Maruth Goyal

(512)-971-6794 | maruth@cs.utexas.edu

Education

The University of Texas at Austin — Austin, TX

Bachelor of Science in Computer Science, Turing Scholars Honors GPA: 3.88/4.00 — *Aug. 2018 – May 2022*

Bachelor of Science in Mathematics GPA: 3.81/4.00 — *Aug. 2018 – May 2022*

Cumulative GPA: 3.82/4.00

Coursework: (i) Cryptography† (ii) Distributed Systems† (iii) Foundations of Machine Learning and Data Science† (iv) Automated Logical Reasoning† (v) Theory of Computation† (vi) Quantum Information Science⋆ (vii) Virtualization (viii) Operating Systems⋆ (ix) Computer Architecture⋆ (x) Theory of Computation (xi) Algorithms and Complexity (xii) Data Structures⋆ (xiii) Discrete Mathematics⋆ (xiv) Topology (xv) Category Theory (xvi) Algebraic Structures (xvii) Linear Algebra⋆ (xviii) Differential Equations⋆ (xix) Vector Calculus⋆ (xx) Probability
† = Graduate Level, ⋆ = Honors

Awards

Undergraduate Research Fellowship, UT Austin	2020
Albert A. Bennett Linear Algebra Competition, 2nd place	2019
Ajit Ramchandani Endowed Presidential Scholarship	2018
University Honors	Fall 2018-2020, Spring 2019-2020

Experience

Undergraduate Research Assistant — September 2021 – Present

The University of Texas at Austin — *Austin, TX*

- Working with Prof. Aditya Akella on precise data-driven query planning.
- Extending prior work on picking query physical plan on-the-fly based on properties of intermediate data.
- Developing an optimizer to pick best strategy to picking plan on-the-fly by looking at strategy which examines properties most relevant to data and query
- Developing abstractions to incorporate code-generation while still planning on-the-fly
- Developing a program synthesizer to synthesize better strategies by examining past queries, data, and plan choices.

Software Engineering Intern — May 2021 – July 2021

Stripe, Inc. — *San Fransisco, CA*

- Worked on the Financial Automation team. Developed an app using Apache Spark to classify transactions based on a set of rules provided by accounting.
- Optimized implementation to get a 3x speedup by rewriting filters as joins, judicious caching, and other tricks to optimize Spark SQL query plan.
- Additionally delivered a dashboard that visualized data-quality metrics for transaction data ingested by the Financial Automation
- Dashboard helps engineers and management detect and diagnose any issues with data quality as soon as possible

Undergraduate Research Assistant — May 2020 – May 2021

The University of Texas at Austin — *Austin, TX*

- Lead team of 5 people with Prof. Isil Dillig on optimization of smart contracts on the blockchain; automatically refactoring data layout to minimize "gas" usage.
- Developed procedure to efficiently generate search space of candidate programs, novel synthesis algorithm, and static analyses for aggressive reduction of the search space.
- Developed a novel synthesis algorithm that exploited symmetry to drastically reduce the search space.
- 15k+ LoC in Java, Kotlin, Solidity, Bash. Used Gradle for building, CheckStyle, SpotBugs, Docker for ensuring standards with collaborators, and JUnit 5 with a CI pipeline for correctness.
- Submitted to **OOPSLA 2022.**

Undergraduate Teaching Assistant — Aug 2019 – Dec 2019, Aug 2020 – Dec 2020

The University of Texas at Austin — *Austin, TX*

- Led weekly discussion sections, graded problem sets, and exams for CS311H: Honors Discrete Mathematics with Prof. Isil Dillig.
- Developed a curriculum and series of assignments for an introduction to machine-checked mathematical proofs in the Coq language.
- Developed a programming assignment for students with the goal of cracking the Vigenere cipher as an application of math learned in class to Computer Science.

- Assisted in designing exam questions

Undergraduate Research Assistant Jan 2019 – Aug 2019

The University of Texas at Austin *Austin, TX*

- Worked on using machine learning for static type inference for gradually-typed languages like JavaScript with Prof. Isil Dillig and Prof. Greg Durrett.
- Assisted design of Graph Neural Network model architecture. Identified major sources of error, and helped optimize model to increase performance.
- Implemented internal translations from TypeScript to a Scala DSL, and helped setup and run experiments.
- Published at **ICLR 2020**

Publications

- ***Maruth Goyal**, James Dong, Yanju Chen, Yuepeng Wang, Yu Feng, Isil Dillig*, Synthesis-Powered Optimization of Smart Contracts via Data Type Refactoring **(in submission)**
- *Jiayi Wei, **Maruth Goyal**, Greg Durrett, Isil Dillig*, LambdaNet: Probabilistic type inference using graph neural networks. In **International Conference on Learning Representations (2020)**

Projects

Machine Unlearning Spring 2021

- Surveyed, and setup experiments to evaluate the current state of the art in Machine Unlearning.
- Provided first experimental evaluations for some algorithms, and attempted to compare them to theoretical guarantees.
- Used both toy, and real world data in order to evaluate and understand the settings in which different algorithms worked.
- Found certain algorithms to not be as useful in practice due combinations of poor performance, or extreme time and memory requirements.
- Implemented a general framework-agnostic set of abstractions and interfaces to make any ML model amenable to machine unlearning.

Survey on Karp-Lipton Style Theorems Spring 2020

- Wrote a survey on the use of theorems in the style of the Karp-Lipton theorem , connecting lower bounds for non-uniform models of computation with bounds for uniform models
- Explored the original result of Karp and Lipton, the super-polynomial lower bound on Σ_2 by Ravi Kannan, and the lower bound for PP by Vinodachandran, among others.
- Further explored Quantum equivalents of Karp-Lipton style theorems with Aaronson's result showing $\mathsf{PP} \subseteq \mathsf{BQP/poly} \implies \mathsf{CH} = \mathsf{MA}$
- Also investigated recent result of Chen, McKay, Murray, and Williams showing an equivalence between Karp-Lipton style collapses and superpolynomial lower bounds for P^{NP}
- Finally, explored these collapses in the context of the Algebraization proof barrier introduced by Aaronson and Wigderson.

Transactional Memory for OS | *C++, x86* Fall 2019

- Implemented support for Intel TSX Hardware Transactional Memory at kernel level.
- Allowed for same critical region to be protected simultaneously by transactions and SpinLocks.
- Based on Rossbach et al. SOSP 2007
- Automatically falls back to SpinLocks in event of I/O, or irrecoverable transaction aborts.

Static Type Inference | *Haskell* Spring 2019

- Implemented constraint-unification based Hindley-Milner inference for a simply typed λ-calculus.
- Supported let-in, if-else, λ functions, function calls, recursion, arithmetic.

Technical Skills

Languages: Java, Python, C++, C, Kotlin, Scala
Frameworks: Flask, PyTorch, NumPy
Developer Tools: Git, Docker, TravisCI, Gradle