

Bhargav Kulkarni

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

Education

University of Utah

2023 — Present

PhD

- 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

2019 — 2023

Bachelors in Computer Science

- 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 prior debugging research and **far faster** than arbitrary-precision methods.

Research Experience

Research Assistant

University of Utah

2023 — Present

- Currently building a verified optimizer for the **Skia** vector graphics engine that powers **Chrome** rendering.
 - Formalized Skia's semantics in the **Lean** theorem prover to verify optimizing rewrites.
- Previously adapted floating-point static analysis techniques to build an accurate floating-point debugger.

Research Intern

NASA Langley Formal Methods Group

2024

- Worked on generating **proof certificates** for the **PVS** automated theorem prover to verify **Herbie's** (a floating-point superoptimizer) **accuracy-aware optimizations**

Summer Research

CMI

Jun 2020 — May 2023

- Worked with Prof. SP Suresh (CMI) and Prof. Anup Basil Mathew (BITS Pilani)
- 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. [\[source\]](#)