Separating Parallel and Functional Correctness
Jacob Burnim, George Necula, Koushik Sen

Overview
- Verifying parallel programs is very challenging.
  - Painful to reason simultaneously about correctness of parallelism and about functional correctness.
  - Functional correctness often largely sequential.
- Goal: Decompose effort of verifying parallelism and verifying functional correctness.
  - Prove parallel correctness simply – not entangled in complex sequential functional correctness.
  - Verify functional correctness in a sequential way.
- Question: What is parallel correctness?

Specifying Determinism
- Previous work: Deterministic specifications.
  - Burnim and Sen, FSE 2009
  - Idea: Parallel correctness means every thread schedule gives semantically equivalent results.
  - Internal nondeterminism, but deterministic output.
  - Assert that parallel code yields semantically equivalent outputs for equivalent inputs.

Our Approach
- For a parallel program, use a sequential but nondeterministic version as a specification.
  - User annotates intended algorithmic nondeterminism.
  - We interpret parallel constructs as nondeterministic and sequential.
- Parallelism is correct if it adds no unintended nondeterminism.
  - I.e., if parallel and nondeterministic sequential versions of the program are equivalent.

Proving Parallel Correctness
- Goal: Prove each execution of a parallel program is equivalent to a nondeterministic sequential (ndseq) execution.
- Added nondeterminism allows prune? (a) to be moved past update (b) without changing the program’s behavior.

Future Work
- Formal proof rules for parallel and nondeterministic sequential equivalence.
- Automated proofs of parallel correctness.
- Combine with verification tools for sequential programs with nondeterminism.
  - Model checking with predicate abstraction (CEGAR).
  - Can verify functional correctness on sequential code!
- Apply above to real parallel benchmarks.
- Applications to debugging?
  - Allow programmer to sequentially debug a parallel execution by mapping a parallel trace to a nondeterministic sequential one.