Smacc: An SMT Memory-Model and Assertion-Checker for C

Jakob Zwirchmayr, jakob@complang.tuwien.ac.at

April 12, 2010

Abstract

The thesis presents the tool SmacC, an approach to software verification and SMT benchmark generation building upon a new state-of-the-art SMT solver, Boolector, developed at FMV institute at JKU, Linz.

The program gets as input a C program that lies in the supported subset of the programming language (ANSI) C and transforms the program to SMT formulas. The SMT representation allows verification of properties that must hold on the program and the generation of SMT benchmarks by dumping the SMT instances.

Part of the goal of this work includes, on the software verification side, to check the input code for certain programming errors and additionally prove or disprove assertion statements in the code.

The other goal of the work was to generate SMT examples for SMT solvers that can be used either as (regression) tests for newly developed SMT solvers, or as benchmarks to compare the performance of different SMT solvers that support the underlying formats (BTOR or SMT-LIB) and theories (bit-vectors, arrays, equality of arrays).

To reach the goals, the tool symbolically executes the programs source code, establishing a (memory-) model for the program, represented as SMT formulas. Then the tool generates SMT formulas and lets the SMT solver decide if certain properties hold on the SMT representation of the program. If properties checked do not hold on the SMT representation, they do not hold on the real program.

Keywords: *SMT, Satisfiability Modulo Theories, Boolector, BTOR, Assertion Proving, Memory Model, Programming Language C, Symbolic Execution, Symbolic Simulation, Benchmarks.*