Chenyang Yuan

yuanchenyang@gmail.com

http://www.github.com/yuanchenyang http://www.chenyang.co

Education

PhD in Electrical Engineering and Computer Science 2018–Present
Massachusetts Institute of Technology, Cambridge, MA GPA: 5/5

MA in EECS 2016–2018
Massachusetts Institute of Technology, Cambridge, MA GPA: 5/5

BA in Computer Science 2012–2016
The University of Berkeley at California, Berkeley, CA GPA: 3.94/4

Papers

Chenyang Yuan and Pablo Parrilo, "Semidefinite Relaxations of Products of Nonnegative Forms on the Sphere", *Preprint, arxiv*

Chenyang Yuan and Pablo Parrilo, "Maximizing Products of Linear Forms, and the Permanent of Positive Semidefinite Matrices", *Mathematical Programming Series A*

J. Thai, **C. Yuan**, A. Bayen, "Resiliency of Mobility-as-a-Service Systems to Denial-of-Service Attacks", *2016 IEEE Transactions on Control of Network Systems*

C. Yuan, J. Thai, A. Bayen, "ZUbers against ZLyfts Apocalypse: An Analysis Framework for DoS Attacks on Mobility-as-a-Service Systems", *2016 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*

Thesis

Chenyang Yuan, "Focused Polynomials, Random Projections and Approximation Algorithms for Polynomial Optimization over the Sphere" *SM Thesis, MIT, 2018*

Programming Skills

Proficient in Python, Julia, Javascript, LaTeX, Emacs, Git
Experience in Java, C, Rust, Haskell, Scheme, HTML, Hadoop, Android, SQL, Assembly

Work Experience

Research Intern, *Lyft* *–August 2013*

- Helped improve internal tools
- Built an API auto-documentation system; designed and build an API Explorer: `https://www.clover.com/api_explorer`
- Created demo app using Clover's API: `https://github.com/clover/example-server`

Undergraduate Student Researcher, *UC Berkeley* *Spring 2015 – Spring 2016*

- I work on traffic research with Professor Alex Bayen. Projects I worked on include inferring route flows of cars from cellular connection data and using queueing theory to investigate possible attacks on on-demand taxi networks by calling taxies and then canceling the calls.

Undergraduate Student Researcher, *UC Berkeley* *Spring 2014 – Fall 2014*

- I worked with Professor Ras Bodik on the synthesis of a layout engine for an experimental browser, Servo. I helped built a backend which generates a layout engine in Rust, which replaces the hand-written layout engine in Servo. I also worked on writing a synthesis algorithm for incremental layout schedules, implemented in Rosette, a domain specific language for interfacing with SAT/SMT solvers.

Software Engineering Intern, *Clover* *June–September 2013*

- Helped improve internal tools
- Built an API auto-documentation system; designed and build an API Explorer: `https://www.clover.com/api_explorer`
- Created demo app using Clover's API: `https://github.com/clover/example-server`

Teaching Experience

TA for Algebraic Techniques and Semidefinite Programming, *MIT* *Spring 2021*
TA for Linear Algebra and Optimization, *MIT* *Fall 2020*

- Undergrad class designed to emphasize linear algebra
-

TA for Nonlinear Optimization, *MIT* *Spring 2020*
TA for Designing Information Devices and Systems, *UC Berkeley* *Fall 2015*

- New class designed to introduce linear algebra and applications to first year students
- Helped create labs, homework questions and write class notes

TA for Discrete Math and Probability, *UC Berkeley* *Spring 2015*

- Teach sections and labs, holds office hours
- On the content team that creates weekly homeworks, discussion sheets and their solutions
- Wrote an interactive browser-based virtual lab for polynomial interpolation

TA for Structure and Interpretation of Computer Programs, *UC Berkeley* *Fall 2013 – Fall 2014*

- Teach sections and labs, holds office hours
- Help write the autograder for projects
- Wrote Javascript interpreters for Scheme and Logic languages used in the class, so that students can interpret code on their browsers without installing interpreters on their machines.
- Ran and maintained the codereview system used to give students composition feedback from readers

Math Competition Trainer, *National University of Singapore High School* *March 2012*

- Compile problems and create training notes
- Conduct classes for grade 8-10 students

Physics Competition Trainer, *National University of Singapore High School* *March-August 2012*

- Prepare PhD-qualifying exam level problems
- Conduct classes for grade 11 students
- Create and grade a test

Software Projects

SumOfSquares.py `https://github.com/yuanchenyang/SumOfSquares.py`
Sum of squares optimization modeller built on top of picos. Features easy access to pseudoexpectation operators for both formulating problems and extracting solutions via rounding algorithms

Linear Algrbra DSL `https://github.com/yuanchenyang/llvm-linear-algebra-dsl`
An open-ended project for a compilers class. First created a set of tools for building domain specific languages (DSLs) using LLVM for code generation and created a DSL for linear algebra operations introducing lots of domain-specific optimizations. Then implemented an edge detector and part of an optical flow estimation algorithm using the DSL.

Facebook Group Archiver `http://archiver.chenyang.co`
A tool for saving Facebook groups in a local database and doing comprehensive searches locally. After the first download, it will sync the local database with the Facebook group during each run. Also includes a web-interface for stats, searching and doing database queries.

Interactive SICP Textbook `http://xuanji.appspot.com/isicp/1-1-elements.html`
Made an interactive version of the classic Structure and Interpretation of Computer Programs book with my friend. I created the asynchronous Javascript-based Scheme interpreter used on the website.

Self-Balancing Robot `http://youtu.be/Ps0Ex3ADR6k`
An open-ended project for my physics electronics lab class, built a self-balancing robot from scratch. Programmed a controller for it on an Arduino board.

WebGL Particle Simulator `http://www.chenyang.co/particles`
A simulation with thousands of particles attracted by gravity, created with WebGL and Javascript.

Python Control Flow Visualizer `http://pyvisualizer.chenyang.co`

An online tool that run python programs and visualize the code branching using D3.js

Scheme on TI-89 `https://github.com/yuanchenyang/TI89-Scheme`
Built a Scheme interpreter from scratch that runs on my TI-89 graphing calculator. It is written in C and supports a small subset of the Scheme language.

Building a Computer from Scratch `https://github.com/yuanchenyang/My-EOCS`
Following the instructions from a book called the Elements of Computing Systems, I built a CPU from logic gates using a hardware simulator. Then I proceeded to create an assembler for the CPU and a VM simulator that takes in VM code (similar to java bytecode) and outputs assembly code.

Logic Gate Simulator `https://github.com/yuanchenyang/Logic-Simulator`
Used Python to create a logic gate simulation system with constraint passing. This system also allows powerful abstractions to be made so that more complicated sets of gates can be created, saved and reused. This project won an honorable mention in the Facebook Battle of the Bay hackathon.

Perfect Strategy for Hog `https://github.com/yuanchenyang/Hog-Perfect-Strategy`
For a project in my CS class, we have to create artificial intelligence agents to compete in a dice game called Hog. I used dynamic programming and recursion to create a prefect strategy that cannot be beaten, thereby winning the contest.

Selected Awards

First Place, Cal vs Stanford Big Hack *Apr 2013*
Created a scheme interpreter in C on my TI-89 graphing calculator
Honorable Mention, 12th Asian Physics Olympiad *May 2011*
One of the 8 students representing Singapore in this competition.
Third Place, Hackers at Berkeley HackJam *Apr 2013*
Made an animation sequence on my TI-89 graphing calculator
Honorable Mention, Facebook Nor-Cal Hackathon 2013 *Oct 2013*
Built a online Python code branching visualizer.
Honorable Mention, Facebook Battle of the Bay Hackathon 2012 *Oct 2012*
Build a logic gate simulator with a graphical interface in Python.
Rank 15, Hackerrank Back to School Hackathon 2013 *Feb 2013*

Selected Coursework

CS: Berkeley: Graduate Algorithms and Theory, Compilers, Security, AI, Randomized Algorithms. MIT: Advanced Algorithms, Inference and Information, Geometric Computing, Algebraic Techniques and Semidefinite Programming
EE: MIT: Dynamic Systems and Control

Math: Berkeley: Complex Analysis, Honors Abstract Algebra. MIT: High-dimensional Statistics

Physics: Berkeley: Analytical Mechanics, Quantum Mechanics, General Relativity, Electronics Lab