Verifying Concurrent Search Structures

Thomas Wies
New York University, New York, NY, US

Abstract

Search structures support the fundamental data storage primitives on key-value pairs: insert a pair, delete by key, search by key, and update the value associated with a key. Concurrent search structures are parallel algorithms to speed access to search structures on multicore and distributed servers. For these data structures to be efficient, the underlying parallel algorithms need to perform fine-grained synchronization between threads. This makes them notoriously difficult to design and implement correctly. Indeed, bugs are routinely found both in actual implementations and in the designs proposed by experts in peer-reviewed publications. Often, these bugs elude testing-based quality control due to complex thread interactions that only manifest after deployment, and under conditions that are difficult to replicate. Given the critical role that concurrent search structures play in today's software infrastructure, it is therefore highly desirable to verify their correctness using formal methods, preferably in an automated fashion.

In this talk, I will present a framework for obtaining linearizability proofs for concurrent search structures that are modular, reusable, and amenable to automation. The framework takes advantage of recent advances in local reasoning techniques based on concurrent separation logic. I will provide an overview of these techniques and discuss there use for verifying both lock-based and lock-free concurrent search structures such as concurrent (skip)lists, hash structures, binary search trees, B trees, and log-structured merge trees.

2012 ACM Subject Classification Theory of computation → Program verification

Keywords and phrases Concurrent search structures

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2024.3

Category Invited Talk

Funding This work is funded in parts by the United States National Science Foundation under grants CCF-2304758 and CCF-1815633. Further funding came from an Amazon Research Award Fall 2021. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not reflect the views of Amazon.

Acknowledgements This talk is based on joint work with many of my students and colleagues, including Siddharth Krishna, Roland Meyer, Nisarg Patel, Dennis Shasha, Alexander Summers, and Sebastian Wolff.

35th International Conference on Concurrency Theory (CONCUR 2024).
Editors: Rupak Majumdar and Alexandra Silva; Article No. 3; pp. 3:1–3:1
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany