

NECLA Benchmarks: Linear and non-linear decision problems in SMTLIB format

Sicun Gao^{1,2}, Malay K. Ganai¹, Franjo Ivančić¹, Sriram Sankaranarayanan^{1,3}, Aarti Gupta¹

¹ NEC Laboratories America, Princeton, NJ

² Carnegie Mellon University, Pittsburgh, PA

³ University of Colorado, Boulder, CO

There are 4 sets of benchmarks, which are generated using our modeling of floating-point programs into a CFG representation with integer and real linear/non-arithmetic operations [2]. The modeling is geared towards analyzing numerical stability properties of such floating-point programs. Such models are analyzed using a BMC framework such as [1, 2, 3, 4]. Each benchmark set contains a number of decision problems corresponding to BMC instances at different depths.

Note that some benchmarks are generated from programs with floating-point computations that are based on discrete-time simulations of hybrid system models written as C programs. The generated model and corresponding benchmarks are conceptually close to prior similar benchmarks in the hybrid systems community. However, hybrid system models have a very compact representation, for example containing system dynamics and local invariants in each mode. Rewriting fixed step-size discrete-time executions of hybrid system models as C programs loses much of this compactness. Therefore, the benchmarks that are provided below are much larger than the corresponding original hybrid system benchmark. Furthermore, the “depth” of a discrete-time hybrid system simulation is measured in the number of discrete time-steps, whereas our models are based on unwinding of the control-flow graph representation of the C program increasing the depth required to reach a particular target. In the future, we may provide compact versions of these benchmarks that are based on a more compact representation of discrete-time hybrid system executions.

- Bouncing-ball-fromC: These benchmarks are based on an unrolling of a C program representation of a fixed time-step discrete-time simulation of a hybrid system representing a bouncing ball trying to reach a target hole. The generated model and corresponding benchmarks are conceptually close to prior similar benchmarks in the hybrid systems community. However, generating the benchmarks in our flow generates larger models than customary for this example.
- EtcS-fromC: These benchmarks are based on an unrolling of a C program representation of a fixed time-step discrete-time simulation of a hybrid system representing a European train control system (ETCS). The generated model and corresponding benchmarks are conceptually close to prior similar benchmarks in the hybrid systems community. However, generating the benchmarks in our flow generates larger models than customary for this example.
- addManyTimes: These benchmarks are based on an unrolling of a C program computing the sum of an array of unknown length of floating-point numbers in a particular range.
- multManyTimes: These benchmarks are based on an unrolling of a C program computing the product of an array of unknown length of floating-point numbers in a particular range

PLEASE DO NOT RE-DISTRIBUTE without permission from NEC Labs America.

Please contact Franjo Ivančić (ivancic@nec-labs.com) for any questions/comments.

References

1. S. Gao, M. K. Ganai, F. Ivančić, S. Sankaranarayanan, A. Gupta, and E. M. Clarke. Integrating ICP with DPLL(T) for Nonlinear Real Arithmetic. *Under submission*, 2010.
2. F. Ivančić, M. K. Ganai, S. Sankaranarayanan, and A. Gupta. Software Model Checking the Precision of Floating-Point Programs. *Under submission*, 2010.
3. M. K. Ganai and A. Gupta. Accelerating high-level bounded model checking. In *Proc. Of ICCAD*, 2006.
4. M. Franzle and C. Herde. HySAT: An efficient proof engine for bounded model checking of hybrid systems. In *Formal Method in System Design*, 2006.