

Thread-Modular Shape Analysis

Mooly Sagiv

Tel-Aviv University

Abstract. Thread-modular static analysis of concurrent systems abstracts away the correlations between the local variables (and program locations) of different threads. This idea reduces the exponential complexity due to thread interleaving and allows us to handle programs with an unbounded number of threads.

Thread-modular static analyses face a major problem in simultaneously requiring a separation of the reasoning done for each thread, for efficiency purposes, and capturing relevant interactions between threads, which is often crucial to verify properties. Programs that manipulate the heap complicate thread-modular analysis. Naively treating the heap as part of the global state, accessible by all threads, has several disadvantages since it still admits exponential blow-ups in the heap and is not precise enough to capture things like ownership transfers of heap objects. An effective thread-modular analysis needs to determine which parts of the heap are owned by which threads to obtain a suitable thread-modular state abstraction.

I will present new thread-modular analysis techniques and adaptations of thread-modular analysis for programs which manipulate the heap. It is shown that the precision of thread-modular analysis is improved by tracking some correlations between the local variables of different threads. I will also describe techniques for reducing the analysis time for common situations. A key observation for handling the heap is using notions of separation and more generally subheaps in order to abstract away correlations between the properties of subheaps.

This is a joint work with Josh Berdine, Byron Cook, Alexey Gotsman, Tal Lev-Ami, Roman Manevich, G. Ramalingam, and Michal Segalov.