The Ctrl-Ergo SMT Solver

Mohamed Iguernlala

OCamlPro SAS, Gif-sur-Yvette F-91190
LRI, Université Paris-Saclay, Orsay F-91405

Ctrl-Ergo is a toy solver developed to implement and experiment new ideas before eventually incorporating them in the Alt-Ergo SMT solver[3]. It has originally been written to validate the results of our Simplex-based extension of Fourier-Motzkin for deciding quantifiers-free linear integer arithmetic formulas[2]. Recently, we extended its SAT solver engine with some optimization features and added quantifiers-free linear rational arithmetic support. More precisely:

1. Ctrl-Ergo's SAT solver is basically a re-implementation of MiniSAT in OCaml. We improved this engine with models reduction techniques based on Tableaux method[4] to allow a more efficient interaction with theories reasoners,

2. For quantifiers-free linear integer arithmetic, we implemented a pre-processor to simplify `If-Then-Else` constructs that are heavily used in some QF_LIA benchmarks. For the theory reasoner part, we use an integer Diophantine solver to turn equations into a rewriting system and a Simplex algorithm on auxiliary optimization problems to efficiently emulate Fourier-Motzkin in order to deduce better bounds for inequations' affine forms[2]. Although it is not incremental, our Simplex-based algorithm is quite efficient on a certain class of QF_LIA problems. An incremental re-formulation has been proposed by Bromberger et al. in their "*Cube Test*" approach[1],

3. To decide quantifiers-free linear rational arithmetic, we use a variant of the Simplex algorithm that works on problems put in general form[5]. We also integrate a small engine based on forward and backward propagation techniques found in Constraints Programming for theory lemmas deduction.

Ctrl-Ergo is today made of approximately 10,000 lines of OCaml code. The sources are available here: https://gitlab.com/OCamlPro-Iguernlala/Ctrl-Ergo. Its optimized SAT solver has already been integrated in recent versions Alt-Ergo. Some further implementation details are provided in my PhD thesis[6].

References

[1] Martin Bromberger and Christoph Weidenbach. New techniques for linear arithmetic: cubes and equalities. *Formal Methods in System Design*, 51(3):433–461, Dec 2017.

[2] François Bobot, Sylvain Conchon, Evelyne Contejean, Mohamed Iguernelala, Assia Mahboubi, Alain Mebsout, and Guillaume Melquiond. A Simplex-Based Extension of Fourier-Motzkin for Solving Linear Integer Arithmetic. In Bernhard Gramlich, Dale Miller, and Ulrike Sattler, editors, *IJCAR 2012 : Proceedings of the 6th International Joint Conference on Automated Reasoning*, Lecture Notes in Computer Science, Manchester, UK, June 2012. Springer Verlag.

[3] Sylvain Conchon, Albin Coquereau, Mohamed Iguernlala, and Alain Mebsout. Alt-Ergo 2.2. In *Proceedings of the 16th International Workshop on Satisfiability Modulo Theories, SMT 2018*, Oxford, UK, 2018.

[4] Leonardo de Moura and Nikolaj BjÃ¸rner. Relevancy propagation. 05 2019.

[5] Leonardo Mendonça de Moura, Bruno Dutertre, and Natarajan Shankar. A tutorial on satisfiability modulo theories. In *Computer Aided Verification, 19th International Conference, CAV 2007, Berlin, Germany, July 3-7, 2007, Proceedings*, pages 20–36, 2007.

[6] Mohamed Iguernelala. *Strengthening the heart of an SMT-solver : Design and implementation of efficient decision procedures. (Renforcement du noyau d'un démonstrateur SMT : Conception et implantation de procédures de décisions efficaces)*. PhD thesis, University of Paris-Sud, Orsay, France, 2013.