ALAN T. LEUNG

alan@leung.work
https://alan.leung.work
https://www.linkedin.com/in/alan-leung-dev/

EMPLOYMENT

Research Software Engineer — **Microsoft** — **November 2017 – Present**

- Building industrial-strength technologies to help users program by example: instead of coding, give examples

Graduate Student Researcher — **University of California, San Diego** — **Fall 2010 – Spring 2017**

- Implemented several research tools in the areas of compilers, program analysis, and program synthesis
- Published 5 papers at top academic conferences: four as project lead, first author, and conference presenter
- Mentored 2 junior graduate students

Research Intern — **Microsoft Research** — **Summer 2012**

- Implemented program transformations to reduce bandwidth utilization of cloud-based mobile apps

Microprocessor Design Engineer — **Intel** — **August 2004 – April 2009**

- Implemented cache logic and circuits for two Intel Itanium 2 microprocessors
- Created software tools for circuit simulation and automation
- Mentored 3 junior engineers
- Awarded outstanding rating (top 5% of employees) in 2008 and Division Recognition Award in 2009

EDUCATION

La Jolla, CA — **University of California, San Diego** — **Fall 2010 – Spring 2017**

- PhD in Computer Science (2017), GPA: 4.0, Advisor: Sorin Lerner
- MS in Computer Science (2013), GPA: 4.0, Powell Fellow

Ithaca, NY — **Cornell University** — **Fall 2000 – Spring 2004**

- BS in Electrical and Computer Engineering, GPA: 4.0, Summa Cum Laude

PROJECTS

- **Parser Synthesis IDE** (2014-2017). Cloud-hosted visual development environment for constructing parsers by example. https://parsimony-ide.github.io. Clojure, ClojureScript, C, React
- **C-to-Verilog Equivalence Checker** (2012 – 2013). Program analysis toolchain to check semantic equivalence between a Verilog implementation and the C source code from which it was generated. Scala, Z3
- **GPU Kernel Test Amplification** (2011 – 2012). Program analysis toolchain to detect races and determinism bugs in highly-parallel GPU code. C, CUDA, Perl, Haskell, LLVM

PUBLICATIONS (SELECTED)

- **Interactive Parser Synthesis by Example**. Alan Leung, John Sarracino, Sorin Lerner. PLDI 2015.
- **Verifying GPU Kernels via Test Amplification**. Alan Leung, Manish Gupta, Yuvraj Agarwal, Rajesh Gupta, Ranjit Jhala, Sorin Lerner. PLDI 2012.

LANGUAGES AND TECHNOLOGIES

Languages

- Proficient: C#, Python, Clojure, ClojureScript
- Experienced: Java, C, Perl, Haskell, Scala

Technologies

- Proficient: functional programming, compilers, program analysis, program synthesis
- Experienced: React, LLVM, CUDA, Linux, git, Datomic, Vagrant, Riemann, Z3