To achieve **automatic bug detection** in **the development process of CUDA-based applications** use **ESBMC GPU** **an SMT-based context-BMC tool**

CUDA allows **parallelization** which makes bug inspection a difficult task

Problem

CUDA programs may contain bugs, *e.g.*, array out-of-bounds, arithmetic under/overflow, division by zero, failed user-specified assertions and specific concurrency errors related to data race and barrier divergence.

Apply the proposed solution to find bugs in CUDA-based applications

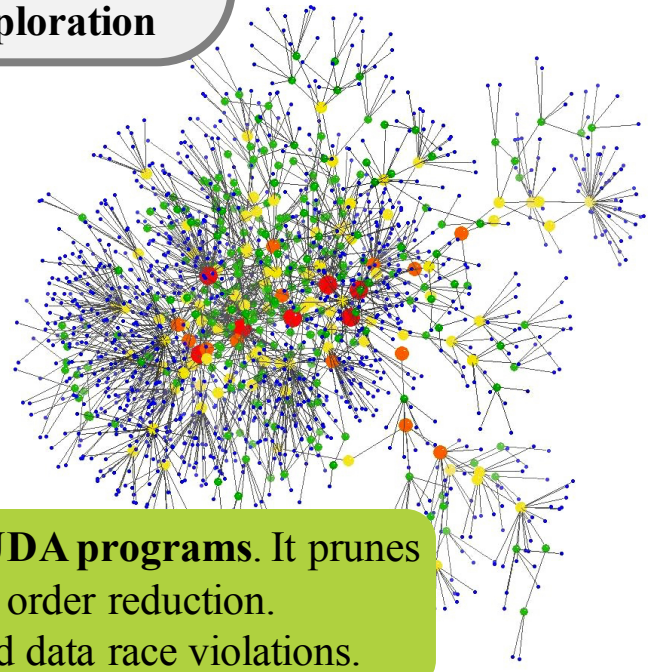
Solution

Combine **symbolic execution**, based on **bounded model checking** (BMC) and **satisfiability modulo theories** (SMT) techniques, with **explicit state-space exploration**

Explicitly explore the possible interleavings (up to the given context bound), while we treat each interleaving itself symbolically w.r.t. a given property (*e.g.*, data race)

GPUs are the core of some safety-critical applications (*e.g.*, autonomous cars), therefore, **correctness is of paramount importance**. However, certain kinds of bugs (*e.g.*, data race conditions) are hard to detect using standard GPU debuggers (*e.g.*, CUDA-GDB).

ESBMC-GPU was evaluated in 154 CUDA-based programs that explore a wide range of CUDA functionalities. **ESBMC-GPU was able to correctly verify 85% of the chosen benchmarks and it also overcame other existing GPU verifiers**.



ESBMC-GPU marks the first application of an SMT-based context-BMC tool for CUDA programs. It prunes space-state exploration through two-threads analysis, state hashing, and monotonic partial order reduction. Furthermore, ESBMC-GPU presents an improved ability to detect array out-of-bounds and data race violations.