Finding Critical Clauses in SMT-based Hardware Verification

Makai Mann, Clark Barrett

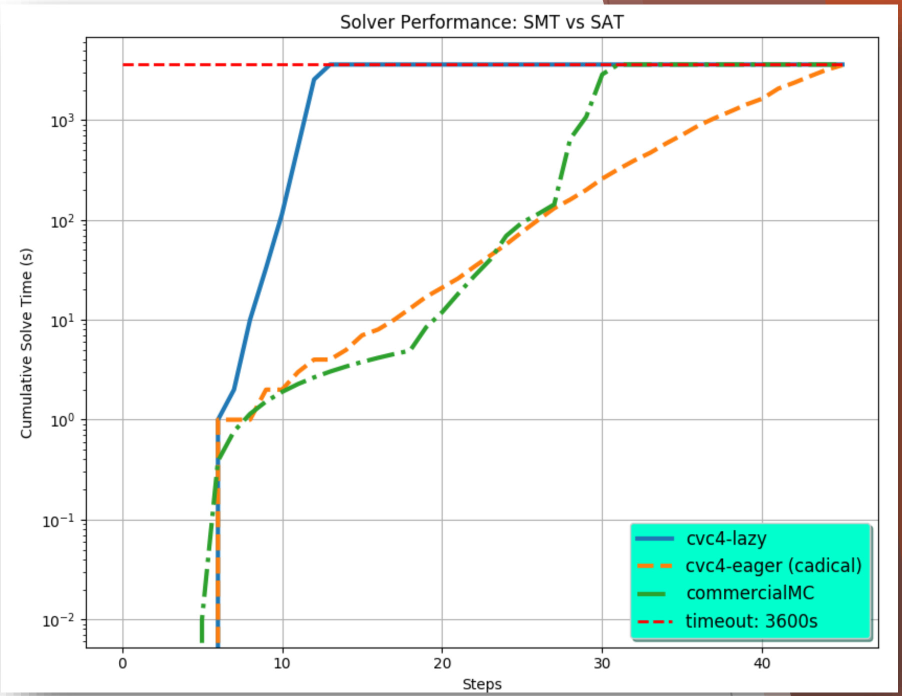
Stanford University

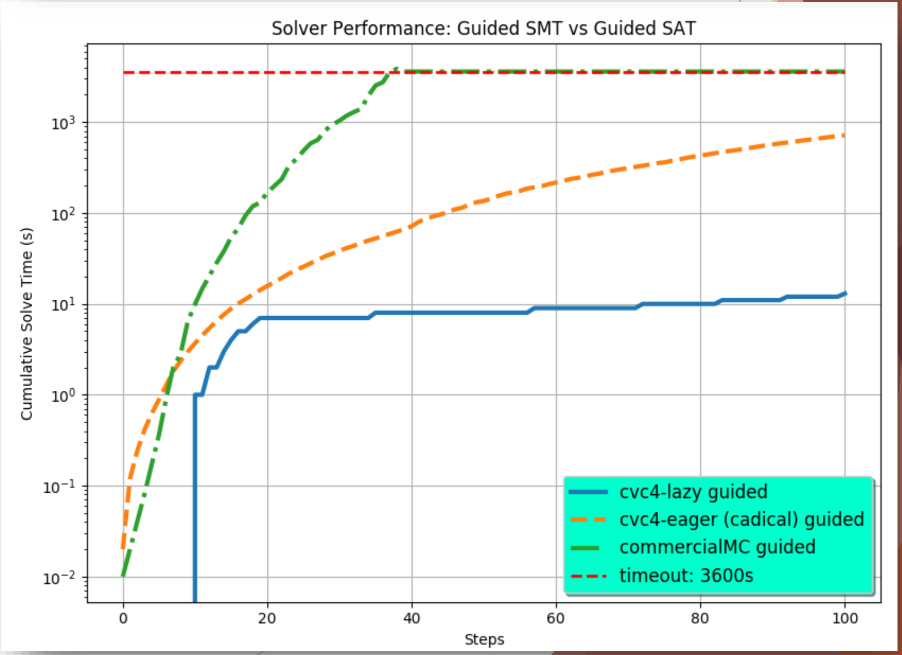
Hardware Verification

- SAT is king
  - Still faces scaling issues, particularly for data-path properties
- Satisfiability Modulo Theories (SMT) can reason at a higher level of abstraction
  - Lazy approaches usually not competitive with SAT (yet)
  - But there's hope

Evidence of Hope

- Checking data integrity of FIFO implementation
  - No packet is dropped
  - No packets are swapped
- Compare to SAT-based, unnamed, commercial model checker
- Helping both solvers
  - Lemmas
  - Encoding Tricks
  - Huge speed-up for lazy SMT





Three Approaches for Identifying Critical Clauses

Modular Techniques

- Identify invariants known at design-time
- Minimize inference solver has to do
- Particularly useful for transformations

Statistical Techniques

- "Offline" learning – learn from previous unroll in BMC
- "Online" learning – learn good splitting literals
- Early-stage research in SAT-based BMC, learning from resolution proofs

Transition Relation Techniques

- Clause lifting in BMC
- Reduce redundant path explorations
- Reachability algorithms
  - Using SMT
  - Guide SMT BMC

Thank you!

- Poster on Thursday