Dipesh Kafle

Email | NUS Email | GitHub | LinkedIn | Website

Software engineer and PhD student focused on programming languages and formal methods, with a strong interest in systems programming and distributed systems.

Education

National Institute of Technology Tiruchirappalli 2019 – 2023
B.Tech in Computer Science and Engineering CGPA: 8.84/10

- Studied algorithms and data structures, discrete mathematics, computer architecture, operating systems, computer networks, databases, theory of computation, and compilers.

National University of Singapore 2025 – Present
Ph.D. in Computer Science Singapore

- Coursework: CS6223 Advanced Topics in Software Testing, CS5223 Distributed Systems, CS5232 Formal Specification and Design Techniques, CS5469 Logic in Computer Science, CS6217 Topics in Programming Languages and Software Engineering.
- Teaching Assistant, CS3213 Foundations of Software Engineering.

Publications

- **Certified Program Synthesis with a Multi-Modal Verifier.**
  ASE, 2026. Joint first author. Paper
- **Velvet: A Foundational Multi-Modal Verifier for Imperative Programs in Lean.**
  CAV, 2026 (Distinguished Paper Award). Paper
- **A Mechanically Verified Garbage Collector for OCaml.**
  Journal of Automated Reasoning, 2025. Paper

Work Experience

Uber 2023 – 2025
Software Engineer I -> Software Engineer II (Promoted 2025) Bengaluru, India

- Worked on backend systems for Uber's AI Solutions initiative, including the initial MVP and later scaling efforts.
- Worked on improving reliability, security, and observability across HITL orchestration services.
- Participated in on-call rotations, handled production incidents, coordinated rollouts and migrations.
- Active code contributor and reviewer; contributed to team documentation and internal guides for testing and debugging backend systems.

IIT Madras 2022 – 2024
Research Intern Remote

- Worked with Dr. KC Sivaramakrishnan and Dr. Kartik Nagar alongside a PhD student on a project that aimed to verify an OCaml style garbage collector with F*/Low*.
- Helped with the integration of the extracted verified code with the OCaml bytecode interpreter, ran real-world OCaml programs and ran benchmarks to analyze performance. Also wrote a next-fit allocator in Rust to work with the generated verified stop-the-world mark and sweep code.

CDAC Bangalore 2023
Research Intern Remote

- Developed a GCC plugin that transformed a familiar code snippet to highly optimized subroutines and another one that tuned loop unrolling heuristics based on linear regression model. Additonally, suggested potential ARM specific optimizations for future exploration.

Tarides 2023
Software Engineering Intern Remote

- Worked on developing Par_incr, a library for incremental computation with support for freshly introduced parallelism constructs in OCaml.

Technical Projects

Velvet

- A Dafny-style multi-modal verifier for imperative programs embedded in the **Lean 4** proof assistant.
- Combines SMT-based automated verification (Z3, CVC5) with Lean's interactive proof mode, enabling programs to be specified, verified, and executed within a unified environment.
- Supports separate reasoning for functional correctness and termination, non-determinism, and direct integration with Lean's `mathlib`.

Par_incr

- A library for incremental computation with support for parallelism in **OCaml**. Other similar libraries lack parallelism constructs. The work is based on the paper Efficient Parallel Self-Adjusting Computation. [Slides]
- Wrote the library from scratch and thoroughly tested it.
- Identified performance bottlenecks through profiling and applied various optimization techniques in OCaml.
- Wrote benchmarks, compared the performance with other similar libraries, and achieved similar if not better performance on average.

Code Character

- A strategy-based programming game where you control troops in a turn-based game with the code you write in one of the multiple programming languages (C++, Python, Java) available in the game.
- Worked on the implementation of the simulator (**C++**)
- Worked on the game driver (**Rust**), including process orchestration, inter-process communication, and concurrent execution.

Talks and Writings

Understanding Memory Management

- Slides, Video

Personal Blog

- What is a Fixed Point Combinator?
- Non Local Jumps with setjmp and longjmp

Skills

Programming: C, C++, Rust, OCaml, Lean4, Java, TypeScript, Python
Areas: Programming Languages, Formal Verification, Systems Programming, Back-End Development, Databases