Large Theory Reasoning with SUMO at CASC

Adam Pease [a,*], Geoff Sutcliffe [b], Nick Siegel [a], and Steven Trac [b]
[a] *Articulate Software, USA*
E-mail: {apease,nsiegel}@articulatesoftware.com
[b] *University of Miami, USA*
E-mail: {geoff,strac}@cs.miami.edu

The Suggested Upper Merged Ontology (SUMO) has provided the TPTP problem library with problems that have large numbers of axioms, of which typically only a few are needed to prove any given conjecture. The LTB division of the CADE ATP System Competition tests the performance of ATP systems on these types of problems. The SUMO problems were used in the SMO category of the LTB division in 2008. This paper presents an analysis of the performance of the 2007 and 2008 CASC entrants on the SUMO problems, illustrating the improvements that can be achieved by various tuning techniques.

Keywords: Automated reasoning, Large theories, Commonsense reasoning

1. Introduction

Many reasoning applications rely on knowledge bases that have large numbers of axioms, of which only a few are relevant to any given query. The LTB division of the CADE ATP System Competition CASC was created in 2008, to the test performance of ATP systems on these types of problems. The SMO problem category of the 2008 LTB division used problems based on the Suggested Upper Merged Ontology (SUMO) [8]. SUMO problems mimic the situation found in many practical reasoning applications in decision support and expert systems. This paper analyses the performance of ATP systems on the SUMO problems, comparing their performance before any tuning done for the 2008 CASC with their performance in the competition and on the SUMO problems in subsequent testing.

*This work has been funded by a number of sources, including the Army Research Institute. We are grateful for their investment.

2. The Suggested Upper Merged Ontology

SUMO is a free, formal ontology of about 1000 terms and 4000 definitional statements. It is a formal theory of common-sense concepts, ranging from general notions of time, space, and action, to specific information about application domains such as economics and physical sciences. It is provided in the SUO-KIF language [11], which is a first-order logic with some higher-order extensions. The original SUMO, consisting of only (what were somewhat arbitrarily considered) "upper level" terms, was first released in 2001. Upper level terms are the most general terms that name and define concepts, and are not tied to any particular specialized domain of knowledge. They include notions like a position in time or space, a process or event, the Aristotelian distinction of object versus substance, and so on. The original SUMO was designed to have roughly 1000 upper level terms, with associated definitions. This number was chosen to keep SUMO at a managable size, and is entirely arbitrary, since there is no objective test for whether a term is "upper level" or not. As new terms are added to SUMO, the lowest level terms in its hierarchy are migrated to the lower level ontologies that depend on them, in order to keep its size close to the 1000 term guideline. A next phase of effort after the creation of SUMO was to create the MId-Level Ontology (MILO) for terms that span individual domains, but are more specific than those in SUMO. Subsequently, several years of extension and application domain ontology development enlarged the combined theory to an approximate total of 21000 terms and 73000 axioms, including 3000 rules. SUMO has been mapped to the WordNet lexicon [3] of over

AI Communications
ISSN 0921-7126, IOS Press.

117000 noun, verb, adjective, and adverb word senses [9], which not only acts as a check on coverage and completeness, but also provides a basis for work in natural language processing [12,2,16]. Translation templates that allow SUMO terms and axioms to be expressed in different natural languages, including German, Hindi, Chinese, Czech, Italian and many others, have been created in order to help prevent any linguistic or cultural bias in the ontology.

SUMO is now in its 75th free version, having undergone eight years of development, review by a community of hundreds of people, and application in expert reasoning and linguistics. Various versions of SUMO have been subjected to formal verification with Vampire [14], which until recently was the only ATP system integrated into SUMO's browsing and inference tool suite, Sigma [11]. The TPTPWorld [18] has now been integrated into Sigma [22], allowing many different TPTP-compliant ATP systems to be used with the preprocessing, optimization, and proof display capabilities of Sigma. SUMO and all the associated tools are available at `www.ontologyportal.org`.

Prior work [10] has described how SUMO can be transformed into a strictly first-order form. SUMO has also been translated into the OWL semantic web language (which is a necessarily lossy translation, given the limited expressiveness of OWL). There are a variety of possibilities for first-order transformation of what appear to be higher-order constructs. Those used in the export of SUMO to the TPTP (see Section 3) include quoting all nested formulae so that they become constants, and expanding schemas for variable-arity relations so that one higher-order axiom may result in many first-order axioms. There are also a variety of possibilities for caching, which are aimed generally at optimizing the performance of ATP systems on SUMO-based problems. By implementing different strategies such as preprocessing code that is used to transform the axioms prior to sending them to an ATP system, optimization options can be tested without needing to understand the code of many different ATP systems. If those ideas are successful, prover developers can then adopt and implement them more efficiently within their different systems. This possibility for future collaboration is now open, given the basic level of compatibility that has been established between SUMO and the ATP systems that participated in the 2008 CASC.

To give a flavor of the types of queries that might be answered using SUMO axioms, an example is presented here, in its original SUO-KIF syntax. The question is "Is a banana slug an invertebrate?". The problem has an example instance of a banana slug ...

```
(instance BananaSlug10-1 Animal)
```

... a statement that animals without a spinal column are not vertebrates ...

```
(=>
  (and
    (instance ?A Animal)
    (not
      (exists (?PART)
        (and
          (instance ?PART SpinalColumn)
          (part ?PART ?A)))))
  (not
    (instance ?A Vertebrate)))
```

... statements that the banana slug does not have a spinal column but does have some body parts ...

```
(not
  (exists (?SPINE)
    (and
      (instance ?SPINE SpinalColumn)
      (part ?SPINE BananaSlug10-1))))

(and
  (instance BodyPart10-1 BodyPart)
  (component BodyPart10-1 BananaSlug10-1))
```

... and the conjecture ...

```
(instance BananaSlug10-1 Invertebrate)
```

A successful reasoning system will find the following axioms in SUMO, and apply them to prove the conjecture. The axioms state that animals are either vertebrates or invertebrates ...

```
(partition Animal Vertebrate Invertebrate)
```

... that a partition of subclasses is not ordered ...

```
(=>
  (partition ?SUPER ?SUB1 ?SUB2)
  (partition ?SUPER ?SUB2 ?SUB1))
```

... and that if an individual is an instance of a partitioned class, and is not an instance of one of the partitions, then it must be an instance of the other partition ...

```
(=>
  (and
    (partition ?SUPER ?SUB1 ?SUB2)
```

```
    (instance ?INST ?SUPER)
    (not (instance ?INST ?SUB1)))
  (instance ?INST ?SUB2))
```

While this example is trivial when the necessary axioms are provided ahead of time, it becomes very challenging in the context of a large knowledge base, where, in a practical situation, the relevant axioms cannot be known ahead of time. For example, there are thousands of rules involving the term `instance`, and a successful ATP system has to hunt through the axioms in order to find the ones that are relevant to the query being posed.

3. SUMO Problems in the TPTP and CASC

The TPTP contains 105 problems based on SUMO, generated from 35 distinct queries and three background theories (and some problems have some additional problem-specific axioms). The three background theories are (i) just the ∼11000 axioms in SUMO, (ii) the ∼17000 axioms in SUMO+MILO, and (iii) the ∼67000 axioms in SUMO+MILO+domain ontologies. These are the TPTP problems `CSR075` to `CSR109`, with the `+1` versions using just the SUMO axioms, the `+2` versions using the SUMO+MILO axioms, and the `+3` versions using all the axioms. All of the problems are designed to be solved using axioms from only SUMO itself. The MILO and domain axioms are distractions for ATP systems, but can contain symbols that are present in the problem-specific query and axioms. Efficient separation of axioms that don't reference relevant symbols, from those that reference relevant symbols but which aren't needed for the problem solution, from those that are needed for the solution, is a key to solving these problems. Identifying ATP systems with this capability is important for their use in Sigma, and these problems are useful tests for that purpose. The problems were released to the ATP community in TPTP v3.4.0, about five months before the 2008 CASC. Easy online access to TPTP problems is available from `http://www.tptp.org`.

For the LTB division of the 2008 CASC (held at IJCAR 2008 [19]), the SUMO inference prizes totaling US$3000 were awarded to the best performances on the SMO category of the division. The LTB division has two ranking classes – the *proof class* requires the ATP system to produce a proof of how its solution was found, while the *assurance class* merely requires that the system report whether or not there is a solution. In each ranking class the winner received $750, the second place $500, and the third place $250 (a system that wins the proof ranking class can also win the assurance ranking class and that was in fact the case for first and second place).

For the competition, 30 of the 105 problems were selected, with 26 of the selected problems having distinct queries (i.e., 4 of the problems differed from others only by using a subset of the axioms – solving the smaller problem would trivially provide a solution to the larger problem, but none of the ATP systems took advantage of this possibility). All the problems were presented together at one time to the ATP systems, allowing them to process or optimize on them collectively, and to determine for themselves in what order to attempt them. Each ATP system was given an overall time limit of 7200 seconds to solve the 30 problems. This "batch" approach marked a departure from the existing CASC divisions, in which problems are presented individually and sequentially. It was also a departure from most previous applications of formal theorem proving with Sigma, where queries are typically presented sequentially by a user. However, one can easily envision classes of applications where a set of problems are presented at the same time. This test format also allowed machine learning approaches to explore optimizations strategies. Both the sequential and batch test modes are interesting and pragmatic.

Although not the focus of this paper, an additional "validation challenge" was created to support the participation of model finders. The challenge asks ATP systems to verify the consistency of, or provide feedback to repair, each of the SUMO, SUMO+MILO, and SUMO+MILO+domain axiom sets. For each set of axioms there is a prize of $100 for completing the challenge with either result. The winners of the challenge were announced and received their awards at the 2008 CASC results presentation.

4. Pre-CASC Testing

In order to test whether it was even reasonable to base a CASC category on the SUMO problems, all the ATP systems in the SystemOnTPTP suite at the time were run on all 105 SUMO problems

in the TPTP, and the results analyzed. The systems were Bliksem 1.12, CARINE 0.734, CiME 2.01, Darwin 1.4.1, DarwinFM 1.4.1, DCTP 1.31, E 0.999, E-KRHyper 1.0, EQP 0.9d, Equinox 1.3, Fampire 1.3, Faust 1.0, FDP 0.9.16, Fiesta 2, Gandalf c-2.6, Geo 2007f, GrAnDe 1.1, iProver 0.2, leanCoP 2.0, LeanTAP 2.3, Mace2 2.2, Mace4 1207, Matita 0.1.0, Metis 2.0, Muscadet 2.7a, Otter 3.3, Paradox 2.3, Prover9 1207, S-SETHEO 0.0, SETHEO 3.3, SNARK 20070805, SOS 2.0, SPASS 3.0, SRASS 0.1, Theo 2006, Vampire 9.0, Waldmeister 806, zChaff 04.11.15, and Zenon 0.5.0. The problems were presented to the ATP systems individually and sequentially. A time limit of 600 seconds was imposed on each ATP system run.

Overall performance, in terms of the number of problems solved for each background theory, the total number of problems solved, the average time over all problems, the average time over the problems solved, and the solving efficiency (the total number of problems solved divided by the average time over the problems solved), is shown in Table 1. The systems that are not listed did not solve any problems. Even the best performing systems did not solve a majority of the 105 problems in the test set: Vampire solved 31, Fampire solved 20, E solved 15, and Metis solved 14, with the other systems in the single digits or having no solutions at all. Systems generally failed because of timeouts, rather than errors in parsing or memory space. It is notable that Zenon and Equinox, both lower ranked overall, were top performers when the largest background axiomatization was used. Generally, the overall performance and the individual axiomatization performances do not align well, indicating sensitivity to the increasing number of unnecessary axioms. The average times taken and the solving efficiencies are also unaligned - the ranking of the systems is very dependent on the problems and the desired performance characteristics. The average time over all problems is appropriate for an interactive context, where a system that quickly gets its answer or abandons its attempt is more attractive than one that often takes nearly the entire allotted time to reach a solution (none of these systems quickly abandoned their proof attempts). The average time over problems solved is appropriate when there is a high confidence that the system will solve each problem. Efficiency provides a balance of ability to solve problems with the average time over problems solved.

System	SU MO	MI LO	Do ms	Tot al	Avg. all	Avg. sol.	Eff.
Vampire 9.0	18	13	0	31	463	135	0.23
Fampire 1.3	20	0	0	20	586	525	0.04
E 0.999	15	0	0	15	549	240	0.06
Metis 2.0	5	5	4	14	528	58	0.24
iProver 0.2	9	0	0	9	559	127	0.07
SPASS 3.0	6	0	0	6	596	524	0.01
leanCoP 2.0	5	0	0	5	572	7	0.75
Darwin 1.4.1	4	0	0	4	577	2	1.88
Equinox 1.3	1	1	1	3	584	30	0.10
Zenon 0.5.0	1	1	1	3	583	12	0.24
Muscadet 2.7a	1	1	0	2	592	203	0.01
SNARK 2007	1	1	0	2	589	6	0.34
Faust 1.0	1	0	0	1	598	351	0.00

Table 1
Pre-CASC Performance Data

The differing strengths of the systems suggested creating a "meta-prover" combining several systems. One strategy is to give Vampire 400 seconds, then give Metis up to 200 seconds if Vampire fails to find a proof. This combined system solves 33 problems, with an average time of 150s and an efficiency of 0.22. The timeslice allocation might be improved further, although great efforts in that direction could be considered overtraining to this problem set.

An analysis was performed to determine what set of systems would cover the maximum number of problems. This is termed a State Of The Art (SOTA) analysis [21]. This analysis is based on a subsumption relationship between the sets of problems solved by the ATP systems. System A subsumes system B if system A solves a strict superset of the problems solved by system B. The result is shown in Figure 1, where an arrow indicates a subsumption relationship. Vampire exclusively solved eight problems solved by no other system. Metis exclusively solved two, and Fampire exclusively solved 1. This analysis suggests creation of a meta-prover composed of Vampire, Metis, and Fampire.

5. The 2008 CASC Results

The SMO category of the LTB division of the 2008 CASC was won by Krystof Hoder for SInE 0.3 [4], in both the proof and assurance ranking

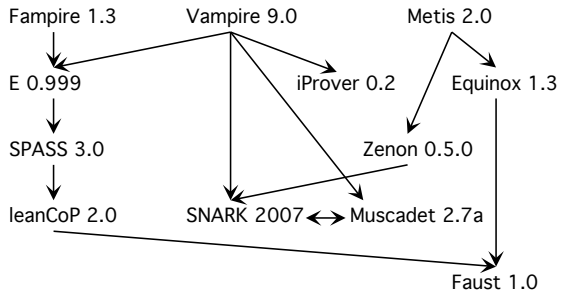

Fig. 1. Pre-CASC SOTA Analysis

classes. Second place in both classes went to Josef Urban for MaLARea 0.3 [23]. Third place went to Konstantin Korovin for iProver 0.5 [5] in the proof class and to Andrei Voronkov for VampireLT-10 in the assurance class. The results were most notable for SInE, which was developed specifically for the 2008 CASC, and which is a meta-prover based on an underlying core first-order ATP system. In the 2008 CASC, E [17] was used as its core system. Although SInE did not have permission to use Vampire as its core ATP system in the official competition, it did enter an unofficial demonstration version using Vampire, for which results that showed it would have performed even better.

MaLARea was notable not only for achieving second place, but also because it incorporates machine learning from previous proofs, and is a meta-prover that uses many state-of-the-art components – E, SPASS [24], Paradox [1], and Mace [6]. MaLARea was able to tune itself to perform well on the problems. The post-CASC testing described in Section 6 does not include MaLARea because those tests were performed sequentially. However, we do envision ongoing research with MaLARea. In particular, we hope to collaborate with the MaLARea developer to provide an incremental learning mode in which each query is saved, so that MaLARea can learn from the queries it has solved so far. While performance on the first few queries will likely not be comparable to the best available systems, over time, given a set of problems based on the same background knowledge, MaLARea should be able to improve.

The validation challenge proved too difficult, or the financial incentive too small, to encourage more than one entrant. However, SInE's author Krystof Hoder was able to find a contradiction in the SUMO+MILO axioms, and therefore won that prize.

6. Post-CASC Testing

The post-CASC testing was done using the same environment as the pre-CASC testing described in Section 4, i.e., the 105 SUMO-based problems were used, the problems were presented to the ATP systems individually and sequentially, and a time limit of 600 seconds was imposed on each run. Old versions of ATP systems were replaced by new versions, as used in the 2008 CASC. This provides an interesting comparison with the pre-CASC performance, and provides insight into changes that might have been influenced by preparation for the competition. The systems were Bliksem 1.12, Darwin 1.4.3, E 1.0pre, Equinox 3.0, Fampire 1.3, Faust 1.0, Geo 2007f, iProver 0.5c, leanCoP 2.0, LeanTAP 2.3, Metis 2.1, Muscadet 3.0, OSHL-S 0.1, Otter 3.3, Prover9 0908, randoCoP 1.1, SInE 0.3, SNARK 20080805r005, SOS 2.0, SPASS 3.01, SRASS 0.1, Theo 2006, Vampire 10.0, VampireLT 10.0, and Zenon 0.5.0.

Table 2 shows the post-CASC performance data of the systems. SInE was the clear winner – it solved significantly more problems, and was significantly faster, than the other ATP systems. It solved 55 problems, compared to a second place of 47. In particular, SInE exclusively solved 13 problems that no other system was able to solve (although it failed to solve some problems that other systems did). One interesting observation that can be seen in the data is that Fampire solved a significant number of problems but solved them at a significantly lower rate than other systems. A number of systems failed to solve any problems, and a number of systems – leanCoP, Equinox, Faust, SPASS, Muscadet, and Zenon solved less than 10 problems.

Table 3 shows the ratios of performance figures between corresponding pre-CASC and post-CASC versions of systems (so that high numbers are good for the total solved and efficiency, low numbers are good for average times). Many systems demonstrated improvement. Overall, iProver had the greatest performance improvement. iProver also exclusively solved two problems (SInE and iProver were the only two systems with exclusive solutions). Not every system improved, some systems whose developments were not tied to large theory processing even saw slight decreases in performance. Others, such as Metis and Fampire, were overtaken as developers who had new ideas, or sim-

System	SU MO	MI LO	Do ms	Tot al	Avg. all	Avg. sol.	Eff.
SInE 0.3	21	16	22	59	267	8	7.39
VampireLT 10.0	16	12	19	47	411	177	0.27
iProver 0.5c	15	14	15	44	378	71	0.62
Vampire 10.0	12	7	13	32	452	115	0.28
E 1.0pre	8	7	11	26	506	219	0.12
Fampire 1.3	7	7	6	20	586	525	0.04
Darwin 1.4.3	4	4	6	14	522	17	0.81
Metis 2.1	6	3	5	14	531	79	0.18
SNARK 2008	6	1	5	12	533	17	0.71
randoCoP 1.1	3	5	2	10	560	184	0.05
SPASS 3.01	2	2	2	6	596	524	0.01
leanCoP 2.0	1	2	2	5	572	7	0.75
Equinox 3.0	0	1	2	3	583	16	0.19
Zenon 0.5.0	0	0	3	3	583	14	0.20
Muscadet 3.0	0	0	2	2	591	150	0.01
Faust 1.0	0	1	0	1	598	351	0.00

Table 2
Post-CASC Performance Data

System	Total solved	Avg. all	Avg. sol.	Eff.
SNARK 2007 & 2008	6.00	0.91	2.90	2.07
iProver 0.2 & 0.5c	4.89	0.68	0.56	8.74
Vampire 9.0 & 10.0	3.56	0.98	0.91	3.93
Darwin 1.4.1 & 1.4.3	3.50	0.90	8.15	0.43
E 0.999 & 1.0pre	1.73	0.92	0.91	1.89
Muscadet 2.7a & 3.0	1.00	1.00	0.74	1.35
Equinox 1.3 & 3.0	1.00	1.00	0.54	1.86
Metis 2.0 & 2.1	1.00	1.01	1.35	0.74

Table 3
Ratios of Pre- and Post-CASC Performance Data

ply more resources to put to the task of handling large theories, created new versions of their systems.

The post-CASC SOTA analysis is considerably more complicated than the pre-CASC analysis, and is shown in Figure 2. Many more problems were solved by more systems. SInE, VampireLT, iProver 0.3, and iProver 0.5c form the smallest set of ATP systems that together solved all problems solved by any system. No one system solved all problems, and no one system could handle all the problems that all other systems could handle collectively.

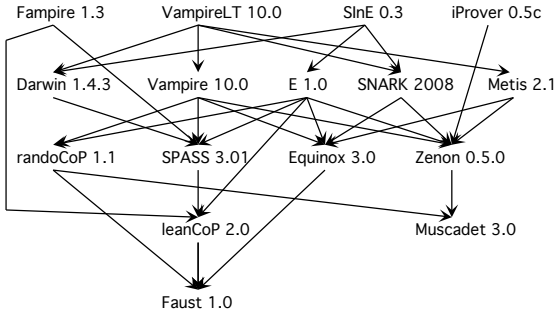

Fig. 2. Post-CASC SOTA Analysis

7. Conclusion

The SMO of the LTB division of CASC motivates the development of high performance reasoning on practical problems that use a broad knowledge base. This has yielded some clear results regarding performance on a new class of problems, as well as providing the application development community with ATP systems that are more closely optimized to the needs of one sort of practical inference.

An intriguing practical development is the SinE system's "relevance detector" preprocessor. In the 2008 CASC it was found that the core ATP system used with SInE did not have a dominant effect on performance. The SInE developer would have liked to have used the highest performance ATP system as its core. However, the Vampire developer did not give permission for it to be used with SInE in the competition, and SInE was able to win using E as its core. This indicates that significant speedups are possible through relatively simple preprocessing optimizations, rather than relying on the use of more complex ATP systems. In particular, the ability to extract a small but sufficient subset of axioms from a large axiomatization, to solve a given problem, is a key capability for the application of automated reasoning over large knowledge bases. In addition to the SInE "relevance detector", there have been other successful efforts in this direction, e.g, [7,20,23,15]. The effects of such approaches can be amplified by combining them with approaches that take advantage of the structure that is typically inherent in large knowledge bases, as done in, e.g., [13].

In order to obtain improvements in a consistent way it will also be necessary to understand *why* different calculi and systems behave so differently

for these types of problems. A deep understanding of the effects of different calculi, search strategies, data structures, preprocessing, etc., is necessary, but getting any reliable results in this direction tends to be extremely difficult. While the 2008 CASC report [19] provides some details, real insights are best obtained by direct communication with the system developers (this is one of the real advantages of entering, or at least attending, CASC!). Therefore, for now, the style of empirical evaluation provided by this paper is a most effective way of building an understanding of automated reasoning for these types of problems.

In the future, the number of SUMO-based problems in the TPTP, and hence the number of problems in the SMO category in CASC, is expected to be increased. This will benefit future empirical testing of ATP systems for such large theory problems. A "stratified" set of problems of different expressiveness is also planned, by extracting the Horn clause and description logic subsets of SUMO, and basing problems on those subsets. The creation of new problems will make it possible to keep some SMO problems hidden until their use in CASC. It remains to be seen whether learning systems will overfit to published TPTP problems, or have to underfit and therefore not achieve significant optimization.

References

[1] K. Claessen and N. Sörensson. New Techniques that Improve MACE-style Finite Model Finding. In P. Baumgartner and C. Fermueller, editors, *Proceedings of the CADE-19 Workshop: Model Computation - Principles, Algorithms, Applications*, 2003.

[2] S. Elkateb, W. Black, H. Rodriquez, M. Alkhalifa, P. Vossen, A. Pease, and C. Fellbaum. Building a WordNet for Arabic. In N. Calzolari, editor, *Proceedings of the 5th International Conference on Language Resources and Evaluation*, 2006.

[3] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

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

[5] K. Korovin. iProver - An Instantiation-Based Theorem Prover for First-order Logic (System Description). In P. Baumgartner, A. Armando, and D. Gilles, editors, *Proceedings of the 4th International Joint Conference on Automated Reasoning*, number 5195 in Lecture Notes in Artificial Intelligence, pages 292–298, 2008.

[6] W.W. McCune. Mace4 Reference Manual and Guide. Technical Report ANL/MCS-TM-264, Argonne National Laboratory, Argonne, USA, 2003.

[7] J. Meng and L. Paulson. Lightweight Relevance Filtering for Machine-Generated Resolution Problems. In G. Sutcliffe, R. Schmidt, and S. Schulz, editors, *Proceedings of the FLoC'06 Workshop on Empirically Successful Computerized Reasoning, 3rd International Joint Conference on Automated Reasoning*, number 192 in CEUR Workshop Proceedings, pages 53–69, 2006.

[8] I. Niles and A. Pease. Towards A Standard Upper Ontology. In C. Welty and B. Smith, editors, *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*, pages 2–9, 2001.

[9] I. Niles and A. Pease. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In H. Arabnia, editor, *Proceedings of the 2003 International Conference on Information and Knowledge Engineering*, pages 412–416, 2003.

[10] L. Paulson and K. Susanto. Source-level Proof Reconstruction for Interactive Theorem Proving. In K. Schneider and J. Brandt, editors, *Proceedings of the 20th International Conference on Theorem Proving in Higher Order Logics*, number 4732 in Lecture Notes in Computer Science. Springer-Verlag, 2007.

[11] A. Pease. The Sigma Ontology Development Environment. In F. Giunchiglia, A. Gomez-Perez, A. Pease, H. Stuckenschmidt, Y. Sure, and S. Willmott, editors, *Proceedings of the IJCAI-03 Workshop on Ontologies and Distributed Systems*, number 71 in CEUR Workshop Proceedings, 2003.

[12] A. Pease and W. Murray. An English to Logic Translator for Ontology-based Knowledge Representation Languages. In C. Zong, editor, *Proceedings of the 2003 International Conference on Natural Language Processing and Knowledge Engineering*, pages 777–783. IEEE Press, 2003.

[13] W. Reif and G. Schellhorn. Theorem Proving in Large Theories. In W. Bibel and P.H. Schmitt, editors, *Automated Deduction, A Basis for Applications*, volume III Applications of *Applied Logic Series*, pages 225–241. Kluwer Academic Publishers, 1998.

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

[15] A. Roederer, Y. Puzis, and G. Sutcliffe. Divvy: A ATP Meta-system based on Axiom Relevance Ordering. In R. Schmidt, editor, *Proceedings of the 22nd International Conference on Automated Deduction*, number 5663 in Lecture Notes in Artificial Intelligence, page To appear. Springer-Verlag, 2009.

[16] J. Scheffczyk, A. Pease, and M. Ellsworth. Linking FrameNet to the Suggested Upper Merged Ontology. In B. Bennett and C. Fellbaum, editors, *Proceedings of the International Conference on Formal Ontology in Information Systems*, pages 289–300. IOS Press, 2006.

[17] S. Schulz. E: A Brainiac Theorem Prover. *AI Communications*, 15(2-3):111–126, 2002.

[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 4th IJCAR Automated Theorem Proving System Competition - CASC. *AI Communications*, 22(1):59–72, 2009.

[20] G. Sutcliffe and Y. Puzis. SRASS - a Semantic Relevance Axiom Selection System. In F. Pfenning, editor, *Proceedings of the 21st International Conference on Automated Deduction*, number 4603 in Lecture Notes in Artificial Intelligence, pages 295–310. Springer-Verlag, 2007.

[21] G. Sutcliffe and C.B. Suttner. Evaluating General Purpose Automated Theorem Proving Systems. *Artificial Intelligence*, 131(1-2):39–54, 2001.

[22] S. Trac, G. Sutcliffe, and A. Pease. Integration of the TPTPWorld into SigmaKEE. 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 103–114, 2008.

[23] 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 D. Gilles, 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.

[24] C. Weidenbach, R. Schmidt, T. Hillenbrand, R. Rusev, and D. Topic. SPASS Version 3.0. In F. Pfenning, editor, *Proceedings of the 21st International Conference on Automated Deduction*, number 4603 in Lecture Notes in Artificial Intelligence, pages 514–520. Springer-Verlag, 2007.