Sankha Narayan Guria

https://sankhs.com

Email: sankha@cs.umd.edu
Mobile: +1 (240) 302 - 3976

Education

- **University of Maryland** — College Park, MD
  Ph.D. in Computer Science · Advisor: *Jeff Foster & David Van Horn* · GPA: 3.75/4.0 — *Aug 2017 – Present*
- **Indian Institute of Technology Jodhpur** — Jodhpur, India
  B.Tech. in System Science · GPA: 8.26/10.0 — *July 2011 – May 2015*

Publications

[1] S. N. Guria, J. S. Foster, and D. Van Horn. "RbSyn: Type- and Effect-Guided Program Synthesis". In: *42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI 2021)*.

[2] M. Kazerounian, S. N. Guria, N. Vazou, J. S. Foster, and D. Van Horn. "Type-Level Computations for Ruby Libraries". In: *40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2019)*.

[3] M. Keil, S. N. Guria, A. Schlegel, M. Geffken, and P. Thiemann. "Transparent Object Proxies in JavaScript". In: *29th European Conference on Object-Oriented Programming (ECOOP 2015)*.

Awards & Achievements

- Dean's Fellowship, University of Maryland · $5000 — *2017 – 2018*
- Summer Dean's Fellowship, University of Maryland · $5000 — *Summer 2018*
- NSF Travel Award, Summer School on Formal Techniques, SRI · $700 — *May 2018*

Experience

- **University of Maryland** — College Park, MD
  Graduate Research Assistant — *June 2018 - Present*
  - Designing a type and effect driven synthesis engine for Ruby programs, using tests as specification [1].
  - Developed practical type systems on RDL using type-level computations that scales to real web applications [2].
- **Facebook** — Menlo Park, CA
  Software Engineering Intern — *May 2021 - Aug 2021*
  - Designed & implemented an effect encapsulation for the Hack language to allow users to define & enforce custom coeffects.
- **Synthetic Minds** — San Francisco, CA
  Research Intern · Advisor: Prof. Rastislav Bodik — *May 2019 - Aug 2019*
  - Designed & implemented a symbolic execution engine for Solidity, capable of running large smart contracts like Augur. It supported solver aided queries like verification, angelic execution and synthesis.
- **BrowserStack** — Mumbai, India
  Software Engineer — *June 2015 - June 2017*
  - One of the two primary developers to build and release *App Live* - the interactive cloud based mobile app testing on real devices product from scratch in 5 months.
  - Scaled the *Automate* product to more than 300,000 sessions/day (~4x growth), changed the engineering culture of the team to rely on automated test suites to ship faster at 99.5% stability.
  - Established organization-wide instrumentation for the cloud infrastructure, built real-time message relay service, optimized real device cloud to achieve ~2x faster user perceived session start-time.
- **University of Freiburg** — Freiburg, Germany
  Research Intern · Advisor: Prof. Peter Thiemann — *May 2014 - July 2014*
  - Developed JavaScript language semantics with transparent proxies against the equality operator and defined an object capability model for security related use cases in contract systems [3].
  - Propositions were implemented on SpiderMonkey VM's interpreter and baseline JIT and proved to run with real-world benchmarks without any performance regressions.
- **Mozilla** — Remote
  Open-source Contributor — *June 2012 - July 2014*
  - Primarily contributed to SpiderMonkey - the JavaScript engine. Shipped new ECMAScript 6 features like Array, Map & Set iteration methods, String#repeat, Object.setPrototypeOf, etc.
  - Implemented a number of JIT optimizations, async I/O in critical paths to reduce browser jank.
  - Proposed and implemented a deterministic algorithm to analyze the browsing and form submission behavior of the user to detect search forms as a part of *Google Summer of Code 2013*.

Projects

- **Automated Verification of Database Model Validations**
  Advisor: Prof. Jeff Foster *Oct 2017 - Dec 2017*
  - Developed a framework to compile database model schema and related methods in Ruby on Rails applications to an equivalent Rosette program, emulating basic database queries with Rosette structs.
  - Verified database validation predicates hold statically, by discharging them as SMT queries to Z3.
- **Specializing JavaScript Programs**
  Advisor: Prof. Peter Thiemann *Feb 2014 - May 2014*
  - Studied program specialization techniques for JavaScript interpreters. Results were added to a JavaScript interpreter written in JS, to type specialize operations by gathering type feedback to make them faster.

Technical Skills

- **Languages:** Ruby, JavaScript, Rust, Python, C, C++, Bash, Racket
- **SAT/SMT Solvers:** Z3
- **Research Tools:** Coq, Rosette

Teaching Experience

- **CMSC430:** Design and Implementation of Programming Languages, *Teaching Assistant* *Spring 2020, Fall 2019*
- **CMSC433:** Programming Language Technologies and Paradigms, *Teaching Assistant* *Spring 2018*
- **CMSC216:** Introduction to Computer Systems, *Teaching Assistant* *Fall 2017*

Service

- **Artifact Evaluation Committee:** Programming Language Design and Implementation (PLDI) Conference *2021*
- **Organizer:** UMD's Programming Languages Reading Group *Spring 2020, Fall 2019*
- **Artifact Evaluation Committee:** Programming Language Design and Implementation (PLDI) Conference *2020*
- **Application Reviewer:** Admissions Committee, Department of Computer Science, UMD *Fall 2020*
- **Artifact Evaluation Committee:** Principles of Programming Languages (POPL) Conference *2020*
- **Mentor:** Tech+Research track of Technica, world's largest women & non-binary hackathon *2019, 2018*