Bhargav Kulkarni

bhargavkishork@gmail.com | github.com/bhargavkulk | linkedin.com/bhargavkulk | bhargavkulk.github.io

Education

University of Utah UT
PhD *2023 — Present*
- Advised by Prof. Pavel Panchekha
- Cumulative GPA: 3.8/4.0; Relevant Courses: PL, verification, compilers, computer architecture
- TA for Compilers, Computer Organization

BITS Pilani India
Bachelors in Computer Science *2019 — 2023*
- Cumulative GPA: 8.9/10.0; Merit scholarship holder
- TA for OS, compilers, networks, architecture

Papers

Mixing Condition Numbers and Oracles for Accurate Floating-point Debugging IEEE ARITH'25
Bhargav Kulkarni, Pavel Panchekha
- This paper introduces **ExplaniFloat**, combining double-double arithmetic and condition numbers for **faster, more accurate numeric debugging**.
- It achieves **80.0%** precision and **96.1%** recall on **546** benchmarks, **more accurate** than double-double oracles and **far faster** than arbitrary-precision methods.

Research Experience

Research Assistant 2023 — Present
University of Utah *Salt Lake City, UT*
- Currently working on building a verified optimizer for the **Skia** vector graphics engine that drives the rendering of the **Chrome** web browser.
  - To verify optimizing rewrites, formalized the semantics of Skia's operations using the **Lean** interactive theorem prover
- Previously worked on adapting techniques from floating-point static analysis to build an accurate and performant floating-point debugging tool.

Research Intern @ NASA Langley Formal Methods Group 2024
NASA *Langley, VA*
- Worked on generating **proof certificates** for the **PVS** automated theorem prover to verify **Herbie's** (a floating-point superoptimizer) **accuracy-aware optimizations**

Research Intern Jun 2020 — May 2023
CMI *India*
- Worked with Prof. SP Suresh and Prof. Anup Basil Mathew @ Chennai Mathematical Institute
- Formalized basic DFA/NFA constructions in Coq/Rocq
- Some initial work adapted into undergraduate Discrete Structures course

Skills and Projects

- **General Programming**: Python, Racket, Java
- **Systems Programming**: C/C++, Bash, Rust
- **Hardware**: Verilog, x86

- **Trinity Game Engine**: A game engine and byte code VM for scripting. [source]
- **Logic in Coq**: Classical propositional logic and natural deduction in Coq/Rocq. [source]
- **CheemScheme**: Scheme dialect in C++ with tail recursion and error reporting. [source]