Vampire 4.2-SMT System Description

Giles Reger[1], Martin Suda[2], Andrei Voronkov[1,2,4], Evgeny Kotelnikov[3], Simon Robillard[3], and Laura Kovács[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 2017.

General Approach. Vampire [7] is an automatic theorem prover for first-order logic. Vampire implements the calculi of ordered binary resolution and superposition for handling equality. It also implements the Inst-gen calculus [3] and a MACE-style finite model builder [11]. Splitting in resolution-based proof search is controlled by the AVATAR architecture [10, 15]. Both resolution and instantiation based proof search make use of global subsumption [3]. 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 resolution and superposition calculi.

A number of standard redundancy criteria and simplification techniques are used for pruning the search space. The reduction ordering is the Knuth-Bendix Ordering. Internally, Vampire works only with clausal normal form. Problems are clausified during preprocessing [12]. Vampire implements many useful preprocessing transformations including the Sine axiom selection algorithm [2]. Vampire is a parallel portfolio solver, executing a schedule of complementary strategies in parallel.

Theory Reasoning. Vampire supports all logics apart from bit vectors. This is thanks to recent support for a first-class boolean sort [5], arrays [4], and datatypes [6], which 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 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 (see [9] for a discussion).
2. AVATAR modulo theories [8] which incorporates Z3 [1] into AVATAR (in this sense Vampire is a wrapper solver). 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.
3. As described in [14, 13], Vampire combines new approaches to unification and instantiation with the aim of leveraging an SMT solver (Z3) for reasoning within a clause.

Additionally, Vampire incorporates a MACE-style finite-model finding method that operates on multi-sorted problems [11]. There are only two cases where Vampire can return sat: Firstly in UF and secondly, if Vampire produces a ground problem after preprocessing it may pass this problem to Z3 and report its result (possibly sat) directly.

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. Last year Vampire won the AUFNIRA, NRA, and UFNIA tracks, and was the best competitive solver (behind Z3) in the AUFLIRA and UFLRA tracks. Vampire also performed well on unknown benchmarks, which are included this year. We hope to perform similarly or better this year. For the new datatype benchmarks we hope to be competitive. 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] Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3: an efficient SMT solver. In *Proc. of TACAS*, volume 4963 of *LNCS*, pages 337–340, 2008.

[2] 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.

[3] 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.

[4] Evgenii Kotelnikov, Laura Kovács, Giles Reger, and Andrei Voronkov. The vampire and the FOOL. In *ACM SIGPLAN 2016 proceedings*, pages 37–48, 2016.

[5] 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.

[6] Laura Kovács, Simon Robillard, and Andrei Voronkov. Coming to terms with quantified reasoning. In Giuseppe Castagna and Andrew D. Gordon, editors, *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017*, pages 260–270. ACM, 2017.

[7] Laura Kovács and Andrei Voronkov. First-order theorem proving and Vampire. In *CAV 2013. Proceedings*, pages 1–35, 2013.

[8] Giles Reger, Nikolaj Bjorner, Martin Suda, and Andrei Voronkov. AVATAR modulo theories. In Christoph Benzmüller, Geoff Sutcliffe, and Raul Rojas, editors, *GCAI 2016. 2nd Global Conference on Artificial Intelligence*, volume 41 of *EPiC Series in Computing*, pages 39–52. EasyChair, 2016.

[9] Giles Reger and Martin Suda. Set of support for theory reasoning. In Thomas Eiter, David Sands, Geoff Sutcliffe, and Andrei Voronkov, editors, *IWIL Workshop and LPAR Short Presentations*, volume 1 of *Kalpa Publications in Computing*, pages 124–134. EasyChair, 2017.

[10] Giles Reger, Martin Suda, and Andrei Voronkov. Playing with AVATAR. In *CADE-25 2015, Proceedings*, pages 399–415, 2015.

[11] Giles Reger, Martin Suda, and Andrei Voronkov. Finding finite models in multi-sorted first order logic. In *SAT 2016, Proceedings*, 2016.

[12] Giles Reger, Martin Suda, and Andrei Voronkov. New techniques in clausal form generation. In Christoph Benzmuller, Geoff Sutcliffe, and Raul Rojas, editors, *GCAI 2016. 2nd Global Conference on Artificial Intelligence*, volume 41 of *EPiC Series in Computing*, pages 11–23. EasyChair, 2016.

[13] Giles Reger, Martin Suda, and Andrei Voronkov. Instantiation and pretending to be an smt solver with vampire. In *SMT Workshop 2017*, 2017.

[14] Giles Reger, Martin Suda, and Andrei Voronkov. Unification with abstraction and theory instantiation in saturation-based reasoning. In *IWIL Workshop and LPAR Short Presentations*, volume 1 of *Kalpa Publications in Computing*. EasyChair, 2017.

[15] Andrei Voronkov. AVATAR: the architecture for first-order theorem provers. In *CAV 2014. Proceedings*, pages 696–710, 2014.