Jan-Paul Vincent Ramos-Dávila

mail@jpramos.me | jpramos.me | jpramos-me | jpramos-me

Education

Cornell University 08/2021 - 05/2025
Bachelor of Arts in Computer Science & Bachelor of Arts in Philosophy Ithaca, NY
Utrecht University 07/2023
Advanced Functional Programming in Haskell Summer School Utrecht, Netherlands

Experience

Amazon (Summer Undergraduate Research Experience @ CMU) 06/2023 - 08/2023
Software Verification Research Intern, advised by Dr. Jonathan Aldrich Pittsburgh, PA

- Developed an algorithm to target Gradual C_0 predicates as second-order predicates, which significantly optimized runtime check assertions in our benchmarks from $O(2^n)$ to $O(n)$.

Carnegie Mellon University 06/2022 - Present
Software Verification Research Intern, advised by Dr. Jonathan Aldrich & Dr. Joshua Sunshine Pittsburgh, PA

- Exploring the application of gradual verification techniques to smart contracts on the *Algorand* blockchain platform in developing Gradually Verified Teal.
- Worked on formal proofs for establishing semantic correspondence to ensure soundness between the static and dynamic verifier for Gradual C_0.
- 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 10/2021 - 12/2022
Programming Languages Undergraduate Researcher, advised by Dr. Adrian Sampson Ithaca, NY

- Worked on a symbolic execution tool for verifying parallelism in Calyx.
- Implemented *Graphicionado Graph Analytics* algorithm in Calyx as a case-study of the language and found/solved soundness bugs in the front-end.

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, LaTeX