

Poirot—A Concurrency Sleuth

Shaz Qadeer

Microsoft Research
qadeer@microsoft.com

Abstract. Concurrent programming is difficult. The challenges are foundational: unlike sequential control flow, asynchronous control flow is difficult to understand and reason about. Not surprisingly, even expert programmers find it difficult to write concurrent software. We desperately need software engineering techniques and tools to move concurrent programming from black art to a rigorous engineering discipline. I believe that automated tools that reduce the cognitive burden of reasoning about concurrency can help tremendously in improving the productivity of concurrent programmers. In collaboration with my colleagues at Microsoft Research, I have developed Poirot (<http://research.microsoft.com/en-us/projects/poirot/>), a tool for answering semantic queries about a concurrent program by statically searching over its executions. Poirot exploits sequential encodings of concurrent semantics, structural under- and over-approximations for sequential control flow, and advances in automated theorem proving to search concurrent program executions efficiently. Poirot is being used in many different applications—bug detection, program understanding, and symbolic debugging. This lecture will present both a demonstration and an explanation of the techniques underlying the search engine inside Poirot.

Poirot is joint work with Akash Lal and Shuvendu Lahiri.