

Monographs in Theoretical Computer Science

An EATCS Series

Editors: W. Brauer G. Rozenberg A. Salomaa

On behalf of the European Association
for Theoretical Computer Science (EATCS)

Advisory Board: G. Ausiello M. Broy C.S. Calude
A. Condon D. Harel J. Hartmanis T. Henzinger
J. Hromkovič N. Jones T. Leighton M. Nivat
C. Papadimitriou D. Scott

H. Ehrig · K. Ehrig
U. Prange · G. Taentzer

Fundamentals of Algebraic Graph Transformation

With 41 Figures

Authors

Prof. Dr. Hartmut Ehrig
Dr. Karsten Ehrig
Ulrike Prange
Dr. Gabriele Taentzer

Technical University of Berlin
Institute for Software Engineering
and Theoretical Computer Science
Franklinstr. 28/29
10587 Berlin, Germany

ehrig@cs.tu-berlin.de
karstene@cs.tu-berlin.de
uprange@cs.tu-berlin.de
gabi@cs.tu-berlin.de

Series Editors

Prof. Dr. Wilfried Brauer
Institut für Informatik der TUM
Boltzmannstr. 3
85748 Garching, Germany
brauer@informatik.tu-muenchen.de

Prof. Dr. Grzegorz Rozenberg
Leiden Institute of Advanced
Computer Science
University of Leiden
Niels Bohrweg 1
2333 CA Leiden, The Netherlands
rozenber@liacs.nl

Prof. Dr. Arto Salomaa
Turku Centre of
Computer Science
Lemminkäisenkatu 14 A
20520 Turku, Finland
asalomaa@utu.fi

Library of Congress Control Number: 2005938509

ACM Computing Classification (1998): D.1.7, D.2.1, D.3.1, F.4.2, F.4.3, G.2.2, I.1.

ISBN-10 3-540-31187-4 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-31187-4 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com

© Springer-Verlag Berlin Heidelberg 2006
Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover Design: KunkelLopka, Heidelberg
Typesetting: by the Authors
Production: LE-TeX Jelonek, Schmidt & Vöckler GbR, Leipzig

Printed on acid-free paper 45/3142/YL 5 4 3 2 1 0

Preface

In the late 1960s and early 1970s, the concepts of graph transformation and graph grammars started to become of interest in picture processing and computer science. The main idea was to generalize well-known rewriting techniques from strings and trees to graphs, leading to graph transformations and graph grammars. In particular, the concepts of algebraic graph transformation gained considerable importance in the early years and have done so even more in the last decade. Today, algebraic graph transformation techniques are playing a central role in theoretical computer science, as well as in several applied areas, such as software engineering, concurrent and distributed systems, and visual modeling techniques and model transformations.

The aim of this book is to present the fundamentals of algebraic graph transformation techniques for the purposes of teaching, research, and development, with respect to the following aspects:

1. Fundamentals in the sense of an introduction with a detailed motivation to algebraic graph transformation, including the main constructions and results, as well as their generalization to high-level replacement systems, with a wide range of applications in computer science and related areas.
2. Fundamentals in the sense of mathematical theories, which are the basis for precise definitions, constructions, and results, and for the implementation of algebraic graph transformation in a tool environment called AGG.
3. Fundamentals in the sense of the integration of data types and process specification techniques, where the concepts of algebraic data types are integrated with graph rewriting, leading to the concept of typed attributed graph transformation.

In accordance with these aims, the book is organized in four parts:

- *Part I: Introduction to Graph Transformation Systems*, where graph transformations based on classical graphs are introduced and the main constructions and results are motivated in detail.

- *Part II: Adhesive High-Level Replacement Categories and Systems*, where the theory is presented in a categorical framework with applications to a large variety of high-level structures, especially transformation systems for various kinds of graphs and Petri nets.
- *Part III: Typed Attributed Graph Transformation Systems*, where the concepts of typed attributed graphs are carefully introduced and the main results are obtained as instantiations of Part II.
- *Part IV: Case Study on Model Transformation, and Tool Support by AGG*, where the concepts of typed attributed graph transformation are applied in a separate case study to visual model transformation, and it is shown how the theory is implemented in the AGG tool.

The book is organized in such a way that the reader can switch, after the introduction in Part I, immediately to Part III; however, the concepts and results in both of these parts are instantiations of the categorical theory presented in Part II.

The material of this book is based on a theory of algebraic graph transformation developed at the Technical University of Berlin in cooperation with several international partners in the EU projects COMPUGRAPH, GETGRATS, APPLIGRAPH and SEGRAVIS. This material can also be seen as being in the tradition of algebraic specification techniques, described in the EATCS series of Monographs in Theoretical Computer Science.

We are most thankful to Hans-Jörg Kreowski, Michael Pfender, Hans-Jürgen Schneider, Barry Rosen, and Grzegorz Rozenberg for creating the algebraic approach to graph transformation in fruitful cooperation with the first author in the 1970s. For the main contributions to the algebraic approach in subsequent years, we would like to thank in addition Paolo Baldan, Roswitha Bardohl, Paolo Bottoni, Andrea Corradini, Gregor Engels, Claudia Ermel, Ingrid Fischer, Annegret Habel, Reiko Heckel, Berthold Hoffmann, Manuel Koch, Barbara König, Martin Korff, Jochen Küster, Juan de Lara, Leen Lambers, Michael Löwe, Ugo Montanari, Mark Minas, Fernando Orejas, Julia Padberg, Karl-Heinz Pennemann, Francesco Parisi-Presicce, Detlef Plump, Leila Ribeiro, Francesca Rossi, Olga Runge, Andy Schürr, Pawel Sobociński, Daniel Varró, Szilvia Varró-Gyapay, Annika Wagner, and Dietmar Wolz.

We would especially like to thank Reiko Heckel for several useful comments concerning the overall structure of the book.

A draft version of the book was carefully studied by the participants of a compact seminar on “Fundamentals of Algebraic Graph Transformation” for advanced students and young researchers in the SEGRAVIS TMR network. We are most grateful to the following members of this seminar, whose comments led to several useful improvements in the final version of this book: Paolo Baldan, Enrico Biermann, Benjamin Braatz, Esther Guerra, Stefan Hänsen, Frank Herrmann, Markus Klein, Barbara König, Sebastian Kuhner, Juan de Lara, Tihamer Levendovsky, Katharina Mehner, Tony Modica,

Mattia Monga, Alessandra Raffaetta, Guilherme Rangel, Giovanni Toffetti Carughi, Daniel Varró, Szilvia Varró-Gyapay, and Jessica Winkelmann.

Finally, we thank Grzegorz Rozenberg and all the other editors of the EATCS monograph series, and those at Springer-Verlag, especially Ronan Nugent, for smooth publication.

Berlin, Summer 2005

Hartmut Ehrig
Karsten Ehrig
Ulrike Prange
Gabriele Taentzer

Contents

Part I Introduction to Graph Transformation Systems

1	General Introduction	5
1.1	General Overview of Graph Grammars and Graph Transformation	5
1.1.1	What Is Graph Transformation?	6
1.1.2	Aims and Paradigms of Graph Transformation	6
1.1.3	Overview of Various Approaches	9
1.2	The Main Ideas of the Algebraic Graph Transformation Approach	10
1.2.1	The DPO Approach	11
1.2.2	The Algebraic Roots	12
1.2.3	From the DPO to the SPO Approach	13
1.2.4	From Graphs to High-Level Structures	14
1.3	The Chapters of This Book and the Main Results	15
1.3.1	Part I: Introduction to Graph Transformation Systems	15
1.3.2	Part II: Adhesive HLR Categories and Systems	15
1.3.3	Part III: Typed Attributed Graph Transformation Systems	16
1.3.4	Part IV: Case Study and Tool Support	17
1.3.5	Appendices	17
1.3.6	Hints for Reading This Book	17
1.4	Bibliographic Notes and Further Topics	17
1.4.1	Concepts of Graph Grammars and Graph Transformation Systems	17
1.4.2	Application Areas of Graph Transformation Systems	19
1.4.3	Languages and Tools for Graph Transformation Systems	19
1.4.4	Future Work	20

2	Graphs, Typed Graphs, and the Gluing Construction	21
2.1	Graphs and Typed Graphs	21
2.2	Introduction to Categories	25
2.3	Pushouts as a Gluing Construction	29
2.4	Pullbacks as the Dual Construction of Pushouts	33
3	Graph Transformation Systems	37
3.1	Basic Definitions for GT Systems	37
3.2	Construction of Graph Transformations	43
3.3	Local Church–Rosser and Parallelism Theorems for GT Systems	47
3.4	Overview of Some Other Main Results for GT Systems	53
3.4.1	Concurrency Theorem	54
3.4.2	Embedding and Extension Theorems	56
3.4.3	Confluence, Local Confluence, Termination, and Critical Pairs	59
3.4.4	Functional Behavior of GT Systems and Termination Analysis	62
3.5	Graph Constraints and Application Conditions	64

Part II Adhesive High-Level Replacement Categories and Systems

4	Adhesive High-Level Replacement Categories	77
4.1	Van Kampen Squares and Adhesive Categories	77
4.2	Adhesive HLR Categories	86
4.3	HLR Properties of Adhesive HLR Categories	96
5	Adhesive High-Level Replacement Systems	101
5.1	Basic Concepts of Adhesive HLR Systems	101
5.2	Instantiation of Adhesive HLR Systems	105
5.2.1	Graph and Typed Graph Transformation Systems	106
5.2.2	Hypergraph Transformation Systems	106
5.2.3	Petri Net Transformation Systems	107
5.2.4	Algebraic Specification Transformation Systems	108
5.2.5	Typed Attributed Graph Transformation Systems	108
5.3	The Local Church–Rosser and Parallelism Theorems	109
5.4	Concurrency Theorem and Pair Factorization	117
6	Embedding and Local Confluence	125
6.1	Initial Pushouts and the Gluing Condition	125
6.2	Embedding and Extension Theorems	130
6.3	Critical Pairs	140
6.4	Local Confluence Theorem	144

7	Constraints and Application Conditions	151
7.1	Definition of Constraints and Application Conditions	152
7.2	Construction of Application Conditions from Constraints	156
7.3	Construction of Left from Right Application Conditions	160
7.4	Guaranteeing and Preservation of Constraints	164

Part III Typed Attributed Graph Transformation Systems

8	Typed Attributed Graphs	171
8.1	Attributed Graphs and Typing	172
8.2	Pushouts as a Gluing Construction of Attributed Graphs	177
8.3	Pullbacks of Attributed Graphs	178
9	Typed Attributed Graph Transformation Systems	181
9.1	Basic Concepts for Typed AGT Systems	181
9.2	Construction of Typed Attributed Graph Transformations	188
9.3	Local Church–Rosser and Parallelism Theorem for Typed AGT Systems	190
9.4	Concurrency Theorem and Pair Factorization for Typed AGT Systems	194
9.4.1	Pair Factorizations	194
9.4.2	Concurrency Theorem	201
10	Embedding and Local Confluence for Typed AGT Systems	207
10.1	Embedding and Extension Theorems for Typed AGT Systems	207
10.2	Critical Pairs for Typed AGT Systems	215
10.3	Local Confluence Theorem for Typed AGT Systems	218
11	Adhesive HLR Categories for Typed Attributed Graphs ...	221
11.1	Attributed Graph Structure Signatures and Typed Attributed Graphs	222
11.2	Definition of Concrete Adhesive HLR Categories	225
11.3	Verification of the Main Results for Typed AGT Systems	232
12	Constraints, Application Conditions and Termination for TAGT Systems	237
12.1	Constraints and Application Conditions for Typed AGT Systems	237
12.2	Equivalence of Constraints and Application Conditions	242
12.3	Termination Criteria for Layered Typed Attributed Graph Grammars	249

13 Typed Attributed Graph Transformation with Inheritance	259
13.1 Attributed Type Graphs with Inheritance	260
13.2 Attributed Clan Morphisms	265
13.3 Productions and Attributed Graph Transformation with Inheritance	271
13.4 Equivalence of Concepts with and without Inheritance	278

Part IV Case Study on Model Transformation, and Tool Support by AGG

14 Case Study on Model Transformation	287
14.1 Model Transformation by Typed Attributed Graph Transformation	287
14.2 Model Transformation from Statecharts to Petri Nets	288
14.2.1 Source Modeling Language: Simple Version of UML Statecharts	289
14.2.2 Target Modeling Language: Petri Nets	290
14.2.3 Model Transformation	293
14.2.4 Termination Analysis of the Model Transformation	301
14.3 Further Case Studies	303
14.3.1 From the General Resource Model to Petri Nets	303
14.3.2 From Process Interaction Diagrams to Timed Petri Nets	304
14.4 Conclusion	304
15 Implementation of Typed Attributed Graph Transformation by AGG	305
15.1 Language Concepts of AGG	305
15.1.1 Graphs	306
15.1.2 Typing Facilities	306
15.1.3 Node and Edge Attributes	307
15.1.4 Rules and Matches	308
15.1.5 Graph Transformations	310
15.1.6 Graph Grammars	312
15.2 Analysis Techniques Implemented in AGG	312
15.2.1 Graph Constraints	312
15.2.2 Critical Pair Analysis	313
15.2.3 Graph Parsing	317
15.2.4 Termination	318
15.3 Tool Environment of AGG	318
15.3.1 Visual Environment	320
15.3.2 Graph Transformation Engine	321
15.3.3 Tool Integration	322
15.4 Conclusion	322

Appendices

A	A Short Introduction to Category Theory	329
A.1	Categories	329
A.2	Construction of Categories, and Duality	330
A.3	Monomorphisms, Epimorphisms, and Isomorphisms	332
A.4	Pushouts and Pullbacks	334
A.5	Binary Coproducts and Initial Objects	340
A.6	Functors, Functor Categories, and Comma Categories	344
A.7	Isomorphism and Equivalence of Categories	350
B	A Short Introduction to Signatures and Algebras	353
B.1	Algebraic Signatures	353
B.2	Algebras	355
B.3	Terms and Term Evaluation	357
C	Detailed Proofs	359
C.1	Completion of Proof of Fact 4.24	359
C.2	Proof of Lemma 6.25	361
C.3	Completion of Proof of Theorem 11.3	362
C.3.1	Well-Definedness	362
C.3.2	Functors	364
C.3.3	Isomorphism	365
C.4	Proof of Lemma 11.17	366
C.4.1	Well-Definedness	366
C.4.2	Pushout Property	367
C.4.3	Initial Pushout	368
C.5	Proof of Theorem 13.12	370
C.6	Proof of Lemma 13.20	373
	References	377
	Index	385