

LNCS 4178

Andrea Corradini
Hartmut Ehrig
Ugo Montanari
Leila Ribeiro
Grzegorz Rozenberg (Eds.)

Graph Transformations

Third International Conference, ICGT 2006
Natal Rio Grande do Norte, Brazil, September 2006
Proceedings



TP301.6-53

I15

2006

Andrea Corradini Hartmut Ehrig
Ugo Montanari Leila Ribeiro
Grzegorz Rozenberg (Eds.)

Graph Transformations

Third International Conference, ICGT 2006
Natal, Rio Grande do Norte, Brazil
September 17-23, 2006
Proceedings



Springer



E200604050

Volume Editors

Andrea Corradini

Ugo Montanari

University of Pisa

Department of Computer Science

56127 Pisa, Italy

E-mail: {andrea,ugo}@di.unipi.it

Hartmut Ehrig

Technical University of Berlin

Department for Software Technology and Theoretical Informatics

10587 Berlin, Germany

E-mail: ehrig@cs.tu-berlin.de

Leila Ribeiro

Universidade Federal do Rio Grande do Sul

Instituto de Informática

91501-970 Porto Alegre, Brazil

E-mail: leila@inf.ufrgs.br

Grzegorz Rozenberg

Leiden University

Leiden Institute of Advanced Computer Science (LIACS)

2333 CA Leiden, The Netherlands

E-mail: rozenber@liacs.nl

Library of Congress Control Number: 2006931633

CR Subject Classification (1998): E.1, G.2.2, D.2.4, F.1, F.2.2, F.3, F.4.2-3

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743

ISBN-10 3-540-38870-2 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-38870-8 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, re-use of illustrations, recitation, broadcasting, reproduction on microfilms 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 to 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

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11841883 06/3142 543210

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Moshe Y. Vardi

Rice University, Houston, TX, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Preface

ICGT 2006 was the 3rd International Conference on Graph Transformation, following the previous two in Barcelona (2002) and Rome (2004), and a series of six international workshops between 1978 and 1998. ICGT 2006 was held in Natal (Rio Grande do Norte, Brazil) on September 17-23, 2006, co-located with the Brazilian Symposium on Formal Methods (SBMF 2006), under the auspices of the Brazilian Computer Society (SBC), the European Association of Software Science and Technology (EASST), the European Association for Theoretical Computer Science (EATCS) and the IFIP WG 1.3 on Foundations of Systems Specification. The conference obtained partial support from Formal Methods Europe and IFIP TC 1 on Foundations of Computer Science.

The scope of the conference concerned graphical structures of various kinds (like graphs, diagrams and visual sentences) that are useful when describing complex structures and systems in a direct and intuitive way. These structures are often enriched with formalisms that model their evolution via suitable kinds of transformations. The field of the conference was concerned with the theory, applications, and implementation issues of such formalisms. Particular emphasis was put on metamodels which can accommodate a variety of graphical structures within the same abstract theory.

The theory is strongly related to areas such as graph theory and graph algorithms, formal language and parsing theory, the theory of concurrent and distributed systems, formal specification and verification, logics, and semantics. The application areas include all those fields of computer science, information processing, engineering, biology and the natural sciences where static and dynamic modelling using graphical structures and graph transformations, respectively, play important roles. In many of these areas tools based on graph transformation technology have been implemented and used.

The proceedings of ICGT 2006 consist of two parts. The first part contains the contributions of the invited speakers followed by 28 accepted papers that were selected out of 62 carefully reviewed submissions. The topics of the papers range over a wide spectrum, including graph theory and graph algorithms, theoretic and semantic aspects, modelling, contributions to software engineering and global computing, applications to biology, and tool issues. The second part contains a short description of a tutorial on foundations and applications of graph transformations, and short presentations of the satellite events of ICGT 2006.

We would like to thank the members of the program committee and the secondary reviewers for their enormous help in the selection process. Moreover, we would like to express our gratitude to the local organizers who did a great job, in particular to the Organizing Committee chair Anamaria Martins Moreira.

July 2006 Andrea Corradini, Hartmut Ehrig, Ugo Montanari (PC co-chair)
Leila Ribeiro (PC co-chair), Grzegorz Rozenberg

Organization

Program Committee

Paolo Baldan	Venice (Italy)
Paolo Bottoni	Rome (Italy)
Bruno Courcelle	Bordeaux (France)
Andrea Corradini	Pisa (Italy)
Hartmut Ehrig	Berlin (Germany)
Gregor Engels	Paderborn (Germany)
Reiko Heckel	Leicester (UK)
Dirk Janssens	Antwerp (Belgium)
Gabor Karsai	Nashville (Tennessee, USA)
Hans-Jörg Kreowski	Bremen (Germany)
Barbara König	Duisburg-Essen (Germany)
Mercè Llabrés	Mallorca (Spain)
Anamaria Martins Moreira	Natal (Brazil)
Ugo Montanari (co-chair)	Pisa (Italy)
Manfred Nagl	Aachen (Germany)
Fernando Orejas	Barcelona (Spain)
Francesco Parisi-Presicce	Rome (Italy) and Fairfax (Virginia, USA)
Mauro Pezzè	Milan (Italy)
John Pfaltz	Charlottesville (Virginia, USA)
Rinus Plasmeijer	Nijmegen (The Netherlands)
Detlef Plump	York (UK)
Leila Ribeiro (co-chair)	Porto Alegre (RS, Brazil)
Grzegorz Rozenberg	Leiden (The Netherlands)
Andy Schürr	Darmstadt (Germany)
Gabriele Taentzer	Berlin (Germany)
Dániel Varró	Budapest (Hungary)
Daniel Yankelevich	Buenos Aires (Argentina)

Secondary Referees

András Balogh	Søren Christensen	Claudia Ermel
Benjamin Braatz	Giovanni Cignoni	Maribel Fernandez
Victor Braberman	Juan de Lara	Esteban Feuerstein
Chiara Braghin	Anne Dicky	Irene Finocchi
Clara Bertolissi	Dino Distefano	Fabio Gadducci
Laura Bocchi	Mike Dodds	Giorgio Ghelli
Roberto Bruni	Juan Echague	Luís Gomes
Alexey Cherkhago	Karsten Ehrig	Annegret Habel

VIII Organization

Tobias Heindel	Marc Lohmann	Guilherme Rangel
Frank Hermann	Ivan Lanese	Christophe Ringeissen
Dan Hirsch	Greg Manning	Francesc Rosselló
Kathrin Hoffmann	Leonardo Mariani	Domenico Saccà
Nicolas Kicillof	Tony Modica	Hans Schippers
Markus Klein	Olaf Muliawan	Gabriel Valiente
Manuel Koch	Nikos Mylonakis	Niels Van Eetvelde
Radu Kopetz	Julia Padberg	Pieter Van Gorp
Maciej Koutny	Jean-Guy Penaud	Gergely Varró
Georgios Lajios	Ulrike Prange	Jessica Winkelmann
Leen Lambers	Paola Quaglia	

Sponsoring Institutions

Brazilian Computer Society (SBC)
European Association for Theoretical Computer Science (EATCS)
European Association of Software Science and Technology (EASST)
Formal Methods Europe (FME)
International Federation for Information Processing (IFIP) WG 1.3

Lecture Notes in Computer Science

For information about Vols. 1–4063

please contact your bookseller or Springer

- Vol. 4185: R. Mizoguchi, Z. Shi, F. Giunchiglia (Eds.), *The Semantic Web – ASWC 2006*. XX, 778 pages. 2006.
- Vol. 4184: M. Bravetti, M. Nuñez, G. Zavattaro (Eds.), *Web Services and Formal Methods*. X, 289 pages. 2006.
- Vol. 4180: M. Kohlase, OMDoc – An Open Markup Format for Mathematical Documents [version 1.2]. XIX, 428 pages. 2006. (Sublibrary LNAI).
- Vol. 4178: A. Corradini, H. Ehrig, U. Montanari, L. Ribeiro, G. Rozenberg (Eds.), *Graph Transformations*. XII, 473 pages. 2006.
- Vol. 4176: S.K. Katsikas, J. Lopez, M. Backes, S. Gritzalis, B. Preneel (Eds.), *Information Security*. XIV, 548 pages. 2006.
- Vol. 4168: Y. Azar, T. Erlebach (Eds.), *Algorithms – ESA 2006*. XVIII, 843 pages. 2006.
- Vol. 4165: W. Jonker, M. Petkovic (Eds.), *Secure, Data Management*. X, 183 pages. 2006.
- Vol. 4163: H. Bersini, J. Carneiro (Eds.), *Artificial Immune Systems*. XII, 460 pages. 2006.
- Vol. 4162: R. Kráľovič, P. Urzyczyn (Eds.), *Mathematical Foundations of Computer Science 2006*. XV, 814 pages. 2006.
- Vol. 4159: J. Ma, H. Jin, L.T. Yang, J.J.-P. Tsai (Eds.), *Ubiquitous Intelligence and Computing*. XXII, 1190 pages. 2006.
- Vol. 4158: L.T. Yang, H. Jin, J. Ma, T. Ungerer (Eds.), *Autonomic and Trusted Computing*. XIV, 613 pages. 2006.
- Vol. 4156: S. Amer-Yahia, Z. Bellahsene, E. Hunt, R. Unland, J.X. Yu (Eds.), *Database and XML Technologies*. IX, 123 pages. 2006.
- Vol. 4155: O. Stock, M. Schaerf (Eds.), *Reasoning, Action and Interaction in AI Theories and Systems*. XVIII, 343 pages. 2006. (Sublibrary LNAI).
- Vol. 4153: N. Zheng, X. Jiang, X. Lan (Eds.), *Advances in Machine Vision, Image Processing, and Pattern Analysis*. XIII, 506 pages. 2006.
- Vol. 4152: Y. Manolopoulos, J. Pokorný, T. Sellis (Eds.), *Advances in Databases and Information Systems*. XV, 448 pages. 2006.
- Vol. 4151: A. Iglesias, N. Takayama (Eds.), *Mathematical Software - ICMS 2006*. XVII, 452 pages. 2006.
- Vol. 4150: M. Dorigo, L.M. Gambardella, M. Birattari, A. Martinoli, R. Poli, T. Stützle (Eds.), *Ant Colony Optimization and Swarm Intelligence*. XVI, 526 pages. 2006.
- Vol. 4149: M. Klusch, M. Rovatsos, T.R. Payne (Eds.), *Cooperative Information Agents X*. XII, 477 pages. 2006. (Sublibrary LNAI).
- Vol. 4146: J.C. Rajapakse, L. Wong, R. Acharya (Eds.), *Pattern Recognition in Bioinformatics*. XIV, 186 pages. 2006. (Sublibrary LNBI).
- Vol. 4144: T. Ball, R.B. Jones (Eds.), *Computer Aided Verification*. XV, 564 pages. 2006.
- Vol. 4139: T. Salakoski, F. Ginter, S. Pyysalo, T. Pahikkala, *Advances in Natural Language Processing*. XVI, 771 pages. 2006. (Sublibrary LNAI).
- Vol. 4138: X. Cheng, W. Li, T. Znati (Eds.), *Wireless Algorithms, Systems, and Applications*. XVI, 709 pages. 2006.
- Vol. 4137: C. Baier, H. Hermanns (Eds.), *CONCUR 2006 – Concurrency Theory*. XIII, 525 pages. 2006.
- Vol. 4136: R.A. Schmidt (Ed.), *Relations and Kleene Algebra in Computer Science*. XI, 433 pages. 2006.
- Vol. 4135: C.S. Calude, M.J. Dinneen, G. Păun, G. Rozenberg, S. Stepney (Eds.), *Unconventional Computation*. X, 267 pages. 2006.
- Vol. 4134: K. Yi (Ed.), *Static Analysis*. XIII, 443 pages. 2006.
- Vol. 4133: J. Gratch, M. Young, R. Aylett, D. Ballin, P. Olivier (Eds.), *Intelligent Virtual Agents*. XIV, 472 pages. 2006. (Sublibrary LNAI).
- Vol. 4130: U. Furbach, N. Shankar (Eds.), *Automated Reasoning*. XV, 680 pages. 2006. (Sublibrary LNAI).
- Vol. 4129: D. McGookin, S. Brewster (Eds.), *Haptic and Audio Interaction Design*. XII, 167 pages. 2006.
- Vol. 4128: W.E. Nagel, W.V. Walter, W. Lehner (Eds.), *Euro-Par 2006 Parallel Processing*. XXXIII, 1221 pages. 2006.
- Vol. 4127: E. Damiani, P. Liu (Eds.), *Data and Applications Security XX*. X, 319 pages. 2006.
- Vol. 4126: P. Barahona, F. Bry, E. Franconi, N. Henze, U. Sattler, *Reasoning Web*. X, 269 pages. 2006.
- Vol. 4124: H. de Meer, J.P. G. Sterbenz (Eds.), *Self-Organizing Systems*. XIV, 261 pages. 2006.
- Vol. 4121: A. Biere, C.P. Gomes (Eds.), *Theory and Applications of Satisfiability Testing - SAT 2006*. XII, 438 pages. 2006.
- Vol. 4119: C. Dony, J.L. Knudsen, A. Romanovsky, A. Tripathi (Eds.), *Advanced Topics in Exception Handling Components*. X, 302 pages. 2006.
- Vol. 4117: C. Dwork (Ed.), *Advances in Cryptology - CRYPTO 2006*. XIII, 621 pages. 2006.
- Vol. 4116: R. De Prisco, M. Yung (Eds.), *Security and Cryptography for Networks*. XI, 366 pages. 2006.

- Vol. 4115: D.-S. Huang, K. Li, G.W. Irwin (Eds.), Computational Intelligence and Bioinformatics, Part III. XXI, 803 pages. 2006. (Sublibrary LNBI).
- Vol. 4114: D.-S. Huang, K. Li, G.W. Irwin (Eds.), Computational Intelligence, Part II. XXVII, 1337 pages. 2006. (Sublibrary LNBI).
- Vol. 4113: D.-S. Huang, K. Li, G.W. Irwin (Eds.), Intelligent Computing, Part I. XXVII, 1331 pages. 2006.
- Vol. 4112: D.Z. Chen, D. T. Lee (Eds.), Computing and Combinatorics. XIV, 528 pages. 2006.
- Vol. 4111: F.S. de Boer, M.M. Bonsangue, S. Graf, W.-P. de Roever (Eds.), Formal Methods for Components and Objects. VIII, 447 pages. 2006.
- Vol. 4110: J. Díaz, K. Jansen, J.D.P. Rolim, U. Zwick (Eds.), Approximation, Randomization, and Combinatorial Optimization. XII, 522 pages. 2006.
- Vol. 4109: D.-Y. Yeung, J.T. Kwok, A. Fred, F. Roli, D. de Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition. XXI, 939 pages. 2006.
- Vol. 4108: J.M. Borwein, W.M. Farmer (Eds.), Mathematical Knowledge Management. VIII, 295 pages. 2006. (Sublibrary LNBI).
- Vol. 4106: T.R. Roth-Berghofer, M.H. Göker, H. A. Güvenir (Eds.), Advances in Case-Based Reasoning. XIV, 566 pages. 2006. (Sublibrary LNBI).
- Vol. 4104: T. Kunz, S.S. Ravi (Eds.), Ad-Hoc, Mobile, and Wireless Networks. XII, 474 pages. 2006.
- Vol. 4099: Q. Yang, G. Webb (Eds.), PRICAI 2006: Trends in Artificial Intelligence. XXVIII, 1263 pages. 2006. (Sublibrary LNBI).
- Vol. 4098: F. Pfenning (Ed.), Term Rewriting and Applications. XIII, 415 pages. 2006.
- Vol. 4097: X. Zhou, O. Sokolsky, L. Yan, E.-S. Jung, Z. Shao, Y. Mu, D.C. Lee, D. Kim, Y.-S. Jeong, C.-Z. Xu (Eds.), Emerging Directions in Embedded and Ubiquitous Computing. XXVII, 1034 pages. 2006.
- Vol. 4096: E. Sha, S.-K. Han, C.-Z. Xu, M.H. Kim, L.T. Yang, B. Xiao (Eds.), Embedded and Ubiquitous Computing. XXIV, 1170 pages. 2006.
- Vol. 4095: S. Nolfi, G. Baldassare, R. Calabretta, D. Marocco, D. Parisi, J.C. T. Hallam, O. Miglino, J.-A. Meyer (Eds.), From Animals to Animats 9. XV, 869 pages. 2006. (Sublibrary LNBI).
- Vol. 4094: O. H. Ibarra, H.-C. Yen (Eds.), Implementation and Application of Automata. XIII, 291 pages. 2006.
- Vol. 4093: X. Li, O.R. Zaïane, Z. Li (Eds.), Advanced Data Mining and Applications. XXI, 1110 pages. 2006. (Sublibrary LNBI).
- Vol. 4092: J. Lang, F. Lin, J. Wang (Eds.), Knowledge Science, Engineering and Management. XV, 664 pages. 2006. (Sublibrary LNBI).
- Vol. 4091: G.-Z. Yang, T. Jiang, D. Shen, L. Gu, J. Yang (Eds.), Medical Imaging and Augmented Reality. XIII, 399 pages. 2006.
- Vol. 4090: S. Spaccapietra, K. Aberer, P. Cudré-Mauroux (Eds.), Journal on Data Semantics VI. XI, 211 pages. 2006.
- Vol. 4089: W. Löwe, M. Südholt (Eds.), Software Composition. X, 339 pages. 2006.
- Vol. 4088: Z.-Z. Shi, R. Sadananda (Eds.), Agent Computing and Multi-Agent Systems. XVII, 827 pages. 2006. (Sublibrary LNBI).
- Vol. 4087: F. Schwenker, S. Marinai (Eds.), Artificial Neural Networks in Pattern Recognition. IX, 299 pages. 2006. (Sublibrary LNBI).
- Vol. 4085: J. Misra, T. Nipkow, E. Sekerinski (Eds.), FM 2006: Formal Methods. XV, 620 pages. 2006.
- Vol. 4084: M.A. Wimmer, H.J. Scholl, Å. Grönlund, K.V. Andersen (Eds.), Electronic Government. XV, 353 pages. 2006.
- Vol. 4083: S. Fischer-Hübner, S. Furnell, C. Lambri-noudakis (Eds.), Trust and Privacy in Digital Business. XIII, 243 pages. 2006.
- Vol. 4082: K. Bauknecht, B. Pröll, H. Werthner (Eds.), E-Commerce and Web Technologies. XIII, 243 pages. 2006.
- Vol. 4081: A. M. Tjoa, J. Trujillo (Eds.), Data Warehousing and Knowledge Discovery. XVII, 578 pages. 2006.
- Vol. 4080: S. Bressan, J. Küng, R. Wagner (Eds.), Database and Expert Systems Applications. XXI, 959 pages. 2006.
- Vol. 4079: S. Etalle, M. Truszczyński (Eds.), Logic Programming. XIV, 474 pages. 2006.
- Vol. 4077: M.-S. Kim, K. Shimada (Eds.), Geometric Modeling and Processing - GMP 2006. XVI, 696 pages. 2006.
- Vol. 4076: F. Hess, S. Pauli, M. Pohst (Eds.), Algorithmic Number Theory. X, 599 pages. 2006.
- Vol. 4075: U. Leser, F. Naumann, B. Eckman (Eds.), Data Integration in the Life Sciences. XI, 298 pages. 2006. (Sublibrary LNBI).
- Vol. 4074: M. Burmester, A. Yasinac (Eds.), Secure Mobile Ad-hoc Networks and Sensors. X, 193 pages. 2006.
- Vol. 4073: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 263 pages. 2006.
- Vol. 4072: M. Harders, G. Székely (Eds.), Biomedical Simulation. XI, 216 pages. 2006.
- Vol. 4071: H. Sundaram, M. Naphade, J.R. Smith, Y. Rui (Eds.), Image and Video Retrieval. XII, 547 pages. 2006.
- Vol. 4070: C. Priami, X. Hu, Y. Pan, T.Y. Lin (Eds.), Transactions on Computational Systems Biology V. IX, 129 pages. 2006. (Sublibrary LNBI).
- Vol. 4069: F.J. Perales, R.B. Fisher (Eds.), Articulated Motion and Deformable Objects. XV, 526 pages. 2006.
- Vol. 4068: H. Schärfe, P. Hitzler, P. Øhrstrøm (Eds.), Conceptual Structures: Inspiration and Application. XI, 455 pages. 2006. (Sublibrary LNBI).
- Vol. 4067: D. Thomas (Ed.), ECOOP 2006 – Object-Oriented Programming. XIV, 527 pages. 2006.
- Vol. 4066: A. Rensink, J. Warner (Eds.), Model Driven Architecture – Foundations and Applications. XII, 392 pages. 2006.
- Vol. 4065: P. Perner (Ed.), Advances in Data Mining. XI, 592 pages. 2006. (Sublibrary LNBI).
- Vol. 4064: R. Büschkes, P. Laskov (Eds.), Detection of Intrusions and Malware & Vulnerability Assessment. X, 195 pages. 2006.

Table of Contents

Invited Papers

Nested Quantification in Graph Transformation Rules	1
<i>Arend Rensink</i>	
Idioms of Logical Modelling	14
<i>Daniel Jackson</i>	
New Algorithms and Applications of Cyclic Reference Counting.....	15
<i>Rafael Dueire Lins</i>	

New Graph Transformation Models

Sesqui-Pushout Rewriting	30
<i>Andrea Corradini, Tobias Heindel, Frank Hermann, Barbara König</i>	
Automata on Directed Graphs: Edge Versus Vertex Marking	46
<i>Dietmar Berwanger, David Janin</i>	
Conflict Detection for Graph Transformation with Negative Application Conditions	61
<i>Leen Lambers, Hartmut Ehrig, Fernando Orejas</i>	
Adaptive Star Grammars	77
<i>Frank Drewes, Berthold Hoffmann, Dirk Janssens, Mark Minas, Niels Van Eetvelde</i>	

Structure Manipulation

Narrowing Data-Structures with Pointers	92
<i>Rachid Echahed, Nicolas Peltier</i>	
Molecular Analysis of Metabolic Pathway with Graph Transformation	107
<i>Karsten Ehrig, Reiko Heckel, Georgios Lajios</i>	
Matrix Approach to Graph Transformation: Matching and Sequences	122
<i>Pedro Pablo Pérez Velasco, Juan de Lara</i>	

String Generating Hypergraph Grammars with Word Order Restrictions	138
<i>Martin Riedl, Sebastian Seifert, Ingrid Fischer</i>	

Borrowed Contexts and Adhesive Categories

Composition and Decomposition of DPO Transformations with Borrowed Context	153
<i>Paolo Baldan, Hartmut Ehrig, Barbara König</i>	
Process Bisimulation Via a Graphical Encoding	168
<i>Filippo Bonchi, Fabio Gadducci, Barbara König</i>	
Toposes Are Adhesive	184
<i>Stephen Lack, Paweł Sobociński</i>	

Extensions for Distributed and Global Computing

Graph Transactions as Processes	199
<i>Paolo Baldan, Andrea Corradini, Luciana Foss, Fabio Gadducci</i>	
Categorical Foundations of Distributed Graph Transformation	215
<i>Hartmut Ehrig, Fernando Orejas, Ulrike Prange</i>	
Dynamic Graph Transformation Systems	230
<i>Roberto Bruni, Hernán Melgratti</i>	
Autonomous Units and Their Semantics — The Sequential Case	245
<i>Karsten Hölscher, Hans-Jörg Kreowski, Sabine Kuske</i>	

Software Engineering Methods and Tools

Termination Analysis of Model Transformations by Petri Nets	260
<i>Dániel Varró, Szilvia Varró-Gyapay, Hartmut Ehrig, Ulrike Prange, Gabriele Taentzer</i>	
Non-functional Analysis of Distributed Systems in Unreliable Environments Using Stochastic Object Based Graph Grammars	275
<i>Odorico Machado Mendizabal, Fernando Luis Dotti</i>	
Temporal Graph Queries to Support Software Evolution	291
<i>Tobias Röttschke, Andy Schürr</i>	
On the Use of Alloy to Analyze Graph Transformation Systems	306
<i>Luciano Baresi, Paola Spoletini</i>	

Model-Driven Development

Non-materialized Model View Specification with Triple Graph Grammars	321
<i>Johannes Jakob, Alexander Königs, Andy Schürr</i>	
Model-Driven Monitoring: An Application of Graph Transformation for Design by Contract	336
<i>Gregor Engels, Marc Lohmann, Stefan Sauer, Reiko Heckel</i>	
Model View Management with Triple Graph Transformation Systems	351
<i>Esther Guerra, Juan de Lara</i>	

Efficient Implementation

Graph Transformation in Constant Time	367
<i>Mike Dodds, Detlef Plump</i>	
GrGen: A Fast SPO-Based Graph Rewriting Tool	383
<i>Rubino Geiß, Gernot Veit Batz, Daniel Grund, Sebastian Hack, Adam Szalkowski</i>	
Realizing Graph Transformations by Pre- and Postconditions and Command Sequences	398
<i>Fabian Büttner, Martin Gogolla</i>	
Heuristic Search for the Analysis of Graph Transition Systems	414
<i>Stefan Edelkamp, Shahid Jabbar, Alberto Lluch Lafuente</i>	

Logics

Satisfiability of High-Level Conditions	430
<i>Annegret Habel, Karl-Heinz Pennemann</i>	
Weakest Preconditions for High-Level Programs	445
<i>Annegret Habel, Karl-Heinz Pennemann, Arend Rensink</i>	

Tutorial and Workshops

Introductory Tutorial on Foundations and Applications of Graph Transformation	461
<i>Reiko Heckel</i>	

Workshop on Graph Computation Models 463
 Yves Métivier, Mohamed Mosbah

Workshop on Graph-Based Tools 465
 Albert Zündorf, Dániel Varró

Workshop on Petri Nets and Graph Transformations 467
 Paolo Baldan, Hartmut Ehrig, Julia Padberg, Grzegorz Rozenberg

3rd International Workshop on Software Evolution Through
Transformations: Embracing Change 470
 Jean-Marie Favré, Reiko Heckel, Tom Mens

Author Index 473

Nested Quantification in Graph Transformation Rules

Arend Rensink

Department of Computer Science, University of Twente
P.O. Box 217, 7500 AE, The Netherlands
rensink@cs.utwente.nl*

Abstract. In this paper we describe a way to integrate Taentzer's rule amalgamation with the recently proposed notions of nested graph conditions. The resulting so-called *quantified graph transformation rules* include (universally and existentially) quantified sub-structures in a flexible way. This can be used for instance to specify a larger-step operational semantics, thus improving the scalability of graph transformation as a technique for software verification.

1 Introduction

The idea presented in this paper is motivated by the goal to use graph transformation as a technique for specifying, and eventually verifying, the dynamic behaviour of software systems. In this setup, each transformation rule corresponds to a single computation step of the system, in which, for instance, a method is called or a variable is assigned. We have observed in previous work, e.g., [13], that such a computation step frequently involves acting upon a structure whose size is not *a priori* known, but instead involves sub-structures of which there may be arbitrarily many copies.

A typical example of this is the encoding of parameter transfer from a method caller to the called method. Obviously this is the same mechanism for all methods, and so we would like to have a single rule that captures it. Unfortunately, the number of parameters is not the same for all methods: in fact, the parameters indeed form a sub-structure with a varying and a priori unknown number of copies.¹ For this reason, it is not possible, using the standard graph transformation formalism, to capture the parameter transfer mechanism in a single rule. Fig. 1 shows an example rule for two parameters.

Although there are workarounds, typically involving the use of auxiliary edges which successively mark all copies of the substructure involved, these have undesirable consequences (besides being inelegant). In particular, such a solution results in a number of small steps that is linear in the number of substructures involved. In particular in a setting where the system under analysis has parallelism, these steps get interleaved with independent actions in other parts of the system, contributing to the state space blow-up (which is the most urgent problem in verification methods in the first place). This,

* The work reported in this paper was carried out in the context of the Dutch NWO project GROOVE (project number 612.000.314).

¹ To be more precise, the number of parameters is fixed and known for each individual method, but not from the more global perspective of our semantics, in which calls to all methods are to be treated as instances of the same mechanism.

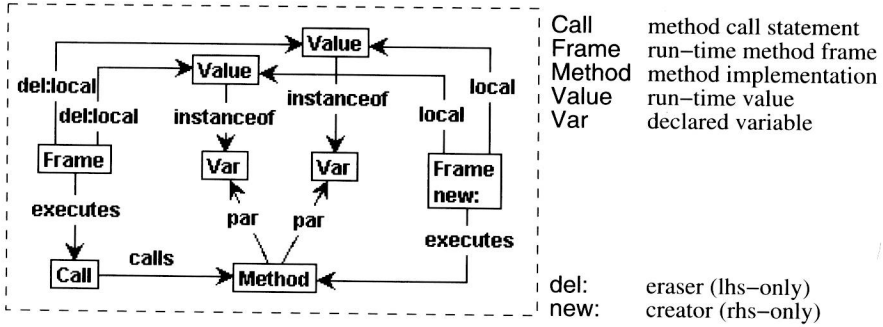


Fig. 1. Method call rule for two parameters

in turn, may be alleviated by further modifying the formalism, for instance by imposing priorities or other forms of control on the set of rules, but at the price of increased complexity of the formalism and hence of the verification task itself.

To make the example even more challenging: if we are interested in a data-flow analysis, we want the method call rule to be enabled if and only if all the arguments to the method are available (and not when control has explicitly reached the point in the program where the method is called). This involves a further condition on the rule, involving the existence of all relevant substructures — in this case, values for all method parameters. This is *not* a standard (positive or negative) application condition, since, once more, the number of substructures that are required to exist is not a priori known.

The problem described above has been studied before. On the practical side, many tools for graph transformation (for instance GREAT [6], FuJaBa [4]) have some notion of graph patterns that may be matched with cardinality greater than 1, i.e., that match to an a priori unknown number of sub-graphs in the host graph. Furthermore, Taentzer [18] has developed an elegant theoretical basis for this type of extension, called *rule amalgamation*. This is based on the concept of an *interaction scheme* which essentially imposes a sub-rule embedding on a set of rules, and a *covering condition* which imposes further conditions on the matches to be considered. Taking the above example, there would be a single so-called *elementary rule* that takes care of the hand-over of a single parameter, and a sub-rule that selects the caller Frame node and creates the called Frame node; the covering condition would be *local-all*, which gathers as many copies of the elementary rule as there are suitable combinations of Var- and Value-nodes attached to the selected Frame-node.

Elegant and natural though this solution is, it does not yet meet all demands. For instance, the data flow analysis rule proposed above cannot be captured by a simple sub-rule embedding, even in the presence of (standard) negative application conditions: the result would be that the rule as a whole is always enabled, with copies of the elementary rule precisely for those parameters for which an argument value is already available. More generally, the problem is to enforce, in a covering condition, that elementary rules are actually enabled for all sub-structures (in the host graph) of a particular kind, i.e., satisfying a certain application condition. In other words, there are further gains to be made in the appropriate combination of covering conditions and application conditions.

Recently there have been proposals, by ourselves in [16] and independently by Habel and Pennemann in [9], to extend the power of application conditions, by generalising the two-level structures originally introduced in [8] to trees of arbitrary nesting depth. As shown in [16], every further level of nesting effectively corresponds to an additional level of quantification in terms of logic.

The core contribution of this paper is to recognise that the two principles of sub-rule embedding on the one hand and condition nesting on the other can be fruitfully combined, giving rise to a notion of *quantified graph transformation rule* that is both natural and powerful, and complements the framework for rule amalgamation so as to solve the problem outlined above.

Summarising, in this paper we combine two pre-existing ideas:

Rule amalgamation, developed by Taentzer [18] and later applied for, e.g., refactoring [1], parallel graph transformation [3] and multi-formalism simulation [2].

Nested graph predicates, recently proposed by Rensink [16] and independently by Habel and Pennemann [9].

In terms of these techniques, a brief explanation of our proposal is that we merge the nesting structure of the nested graph predicates with the sub-rule embeddings of interaction schemes, as used in rule amalgamation, so that the left hand sides of the rules are part of a nested graph predicate that simultaneously acts as an application condition. Another way to put it (slightly more loosely) is to say that we present a way to use nested graph predicates as a language for covering conditions, in the sense of Taentzer [18].

The paper is structured as follows: Sect. 2 provides the necessary technical concepts, Sect. 3 illustrates their use on the basis of a number of examples, and Sect. 4 concludes the paper.

2 Definitions

We first recall the notion of graph predicates from [16] and in passing establish the connection to [9]. We assume some category of graphs \mathbf{Graph} with an initial element \emptyset , objects G, H and morphisms f, g etc. For concrete examples, we will take the common edge-labelled graphs $\langle N, E, L, s, t, \ell \rangle$, with N as set of nodes, E a set of edges, L a set of labels, and $s, t: E \rightarrow N$ and $\ell: E \rightarrow L$ the source and target mapping and labelling function, respectively; morphisms will be the homomorphisms over this structure.

We characterise graph predicates as rooted diagrams in the category \mathbf{Graph} . Given such a diagram d and an object G in d , $root_d$ will denote the object at the root of d , and $out_d(G)$ will denote the morphisms in d that originate in G . (Note that a rooted diagram has a well-defined root even if it has no arrows.) Furthermore, $sub_d(G)$ will denote the reachable sub-diagram rooted in G , and for arbitrary $f: G \rightarrow root_d$, $c = d \circ f$ is the diagram with $root_c = G$, $init_c = \{g \circ f \mid g \in init_d\}$ and $sub_c(H) = sub_d(H)$ for all graphs H in d except for $root_d$. (In other words, $d \circ f$ is obtained from d by using f 's source as the new root, concatenating f with the initial arrows of d and leaving the remainder of d unchanged.)

Definition 1 (graph predicate). Let $G \in \text{Graph}$ be arbitrary. A graph predicate over G is a tree-shaped diagram p in Graph rooted in G . p is called ground if $G = \emptyset$.

Predicate satisfaction is a binary relation \models between predicates and graph morphisms: $p \models f$ expresses that f satisfies p . \models is defined as the smallest relation such that $p \models f$ whenever the following conditions hold:

- $\text{root}_p = H$;
- There are $g: G \rightarrow K \in \text{init}_p$ and $h: K \rightarrow H$ such that $f = h \circ g$ and $\text{sub}_p(K) \not\models h$.

Moreover, if we say that a graph G satisfies p , denoted $p \models G$, if $p \models f: \emptyset \rightarrow G$. (Note that this implies that p is ground.)

It might take some getting used to that the subjects are morphisms f rather than graphs. The intuition is that the source of f , which corresponds to the root of the predicate diagram, only identifies the common context or pattern; typically a subgraph which is already known to be present. The predicate itself actually states something about the target graph of the morphism. This intuition is confirmed by the special case where the source of f is the empty graph: in that case f really contains only the information in its target.

Satisfaction as defined in Def. 1 is slightly tricky in that it seems to rely on a smallest fixpoint construction for a function that is not monotonic, but rather anti-monotonic, due to the negation in the second bullet above. However, this is only superficially true, since in the sub-clause the satisfaction predicate is applied to a strictly smaller diagram; hence we can conduct proofs on the depth of the predicate diagram. (Actually, the above definition and most of the developments of this paper would still work for dags, and even for diagrams with cycles, as long as for any arrow in the diagram, the length of every path from the root to that arrow has the same parity, i.e., either all paths have odd length or all paths have even length. This implies that, in particular, all cycles must have even length. However, we restrict to tree-shaped diagrams in this paper.)

Alternatively, and perhaps more understandably, predicate satisfaction can be formulated in terms of two distinct satisfaction relations, \models_\exists and \models_\forall , as follows:

Definition 2. Existential and universal satisfaction are the smallest pair of binary relations \models_\exists and \models_\forall between graph predicates p and graph morphisms $f: G \rightarrow H$ such that

- $p \models_\exists f$ whenever the following conditions hold:
 - $\text{root}_p = H$;
 - There are $g: G \rightarrow K \in \text{init}_p$ and $h: K \rightarrow H$ such that $f = h \circ g$ and $\text{sub}_p(K) \models_\forall h$.
- $p \models_\forall f$ whenever the following conditions hold:
 - $\text{root}_p = H$;
 - For all $g: G \rightarrow K \in \text{init}_p$ and all $h: K \rightarrow H$ such that $f = h \circ g$, $\text{sub}_p(K) \models_\exists h$.

Note that the concept of “smallest pair of binary relations” is indeed well-defined, under a pairwise ordering on sets. Also note that the problem due to the negation in the inductive definition of Def. 1 is no longer present in this formulation; instead, we can mark every arrow of a predicate diagram as \exists or \forall , depending on their distance from