

Contents

| | |
|--|------------|
| Abstract | v |
| Resumen | vii |
| Resum | ix |
| | |
| I Introduction | 1 |
| 1 Preamble | 3 |
| 1.1 Motivation | 3 |
| 1.2 Analysis Techniques | 5 |
| 1.3 Contributions and Main Goals | 10 |
| 1.4 Structure of this thesis | 13 |
| | |
| II Program Slicing | 17 |
| 2 Preliminary Definitions and Notation | 19 |
| 2.1 Program Slicing with System Dependence Graphs | 22 |
| 2.2 Program Slicing Algorithm | 24 |
| 2.3 Program Slicing of Object-Oriented Programs | 26 |
| 2.4 Language-Independent Program Slicing: Observation-Based Slicing | 29 |
| 3 Flow Dependence for Java Object-Oriented Programs | 31 |
| 3.1 Limitations of the JSysDG | 31 |
| 3.2 A novel definition of flow dependence | 33 |
| 3.3 Implementation | 47 |
| 3.4 Experimental Results | 50 |
| 3.5 Related Work | 53 |
| 4 Field-Sensitive Slicing with Constrained Graphs | 57 |
| 4.1 The CE-PDG | 59 |
| 4.2 Dealing with recursive data structures | 62 |
| 4.3 Slicing the CE-PDG | 63 |
| 4.4 The CE-SDG | 79 |
| 4.4.1 Summary edges and grammar productions | 80 |
| 4.4.2 Summary constraints for unknown source code functions | 84 |
| 4.4.3 Dealing with recursion | 85 |
| 4.4.4 Slicing the CE-SDG | 86 |

| | | |
|------------|---|------------|
| 4.5 | Implementation | 88 |
| 4.6 | Experimental Results | 88 |
| 4.7 | Related Work | 92 |
| 5 | Overcoming SDG Limits: The Expression Dependence Graph | 95 |
| 5.1 | Representation problems of the SDG | 95 |
| 5.2 | From ASTs to EDGs | 100 |
| 5.3 | Slicing the EDG | 113 |
| 5.4 | Solving SDG limitations | 113 |
| 5.5 | Implementation | 117 |
| 5.6 | Empirical evaluation | 119 |
| 5.7 | Related Work | 122 |
| 6 | Quasi-Minimal Slicing to Compare Program Slicers | 125 |
| 6.1 | Using ASTs to Improve Granularity | 127 |
| 6.2 | A Method to Produce Quasi-Minimal Slices | 128 |
| 6.2.1 | Phase 1: Combining static program slicers | 128 |
| 6.2.2 | Phase 2: Increasing precision via an AST-adapted ORBS algorithm | 130 |
| 6.3 | Implementation | 134 |
| 6.3.1 | Phase 1: <i>Slicerl</i> and <i>e-Knife</i> | 134 |
| 6.3.2 | Phase 2: CutEr, Cover, and SecEr | 134 |
| 6.4 | Experimental Evaluation and Results | 136 |
| 6.4.1 | Phase 1: Behaviour of Slicerl and e-Knife | 136 |
| 6.4.2 | Phase 2: Behaviour of ORBS and CutEr | 136 |
| 6.4.3 | Empirical evaluation | 138 |
| 6.4.4 | A suite of minimal slices | 141 |
| 6.5 | Related Work | 143 |
| III | Testing & Verification | 145 |
| 7 | Preliminary Definitions and Notation | 147 |
| 7.1 | Analysis tools for Erlang | 147 |
| 7.1.1 | Type inference in Erlang: TypEr | 147 |
| 7.1.2 | Property-based testing in Erlang: PropEr | 148 |
| 7.1.3 | Concolic testing in Erlang: CutEr | 148 |
| 7.2 | Design by contract | 149 |
| 8 | Software Evolution Control in Erlang | 151 |
| 8.1 | A novel approach to Automatic Regression Testing: Point of Interest Testing | 153 |
| 8.2 | Traced information in POI testing | 155 |
| 8.2.1 | Possible POI testing configurations | 157 |
| 8.3 | POI testing adapted to Erlang | 159 |
| 8.3.1 | Initial ITC generation | 159 |
| 8.3.2 | Recording the traces of the point of interest | 161 |
| 8.3.3 | Extraction of additional trace information | 166 |

| | | |
|-----------|---|------------|
| 8.3.4 | Test case generation using ITC mutation | 170 |
| 8.4 | POI testing with multiple POIs | 171 |
| 8.4.1 | Code transformation with multiple POIs | 172 |
| 8.4.2 | Differences in trace equality | 172 |
| 8.5 | POI testing in concurrent environments | 175 |
| 8.6 | Implementation | 175 |
| 8.7 | Experimental Evaluation | 178 |
| 8.8 | Related Work | 180 |
| 9 | Design-by-contract verification in Erlang | 185 |
| 9.1 | Contracts in Erlang | 186 |
| 9.1.1 | Contracts for sequential Erlang | 186 |
| 9.1.2 | Contracts for concurrent Erlang | 192 |
| 9.2 | Implementation | 198 |
| 9.2.1 | Architecture | 198 |
| 9.2.2 | Instrumentation | 200 |
| 9.3 | Related Work | 202 |
| IV | Developed Tools | 205 |
| 10 | Developed Tools and User Guides | 207 |
| 10.1 | JavaSlicer | 207 |
| 10.1.1 | Installation and first steps | 207 |
| 10.1.2 | Use case | 210 |
| 10.2 | e-Knife (a CE-EDG Slicer for Erlang) | 211 |
| 10.2.1 | Installation and first steps | 212 |
| 10.2.2 | Use case | 214 |
| 10.3 | SecEr | 216 |
| 10.3.1 | Installation and first steps | 216 |
| 10.3.2 | Use cases | 223 |
| 10.4 | EDBC | 231 |
| 10.4.1 | Installation and first steps | 232 |
| 10.4.2 | Use cases | 235 |
| V | Conclusions and Future Research | 239 |
| 11 | Conclusions | 241 |
| 12 | Future Lines of Research | 247 |
| 12.1 | Program Slicing | 247 |
| 12.2 | Testing and Verification | 249 |
| | Bibliography | 251 |