

june wunder

Programming languages researcher — type-and-effect systems, static analysis, type inference

✉ june@junewunder.com | 🌐 junewunder.com | 🐙 [junewunder](https://github.com/junewunder) | 🌐 [june-wunder](https://www.linkedin.com/company/june-wunder) | 📍 Greater Boston (Remote Only)

SUMMARY

Programming languages PhD candidate at Boston University building type-and-effect systems for domains that normally rely on convention and runtime testing. Researching event-driven systems, reactive programs, web development, concurrency/highly multithreaded programs. Papers at POPL and ESOP with another under review; each with a Haskell or OCaml implementation. Seeking remote research or research-engineering roles starting March 2027.

EDUCATION

Boston University <i>Ph.D., Computer Science — Programming Languages</i>	Sep 2021 – Dec 2026 (expected) Boston, MA
University of Vermont <i>B.S., Computer Science</i>	2016 – 2020 Burlington, VT

PUBLICATIONS

A Type-and-Effect System for Temporal Dependency Analysis of Render-based Reactive Programs

june wunder, Ankush Das, Marco Gaboardi

Under double-blind review, 2026. [arXiv:2607.27074](https://arxiv.org/abs/2607.27074) | [code](#)

Pipelines and Beyond: Graph Types for ADTs with Futures

Francis Rinaldi, **june wunder**, Arthur Azevedo de Amorim, Stefan K. Muller

POPL 2024. DOI | [arXiv:2311.06984](https://arxiv.org/abs/2311.06984) | [artifact](#)

Bunched Fuzz: Sensitivity for Vector Metrics

june wunder, Arthur Azevedo de Amorim, Patrick Baillot, Marco Gaboardi

ESOP 2023. DOI | [arXiv:2202.01901](https://arxiv.org/abs/2202.01901)

RESEARCH & TEACHING EXPERIENCE

Boston University <i>Ph.D. Candidate, Programming Languages Research (Research Assistant, 2020–21)</i>	Sep 2020 – Present Boston, MA
--	----------------------------------

- Advised by Marco Gaboardi. Worked closely with Arthur Azevedo de Amorim and Ankush Das.
- **Willow**: designed a core calculus, time-aware operational semantics, and type-and-effect system for React-style reactive programming, then implemented a prototype checker with automatic effect inference in Haskell. Willow tracks the timing of state changes as effects, forming temporal dependency graphs that statically detect render cascades and inter-render loops.
- **GML / Graph Types**: built the type inference engine in OCaml for a system that statically captures every possible execution graph of a parallel program with futures, catching multithreading bugs such as deadlock before runtime.
- **Bunched Fuzz**: extended linear-logic-based sensitivity typing to arbitrary L^p semimetric spaces.
- **Teaching Fellow** for Boston University's undergraduate programming languages course CS320. Fall 2021 and Fall 2024.

University of Vermont — PLAID Lab <i>Undergraduate Research Assistant for David Darais and Joe Near</i>	Jan 2019 – May 2020 Burlington, VT
---	---------------------------------------

- Learned Agda programming language and practiced proof engineering skills.
- Investigated formalization of Fuzz language for differential privacy.
- Explored possible abstractions for language verification in Agda.

INDUSTRY EXPERIENCE

Internships **DPella** — production Haskell, differential-privacy analytics (2021) | **Renewable Energy Solutions** — React (2020) | **Northwestern Mutual** — React (2019) | **NextCapital** — Ruby (2018) | **BeaconVT** — Express.js (2017)

SKILLS

Languages Haskell, OCaml, TypeScript / JavaScript, Rust, Rocq (Coq), Agda, LaTeX
Frameworks React, Svelte, Next.js