Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 240 (2009) 113–128
www.elsevier.com/locate/entcs

Combining Decision Procedures by (Model-)Equality Propagation

Diego Caminha B. de Oliveira[b,1,2] David Déharbe[a,1,3]
Pascal Fontaine[b,1,4]

[a] *Departamento de Informática e Matemática Aplicada*
Universidade Federal do Rio Grande do Norte
Natal, RN, Brazil

[b] *LORIA - INRIA*
Université de Nancy
Nancy, France

Abstract

SMT (Satisfiability Modulo Theories) solvers are automatic verification engines suitable to discharge important classes of proof obligations generated in applying formal construction of software and hardware designs. In this paper, we present a new approach to combine decision procedures and propositional solvers into an SMT-solver. This approach is based on the generation of model equalities by decision procedures. We show the soundness and completeness of the proposed approach using an original abstract framework to represent and reason about SMT-solvers. We then present an algorithmic version of the new SMT-solving approach and discuss practical aspects of our implementation.

Keywords: automatic theorem proving, SMT solvers, combination of decision procedures

1 Introduction

The application of formal methods to the design of computing systems often results in the generation of verification conditions that need to be proved in order to guarantee the correctness of the result. Such verification conditions express properties of models or relations between models and may be expressed in a wide range of logics: from propositional to high order logic, but also process algebra and temporal logic. Hence the level of automation for verification in a specific formalism is tightly dependent on the availability of tools to support reasoning in such logics.

[1] The research presented in this paper has been partially financed by CNPq/INRIA project Da Capo and CNPq project Nº 307597/2006-7.

[2] Email:diego.caminha@loria.fr

[3] Email:david@dimap.ufrn.br

[4] Email:pascal.fontaine@loria.fr

doi:10.1016/j.entcs.2009.05.048

The work described in this paper addresses the verification of satisfiability modulo theories (SMT) of quantifier-free formulas, i.e. verification conditions expressed in a first-order logic using symbols from a combination of theories, such as uninterpreted functions, fragments of integer or real arithmetics, set and array theories, etc. This applies to a number of verification applications, e.g. the application of formal program transformations such as refinement [15] or refactoring laws [6], verification of refinement properties in *posit-and-prove* software engineering efforts [1,2], or static analysis of annotations in design-by-contract languages [14]. Even verification efforts in more expressive logics often require proving lemmas that may be tackled by SMT-solvers (see for instance [4]).

SMT-solvers can for example handle a formula like

$$x \leq y \wedge y \leq x + f(x) \wedge P(h(x) - h(y)) \wedge \neg P(0) \wedge f(x) = 0 \tag{1}$$

which contains linear arithmetics on reals $(0, +, -, \leq)$, and uninterpreted symbols (P, h, f). SMT-solvers use decision procedures for the disjoint languages (for instance, congruence closure for uninterpreted symbols [17], and simplex for linear arithmetics) and combine them to build a decision procedure for the union of the languages. The combination of decision procedures works either through some guessing, or through the exchange of information between the decision procedures. In the general case, the information exchanged between the decision procedures is a set of disjunctions of equalities, and handling them requires often complex and costly case splitting. In the special case of convex theories, exchanging only equalities (and not disjunctions) is enough to ensure the completeness of the combination. We here show that even in the general case (i.e. with non-convex theories) exchanging equalities is also sufficient for completeness thanks to the cooperation with the propositional reasoning engine of the SMT-solver.

The next section introduces notations and the basics of SMT-solvers. In Section 3 we present an abstract framework for describing SMT-solvers. It only serves to discuss the soundness and completeness of the combination framework we describe in this paper. It is not as detailed as the DPLL($\mathcal{T}$) framework [18] since it is not meant to be a precise description of solvers. By contrast to the DPLL($\mathcal{T}$) framework and for simplicity, our schema highlights the distinction between the Boolean reasoning and the theory reasoning. It is not difficult to understand DPLL($\mathcal{T}$) as being a refinement of our schema.

Section 4 uses the framework introduced in Section 3 to discuss the soundness and completeness of various approaches to SMT solving. In particular, we present a new approach that consists in only exchanging equalities: either those equalities are deduced by the decision procedures, or they are assumed by generalising models. The approach is suitable for any decision procedure capable of finding models; many decision procedures inherently have this capability. A concrete algorithm using this approach is presented in Section 5. This algorithm is a simplification of our implementation. A concrete example is discussed in Section 6.