

# Jan-Paul Vincent Ramos-Dávila

✉ mail@jpramos.me | 🏠 https://jpramos.me | 🌐 jpramos-me | 🌐 jpramos-me

## Education

### Cornell University

Ithaca, NY

B.A. in Computer Science, Conc. in Programming Languages & B.A. in Philosophy, Conc. in Logic

August 2021 - May 2025

## Research Experience

### Carnegie Mellon University

Pittsburgh, PA

Software Verification Research Intern, Advised by [Dr. Jonathan Aldrich](#) & [Dr. Jonathan Sunshine](#)

May 2022 - Present

- Exploring the application of gradual verification to smart contracts on the *Algorand* blockchain platform in developing [Gradually Verified Teal](#). [Funded by Algorand](#).
- Investigated second-order logic optimization techniques for naive runtime assertions in Gradual  $C_0$ , significantly optimizing benchmark results. [Funded by Amazon](#)
- Core contributor on the early development of a [Gradual Verification framework](#). Implemented a Property Based Testing tool<sup>1</sup> for evaluating soundness of [Gradual  \$C\_0\$](#)  and developed formal proofs for establishing semantic correspondence between static and dynamic checks.

### Cornell University

Ithaca, NY

Programming Languages Undergraduate Researcher, Advised by [Dr. Adrian Sampson](#)

October 2021 - December 2022

- Worked on a symbolic execution tool for verifying parallelism in Calyx.
- Implemented *Graphicionado Graph Analytics* algorithm in [Calyx](#) as a case study of the language and found/solved soundness bugs in the front-end. Worked in the [Computer Architecture & Programming Abstractions](#) group

## Publications

- [1] Jan-Paul Ramos-Dávila **Evaluating Soundness of a Gradual Verifier with Property Based Testing**<sup>1</sup> In *Principles of Programming Languages Student Research Competition* [Link] [Presentation] [Poster] & In *Cornell Undergraduate Research Journal*, CURJ Vol. 2 No. 1 [Link]

## Projects

### Incremental Specification Mining

[Cornell CS 6156 Runtime Verification](#)

- Instrumentation for Maven-based projects that *incrementally* mines specifications for runtime verification.
- Significantly decreases overhead for evolutionary-aware specification miners. Supports integration with [Javert](#) and [BDDMiner](#).

### EtaC

[Cornell CS 4120 Intro to Compilers](#)

- Compiler for the [Eta programming language](#) in OCaml with ~ 7,200 lines of code.
- Made use of [Jane Street's expect testing suite](#) for ~ 90% code coverage.

### RNAfoldml

[Cornell CS 3110 Functional Programming](#)

- OCaml package that enables users to input both RNA sequences in FASTA format and a set of constraints to predict RNA secondary structure.

### Diffeq-lang

[Senior High School Project](#)

- Domain-specific language for automatically solving and graphing differential equations via a [web interface](#).

## Awards and Honors

2023

**Winner, Third Place:** ACM SIGPLAN Symposium POPL SRC<sup>1</sup>

2022

**Travel Scholarship:** Second Prize in ACM SIGPLAN Conference PLDI

2020 & 2021

**Finalist, Mathematics:** Regeneron International Science and Engineering Fair

## Other Experience

## Utrecht University

Student, [Advanced Functional Programming Summer School](#)

- Participated in a mix of lectures, labs and a busy social program, focused in Master's level Haskell programming.

*Utrecht, Netherlands*

July 2023

## Volunteer Services

---

### ACM SIGPLAN ICFP 2023

Student Volunteer

*Seattle, WA*

2023

## Skills

---

### Languages

OCaml, Scala, Python,  $\LaTeX$ , Haskell, JavaScript, Java, C, Racket, Rust, Assembly, English, Español, Italiano

### Tools

Unix, Git, VSCode, IntelliJ IDEA, Shell, Neovim, Docker, Heroku