

# Shaurya Gomber

[sgomber.github.io](https://sgomber.github.io) | [sgomber31@gmail.com](mailto:sgomber31@gmail.com) | [linkedin.com/in/sgomber](https://linkedin.com/in/sgomber) | [github.com/sgomber](https://github.com/sgomber)

## EDUCATION

### University of Illinois Urbana-Champaign

*PhD in Computer Science; GPA: 4.0/4.0*

*Advisor: Prof. Gagandeep Singh; Research Areas: Program Analysis & Automated Reasoning*

Illinois, USA

*Aug 2024 - Present*

### University of Illinois Urbana-Champaign

*Masters of Science (thesis-track), Computer Science; GPA: 4.0/4.0*

*Advisor: Prof. Gagandeep Singh; Thesis: Neural Abstract Interpretation*

Illinois, USA

*Aug 2022 - May 2024*

### Indian Institute of Technology Guwahati

*B. Tech, Computer Science and Engineering; GPA: 9.66/10 (Insti. Rank: 3)*

Assam, India

*July 2016 - June 2020*

## RESEARCH INTEREST

My current research focuses on developing efficient and precise techniques for program analysis by leveraging the synergy between learning-based and symbolic methods. I also have a keen interest in automated reasoning tools, such as SAT/SMT solvers, and the analysis of neuro-symbolic systems.

## PUBLICATIONS

S=IN SUBMISSION, P=POSTER

[S.1] Synergistic Synthesis of Ranking Function and Invariants for Termination Analysis [In Submission]

*Yasmin Sarita, Avaljot Singh, Shaurya Gomber, Mahesh Viswanathan, Gagandeep Singh*

[Arxiv](#)

[P.1] Neural Abstract Interpretation

*Shaurya Gomber*

SRC @ PLDI'24

[Abstract](#), [Poster](#)

## WORK EXPERIENCE

### Amazon Web Services (AWS), Automated Reasoning Group

*Applied Scientist Intern*

Santa Clara, California, USA

*June 2023 - Aug 2023*

- Worked on the tool [Zelkova](#), that reasons about AWS access control policies by encoding them and verification conditions as SMT queries.
- Some semantics, such as conditions relying on type-casting (e.g., allowing access if a string S in the request context is numerically less than 42), are challenging to encode in SMT. We developed efficient encodings using [SMT0](#) (SMT with Oracles) to handle such *type-casting semantics*.
- This *solved*  $\sim 30k$  such production queries (unsolved before) with avg. query solving time 1 min.
- Also contributed to [CVC5](#)'s [SMT0](#) solver by identifying/fixing bugs and improving the oracles I/O interface.
- Tech Stack: Java, Scala, Python, SMT Solvers (Z3, CVC5, etc.)

### D.E. Shaw & Co.

*Software Engineering Intern & Software Engineer II*

*May 2019 - July 2019 & June 2020 - June 2022*

Hyderabad, India

- Enhanced the firm's proprietary low-latency electronic trading system, which processed terabytes of trading data daily, with features like on-demand arbitrary data computations to optimize trader workflows.
- Reviewed major projects, contributed to design discussions for core trading system components, and mentored two new SDE-1s.
- As an intern, I implemented a *type-safe low-latency API* in Java to read and write on the firm's database, achieving a 60x run-time improvement in production-critical scripts, leading to a Pre-Placement Offer.
- Tech Stack: Java & C++ (backend), React (frontend). Misc: Git, Python, Bash, Grafana, Numpy, Matplotlib.

## TEACHING EXPERIENCE

Teaching Assistant, CS421 Programming Languages & Compilers, UIUC [🔗](#)

Spring '23, Fall '23

Teaching Assistant, CS225 Data Structures & Algorithms, UIUC [🔗](#)

Fall '22

## TALKS

|  |                                    |
|--|------------------------------------|
| <b>Neural Abstract Interpretation</b><br><i>Formal Methods Seminar, Fall 2024, UIUC</i>                                      | Apr 2024<br><a href="#">Slides</a> |
| <b>Verification &amp; Certified Training of PINNs</b><br><i>CS598 Scientific Machine Learning, Fall 2023, UIUC</i>           | Nov 2023<br><a href="#">Slides</a> |
| <b>Satisfiability and Synthesis Modulo Oracles</b><br><i>Formal Methods Seminar, Fall 2023, UIUC</i>                         | Nov 2023<br><a href="#">Slides</a> |
| <b>Neural Approximations of Abstract Transformers</b><br><i>CS477 Formal Software Development Methods, Spring 2023, UIUC</i> | May 2023<br><a href="#">Slides</a> |
| <b>Synthesizing Abstract Transformers</b><br><i>Formal Methods Seminar, Spring 2023, UIUC</i>                                | Mar 2023<br><a href="#">Slides</a> |
| <b>Monotonic Neural Networks</b><br><i>CS521 Trustworthy AI Systems, Fall 2022, UIUC</i>                                     | Nov 2022<br><a href="#">Slides</a> |

## ACADEMIC SERVICE

- **Artifact Evaluation Committee:** PLDI'24

## AWARDS

|   |           |
|---|-----------|
| <b>Richard T. Cheng Endowed Fellowship</b> <a href="#">↗</a>      | 2024-2025 |
| <b>Institute Merit Scholarship IIT Guwahati</b> <a href="#">↗</a> | 2018-2019 |

## SELECTED PROJECTS

|   |                     |
|---|---------------------|
| <b>Verification and Certified Training of PINNs</b>   <i>Python, Tensorflow, Numpy</i>   <a href="#">Code</a>   | Oct 2023 – Nov 2023 |
| <ul style="list-style-type: none"><li>• PINNs (Physics-Informed Neural Networks) solve partial differential equations by embedding governing equations as constraints, with residual error representing prediction discrepancies.</li><li>• Computed worst-case residual error bounds using Interval Bounds Propagation (IBP) and used Projected Gradient Descent (PGD) and Certified Training with the computed error bounds to train the PINNs.</li><li>• Demonstrated that PGD and Certified Training yield PINNs with provably lower worst-case residual errors compared to vanilla training.</li></ul> |                     |
| <b>Monotonic Neural Networks</b>   <i>Python, Keras, Tensorflow, Numpy</i>   <a href="#">Code</a>   | Oct 2022 – Nov 2022 |
| <ul style="list-style-type: none"><li>• Monotonic Neural Networks are ones whose outputs vary monotonically with respect to the input features.</li><li>• Implemented a gradients-based loss method that enforces monotonicity while training neural networks.</li><li>• Demonstrated its effectiveness on networks that predict house rents (monotonically increasing with number of rooms) and life expectancy (monotonically decreasing with the cholesterol levels).</li></ul>  |                     |
| <b>Efficient SAT Solver</b>   <i>Python</i>   <a href="#">Code</a>  | Mar 2020 – May 2020 |
| <ul style="list-style-type: none"><li>• Implemented the <i>CDCL (Conflict Driven Clause Learning)</i> SAT algorithm, with state-of-the-art optimizations like 2-watched literals, Decision Heuristics (VSIDS, DLIS) and Restart Heuristics (Geometric, Luby).</li><li>• Devised new heuristics based on intelligent data structures (Priority Queues) and innovative restart strategies.</li></ul>  |                     |
| <b>Traffic Light Model Checker</b>   <i>NuSMV Model Checker</i>   <a href="#">Code</a>  | Oct 2019 – Nov 2019 |
| <ul style="list-style-type: none"><li>• Used the <a href="#">NuSMV Model Checker</a> to simulate the traffic lights model at a road junction.</li><li>• Verified its correctness by checking temporal-logic safety constraints in Linear Temporal Logic (LTL) like two lights can not be on simultaneously.</li></ul>   |                     |
| <b>Compiler Construction</b>   <i>C++, Flex, Bison, SPIM simulator</i>   <a href="#">Code</a>   | Jan 2019 – Apr 2019 |
| <ul style="list-style-type: none"><li>• Developed a <i>compiler for a C-like language</i> from scratch by implementing all stages of compilation.</li><li>• The stages include: lexical analysis, parsing, intermediate code generation, and target MIPS code generation.</li><li>• It supported: Function calls, expressions (relational, arithmetic and logical), if-else, switch, for and while loops.</li></ul>   |                     |

## RELEVANT COURSEWORK

---

- **Formal Methods & PL:** Logic in Computer Science, Programming Languages Design, Formal Methods for System Verification, Basics of Automated Reasoning, Program Verification, Trustworthy AI Systems, Formal Software Development Methods, Formal Languages and Automata Theory, Theory of Computation, Compilers (with lab)
- **Mathematics:** Discrete Mathematics, Linear Algebra, Real Analysis, Multi-variate calculus, Introduction to Differential Equations, Convex Optimization, Probability Theory and Random Processes
- **Misc. Computer Science:** Operating Systems (with lab), Networks (with lab), Algorithms, Databases (with lab), Computer Architecture, Digital Design, Parallel Computer Architecture, Computer Vision, Applied Machine Learning, Scientific Machine Learning

## MENTORSHIP & SERVICE

---

- **Mentored CSE freshers** under the Mentor-Mentee program of the [SAATHI Counselling Club](#) of IIT Guwahati.
- **Placement Lectures Coordinator, IITG (Fall 2019):** Organized the lectures (content, schedule etc.) and taught Data Structures & Algorithms to the candidates appearing for placements.
- **Treasurer, CSEA (2019-20):** Served as the treasurer of Computer Science and Engineering Association, IIT Guwahati and was responsible for the fund management and allocation for the CSEA events.

## ACHIEVEMENTS

---

- **Microsoft Code.Fun.Do 2019:** One of 10 national finalists (300+ teams) Topic: Blockchain-based Voting System
- **Inter IIT Tech Meet 2018:** Represented IIT Guwahati in the coding hackathon event at IIT Bombay
- **ACM ICPC 2018:** Represented IIT Guwahati in the India regionals held at Amritapuri, Kerala
- **KVPY 2015:** Secured All India Rank 178 among 1.5 million candidates (top 0.01%) in a Science Aptitude exam conducted by IISc Bangalore.
- **IIT JEE Mains 2016:** All India Rank 2323 among 1.5 million candidates (top 0.15%) in the preliminary phase of engineering entrance exam.
- **IIT JEE Advanced 2016:** All India Rank 902 among 1.5 million candidates (top 0.06%) in the advanced phase of engineering entrance exam.