



UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

itee^{PhD}
information technology
electrical engineering



DIE
TI

UNI
NA

Francesco Altiero

Tree Kernels applications for Regression Test Prioritization

Tutor: prof. Adriano Peron
Cycle: XXXVI

co-Tutor: prof. Anna Corazza
Year: First

My background

- M.Sc. Degree in Computer Science @ DIETI - Federico II
 - Thesis: *Source Code Similarity using Abstract Syntax Trees and Partial Tree Kernels*
- Ph.D. fellowship funded by *UNINA*, started in November 2020
- My Research Group:
 - **Prof. Adriano Peron**, *DIETI – UNINA*
 - **Prof. Sergio di Martino**, *DIETI – UNINA*
 - **Prof. Anna Corazza**, *DIETI – UNINA*
 - **Luigi Libero Lucio Starace**, *ITEE PHD - UNINA*

Research field of interest

Software Verification

Regression Test Prioritization

Machine Learning applications to Software Testing

Empirical Software Engineering

Source Code Analysis

Code Clone Detection

Code Plagiarism Detection

Summary of study activities

	Courses	Seminars	Research	Tutorship
First Year	23,5	7,1	33	0
Expected	20 - 40	5 - 10	10 - 35	0 - 1,6

(Some) courses I attended:

- Combinatorial Optimization (6 ECTS)
- Data Analytics (6 ECTS)
- Scientifying Programming and Data Visualization with Python (2 ECTS)

Conferences attended:

- International Conference on Testing Software and Systems (ITCSS 2020)

Research activity: Overview (1/2)

The Problem: **Regression Test Prioritization**

What: **permute test-cases** to execute tests which **uncover faults** before others

When: **limited resources** for the verification phase to execute the **whole test-suite** of a software upon a **new release**

Whom: **software companies** which have **strict time-to-market constraints** or **pay** the usage of **testing environments**

Research activity: Overview (2/2)

- *Objective*: prioritize test-suite by inspecting **source code changes** between versions
- *Contribution*: define measures to evaluate **code similarity** employing **Tree Kernels** on **Abstract Syntax Trees**
- *Methodology*:
 1. Collect a **benchmark dataset**
 2. Prioritize test-cases which **cover** more **dissimilar** portions
 3. **Evaluate** prioritization performance with common metrics (e.g., **APFD**)
 4. **Statistically compare** results with those of state-of-art prioritization techniques

Products

[P1]	<p>Conference Paper</p> <p><i>Inspecting Code Churns to Prioritize Test Cases</i></p> <p><i>Altiero F., Corazza A., Di Martino S., Peron A. & Starace L. L. L.</i></p> <p>International Conference on Testing Software and Systems (ICTSS 2020)</p> <p>Status: <i>published</i></p>
[P2]	<p>Software Prototype</p> <p><i>Prioritization Platform</i></p> <p>Java implementation of the designed pipeline to perform prioritization experiments on benchmark software projects and produce performance metrics</p>
[P3]	<p>Software Extension</p> <p><i>Open Clover</i></p> <p>https://openclover.org/</p> <p>Added option to obtain <i>per-test</i> coverage reporting in XML output format</p>

Next Year...

- Deploy new techniques to assess code similarity
- Evaluate Tree Kernel prioritization techniques on software developed with *CI/CD* and *Agile methodologies*
- Extend *Prioritization Platform* adding new similarity-based techniques to prioritize
- Apply developed techniques to other topics, e.g., *Code Plagiarism*

Thank you all for your
attention!