

Janusz Laski
William Stanley

Software Verification and Analysis

An Integrated, Hands-On Approach



Springer

Contents

Introduction: What Do We Want To Know About the Program?.....

1	What is the Program Doing: Specification	1
2	How to Make Sure That the Program is Doing it Right: Verification	5
3	Trying to Show That the Program is Incorrect: Testing	8
4	Trying to Locate the Cause of Incorrectness: Debugging	11
5	What One Can Tell About The Program Without Executing It: Static Analysis	15
6	The Scope of the SAT Methods	17
7	Conclusions	20
	Exercises	21
	References	21

Part I The Semantic Analysis

1	Why Not Write Correct Software the First Time?	25
1.1	Express Yourself Precisely: The Precondition	25
1.2	The Postcondition	28
1.3	The Principles of Top-Down Refinement	32
1.4	The Example Continued	33
1.5	Conclusions	36
	References	37
2	How to Prove a Program Correct: Programs Without Loops	39
2.1	Program Correctness	39
2.2	The Weakest Precondition $wp(S, Q)$	42
2.3	Finding the $wp(S, Q)$	43
2.3.1	The Assignment Axiom	43
2.3.2	A Sequence of Assignments: The Composition Rule	44
2.4	SPARK Experiments	45
2.5	Programs With Many Paths	49

2.6	The Derivation of Partial Weakest Precondition (pwp) and Path Traversal (tr).....	52
2.7	The Assertion Method	56
2.8	Conclusions	60
	Exercises	60
	References	61
3	How to Prove a Program Correct: Iterative Programs	63
3.1	When You Cannot Verify All Paths: Programs with Loops.....	63
3.2	From the Particular to the General: Mathematical Induction.....	65
3.3	Loop Invariants	66
3.4	Where Do Invariants Come From: Goal Invariant.....	70
3.5	Supporting the Proof: Using the Proof Checker.....	72
3.6	Does the Loop Terminate? Variants.....	76
3.7	Conclusions	77
	Exercises.....	78
	References	79
4	Prepare Test for Any Implementation: Black-Box Testing	81
4.1	Testing Principles	81
4.2	Functionality Testing	85
4.2.1	Special Values.....	86
4.2.2	Fixed Points	86
4.2.3	Special Classes.....	87
4.2.4	Boundary Analysis.....	87
4.3	Partition Testing.....	88
4.4	An Example	89
4.5	Random Testing.....	95
4.6	Conclusions	97
	Exercises.....	98
	References	99
Part II Static Analysis		
5	Intermediate Program Representation	103
5.1	Introduction	103
5.2	Program Parse and Syntax Trees	104
5.3	Program Control Flowgraph.....	104
5.4	Labeled Flowgraphs	109
5.5	Deriving the Flowgraph.....	112
5.6	Paths in Flowgraphs.....	116
5.7	Conclusions	123

Exercises.....	123
References	123
6 Program Dependencies.....	125
6.1 Motivations.....	125
6.2 Dominators and Attractors	128
6.3 Control Dependency: Structured Control.....	131
6.4 Control Dependency: Arbitrary Control.....	135
6.5 Computing Control Dependency	137
6.6 Data and General Dependency	139
6.7 Conclusions	141
Exercises.....	142
References	142
7 What Can One Tell About a Program Without Its Execution: Static Analysis	143
7.1 Motivations	143
7.2 Control Flow Anomalies	145
7.3 Data Flow Anomalies	147
7.3.1 Undefined-Referenced (UR) Anomaly: The Use of Uninitialized Variables	149
7.3.2 Double Definition (DD) Anomaly.....	151
7.3.3 Redundant Statement (RS) Anomaly.....	151
7.3.4 Loop Analysis.....	151
7.4 Modeling Procedure Calls	152
7.5 Signature Anomalies	158
7.6 Descriptive Static Analysis.....	163
7.6.1 Control Flow Queries.....	164
7.6.2 Data flow and Dependency Queries	164
7.6.3 Structural Testing Queries	165
7.6.4 System (Program) and Visibility Queries	165
7.7 Events on Program Paths	165
7.8 Conclusions	168
Exercises.....	169
References	169

Part III Dynamic Analysis

8 Is There a Bug in the Program? Structural Program Testing	173
8.1 Introduction	173
8.2 Code Coverage Criteria	174
8.3 Testing Scenario	179
8.4 Faults and Errors.....	185
8.5 Fault Detection Power of Code Coverage Testing.....	191

8.6	Program Dependencies in Software Testing.....	193
8.7	Conclusions	198
	Exercises.....	201
	References	201
9	Dynamic Program Analysis.....	203
9.1	Introduction	203
9.2	Operational Semantics: States and Computations.....	204
9.3	Dynamic Analysis Concepts.....	208
9.4	An Application: Dynamic Program Slicing	211
9.5	An Application: Handling Dynamic Data Structures.....	214
9.6	Conclusions	217
	Exercises.....	218
	References	219
	Index.....	221