

# 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.

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

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

### 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.

Winter '21/Fall '22 Implemented *Graphicionado Graph Analytics* algorithm in Calyx.

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

### Google

07/2021 - 08/2021

Computer Science Summer Institute

Learned programming fundamentals in JavaScript directly from Google engineers and got an inside look at Google employee tools used for web development. Developed a peer-to-peer instant messaging system with the Express.js framework and Heroku hosting.<sup>[7]</sup>

## Publications

POPL 2023 (SRC) **Evaluating Soundness of a Gradual Verifier with Property Based Testing** [↗](#)  
Jan-Paul Ramos-Dávila (Extended Abstract)

CURJ Vol. 2 No. 1 **Evaluating Soundness of a Gradual Verifier with Property Based Testing** [↗](#)  
Jan-Paul Ramos-Dávila

## Coursework Projects

### RNAfoldml 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>

### JAECCE Eta Compiler CS 4120 Introduction to Compilers

Compiler for the *Eta* programming language in OCaml. Funtional design with the use of *GADTs*.<sup>[1][9]</sup>

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

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

## Honors

---

<b>Winner, Third Place</b> , ACM SIGPLAN Symposium POPL Student Research Competition	2023
<b>Travel Scholarship</b> , ACM SIGPLAN Conference PLDI	2022
<b>Finalist, Mathematics</b> , Regeneron International Science and Engineering Fair	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>, Haskell<sup>9</sup>, English, Español, Italiano

**Tools:** Unix, Git, VSCode, IntelliJ IDEA, Neovim, Docker, Heroku, L<sup>A</sup>T<sub>E</sub>X