

Jan-Paul Vincent Ramos-Dávila

✉ mail@janpaul.pl | 🏠 https://janpaul.pl | 🌐 janpaulpl | 📄 janpaulpl | 📁 conf.researcher

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

Experience

NASA, Langley Formal Methods

Hampton, VA (R)

Research Assistant, Program Verification, Advised by Dr. Alwyn Goodloe

June 2024 - Present

- Mechanized proofs that model correct behaviors behind a Software Defined Delay-Tolerant Network's Match-Action pipeline algorithm for NASA's Interplanetary Overlay Network framework.
- Developed a Network Calculus IR formally verified in Coq (*RocqNet*). Wrote an interpreter for a subset of P4 that targets RocqNet.

Carnegie Mellon University, S3D

Pittsburgh, PA

Research Assistant, PL/Program Verification, Advised by Dr. Jonathan Aldrich & Dr. Jenna DiVincenzo

May 2022 - May 2024

- Core contributor on the early development of the Gradual Verification framework [1]. Empirically evaluated the soundness of Gradual C_0 [4] and provided formal proofs of completeness between the dynamic and static verifiers [3].
- Explored the application of Gradual Verification to smart contracts on the *Algorand* and *Ethereum* blockchain platforms and developed a prototype for Gradually Verified Teal [2].

Cornell University, CIS

Ithaca, NY

Teaching Assistant, CS 4114 Systems Programming, Taught by Dr. Ken Birman

January 2024 - May 2024

- Graded students' assignments, held weekly office hours, and ran coding workshops each week with hands-on demos building and debugging C++/Linux applications.

Teaching Assistant, CS 4/5110 Programming Languages and Logics, Taught by Dr. Adrian Sampson

August 2024 - December 2024

- Examination czar in charge of the infrastructure of midterms, graded students' assignments, and held weekly office hours.

Research Assistant, Programming Languages, Advised by Dr. Adrian Sampson

October 2021 - December 2022

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

Publications & Presentations

- [1] Ramos-Dávila, J., Goodloe, A., *Type Preserving Compilation for Formally Verified Software Defined Delay-Tolerant Networks*, In IEEE Workshop on Optimizing Interplanetary Communication Through Network Autonomy, ACM SIGPLAN/SIGLOG Certified Programs and Proofs (CPP '25, co-located with POPL '25) (*In Submission*)
- [2] DiVincenzo, J., McCormack, I., Gouni, H., Gorenburg, J., Ramos-Dávila, J., Zhang, M., Zimmerman, C., Sunshine, J., Tanter, É., Aldrich, J., *Gradual C0: Symbolic Execution for Gradual Verification*, In ACM Transactions on Programming Languages and Systems (*To Appear*)
- [3] Singh, K., Sun, H., Ramos-Dávila, J., Aldrich, J., DiVincenzo, J. *Gradual Verification of Smart Contracts*, In ACM SIGPLAN Workshop on Principles of Secure Compilation (PRISC, POPL '24 Workshop) [Preprint] [Presentation]
- [4] Ramos-Dávila, J., *Optimization of a Gradual Verifier: Lazy evaluation of Iso-recursive Predicates as Equi-recursive at Runtime*, In 51st ACM SIGPLAN Symposium on Principles of Programming Languages Student Research Competition (POPL '24 SRC), Midwest Programming Languages Summit 2023 (MWPLS '23) [Poster] [Abstract]
- [5] Ramos-Dávila, J., *Evaluating Soundness of a Gradual Verifier with Property Based Testing*, In 50th ACM SIGPLAN Symposium on Principles of Programming Languages Student Research Competition (POPL '23 SRC), Cornell Undergraduate Research Journal, 2(1), 17–27. <https://doi.org/10.37513/curj.v2i1.696> [Paper] [Presentation] [Poster]

Projects

- **A Verified IR for Calyx.** (*Cornell CS 6861 Kleene Algebra*) Verifying the correctness of parallelism in Calyx with KATs. [Paper]
- **Optimization of a Concurrent PL Model Checker.** (*Cornell CS 6120 Advanced Compilers*) Reduction of state explosion for the Harmony Concurrent Programming Language's model checker. [Repo]
- **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. [Repo]
- **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. [Repo]

Awards & Grants

next page

Travel Scholarship: Verification Mentoring Workshop @ CAV	'24
Fellow: Amazon Summer Undergraduate Research Experience (CMU)	'23
Winner, Third Place: ACM SIGPLAN POPL SRC [4]	'23
Travel Scholarship: Programming Languages Mentoring Workshop @ ACM SIGPLAN PLDI	'22
Finalist, Mathematics: Regeneron International Science and Engineering Fair	'20 & '21

Academic Service

Seoul, KR	Video Co-Chair: ACM SIGPLAN PLDI 2025	Jun. '25
Denver, CO	Video Co-Chair: ACM SIGPLAN POPL 2025	Jan. '25
Milan, IT	Virtualization Chair: ACM SIGPLAN ICFP 2024	Sep. '24
Copenhagen, DK	Virtualization Chair: ACM SIGPLAN PLDI 2024	Jun. '24
London, UK	AV Committee: ACM SIGPLAN POPL 2024	Jan. '24
Cascais, PT	Video Co-Chair: ACM SIGPLAN SPLASH 2023	Oct. '23
Seattle, WA	Student Volunteer: ACM SIGPLAN ICFP 2023	Sept. '23

Skills

Languages	OCaml, Scala, Python, Haskell, JavaScript, Java, C, Racket, Rust, English, Español, Italiano
Tools	LaTeX, Coq IDE, Agda-mode, Unix, Git, Shell, Neovim, Emacs, Docker, Heroku, HTML/CSS, Flask
PL Education	Oregon Programming Languages Summer School 2024 (Boston University) Advanced Functional Programming Summer School 2023 (Utrecht University)