Reasoning about Program Data Structure Shape: from the Heap to Distributed Systems

Shape analysis is a static program-analysis technique that discovers and verifies properties of a program's dynamically allocated data structures. For example, shape analysis can infer that a linked list is acyclic, and prove that a program cannot free an element more than once. More generally, shape analysis provides a method to establish properties of systems whose states can be modeled as relations that evolve over time. A shape analyzer discovers quantified invariants of the elements of such systems.

In this talk, I will describe the road from analyzing dynamically allocated data structures to analyzing network protocols and other distributed systems. The survey includes both sound techniques and complete techniques. Some of these fundamental techniques inspired tools that are deployed in industry.

Biography

Mooly Sagiv is a full professor in the School of Computer Sciences at Tel-Aviv University. He is a leading researcher in the area of large scale (inter-procedural) program analysis, and one of the key contributors to shape analysis. His fields of interests include programming languages, compilers, abstract interpretation, profiling, pointer analysis, shape analysis, inter-procedural dataflow analysis, program slicing, and language-based programming environments. Sagiv is a recipient of a 2013 senior ERC research grant for Verifying and Synthesizing Software Composition. Prof. Sagiv served as Member of the Advisory Board of Panaya Inc. He received best-paper awards at PLDI’11 and PLDI’12 for his work on composing concurrent data structures and a ACM SIGSOFT Retrospective Impact Paper Award (2011) for program slicing.