# Contents

*Preface*  
ix  

## Part I: Introduction and Overview  
1  
1.1 Central Questions  
1.2 Structure of this Chapter  
1.3 Basic Concepts of Concurrency  
1.4 Why Concurrent Programs Should be Proved Correct  
1.5 The Approach of this Book  
1.6 Compositionality  
1.7 From Noncomp. to Compo Proof Methods – a historical perspective  

## Part II: The Inductive Assertion Method  
71  
2  
2.1 Objectives of Part II  
2.2 Structure of this Chapter  
2.3 Sequential Transition Diagrams and Systems  
2.4 Specification and Correctness Statements  
2.5 A Proof Method for Partial Correctness  
2.6 Soundness  
2.7 Semantic Completeness of the Inductive Assertion Method  
2.8 Proving Convergence  
2.9 Proving Absence of Runtime Errors  
2.10 Historical Notes  

## 3 The Inductive Assertion Method for Shared-Variable Concurrency  
119  
3.1 Objective and Structure of this Chapter  
3.2 A Characterisation of Concurrent Execution
Table of Contents

3.3 Is this Characterisation of Concurrent Execution Justified? 132
3.4 The Generalisation of Floyd's Approach to Nondeterministic Interleavings 134
3.5 Concurrent Transition Systems with Shared Variables 137
3.6 Proving Convergence for Shared-Variable Concurrency 188
3.7 Proving Deadlock Freedom 203
3.8 Proving Absence of Runtime Errors 206
3.9 Historical Notes 208

4 The Inductive Assertion Method for Synchronous Message Passing 221
4.1 Objective and Introduction 221
4.2 Structure of this Chapter 223
4.3 Syntax and Semantics of Synchronous Transition Diagrams 223
4.4 Proof Methods for Partial Correctness 227
4.5 Semantic Completeness 249
4.6 Technical Note: Modifications Towards Compositionality 264
4.7 A Modular Method for Proving Convergence 269
4.8 Verifying Deadlock Freedom 277
4.9 Proving Absence of Runtime Errors 279
4.10 Historical Notes 282

5 Expressibility and Relative Completeness 291
5.1 Objective 291
5.2 Structure of this Chapter 292
5.3 Syntactic Notions 292
5.4 Partial Correctness of Syntactic Transition Diagrams 298
5.5 Relative Completeness of Floyd's Inductive Assertion Method 300
5.6 Relative Completeness of the Method of Owicky & Gries 309
5.7 Relative Completeness of the Method of Apt, Francez & de Roever 312
5.8 Historical Notes 316

6 Introduction to Compositional Reasoning 354
6.1 Motivation 354
6.2 Introduction to Part III and to this Chapter 356
6.3 Assume-Guarantee-based Reasoning 359
6.4 Assumption-Commitment-based Reasoning 361
6.5 Rely-Guarantee-based Reasoning 363

Part III: Compositional Methods based on Assertion Networks 353

Picture Gallery 319

© in this web service Cambridge University Press www.cambridge.org
## Contents

7 Compositional Proof Methods: Synchronous Message Passing 367
  7.1 Objective and Introduction 367
  7.2 Structure of the Chapter 368
  7.3 Top-level Synchronous Message Passing 369
  7.4 A Compositional Proof Method for Nested Parallelism 379
  7.5 Assumption-Commitment-based Reasoning 397
  7.6 Historical Notes 429

8 Compositional Proof Methods: Shared-Variable Concurrency 438
  8.1 Introduction and Overview 438
  8.2 Concurrent Transition Diagrams 439
  8.3 Top-Level Shared-Variable Concurrency 440
  8.4 The Rely-Guarantee Method 447
  8.5 Historical Notes 479

Part IV: Hoare Logic 487

9 A Proof System for Sequential Programs Using Hoare Triples 488
  9.1 Introduction and Overview of Hoare Logics 488
  9.2 Structure of this Chapter 497
  9.3 Syntax and Informal Meaning of GCL\(^+\) Programs 498
  9.4 Semantics of GCL\(^+\) 501
  9.5 A Proof System for GCL\(^+\) Programs 506
  9.6 Soundness and Relative Completeness 511
  9.7 Proof Outlines 517
  9.8 Alternative Definitions of Proof Outlines 521
  9.9 Examples of Verification during Program Development 522
  9.10 Historical Notes 526

10 A Hoare Logic for Shared-Variable Concurrency 531
  10.1 Introduction and Overview 531
  10.2 Syntax and Informal Meaning of SVL Programs 532
  10.3 Semantics of SVL\(^+\) 537
  10.4 A Proof System for SVL Programs 540
  10.5 An Extended Example: Concurrent Garbage Collection 563
  10.6 Completeness of the Owicki & Gries Method 584

11 A Hoare Logic for Synchronous Message Passing 600
  11.1 Structure of this Chapter 600
  11.2 Syntax and Informal Meaning of DML Programs 601
  11.3 Semantics of DML 606
  11.4 A Hoare Logic for Synchronous Message Passing 608
  11.5 Soundness and Relative Completeness of this Hoare Logic 630