

Contents

1	Introduction	1
----------	---------------------	----------

Part I Preliminaries

2	Overview of Formal Verification	9
2.1	Theorem Proving	9
2.2	Temporal Logic and Model Checking	12
2.3	Program Logics, Axiomatic Semantics, and Verification Conditions	17
2.4	Bibliographic Notes	22
3	Introduction to ACL2	25
3.1	Basic Logic of ACL2	25
3.2	Ground Zero Theory	27
3.2.1	Terms, Formulas, Functions, and Predicates	30
3.2.2	Ordinals and Well-Founded Induction	32
3.3	Extension Principles	35
3.3.1	Definitional Principle	36
3.3.2	Encapsulation Principle	40
3.3.3	Defchoose Principle	42
3.4	The Theorem Prover	44
3.5	Structuring Mechanisms	45
3.6	Evaluators	46
3.7	The ACL2 Programming Environment	47
3.8	Bibliographic Notes	48

Part II Sequential Program Verification

4	Sequential Programs	53
4.1	Modeling Sequential Programs	53
4.2	Proof Styles	55
4.2.1	Stepwise Invariants	55
4.2.2	Clock Functions	56
4.3	Comparison of Proof Styles	57

4.4	Verifying Program Components and Generalized Proof Obligations	59
4.5	Discussion	62
4.5.1	Overspecification	62
4.5.2	Forced Homogeneity	63
4.6	Summary	64
4.7	Bibliographic Notes	64
5	Operational Semantics and Assertional Reasoning	65
5.1	Cutpoints, Assertions, and VCG Guarantees	65
5.2	VCG Guarantees and Symbolic Simulation	68
5.3	Composing Correctness Statements	70
5.4	Applications	72
5.4.1	Fibonacci Implementation on TINY	73
5.4.2	Recursive Factorial Implementation on the JVM	75
5.4.3	CBC-Mode Encryption and Decryption	75
5.5	Comparison with Related Approaches	76
5.6	Summary	78
5.7	Bibliographic Notes	78
6	Connecting Different Proof Styles	81
6.1	Soundness of Proof Styles	82
6.2	Completeness	84
6.3	Remarks on Mechanization	88
6.4	Discussion	88
6.5	Summary and Conclusion	90
6.6	Bibliographic Notes	91
Part III Verification of Reactive Systems		
7	Reactive Systems	95
7.1	Modeling Reactive Systems	96
7.2	Stuttering Trace Containment	97
7.3	Fairness Constraints	99
7.4	Discussion	103
7.5	Summary	106
7.6	Bibliographic Notes	107
8	Verifying Concurrent Protocols Using Refinements	109
8.1	Reduction via Stepwise Refinement	110
8.2	Reduction to Single-Step Theorems	110
8.3	Equivalences and Auxiliary Variables	114
8.4	Examples	116
8.4.1	An ESI Cache Coherence Protocol	116
8.4.2	An Implementation of the Bakery Algorithm	119
8.4.3	A Concurrent Deque Implementation	124

8.5	Summary	129
8.6	Bibliographic Notes	129
9	Pipelined Machines	131
9.1	Simulation Correspondence, Pipelines, and Flushing Proofs	131
9.2	Reducing Flushing Proofs to Refinements	134
9.3	A New Proof Rule	136
9.4	Example	137
9.5	Advanced Features	141
9.5.1	Stalls	141
9.5.2	Interrupts	141
9.5.3	Out-of-Order Execution	142
9.5.4	Out-of-Order and Multiple Instruction Completion	142
9.6	Summary	143
9.7	Bibliographic Notes	144

Part IV Invariant Proving

10	Invariant Proving	149
10.1	Predicate Abstractions	151
10.2	Discussion	153
10.3	An Illustrative Example	154
10.4	Summary	156
10.5	Bibliographic Notes	157
11	Predicate Abstraction via Rewriting	159
11.1	Features and Optimizations	163
11.1.1	User-Guided Abstraction	164
11.1.2	Assume Guarantee Reasoning	164
11.2	Reachability Analysis	165
11.3	Examples	166
11.3.1	Proving the ESI	166
11.3.2	German Protocol	168
11.4	Summary and Comparisons	169
11.5	Bibliographic Notes	171

Part V Formal Integration of Decision Procedures

12	Integrating Deductive and Algorithmic Reasoning	175
13	A Compositional Model Checking Procedure	179
13.1	Formalizing a Compositional Model Checking Procedure	180
13.1.1	Finite State Systems	180
13.1.2	Temporal Logic formulas	181
13.1.3	Compositional Procedure	181

13.2	Modeling LTL Semantics	183
13.3	Verification	187
13.4	A Deficiency of the Integration and Logical Issues	190
13.5	A Possible Remedy: Integration with HOL4	192
13.6	Summary and Discussion	193
13.7	Bibliographic Notes	194
14	Connecting External Deduction Tools with ACL2	195
14.1	Verified External Tools	196
14.1.1	Applications of Verified External Tools	200
14.2	Basic Unverified External Tools	203
14.2.1	Applications of Unverified External Tools	204
14.3	Unverified External Tools for Implicit Theories	205
14.4	Remarks on Implementation	209
14.4.1	Basic Design Decisions	209
14.4.2	Miscellaneous Engineering Considerations	211
14.5	Summary	214
14.6	Bibliographic Notes	215
 Part VI Conclusion		
15	Summary and Conclusion	219
References		223
Index		237



<http://www.springer.com/978-1-4419-5997-3>

Scalable Techniques for Formal Verification

Ray, S.

2010, XIV, 243 p., Hardcover

ISBN: 978-1-4419-5997-3