Jan-Paul Vincent Ramos-Dávila

✉ mail@janpaul.pl | ⌂ https://janpaul.pl | ⊙ janpaulpl | ▣ janpaulpl

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
- Developing Coq proofs for statically proving properties of a DTN protocol to ensure end-to-end data delivery.

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.
  - *Verification*: Implemented a Property Based Testing tool for evaluating soundness of Gradual C_0 and developed formal proofs for establishing semantic correspondence between static and dynamic verification.
  - *Performance*: Significantly optimized benchmark results by using second-order logic equivalences for runtime assertions.
- Exploring the application of Gradual Verification to smart contracts on the *Algorand* and *Ethereum* blockchain platforms. Developed a prototype for Gradually Verified Teal and Gradually Verified Ethereum.

Cornell University, CIS   *Ithaca, NY*
Teaching Assistant, CS 4/5110 Programming Languages and Logics, Taught by Dr. Adrian Sampson   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 4110 Systems Programming, Taught by Dr. Ken Birman   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] 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 (**In Submission**)

[2] Singh, K., Sun, H., **Ramos-Dávila, J.**, Aldrich, J., DiVincenzo, J. *Gradual Verification of Smart Contracts*, In Workshop on Principles of Secure Compilation (PRiSC, co-located with POPL '24)

[3] **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)

[4] **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 [Pub] [Presentation] [Poster]

Projects

- **Kleene Algebra with Tests for Calyx.** (*Cornell CS 6861 Kleene Algebra*) Development of a KATs front-end for the Calyx hardware synthesis framework.
- **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.
- **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 and Honors

Fellow: Amazon Summer Undergraduate Research Experience (CMU)   *'23*
Winner, Third Place: ACM SIGPLAN POPL SRC   *'23*
Travel Scholarship: ACM SIGPLAN PLDI   *'22*
Sponsor Prize: Cornell BRH Hackathon [Course2Career]   *'21*
Finalist, Mathematics: Regeneron International Science and Engineering Fair   *'20 & '21*

Academic Service

Copenhagen, DK   **Virtualization Chair**: ACM SIGPLAN PLDI 2024   *Jun. '24*
London, UK   **AV Team**: 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
More Education   Oregon Programming Languages Summer School 2024 (*Boston University*)
Advanced Functional Programming Summer School 2023 (*Utrecht University*)