Grand Challenges for Automated Reasoning Position Statement

Cesare Tinelli[1]

Department of Computer Science
The University of Iowa
USA
`tinelli@cs.uiowa.edu`

It has been argued in many places and by several researchers that the key to scaling Automated Reasoning to real world applications will lie into the effective integration of special-purpose reasoners into general-purpose reasoning framework. In this respect, I think that one of the grand challenges for the field in the next years will be:

> *efficiently integrating, without loss of completeness, ground satisfiability procedures into first-order reasoning systems.*

By ground satisfiability procedure I mean the following. Given a logical theory T, a ground satisfiability procedure for T is a procedure that decides the satisfiability in T of conjunctions of literals. More precisely, it is a procedure that, for all conjunctions φ of literals in the signature of T, is always able to determine whether there is a model of T that satisfies (the existential closure of) φ. Ground satisfiability procedures have been discovered for many theories of interests in computer science. Moreover, several of them are also quite efficient in practice, if not not in theory.

By integrating a ground satisfiability procedure for a theory T into first-order reasoning system, I mean modularly incorporating the procedure into a more general mechanism, such as for instance a first-order calculus, so as to obtain a system for the satisfiability in T of arbitrary first-order formulas. Because of basic undecidability results, we know that such a system will not be terminating in general. The challenge however requires that it be complete, that is, able to recognize all first-order formulas unsatisfiable in T.

In addition to *completeness*, the key requirements in this challenge are the time/space *efficiency* of the integrated system and the use of *ground satisfiability* procedures, as opposed to non-ground satisfiability procedures. The challenge lies exactly in having all of these desiderata together, because without one or more of them the challenge can be considered as already met. In the following I argue why this is the case, and why it is important to achieve all of the challenge's requirements.

Efficiency. I do realize that efficiency is not an objective requirement unless it is defined in quantitative terms. Therefore some reasonable and measurable notion of efficiency must be discussed and established before this challenge can be even considered. Assuming however that this has been done and appealing to our intuitive appreciation of what an efficient system should be, let me comment on why efficiency is part of the challenge.

If one does not care about efficiency, the challenge is easily achievable by ground instantiation, at least for "Herbrand theories".[1] Given a formula to be checked for satisfiability, one can Skolemize it, enumerate its ground instances, and check the satisfiability of each of them. Note that this requires the ground satisfiability procedure for T to accept formulas containing arbitrary Skolem symbols in addition to the symbols of T. Recent results, however, show that any ground satisfiability procedure can be modularly extended to do that.

Since improving efficiency of automated reasoning is the very reason of using special-purpose reasoners like ground satisfiability procedures in the first place, this is a requirement that we certainly cannot lift.

Completeness. At least at the theoretical level, forgoing completeness almost trivializes the challenge. In fact, using past results on so called *theory reasoning* calculi, a ground satisfiability procedure for a theory T can be easily integrated into a general first-order calculus to obtain a sound and complete system for the satisfiability in T of quantifier-free formulas. The integrated system, however, cannot deal with (universally) quantified formulas, and as a consequence it is incomplete with respect to the class of all first-order formulas.

The difficulty lies exactly in dealing with quantifiers in a complete way. Theory reasoning calculi are in fact complete, but at the cost of requiring more than a mere ground satisfiability procedure (see later). Some existing theorem provers embedding ground satisfiability procedures[2] do provide a limited treatment quantifiers. They use simple but incomplete heuristics to eliminate quantifiers by guessing "right" instantiations. The challenge would be in going past these heuristics and devise complete ways to deal with quantifiers while at the same time using decision procedure that accept only quantifier-free formulas.

Ground satisfiability. In a sense, having decision procedures for the satisfiability of sets of literals is not so much a desideratum as a necessity. Designing and implementing efficient decision procedures for *non-ground* satisfiability in a theory T is extremely hard, if possible at all. There is a quite large and ever increasing number of efficient ground satisfiability procedures in the literature, and for more and more theories of practical interest in AR applications. On the other hand, one is hard pressed to find efficient decision (or even semi-decision) procedures for non-ground satisfiability. This is possibly the leading reason the theory reasoning paradigm has not fulfilled so far its promise of improving the efficiency of automated reasoning. The basic research in theory reasoning has succeeded in integrating in a complete

---

[1] By Herbrand theory I mean a theory T such that a formula is satisfiable in T iff it is satisfiable in a term-generated model of T. For instance all, but not only, the universal theories are Herbrand in this sense.

[2] Step, Simplify, and PVS come to mind.

way theory-specific *background reasoners* into general-purpose calculi. But it has done so by postulating the use of background reasoners that answer more general satisfiability problems than those decided by ground satisfiability procedures. In basically all theory reasoning calculi, the background reasoner for a theory T is given a conjunction φ of literals and is asked for a *refuter*, a substitution σ that makes the existential closure of $\varphi\sigma$ unsatisfiable in T. Implementing efficiently background reasoners of this sort has proven almost impossible except for very simple background theories. The implementation task is a lot simpler and actually feasible, as proven by the many systems already in circulation, if the background reasoner is asked only to say whether the existential closure of φ itself is satisfiable in the theory. At an abstract level then, we could say that the challenge consists in moving the computation of refuters from the background reasoner into the main calculus, while still achieving completeness overall.