Nominal Techniques for Software Specification and Verification

Maribel Fernández
Department of Informatics, King's College London, UK

Abstract

In this talk we discuss the nominal approach to the specification of languages with binders and some applications to programming languages and verification.

2012 ACM Subject Classification Theory of computation → Logic and verification; Theory of computation → Lambda calculus; Theory of computation → Equational logic and rewriting

Keywords and phrases Binding operator, Nominal Logic, Nominal Rewriting, Unification, Equational Theories, Type Systems

Digital Object Identifier 10.4230/LIPIcs.FSCD.2023.1

Category Invited Talk

Funding This work is partially funded by the Royal Society (International Exchanges, grant number IES\R2\212106).

Acknowledgements The work described in this abstract is the result of collaborations with several researchers, cited below. Special thanks to my PhD students and co-authors.

1 Overview

The nominal approach to the specification of languages with binding operators, introduced by Gabbay and Pitts [27, 21, 20], has its roots in nominal set theory [26]. Its user-friendly syntax and first-order presentation (indeed, nominal logic [25] is defined as a theory in first-order logic) makes formal reasoning about binding operators similar to conventional on-paper reasoning.

Nominal logic uses the well-understood concept of *permutation groups acting on sets* to provide a rigorous, first-order treatment of common informal practice to do with fresh and bound names. Nominal matching and nominal unification [34, 35] (which work modulo α-equivalence) are decidable and efficient algorithms exist [7, 8, 22, 9], which are the basis for efficient implementations of nominal rewriting [19, 17, 18].

A number of systems (such as Nominal Isabelle [33]) highlighted the benefits of the nominal approach, which gave rise to elegant formalisations of Gödel's theorems [24] and the π-calculus [5] and to advances in programming language semantics [23]. However, there are still some obstacles to the inclusion of nominal features in programming languages and verification environments.

In this talk, I will present our current work towards incorporating nominal techniques into two widely-used rule-based first-order verification environments: the K specification framework [29] and the Maude programming language [11, 12].

An important component of rule-based programming and verification environments is the algorithm used to check equivalence of terms and to solve equations (unification). In practice, unification problems arise in the context of equational axioms (e.g., to take into account associative and commutative (AC) operators [32, 31, 13, 14, 6]). The first part of the talk will discuss notions of α-equivalence modulo associativity and commutativity axioms [1],

8th International Conference on Formal Structures for Computation and Deduction (FSCD 2023).
Editors: Marco Gaboardi and Femke van Raamsdonk; Article No. 1; pp. 1:1–1:4
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

extensions of nominal matching and unification to deal with AC operators [2], and the use of nominal narrowing [3] to deal with equational theories presented by convergent nominal rewriting rules.

Another important component of these environments is the type system. In the second part of the talk, I will discuss type systems for nominal languages (including polymorphic systems [15] and intersection systems [4]). Dependent type theories, the dominant approach to formalising programming languages, have been extended with nominal features [10, 28, 30]. A lambda-less nominal dependent type system is available [16] and we are currently working on a type checker for this system.

The talk is structured as follows: we will start with the definition of nominal logic (including the notions of fresh atoms and alpha-equivalence) followed by a brief introduction to nominal matching and unification. We will then define nominal rewriting, a generalisation of first-order rewriting that provides in-built support for alpha-equivalence following the nominal approach. Finally, we will discuss notions of nominal unification and rewriting modulo AC operators and briefly overview typed versions of nominal languages.

References

1 Mauricio Ayala-Rincón, Washington de Carvalho Segundo, Maribel Fernández, Daniele Nantes-Sobrinho, and Ana Oliveira. A formalisation of nominal α-equivalence with A, C, and AC symbols. *Theor. Comput. Sci.*, 781:3–23, 2019. doi:10.1016/j.tcs.2019.02.020.

2 Mauricio Ayala-Rincón, Washington de Carvalho Segundo, Maribel Fernández, Gabriel Ferreira Silva, and Daniele Nantes-Sobrinho. Formalising nominal C-unification generalised with protected variables. *Math. Struct. Comput. Sci.*, 31(3):286–311, 2021. doi:10.1017/S0960129521000050.

3 Mauricio Ayala-Rincón, Maribel Fernández, and Daniele Nantes-Sobrinho. Nominal Narrowing. In *1st International Conference on Formal Structures for Computation and Deduction, FSCD 2016*, page 11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

4 Maurício Ayala-Rincón, Maribel Fernández, Ana Cristina Rocha-Oliveira, and Daniel Lima Ventura. Nominal essential intersection types. *Theoretical Computer Science*, 737:62–80, 2018. doi:10.1016/j.tcs.2018.05.008.

5 Jesper Bengtson and Joachim Parrow. Formalising the pi-calculus using nominal logic. *LMCS*, 5(2), 2009. URL: http://arxiv.org/abs/0809.3960.

6 Alexandre Boudet, Evelyne Contejean, and Hervé Devie. A New AC Unification Algorithm with an Algorithm for Solving Systems of Diophantine Equations. In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science (LICS '90), Philadelphia, Pennsylvania, USA, June 4-7, 1990*, pages 289–299. IEEE Computer Society, 1990.

7 Christophe Calvès and Maribel Fernández. A polynomial nominal unification algorithm. *Theor. Comp. Sci.*, 403:285–306, August 2008.

8 Christophe Calvès and Maribel Fernández. Matching and alpha-equivalence check for nominal terms. *Journal of Comp. Syst. Sci.*, 76(5):283–301, 2010.

9 Christophe Calvès and Maribel Fernández. The first-order nominal link. In *Logic-Based Program Synthesis and Transformation - 20th International Symposium, LOPSTR 2010, Hagenberg, Austria, July 23-25, 2010, Revised Selected Papers*, volume 6564 of *LNCS*, pages 234–248. Springer, 2011.

10 James Cheney. A dependent nominal type theory. *Logical Methods in Computer Science*, 8(1), 2012.

11 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*, volume 4350 of *LNCS*. Springer, 2007. doi:10.1007/978-3-540-71999-1.

12 Francisco Durán, Steven Eker, Santiago Escobar, Narciso Martí-Oliet, José Meseguer, Rubén Rubio, and Carolyn L. Talcott. Programming and symbolic computation in Maude. *J. Log. Algebr. Meth. Program.*, 110, 2020.

13 François Fages. Associative-Commutative Unification. In Robert E. Shostak, editor, *7th International Conference on Automated Deduction, Napa, California, USA, May 14-16, 1984, Proceedings*, volume 170 of *LNCS*, pages 194–208. Springer, 1984.

14 François Fages. Associative-Commutative Unification. *J. of Sym. Computation*, 3(3):257–275, 1987.

15 Elliot Fairweather and Maribel Fernández. Typed nominal rewriting. *ACM Transactions on Computational Logic*, 19(1):6:1–6:46, 2018. `doi:10.1145/3161558`.

16 Elliot Fairweather, Maribel Fernández, Nora Szasz, and Alvaro Tasistro. Dependent types for nominal terms with atom substitutions. In *Typed Lambda Calculus and Applications (Proceedings of TLCA)*, pages 180–195, 2015. `doi:10.4230/LIPIcs.TLCA.2015.180`.

17 Maribel Fernández and Murdoch J. Gabbay. Nominal rewriting. *Inf. Comput.*, 205(6):917–965, 2007.

18 Maribel Fernández and Murdoch J. Gabbay. Closed nominal rewriting and efficiently computable nominal algebra equality. In *LFMTP*, 2010.

19 Maribel Fernández, Murdoch J. Gabbay, and Ian Mackie. Nominal rewriting systems. In *PPDP*, pages 108–119. ACM Press, August 2004.

20 Murdoch J. Gabbay. Foundations of nominal techniques: logic and semantics of variables in abstract syntax. *Bulletin of Symbolic Logic*, 2011.

21 Murdoch J. Gabbay and Andrew M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13(3–5):341–363, July 2001.

22 Jordi Levy and Mateu Villaret. An efficient nominal unification algorithm. In *Proceedings of the 21st International Conference on Rewriting Techniques and Applications (RTA 2010)*, volume 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 209–226. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2010.

23 Andrzej S. Murawski and Nikos Tzevelekos. Nominal game semantics. *FTPL*, 2:4:191–269, 2016. `doi:10.1561/2500000017`.

24 Lawrence C. Paulson. A mechanised proof of Gödel's incompleteness theorems using Nominal Isabelle. *J. Autom. Reasoning*, 55(1):1–37, 2015. `doi:10.1007/s10817-015-9322-8`.

25 Andrew M. Pitts. Nominal logic: A first order theory of names and binding. In *TACS*, volume 2215 of *LNCS*, pages 219–242. Springer, 2001.

26 Andrew M Pitts. *Nominal sets: Names and symmetry in computer science*. Cambridge UP, 2013.

27 Andrew M. Pitts and Murdoch J. Gabbay. A metalanguage for programming with bound names modulo renaming. In *Proceedings of the 5th international conference on the mathematics of program construction (MPC 2000)*, volume 1837 of *LNCS*, pages 230–255. Springer, December 2000. URL: `http://www.gabbay.org.uk/papers.html#metpbn`.

28 Andrew M. Pitts, Justus Matthiesen, and Jasper Derikx. A dependent type theory with abstractable names. *Electr. Notes Theor. Comput. Sci.*, 312:19–50, 2015. `doi:10.1016/j.entcs.2015.04.003`.

29 Grigore Rosu and Traian-Florin Serbanuta. An overview of the K semantic framework. *J. Log. Algebr. Program.*, 79(6):397–434, 2010. `doi:10.1016/j.jlap.2010.03.012`.

30 Ulrich Schöpp and Ian Stark. A Dependent Type Theory with Names and Binding. In *CSL*, pages 235–249, 2004.

31 Mark Stickel. A Unification Algorithm for Associative-Commutative Functions. *J. of the ACM*, 28(3):423–434, 1981.

32 Mark E. Stickel. A Complete Unification Algorithm for Associative-Commutative Functions. In *Advance Papers of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, September 3-8, 1975*, pages 71–76, 1975.

33 Christian Urban. Nominal techniques in Isabelle/HOL. *J. Autom. Reason.*, 40(4):327–356, May 2008.

34 Christian Urban, Andrew M. Pitts, and Murdoch J. Gabbay. Nominal unification. In *CSL*, volume 2803 of *LNCS*, pages 513–527. Springer, December 2003. URL: `http://www.gabbay.org.uk/papers.html#nomu`, `doi:10.1016/j.tcs.2004.06.016`.

35 Christian Urban, Andrew M. Pitts, and Murdoch J. Gabbay. Nominal unification. *Theor. Comp. Sci.*, 323(1–3):473–497, 2004. `doi:10.1016/j.tcs.2004.06.016`.