, D. Dietrich, A. Meier, and C-P. Wirth. A Generic Modular Data Structure for Proof Attempts Alternating on Ideas and Granularity. In M. Kohlhase, editor, *Proceedings of*

the 4th International Conference on Mathematical Knowledge Management, number 3863 in Lecture Notes in Artificial Intelligence, pages 126–142. Springer-Verlag, 2006.

[2] G. Bancerek, N. Endou, and Y. Sakai. On the Characterizations of Compactness. *Journal of Formalized Mathematics*, 9(4):733–738, 2001.

[3] A. Cowell, D. McGuinness, C. Varley, and D. Thurman. Knowledge-Worker Requirements for Next Generation Query Answering and Explanation Systems. In M. Zhou and M. Maybury, editors, *Proceedings of the Workshop on Intelligent User Interfaces for Intelligence Analysis*. ACM Press, 2006.

[4] K. Hoder. *Automated Reasoning in Large Knowledge Bases*. PhD thesis, Charles University in Prague, Prague, Czech Republic, 2008.

[5] J. Hurd. First-Order Proof Tactics in Higher-Order Logic Theorem Provers. In M. Archer, B. Di Vito, and C. Munoz, editors, *Proceedings of the 1st International Workshop on Design and Application of Strategies/Tactics in Higher Order Logics*, number NASA/CP-2003-212448 in NASA Technical Reports, pages 56–68, 2003.

[6] P. Jaccard. Étude Comparative de la Distribution Florale Dans une Portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[7] A. Nonnengart and C. Weidenbach. Computing Small Clause Normal Forms. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 335–367. Elsevier Science, 2001.

[8] T. Opsahl and P. Panzarasa. Clustering in Weighted Networks. *Social Networks*, 31(2):155–163, 2009.

[9] P. Pinheiro da Silva, D.L. McGuinness, and R. Fikes. A Proof Markup Language for Semantic Web Services. *Information Systems*, 31(4-5):381–395, 2006.

[10] P. Pinheiro da Silva, G. Sutcliffe, C. Chang, L. Ding, N. del Rio, and D. McGuinness. Presenting TSTP Proofs with Inference Web Tools. In R. Schmidt, B. Konev, and S. Schulz, editors, *Proceedings of the Workshop on Practical Aspects of Automated Reasoning, 4th International Joint Conference on Automated Reasoning*, number 373 in CEUR Workshop Proceedings, pages 81–93, 2008.

[11] Y. Puzis, Y. Gao, and G. Sutcliffe. Automated Generation of Interesting Theorems. In G. Sutcliffe and R. Goebel, editors, *Proceedings of the 19th International FLAIRS Conference*, pages 49–54. AAAI Press, 2006.

[12] A. Riazanov and A. Voronkov. The Design and Implementation of Vampire. *AI Communications*, 15(2-3):91–110, 2002.

[13] P. Rudnicki and A. Trybulec. On the Integrity of a Repository of Formal Mathematics. In A. Asperti, B. Buchberger, and J.H. Davenport, editors, *Proceedings of the 2nd International Conference on Mathematical Knowledge Management*, number 2594 in Lecture Notes in Computer Science, pages 162–174. Springer-Verlag, 2003.

[14] M. Schiller and C. Benzmüller. Presenting Proofs with Adapted Granularity. In B. Mertsching, editor, *Proceedings of the 32nd Annual Conference on Artificial Intelligence*, number 5803 in Lecture Notes in Artificial Intelligence, pages 289–297, 2009.

[15] S. Schulz. A Comparison of Different Techniques for Grounding Near-Propositional CNF Formulae. In S. Haller and G. Simmons, editors, *Proceedings of the 15th International FLAIRS Conference*, pages 72–76. AAAI Press, 2002.

[16] J.K. Slaney, A. Binas, and D. Price. Guiding a Theorem Prover with Soft Constraints. In R. López de Mántaras and L. Saitta, editors, *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 221–225, 2004.

[17] M.E. Stickel. SNARK - SRI's New Automated Reasoning Kit. http://www.ai.sri.com/ stickel/snark.html.

[18] G. Sutcliffe. TPTP, TSTP, CASC, etc. In V. Diekert, M. Volkov, and A. Voronkov, editors, *Proceedings of the 2nd International Computer Science Symposium in Russia*, number 4649 in Lecture Notes in Computer Science, pages 7–23. Springer-Verlag, 2007.

[19] G. Sutcliffe. The CADE-21 Automated Theorem Proving System Competition. *AI Communications*, 21(1):71–82, 2008.

[20] 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.

[21] 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.

[22] G. Sutcliffe and S. Melville. The Practice of Clausification in Automatic Theorem Proving. *South African Computer Journal*, 18:57–68, 1996.

[23] 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, 2006.

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

[25] 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.

[26] J. Zimmer, A. Meier, G. Sutcliffe, and Y. Zhang. Integrated Proof Transformation Services. In C. Benzmüller and W. Windsteiger, editors, *Proceedings of the Workshop on Computer-Supported Mathematical Theory Development, 2nd International Joint Conference on Automated Reasoning*, Electronic Notes in Theoretical Computer Science, 2004.