


Jan-Paul Vincent Ramos-Dávila

 mail@jpramos.me |  jpramos.me |  [jpramos-me](https://github.com/jpramos-me) |  [jpramos-me](https://www.linkedin.com/in/jpramos-me)

Education

Cornell University

Bachelor of Arts in Computer Science & Bachelor of Arts in Philosophy

08/2021 - 05/2025

Ithaca, NY

Experience

Amazon

Software Verification Research Intern, advised by Dr. Jonathan Aldrich

06/2023 - 08/2023

Pittsburgh, PA

- λ. Developed an algorithm to target Gradual C_0 predicates as second-order predicates, which significantly optimized runtime check assertions in our benchmarks by a factor of $O(n)$.
- λ. Participated in an industry experience workshop led by Amazonians Myles Shiroma & Korin Torrence Johnson.

Carnegie Mellon University

Software Verification Research Intern, advised by Dr. Jonathan Aldrich & Dr. Joshua Sunshine

06/2022 - Present

Pittsburgh, PA

- λ. Exploring the application of gradual verification techniques to smart contracts on the *Algorand* blockchain platform in developing Gradually Verified Teal.
- λ. Worked on formal proofs for establishing semantic correspondence to ensure soundness between the static and dynamic verifier for Gradual C_0 .
- λ. Fixed optimization bugs and implemented a Property Based Testing tool for evaluating the soundness of Gradual C_0 source code. Presented work at POPL '23 SRC.

Cornell University

Programming Languages Undergraduate Researcher, advised by Dr. Adrian Sampson

10/2021 - 12/2022

Ithaca, NY

- λ. Worked on a symbolic execution tool for verifying parallelism in Calyx.
- λ. Fixed compiler front-end bugs and implemented *Graphicionado Graph Analytics* algorithm in Calyx.

Publications

Evaluating Soundness of a Gradual Verifier with Property Based Testing Jan-Paul Ramos-Dávila In Principles of Programming Languages Student Research Competition (POPL 2023 [↗](#)) (POPL Video [↗](#)) (POPL Poster [↗](#)) & In Cornell Undergraduate Research Journal (CURJ Vol. 2 No. 1 [↗](#)).

Notable 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 mining. Supports integration with Javert and BDDMiner.

EtaC (Cornell CS 4120 Intro to Compilers [↗](#)) Compiler for the Eta programming language in OCaml with $\sim 7,200$ lines of code. Made use of Jane Street's expect testing suite for $\sim 90\%$ 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 web interface.

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

Academic Service

Student Volunteer, ACM SIGPLAN ICFP 2023

Seattle, WA

Technical Skills

Languages: OCaml, Python, Scala, Haskell, Rust, Racket, Java, JavaScript, C, English, Español, Italiano

Tools: Unix, Git, VSCode, Bash, IntelliJ IDEA, Neovim, Docker, Heroku, L^AT_EX