Jinseong Jeon

425 Urban Plz, Kirkland, WA 98033

jsjeon@cs.umd.edu
https://jsjeon.github.io/

Education

- **University of Maryland, College Park** — College Park, MD
  Ph.D. in Computer Science — Feb 2016
- **KAIST** — Daejeon, Korea
  M.S. in Computer Science — Feb 2007
- **KAIST** — Daejeon, Korea
  B.S. in Computer Science — Feb 2005

Experience

- **Google LLC** — Kirkland, WA
  Tech Lead / Staff Software Engineer — Mar 2023 / Nov 2024 – Present
  - *Kotlin compiler at Google*: Leading Kotlin frontend and JVM compiler efforts within Google
    * Spearheaded K2 adoption in Google monorepo
    * Guided Android Studio migration to K2 and K2 IDE plugin
    * Drove tooling migrations: Android Lint and Metalava in AndroidX

- **Google LLC** — Kirkland, WA
  Senior Software Engineer — May 2019 – Oct 2024
  - *IntelliJ IDEA* (contributions): Designed and implemented K2 UAST from scratch:
    * Unified AST for Java and Kotlin, backbone of Android Lint
    * Landed to AndroidX: average 1.31x, up to 1.72x faster
    * Adopted by external companies: Meta, Square, and Mozilla
  - *Kotlin compiler* (contributions): Contributed to K2, the new Kotlin compiler frontend, end-to-end:
    * Resolution: types, declarations, call targets, SAM/`suspend` conversion, etc.
    * Static analyses: control-flow / data-flow analysis, diagnostics
    * Conversion to backend IR

- **Google LLC** — Kirkland, WA
  Software Engineer — Feb 2016 – Apr 2019
  - *Android Compiler Toolchain*: D8 dexer and R8 shrinker (contributions): Researched; designed; implemented; and deployed optimizations and obfuscations, such as:
    * local type/nullability analysis, call-site optimization (e.g., remove Kotlin intrinsics calls),
    * `StringBuilder` optimization, compile-time reflection simplification, constant/call canonicalization,
    * Kotlin `@Metadata` rewriting, identifier string obfuscation, package obfuscation
  - *Google Compute Engine*: Sole-tenant nodes, Committed use discounts

- **University of Maryland, College Park** — College Park, MD
  Research Assistant — Jun 2011 – Feb 2016
  - Pasket: *Synthesizing Framework Models for Symbolic Execution* [1, 3, 4, 5]: Researched and developed scalable synthesis of models for object-oriented, event-driven frameworks, such as Java Swing and Android, that enable other static analysis tools to analyze real-world apps effectively and efficiently
  - *Redexer: Dalvik Bytecode Instrumentation Framework* [6]: Developed a general-purpose bytecode rewriting framework for Android, which is composed of a rich set of utilities that let programmers parse, manipulate, and generate Dalvik bytecode from scratch

- **Google Inc.** — Mountain View, CA
  Software Engineering Intern — May – Aug 2015

– *Espresso Test Recorder*: Designed and prototyped an Android Studio plugin that records user interactions via instrumentation and synthesizes repeatable Espresso test code from the logs

• **Microsoft Research** Redmond, WA
Research Intern *May – Aug 2014*

– *AppFormer: Synthesizing Cross-Platform Mappings from Examples*: Researched automatic creation of platform-to-platform mappings (e.g., Android to Windows Phone) by logging example apps' behaviors and summarizing them via template-based program synthesis

• **KAIST** Daejeon, Korea
Research Assistant *Mar 2006 – Feb 2007*

– *RTFA: Layout Transformation for Heap Objects* [2, 7]: Developed a compiler optimization that infers data structure access patterns and transforms heap layouts to improve program performance by increasing cache hit ratios (*won an Outstanding Master's Thesis Award from the department*).

– *Raccon: Buffer Overrun Analyzer for C Programs*: Modified a buffer overrun analyzer for C programs so that it can distinguish k different call contexts during analysis

Awards and Activities

Professional Activities:

- Artifact Evaluation Committee, PLDI '15, PLDI '20 2015, 2020
- Reviewer, IEEE Transactions on Mobile Computing (TMC) 2015, 2018
- Reviewer, POPL '15, ICSE '15 2014
- Reviewer, Journal of Information Security and Applications (JISA) 2014
- Reviewer, IEEE Transactions on Dependable and Secure Computing (TDSC) 2013

Publications

Journal Articles

[1] Jinseong Jeon, Xiaokang Qiu, Armando Solar-Lezama, and Jeffrey S. Foster. An Empirical Study of Adaptive Concretization for Parallel Program Synthesis. *Formal Methods in System Design (FMSD)*, 50(1):75–95, Mar 2017.

[2] Jinseong Jeon, Keoncheol Shin, and Hwansoo Han. Abstracting Access Patterns of Dynamic Memory Using Regular Expressions. *ACM Transactions on Architecture and Code Optimization (TACO)*, 5(4):18:1–18:28, Mar 2009.

Conference/Workshop Proceedings

[3] Jinseong Jeon, Xiaokang Qiu, Jonathan Fetter-Degges, Jeffrey S. Foster, and Armando Solar-Lezama. Synthesizing Framework Models for Symbolic Execution. In *38th International Conference on Software Engineering (ICSE '16)*, May 2016.

[4] Jinseong Jeon, Xiaokang Qiu, Jeffrey S. Foster, and Armando Solar-Lezma. JSKETCH: Sketching for Java. In *10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE '15)*, Sep 2015.

[5] Jinseong Jeon, Xiaokang Qiu, Armando Solar-Lezama, and Jeffrey S. Foster. Adaptive Concretization for Parallel Program Synthesis. In *the 27th International Conference on Computer Aided Verification (CAV '15)*, Jul 2015.

[6] Jinseong Jeon, Kristopher K. Micinski, Jeffrey A. Vaughan, Ari Fogel, Nikhilesh Reddy, Jeffrey S. Foster, and Todd Millstein. Dr. Android and Mr. Hide: Fine-grained Permissions in Android Applications. In *ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '12)*, pages 3–14, Oct 2012.

[7] Jinseong Jeon, Keoncheol Shin, and Hwansoo Han. Layout Transformations for Heap Objects Using Static Access Patterns. In *Compiler Construction (CC '07)*, pages 187–201, Mar 2007.