Aart Middeldorp (Ed.)

LNCS 2051

# Rewriting Techniques and Applications

12th International Conference, RTA 2001
Utrecht, The Netherlands, May 2001
Proceedings

Springer

Aart Middeldorp (Ed.)

# Rewriting Techniques and Applications

12th International Conference, RTA 2001
Utrecht, The Netherlands, May 22-24, 2001
Proceedings



Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editor

Aart Middeldorp
University of Tsukuba
Institute of Information Sciences and Electronics
Tsukuba 305-8573, Japan
E-mail: ami@is.tsukuba.ac.jp

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Rewriting techniques and applications : 12th international conference ; proceedings / RTA 2001, Utrecht, The Netherlands, May 22 - 24, 2001. Aart Middeldorp (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2051)
ISBN 3-540-42117-3

CR Subject Classification (1998): F.4, F.3.2, D.3, I.2.2-3, I.1

ISSN 0302-9743
ISBN 3-540-42117-3 Springer-Verlag Berlin Heidelberg New York



Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de


Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP Berlin, Stefan Sossna
Printed on acid-free paper SPIN 10781501 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 2051
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer
*Berlin*
*Heidelberg*
*New York*
*Barcelona*
*Hong Kong*
*London*
*Milan*
*Paris*
*Singapore*
*Tokyo*

# Preface

This volume contains the proceedings of the *12th International Conference on Rewriting Techniques and Applications* (RTA 2001), which was held May 22-24, 2001 at Utrecht University in The Netherlands. RTA is the major forum for the presentation of research on all aspects of rewriting. Previous RTA conferences were held in Dijon (1985), Bordeaux (1987), Chapel Hill (1989), Como (1991), Montreal (1993), Kaiserslautern (1995), Rutgers (1996), Sitges (1997), Tsukuba (1998), Trento (1999), and Norwich (2000).

There were 55 submissions from Argentina ($\frac{2}{3}$), Australia (1), France ($12\frac{2}{3}$), Germany ($11\frac{2}{3}$), Israel ($1\frac{1}{3}$), Italy (2), Japan ($8\frac{1}{2}$), The Netherlands (6), Slovakia ($\frac{1}{3}$), Spain (4), UK ($2\frac{5}{6}$), USA (3), and Venezuela (1), of which the program committee selected 23 regular papers and 2 system descriptions for presentation. In addition, there were invited talks by Arvind (Rewriting the Rules for Chip Design), Henk Barendregt (Computing and Proving), and Michael Rusinowitch (Rewriting for Deduction and Verification).

The program committee awarded the *best paper* prize to Jens R. Woinowski for his paper *A Normal Form for Church-Rosser Language Systems.* In this paper the surprising and important result is shown that all Church-Rosser languages can be defined by string rewrite rules of the form $uvw \to uxw$ with $v$ being nonempty and $x$ having a maximum length of one.

Many people helped to make RTA 2001 a success. I am grateful to the members of the program committee and the external referees for reviewing the submissions and maintaining the high standards of the RTA conferences. It is a particular pleasure to thank Vincent van Oostrom and the other members of the local organizing committee for organizing an excellent conference in a rather short period. Finally, I thank the organizers of the four events that collocated with RTA 2001 for making the conference even more attractive:

- 4th International Workshop on Explicit Substitutions: Theory and Applications to Programs and Proofs (Pierre Lescanne),
- 5th International Workshop on Termination (Nachum Dershowitz),
- International Workshop on Reduction Strategies in Rewriting and Programming (Bernhard Gramlich and Salvador Lucas),
- IFIP Working Group 1.6 on Term Rewriting (Claude Kirchner).

March 2000 Aart Middeldorp

# Conference Organization

## Program Chair

| | |
|---|---|
| Aart Middeldorp | *University of Tsukuba* |

## Conference Chair

| | |
|---|---|
| Vincent van Oostrom | *Utrecht University* |

## Program Committee

| | |
|---|---|
| Zena Ariola | *Eugene* |
| David Basin | *Freiburg* |
| Mariangiola Dezani-Ciancaglini | *Torino* |
| Philippe de Groote | *Nancy* |
| Christine Paulin-Mohring | *Orsay* |
| Ian Mackie | *Palaiseau* |
| José Meseguer | *Menlo Park* |
| Aart Middeldorp | *Tsukuba* |
| Robert Nieuwenhuis | *Barcelona* |
| Enno Ohlebusch | *Bielefeld* |
| Friedrich Otto | *Kassel* |
| Sándor Vágvölgyi | *Szeged* |
| Joe Wells | *Edinburgh* |

## Local Organizing Committee

| | |
|---|---|
| Jan Bergstra | *Utrecht* |
| Hans Zantema | *Eindhoven* |
| Vincent van Oostrom | *Utrecht* |

## RTA Steering Committee

| | | |
|---|---|---|
| Leo Bachmair | *Stony Brook* | |
| Nachum Dershowitz | *Tel Aviv* | (chair) |
| Hélène Kirchner | *Nancy* | |
| José Meseguer | *Menlo Park* | (publicity chair) |
| Tobias Nipkow | *Munich* | |
| Michael Rusinowitch | *Nancy* | |

## Sponsors

Centrum voor Wiskunde en Informatica (CWI)
Instituut voor Programmatuurkunde en Algoritmiek (IPA)
International Federation for Information Processing (IFIP)
University of Amsterdam, Informatics Institute
University of Tsukuba, Institute of Information Sciences and Electronics
Utrecht University, Department of Philosophy
Utrecht University, Lustrum Committee
ZENO Institute for Philosophy, The Leiden-Utrecht Research Institute

## List of Referees

Yohji Akama
Thomas Arts
Jürgen Avenhaus
Steffen van Bakel
Franco Barbanera
László Bernátsky
Gavin Bierman
Stefan Blom
Roel Bloo
Gerhard Buntrock
Wei Ngan Chin
Adam Cichon
Manuel Clavel
Evelyne Contejean
Mario Coppo
Andrea Corradini
Roberto Di Cosmo
Dan Dougherty
Frank Drewes
Catherine Dubois
Irène Durand
Steven Eker
Maribel Fernández
Wan Fokkink
Zoltán Fülöp
Philippa Gardner
Simon Gay
Neil Ghani
Robert Giegerich
Jürgen Giesl
Isabelle Gnaedig
Guillem Godoy
Eric Goubault
Jean Goubault-Larrecq
Bernhard Gramlich
Stefano Guerrini
Pál Gyenizse
Chris Hankin
Michael Hanus
Thérèse Hardin
Dieter Hofbauer
Markus Holzer
Maria Huber
Benedetto Intrigila
Jean-Pierre Jouannaud
Fairouz Kamareddine
Yoshinobu Kawabe
Richard Kennaway
Delia Kesner
Felix Klaedtke
Yves Lafont
François Lamarche
Ingo Lepper
Pierre Lescanne
Jordi Levy
Ugo de'Liguoro
Luigi Liquori
Markus Lohrey
Salvador Lucas
Klaus Madlener
Luis Mandel
Claude Marché
Maurice Margenstern
Gundula Niemann
Yoshikatsu Ohta
Peter Ølveczky
Vincent van Oostrom
Adriano Peron
Jorge Sousa Pinto
Adolfo Piperno
François Pottier
Femke van Raamsdonk
Christophe Ringeissen
Albert Rubio
Michael Rusinowitch
Kai Salomaa
Andrea Sattler-Klein
Manfred Schmidt-Schauß
Marco Schorlemmer
Aleksy Schubert
Carsten Schürmann
Helmut Seidl
Paula Severi
Mark-Oliver Stehr
Magnus Steinby
Georg Struth
Taro Suzuki
Ralf Treinen
Marisa Venturini Zilli
Laurent Vigneron
Fer-Jan de Vries
Uwe Waldmann
Andreas Weiermann
Benjamin Werner
Hans Zantema

# Author Index

# Lecture Notes in Computer Science

For information about Vols. 1–1964
please contact your bookseller or Springer-Verlag

Vol. 1958: S. Fischer-Hübner, IT-Security and Privacy. XIII, 351 pages. 2001.

Vol. 1965: Ç. K. Koç, C. Paar (Eds.), Cryptographic Hardware and Embedded Systems – CHES 2000. Proceedings, 2000. XI, 355 pages. 2000.

Vol. 1966: S. Bhalla (Ed.), Databases in Networked Information Systems. Proceedings, 2000. VIII, 247 pages. 2000.

Vol. 1967: S. Arikawa, S. Morishita (Eds.), Discovery Science. Proceedings, 2000. XII, 332 pages. 2000. (Subseries LNAI).

Vol. 1968: H. Arimura, S. Jain, A. Sharma (Eds.), Algorithmic Learning Theory. Proceedings, 2000. XI, 335 pages. 2000. (Subseries LNAI).

Vol. 1969: D.T. Lee, S.-H. Teng (Eds.), Algorithms and Computation. Proceedings, 2000. XIV, 578 pages. 2000.

Vol. 1970: M. Valero, V.K. Prasanna, S. Vajapeyam (Eds.), High Performance Computing – HiPC 2000. Proceedings, 2000. XVIII, 568 pages. 2000.

Vol. 1971: R. Buyya, M. Baker (Eds.), Grid Computing – GRID 2000. Proceedings, 2000. XIV, 229 pages. 2000.

Vol. 1972: A. Omicini, R. Tolksdorf, F. Zambonelli (Eds.), Engineering Societies in the Agents World. Proceedings, 2000. IX, 143 pages. 2000. (Subseries LNAI).

Vol. 1973: J. Van den Bussche, V. Vianu (Eds.), Database Theory – ICDT 2001. Proceedings, 2001. X, 451 pages. 2001.

Vol. 1974: S. Kapoor, S. Prasad (Eds.), FST TCS 2000: Foundations of Software Technology and Theoretical Computer Science. Proceedings, 2000. XIII, 532 pages. 2000.

Vol. 1975: J. Pieprzyk, E. Okamoto, J. Seberry (Eds.), Information Security. Proceedings, 2000. X, 323 pages. 2000.

Vol. 1976: T. Okamoto (Ed.), Advances in Cryptology – ASIACRYPT 2000. Proceedings, 2000. XII, 630 pages. 2000.

Vol. 1977: B. Roy, E. Okamoto (Eds.), Progress in Cryptology – INDOCRYPT 2000. Proceedings, 2000. X, 295 pages. 2000.

Vol. 1978: B. Schneier (Ed.), Fast Software Encryption. Proceedings, 2000. VIII, 315 pages. 2001.

Vol. 1979: S. Moss, P. Davidsson (Eds.), Multi-Agent-Based Simulation. Proceedings, 2000. VIII, 267 pages. 2001. (Subseries LNAI).

Vol. 1980: M. Agosti, F. Crestani, G. Pasi (Eds.), Lectures on Information Retrieval. Proceedings, 2000. XI, 311 pages. 2001.

Vol. 1981: J.M.L.M. Palma, J. Dongarra, V. Hernández (Eds.), Vector and Parallel Processing – VECPAR 2000. Proceedings, 2000. XVI, 580 pages. 2001.

Vol. 1983: K.S. Leung, L.-W. Chan, H. Meng (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2000. Proceedings, 2000. XVI, 573 pages. 2000.

Vol. 1984: J. Marks (Ed.), Graph Drawing. Proceedings, 2001. XII, 419 pages. 2001.

Vol. 1985: J. Davidson, S.L. Min (Eds.), Languages, Compilers, and Tools for Embedded Systems. Proceedings, 2000. VIII, 221 pages. 2001.

Vol. 1987: K.-L. Tan, M.J. Franklin, J. C.-S. Lui (Eds.), Mobile Data Management. Proceedings, 2001. XIII, 289 pages. 2001.

Vol. 1988: L. Vulkov, J. Waśniewski, P. Yalamov (Eds.), Numerical Analysis and Its Applications. Proceedings, 2000. XIII, 782 pages. 2001.

Vol. 1989: M. Ajmone Marsan, A. Bianco (Eds.), Quality of Service in Multiservice IP Networks. Proceedings, 2001. XII, 440 pages. 2001.

Vol. 1990: I.V. Ramakrishnan (Ed.), Practical Aspects of Declarative Languages. Proceedings, 2001. VIII, 353 pages. 2001.

Vol. 1991: F. Dignum, C. Sierra (Eds.), Agent Mediated Electronic Commerce. VIII, 241 pages. 2001. (Subseries LNAI).

Vol. 1992: K. Kim (Ed.), Public Key Cryptography. Proceedings, 2001. XI, 423 pages. 2001.

Vol. 1993: E. Zitzler, K. Deb, L. Thiele, C.A.Coello Coello, D. Corne (Eds.), Evolutionary Multi-Criterion Optimization. Proceedings, 2001. XIII, 712 pages. 2001.

Vol. 1995: M. Sloman, J. Lobo, E.C. Lupu (Eds.), Policies for Distributed Systems and Networks. Proceedings, 2001. X, 263 pages. 2001.

Vol. 1997: D. Suciu, G. Vossen (Eds.), The World Wide Web and Databases. Proceedings, 2000. XII, 275 pages. 2001.

Vol. 1998: R. Klette, S. Peleg, G. Sommer (Eds.), Robot Vision. Proceedings, 2001. IX, 285 pages. 2001.

Vol. 1999: W. Emmerich, S. Tai (Eds.), Engineering Distributed Objects. Proceedings, 2000. VIII, 271 pages. 2001.

Vol. 2000: R. Wilhelm (Ed.), Informatics: 10 Years Back, 10 Years Ahead. IX, 369 pages. 2001.

Vol. 2001: G.A. Agha, F. De Cindio, G. Rozenberg (Eds.), Concurrent Object-Oriented Programming and Petri Nets. VIII, 539 pages. 2001.

Vol. 2002: H. Comon, C. Marché, R. Treinen (Eds.), Constraints in Computational Logics. Proceedings, 1999. XII, 309 pages. 2001.

Vol. 2003: F. Dignum, U. Cortés (Eds.), Agent Mediated Electronic Commerce III. XII, 193 pages. 2001. (Subseries LNAI).

Vol. 2004: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing. Proceedings, 2001. XII, 528 pages. 2001.

Vol. 2006: R. Dunke, A. Abran (Eds.), New Approaches in Software Measurement. Proceedings, 2000. VIII, 245 pages. 2001.

Vol. 2007: J.F. Roddick, K. Hornsby (Eds.), Temporal, Spatial, and Spatio-Temporal Data Mining. Proceedings, 2000. VII, 165 pages. 2001. (Subseries LNAI).

Vol. 2009: H. Federrath (Ed.), Designing Privacy Enhancing Technologies. Proceedings, 2000. X, 231 pages. 2001.

Vol. 2010: A. Ferreira, H. Reichel (Eds.), STACS 2001. Proceedings, 2001. XV, 576 pages. 2001.

Vol. 2011: M. Mohnen, P. Koopman (Eds.), Implementation of Functional Languages. Proceedings, 2000. VIII, 267 pages. 2001.

Vol. 2012: D.R. Stinson, S. Tavares (Eds.), Selected Areas in Cryptography. Proceedings, 2000. IX, 339 pages. 2001.

Vol. 2013: S. Singh, N. Murshed, W. Kropatsch (Eds.), Advances in Pattern Recognition – ICAPR 2001. Proceedings, 2001. XIV, 476 pages. 2001.

Vol. 2015: D. Won (Ed.), Information Security and Cryptology – ICISC 2000. Proceedings, 2000. X, 261 pages. 2001.

Vol. 2016: S. Murugesan, Y. Deshpande (Eds.), Web Engineering. IX, 357 pages. 2001.

Vol. 2018: M. Pollefeys, L. Van Gool, A. Zisserman, A. Fitzgibbon (Eds.), 3D Structure from Images – SMILE 2000. Proceedings, 2000. X, 243 pages. 2001.

Vol. 2020: D. Naccache (Ed.), Topics in Cryptology – CT-RSA 2001. Proceedings, 2001. XII, 473 pages. 2001

Vol. 2021: J. N. Oliveira, P. Zave (Eds.), FME 2001: Formal Methods for Increasing Software Productivity. Proceedings, 2001. XIII, 629 pages. 2001.

Vol. 2022: A. Romanovsky, C. Dony, J. Lindskov Knudsen, A. Tripathi (Eds.), Advances in Exception Handling Techniques. XII, 289 pages. 2001

Vol. 2024: H. Kuchen, K. Ueda (Eds.), Functional and Logic Programming. Proceedings, 2001. X, 391 pages. 2001.

Vol. 2025: M. Kaufmann, D. Wagner (Eds.), Drawing Graphs. XIV, 312 pages. 2001.

Vol. 2026: F. Müller (Ed.), High-Level Parallel Programming Models and Supportive Environments. Proceedings, 2001. IX, 137 pages. 2001.

Vol. 2027: R. Wilhelm (Ed.), Compiler Construction. Proceedings, 2001. XI, 371 pages. 2001.

Vol. 2028: D. Sands (Ed.), Programming Languages and Systems. Proceedings, 2001. XIII, 433 pages. 2001.

Vol. 2029: H. Hussmann (Ed.), Fundamental Approaches to Software Engineering. Proceedings, 2001. XIII, 349 pages. 2001.

Vol. 2030: F. Honsell, M. Miculan (Eds.), Foundations of Software Science and Computation Structures. Proceedings, 2001. XII, 413 pages. 2001.

Vol. 2031: T. Margaria, W. Yi (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. Proceedings, 2001. XIV, 588 pages. 2001.

Vol. 2032: R. Klette, T. Huang, G. Gimel'farb (Eds.), Multi-Image Analysis. Proceedings, 2000. VIII, 289 pages. 2001.

Vol. 2033: J. Liu, Y. Ye (Eds.), E-Commerce Agents. VI, 347 pages. 2001. (Subseries LNAI).

Vol. 2034: M.D. Di Benedetto, A. Sangiovanni-Vincentelli (Eds.), Hybrid Systems: Computation and Control. Proceedings, 2001. XIV, 516 pages. 2001.

Vol. 2035: D. Cheung, G.J. Williams, Q. Li (Eds.), Advances in Knowledge Discovery and Data Mining – PAKDD 2001. Proceedings, 2001. XVIII, 596 pages. 2001. (Subseries LNAI).

Vol. 2037: E.J.W. Boers et al. (Eds.), Applications of Evolutionary Computing. Proceedings, 2001. XIII, 516 pages. 2001.

Vol. 2038: J. Miller, M. Tomassini, P.L. Lanzi, C. Ryan, A.G.B. Tettamanzi, W.B. Langdon (Eds.), Genetic Programming. Proceedings, 2001. XI, 384 pages. 2001.

Vol. 2039: M. Schumacher, Objective Coordination in Multi-Agent System Engineering. XIV, 149 pages. 2001. (Subseries LNAI).

Vol. 2040: W. Kou, Y. Yesha, C.J. Tan (Eds.), Electronic Commerce Technologies. Proceedings, 2001. X, 187 pages. 2001.

Vol. 2042: K.-K. Lau (Ed.), Logic Based Program Synthesis and Transformation. Proceedings, 2000. VIII, 183 pages. 2001.

Vol. 2043: , D. Craeynest, A. Strohmeier (Eds.), Reliable Software Technologies – Ada-Europe 2001. Proceedings, 2001. XV, 405 pages. 2001.

Vol. 2044: S. Abramsky (Ed.), Typed Lambda Calculi and Applications. Proceedings, 2001. XI, 431 pages. 2001.

Vol. 2045: B. Pfitzmann (Ed.), Advances in Cryptology – EUROCRYPT 2001. Proceedings, 2001. XII, 545 pages. 2001.

Vol. 2048: J. Pauli, Learning Based Robot Vision. IX, 288 pages. 2001.

Vol. 2051: A. Middeldorp (Ed.), Rewriting Techniques and Applications. Proceedings, 2001. XII, 363 pages. 2001.

Vol. 2052: V.I. Gorodetski, V.A. Skormin, L.J. Popyack (Eds.), Information Assurance in Computer Networks. Proceedings, 2001. XIII, 313 pages. 2001.

Vol. 2053: O. Danvy, A. Filinski (Eds.), Programs as Data Objects. Proceedings, 2001. VIII, 279 pages. 2001.

Vol. 2054: A. Condon, G. Rozenberg (Eds.), DNA Computing. Proceedings, 2000. X, 271 pages. 2001.

Vol. 2055: M. Margenstern, Y. Rogozhin (Eds.), Machines, Computations, and Universality. Proceedings, 2001. VIII, 321 pages. 2001.

Vol. 2056: E. Stroulia, S. Matwin (Eds.), Advances in Artificial Intelligence. Proceedings, 2001. XII, 366 pages. 2001. (Subseries LNAI).

Vol. 2057: M. Dwyer (Ed.), Model Checking Software. Proceedings, 2001. X, 313 pages. 2001.

Vol. 2059: C. Arcelli, L.P. Cordella, G. Sanniti di Baja (Eds.), Visual Form 2001. Proceedings, 2001. XIV, 799 pages. 2001.

# Table of Contents

## Invited Talks



## Regular Papers

## System Descriptions

# Computing and Proving

Henk Barendregt

University of Nijmegen
henk@cs.kun.nl

**Abstract.** Computer mathematics is the enterprise to represent substantial parts of mathematics on a computer. This is possible also for arbitrary structures (with non-computable predicates and functions), as long as one also represents proofs of known properties of these. In this way one can construct a 'Mathematical Assistant' that verifies the well-formedness of definitions and statements, helps the human user to develop theories and proofs.
An essential part of the enterprise consists of a reliable representation of computations $f(a) = b$, say for $a$, $b$ in some concrete set $A$. We will discuss why this is so and present two reliable ways to do this. One consists of following the trace of the computation in the formal system used to represent the mathematics 'from the outside'. The other way consist of doing this 'from the inside', building the assistant around a term rewrite system. The two ways will be compared.
Other choices in the design of a Mathematical Assistant are concerned with the following qualities of the system
1. reliability;
2. choice of ontology;
3. choice of quantification strength;
4. constructive or classical logic;
5. aspects of the user interface.
These topics have been addressed by a number of 'competing' projects, each in a different way. From many of these systems one can learn, but a system that is satisfactory on all points has not yet been built. Enough experience through case studies has been obtained to assert that now time is ripe for building a satisfactory system.

A. Middeldorp (Ed.): RTA 2001, LNCS 2051, p. 1, 2001.

# Rewriting for Deduction and Verification

Michael Rusinowitch

LORIA — INRIA
615, rue du Jardin Botanique
BP 101, 54602 Villers-lès-Nancy Cedex France
rusi@loria.fr

**Abstract.** Rewriting was employed even in early theorem-proving systems to improve efficiency. Theoretical justifications for its completeness were developed through work on Knuth-Bendix completion and convergent reduction systems, where the key-notions of termination and confluence were identified. A first difficulty with reduction in automated theorem-proving is to ensure termination of simplification steps when they are interleaved with deduction. Term orderings have been widely investigated and provide us with good practical solutions to termination problems. A second difficulty is to keep completeness with unidirectional use of equations, both for deduction and simplification, which amounts to restoring a confluence property. This is obtained by extending the superposition rule of Knuth-Bendix procedure to first-order clauses.
Rewrite-based deduction has found several applications in formal verification. We shall outline some of them in the presentation. In computer-assisted verification, *decision procedures* are typically applied for eliminating trivial subgoals (represented, for example, as sequents modulo a background theory). The computation of limit sets of formulas by iterating superposition rules generalizes Knuth-Bendix completion and permits the uniform design of these decision procedures. Rewriting combined with a controlled instantiation mechanism is also a powerful *induction* tactic for verifying safety properties of infinite-state systems. A nice feature of the so-called rewriting induction approach is that it allows for the refutation of false conjectures too. A more recent application of rewriting to verification concerns *security protocols*. These protocols can be compiled to rewrite systems, since rewriting nicely simulates the actions of participants and malicious environments.

A. Middeldorp (Ed.): RTA 2001, LNCS 2051, p. 2, 2001.

# Universal Interaction Systems with Only Two Agents

Denis Bechet

LIPN - UPRES-A 7030
Institut Galilée
Université Paris 13 & CNRS
99, avenue J.-B. Clément
93430 VILLETANEUSE
France
Denis.Bechet@lipn.univ-paris13.fr

**Abstract.** In the framework of interaction nets [6], Yves Lafont has proved [8] that every interaction system can be simulated by a system composed of 3 symbols named $\gamma$, $\delta$ and $\epsilon$. One may wonder if it is possible to find a similar universal system with less symbols. In this paper, we show a way to simulate every interaction system with a specific interaction system constituted of only 2 symbols. By transitivity, we prove that we can find a universal interaction system with only 2 agents. Moreover, we show how to find such a system where agents have no more than 3 auxiliary ports.

## 1 Introduction

In [6], Yves Lafont introduces *interaction nets*, a programming paradigm inspired by Girard's proof nets for *linear logic* [3]. Some translations from $\lambda$-calculus into interaction nets [9,4,5] or from proof nets [7,10,2,1,11] show that universal interaction systems are interesting for computation. We can explain this interest for these translations by the fact that computation with interaction nets is purely local and naturally confluent. Reductions can be made in parallel. Moreover, the number of steps that are necessary to reduce completely a net is independent of the way one may choose. From the point of view of $\lambda$-calculus, translations used in [4,5] captures optimal reduction.

In [8], Lafont introduces a universal interaction system with only three different symbols $\gamma$, $\delta$ and $\epsilon$. $\delta$ and $\epsilon$ are respectively a duplicator and an eraser and $\gamma$ is a constructor. This system preserves the complexity of computation for a particular system. The number of steps that are necessary to reduce a simulated interaction net is just (at most) multiplied by a constant (which depends only on the simulated system and not on the size of the simulated net).

One may wonder if it is possible to find a simpler universal interaction system with only 2 symbols. This paper answers yes to this question. In fact, we prove that we can simulate a particular interaction system with only two symbols. By simulating a universal system, we prove that a universal system constituted of only two symbols exists.

A. Middeldorp (Ed.): RTA 2001, LNCS 2051, pp. 3–14, 2001.