AVATAR: a new Architecture for First-Order Theorem Provers

Giles Reger Martin Suda Andrei Voronkov

University of Manchester, Manchester, UK
{regerg,sudam,voronkov}@cs.man.ac.uk

Abstract: AVATAR is a new architecture for first-order resolution and superposition theorem provers which tightly integrates the saturation loop with a SAT solver (or an SMT solver) to efficiently implement the clause splitting rule. AVATAR employes the SAT solver to pick splitting branches, thus delegating the propositional essence of the given problem to the dedicated solver. This leads to a combination which is shown to be highly successful in practice. Moreover, replacing the SAT solver with an SMT solver opens up the possibility for efficient reasoning with both quantifiers and theories.

1 Introduction

Modern first-order resolution and superposition theorem provers use saturation algorithms to search for a refutation in clauses derivable from the input. On hard problems, this search space often grows rapidly and performance degrades especially fast when long clauses are generated. One approach that has proved successful in taming the search space is *splitting* where clauses are split into components with disjoint variables and the components are asserted in turn. This reduces the length of clauses in the search space at the cost of keeping track of splitting decisions.

Two main approaches to implementing splitting have been described in the literature. In splitting with backtracking (introduced in SPASS [5]) a conceptual splitting tree is traversed in parallel to the saturation process and one explicitly keeps track of how clauses depend on splitting decisions. Deriving a conditional contradiction means that the currently explored branch of the tree is inconsistent and a new candidate branch must be selected. Splitting without backtracking [3] (as seen in Vampire [2]), on the other hand, introduces new propositional variables to represent the splitting components and replaces a split clause by several shorter clauses while preserving satisfiability.

AVATAR (Advanced Vampire Architecture for Theories and Resolution) [4] combines ingredients from both previous approaches to splitting in a clever way with the key addition of employing an efficient SAT solver to make splitting decisions. We provide an overview of this new architecture (Sect. 2), report on experimental results which demonstrate its practical potential (Sect. 3), and outline areas of our current research on this topic (Sect. 4).

2 AVATAR overview

The AVATAR architecture (see Fig. 1) consist of three parts: the first order solver FO, the SAT solver, and the intermediate Splitting interface. The FO solver implements the standard saturation loop except that it deals with first order clauses *with assertions* (see below). Splitting interface splits clauses into components, maintains an injective mapping $[\_]$ abstracting clause components by propositional variables and communicates with the SAT solver. The SAT

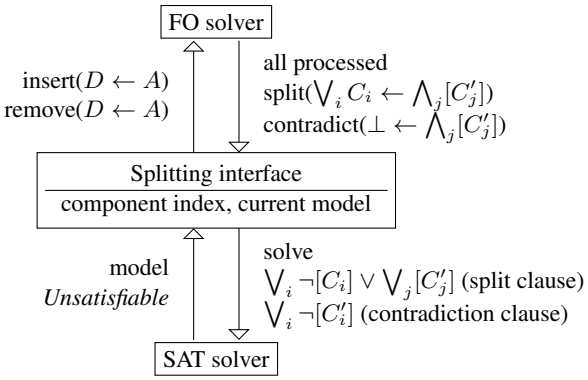

Figure 1: The AVATAR architecture.

solver receives information about split clauses and derived (conditional) contradictions and either produces a model, which defines a new branch of the conceptual split tree to explore, or reports *Unsatisfiable*, which means the original problem is unsatisfiable and the computation terminates.

When AVATAR encounters a clause C splittable into components: $C = C_1 \vee \ldots \vee C_n$, the clause is abstracted into the propositional *split clause* $[C_1] \vee \ldots \vee [C_n]$ and passed to the SAT solver. This ensures that at least one of the variables $[C_i]$ will be true in the subsequent model.

When $[C_i]$ is true in the model, a clause with assertion $C_i \leftarrow [C_i]$ is inserted into the FO solver. In general, a clause with assertions $D \leftarrow A$ consists of a first-order clause D and a conjunction of propositional variables A, the assertions. The FO solver performs inferences on the first-order parts as usual, but collects assertions such that the assertion of inference's conclusion equals the conjunction of assertions of the inference's premises.

Unless the given problem is satisfiable, the FO solver eventually derives a contradiction clause $\bot \leftarrow A$, which in general depends on some assertions $A = [C'_1] \wedge \ldots \wedge [C'_m]$. The Splitting interface turns this information into a propositional *contradiction clause* $\neg[C'_1] \vee \ldots \vee \neg[C'_m]$ and passes it to the SAT solver. Because the assertion A was true in the current model, this addition forces the SAT solver to compute a different model. This corresponds to advancing the search to a different branch in the conceptual split tree.

The above description of AVATAR omits some important details such as the maintenance of the component index, special treatment of ground components, and, most importantly, the handling of clause reductions and deletions by the FO solver which, similarly to splitting with backtracking, are in general *conditional* and may need to be undone when the current model changes. We refer the reader to the original paper [4] for a full account of these aspects.

3 Experimental evaluation

Our experiments [4] have shown that AVATAR shows outstanding results both in terms of its average performance and in the number of problems that it can solve and that were previously unsolvable by any existing prover.

Out of 6892 unsatisfiable problems from TPTP problem library the best splitting strategy from previous work [1] was able to solve 4381 problems while a strategy based on AVATAR solved 4716 problems. Moreover, Vampire with AVATAR was able to solve 421 problems unsolvable by Vampire without AVATAR and by any other prover. In comparison, Vampire using splitting with and without backtracking was able to solve only 17 problems unsolvable by any strategy using AVATAR. Solving over 400 previously unsolvable problems is a remarkable result since all these problems are very hard. In the past, the implementation of various novelties in Vampire would normally result in solving from a few to about 30 previously unsolvable problems.

The experimental results were so successful that all previously implemented code for handling splitting was completely removed from the latest versions of Vampire, resulting in considerable simplifications in its code and better maintainability.

4 Current and future work

Although AVATAR has already proved highly successful, its full power, and the best way to use it, is not yet fully understood. We, therefore, continue the research on AVATAR, currently focusing mainly on the following aspects.

Tuning the architecture There are several ways of configuring the basic architecture out of which none is obviously better than the others. An example of a configurable option is when exactly in the execution of a saturation algorithm to split a clause, which could as be a soon as the clause is generated or only just before the clause is selected to participate in inferences.

The effect of such options on prover performance is hard to predict, because it is typically connected with intrinsic tradeoffs and certain choices may help on some problems but not on others. We are currently investigating which option value combinations lead to the best performance in the number of solved problems, but also which option values are necessary to solve problems that cannot be solved otherwise.

Nice models Another factor which influences the performance of AVATAR is the quality of models produced by the SAT solver. Although it is not obvious what the best model looks like, it is clear that some models are better than others. For instance, introducing unnecessary assertions may lead to a slowdown in the FO solver without any compensating benefit in obtaining the ultimate contradiction.

We are currently experimenting with partial models and a heuristical minimisation procedure which eagerly removes literals not needed for satisfying the clauses in the SAT solver. Smaller models are expected to speed up the proving process provided the actual minimisation time does not become detrimental.

Theory reasoning As already explained in the original paper [4], the AVATAR architecture naturally allows for a combination of first order reasoning with theories when the SAT solver is replaced by an STM solver. Ground components composed of theory literals can then be made accessible to the SMT solver, allowing it to exclude theory-inconsistent models.

We have already obtained encouraging results for the theory of equality and uninterpreted functions. There is a plan to integrate other theories in the near future. We believe that AVATAR is an important step towards efficient reasoning with both quantifiers and theories, which is one of the key areas in modern applications of theorem provers in program analysis and verification.

References

[1] Krytof Hoder and Andrei Voronkov. The 481 ways to split a clause and deal with propositional variables. In Maria Paola Bonacina, editor, *Automated Deduction CADE-24*, volume 7898 of *Lecture Notes in Computer Science*, pages 450–464. Springer, 2013.

[2] Laura Kovács and Andrei Voronkov. First-order theorem proving and Vampire. In Natasha Sharygina and Helmut Veith, editors, *CAV 2013*, volume 8044 of *Lecture Notes in Computer Science*, pages 1–35, 2013.

[3] A. Riazanov and A. Voronkov. Splitting without backtracking. In B. Nebel, editor, *17th International Joint Conference on Artificial Intelligence, IJCAI'01*, volume 1, pages 611–617, 2001.

[4] Andrei Voronkov. AVATAR: The architecture for first-order theorem provers. In Armin Biere and Roderick Bloem, editors, *Computer Aided Verification*, volume 8559 of *Lecture Notes in Computer Science*, pages 696–710. Springer International Publishing, 2014.

[5] C. Weidenbach. Combining superposition, sorts and splitting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 27, pages 1965–2013. Elsevier Science, 2001.