



## Luca Negrini

**Nationality:** Italian

(+39) 3495128700

**Date of birth:** 01/10/1993

**Gender:** Male

**Email address:** [luca.negrinii@gmail.com](mailto:luca.negrinii@gmail.com)

**Website:** <https://lucaneg.github.io/>

**LinkedIn :** <https://www.linkedin.com/in/luca-negrini-4a9813160/>

### WORK EXPERIENCE

---

#### Software Engineer and Research Scientist

**Corvallis Srl (previously JuliaSoft Srl)** [ 04/04/2018 – Current ]

- <https://corvallis.it/>

City: Verona

Country: Italy

- Development of the Julia Static Analyzer (now part of CodeSonar by GrammaTech, Inc.): software development and engineering in Java and C#
- Research topics: static analysis of object oriented software, tools for static analysis, abstract interpretation

### EDUCATION AND TRAINING

---

#### PhD Internship

**INRIA Paris, Team ANTIQUE** [ 03/01/2022 – 03/04/2022 ]

Address: 45 Rue d'Ulm, 75005 Paris (France)

<https://team.inria.fr/antique/>

Static analysis for Data Science: analyzing Jupyter Notebooks

Supervisor: [Caterina Urban](#)

#### PhD in Computer Science

**Università Cà Foscari Venezia** [ 09/2019 – Current ]

Address: Dorsoduro 3246, 30123 Venezia (Italy)

<https://www.unive.it/>

Level in EQF: EQF level 8

Design and implementation of a modular multi-language static analyzer

Supervisor: [Agostino Cortesi](#)

#### Master Degree in Computer Science

**Università degli Studi di Verona** [ 09/2015 – 03/2018 ]

Address: Via S. Francesco 22, 37129 Verona (Italy)

<https://www.univr.it/>

Final grade : 110/110 cum laude – Level in EQF: EQF level 7

## **Bachelor Degree in Computer Science**

*Università degli Studi di Verona* [ 09/2012 – 03/2016 ]

Address: Via S. Francesco 22, 37129 Verona (Italy)

<https://www.univr.it/>

Final grade : 90/110 – Level in EQF: EQF level 6

## **High School diploma in IT**

*ITIS Guglielmo Marconi* [ 09/2007 – 06/2012 ]

Address: Piazzale R. Guardini 1, 37138 Verona (Italy)

<https://www.marconiverona.edu.it/>

Final grade : 93/100 – Level in EQF: EQF level 4

## **PUBLICATIONS**

---

### **Static analysis for dummies: experiencing LiSA**

[2021]

<https://dl.acm.org/doi/10.1145/3460946.3464316>

Authors: P. Ferrara, L. Negrini, V. Arceri, A. Cortesi.

In: Proceedings of the 10th ACM SIGPLAN International Workshop on the State Of the Art in Program Analysis.

Conference: SOAP 2021, Online, June 22-23, 2021

### **Twinning Automata and Regular Expressions for String Static Analysis**

[2021]

[https://link.springer.com/chapter/10.1007/978-3-030-67067-2\\_13](https://link.springer.com/chapter/10.1007/978-3-030-67067-2_13)

Authors: L. Negrini, V. Arceri, P. Ferrara, A. Cortesi.

In: Proceedings of the 22nd International Conference on Verification, Model Checking, and Abstract Interpretation.

Conference: VMCAI 2021, Online, January 17-22, 2021

### **SARL: OO Framework Specification for Static Analysis**

[2020]

[https://link.springer.com/chapter/10.1007/978-3-030-63618-0\\_1](https://link.springer.com/chapter/10.1007/978-3-030-63618-0_1)

Authors: P. Ferrara, L. Negrini.

In: Software verification. Springer, Cham, 2020. PP. 3-20.

Conference: VSTTE 2020, Virtual conference, July 20-21, 2020

### **SARL: Framework Modeling for Static Analysis**

[2018]

[https://staticanalysis.org/tapas2018/abstracts/TAPAS\\_2018\\_Negrini\\_Ferrara.pdf](https://staticanalysis.org/tapas2018/abstracts/TAPAS_2018_Negrini_Ferrara.pdf)

Authors: L. Negrini, P. Ferrara.

In: Proceedings of the 9th Workshop on Tools for Automatic Program Analysis (TAPAS2018).

Conference: TAPAS 2018, Freiburg im Breisgau, Germany, August 28, 2018

## **TALKS AND SEMINARS**

---

### **Modular Multi-language analysis in LiSA**

[ 19/01/2022 ] INRIA Paris, ANTIQUE internal seminar, Online

## Modular Multi-language analysis in LiSA

[ 30/06/2021 ] Università Cà Foscari Venezia, Online seminar

## Twinning Automata and Regular Expressions for String Static Analysis

[ 18/01/2021 ] VMCAI 2021, Virtual conference

## SARL: OO Framework Specification for Static Analysis

[ 20/07/2020 ] VSTTE 2020, Virtual conference

## SARL: Framework Modeling for Static Analysis

[ 28/08/2018 ] TAPAS 2018, Freiburg im Breisgau, Germany

## TEACHING

---

### Lectures (4 hours) and online tutoring

[ Mar - May 2022 ] Università Cà Foscari Venezia, "Software Correctness, Security and Reliability" course

### Online tutoring

[Sept 2021 - Jan 2022] Università Cà Foscari Venezia, "Programmazione ad Oggetti - Mod. 1" course

### Lectures (4 hours) and online tutoring

[ Mar - May 2021 ] Università Cà Foscari Venezia, "Software Correctness, Security and Reliability" course

### Online tutoring

[ Mar - May 2020 ] Università Cà Foscari Venezia, "Software Correctness, Security and Reliability" course

## COMMUNITY SERVICES

---

### Reviewer

VALID 2021 - The Thirteenth International Conference on Advances in System Testing and Validation Lifecycle

## TOOLS AND SOFTWARE

---

### LiSA

As part of my PhD research project, I started the development of LiSA, together with the [Software and System Verification group](#) @ Ca' Foscari University of Venice, Italy. LiSA (Library for Static Analysis) eases the creation and implementation of static analyzers based on the Abstract Interpretation theory. LiSA provides an analysis engine that works on a generic and extensible control flow graph representation of the program to analyze. Abstract interpreters in LiSA are built for analyzing such representation, providing a unique analysis infrastructure for all the analyzers that will rely on it.

Building an analyzer upon LiSA boils down to writing a parser for the language that one aims to analyze, translating the source code or the compiled code towards the control flow graph representation of LiSA. Then, simple checks iterating over the results provided by the semantic analyses of LiSA can be easily de- fined to translate semantic information into warnings that can be of value for the final user.

LiSA is distributed under the MIT license, and is available on [GitHub](#).

<https://unive-ssv.github.io/lisa/>

## Tarsis

Tarsis is a new abstract domain for string values based on finite state automata. Standard finite state automata abstract domain has been shown to provide precise abstractions of string values when all the components of such strings are known, but with high computational cost. Instead of considering standard finite automata built over an alphabet of single characters, Tarsis considers automata that are built over an alphabet of strings, comprising a special value to represent statically unknown strings. Tarsis is maintained by the [Software and System Verification group](#) @ Ca' Foscari University of Venice, Italy, and it is available on [GitHub](#).

## MASTER THESIS

---

### Automatic Application Splitting

Supervisor: Prof. Fausto Spoto

Co-supervisor: Pietro Ferrara, PhD

Design and implementation of advanced algorithms and application of machine learning to obtain automatic application splitting for scaling up interprocedural static analyses to industrial software. Implementation and experimental evaluation have been performed with the Julia Static Analyzer for Java and C#.

## LANGUAGE SKILLS

---

Mother tongue(s): **Italian**

Other language(s):

**English**

**LISTENING B2 READING B2 WRITING B2**

**SPOKEN PRODUCTION B2 SPOKEN INTERACTION B2**