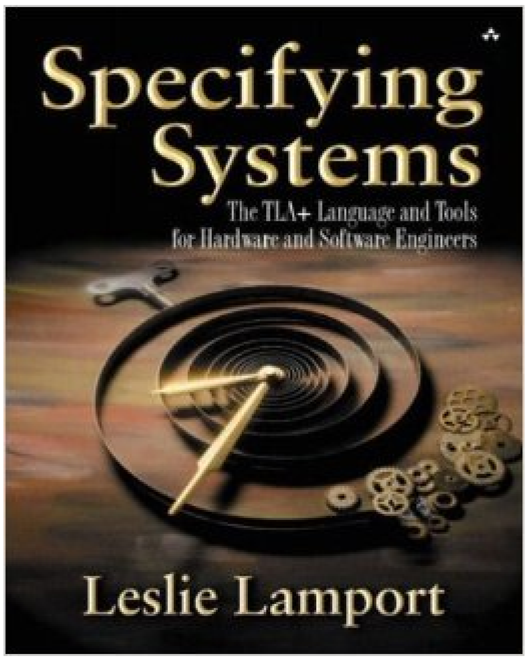
Specifying Systems: The TLA+ Language And Tools For Hardware And Software Engineers


Specifying Systems
The TLA+ Language and Tools for Hardware and Software Engineers
Leslie Lamport

Synopsis

This work shows how to write unambiguous specifications of complex computer systems. The first part provides a concise and lucid introduction to specification, explaining how to describe, with mathematical precision, the behavioural properties of a system - what that system is allowed to do. The emphasis here is on safety properties. The second part covers more advanced topics, including liveness and fairness, real time properties, and composition. The books final two parts provide a complete reference manual for the TLA+ language and tools, as well as a mini-manual.

Book Information

Paperback: 384 pages

Publisher: Addison-Wesley Professional; 1 edition (July 29, 2002)

Language: English

ISBN-10: 032114306X

ISBN-13: 978-0321143068

Product Dimensions: 7.3 x 1 x 9 inches

Shipping Weight: 1.1 pounds (View shipping rates and policies)

Average Customer Review: 5.0 out of 5 starsÂ Â See all reviewsÂ (4 customer reviews)

Best Sellers Rank: #213,252 in Books (See Top 100 in Books)   #20 inÂ Books > Computers & Technology > Programming > Parallel Programming   #30 inÂ Books > Computers & Technology > Hardware & DIY > Microprocessors & System Design > Computer Design   #110 inÂ Books > Science & Math > Mathematics > Pure Mathematics > Logic

Customer Reviews

How is it possible that nobody has reviewed this book yet?This has been an eye opener book for concurrency and distributed software design. I learned more with this book (and the TLA+ toolbox) than with the "classic" from Lynch on distributed algorithms. I do not know if Lynch book also improves qualitatively when used with the Tempo toolbox. I guess it is worth trying.The book is challenging but it is really well-written. I’ll recommend the book to anyone that wants to challenge his understanding of computer systems (hardware, software, concurrency, etc). You will never evaluate your designs so naively.I would like to have a new edition of this book that covers TLA+2.BTW, You can get the pdf from Lamport web page.

One of the best books I have ever read on any topic. Even if you don’t care about the subject matter (modeling and model-checking), the blazing clarity and simplicity will delight you. Classically reductionist, it boils all the complexity of systems (including distributed, concurrent, parallel, Byzantine systems, the kinds of things that send most practitioners running away in horror) down to a handful of primitives in ordinary Boolean logic. Another one of those books (like SICP and VCLADF and the Feynman lectures) that a high-schooler can understand but the average PhD would benefit from.

For a long time, I’ve been thinking about ways to improve the reliability of the software we write. We use many techniques to strive for this goal, sometimes in a roundabout way, but the ultimate approach would be mathematical proofs of correctness.Of the approaches I’ve read over the years, Lamport’s TLA+ comes the closest to this promise a system specification that’s workable. This book provides a great introduction with practical, real-world examples.

This is a very good introduction to temporal reasoning, with a language designed to encourage habits essential for describing large systems readably. The book will benefit both novice and expert readers, and also people that do not plan to specify systems themselves, but need to understand the process, so that they can communicate with those that will. The material is presented precisely in simple mathematics, introduced as needed. The writing is engaging, not at all boring nor boilerplate, which has been a challenge for many other authors.To convince yourself, you can start by reading the PDF version online, available by the author here: http://research.microsoft.com/en-us/um/people/lamport/tla/book.htmlCompared to the relevant papers by Lamport and coauthors, the book is geared more towards beginners and users. It is not a collection of all the results, but has a complementary purpose. In particular, composition of open systems and timing constraints are developed in more detail outside this book.

Download to continue reading...

Kaufmann Series in Computer Architecture and Design) Computer Organization and Design: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design) Getting Started with 3D Printing: A Hands-on Guide to the Hardware, Software, and Services Behind the New Manufacturing Revolution Debugging: The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems Linux Enterprise Cluster: Build a Highly Available Cluster with Commodity Hardware and Free Software Make: FPGAs: Turning Software into Hardware with Eight Fun and Easy DIY Projects Complete CompTIA A+ Guide to IT Hardware and Software (7th Edition) Computer Networking from LANs to WANs: Hardware, Software and Security (Networking) Software Components With Ada: Structures, Tools, and Subsystems (The Benjamin/Cummings Series in Ada and Software Engineering) Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for Software Protection Software Engineering Classics: Software Project Survival Guide/ Debugging the Development Process/ Dynamics of Software Development (Programming/General) The Pocket Universal Principles of Design: 150 Essential Tools for Architects, Artists, Designers, Developers, Engineers, Inventors, and Makers