Rho-Calculi for Computation and Logic

Claude Kirchner[1]

1 Inria
Domaine de Voluceau – Rocquencourt B.P. 105 – 78153 Le Chesnay, France
claude.kirchner@inria.fr

Abstract

The rho-calculi provide enlightening concepts for both computing and reasoning as well as their combination. They consist in the generalization of lambda-calculus to structures like terms, propositions or graphs and we will show how their interrelations with deduction provide powerful frameworks for the next generation of proof assistants.

1998 ACM Subject Classification F.4.1 Mathematical Logic – Lambda calculus and related systems, I.2.3 Deduction and Theorem Proving, F.4.2 Grammars and Other Rewriting Systems

Keywords and phrases Rewriting calculus, rho-calculus, deduction modulo

Digital Object Identifier 10.4230/LIPIcs.RTA.2012.2

Category Invited Paper

> *It is reasonable to hope that the relationship between computation and mathematical logic will be as fruitful in the next century as that between analysis and physics in the last. The development of this relationship demands a concern for both applications and for mathematical elegance.*
>
> John McCarthy
> A Basis for a Mathematical Theory of Computation, 1963

1 Structures for computation

When one wants to caracterize informatics as a science, one main concept is prominent: computation. When the computation of 2+2 leads to 4, this transformation can be described and performed in many different ways, for instance using lambda-calculus or using rewriting. If the structure of the information to be transformed is available, it is well known and widely used that this may result in a huge difference in the complexity of the computation: computing with Church numbers in the lambda-calculus is not the most efficient way to implement arithmetic. Rewriting is one of the universal computational model that allows the embedding of structures as a first class concept.

These structures can be very simple, yet quite useful, typically when just plain tree syntax is used. But they can be also quite elaborated, for example when satisfying some theory like associativity, commutativity, groups, list, arrays, arithmetic, . . .

To handle explicitly structures and fonctions, we defined the rho-calculus [11, 9] (also called rewriting calculus) and indeed a family of such calculi [12, 10, 23, 2, 18, 22, 1] to mention just a few. Related calculi should be mentioned like [21, 20].

These calculi, offering a natural integration of structures into the theory of functions and in particular in lambda-calculus, provide a useful background for rewriting theory, implementation and applications.

23rd International Conference on Rewriting Techniques and Applications (RTA'12).
Editor: A. Tiwari; pp. 2–4
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

But they offer also the right background to base the logical integration of computation into deduction, a deep and non-trivial goal as emphasized by the quote of John McCarthy.

2 Computation for logic

The complementarity of computation and deduction is known and used since the rise of logic. It has not always been well understood or accepted, like in the now settled controversy about the use of computations in the proof of the fourth color theorem. A formal treatment of this combination, called deduction modulo and initiated in [15], allows us to define deduction systems like the sequent calculus or natural deduction modulo some congruence defined by a theory implemented by a computational system such as rewriting. The consequences of reasoning modulo are deep. Typically, cut elimination does no hold for any congruence but sufficient conditions on the theories can be given [16, 8] allowing one to deal modulo important theories [17, 14, 19]. The second fondamental consequence is that the deduction modulo logical systems allow proving exactly the same theorems as when integrating the modulo part in the theory, but the proof are extremely different. Indeed they can be in some cases arbitrary shorter as shown in [7, 6].

An interesting point is that the proof terms representing proofs in deduction modulo are rho-terms [5], thus providing a complete picture of the articulation between structures, computations and proofs.

3 Next steps

This strong relationship between the rho-calculi and logic opens many research and application tracks. One is related to the implementation of these concepts to design proof assistants where the users are freed of the computational steps of the proof they are conducting, in the lines of the prototypes `https://www.rocq.inria.fr/deducteam/Dedukti` [3] or `http://rho.loria.fr/lemuridae.html` [4]. The relationship between rho-calculi and rewriting logic [13] has to be deepened. Challenging open questions like unification and matching in rho-calculi will lead to better understanding of the automation of reasoning, opening new trends for implementations and applications. Finally the relationship between computation and deduction and in particular the relationship with proof complexity is a challenging open research area.

Acknowledgements I want to thank all the colleagues with whom I have been sharing these ideas and work on rho-calculi and deduction modulo and in particulier Horatiu Cirstea, Nachum Dershowitz, Gilles Dowek, Hélène Kirchner and Luigi Liquori for many enlightening discussions.

References

1 Paolo Baldan, Clara Bertolissi, Horatiu Cirstea, and Claude Kirchner. A rewriting calculus for cyclic higher-order term graphs. *Mathematical Structures in Computer Science*, 2006.
2 Clara Bertolissi. *The graph rewriting calculus : properties and expressive capabilities*. Thèse de doctorat, Institut National Polytechnique de Lorraine - INPL, Oct 2005.
3 Mathieu Boespflug. *Conception d'un noyau de vérification de preuves pour le lambda-Pi-calcul modulo*. PhD thesis, École Polytechnique, Palaiseau, January 2011.
4 Paul Brauner. *Fondements et mise-en-oeuvre de la Super Déduction Modulo*. Phd thesis, INPL, Jun 2010.

5 Paul Brauner, Clément Houtmann, and Claude Kirchner. Principles of superdeduction. In Luke Ong, editor, *Proceedings of LICS*, jul 2007.

6 Guillaume Burel. *Bonnes démonstrations en déduction modulo*. PhD thesis, Université Henri Poincaré (Nancy 1), 2009.

7 Guillaume Burel. Efficiently simulating higher-order arithmetic by a first-order theory modulo. *Logical Methods in Computer Science*, 7(1), 2011.

8 Guillaume Burel and Claude Kirchner. Cut elimination in deduction modulo by abstract completion. In Sergei Artemov and Anil Nerode, editors, *Proc. of the Symposium on Logical Foundations of Computer Science*, Lecture Notes in Computer Science, New-York, USA, jun 2007. Springer Verlag.

9 Horatiu Cirstea. *Rewriting Calculus: Foundations and Applications*. PhD thesis, Université Henri Poincaré - Nancy I, 2000.

10 Horatiu Cirstea, Germain Faure, and Claude Kirchner. A rho-calculus of explicit constraint application. In *Proceedings of the 5th workshop on rewriting logic and applications*. Electronic Notes in Theoretical Computer Science, 2004.

11 Horatiu Cirstea and Claude Kirchner. The rewriting calculus — Part I *and* II. *Logic Journal of the Interest Group in Pure and Applied Logics*, 9:427–498, May 2001.

12 Horatiu Cirstea, Claude Kirchner, and Luigi Liquori. Matching Power. In Aart Middeldorp, editor, *12th International Conference on Rewriting Techniques and Applications (RTA'2001)*, volume 2051 of *Lecture Notes in Computer Science*, pages 77–92, Utrecht (The Netherlands), May 2001. Springer. Also available as Technical Report A01-R-201, LORIA, Nancy (France).

13 Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn L. Talcott, editors. *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, volume 4350 of *Lecture Notes in Computer Science*. Springer, 2007.

14 Gilles Dowek, Thérèse Hardin, and Claude Kirchner. HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic. *Mathematical Structures in Computer Science*, 11(1):21–45, 2001.

15 Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31(1):33–72, Nov 2003.

16 Gilles Dowek and Benjamin Werner. Proof normalization modulo. *The Journal of Symbolic Logic*, 68(4):1289–1316, 2003.

17 Gilles Dowek and Benjamin Werner. Arithmetic as a theory modulo. In Jürgen Giesl, editor, *Term Rewriting and Applications, 16th International Conference, RTA 2005, Nara, Japan, April 19-21, 2005, Proceedings*, volume 3467 of *Lecture notes in Computer Science*, pages 423–437, 2005.

18 Germain Faure. *Structure et modèles de calcul de réécriture*. Thèse de doctorat, Université Henri Poincaré Nancy, 2007.

19 Olivier Hermant. Resolution is cut-free. *Journal of Automated Reasoning*, 44(3):245–276, 03 2010.

20 C. Barry Jay and Delia Kesner. First-class patterns. *J. Funct. Program.*, 19(2):191–225, 2009.

21 Jan Willem Klop, Vincent van Oostrom, and Roel C. de Vrijer. Lambda calculus with patterns. *Theor. Comput. Sci.*, 398(1-3):16–31, 2008.

22 Luigi Liquori and Bernard P. Serpette. irho: an imperative rewriting calculus. In *6th Conference on Principles and Practice of Declarative Programming – PPDP'04*, pages 167–178. ACM, 2004.

23 Benjamin Wack. *Typage et déduction dans le calcul de réécriture*. Thèse de doctorat, Université Henri Poincaré - Nancy I, October, 7- 2005.