

# Chungha Sung

Los Angeles, CA, USA  
sch8906@gmail.com  
(+1) 540-750-9013

## EDUCATION

---

- University of Southern California (USC)** *Aug. 2017 – Present*  
Los Angeles, USA  
*Ph.D. Candidate in Computer Science*  
Reliable & Secure Software lab under Dr. Chao Wang
- Virginia Polytechnic Institute and State University (Virginia Tech)** *Aug. 2014 – Dec. 2016*  
Blacksburg, USA  
*Master of Science in Computer Engineering (GPA: 3.73/4.0)*
- Sung Kyun Kwan University (SKKU)** *Mar. 2007 – Aug. 2013*  
Seoul, Korea  
*Bachelor of Science in Engineering (GPA: 4.03/4.5)*  
Parallel Architecture and Programming Language Lab (PAPL) under Dr. Jae W. Lee
- Gangwon Science High School** *Mar. 2005 – Feb. 2007*  
Gangwondo, Korea  
*Grade skipping and early graduation (1 year)*

## EXPERIENCE

---

- Research Assistant in Reliable & Secure Software Lab, USC** *Aug. 2017 – Present*  
- Advisor: Prof. Chao Wang
- Research Intern in Microsoft Research (MSR) in India** *May. 2017 – Aug. 2017*  
- Manager: Dr. Akash Lal, Dr. Kaushik Rajan  
- Extended a Scope query optimization tool to support uninterpreted functions and various combinations of columns (F#, C#, Scope query, Visual studio)
- Research Assistant in Reliable & Secure Software Lab, Virginia Tech** *Jan. 2015 – May. 2017*  
- Static Analysis of web application to optimize testing process (C++, JavaScript, nodejs, Z3-SMT solver)  
- Static Analysis for verification of interrupt-driven programs (C++, LLVM, Z3-SMT solver)
- Teaching Assistant in ECE department, Virginia Tech** *Sep. 2014 – Dec. 2014*  
- Managed lab sessions and graded course assignments in Microcontroller Interfacing class (Sophomore course)
- Research Assistant in Parallel Architecture and Programming Lab, SKKU** *Mar. 2013 – Aug. 2014*  
- Advisor: Prof. Jae W. Lee  
- Implemented a graphical visualization program
- Google Summer of Code 2013 (GSOC 2013)** *May. 2013 – Sep. 2013*  
- Contributed to publishing a graphical open-source package named "RIGHT" in R project
- Teaching Assistant in Semiconductor Systems Engineering department, SKKU** *Spring - 2013, 2012*  
- Made a course project running on FPGA with VHDL and managed lab sessions in logic design laboratory (Sophomore course)  
- Made three ARM-core based sorting optimization projects with Verilog HDL and graded all course assignments in Digital System class (Junior course)
- Summer Internship in Ahn Lab (The largest security vendor in Korea)** *Jul. 2012 – Aug. 2012*  
- Changed boot loader sequence of server for making server maintenance easier

## PUBLICATION

---

[Conference papers]

**Chungha Sung**, Markus Kusano, Chao Wang, “**Modular verification of interrupt-driven programs**”, *The 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Oct 2017

**Chungha Sung**, Markus Kusano, Nishant Sinha, Chao Wang, “**Static DOM Event Dependency Analysis for Testing Web Applications**”, *The 24th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE)*, Oct 2016

[Posters]

**Chungha Sung**, TaeJoon Song, Jae W. Lee, and Junghoon Lee, “**RIGHT: an HTML canvas and JavaScript-based interactive data visualization package for linked graphics**”, *The R User Conference (UseR)*, UCLA, Los Angeles, California, July 2014

## AWARDS AND GRANTS

---

[Scholarships]

Full Merit-based awards from SKKU for full academic years

Mar. 2007 – Aug. 2013

Scholarship from Samsung Electronics for 2 years

Mar.2007 – Dec.2008

[Awards]

ACM SIGSOFT travel grant for FSE conference (\$1,000)

Nov. 2016

CAV (International Conference on Computer Aided Verification) travel award (\$1,500)

July. 2016

First prize for undergraduate graduation thesis and project award from SKKU

June. 2013

Dean's List award (for top 5% students in one semester) from SKKU

Apr. 2012

## SKILLS

---

**Language experience:** C, C++, C#, F#, JAVA, JavaScript, HTML, R, Matlab, Datalog, Assembly x86/x64, Verilog, VHDL

**Tool experience:** Z3-SMT solver, Visual Studio, nodejs, LLVM, eclipse, Cadence circuit tool, Modelsim, Altera quartus

**Version Controller:** git, svn, cvs

## PROJECT SUMMARIES

---

### [Extension of a Scope query optimization tool]

During the internship at MSR, I extended a scope query optimization tool. The tool consists of two parts: one part is parser and another part is template generation. Basically, it parses a query file and then prints a C-style code template, and the template is fed into a programming synthesis tool to optimize it. Then, the optimized C-style program is translated into an optimized query. However, the tool only supported a small set of queries because many queries use many combinations of columns and uninterpreted functions. So, I changed more than half of the tool, both parser and template generation parts, to support various combinations of columns and uninterpreted functions. To achieve this, I got used to F#, C# and Scope query.

### [Modular verification of interrupt-driven programs]

Since there has been relatively lack of verification methods for interrupt-driven programs even the behavior of interrupts is similar with that of threads, we introduced a more accurate verification method for interrupt-driven programs by leveraging a thread-modular analysis for concurrent programs. By modeling interrupt behavior such as priority information and preemption information with Z3 SMT solver, I combined this information with a thread-modular abstract interpretation analysis. Therefore, I could remove infeasible interference for interrupt-driven programs and could achieve a more accurate analysis without losing computation overhead.

I compared our analysis to state-of-the-art techniques one of which is a thread-modular analysis for concurrent programs, and another is a model checking based interrupt-driven analysis tool, with about 35 C-base interrupt-driven programs. Finally, my tool showed its higher accuracy than other two tools with getting a little overhead.

#### **[Static DOM Event Dependency Analysis for Testing Web Applications]**

Even though there were many approaches to analyze web applications, it was hard to analyze the web application accurately due to its inherent characteristic of user-driven events using the DOM (Document Object Model). This project tries to model the DOMs to find out dependencies between two events which are attached to the DOMs. To achieve this, I made a JavaScript parser from the scratch using JavaScript and nodejs, and modified JS\_WALA to build a CFG (Control Flow Graph). Based on the CFG, translated datalog facts are printed. My analysis is inter-procedural and flow-insensitive pointer analysis. Finally, from the facts I got and the rules I made, I can figure out the dependencies among all events over-approximately.

To show the usefulness my work, I hooked up the automated testing tool named Artemis to shrink the searching space of it using POR (Partial Order Reduction) with the dependency information. I tested with about 20 real world JavaScript games which are highly user-driven events demanded program, and the result shows that the overhead of this analysis is really negligible and we got about 16% improvement of code coverage in average.

#### **[RIGHT: an HTML canvas and JavaScript-based interactive data visualization package for linked graphics]**

Our team developed an interactive visualization graphic package as an open source in R project during the Google summer of Code 2013 period to help people easily deal with various aspects of data by inserting new data or deleting outliers while watching the graphs. To support these features easily our team used the browser to show the graph using HTML and JavaScript. The advantage of using the browser to support the graphic package is people can easily use this package on the various platforms such as mobile devices. Furthermore, we try to support server-side offload using the package named Shiny with the large number of data which are hard to compute at the client-side platform.

#### **[Development of boot loader to support easy maintenance of a server]**

I worked on prototyping new boot loader for an appliance that should be compatible with x86/MIPS architecture and Linux-based operating systems and simplified customer support and troubleshooting, ensuring that the boot loader would provide multi-level booting, securing the system to protect not only back side information, but customers' information as well.

I wrote the boot loader from scratch, as some of these features were not preexisting projects including GNU GRUB.

### **OTHER EXPERIENCES**

---

Vice president of Korean Student Association (KSA) in Virginia Tech	<i>Jun. 2015 – May. 2016</i>
PR student ambassador of the department of Semiconductor Systems Engineering	<i>Mar. 2007 – Aug. 2013</i>
3rd prize in individual kendo (Japanese fencing) competition of the League of Seoul Universities: 2 times	<i>Dec. 2008, Dec. 2011</i>
Administrative Duties during Military service: sergeant squad leader	<i>Jan. 2009 – Nov. 2010</i>
Voluntary service in the countryside (Choongnam yeongi-gun, Korea)	<i>Aug. 2008</i>