

Jeremy Lacomis

📍 Pittsburgh, PA ✉ j.lacomis@gmail.com ☎ (570) 881-1283 🔗 jeremylacomis.com 🌐 jlacomis

About Me

I am a Postdoctoral Fellow at Carnegie Mellon University with an interest in automated software engineering. I have extensive experience with decompilers and low-level source code, including automatically easing the ability for reverse-engineers to effectively understand decompiled code. I am looking for opportunities to apply my experience to real-world problems.

Quick Guide

- Deep understanding of compilation, low-level programming, and decompilation.
- Experience combining machine learning with insights to lower the cognitive load of reverse-engineers.
- Comfortable with Python, C, Java, OCaml, and x86-64 assembly language.

Education

PhD	Carnegie Mellon University , Software Engineering	2017 - 2023
	<ul style="list-style-type: none">• Dissertation: Automatically Annotating Decompiled Code with Meaningful Names and Types• Advisors: Claire Le Goues and Bogdan Vasilescu	
BA	University of Virginia , Computer Science	2015 - 2017
AS	Piedmont Virginia Community College , Computer Science	2010 - 2015

Experience

Carnegie Mellon University , Mark Stehlik Postdoctoral Teaching Fellow	Pittsburgh, PA June 2023 – Present
<ul style="list-style-type: none">• Instructor of record for four semesters of an introductory software engineering course.• Designed new homework assignments, labs, and exams from scratch.• Organized and led teams of 10 TAs with up to two other co-instructors.	
GrammaTech , Software Engineer Intern	Ithaca, NY May 2019 – Aug 2019
<ul style="list-style-type: none">• Completely refactored the Python interface to the GrammaTech Intermediate Representation for Binaries (GTIRB), adding documentation and automated tests.• Added functionality to the Python interface for serializing and deserializing GTIRB tuples, vectors, and <code>intt64_t</code> types.• Did preliminary experimentation with GTIRB to identify ways to automatically reduce the attack surface of binaries without access to their source code.	

Selected Publications

A Human Study of Automatically Generated Decompiler Annotations	DSN 2025 (accepted)
Yuwei Yang, Sykler Grandel, Jeremy Lacomis , Edward J. Schwartz, Bogdan Vasilescu, Claire Le Goues, Kevin Leach	
<ul style="list-style-type: none">• Performed a human study of 40 reverse-engineers to evaluate the effectiveness of automated decompiler annotations.• Empirically demonstrated a lack of correlation between current best-known ML performance metrics and the effectiveness of variable type and name recovery.	

- Found that users prefer code with annotations over code without annotations, even when the annotations do not significantly increase accuracy or reduce the time taken to understand decompiled code.

A Taxonomy of C Decompiler Fidelity Issues

USENIX 2024

Luke Dramko, **Jeremy Lacomis**, Edward J. Schwartz, Claire Le Goues, Bogdan Vasilescu

- Created a comprehensive, hierarchical taxonomy of fidelity issues in decompiled code beyond names and types
- Compared the effectiveness of four different modern decompilers

VarCLR: Variable Semantic Representation Pre-training via Contrastive Learning

ICSE 2022

Qibin Chen, **Jeremy Lacomis**, Edward J. Schwartz, Claire Le Goues, Graham Neubig, Bogdan Vasilescu

- Presented VarCLR, a novel method based on contrastive learning that learns general-purpose variable semantic representations
- Created GITHUBRENAMES, a weakly-supervised dataset of variable renamings collected directly from GITHUB

Augmenting Decompiler Output with Learned Variable Names and Types

USENIX 2022

Distinguished Paper Award

Qibin Chen, **Jeremy Lacomis**, Edward J. Schwartz, Claire Le Goues, Graham Neubig, Bogdan Vasilescu

- Introduced the **DecompI**ler variable **ReTY**per (DIRTY), a Transformer-based neural network model designed to recover both semantic and syntactic variable types in decompiled code
- Collected the **Dataset for Idiomatic ReTyping** (DIRT), a dataset of nearly 1 million public C functions for training models for renaming and retyping variables

DIRE: A Neural Approach to Decompiled Identifier Naming

ASE 2019

Jeremy Lacomis, Pengcheng Yin, Edward J. Schwartz, Miltiadis Allamanis, Claire Le Goues, Graham Neubig, Bogdan Vasilescu

- Introduced the **Decompiled Identifier Renaming Engine** (DIRE), a technique for assigning meaningful names to decompiled variables using neural network techniques
- Presented a novel technique for generating corpora suitable for training both lexical and graph-based probabilistic models of variable names in decompiled code

Projects

DIRTY

github.com/CMUSTRUDEL/DIRTY 

- A Transformer-based neural network model designed to recover both semantic and syntactic variable types in decompiled code
- Configurable to also predict variable names at the same time
- Tools Used: Python, PyTorch, Hex-Rays

DIRE

github.com/CMUSTRUDEL/DIRE 

- A tool for assigning meaningful names to decompiled variables using neural network techniques
- Tools Used: Python, PyTorch, Hex-Rays

Professional Service and Affiliations

Selection Committee Chair, REUSE@CMU	2019 - 2023
Mentor, SCS Graduate Application Support Program	2020
Member, ISR DEI Committee	2020 - 2023
Journal Referee, Automated Software Engineering	2024

References

Claire Le Goues, Professor of Computer Science: Carnegie Mellon University

clegoues@andrew.cmu.edu

Zachary Fry, Senior Scientist III: GrammaTech

zfry@grammatech.com