Vampire 4.1-SMT System Description

Giles Reger[1], Martin Suda[2], Andrei Voronkov[1,3,4],
Evgeny Kotelnikov[3], and Laura Kovacs[2,3]

[1] University of Manchester, Manchester, UK
[2] Institute for Information Systems, Vienna University of Technology, Austria
[3] Chalmers University of Technology, Gothenburg, Sweden
[4] EasyChair

Abstract. A system description for SMTCOMP 2016.

General Approach. Vampire [7] is an automatic theorem prover for first-order logic. Vampire implements the calculi of ordered binary resolution [1] and superposition for handling equality [8]. It also implements the Inst-gen calculus [4] and a MACE-style finite model builder [10]. Splitting in resolution-based proof search is controlled by the AVATAR architecture [9, 11]. Both resolution and instantiation based proof search make use of global subsumption [4]. It should be noted, to avoid confusion, that unlike the standard SMT approach of instantiation, Vampire deals directly with non-ground clauses via the first-order standard resolution and superposition calculi.

A number of standard redundancy criteria and simplification techniques are used for pruning the search space: subsumption, tautology deletion, subsumption resolution and rewriting by ordered unit equalities. The reduction ordering is the Knuth-Bendix Ordering. Substitution tree and code tree indexes are used to implement all major operations on sets of terms, literals and clauses. Internally, Vampire works only with clausal normal form. Problems are clausified during preprocessing. Vampire implements many useful preprocessing transformations including the Sine axiom selection algorithm [3].

Theory Reasoning. Vampire supports all logics apart from bit vectors. This is thanks to recent support for a first-class boolean sort [6] and arrays [5]. Both additions are supported by special inference rules and/or preprocessing steps. However, Vampire has no special support for ground problems (see Z3 point below) and is therefore not entered into any *quantifier-free* divisions.

The two main techniques Vampire uses for theory reasoning are:

1. The addition of *theory axioms*. The main technique Vampire uses for non-ground theory reasoning is to add axioms of the theory. This is clearly incomplete but can be effective for a large number of problems.
2. Incorporating Z3 [2] into AVATAR. In this setup the ground part of the problem is passed to Z3 along with a propositional naming of the non-ground part (with no indication of what this names) and the produced model is used to select a sub-problem for Vampire to solve. The result is that Vampire

only deals with problems that have theory-consistent ground parts. In the extreme case where the initial problem is ground, Z3 will be passed the whole problem. To reiterate, we never pass Z3 anything which is non-ground.

Additionally, Vampire incorporates a MACE-style finite-model finding method that operates on multi-sorted problems [10]. Vampire has no support for satisfiability with theories so in all divisions other than UF it can only report `unsat`.

Availability and Licensing. The online home for Vampire is vprover.org where instructions for how to obtain Vampire and information about its licence can be found. In the first instance, please direct any queries to the first author.

Expected Performance. This is the first year Vampire is competing in SMT-COMP and has only recently supported problems in the SMT-LIB format. However, it has a strong track record in the CASC competition. The UF division is closest to Vampire's traditional area of expertise but we believe we are strong in all divisions we have entered. Generally, Vampire should perform best in quantifier-heavy problems; if a problem is mostly-ground there is less that Vampire can achieve in comparison to a traditional SMT solver.

References

1. L. Bachmair and H. Ganzinger. Resolution theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 2, pages 19–99. Elsevier Science, 2001.
2. Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3: an efficient SMT solver. In *Proc. of TACAS*, volume 4963 of *LNCS*, pages 337–340, 2008.
3. Krystof Hoder and Andrei Voronkov. Sine qua non for large theory reasoning. In *CADE-23 2011. Proceedings*, volume 6803 of *Lecture Notes in Computer Science*, pages 299–314. Springer, 2011.
4. Konstantin Korovin. Inst-gen - A modular approach to instantiation-based automated reasoning. In *Programming Logics - Essays in Memory of Harald Ganzinger*, volume 7797 of *Lecture Notes in Computer Science*, pages 239–270. Springer, 2013.
5. Evgenii Kotelnikov, Laura Kovács, Giles Reger, and Andrei Voronkov. The vampire and the FOOL. In *ACM SIGPLAN 2016 proceedings*, pages 37–48, 2016.
6. Evgenii Kotelnikov, Laura Kovács, and Andrei Voronkov. A first class boolean sort in first-order theorem proving and TPTP. In *CICM 2015, Proceedings*, pages 71–86, 2015.
7. Laura Kovács and Andrei Voronkov. First-order theorem proving and Vampire. In *CAV 2013. Proceedings*, pages 1–35, 2013.
8. R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 7, pages 371–443. Elsevier Science, 2001.
9. Giles Reger, Martin Suda, and Andrei Voronkov. Playing with AVATAR. In *CADE-25 2015, Proceedings*, pages 399–415, 2015.
10. Giles Reger, Martin Suda, and Andrei Voronkov. Finding finite models in multi-sorted first order logic. In *SAT 2016, Proceedings*, 2016.
11. Andrei Voronkov. AVATAR: the architecture for first-order theorem provers. In *CAV 2014. Proceedings*, pages 696–710, 2014.