Curriculum Vitæ et Studiorum

Massimo Nocentini

Dipartimento di Statistica, Informatica e Applicazioni

50134, Florence, Italy

massimo.nocentini@unifi.it

https://github.com/massimo-nocentini

October 28, 2016

1 Personal data

Born in Italy, 8 January 1986; living in Via Ronco Corto 98, Florence 50143, Italy.

2 Education

Currently I'm a PhD student at the University of Florence, Dipartimento di Statistica, Informatica e Applicazioni (DiSIA), working with prof. Donatella Merlini[1].

- 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.

3 Scientific activity

His research activity is mainly concerned with the study of formal methods and their application to the analysis of algorithms and data structures, supported by implementations written using symbolic and functional languages. A solid base for such

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

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[2], Knuth[3] and Graham et al. [4] exist on those topics.

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 [5] and Stanley[6]. Moreover, he wants to apply such techniques and interpretations to the analysis of algorithms as far as the analytic aspect is concerned, and to data structures for the combinatorial one.

He believes that all this abstract and formal context should be paired up with sounding computer programs to show the beauty and elegance of such topics; this parallel path allows him to enhance and deepen his knowledge in functional and symbolic programming, using languages like Lisp, Python and Haskell in his daily work. For this reason, during the first year of his PhD, he wrote a bunch of Jupyter notebooks about Gray codes, backtracking algorithms applied to tiling problems and the generation of recursive structures, and finally, application of bit-masking techniques to speed up symbolic computations[7].

At the same time, he continues to work on Riordan Arrays, studied in his master thesis[8], focusing on new characterizations to spot properties of their structure. One example is the h-characterization $\mathcal{R}_{h(t)}$ of a Riordan array $\mathcal{R}$, developed and explored within the thesis. Another path that he is following is the study of two important objects, A-sequence $\{a_n\}_{n\in\mathbb{N}}$ and A-matrix $\{a_{ij}\}_{i,j\in\mathbb{N}}$ respectively, generalizing them in order to discover new combinatorial identities. This approach is supported by a framework written using the Python language that performs unfolding of recurrence relations from the symbolic point of view: this is a work in progress, yet ready to be stressed against relations of general interest.

The other topic of his thesis shows his interest in the description and formalization of Riordan arrays under the light of modular arithmetic. He has shown congruences about *Pascal* array $\mathcal{P}$ and its inverse $\mathcal{P}^{-1}$. He has also proved a formal characterization for the *Catalan* array $\mathcal{C}$. These results were presented in a talk contributed at a recent conference held in Lecco and is the topic of a submitted paper.

Currently, he is working with prof. Donatella Merlini on advanced topic involving Riordan arrays: in particular, on binary words avoiding patterns, lattice paths enumeration problems and transformations of infinite sequences of numbers. He would

[2] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*, Cambridge University Press, 2009.

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

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

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

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

[7] `https://github.com/massimo-nocentini/competitive-programming/tree/master/tutorials`

[8] Massimo Nocentini, *Patterns in Riordan Arrays*, October 2015, University of Florence

deepen his understanding of such topics in order to make them central in his PhD thesis; moreover, he is designing a symbolic framework to implement a subset of most important and useful definitions taken from literature on this field.

3.1 Papers submitted and in preparation

- Donatella Merlini, Massimo Nocentini. *Colouring Catalan triangle*, submitted.
- Donatella Merlini, Massimo Nocentini. *Algebraic generating functions for languages avoiding Riordan patterns*, in preparation.
- Donatella Merlini, Massimo Nocentini. *Patterns in Riordan arrays*[9], Second International Symposium on Riordan Arrays and Related Topics, Lecco, 2015.
- *Recurrence unfolding*[10]: We provide a framework, written using the Python language on top of SymPy module, to perform arbitrary unfolding of recurrence relations. The main idea is to consider a set of possibly mutually defined recurrence relations, call it Ω, and use each one of them as a *rewriting rule* in the sense of using the left-hand side ("lhs" for short) as a term to be matched in order to instantiate the right-hand side ("rhs", respectively) accordingly; finally, use the new rhs as a replacement for the term that starts the matching.

  Unlike "plain" substitution, we perform an extended matching strategy on the lhs, allowing the mathematician to write relations that include a coefficient in the lhs: so a term in the rhs matches successful if it is possible to find a substitution for free variables that makes equal both the indexed symbol and the coefficient.

  To the time of this document, we have a working prototype for relations that involve indexed terms of the form $f_{n_1,\ldots,n_k}$ for desired $k \in \mathbb{N}$, with the constraint that the recurrence relation use constant coefficients. We're working to fully handle arbitrary recurrence relations. For the sake of clarity, we show, first, an application to the sequence of Fibonacci numbers[11], according to the unary-indexed recurrence $f_{n+2} = f_{n+1} + f_n$; second, an application to the Pascal array [12], according to the doubly-indexed recurrence $d_{n+1,k+1} = d_{n,k} + d_{n,k+1}$.

  We aim to show possibly new or hard to recognize identities over classes of combinatorial objects counted by relations under study; therefore, this prototype could be seen as an helper for the mathematician to understand how a recurrence behaves doing unfolding, leaving to him/her the analytic check of spotted patterns seen while unfolding the recurrence. In preparation.

[9] `http://www1.mate.polimi.it/~munarini/RART2015/Abstracts/RART2015_Nocentini.pdf`
[10] `https://github.com/massimo-nocentini/recurrences-unfolding`
[11] `http://nbviewer.jupyter.org/github/massimo-nocentini/recurrences-unfolding/blob/master/notebooks/fibonacci-numbers-unary-indexed-unfolding.ipynb`
[12] `http://nbviewer.jupyter.org/github/massimo-nocentini/recurrences-unfolding/blob/master/notebooks/pascal-array-doubly-indexed-unfolding.ipynb`

- *Riordan Arrays*: Following ideas of unfolding recurrence relations, we consider matrices in the Riordan group, leaving them completely symbolical, focusing only on their A-sequences which dictates relations among coefficients. We choose a row r and start to unfold coefficients from row $r + 1$ to the last, in order to build a matrix that depends on a small set of coefficients, namely those lying on the former $r + 1$ rows. This allow us to rewrite the original matrix as a sum of "inner" matrices, all of them sharing the same relation among coefficients. We're working on a prototype that built such splitting, collecting material in a notebook [13]. In preparation.

- *OEIS mining*: We develop a *web-crawler* targeting the OEIS [14], in order to fetch sequences according to the *cross-references* section included in search results. Collected informations can be represented as a graph, where each node represent a sequence and there is a reference between two sequences if they appears in the cited section. Such graph can be manipulated in order to remove not interesting sequences for the study of interest, in order to apply graph mining techniques. This is an ongoing project, however we have a first application[15]. In preparation.

- *OEIS search result pretty printer*: On top of the web-crawler defined in the previous topic, we provide a bunch of Python functions to implement a *pretty printer* for search result returned by OEIS. Use Jupyter notebook as working environments, we provide an API in order to query the OEIS by either sequence identifier or segment or open content, parsing results as json documents and returning an IPython display object that shows them directly in your notebook. This approach has the following ideas: simple API to perform a search, filtering all sections provided by the OEIS, list and table representation for sequences that have one and two dimensions, respectively. We believe that having an integrated environment, without neither switching between browser tabs nor put references to external contents, allows us to build reproducible, self-contained notebooks. We provide a tutorial[16]. In preparation.

3.2 Conferences

- *ECOOP*[17], July 2016 Rome, Italy: attended as volunteer student.

[13] `http://nbviewer.jupyter.org/github/massimo-nocentini/recurrences-unfolding/blob/master/notebooks/matrix-recurrences-unfolding.ipynb`

[14] `http://oeis.org/`

[15] `http://nbviewer.jupyter.org/github/massimo-nocentini/competitive-programming/blob/master/tutorials/oeis-mining.ipynb?flush_cache=true`

[16] `http://nbviewer.jupyter.org/github/massimo-nocentini/competitive-programming/blob/master/tutorials/oeis-interaction.ipynb?flush_cache=true`

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

- *Second International Symposium on Riordan Arrays and Related Topics*[18], July 2015 Lecco, Italy: contributed talk about modular Catalan triangle $\mathcal{C}_{\equiv_2}$

3.3 Seminars

Given a summary of my first year PhD's activities, the talk is available on-line[19].

3.4 Teaching

Given a class about *SymPy*[20] 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, we translate lab sessions using SymPy objects in notebooks, freely available[21].

4 Working activity

During his studies he worked in middle-size software houses developing mainly client-server applications using industrial-strength languages such as Java and C#, for about eight years.

[18] https://www.mate.polimi.it/RART2015/
[19] http://massimo-nocentini.github.io/PhD/first-year-summary/talk.html#/
[20] http://www.sympy.org/en/index.html
[21] https://github.com/massimo-nocentini/pacc/tree/master/paa-course