Modular Termination and Combinability for Superposition Modulo Counter Arithmetic

Christophe Ringeissen and Valerio Senni

LORIA-INRIA Nancy Grand Est
FirstName.LastName@loria.fr

Abstract. Modularity is a highly desirable property in the development of satisfiability procedures. In this paper we are interested in using a dedicated superposition calculus to develop satisfiability procedures for (unions of) theories sharing counter arithmetic. In the first place, we are concerned with the termination of this calculus for theories representing data structures and their extensions. To this purpose, we prove a modularity result for termination which allows us to use our superposition calculus as a satisfiability procedure for combinations of data structures. In addition, we present a general combinability result that permits us to use our satisfiability procedures into a non-disjoint combination method à la Nelson-Oppen without loss of completeness. This latter result is useful whenever data structures are combined with theories for which superposition is not applicable, like theories of arithmetic.

1 Introduction

Software verification tasks require the availability of solvers that are able to discharge proof obligations involving data-structures together with arithmetic constraints and other mathematical abstractions, such as size abstractions. Besides, the use of Satisfiability Modulo Theories (SMT) solvers allows us to focus on the development of satisfiability procedures for such mixed theories. In this setting, the problem of designing the satisfiability procedures is often addressed with success by using approaches based on combination [15].

Problems arise when we consider combinations involving theories whose signatures are non-disjoint. This is especially the case when we consider theories sharing some algebraic constraints [14,16,17,18,20,21]. In order to combine satisfiability procedures for the single theories to handle constraints in their non-disjoint union one needs to rely on powerful methods such as the combination framework of [9,10]. These methods are based on semantic properties of the considered theories, such as compatibility and computability of bases of the shared entailed equalities, which often require complex proofs.

A further issue concerns the development of correct and efficient satisfiability procedures for the single theories, possibly using a systematic approach. In this regard, the use of superposition calculus has proved to be effective

* The author acknowledges support from ERCIM during his stay at LORIA-INRIA.
to deal with classical data structures, which do not involve arithmetic constraints [1,2,4,5,6,13].

In this paper we address both aspects by: (1) considering a superposition calculus with a built-in theory of counter arithmetic [17,18] and (2) providing modularity results for termination and combinability, based on conditions on the saturations of the component theories that can be checked automatically.

Our contributions are twofold. First, we prove a modular termination result for extending the applicability of the superposition calculus to theories that share a theory of counter arithmetic. This generalizes, to the non-disjoint case, the results in [1], where the authors consider the standard superposition calculus and signature-disjoint theories. This result allows us to drop some of the complex conditions required by the combination framework when we deal with theories that can be treated uniformly through superposition.

Second, we prove a general compatibility result that allows us to use our superposition-based satisfiability procedures into the combination framework of [10]. We prove that any satisfiability procedure obtained by using our modular termination result is able to compute a finite basis of the shared entailed equalities. In addition, we provide a sufficient condition on the form of the saturations of the theories that allows us to conclude compatibility of the component theories with respect to the shared theory and, thus, completeness of their combination.

As an outcome, we have less and simpler restrictions on combinability and we are able to obtain satisfiability procedures both by a uniform approach for theories that can be treated well by superposition (e.g., data structures) and by combination with other solvers for theories which are not ‘superposition-friendly’ (such as theories of arithmetic).

To show the application of our results in practice, we introduce a class of new theories modeling data structures and equipped with a counting operator that allows us to keep track of the number of the modifications (writes, constructors, etc.) performed on a data structure. In these theories we are able to distinguish between versions of the same data structure obtained by some update.

The paper is organized as follows. In Section 2 we briefly introduce an example in which we use data structures equipped with a mechanism to count the update operations. In Section 3 we introduce some basic notions and recall the superposition calculus for counter arithmetic. In Section 4 we present our modular termination result. In Section 5 we present our general compatibility result. In Section 6 we discuss in details how these results can be applied to our motivating example. Section 7 concludes with some perspectives.

2 A Motivating Example

Let us now consider an example where we show the application of our technique to the analysis of a function \textit{minmax}, defined as follows:

\begin{verbatim}
function minmax (l : LIST) : RECORD {
    while (l != nil) {
        e := car(l);
    }
}
\end{verbatim}