Curriculum Vitæ et Studiorum

Massimo Nocentini

massimo.nocentini@gmail.com, massimo.nocentini@unifi.it

https://github.com/massimo-nocentini

September 19, 2018

Personal data

I was born in Italy on 8 January 1986 and I live in Florence, Italy.

Education

Currently I'm a PhD student at the University of Florence, affiliated to the Dipartimento di Statistica, Informatica e Applicazioni (DiSIA [1]), supervised by prof *Donatella Merlini* [2]; in the past,

- Master Laurea degree in Computer Science, thesis title *Patterns in Riordan arrays*, supervised by prof. Donatella Merlini, University of Florence, 2015.
- Laurea degree in Computer Science, thesis title *Analysis of metabolic networks based on connection properties*, supervised by prof. Pierluigi Crescenzi, University of Florence, 2012.
- Maturity exam on Computer Science, Meucci Technical Institute, ABACUS project, Florence, 2005.

[1] https://www.disia.unifi.it/

[2] http://local.disia.unifi.it/merlini/

Scientific activity

His research activity concerns (i) the study of *formal methods and their applications* to the analysis of algorithms and data structures, (ii) supporting them with *software abstractions* implemented using functional programming languages, (iii) toward the field of *mechanized mathematics*. A solid base for such methods comes from the field of analytic combinatorics, which comprises tools such as generating functions, Riordan arrays and the symbolic method. Many interesting books by Flajolet and Sedgewick[3], Knuth[4] and Graham et al. [5] exist on those topics; moreover, see Harrison[6], Friedman and Felleisen [7] and Byrd et al.[8] for implementation aspects.

He desires to have a solid grasp of such powerful techniques in order to think about combinatorial *interpretations* of analytic results about classes of abstract objects in order to show *combinatorial meanings* and, possibly, characterizations in terms of lattice paths, urn models, bracelet configurations, boards tiling and so on, in the spirit of Benjamin and Quinn [9] and Stanley[10].

He believes that abstract and formal contexts should be paired up with sounding computer programs that show their beauty and elegance; this parallel path allows him to code in Lisp, Python, OCaml, Smalltalk and Haskell during his daily work.

[3] Flajolet and Sedgewick. *Analytic Combinatorics*, Cambridge University Press, 2009.

[4] Knuth. *The Art of Computer Programming*, vol. 1-3, Addison-Wesley, 1973.

[5] Graham, Knuth and Patashnik. *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley, 1994.

[6] Harrison. *Handbook of Practical Logic and Automated Reasoning*, Cambridge University Press, 2009; and *The HOL Light theorem prover*, User manual, 2017.

[7] Friedman and Felleisen. *The Little Schemer* and *The Seasoned Schemer* and *The Little MLer*, MIT Press.

[8] Byrd, Friedman and Kiselyov. *The Reasoned Schemer*, MIT Press.

[9] Benjamin and Quinn. *Proofs that really counts*, Mathematical Association of America, 2003.

[10] Stanley. *Enumerative combinatorics. Vol. 1&2*, Cambridge University Press.

Currently, he is supervised and collaborates with prof *Donatella Merlini* on advanced topic about *Riordan arrays*, in particular on binary words avoiding patterns, lattice paths enumeration problems and transformations of infinite sequences of numbers; he would deepen his understanding of those concepts in order to make them central in his PhD thesis.

Moreover, he is collaborating with prof *Marco Maggesi* [11] to enhance the HOL Light theorem prover with an extension of the goals and tactics mechanism to support the relational paradigm, in the spirit of µkanren[12].

[11] http://web.math.unifi.it/users/maggesi/

[12] Hemann and Friedman. µ*Kanren: a Minimal Functional Core for Relational Programming*, Scheme2013, Alexandria.

Papers

- Donatella Merlini, Massimo Nocentini. *Functions and Jordan canonical forms of Riordan matrices*, currently under review by the journal *Linear Algebra and its Applications*, 2018.
- Donatella Merlini, Massimo Nocentini. *Algebraic generating functions for languages avoiding Riordan patterns*, in *Journal of Integer Sequences*, Volume 21, Article 18.1.3, 2018.
- Donatella Merlini, Massimo Nocentini. *Colouring Catalan triangle*.

Conferences

- *ESUG*, September 2018, Cagliari, Italy [13]: volunteer student and contributed the talk *Relational Programming in Smalltalk* [14].
- *ICFP*, September 2017, Oxford, UK [15]: volunteer student.
- *EuroPython*, July 2017, Rimini, Italy [16]: participant.
- *ECOOP*, July 2016, Rome, Italy [17]: volunteer student.
- *Second International Symposium on Riordan Arrays and Related Topics*, July 2015 Lecco, Italy [18]: contributed a talk about modular Catalan triangle $\mathcal{C}_{\equiv_2}$.

[13] https://esug.github.io/2018-Conference/conf2018.html

[14] https://github.com/massimo-nocentini/microkanrenst/releases/download/v1.0/esug.pdf

[15] https://conf.researchr.org/home/icfp-2017

[16] https://ep2017.europython.eu/

[17] http://2016.ecoop.org/

[18] https://www.mate.polimi.it/RART2015/

Seminars and Schools

- *Logic and Relational Programming* at Logic Department, University of Florence [19].
- *summary of 2nd year* PhD activities, University of Florence [20].
- *Algebraic gf avoiding Riordan patterns* at AORC Open School, Sungkyunkwan University [21].
- *EOIS tools* at AORC Open School, Sungkyunkwan University [22].
- *summary of 1st year* PhD activities, University of Florence [23].

[19] http://massimo-nocentini.github.io/PhD/mkpy/talk.html#

[20] http://massimo-nocentini.github.io/PhD/second-year-summary/talk.html#

[21] http://shb.skku.edu/_custom/skk/_common/board/download.jsp?attach_no=29038

[22] http://massimo-nocentini.github.io/PhD/skku-aorc-2017/oeistools.html#

[23] http://massimo-nocentini.github.io/PhD/first-year-summary/talk.html#

Teaching

He did two classes about *SymPy* to introduce symbolic manipulations on top of the Python language, within a course on *Analysis of Algorithms* taught by Donatella Merlini at the University of Florence; in addition, he translated lab sessions code from Maple to Python collected in notebooks available online [24].

[24] https://github.com/massimo-nocentini/pacc/tree/master/paa-course

Github

Recently he joined the *Square Bracket Associates* organization; moreover, he pushes into the following repos:

https://github.com/massimo-nocentini/on-scheme holds stuff about the Scheme dialect of Lisp. In particular, here he explores *continuations*, μ*Kanren*, *union-find structures*, *lazy streams memoization* and *abstract computing devices (Landin's SECD machine)*.

https://github.com/massimo-nocentini/microkanrenst holds his Smalltalk implementation of μKanren, shown at ESUG2018; precisely, it provides *triangular substitutions*, *complete (but unfair) solutions enumeration* and *structural induction by heavy double dispatchings*.

https://github.com/massimo-nocentini/Booklet-microKanren holds docs and explanations of the relational paradigm provided in the repo before, using Pillar as writing environment, just started.

https://github.com/massimo-nocentini/kanren-light holds a parallel goals and tactics mechanism for Harrison's *HOL Light* theorem prover that is inspired by μKanren, co-authored with prof Marco Maggesi.

https://github.com/massimo-nocentini/simulation-methods holds the implementation of the framework of matrices functions, namely the process that lifts a scalar function to a matrix function using eigenvalues and generalized Lagrange bases.

https://github.com/massimo-nocentini/oeis-tools holds a suite of tools to mine the Online Encyclopedia of Integer Sequences, providing a crawler, a pretty printer and a grapher of relations among sequences of numbers; implemented in Python, allows him to play with `async` and `await` primitives.

https://github.com/massimo-nocentini/competitive-programming holds his solutions to a few problems from the UVa only judge; moreover, some jupyter notebooks are provided about bitmasking, backtracking, tilings, Gray codes, recursively-defined structures and dynamic programming.

https://github.com/massimo-nocentini/recurrences-unfolding holds a Python implementation of a framework that allows arbitrary unfoldings of recurrence relations, such as the famous $f_{n+2} = f_{n+1} + f_n$; precisely, the main idea is to use each rec relation as a

rewriting rule that can be used in pattern matching to instantiate vars in subscripts.

https://github.com/massimo-nocentini/on-python holds study material to explore features provided by the Python language; precisely, he takes into account coroutines, metaprogramming, a double dispatcher and collects some pointers.

https://github.com/massimo-nocentini/microkanrenpy a Pythonic implementation of μKanren, providing *complete* and *fair* solutions enumerations and *impure logical operators*, paired up with a decent doc and a test suite that covers *all* questions of *The Reasoned Schemer* book.

https://github.com/massimo-nocentini/master-thesis holds the TEX sources and implementations of his Master Thesis defended at the University of Florence; precisely, he proposes another characterization of Riordan arrays that exposes their A-sequence's generating functions for the Bell subgroup.

https://github.com/massimo-nocentini/cagd holds Julia and Python implementations of de Casteljau algorithm, Bezier curves, BSplines, tensor product and triangular Bezier patches; he worked on this to fullfil the course of Computer Aided Graphic Design at University of Florence, given by prof Alessandra Sestini and prof Costanza Conti.

https://github.com/massimo-nocentini/theory-of-programming-languages holds a study of type inference according to Benjamin Pierce [25] and the corresponding implementation using SML/NJ.

[25] Pierce. *Types and Programming Languages*, MIT Press.

https://github.com/massimo-nocentini/on-the-little-schemer holds study material and translation of *The Little Schemer* using the *CommonLisp* language in order to understand higher-order programming and play with `lisp-unit`; moreover, a Java implementation of the `Y` combinator is given.

https://github.com/massimo-nocentini/reasoning-about-little-books holds study material using the SML language, some derivation of the `Y` combinator are given (even an older one [26]) and most definitions from the "little books" had been reworked.

[26] Friedman and Felleisen. *The Little LISPer*, MIT Press.

https://github.com/massimo-nocentini/network-reasoner holds a simulator of gas distribution networks, according to various parameters; joint work with engineer Fabio Tarani and implemented in C# on the Mono platform.

Working activity

During his studies he worked in middle-sized software houses [27] developing mainly client-server applications using industrial-strength languages such as Java and C#, for about eight years, part-time relationships in parallel with his studies.

[27] Formerly at `https://www.commitsoftware.it/`, lately at `http://www.negens.com/site/home.html`.