# Europass Curriculum Vitae

## Personal information

| | |
|---|---|
| First name(s) / Surname(s) | Đurica Nikolić |
| Address(es) | Dipartimento di Informatica, Strada Le Grazie 15, 37134 Verona, Italy<br>Ca' Vignal 2, piano 1, stanza 69 |
| Telephone(s) | +39 3337755512 Skype: nikolic.durica |
| E-mail | durica.nikolic@univr.it |
| Homepage | http://profs.sci.univr.it/~nikolic/<br>http://www.cosbi.eu/index.php/people/people-research/1135-durica-nikolic |
| Nationality | Serbian |
| Date of birth | February 20th, 1983 |
| Gender | male |

## Current Position

- **PhD Student in Computer Science at University of Verona**
- **Junior Researcher at The Microsoft Research – University of Trento Centre for Computational and Systems Biology**

## Working and Accademic Experience

| | |
|---|---|
| Dates | Febryary 1st, 2012 - |
| Occupation | **Junior Researcher** |
| Institute | **Microsoft Research – University of Trento Centre for Computational and Systems Biology** |
| Research Area | Qualitative and Quantitative Static Analysis of Modeling Languages for Biological Systems |

| | |
|---|---|
| Dates | January 1st, 2010 - |
| Occupation | **PhD Student in Computer Science, University of Verona** |
| Research Area | Languages, Logic and Formal methods |
| Supervisors | Prof. Roberto Giacobazzi, Prof. Nicola Fausto Spoto |
| Main activities | Static Analysis, Abstract Interpretation, Program Verification, Model Checking, Program Slicing, Semantics |

### Project

**Julia (www.juliasoft.com)**, a static analyzer for Java bytecode, finds bugs in **Java** and **Android** programs well before they are run. It is is a semantical tool, based on a mathematical theory known as **abstract interpretation**. Julia checks all possible executions of a software and finds all possible bugs, inside the categories considered by the tool. The analyses that can be performed are: ***checks*** *- a* very simple and quick analysis that detects a great amount of information about the analyzed program; ***nullness*** - performs a deep, complete nullness analysis of the program which covers all program's executions; ***termination*** – this analysis proves termination of program's methods: they will not hang unexpectedly.

I have been working on different static analyses which deal with memory management and which improve the precision of principal analyses of Julia. Among others, I have been working on both theoretical formalization and implementation of **Array Initialization Analysis**, **Reachability Analysis of Program Variables**, **Definite Aliasing Analysis**. Our analyses are abstract interpretation-based, and our abstract domains are quite simple, which turn our analyses very fast. Moreover, experimental evaluations performed on real life programs (e.g., Android programs written by Google) of these analyses shows that they are very efficient, since they improve the precision of both nullness and termination analyses.

## Background

During the first year of my PhD I studied the theory of a lot of techniques for formal verification on programs. In particular, I concentrated on Abstract Interpretation and Model Checking, and I studied the way these two formal methods can interact. I gained some knowledge about the most used model checkers, such as SLAM and BLAST. These two tools use the well-known CEGAR (Counter-Example Guided Abstraction Refinement) technique, which was the main subject of the first year of my PhD.

## Publications

**Published papers:**

- Đ. Nikolić, R. Zunino, C. Priami. **A Rule-based and Imperative Language for Biochemical Modeling and Simulation**. Invited paper in Proceedings of the 10th International Conference on Software Engineering and Formal Methods (SEFM 2012). October 1st–5th 2012, Thessaloniki, Greece.
- Đ. Nikolić, F. Spoto. **Definite Expression Aliasing Analysis for Java Bytecode**. In Proceedings of the 9th International Colloquium on Theoretical Aspects of Computing (ICTAC 2012). September 24th–27th 2012, Bangalore, India.
- R. Giacobazzi, I. Mastroeni, Đ. Nikolić. **Strong Preservation by Model Deformation**. In Proceedings of the 6th International Symposium on Theoretical Aspects of Software Engineering (TASE 2012). July 4th-6th 2012, Beijing, China.
- Đ. Nikolić, F. Spoto. **Reachability Analysis of Program Variables**. In Proceedings of the 6th International Joint Conference on Automated Reasoning (IJCAR 2012). June 26th–July 1st 2012, Manchester, UK.
- Đ. Nikolić, F. Spoto. **Automaton-based Array Initialization Analysis**. In Proceedings of the 6th International Conference on Language and Automata Theory and Applications (LATA 2012). March 5th–9th 2012, A Coruña, Spain.
- I. Mastroeni, Đ. Nikolić. **Abstract Program Slicing: From Theory Towards an Implementation.** In Proceedings of the 12th International Conference on Formal Engineering Methods (ICFEM'10). November 16th -19th 2010, Shanghai, China.

**Submitted papers:**

- Đ. Nikolić, F. Spoto. **Reachability Analysis of Program Variables**. Submitted for publication, May 2012.
- Đ. Nikolić, F. Spoto. **Inference of Class Invariants for Arrays.** Submitted for publication, February 2012.

## International Schools

1. **Tools for Practical Software Verification**, **8th LASER Summer School on Software Engineering** by Edmund Clarke (Carnegie Mellon), Patrick Cousot (École Normale Supérieure), Patrice Godefroid (Microsoft Research), Rustan Leino (Mcrosoft Research), Bertrand Meyer (ETH Zurich), César Muñoz (NASA Langley Research Center), Christine Paulin-Mohring (Université Paris-Sud) and Andrei Voronkov (University of Manchester). September 4th – September 11th 2011, Elba Island, Italy

2. **International School on Abstract Interpretation** by Roberto Giacobazzi. June 1st – June 20th 2010, Universidad Politecnica de Madrid, Madrid, Spain

3. **First International Summer School on Information Security and Protection** by Christian Collberg, Jack Davidson, Roberto Giacobazzi and Yuan Xiang Gu. July 26th – July 31st 2010, Institute of Automation, Chinese Academy of Science, Beijing, China.

## Courses Attended

1. **Static Analysis and Software Protection** by Roberto Giacobazzi, January 2011 – April 2011, University of Verona, Italy

2. **Automatic Verification of Programs** by Maria Paola Bonacina, January 2011 – April 2011, University of Verona, Italy

3. **Special Topics in Artificial Intelligence** by Maria Paola Bonacina, April 2011 – May 2011, University of Verona, Italy

## Teaching

1. **Teaching Assistant** in Fundamentals of Computer Science, October 2011 – February 2012; October 2010 – February 2011

2. **Tutor** in Programming 1, October 2011 – June 2012; October 2010 – June 2011; October 2009 – June 2010

| | |
|---|---|
| Dates | September 1st, 2009 – December 31st, 2009 |
| Occupation | Research Associate |
| Project | **Abstract Slicing Obfuscation and De-Obfuscation** |

| | |
|---|---|
| Tutor | Roberto Giacobazzi |

## Education

| | |
|---|---|
| Dates | October 1st, 2007 – July 24th, 2009 |
| Title of qualification awarded | **Master in Computer Science** |
| Thesis | **An Abstract Interpretation-Based Framework for Program Slicing** |
| Valuation | **110/110 cum laude** |
| Principal subjects/occupational skills covered | Semantics, Security, Abstract Interpretation |
| Name and type of organization providing education and training | **Faculty of Mathematical, Physical and Natural Science, University of Verona, Italy** |
| Interesting Projects: | ***ZigBee**: implementation of a middleware which detects ZigBee devices and which is used by the main node for a communication with detected nodes. In collaboration with prof. Franco Fummi* |

***Thesis – An Abstract Interpretation-Based Framework for Program Slicing (Costruzione di un modello per lo Slicing basato su Interpretazione Astratta)**:* A review of the most well-known techniques of program slicing. An introduction to a framework used for definition of different forms of Program Slicing and for their comparison. We insert 4 forms of Conditional Program Slicing into existing hierarchy. We define a novel technique of Program Slicing based on Abstract Interpretation and we call it Abstract Program Slicing. We generalize the existing framework by defining Abstract Formal Framework and Abstract Unified Equivalence which permit insertion of new forms of slicing, Abstract Slicing, into existing hierarchy. We show that abstract forms of slicing are weaker than the corresponding concrete forms.Supervisor: dr. **Isabella Mastroeni.**

| | |
|---|---|
| Dates | October 1st, 2005 – July 18th, 2007 |
| Title of qualification awarded | **Bachelor of Computer Science** |
| Valuation | **110/110 cum laude** |
| Principal subjects/occupational skills covered | Mathematical Analysis, Algebra, Programming (Java), Operative Systems, Fundamentals of Informatics |
| Name and type of organization providing education and training | **Faculty of Mathematical, Physical and Natural Science, University of Verona, Italy** |

***Thesis – An efficient solution for Really Simple Syndication in a dynamic web page***: an optimization of a Java class which handles the notification page of all of the websites of the faculties of University of Verona. I implemented a data structure which memorizes all the relevant news until a new insertion or a modification of an existing database entry occurs, This way it is not necessary to comunicate with the central database any time that a request for the webpage is sent by a student. When a professor or a secretary inserts a new or modifies an existing news, the data structure is deleted. When the first request arrives, another data structure is generated. Moreover, I inserted the RSS service in these pages. Supervisor: dr. **Roberto Posenato**.

| | |
|---|---|
| Dates | October 1st, 2002 – July 10th, 2005 |
| Title of qualification awarded | **3 years of Electronic Engineering Course** |
| Principal subjects/occupational skills covered | Mathematical Analysis, Programming (Java, C, C++), Electronics, Electrical Engineering |
| Name and type of organization providing education and training | Faculty of Electronic Engineering, University of Niš, Serbia |

## Personal skills and competences

| | |
|---|---|
| Mother tongue(s) | **Serbian** |

Other language(s)

| Self-assessment | **Understanding** | | **Speaking** | | **Writing** |
|---|---|---|---|---|---|
| | Listening | Reading | Spoken interaction | Spoken production | |
| **English** | Excellent | Excellent | Excellent | Excellent | Excellent |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Italian** | | Excellent | | Excellent | | Excellent | | Excellent | | Excellent |

Computer skills and competences

Programming Languages: Java, C, C++, Pascal, Delphi, Php