Book Review for AMS Reviews

Review of *Directed Algebraic Topology and Concurrency*
by Lisbeth Fajstrup, Eric Goubault, Emmanuel Haucourt, Samuel Mimram, Martin Raussen
Springer, 2016

Our society increasingly depends on complex, software-intensive systems in all areas of our daily lives. Such systems often interact concurrently, so as to jointly provide a service or to increase performance. However, comprehending and analyzing the interactions of concurrent processes is a challenge due to the sheer number of ways in which process executions can interleave. Fortunately, the number of executions and thus a system's state space can be cut drastically when noting that many actions between processes are typically independent. Executions that differ only in their ordering of independent actions are equivalent and, for analyzing properties of concurrent systems such as the reachability of deadlocks, it suffices to consider one representative of each equivalence class only. This idea underlies true-concurrency models of concurrent systems, e.g., asynchronous transition systems and event structures, and is also utilized in interleaving models, e.g., in the context partial-order reduction.

The reviewed book introduces a more recently developed true-concurrency model, which applies concepts and methods from algebraic topology. It considers a simple, imperative concurrent programming language, together with mutexes and semaphores to coordinate interaction between processes and access to resources. A program's state space is captured as a higher-dimensional space, with one dimension for each process. Program executions correlate to paths in this space, where the semaphores describe regions through which no execution may traverse. The above-mentioned equivalence on executions then corresponds to the topological notion of homotopy, i.e., two executions are equivalent if one can be continuously deformed to the other without crossing forbidden regions. One difference to homotopy as studied in classic topology is that program executions always go forward in time. It is the central concern of this book to extend notions and results in algebraic topology to the directed case, e.g., by introducing *dihomotopy*. The resulting theory leads to a novel geometrical semantics for concurrent programs that is described in the formalism of category theory, and to new algorithms for compactly representing state spaces and calculating program properties such as deadlock.

Most of the book's presentation is restricted to programs without loops and where resource consumption only depends on state, which implies that standard formalizations of, e.g., producer-consumer problems are not expressible (Ch. 2). Concurrent programs are given first a discrete semantics as asynchronous graphs, which describe the commutation of two actions explicitly and are equipped with a notion of dihomotopy. The semantics is generalized to *precubical sets*, which take the commutation of n actions into account (Ch. 3) and can be seen as topological spaces. Precubical sets consist of n-dimensional cubes, for each $n \in \mathbb{N}$, together with their faces. Directed spaces and paths as introduced by Grandis are then considered, leading to a *directed geometrical semantics*, i.e., a continuous semantics corresponding to the precubical semantics (Ch. 4). Alongside, the authors lift categorical constructions in algebraic topology to the directed case and study the *fundamental category* of a directed space, whose objects are the points in the space and whose morphisms are the directed paths up to dihomotopy. Three algorithms based on the geometric semantics are also developed (Ch. 5): for determining a compact representation of cubical regions, for detecting deadlocks, and for factorizing a program's geometric semantics as a product of subspaces.

The book concludes with two chapters on advanced topics and a very brief chapter on applications. Ch. 6 introduces the *category of components*, which is a notion of connected components in directed topology and results from a non-trivial reduction from the fundamental category via weak isomorphisms. This permits the identification of actions that do not matter in a program's execution due to action commutation. An algorithm that utilizes this for computing an approximation of the category of components is provided, too. Ch. 7 constructs combinatorial models for the space of directed paths modulo dihomotopy, which enables efficient computations. Finally, Ch. 8 hints at applications: for example, proving serializability and deadlock freedom in distributed databases, and studying semantic equivalences such as bisimulation for higher-dimensional automata using topological invariants.

This introductory book is authored by internationally leading researchers in the application of algebraic topology to concurrency. It is published at a time when sufficiently many interesting results have been established in this young field, so as to demonstrate its relevance and importance. The authors do a brilliant job in fascinating readers with the intuitive geometric interpretations of concurrent executions, using many concise examples and visualizations. While sometimes being quite technical, the examples always provide good intuition and insight, and a wealth of remarks offer much additional context and understanding. The authors also extensively discuss current limitations and open research questions, and provide pointers to the research literature where further details and proofs can be found.

To be able to address a reasonably large audience of computer scientists and mathematicians, the authors' selections of content and presentation style are necessarily a compromise. For the book to be accessible, readers should have basic knowledge of concurrency theory and algebraic topology, and good knowledge in (applying) category theory. As a computer scientist, I wished the authors would have formally related their geometrical model of concurrent programs to more mainstream models such as event structures, provided more algorithmic details and some benchmarking results wrt. representative concurrent programs, and presented at least one case study in detail. Despite these shortcomings, I applaud the authors for having delivered what I believe is the first book in a very promising field.

I highly recommend this book to mathematicians specializing in topology and seeking a challenging new application, and to those concurrency theoreticians who have a firm background in category theory. The exciting research field of directed algebraic topology and concurrency will likely see rapid advances over the next few years, and I am hoping to then hold a new edition of the book in my hands that I can recommend not only to colleagues but also to students.

GERALD LÜTTGEN
Software Technologies Research Group
University of Bamberg
D–96045 Bamberg, Germany
E-mail: gerald.luettgen@swt-bamberg.de