

# Jan-Paul Vincent Ramos-Dávila

[jpramos.me](https://jpramos.me) | [jvr34@cornell.edu](mailto:jvr34@cornell.edu) | [github/jpvinnie](https://github.com/jpvinnie) | [linkedin/jpv-ramos](https://www.linkedin.com/in/jpv-ramos)

## Education

### Cornell University

2021 - 2025

Bachelor of Arts in Computer Science & Philosophy

## Experience

### Carnegie Mellon University

06/2022 - Present

Research Intern

Advised by Dr. Jonathan Aldrich & Dr. Joshua Sunshine on Gradual Verification: a state of the art verification technique that leverages partial specifications for scalability.

Spring '23 Developed formal proofs for establishing semantic correspondence to ensure soundness between the static and dynamic verifiers.

Summer/Fall '22 Fixed optimization bugs and implemented a Property Based Testing tool for evaluating the soundness of Gradual  $C_0$ .<sup>[3][8][H1]</sup>

### Cornell University

10/2021 - 12/2022

Undergraduate Research Assistant

Advised by Dr. Adrian Sampson on the developing the Calyx Compiler Infrastructure for Accelerator Generators. Calyx's control language simplifies encoding of high-level semantics.

Spring '22 Worked on a symbolic execution tool for verifying parallelism.<sup>[2][4][5]</sup>

Winter '21/Fall '22 Fixed compiler front-end bugs and implemented *Graphicionado Graph Analytics* algorithm in Calyx.

## Publications

CURJ 2023 **Evaluating Soundness of a Gradual Verifier with Property Based Testing** [↗](#)

Jan-Paul Ramos-Dávila

In *Cornell Undergraduate Research Journal Vol. 2 No. 1*

## Presentations

POPL 2023 **Evaluating Soundness of a Gradual Verifier with Property Based Testing** [↗](#) [↗](#)

Jan-Paul Ramos-Dávila

In *Principles of Programming Languages Student Research Competition*<sup>[H1]</sup>

## Coursework Projects

**How Do Code and Mined Specs Co-evolve?** *Cornell CS 6156 Runtime Verification*

Empirical study on temporal behaviors of specifications for code verification in open-source codebases.<sup>[6]</sup>

**Eta Compiler** *Cornell CS 4120 Introduction to Compilers*

Compiler for the *Eta* programming language. Funtional design with the use of *GADTs*.<sup>[1]</sup>

**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.<sup>[1]</sup>

**Diffeq-lang** *Senior High School Project* [↗](#)

Domain Specific Language for solving differential equations.<sup>[7][H2]</sup>

## Honors

---

<b>Winner, Third Place</b> , ACM SIGPLAN Symposium POPL SRC <sup>H<sub>1</sub></sup>	2023
<b>Travel Scholarship</b> , ACM SIGPLAN Conference PLDI	2022
<b>Finalist, Mathematics</b> , Regeneron International Science and Engineering Fair <sup>H<sub>2</sub></sup>	2020 & 2021

## Technical Skills

---

**Languages:** OCaml<sup>1</sup>, Python<sup>2</sup>, Scala<sup>3</sup>, Rust<sup>4</sup>, Racket<sup>5</sup>, Java<sup>6</sup>, JavaScript<sup>7</sup>, C<sup>8</sup>, English, Español, Italiano  
**Tools:** Unix, Git, VSCode, IntelliJ IDEA, Neovim, Docker, Heroku, L<sup>A</sup>T<sub>E</sub>X