

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 .

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.

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**

(Video [↗](#)) (Poster [↗](#))

In *Principles of Programming Languages Student Research Competition* [↗](#)

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.

Eta Compiler *Cornell CS 4120 Introduction to Compilers*

Compiler for the *Eta* programming language, developed in OCaml.

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 solving differential equations.

Honors

Winner, Third Place, ACM SIGPLAN Symposium POPL SRC 2023
Travel Scholarship, ACM SIGPLAN Conference PLDI 2022
Finalist, Mathematics, Regeneron International Science and Engineering Fair 2020 & 2021

Technical Skills

Languages: OCaml, Python, Scala, Rust, Racket, Java, JavaScript, C, English, Español, Italiano
Tools: Unix, Git, VSCode, IntelliJ IDEA, Neovim, Docker, Heroku, L^AT_EX