Apoorva Bhagwat

+1 (513)-641-7503
bhagwat@cmu.edu

Work experience

Google **Seattle, WA**
Software Engineering Intern Summer 2017
- Developed a performance analysis tool for the Google Cloud Storage (Performance and Reliability) team.

Google **Mountain View, CA**
Software Engineering Intern Summer 2016
- Devloped a tool for automated analysis and interpretation of integration test results for the Borg team. *C++ and Python*

LibertyX **Boston, MA**
Software Engineering Intern Summer 2015
- Developed an asynchronous client-server framework based on WebSockets for smooth integration with LibertyX's partner bitcoin services. *Node.js and Python*

Carnegie Mellon University **Pittsburgh, PA**
Teaching Assistant and Head TA 2015-2017
- 15-122 (Imperative Programming) 1 semester
- 15-251 (Theoretical Computer Science) 5 semesters

Education

Bachelor of Science in Computer Science
with minors in Math and Chinese (Expected May 2018)
School of Computer Science, Carnegie Mellon University
GPA : 4.0/4.0

Skills

- **Programming** : C, C++, Python, OCaml, Java, x86 assembly, Standard ML, JavaScript, HTML/CSS3, LaTeX
- **Tools** : NumPy, SageMath, Git
- **Mathematics** : linear algebra, probability, abstract algebra, analysis, topology, computational complexity
- **Languages** : English, Hindi, Marathi (native/bilingual), Sanskrit, Mandarin (elementary)

Selected Honors

- Among the top 480 in the William Lowell Putnam mathematical competition.
- Represented India at the 12^{th} International Linguistics Olympiad in Beijing.

Projects

- **Parametric polymorphism for C0** Fall 2016
  After writing a compiler for the C0 programming language (which is a well-defined, type-safe subset of C), we extended the language with parametric polymorphism (i.e. generic types), and implemented a compiler for this extension. The result was a low-level C-like language that supports generic programming. *Compiler written in OCaml*
- **Graph Clustering** (**Research**) Spring 2017
  Under the guidance of Prof Avrim Blum, I investigated a novel notion of graph clustering called (α, β)-clustering that is useful in community detection. We proved that a natural problem related to this formulation is NP-hard.
- **Surroundify** Fall 2015
  At HackCMU 2015, we developed a web app that lets users utilize multiple machines to play music synchronously and create a surround-sound effect. The system supports surround-sound effects specified by the user. *Backend written in Python, using Flask*.
- **Tool for automated grade entry** 2016-2017
  This is an ongoing project to automatically digitize grades from homework submissions, using tools from computer vision (useful for TAs of large classes). *Written in Python using OpenCV*.
- **Universal Register Machines** Fall 2015
  As a final project for a theory of computation class, I implemented an optimizing register machine interpreter (that speeds up execution of nested register machine loops from exponential to constant time), and a universal register machine (i.e. a register machine program that runs other register machine programs). *Written in Python*.
- **Programming assignment design** Spring 2016
  Designed a programming assignment for a CS theory class at CMU and managed its autograding infrastructure. This assignment involved proving the Turing-completeness of register machines by incrementally building primitives to simulate Turing machines.
- **Linguistics olympiad problem design** 2014-2016
  I have been contributing to India's national linguistics olympiad as a problem designer and tester. Here is one of our past papers.

Selected Coursework

Compiler Design	Abstract Algebra
Computer Systems	Real Analysis
Machine Learning	Automata, Algebra and Logic (a.k.a. CDM)
Artificial Intelligence	Complexity Theory
Parallel Data Structures and Algorithms	Theory of Programming Languages