

Playing with AVATAR

Giles Reger, Martin Suda, Andrei Voronkov

University of Manchester

Abstract

This paper presents and explores experimental results examining the usage of AVATAR (Advanced Vampire Architecture for Theories and Resolution) [3]. Various options and extensions are investigated within the context of problems taken from the CASC competition (<http://www.cs.miami.edu/tptp/CASC/24/>).

AVATAR is a new architecture for first-order resolution and superposition theorem provers that places a SAT (or SMT) solver at the centre of the theorem prover to direct exploration of the search space. This architecture was introduced as a new technique for *splitting* clauses. Splitting addresses the issue of a rapidly growing search-space containing multi-literal and heavy clauses by splitting the search space $S \cup \{C_1 \vee C_2\}$ into $S \cup \{C_1\}$ and $S \cup \{C_2\}$ for variable disjoint C_1 and C_2 .

The AVATAR framework carries out splitting in the following way. If a new clause can be split into components with disjoint sets of variables then a SAT clause is constructed by replacing each component (consistently) by a propositional literal. A (partial) model of these SAT clauses is then used to select which components should be asserted in the first-order prover. Clauses derived from these asserted clauses keep track of this dependency and then if the empty clause is produced with dependencies this information is added as a clause and the model recomputed. At any point a previously asserted clause may be unasserted and it, and any clauses produced from it, should be (temporarily) removed. Therefore, the relation between first-order solver and SAT solver is complex and needs to be explored experimentally.

Previous work [2] explored various options for splitting (leading to 481 different strategies), aiming to understand how different options affected performance of the prover. These results were partially extended for AVATAR in [3] but some questions remain unanswered. The aim of this paper is to extend the experimental analysis of splitting in AVATAR.

This paper examines the effects on efficiency of 1) adding different components to the SAT solver, 2) minimising the produced model, and 3) varying how the SAT solver constructs a model. In the first case, we can choose to add information about nonsplittable clauses to the SAT solver. If these clauses have dependencies, or may become components later, this may give the SAT solver more useful information when constructing a model. Additionally we add a feature that delays adding clauses to the SAT solver if they can be made trivially true i.e. contain a new variable. In the second case, we can minimise the model in various ways i.e. construct a model only of the split clauses, or a minimal model of all clauses. There is then the decision of whether to eagerly remove asserted clauses that are no longer in a partial model. In the final case, we consider two SAT solvers – a home-grown solver and lingeling [1].

The full paper will be available after the workshop.

References

- [1] A. Biere. Lingeling, Plingeling and Treengeling Entering the SAT Competition 2013. In A. Balint, A. Belov, M. Heule, and M. Jrvialo, editors, *Proceedings of SAT Competition 2013*, 2013.
- [2] K. Hoder and A. Voronkov. The 481 ways to split a clause and deal with propositional variables. In M. P. Bonacina, editor, *Automated Deduction CADE-24*, volume 7898 of *Lecture Notes in Computer Science*, pages 450–464. Springer, 2013.
- [3] A. Voronkov. AVATAR: the architecture for first-order theorem provers. In *CAV 2014*, *To appear*.