MARGARIDA FERREIRA

December 2024

margarida@cmu.edu marghrid.github.io

Research Interests

I am interested in **formal methods** and **program synthesis**, particularly in their application to complex networked systems. My research focuses on automatically generating correct implementations of stateful programs based on noisy and/or incomplete execution traces.

Education

PhD. Computer Science • *Carnegie Mellon University & Técnico Lisboa* • 2021–26
Advised by Professors Ruben Martins and Inês Lynce. Dual Degree PhD fellowship by CMU Portugal with Técnico Lisboa. Expected graduation: August 2026.

MSc. Computer Science and Engineering • *Técnico Lisboa* • 2018–20
Specialization in Artificial Intelligence and Algorithms. First year at the Technical University of Munich under Erasmus+ scholarship. Master thesis *Forest: An Interactive Multi-tree Synthesizer for Regular Expressions*, advised by Professor Inês Lynce and Dr. Miguel Neves (OutSystems).

BSc. Computer Science and Engineering • *Técnico Lisboa* • 2015–18
Top 2% GPA (18/20). Final-year research project *Satisfying Cooperative Path-finding*, advised by Professor Inês Lynce.

Publications

Program Synthesis From Partial Traces. **M. Ferreira**, V. Nicolet, J. Dodds, D. Kroening. **Under submission**.

Reverse-Engineering Congestion Control Algorithm Behavior. **M. Ferreira**, R. Ware, Y. Kothari, I. Lynce, R. Martins, A. Narayan, J. Sherry. **IMC 2024**.

Counterfeiting Congestion Control Algorithms. **M. Ferreira**, A. Narayan, I. Lynce, R. Martins, J. Sherry. **HotNets 2021**.

Forest: An Interactive Multi-tree Synthesizer for Regular Expressions. **M. Ferreira**, M. Terra-Neves, M. Ventura, I. Lynce, and R. Martins. **TACAS 2021**.

Experience

Applied Scientist Intern • *Amazon Web Services* • Fall 2024
I worked with the Automated Reasoning for Cloud Operations team, on building an efficient anomaly detection filter for large volumes of structured logs.

Applied Scientist Intern • *Amazon Web Services* • Summer 2023
I worked with the Automated Reasoning for Cloud Operations team, on automatically synthesizing provably correct scripts that allow users to manage their AWS infrastructure.

Junior researcher • *INESC-ID* • 2020–2021
I worked on synthesizing regular expressions from examples and on reverse engineering congestion control algorithms in the Automated Reasoning and Software Reliability group.

Research Intern • *Outsystems* • 2019–2020
I worked on the synthesis of regular expression form validations under the supervision of Miguel Neves and Miguel Ventura at the Engineering Department, Artificial Intelligence division.

Skills

I program mainly in **Python** and **Rust**. I am very familiar with Python's scientific libraries **NumPy**, **SciPy**, and **Matplotlib**. I also have experience with **Java** and **C++17**. I have extensive experience with constraint-solving frameworks, **SAT** and **SMT** (**Z3**, **CVC5**), and their APIs. I have limited experience with the solver-aided language **Rosette**, and proof assistants **Coq** and **Lean**.

Projects

Syren • 2023–24

We propose a new method to synthesize programs from partial execution traces. We combine compiler-like optimizing rewrites with programming-by-example to infer the program's hidden control- and data-flow. We show the applicability of our method in synthesizing cloud computing scripts from logs and shell scripts from kernel traces.

Jetstream • 2021–22

Jetstream simplifies, restructures, and verifies the semantics of virtual switch rulesets. We use SMT to find and remove redundancies from the ruleset, provably maintaining the original semantics. Our analysis of production rulesets shows that JetStream improves packet classification performance by up to 70% using state-of-the-art packet classifiers.

CCA synthesis • 2020–24

We use program synthesis to reverse-engineer congestion control algorithm implementations based on traces collected using the original implementation. Our synthesis procedure uses SMT to encode the space of candidate congestion control algorithms and tries to find one whose behavior minimizes the error against the given trace. Using this approach, we can synthesize simplified versions of several kernel CCAs using only a few traces.

Forest • 2019–20

Forest automatically synthesizes regular expressions that match a desired pattern expressed using examples. It uses an SMT solver to explore and prune the search space and to synthesize capture conditions that ensure the validity of numerical values in the input. We use Z3's regex theory to implement user interaction based on distinguishing inputs. Experimental results show that Forest outperforms previous state-of-the-art regex synthesizers.

Misc

I am an inventor of U.S. Patent Application No. 18/538,920 • Dec 2023

AWS published a blog post about my work as a PhD summer intern at AWS • Oct 2023

CMU Portugal published an article about my experience in the dual degree • May 2022

I co-designed and taught the AfterSchool Artificial Intelligence Course by Treetree2, for high school and middle school students who want to learn Programming and AI • Jan–Feb 2022

Fellowship & Honors

Dual Degree PhD Fellowship • *CMU Portugal* • 2021–26

Excellent Teacher Award (for very positive student evaluations) • *Técnico Lisboa* • 2020–21

Erasmus+ Scholarship • *European Commission* • 2018–19

New Talents in Artificial Intelligence Fellowship • *Calouste Gulbenkian Foundation* • 2017–18

Academic Excellence Certificates (for top 10% GPA) • *Técnico Lisboa* • 2015–2018, 2020