Principles and Specifications of Concurrent Systems

Leslie Lamport

Sections colored like this have

not yet been written.

Version of 28 August 2017

The *Principles* and *Specification* Tracks

?

















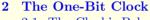






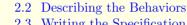


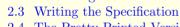


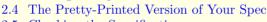










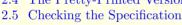


1 Introduction

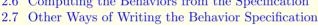
1.3 Specification

1.1 Concurrent Computation 1.2 Modeling Computation

1.4 Systems and Languages















3.1 Representing the Problem in TLA⁺



























Proving Invariance 4.9.1

Divisors

The GCD Operator

CHOOSE and the Maximum of a Set

Verifying GCD1-GCD3 4.9.2

The Generalized Die Hard Problem 5.1 The PlusCal Representation 5.2 Checking the Algorithm 5.3 The TLA⁺ Translation Alternation 6.1 The Problem 6.2 The One-Bit Clock Revisited 6.3 Specifying Alternation: Safety 6.4 Specifying Alternation: Liveness 6.5 The Two-Phase Handshake Protocol 6.6 Refinement 6.7 Refinement and Stuttering 6.7.1 Adding Steps 6.7.2Temporal Logic and Stuttering 6.7.3 A Finer-Grained Algorithm 6.8 Temporal Logic and Refinement 6.9 Alternation Revisited 6.10 Round-Robin Synchronization 6.10.1 The One-Bit Clock Revisited Again 6.10.2 An N-Valued Clock 6.10.3 An Implementation of the N-Valued Clock 6.10.4 Round-Robin Synchronization The *Principles* Track

7 Mutual Exclusion

- - 7.1 The Problem
 - 7.2 The One-Bit Protocol

 - 7.2.1 The Protocol
 - 7.2.2
 - An Assertional Proof

4.9.3 Proving Termination 4.10 Euclid's Algorithm for Sets

- 7.2.3 Using TLC to Check an Inductive Invariant
- 7.3 The Two-Process One-Bit Algorithm

 - The Two-Process Algorithm
 - 7.3.1 7.3.2
 - Busy Waiting Versus Synchronization Primitives

 - Requirement (c) 7.3.3
- 7.4 Proving Liveness
- 7.5 An Informal Proof 7.6 A More Formal Proof
- 7.7 The N-Process One-Bit Algorithm

? ← ←

 \mathbf{S}

		7.8 The Bakery Algorithm
		7.8.1 The Big-Step Algorithm
		7.8.2 Choosing the Grain of Atomicity
		7.8.3 The Atomic Bakery Algorithm
		7.8.4 The Real Bakery Algorithm
		7.9 Mutual Exclusion in Modern Programs
?	8	The Bounded Channel and Bounded Buffer
		8.1 The Bounded Channel
←		8.1.1 The Specification
→		8.1.2 Safety
\mathbf{C}		8.1.3 Liveness
C		8.1.4 Implementing The Bounded Channel
I		8.2 The Bounded Buffer
\mathbf{S}		8.2.1 Modular Arithmetic
5		8.2.2 The Algorithm
		8.3 The Bounded Buffer Implements the Bounded Channel
		8.3.1 The Refinement Mapping
		8.3.2 Showing Implementation
		8.3.3 Liveness
		8.4 A Finer-Grained Bounded Buffer
		8.5 Further Refinement
		8.6 What is a Process?
	\mathbf{T}	The Specification Track
	9	An Input/Output Specification
		9.1 The Example
		9.2 Sorting
		9.3 Votes
		9.4 The Borda Ranking
		9.5 The Condorcet Ranking
		9.6 Transitive Closure
		9.6.1 A Mathematical Definition
		9.6.2 A Definition TLC Can Execute Faster
		9.6.3 Warshall's Algorithm
		9.7 The Condorcet Ranking Revisited
	Т	The TLA ⁺ Proof Track
		THE LETT I TOOL TIGER
	10	O About Proofs and Proving
		10.1 About Proofs

11 Correctness of Euclid's Algorithm

- 11.1 Proving Safety
- 11.2 Proving Properties of the GCD

12 The Proof Language

- 12.1 What a Theorem Asserts
 - 12.2 The Hierarchical Structure of a Proof
 - 12.2.1 Writing Structured Proofs
 - 12.2.2 Reading Structured Proofs
 12.3 The State of a Proof
 - 12.3.1 Steps That Can Have a Proof
 - 12.3.2 Steps That Cannot Have a Proof
 - 12.4 Proof Obligations
 - 12.5 Further Details
 - 12.5.1 Additional Language Features
 - 12.5.2 Importing
 - 12.5.3 Recursively Defined Functions and Operators
 - 12.5.4 The Fine Print

13 The Bounded Buffer Proof

Math

? ← ← C

 \mathbf{S}

13 Arithmetic and Logic

- 13.1 Arithmetic
- 13.2 Mathematical Logic
- 13.3 Propositional Logic
 - $13.3.1 \land \text{and} \lor$
 - 13.3.2 Other Propositional Operators
- 13.4 Predicate Logic
- 13.5 The CHOOSE Operator

14 Sets

- 14.1 An Introduction to Sets
- 14.2 Simple Set Operators
- 14.3 Set Constructors
- 14.4 Subset and Union
- 14.5 Collections too Big to Be Sets
- 14.6 Bags

15 Function	ons
15.1 Fu	nctions and Their Domains
$15.2~\mathrm{Wr}$	iting Functions
15.3 Set	s of Functions
	e except Construct
15.5 Tu	
$15.6 \mathrm{Re}$	
$15.7\mathrm{Str}$	ings
16 Miscel	laneous Constructs
16.1 Co	nditional Constructs
	.1.1 IF / THEN / ELSE
	.1.2 CASE
16.2 De	finitions
	.2.1 Simple Operator Definitions
	.2.2 Function Definitions
	.2.3 Recursive Operator Definitions
	.2.4 Recursive Or Inductive?
	e let / in Construct
16.4 Th	e lambda Construct
17 Tempo	oral Logic
$17.1~\mathrm{Un}$	derstanding Temporal Formulas
17.2 Pro	oof Rules and Proofs
17.3 Ru	les for Proving Safety
17.4 Lea	
	.4.1 The Leads-To Induction Rule
	.4.2 The $\square \rightsquigarrow \text{Rule}$
17	.4.3 Proving \sim Formulas by Contradiction
17.5 Fai	rness
	.5.1 The ENABLED Operator
	.5.2 Weak Fairness
	.5.3 Strong Fairness
	.5.4 Proving \sim Properties with Fairness
17	.5.5 Proving Fairness
Topics	
18 Variab	le Hiding and Auxiliary Variables
19 Reduc	
	0.43.74.4

? ← ← C

I S

20 Debugging With TLC

20.1 Print Statements

 $20.2\,\mathrm{Having}\;\mathrm{TLC}\;\mathrm{Set}$ and Read Values

20.3 Using LET

20.4 The Perils of Lazy Evaluation

? ←
C
I