ESBMC 1.24.1
(Competition Contribution)

Jeremy Morse, **Mikhail Ramalho**,
Lucas Cordeiro, Denis Nicole, Bernd Fischer

ESBMC: SMT-based BMC of single- and multi-threaded software

- exploits SMT solvers and their background theories:
  - optimized encodings for pointers, bit operations, unions and arithmetic over- and underflow
  - efficient search methods (non-chronological backtracking, conflict clauses learning)

ESBMC: SMT-based BMC of single- and multi-threaded software

- exploits SMT solvers and their background theories:
  - optimized encodings for pointers, bit operations, unions and arithmetic over- and underflow
  - efficient search methods (non-chronological backtracking, conflict clauses learning)
- supports verifying multi-threaded software that uses pthreads threading library
  - interleaves only at “visible” instructions
  - *lazy exploration* of the reachability tree
  - optional context-bound

ESBMC: SMT-based BMC of single- and multi-threaded software

- exploits SMT solvers and their background theories:
  - optimized encodings for pointers, bit operations, unions and arithmetic over- and underflow
  - efficient search methods (non-chronological backtracking, conflict clauses learning)
- supports verifying multi-threaded software that uses pthreads threading library
  - interleaves only at “visible” instructions
  - *lazy exploration* of the reachability tree
  - optional context-bound
- derived from CBMC (v2.9) and has inherited its object-based memory model

ESBMC verification support

- built-in properties:
  - arithmetic under- and overflow, pointer safety, array bounds, division by zero, memory leaks, atomicity and order violations, deadlocks, data races

ESBMC verification support

- built-in properties:
  - arithmetic under- and overflow, pointer safety, array bounds, division by zero, memory leaks, atomicity and order violations, deadlocks, data races
- user-specified assertions:
  - *(__ESBMC_assume, __ESBMC_assert)*
- built-in scheduling functions:
  - *(__ESBMC_atomic_begin, __ESBMC_atomic_end, __ESBMC_yield)*

ESBMC verification support

- built-in properties:
  - arithmetic under- and overflow, pointer safety, array bounds, division by zero, memory leaks, atomicity and order violations, deadlocks, data races
- user-specified assertions:
  - *(__ESBMC_assume, __ESBMC_assert)*
- built-in scheduling functions:
  - *(__ESBMC_atomic_begin, __ESBMC_atomic_end, __ESBMC_yield)*
- support for several C++ features: polymorphism, inheritance, exception handling, templates and STL (using models)

Differences to ESBMC 1.22

- ESBMC 1.24.1 is largely a **bugfixing release**, but also:
  - Improved new intermediate representation
    → increased ESBMC's speed by 2*x*
  - Support for boolector (>= 2.0), replaces Z3 as default solver
  - Descresead memory usage by ~23%.

- Several bug fixes on both sequential and parallel k-induction approach.

Results

- First place on BitVectors and Sequentialized

Results

- First place on BitVectors and Sequentialized

- Overall:
  - Second highest correct results (3898)
  - Highest number of false correct (318) and third higher false incorrect (122)
  - MemorySafety and Termination: fail to conform on the new report scheme

Thank you!

www.esbmc.org