Proving a concurrent program correct by demonstrating it does nothing

Shaz Qadeer

Abstract:

CIVL is a verifier for concurrent programs implemented as a conservative extension to the Boogie verification system. CIVL allows the proof of correctness of a concurrent program—shared-memory or message-passing— to be described as a sequence of program layers. The safety of a layer implies the safety of the layer just below, thus allowing the safety of the highest layer to transitively imply the safety of the lowest.

The central theme in CIVL is reasoning about atomic actions. Different layers of a program describe its behavior using atomic actions, higher layers with coarse-grained and lower layers with fine-grained atomic actions. A proof of correctness amounts to a demonstration that the highest layer is SKIP, the atomic action that does nothing. The formal and automated verification justifying the connection between adjacent layers combines several techniques—linear variables, reduction based on movers, location invariants, and changing state representation via variable introduction and elimination.

CIVL is available in the master branch of Boogie together with more than fifty micro-benchmarks. CIVL has been used to refine a realistic concurrent garbage-collection algorithm from a simple high-level specification down to a highly-concurrent implementation described in terms of individual memory accesses.

Bio:

Shaz Qadeer is a Research Scientist at Facebook. He worked at Microsoft Azure and Microsoft Research before joining Facebook. His current work is focused on secure distributed systems. He is also interested in domain-specific programming languages, program verification, software testing, algorithms and decision procedures.