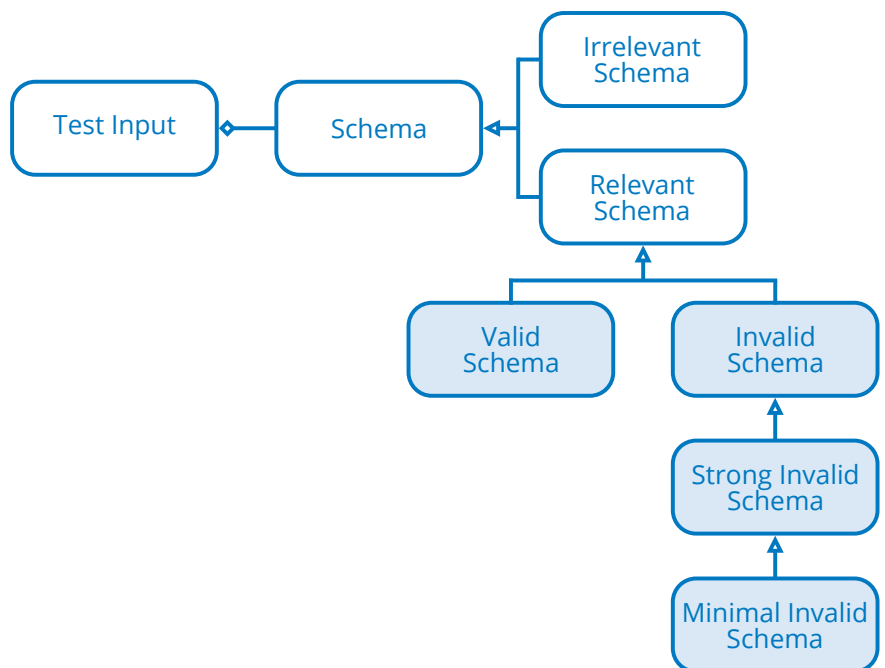


Konrad Anton Fögen

Combinatorial Robustness Testing based on Error-Constraints



Combinatorial Robustness Testing based on Error-Constraints

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der
RWTH Aachen University zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Konrad Anton Fögen, M. Sc.
aus Höxter

Berichter: Universitätsprofessor Dr. rer. nat. Horst Lichter
Professor Angelo Gargantini, Ph.D.

Tag der mündlichen Prüfung: 10. Februar 2021

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek verfügbar.

Aachener Informatik-Berichte, Software Engineering

herausgegeben von
Prof. Dr. rer. nat. Bernhard Rumpe
Software Engineering
RWTH Aachen University

Band 47

Konrad Anton Fögen
RWTH Aachen University

**Combinatorial Robustness Testing
based on Error-Constraints**

Shaker Verlag
Düren 2021

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

Zugl.: D 82 (Diss. RWTH Aachen University, 2021)

Copyright Shaker Verlag 2021

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in Germany.

ISBN 978-3-8440-7929-6

ISSN 1869-9170

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren

Phone: 0049/2421/99011-0 • Telefax: 0049/2421/99011-9

Internet: www.shaker.de • e-mail: info@shaker.de

Kurzfassung

Robustheit ist eine wichtige Eigenschaft einer Software, welche zusätzlich zur Funktionalität getestet werden muss. Dies erfordert ungültige Werte und ungültige Wertekombinationen, um die Reaktion einer Software auf die Eingabe dieser Werte beobachten zu können.

Kombinatorisches Testen (CT) ist eine effektive spezifikationsbasierte Testmethode, die auf einem Eingabeparametermodell (IPM) basiert, mit dem Eingaben für das Testen ausgewählt werden. Die Effektivität von CT verschlechtert sich jedoch, wenn ungültige Werte oder ungültige Wertekombinationen vorhanden sind. Dieses Phänomen wird als Maskierungseffekt ungültiger Eingaben bezeichnet und ist in der Forschung bereits bekannt. Das Phänomen führte zu Erweiterungen der CT Testmethode, welche wir als Kombinatorisches Testen für Robustheit (CRT) bezeichnen. Das Ziel von CRT ist es, das Erkennen von Fehlern zu verbessern, indem der Maskierungseffekt ungültiger Eingaben vermieden wird. Vermieden wird er durch die Trennung von Tests mit gültigen Werten und gültigen Wertekombinationen und Tests mit ungültigen Werten und ungültigen Wertekombinationen.

Obwohl CRT eine vielversprechende Erweiterung von CT ist, argumentieren wir, dass sie immer noch unzureichend erforscht ist. Zum Beispiel wird in verwandten Arbeiten das IPM mit semantischen Informationen erweitert, um Werte als ungültig markieren zu können. Wertekombinationen können allerdings nicht direkt als ungültig markiert werden.

In dieser Arbeit entwickeln wir daher die Idee des CRT weiter. Das Ziel ist eine neue CRT Testmethode mit einem Modellierungsansatz, um ungültige Werte und ungültige Wertkombinationen gleichermaßen gut zu spezifizieren. Dieser Modellierungsansatz soll auch in explizite Testabdeckungskriterien und Testauswahlstrategien integriert werden. Zudem soll dieser Modellierungsansatz durch automatisierte Techniken weiter unterstützt werden.

Zunächst führen wir ein kontrolliertes Experiment durch, um zu überprüfen, ob CRT als Erweiterung notwendig ist, oder ob CT bereits geeignet ist, um Robustheit von Software zu testen. Basierend auf den Ergebnissen verfeinern wir das t -Faktor Fehlermodell, sodass Robustheitsfehler und der inhärente Maskierungseffekt ungültiger Eingaben berücksichtigt werden.

Dann entwickeln wir eine neue Testmethode für CRT und führen das Robustheits-Eingabeparametermodell (RIPM) ein, welches die Struktur der IPMs um das Konzept der Error-Constraints erweitert. Dabei handelt es sich um einen zusätzlichen Satz logischer Ausdrücke zur Beschreibung der Gültigkeit von Werten und Wertekombinationen.

Mit dem verfeinerten t -Faktor Fehlermodell und der neuen RIPM Struktur werden neue Testabdeckungskriterien, welche die zusätzlichen semantischen Informationen einbeziehen, und neue Testauswahlstrategien, welche die Testabdeckungskriterien erfüllen, entwickelt.

Das neue Konzept der Error-Constraints erfordert zusätzlichen Aufwand. Daher entwickeln wir zwei zusätzliche Techniken, welche die Modellierung der Error-Constraints unterstützen. Zuerst entwickeln wir eine Technik zur Identifizierung und Reparatur von Inkonsistenz innerhalb der Error-Constraints. Zudem entwickeln wir eine Technik zur automatischen Generierung von Error-Constraints, welche auf der Konformität mit einem anderen System basiert.

Zu guter Letzt werden alle oben genannten Konzepte und Techniken operationalisiert und in ein Framework zu Testautomatisierung integriert, welches einen Prozess, eine Architektur und eine Java-basierte Referenzimplementierung umfasst.

Abstract

Robustness is an important property of a software, which must be tested in addition to a software's functionality. This requires invalid values and invalid value combinations to be able to observe a software's reaction to them.

Combinatorial testing (CT) is an effective specification-based test method that is based on an input parameter model (IPM) with which test inputs are selected. But its effectiveness deteriorates in the presence of invalid values or invalid value combinations. This phenomenon is called invalid input masking effect and is already acknowledged in some research. The phenomenon led to extensions of CT that we call combinatorial robustness testing (CRT). The objective of CRT is to improve the fault detection by avoiding invalid input masking. This is achieved by separating the testing of valid values and valid value combinations from the testing of invalid values and invalid value combinations.

While CRT is a promising extension of CT, it is still insufficiently researched. For instance, in related work, IPMs are extended with additional semantic information to specify invalid values. However, invalid value combinations cannot be specified directly.

Therefore, the objective of this work is to further expand the idea of CRT. The aim is to develop a new CRT test method with a modeling approach to specify invalid values and invalid value combinations equally well. This modeling approach should also be incorporated into explicit test adequacy criteria and test selection strategies. Furthermore, this modeling approach shall be supported by automated techniques.

First, we conduct a controlled experiment to check if CRT is necessary at all or if CT is already appropriate to test robustness. Based on the result, we continue and develop a refined t -factor fault model that incorporates robustness faults and the inherent invalid input masking effect.

Next, we develop a new test method for CRT and introduce a new structure that extends the structure of IPMs. It is called robustness input parameter model (RIPM) and contains the concept of error-constraints which is an additional set of logical expressions to describe the validity of values and value combinations.

With the refined t -factor fault model and the new RIPM structure, new test adequacy criteria that incorporate the additional semantic information and new test selection strategies that satisfy the test adequacy criteria are developed.

The new concept of error-constraints requires additional effort in modeling. Therefore, we develop two techniques to support the modeling of them. First, we develop a technique to identify and repair inconsistencies among error-constraints. Second, we develop a technique to automatically generate error-constraints based on the conformance to another system.

Last but not least, all aforementioned concepts and techniques are operationalized and integrated in a test automation framework which includes a process, an architecture, and a Java-based reference implementation.

Contents

I. Foundations	1
1. Introduction	3
1.1. Research Motivation	3
1.2. Research Context	5
1.3. Research Objective	6
1.4. Structure of this Work	9
1.5. List of Publications	11
2. Running Example: Ordering Web Service	13
3. Conceptual Foundations	17
3.1. A Taxonomy of Dependable Programs	17
3.1.1. Dependable Programs	17
3.1.2. Combinatorial Testing	25
3.2. An Extended Taxonomy of Robust Programs	38
3.2.1. Robust Programs	38
3.2.2. Exception Handling	43
3.2.3. Combinatorial Robustness Testing	50
4. Related Work	55
4.1. Combinatorial Robustness Testing	55
4.1.1. General Contributions	56
4.1.2. CT Test Methods	58
4.1.3. CRT Test Methods	60
4.2. Non-Combinatorial Robustness Testing	69
II. A Combinatorial Robustness Test Method	71
5. A Combinatorial Robustness Fault Model	73
5.1. Fault Detection Effectiveness in the Presence of Invalid Input Masking	74
5.1.1. Motivation	74
5.1.2. Applying the t-Factor Fault Model	75
5.1.3. Experiment Design	81
5.1.4. Results and Discussion	84
5.1.5. Threats to Validity	90

5.2.	A Classification of Robustness Fault Characteristics	91
5.2.1.	Configuration-independent Robustness Faults	91
5.2.2.	Configuration-dependent Robustness Faults	94
5.3.	A Case Study on Minimal Failure-causing Schemata	97
5.3.1.	Motivation	97
5.3.2.	Case Study Design	98
5.3.3.	Results and Discussion	100
5.3.4.	Threats to Validity	104
5.4.	A Refined t-Factor Fault Model for Robustness Faults	105
6.	A Combinatorial Robustness Input Parameter Model	111
7.	Combinatorial Robustness Test Adequacy Criteria	119
7.1.	Existing t-wise Combinatorial Test Adequacy Criteria	119
7.2.	Combinatorial Test Adequacy Criteria for Strong Invalid Test Inputs	127
8.	Combinatorial Robustness Test Selection Strategies	143
8.1.	Test Selecting Strategies for Valid Test Inputs	143
8.2.	Test Selection Strategies for Strong Invalid Test Inputs	147
III.	Supporting Techniques for Combinatorial Robustness Testing	157
9.	Detection and Repair of Over-Constrained RIPMs	159
9.1.	Detection and Manual Repair of Over-Constrained RIPMs	160
9.1.1.	A Process for Detection and Manual Repair	160
9.1.2.	Identifying Missing Invalid Schemata	164
9.1.3.	Explaining Conflicting Constraints	166
9.2.	Semi-Automatic Repair of Over-Constrained RIPMs	169
9.2.1.	Automatic Diagnosis of Over-Constrained RIPMs	170
9.2.2.	Automatic Relaxation of Conflicting Constraints	174
9.2.3.	Selection and Application of Diagnosis Hitting Sets	177
9.3.	Strong Invalid Test Input Selection from Over-Constrained RIPMs	178
9.3.1.	General Idea of Soft Constraint Handling Strategies	178
9.3.2.	Basic Soft Constraint Handling Strategy	180
9.3.3.	Diagnostic Soft Constraint Handling Strategy	182
10.	Automatic Generation of Error-Constraints	187
10.1.	A Process for Automatic Generating of Error-Constraints	187
10.2.	Identifying Non-Conforming Schemata	191
10.3.	Generating New Error-Constraints	196
11.	A Framework for Automated Combinatorial Robustness Testing	199
11.1.	Orchestrated Process	199
11.2.	Framework Architecture	203

11.3. Implementation of coffee4j	205
IV. Evaluation and Conclusion	209
12. Overview of Evaluation	211
13. Evaluating Combinatorial Robustness Test Selection Strategies	215
13.1. Theoretical Evaluation	215
13.2. Experimental Evaluation	219
13.2.1. Motivation	219
13.2.2. Experiment Design	219
13.2.3. Experiment Setup	220
13.2.4. Experiment Scenarios	221
13.2.5. Results and Discussion	223
13.2.6. Threats to Validity	226
13.3. Case Study-based Evaluation	227
13.3.1. Motivation	227
13.3.2. Case Study Design	228
13.3.3. Results and Discussion	230
13.3.4. Threats to Validity	242
14. Evaluating Detection and Repair of Over-constrained RIPMs	245
14.1. Evaluating the Detection and Manual Repair of Over-Constrained RIPMs	245
14.1.1. Experiment Design and Setup	245
14.1.2. Results and Discussion	246
14.1.3. Threats to Validity	248
14.2. Evaluating Semi-Automatic Repair of Over-Constrained RIPMs	249
14.2.1. Experiment Design and Setup	249
14.2.2. Results and Discussion	249
14.2.3. Threats to Validity	251
14.3. Evaluating Strong Invalid Test Input Selection from Over-Constrained RIPMs	253
14.3.1. Experiment Design and Setup	253
14.3.2. Results and Discussion	254
14.3.3. Threats to Validity	258
15. Evaluating Automatic Generation of Error-Constraints	259
15.1. Experiment Design and Setup	259
15.1.1. Fault Characterization Algorithms	260
15.1.2. Scenarios	260
15.1.3. Experiments	261
15.1.4. Data Collection Procedure	262
15.2. Results and Discussion	263
15.2.1. Experiment 1	263

15.2.2. Experiment 2	264
15.2.3. Experiment 3	265
15.2.4. Experiment 4	265
15.2.5. Summary	267
15.3. Threats to Validity	269
16. Conclusion	271
16.1. Summary	271
16.2. Future Work	274
V. Appendix	277
A. Data of FDE in the Presence of Invalid Input Masking	279
B. Data of Evaluating Combinatorial Robustness Test Selection Strategies	335
C. Data of Evaluating Semi-Automatic Repair of Over-Constrained RIPMs	341
D. Data of Evaluating Automatic Generation of Error-Constraints	345
Abbreviations	351
Symbols and Functions	353
Glossary	357
List of Figures	365
List of Listings	367
List of Tables	369
Supervised Bachelor and Master Theses	375
Bibliography	377