Omkar Vijaykumar Tuppe

Mumbai, India | omkarvt@gmail.com | +919665836934
LinkedIn: https://www.linkedin.com/in/omkar-tuppe-845a29106
GitHub: https://github.com/0Ccipher

SUMMARY

PhD researcher at Computer Science & Engineering at IIT Bombay, specializing in program analysis, formal verification, and concurrent software testing. Built LLVM-IR-based prototype tools, **GPUMC, TranSER, and ConsChecker**, by extending existing infrastructures to implement and evaluate novel concurrency verification algorithms targeting GPU kernels, transactional systems, and relaxed memory models.
Published research in **POPL, CAV, TACAS, and ATVA** spans programming languages, model checking, and static/dynamic analysis. Passionate about applying formal methods, compiler optimizations, and AI techniques to improve software correctness and reliability. Actively seeking opportunities to apply and expand my skills.
Open to roles in formal verification, compilers, program analysis, or software engineering where I can build, enhance, and apply my expertise.

EDUCATION

Indian Institute of Technology Bombay — PhD in Computer Science & Engineering
(2019 – Present, Expected Graduation: Apr 2026) | CPI: 8.15/10
Solapur University — Bachelor of Engineering in Computer Science & Engineering
(July 2014 – Mar 2018) | CPI: 9.8/10

RESEARCH & DEVELOPMENT *(Jan 2021 - Present)*

- **MP+SMCheck: Model Checker for hybrid concurrent programs** *(Sep 2024 - Present)*
  - Designed a formal verification algorithm for concurrent programs combining message passing and shared memory; currently evaluating the algorithm's performance on Go-inspired benchmarks.
- **GPUMC**: **Stateless Model Checker for GPU memory models** *(Jan 2024 - Oct 2024)*
  - Engineered an LLVM-based tool to verify NVPTX and Vulkan-style GPU programs under the SCR11 memory model, exploring the execution state space by customizing the scheduler.
  - Achieved **100x speedup** and **10x lower memory usage** over baselines.
  - Integrated heterogeneous race detection and repair techniques for GPU kernels.
- **TranSER**: **Dynamic Partial Order Reduction for Transactional Programs** *(Dec 2023 - Apr 2024)*
  - Introduced a stateless model checking algorithm for serializable transactional programs used in distributed storage and database systems.
  - Implemented the algorithm in over 3000 lines of C++ code, using the LLVM interpreter to explore interleavings systematically.
  - Validated on **OLTP Benchmarks**, outperforming baselines by **100x in bug detection**.
- **DynamicDyckReachability**: **On-the-fly static analysis via Dyck Reachability** *(Sep 2022- Jun 2023)*
  - Developed a dynamic Dyck-reachability graph algorithm for on-the-fly static analysis.
  - Evaluated on the **DaCapo benchmarks** (alias analysis) and the **SPECjvm2008 benchmark suite** (data-dependence analysis), achieving up to **1000x speedup** over baseline methods.
  - The algorithm efficiently handles code additions and deletions, with negligible update time, making it well-suited for continuous analysis and integration within an IDE.
- **ConsChecker**: **Model Checker for Causal Consistency** *(Jun 2022 - Oct 2022)*
  - Built an LLVM-IR-based model checker to verify **causal consistency** in concurrent programs.
  - Evaluated on the SV-COMP verification competition benchmark suite to demonstrate the effectiveness and performance.
- **ARMv8-A BMC: Verification of ARMv8-A concurrency** *(Dec 2021 - Jun 2022)*
  - Collaborated on developing a bounded context-switching model checker for ARMv8-A assembly using sequentialization-based symbolic execution.

- **TranChecker**: **Model Checker of transactional programs** *(Jan 2021 - Feb 2022)*
    - Engineered verification algorithm for transactional programs operating under weak isolation guarantees (Read Committed, Read Atomic, Causal Convergence).
    - Implemented LLVM-IR-level analysis via a custom LLVM interpreter in C++, enabling systematic exploration of valid concurrent executions.

TECHNICAL COMPETENCIES

- **Expertise:** Formal Verification, Memory Models, Program Analysis, Software Engineering
- **Programming Languages:** C, C++, Python, HTML, CSS
- **Development Tools:** Git, CMake, GDB, Valgrind
- **Research Tools:** LLVM-lli, JavaPathFinder, LaTeX, Coq, Z3, LLVM
- **Platforms**: Linux, ARMv8-A, PTX

INTERNSHIPS

Research Intern, Uppsala University, Sweden *(Apr–June 2023, May–July 2022)*

- Built a sleep-set-based stateless model checking algorithm for C concurrent programs under sequential consistency using the LLVM Interpreter to prune redundant interleavings efficiently.
- Contributed to the development of an algorithm for the verification of concurrent programs using the partial order reduction technique.

PUBLICATIONS

- GPUMC: A Stateless Model Checker for GPU Weak Memory Concurrency, *CAV 2025* - with Soham Chakraborty, S Krishna, and Andreas Pavlogiannis
- Dynamic Partial Order Reduction for Transactional Programs on Serializable Platforms, *ATVA 2024* - with Parosh Abdulla, Ashutosh Gupta, and S Krishna
- On-The-Fly Static Analysis via Dynamic Bidirected Dyck Reachability, *POPL 2024* - with S Krishna, Aniket Lal, and Andreas Pavlogiannis
- Optimal Stateless Model Checking for Causal Consistency, *TACAS 2023* - with Parosh Abdulla, Mohamed Faouzi Atig, S Krishna, and Ashutosh Gupta

LEADERSHIP & TEACHING

- **Student Mentor**: Guided 12+ junior researchers in the IITB Research Scholar Companion program during COVID-19 *(June 2020 – May 2021)*.
- **Conference Speaker:** Presented research at CAV 2025, ARCS 2025, RISC 2025, ATVA 2024, TCS Research Expo 2024, FSTTCS-RHPL 2023, TACAS 2023
- **Teaching Assistant:** Data Structures & Algorithms, Automated Reasoning, Logic in CS, Analysis of Concurrent Programs, Automata Theory