Book Review

Automated Verification of Concurrent Search Structures
Siddharth Krishna, Nisarg Patel, Dennis Shasha, and Thomas Wies
www.morganclaypool.com
Synthesis Lectures on Computer Science
ISBN: 9781636391281 paperback
ISBN: 9781636391298 ebook
ISBN: 9781636391304 hardcover

Algorithms are evolved, tested, applied and improved to ensure the functioning of systems and packages more effectively. Formal verification mechanisms support the mathematical techniques to prove, in a rigorous and machine-checkable manner. The software formulation and development and its success depend how efficiently we deploy formal verification. Against this backdrop the authors have made convincible preparation of a synthesis on Automated Verification of Concurrent Structures.

This book has 14 chapters. In the introduction the authors have explained concurrent algorithms and logic behind such algorithms.

The second chapter is spared to introduce the preliminaries. Normally a book is expected to provide the preliminaries as a part of a chapter. However, this book has an exclusive chapter on preliminaries which explain the notations used throughout the book. They have deployed a machine learning based programming language which is explained together with template algorithms.

To describe the status of the concurrent data structures used, the authors have introduced the separation logic that describes a set of states which satisfies a proposition when the state is described by the proposition. They started with an illustration to explain the SEPARATING CONJUNCTIONS and propositions.

In the fourth chapter, the auxiliary variables a formal technique where the prover adds state (variables or resources) to a program that capture knowledge about the history of a computation is explained. They examined the technique of using ghost state to construct proofs of concurrent algorithms and described how they have used the ghost state in Iris to verify template search. It is used to verify a template algorithm for a single node structure. In the fifth chapter on The Keyset Resource Algebra they defined a keyset resource algebra (RA) that can be used for many single-copy search structures, and demonstrated it by verifying a two-node template. The disjoint keysets are defined with the proof of TWO-NODE TEMPLATE

The next chapter outlined the edgeset framework that allows one to view a single-copy search structure as an abstract graph whose nodes are labeled by their contents and edges are labeled by edgesets. The single copy EDGESETS framework is illustrated with example.

The seventh chapter *Flow Framework* motivated the flow framework and presents the key aspects of the framework that we use in this monograph.

In the chapter 8, the authors demonstrated how to simplify the verification of concurrent search structures by abstracting concurrency algorithms underlying diverse implementations such as B-trees and hash tables into templates that can be verified once and for all. In the ninth chapter they verified the multicopy stricture.

In the tenth chapter on The Edgeset Framework for Multicopy Structures the authors introduced the edgeset framework for multicopy structures and formally define the specifications that each operation on a multicopy structure must satisfy. In the next chapter on Reasoning about Non-Static and Non-Local Linearization Points argue that the concurrent execution of upsert and search operations that satisfy the template-level specifications and they did it by letting the upserts in the equivalent sequential execution occur in the same order as their atomic commit points in the concurrent execution. The template proof that was explained earlier now how to decouple it is explained. In the chapter on Verifying the LSM DAG Template the authors have described a template for multicopy structures that generalizes the LSM (log-structured merge) tree discussed in the ninth chapter to arbitrary directed acyclic graphs (DAGs). The authors have proved the linearizability of the LSM DAG template by verifying that all operations satisfy the template-level atomic triples.

In the thirteenth chapter on Proof Mechanization and Automation authors assessed the techniques presented so far by mechanically verifying the template algorithms in the earlier chapters. Further this part provided a summary of the development using experiments conducted on systems. Finally, they provided the summary of templates and instantiations verified in Iris/Coq and GRASShoppe.

In the last chapter they presented the Related Work, Future Work, and Conclusion where the important discussion on correctness criteria for concurrent data structures is presented. The deductive verification of concurrent programs, proof mechanization and automation and template algorithms are supported with earlier literature. The significant part of this book is a wider discussion on future possible research in this direction. An extensive bibliography is supported at the end.

This book is an example of how a technical book can be simplified and presented to basic readers. The book is supported by many illustrations and table data.

Hathairat Ketmaneechairat
King Mongkut's University of Technology North Bangkok
Thailand