



M. Gogolla, H.-J. Kreowski, B. Krieg-Brückner,
J. Peleska, B.-H. Schlingloff, H. Szczerbicka (Series Editors)

A GENERIC CALCULUS OF TRANSFORMATIONS

Burkhart Wolff

Revised Version, July 1999

Vom Fachbereich Mathematik und Informatik
der Universität Bremen
zur Verleihung des akademischen Grades eines
Doktors der Ingenieurwissenschaften (Dr. Ing.)
genehmigte Dissertation

Gutachter: Prof. Dr. Bernd Krieg-Brückner
Prof. Dr. Tobias Nipkow

Kolloquium: 16. Juli 1997

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Wolff, Burkhard:

A Calculus of Transformations / Burkhard Wolff.

- Als Ms. gedr. - Aachen : Shaker, 1999

(BISS Monographs ; Bd. 4)

Zugl.: Bremen, Univ., Diss., 1997

ISBN 3-8265-3654-1

Contact Address

Prof. Dr. Bernd Krieg-Brückner
Bremen Institute of Safe Systems
TZI, FB3 Mathematik und Informatik
Universität Bremen
Postfach 330 440
D - 28334 Bremen

Tel.: (+49) 421-218-3660
Fax: (+49) 421-218-3054

Telex: 245 811 Uni D
biss@informatik.uni-bremen.de
<http://www.informatik.uni-bremen.de/~biss>

Copyright Shaker Verlag 1999

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISBN 3-8265-3654-1

ISSN 1435-8611

Shaker Verlag GmbH • Postfach 1290 • 52013 Aachen
Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9
Internet: www.shaker.de • eMail: info@shaker.de

Editorial

The area of correct software development with formal methods has made significant progress in the past years, particularly in the area of system support. Efficient systems for interactive proofs have been created such as the tactical proof system Isabelle with its Higher-Order Logic (HOL) instantiation. In the context of the project UniForM Workbench, sponsored by the German Ministry of Research, BMBF, several extensions have been developed, in particular for development by transformation and the proof of correctness of transformation rules, also w.r.t. graphical user interfaces. HOL allows a fairly readable representation of object logics; a completely formal representation and a correctness proof of the theories (such as the process algebra of CSP) that is completely supported and controlled by the system have become realistic. Burkhart Wolff was involved in all these developments in a leading role.

Thus quality assurance in software development will make a big leap forward: the user can formally specify programs, prove essential properties in a formal way, and develop them correctly by stepwise transformation. However, there is always the challenge: how safe or rather formally correct are the tools?

Burkhart Wolff's dissertation starts here. Motivated by his theoretical and practical experiences in the BMBF-projects KORSO and UniForM, he develops a generic framework that allows a flexible, direct presentation of contextual bindings (environments, type contexts with dependent types, bindings to signatures, parametrisation, etc.) in the representation of the object logic. Thus meta-variables can be bound to such objects. Side-conditions for the integrity of bindings are considered in a generalised notion of *filling* (schema substitution). The common notions of *term rewriting* are formalised in this basic calculus; this finally leads to the desired concept of *transformation* as an Isabelle/HOL theory.

This way, the result of the work has theoretical relevance to begin with: the essential results and similar approaches from the literature can be subsumed in the new theory; relevant *algebraic properties* of the operations introduced, and the *confluence* of orthogonal transformations could be proved. It has to be emphasised especially that the sizeable structured set of core-theories (ca. 250 theorems and lemmata) has not only been proved in a mathematically rigorous way, but *formally proved*, i.e. supported and controlled by the proof system Isabelle. Moreover, the result has practical potential for the implementation of tools, e.g. the generation of efficient dedicated theorem provers for given object languages by partial evaluation or transformational optimisation.

The dissertation is now available in a revised version.

July 1999

Prof. Dr. Bernd Krieg-Brückner

Vorwort des Herausgebers

Das Gebiet der Entwicklung korrekter Software mit formalen Methoden hat in den letzten Jahren bedeutende Fortschritte gemacht, erfreulicherweise besonders im Bereich der Systemunterstützung. Für das interaktive Beweisen sind effiziente Systeme entstanden, wie z.B. das taktische Beweissystem Isabelle mit der Higher-Order Logic (HOL) Instanziierung. Im Rahmen des BMBF-Projektes UniForM Workbench wurden einige Erweiterungen entwickelt, insbesondere für die Entwicklung durch Transformation und den Beweis der Korrektheit der Transformationsregeln sowie grafische Benutzungsoberflächen. HOL erlaubt eine recht gut lesbare Repräsentation von Objekt-Logiken; die vollständig formale Repräsentation und der durchgängig vom System unterstützte und überprüfte Beweis der Korrektheit von Theorien (wie zum Beispiel der Prozeß-Algebra von CSP) sind realistisch geworden. An all diesen Entwicklungen war Burkhard Wolff maßgeblich beteiligt.

Die Qualitätssicherung der Software-Entwicklung wird so also einen großen Schritt weiterkommen: der Benutzer kann Programme formal spezifizieren, wesentliche Eigenschaften formal beweisen und durch Transformationen schrittweise korrekt entwickeln. Allerdings stellt sich aber auch immer wieder die Herausforderung: wie sicher bzw. formal korrekt sind denn die Werkzeuge?

Hier setzt die Dissertation von Burkhard Wolff an. Von seinen theoretischen und praktischen Erfahrungen in den BMBF-Projekten KORSO und UniForM motiviert, entwickelt er ein generisches Rahmenwerk, das eine flexible, direkte Präsentation von kontextuellen Bindungen (Umgebungen, Typ-Kontexten mit abhängigen Typen, Bindungen an Signaturen, Parametrisierung etc.) innerhalb der Objektlogik-Repräsentation erlaubt. Somit können Meta-Variablen an derartige Objekte gebunden werden. Nebenbedingungen für die Integrität von Bindungen werden in einem verallgemeinerten Begriff des *fillings* (i.e. der Schema-Substitution) berücksichtigt. Dabei bleiben die Vorteile einer sog. flachen Kodierung teilweise erhalten. In dem so entstandenen Basis-Kalkül werden die gängigen Begriffe der *Termersetzung* formalisiert; dies führt schließlich zu dem erstrebten Konzept der *Transformation* als Isabelle/HOL Theorie.

Damit hat das Ergebnis der Arbeit zunächst eine theoretische Bedeutung: die wesentlichen Ergebnisse und ähnliche Ansätze aus der Literatur können in die neue Theorie eingeordnet werden; wesentliche *algebraische Eigenschaften* der eingeführten Operationen und die *Konfluenz* von orthogonalen Transformationen konnten gezeigt werden. Es muß besonders betont werden, daß das der Kern des umfangreichen Theoriegebäudes (ca. 250 Theoreme und Lemmata) nicht nur mathematisch rigoros, sondern *formal*, d.h. vom Beweissystem Isabelle unterstützt und überprüft, *bewiesen* wurde. Ferner hat das Ergebnis praktisches Potential für die Implementierung von Werkzeugen, z.B. die Generierung von effizienten, dedizierten Theorembeweisern für gegebene Objektsprachen durch partielle Auswertung bzw. transformationelle Optimierung.

Die Dissertation liegt nun in einer überarbeiteten Fassung vor.

Abstract

Binding structures enrich traditional abstract syntax by providing support for representing binding mechanisms (based on deBruijn indices), term-schemata and a very clean algebraic theory of substitution. We provide the following main results:

- 1.) The establishment of a *generic* binding structure with the novel concept of *effect-binding* that enables the representations of both signatures and formulas (i.e. specifications) inside one term meta-language,
- 2.) The foundation of a formal (machine-checked) *substitution theory* of effect-binding. The substitution theory offers two views on its subject: an operational and an algebraic one; hence, it is well-suited both for mechanisation and meta-theoretic reasoning. This may contribute to the construction of tools such as theorem provers, program transformers, static analysers, evaluators and optimising compilers,
- 3.) The foundation of a rigorous meta-theory for rewriting on effectbinding-structures. The resulting rewrite notion *transformation* extends combinatory rewrite systems to rewrites on specifications.

The corner stone of this theory is a confluence proof for orthogonal transformations (partly implemented in the proof assistant Isabelle).

Zusammenfassung

Bindungsstrukturen bestehen zum einen aus traditionellen abstrakten Syntaxen und zum anderen aus Bindungsmechanismen (basierend auf deBruijn-Indizes) samt einer algebraischen Theorie der Substitution. Außerdem sehen sie Mechanismen zur Bildung von Termschemata vor.

Diese Arbeit liefert die folgenden Hauptbeiträge:

- 1.) Die Entwicklung einer *generischen* Bindungsstruktur mit dem neuartigen Konzept der Effektbindung, was sowohl die Repräsentation von Signaturen als auch Formeln (d.h., Spezifikationen) in einer kompakten Termsprache ermöglicht.
- 2.) Die Untersuchung einer formalen (maschinell überprüften) *Theorie der Substitution* für Effektbindung, die für die Mechanisierung geeignet ist. Dies kann als theoretische und praktische Grundlage zur Konstruktion von Werkzeugen wie Theorem-Beweisern, Programm-Transformationssystemen, statischen Analysesystemen, partiellen Auswertern oder optimierenden Compilern dienen.
- 3.) Die rigorose Begründung einer Termersetzungstheorie über Effektbindungsstrukturen. Der resultierende Begriff der *Transformation* erweitert den der kombinatorischen Reduktionssysteme zu Ersetzungssystemen auf Spezifikationen.

Der Schlußstein dieser Theorie (in Teilen implementiert im Theorembeweiser Isabelle) ist der Nachweis, daß orthogonale Transformationssysteme konfluent sind.

Contents

1 Introduction	1
1.1 Representing Logical Languages	1
1.2 Rewriting Notions.....	11
1.3 Why Formal Meta-theory?	18
1.4 Overview	20
2 Foundational Notions and the Technical Setting.....	21
2.1 Higher Order Logic	21
2.2 Conservative Extension Schemes	25
2.3 The HOL-Library.....	26
2.4 Isabelle.....	27
2.5 Theorem Proving in Isabelle	29
2.6 Ida.....	31
Part I	39
3. T-Terms, Induction and Recursion.....	40
3.1 The S-Expression Model for α T.....	41
3.2. The Type Definition of α T	42
3.3. Induction and Recursion over α T	44
3.4 The Class of Binding Operators	46
3.5 Example of an Instantiation	48
3.6 Summary	49
4. Adjustments.....	50
4.1 The Binding Morphism in Theory TBind	51
4.2 The Instantiation of the Binding Morphism for Adjustments	53
4.3 Multiple Adjustments	59
4.4 Cyclic Adjustments.....	62
4.5 Collecting Information.....	63
4.6 Normal Forms of m-adjustments.....	66
4.7 Summary	68
5. On Substitution Closures.....	69
5.1 Substitution Closures	69
5.2 Operations Shift and Spread	71

5.3 Extensions to Multi-Operations	75
5.4 Summary	77
6. Substitutions	78
6.1 Drive and Substitution	78
6.2 Example: Derivation of λv	85
6.3 Example: Standard deBruijn- λ -Calculus Substitution	87
6.4 Example: T-Structures and Explicit Names	87
6.5 Summary	88
7. Filling Closures	90
7.1 Filling Closures	91
7.2 Shifts on Filling Closures	92
7.3 Openings on Filling Closures	93
7.4 Conclusion	94
8. On Fillings (Substitutions on Schema-Variables)	95
8.1 Filling Drive	97
8.2 Schema-Substitution	98
8.3 Linear Conformity	98
8.4 Algebraic Properties of Filling	99
8.5 Summary	100
Part II	101
9. Transformations	103
9.1 Basic Syntactic Notions for Rules	103
9.2 A Labelled Inference System for Transformational Rewriting	106
9.3 More on Syntax: Orthogonality	111
9.4 Abstract Reduction Systems	112
9.5 Summary	115
10. Confluence for Orthogonal Systems	116
10.1 Internalizing Filling Closures	117
10.2 Quasi-Congruence of Transformational Rewriting	119
10.3 Parallel Moves Rewriting Relation	123
10.4 Parallel Moves includes Rew_rel	124
10.5 The Transitive Closure of Rew_rel includes Parallel Moves	125
10.6 Parallel Moves is Diamond	128
10.7 Example: β -reduction is Confluent	131

10.8 Example: Micro-ML	132
10.9 Example: Module-Systems	134
10.10 Example: Program Development Calculi	135
10.11 Summary and a Critique.....	137
11. Conclusion.....	139
11.1 What has been achieved?.....	139
11.2 Future work	140
Bibliography	143
Index for Mathematical Symbols	149
Index (General)	151
Appendix.....	156