Memory Management in Concurrent Algorithms
(Invited Talk)

Maged M. Michael

IBM Thomas J. Watson Research Center

Many shared memory concurrent algorithms involve accessing dynamic nodes of shared structures optimistically, where a thread may access dynamic nodes while they are being updated by concurrent threads. Optimistic access is often necessary to enable non-blocking progress, and it is desirable for increasing concurrency and reducing conflicts among concurrent operations.

Optimistic access to dynamic nodes and structures makes the memory management of such nodes one of the most complex aspects of concurrent algorithms. For example, a thread operating on a shared dynamic structure may access a dynamic node after it was removed from the structure by another thread, and possibly even after it was reclaimed for unrelated reuse. Unless the algorithm and the memory management method it employs are carefully designed to handle such situations correctly, they can lead to various errors such as memory access violations, return of incorrect results, and/or corruption of shared data.

The main purpose of memory management in the context of concurrent algorithms is to balance the goal of enabling and maximizing the flexible reuse of dynamic nodes that are no longer needed, with the goal of maximizing the flexibility of safe access to dynamic nodes.

The verification of a concurrent algorithm is incomplete without taking into account the algorithm's memory management properties. Some such properties need to be verified explicitly, as they are not necessarily covered by conventional safety and progress properties, such as linearizability and deadlock-freedom. Understanding the subtleties of memory management properties of concurrent algorithms and the relations between these properties can help make verification of concurrent algorithms more effective and complete.

The talk discusses memory management problems and issues in concurrent algorithms, including the tightly-related ABA problem, categories of memory management and ABA-prevention solutions, types of memory access properties of concurrent algorithms, memory bounds, progress properties of memory management methods, and relations among these various properties.

T. Touili, B. Cook, and P. Jackson (Eds.): CAV 2010, LNCS 6174, p. 23, 2010.