

CP meets SMT

François Bobot, Sébastien Bardin, Bruno Marre

CEA, LIST, Software Reliability Laboratory, PC 174, 91191 Gif-sur-Yvette France

The field of automatic reasoning evolved in many different directions, from the development of very powerful higher-order logics aiming at the mechanization of mathematics, to the design of efficient decision procedures tailored to propositional logic (SAT DPLL [3]). Between these two extremes, CP and SMT are interested in efficient decision procedures for particular first-order theories with more or less restricted forms of quantification.

Much attention has been devoted in the CP community to the usage of domains as a way to facilitate communication between theories. Moreover the precision propagators have been improved continuously along the years, in order to reduce the quantity of labelling needed, since in classical CP no information is reused between different labelling.

The SMT community, with DPLL(X) [10, 5] and Nelson-Oppen [9] framework, devoted to combine existing efficient decision procedure for different theories, by defining rigorous interfaces. The communication between theories is quite poor and not theory specific. However the search engine of an SMT solver is based on a SAT-solver with all the techniques elaborated in this area. For instance, SMT solvers are able to learn from conflicts facts that help solving other branches, using the CDCL method [12].

During recent years, SMT solvers keep growing with the addition of new decision procedures. Some have proposed to adapt the CDCL approach to theories beyond propositional logic [2]. De Moura and Jovanović [8] revisited a complete but impractical technique for solving nonlinear problems called CAD; they turned it into a usable decision procedure using the CDCL approach for focusing the search and reducing the amount of computations. New decision procedures on floating-point numbers also used this approach [6]. In the meantime, a proposal extended the use of CDCL to the whole SMT framework and called it Model-Constructing Satisfiability (MCSat) calculus [4, 7]. This approach, however, keeps a strong link with propositional logic: it is still based on CNF formulas and keeps the domains inside the theories. For its part, the CP community extended the use of global constraints which can be considered as theories [1]. But these works have no mechanism of learning, or are restricted in a way similar to DPLL(T) [11].

In this presentation we will summarize the limitation of traditional SMT and CP solvers. Then we will present our work in designing a framework that gather an event driven engine from CP for propagation and a computing conflicts engine for learning. The framework uses domains and specific language conflicts as first-class citizen. Defining domains and conflicts as first-class citizen allows one theories to work with domain and conflicts of another theory in a formalized way. These concepts will be instantiated on some theories like linear arithmetic and an implementation will be presented.

References

- [1] Nicolas Beldiceanu. Global constraints as graph properties on a structured network of elementary constraints of the same type. In Rina Dechter, editor, *CP 2000*, volume 1894 of *LNCS*, pages 52–66. Springer Berlin Heidelberg, 2000.
- [2] Scott Cotton. Natural domain SMT: a preliminary assessment. In *Formal Modeling and Analysis of Timed Systems*. 2010.
- [3] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.
- [4] Leonardo de Moura and Dejan Jovanović. A model-constructing satisfiability calculus. In *VMCAI 2013*.
- [5] Harald Ganzinger, George Hagen, Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. DPLL(T): fast decision procedures. In Rajeev Alur and Doron A. Peled, editors, *Computer Aided Verification*, number 3114 in *LNCS*, pages 175–188. Springer Berlin Heidelberg, January 2004.
- [6] Leopold Haller, Alberto Griggio, Martin Brain, and Daniel Kroening. Deciding floating-point logic with systematic abstraction. In *FMCAD*, 2012.
- [7] Dejan Jovanović, Clark Barrett, and Leonardo de Moura. The design and implementation of the model constructing satisfiability calculus. In *Proceedings of 13th International Conference on Formal Methods in Computer-Aided Design, FMCAD 2013*, Portland, Oregon, USA, 2013.
- [8] Dejan Jovanović and Leonardo de Moura. Solving non-linear arithmetic. In *IJCAR 2012*.
- [9] Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 1(2):245–257, 1979.
- [10] Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. Solving SAT and SAT modulo theories: From an abstract Davis–Putnam–Logemann–Loveland procedure to DPLL(T). *J. ACM*, 53(6):937–977, November 2006.
- [11] Olga Ohrimenko, Peter J. Stuckey, and Michael Codish. Propagation via lazy clause generation. *Constraints*, 14(3):357–391, September 2009.
- [12] Lintao Zhang, Conor F Madigan, Matthew H Moskewicz, and Sharad Malik. Efficient conflict driven learning in a boolean satisfiability solver. In *Proceedings of the 2001 IEEE/ACM international conference on Computer-aided design*, pages 279–285. IEEE Press, 2001.