Boolector at the SMT Competition 2017

Aina Niemetz, Mathias Preiner, and Armin Biere

Institute for Formal Models and Verification
Johannes Kepler University, Linz, Austria

Abstract—This paper serves as solver description for our SMT solver Boolector as entered into the SMT Competition 2017. We only list important differences to the version of Boolector that entered the SMT Competition 2016 [5]. For further and more detailed information, we refer to [3,4,7] or source code.

Overview

For the SMT competition 2017, we improved several key components and features of our SMT solver Boolector [4]. In 2016, we added experimental support for quantified bit-vectors and at the SMT competition 2016, for the first time Boolector entered the BV and UFBV divisions of the main track. Since then, we considerably improved Boolector’s quantified bit-vector engine [7,8], which is now stable but currently does not support uninterpreted functions (UF). It enters the BV division of the main track of SMT-COMP’16 and will be included in the upcoming (and all future) official release(s) of Boolector.

In 2016, we also introduced an additional engine for quantifier-free bit-vectors, which implements a novel propagation-based local search approach [6] without bit-blasting. A combination of this approach with bit-blasting within a sequential portfolio setting as described in [3,6] entered SMT-COMP’16 as configuration Boolector preprop. For the SMT competition 2017, this combination is our default approach for the QF_BV division of the main track.

At the SMT competition 2017, additionally to the default configuration Boolector, Boolector enters the QF_BV main track in an experimental configuration Boolector+CaDiCaL, which uses version sc17 of our novel SAT solver CaDiCaL [2] as back-end. The default configuration Boolector, which also enters the division QF_ABV, QF_UFBV, QF_AUFBV and BV, uses an internal version (version bbe) of our SAT solver Lingeling [1], which is identical to the version submitted to the SAT competition 2017, as back-end.

Configurations

At the SMT competition 2017, we submitted two configurations of our SMT solver Boolector: Boolector and Boolector+CaDiCaL. For both configurations, we enabled unconstrained optimization.

Boolector

This configuration of Boolector enters the QF_BV, QF_UFBV, QF_ABV, QF_AUFBV, and BV divisions of the main track and uses our SAT solver Lingeling [1] (version bbe) as back-end.

Boolector+CaDiCaL

This configuration of Boolector enters only the QF_BV division of the main track and uses our novel SAT solver CaDiCaL [2] (version sc17) as back-end.

References