SMT-RAT [2] is an open-source C++ toolbox for strategic and parallel SMT solving consisting of a collection of SMT compliant implementations of methods for solving quantifier-free first-order formulas with a focus on nonlinear real and integer arithmetic. Further supported theories include linear real and integer arithmetic, difference logic, bit-vectors and pseudo-Boolean constraints. A more detailed description of SMT-RAT can be found at [https://github.com/smtrat/smtrat/wiki](https://github.com/smtrat/smtrat/wiki). There will be two versions of SMT-RAT that employ different strategies that we call SMT-RAT and SMT-RAT-MCSAT.

SMT-RAT focuses on nonlinear arithmetic. As core theory solving modules, it employs interval constraint propagation (ICP) as presented in [6], virtual substitution (VS) [1] and the cylindrical algebraic decomposition (CAD) [10]. For ICP, lifting splitting decisions and contraction lemmas to the SAT solving and aided by the other approaches for nonlinear constraints in case it cannot determine whether a box contains a solution or not. For nonlinear integer problems, we employ bit blasting up to some fixed number of bits [9] and use branch-and-bound [8] afterwards. The SAT solving takes place in an adaption of the SAT solver minisat [5] and we use it for SMT solving in a less-lazy fashion [12].

For linear inputs we use the Simplex method equipped with branch-and-bound and cutting-plane procedures as presented in [4]. Furthermore, we apply several preprocessing techniques, e.g., using factorizations to simplify constraints, applying substitutions gained by constraints being equations or breaking symmetries. We also normalize and simplify formulas if it is obvious.

SMT-RAT-MCSAT uses our implementation of the MCSAT framework [3] that is still being worked on. It is equipped with multiple explanation backends based on the following: NLSAT-style CAD-based; Fourier-Motzkin variable elimination; Virtual substitution as in [13]; OneCell CAD as in [11]. The general MCSAT framework is integrated in our adapted minisat [5] solver, but is not particularly optimized yet. The latest addition has been making the variable ordering fully dynamic as suggested in [7].

Authors Erika Ábrahám, Gereon Kremer, Jasper Nalbach, Rebecca Haehn, Florian Corzilius, Sebastian Junges, Stefan Schupp.
References


