# Contents

## Part I Foundations

### 1 Formal Methods for Software Construction

by Reiner Hähnle

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 What KeY Is</td>
<td>1</td>
</tr>
<tr>
<td>1.2 About This Book</td>
<td>6</td>
</tr>
<tr>
<td>1.3 The Case for Formalisation</td>
<td>7</td>
</tr>
<tr>
<td>1.4 Creating Formal Requirements</td>
<td>11</td>
</tr>
<tr>
<td>1.5 Proof Obligations</td>
<td>14</td>
</tr>
<tr>
<td>1.6 Proving Correctness of Programs</td>
<td>15</td>
</tr>
</tbody>
</table>

### 2 First-Order Logic

by Martin Giese

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Types</td>
<td>21</td>
</tr>
<tr>
<td>2.2 Signatures</td>
<td>25</td>
</tr>
<tr>
<td>2.3 Terms and Formulae</td>
<td>31</td>
</tr>
<tr>
<td>2.4 Semantics</td>
<td>32</td>
</tr>
<tr>
<td>2.4.1 Models</td>
<td>35</td>
</tr>
<tr>
<td>2.4.2 The Meaning of Terms and Formulae</td>
<td>40</td>
</tr>
<tr>
<td>2.4.3 Partial Models</td>
<td>44</td>
</tr>
<tr>
<td>2.5 A Calculus</td>
<td>46</td>
</tr>
<tr>
<td>2.5.1 An Example Proof</td>
<td>48</td>
</tr>
<tr>
<td>2.5.2 Ground Substitutions</td>
<td>50</td>
</tr>
<tr>
<td>2.5.3 Sequent Proofs</td>
<td>51</td>
</tr>
<tr>
<td>2.5.4 The Classical First-Order Rules</td>
<td>55</td>
</tr>
<tr>
<td>2.5.5 The Equality Rules</td>
<td>58</td>
</tr>
<tr>
<td>2.6 Soundness, Completeness</td>
<td>64</td>
</tr>
<tr>
<td>2.7 Incompleteness</td>
<td>65</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Type Hierarchy and Signature</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Syntax of JAVA CARD DL Terms</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Syntax of JAVA CARD DL Updates</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Syntax of JAVA CARD DL Formulae</td>
</tr>
<tr>
<td>3.3</td>
<td>Semantics</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Kripke Structures</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Semantics of JAVA CARD DL Updates</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Semantics of JAVA CARD DL Terms</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Semantics of JAVA CARD DL Formulae</td>
</tr>
<tr>
<td>3.3.5</td>
<td>JAVA CARD-Reachable States</td>
</tr>
<tr>
<td>3.4</td>
<td>The Calculus for JAVA CARD DL</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Sequents, Rules, and Proofs</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Soundness and Completeness of the Calculus</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Rule Schemata and Schema Variables</td>
</tr>
<tr>
<td>3.4.4</td>
<td>The Active Statement in a Modality</td>
</tr>
<tr>
<td>3.4.5</td>
<td>The Essence of Symbolic Execution</td>
</tr>
<tr>
<td>3.4.6</td>
<td>Components of the Calculus</td>
</tr>
<tr>
<td>3.5</td>
<td>Calculus Component 1: Non-program Rules</td>
</tr>
<tr>
<td>3.5.1</td>
<td>First-Order Rules</td>
</tr>
<tr>
<td>3.5.2</td>
<td>The Cut Rule and Lemma Introduction</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Non-program Rules for Modalities</td>
</tr>
<tr>
<td>3.6</td>
<td>Calculus Component 2: Reducing JAVA Programs</td>
</tr>
<tr>
<td>3.6.1</td>
<td>The Basic Assignment Rule</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Rules for Handling General Assignments</td>
</tr>
<tr>
<td>3.6.3</td>
<td>Rules for Conditionals</td>
</tr>
<tr>
<td>3.6.4</td>
<td>Unwinding Loops</td>
</tr>
<tr>
<td>3.6.5</td>
<td>Replacing Method Calls by Their Implementation</td>
</tr>
<tr>
<td>3.6.6</td>
<td>Instance Creation and Initialisation</td>
</tr>
<tr>
<td>3.6.7</td>
<td>Handling Abrupt Termination</td>
</tr>
<tr>
<td>3.7</td>
<td>Calculus Component 3: Invariant Rules for Loops</td>
</tr>
<tr>
<td>3.7.1</td>
<td>The Classical Invariant Rule</td>
</tr>
<tr>
<td>3.7.2</td>
<td>Loop Invariants and Abrupt Termination in JAVA CARD DL</td>
</tr>
<tr>
<td>3.7.3</td>
<td>Implementation of Invariant Rules</td>
</tr>
<tr>
<td>3.7.4</td>
<td>An Improved Loop Invariant Rule</td>
</tr>
<tr>
<td>3.8</td>
<td>Calculus Component 4: Using Method Contracts</td>
</tr>
<tr>
<td>3.9</td>
<td>Calculus Component 5: Update Simplification</td>
</tr>
<tr>
<td>3.9.1</td>
<td>General Simplification Laws</td>
</tr>
<tr>
<td>3.9.2</td>
<td>Update Normalisation</td>
</tr>
<tr>
<td>3.9.3</td>
<td>Update Application</td>
</tr>
<tr>
<td>3.10</td>
<td>Related Work</td>
</tr>
</tbody>
</table>
4 Construction of Proofs
by Philipp Rümmer
4.1 Taclets by Example
4.2 Schema Variables
  4.2.1 The Kinds of Schema Variables in Detail
  4.2.2 Schematic Expressions
  4.2.3 Instantiation of Schema Variables and Expressions
  4.2.4 Substitutions Revisited
  4.2.5 Schema Variable Modifiers
  4.2.6 Schema Variable Conditions
  4.2.7 Generic Types
  4.2.8 Meta-operators
4.3 Instantiations and Meta Variables
4.4 Systematic Introduction of Taclets
  4.4.1 The Taclet Language
  4.4.2 Managing Rules: An Excursion to Taclet Options
  4.4.3 Well-Formedness Conditions on Taclets
  4.4.4 Implicit Bound Renaming and Avoidance of Collisions
  4.4.5 Applicability of Taclets
  4.4.6 The Effect of a Taclet
  4.4.7 Taclets in Context: Taclet-Based Proofs
4.5 Reasoning About the Soundness of Taclets
  4.5.1 Soundness in Sequent Calculi
  4.5.2 A Basic Version of Meaning Formulae
  4.5.3 Meaning Formulae for Rewriting Taclets
  4.5.4 Meaning Formulae in the Presence of State Conditions
  4.5.5 Meaning Formulae for Nested Taclets
  4.5.6 Elimination of Schema Variables
  4.5.7 Introducing Lemmas in KeY

Part II Expressing and Formalising Requirements

5 Formal Specification
by Andreas Roth and Peter H. Schmitt
5.1 General Concepts
  5.1.1 Operation Contracts
  5.1.2 Invariants
5.2 Object Constraint Language
  5.2.1 OCL by Example
  5.2.2 OCL Syntax
  5.2.3 OCL Semantics
  5.2.4 Advanced Topics
5.3 JAVA Modeling Language ........................................ 277
      5.3.1 JML by Example ........................................ 278
      5.3.2 JML Expressions ....................................... 282
      5.3.3 Operation Contracts in JML ............................. 284
      5.3.4 Invariants in JML ..................................... 287
      5.3.5 Model Fields and Model Methods ........................ 289
      5.3.6 Supporting Verification with Annotations ............... 291
5.4 Comparing OCL and JML ......................................... 292

6 Pattern-Driven Formal Specification
by Richard Bubel and Reiner Hähnle ................................. 295
   6.1 Introduction ............................................. 295
   6.2 The Database Query Specification Pattern .................. 297
      6.2.1 Relational Database Query ............................ 297
      6.2.2 Pattern Usage Example ................................. 304
   6.3 Specification Patterns ..................................... 306
      6.3.1 Format of Specification Patterns ....................... 306
      6.3.2 Application of Specification Patterns .................. 307
      6.3.3 Other Pattern Usage Scenarios .......................... 307
   6.4 Simplification of Pattern-Generated Constraints ............ 308
   6.5 Support for Specification Patterns in KeY .................... 310
   6.6 Conclusion and Future Work ................................ 313

7 Natural Language Specifications
by Kristofer Johannisson ............................................. 317
   7.1 Feature Overview .......................................... 317
      7.1.1 Translating OCL to Natural Language ................... 317
      7.1.2 Multilingual Specification Editor ....................... 318
      7.1.3 Suggested Use Cases .................................. 320
   7.2 The Grammatical Framework ................................ 323
      7.2.1 GF Examples ........................................ 324
   7.3 System Overview .......................................... 326
   7.4 The Multilingual Editor ................................... 327
      7.4.1 Syntax-Directed Editing ............................... 327
      7.4.2 Top-Down Editing: Refinement ........................ 327
      7.4.3 Bottom-Up Editing: Wrapping ........................... 328
      7.4.4 Other Editor Features ................................ 329
      7.4.5 Expressions and Sentences ............................. 329
      7.4.6 Subtyping ............................................ 330
   7.5 Translation of Domain Specific Concepts ..................... 331
      7.5.1 Grammar Generation ................................... 331
      7.5.2 Customising the Translation ............................ 331
   7.6 Further Reading .......................................... 332
   7.7 Summary ................................................ 333
8 Proof Obligations
by Andreas Roth ........................................ 335
8.1 Design Validation ..................................... 337
  8.1.1 Disjoint Preconditions ............................ 337
  8.1.2 Behavioural Subtyping of Invariants ............... 338
  8.1.3 Behavioural Subtyping of Operations ............... 339
  8.1.4 Strong Operation Contract ......................... 342
8.2 Observed-State Correctness ........................... 344
  8.2.1 Observed States vs. Visible States ................. 345
  8.2.2 Assumptions Before Operation Calls ................. 348
  8.2.3 Operation Calls .................................. 349
  8.2.4 Assertions After Operation Calls .................... 351
  8.2.5 Static Initialisation ................................ 353
8.3 Lightweight Program Correctness ....................... 355
  8.3.1 Invariants ........................................ 355
  8.3.2 Postconditions and Termination ..................... 356
  8.3.3 Modifies Clauses .................................. 357
8.4 Proving Entire Correctness ............................ 359
8.5 Modular Verification .................................... 363
  8.5.1 Visibility-Based Approach ......................... 363
  8.5.2 Encapsulation-Based Approach ....................... 365
  8.5.3 Verification Strategies ............................ 371
  8.5.4 Components and Modular Proofs ...................... 372

9 From Sequential JAVA to JAVA CARD
by Wojciech Mostowski .................................... 375
9.1 Introduction ........................................... 375
9.2 Motivation ............................................ 376
9.3 JAVA CARD Memory, Atomicity, and Transactions ........ 377
9.4 Strong Invariants: The “Throughout” Modality ........ 379
  9.4.1 Additional Calculus Rules for “Throughout” ......... 380
9.5 Handling Transactions in the Logic ..................... 382
  9.5.1 Rules for Beginning and Ending a Transaction .. 382
  9.5.2 Rules for Conditional Assignment ................... 386
9.6 Examples ............................................. 387
9.7 Non-atomic JAVA CARD API Methods .................... 392
  9.7.1 Transaction Suspending and Resuming ................ 394
  9.7.2 Conditional Assignments Revised .................... 396
9.8 Summary ............................................. 398
  9.8.1 Related Work ....................................... 398
9.9 Implementation of the Rules .......................... 399
  9.9.1 New Modalities ..................................... 399
  9.9.2 Transaction Statements and Special Methods ....... 399
  9.9.3 Taclet Options ..................................... 402
  9.9.4 Implicit Fields ..................................... 402
Part III Using the KeY System

10 Using KeY
by Wolfgang Ahrendt

10.1 Introduction
10.2 Exploring Framework and System Simultaneously
  10.2.1 Exploring Basic Notions And Usage
  10.2.2 Exploring Terms, Quantification, and Instantiation
10.3 Generating Proof Obligations

11 Proving by Induction
by Angela Wallenburg

11.1 Introduction
11.2 The Need for Induction
  11.2.1 A First Look at an Induction Rule
  11.2.2 A Small Example
11.3 Basics of Induction in KeY
  11.3.1 Induction Rule
  11.3.2 Induction Variable
  11.3.3 Induction Formula
  11.3.4 Induction Principle
11.4 A Simple Program Loop Example
  11.4.1 Preparing the Proof
  11.4.2 The Proof in JAVA CARD DL
  11.4.3 Making the Proof in the KeY System
11.5 Choosing the Induction Variable
  11.5.1 The Difficulty of Guiding Induction Proofs
  11.5.2 How to Choose the Induction Variable
11.6 Different Induction Rules
  11.6.1 Customised Induction Rules
  11.6.2 The Noetherian Induction Rule
  11.6.3 Soundness of Induction Rules
11.7 Generalisation of Induction Formulae
  11.7.1 Cubic Sum Example
11.8 Summary: The Induction Proving Process
11.9 Conclusion
14.5.2 Security Properties ........................................ 543
14.5.3 Only ISOExceptions at Top Level ...................... 545
14.5.4 Atomicity and Transactions ............................. 557
14.5.5 No Unwanted Overflow .................................. 563
14.5.6 Other Properties .......................................... 564
14.6 Lessons ....................................................... 565
14.6.1 Related Work ............................................... 567

15 The Schorr-Waite-Algorithm
by Richard Bubel .................................................. 569
15.1 The Algorithm in Detail ..................................... 569
15.1.1 In Theory ................................................... 569
15.1.2 In Practice ................................................ 571
15.2 Specifying Schorr-Waite .................................... 573
15.2.1 Specifying Reachability Properties ..................... 574
15.2.2 Specification in JAVA CARD DL ......................... 578
15.3 Verification of Schorr-Waite Within KeY .................. 582
15.3.1 Replacing Arguments of Non-rigid Functions Behind Updates ........................................ 583
15.3.2 The Proof ................................................ 584
15.4 Related Work ................................................ 586

A Predefined Operators in JAVA CARD DL
by Steffen Schlager ................................................. 591
A.1 Syntax ........................................................ 591
A.1.1 Built-in Rigid Function Symbols .......................... 591
A.1.2 Built-in Rigid Function Symbols whose Semantics Depends on the Chosen Integer Semantics .......... 592
A.1.3 Built-in Non-Rigid Function Symbols .................... 593
A.1.4 Built-in Rigid Predicate Symbols ......................... 594
A.1.5 Built-in Rigid Predicate Symbols whose Semantics Depends on the Chosen Integer Semantics .......... 594
A.1.6 Built-in Non-rigid Predicate Symbols .................... 595
A.2 Semantics ..................................................... 595
A.2.1 Semantics of Built-in Rigid Function Symbols .......... 595
A.2.2 Semantics of Built-in Predicate Symbols ................ 597

B The KeY Syntax
by Wojciech Mostowski ........................................... 599
B.1 Notation, Keywords, Identifiers, Numbers, Strings ........ 600
B.2 Terms and Formulae ........................................... 602
B.2.1 Logic Operators ............................................ 602
B.2.2 Atomic Terms .............................................. 605
B.3 Rule Files .................................................... 612
B.3.1 Library and File Inclusion ................................. 612