

Kihong Heo

Post-doctoral Researcher
Department of Computer and Information Science
University of Pennsylvania
3330 Walnut St, Philadelphia, PA 19104, USA
✉ : kheo@cis.upenn.edu 🌐 : <http://www.cis.upenn.edu/~kheo>

Research Interests

My research aims to semantic-based static analysis for safe and reliable software. The central challenge is to build scalable, precise and user-friendly static analysis systems. Technically, I am interested in combining logical and probabilistic reasoning including data-driven adaptive program analysis, interactive program reasoning system, and learning-based program synthesis/transformation.

Education

Computer Science and Engineering, Seoul National University Mar 2009 – Aug 2017
Ph.D. in Computer Science and Engineering
Dissertation: Selectively Sensitive Static Analysis by Impact Pre-analysis and Machine learning
Outstanding Dissertation Award
Advisor: Prof. Kwangkeun Yi

Computer Science and Engineering, Seoul National University Mar 2005 – Feb 2009
B.S. in Computer Science and Engineering

Experience

University of Pennsylvania Jul 2017 – present
Post-doctoral Researcher
Advisor: Prof. Mayur Naik

Facebook Apr 2017 – Jun 2017
Research Scientist (contingent)

The Hong Kong University of Science & Technology Sep 2011 – Feb 2012
Visiting Student

Research Project

- ▶ **Petablox: Declarative Program Analysis for Big Code** 2017 – present
I have been a core developer of Petablox, a declarative program analysis system for Big Code. The main goal of Petablox is to leverage collective knowledge amassed from analyzing existing programs and human feedback. In particular, I have been working on an interactive program reasoning system. Since program analyses necessarily make approximations, they often produce many false alarms that hinder program reasoning. To address the problem, this approach

associates each program analysis alarm with a confidence value by performing Bayesian inference on a probabilistic model derived from the analysis rules. In each iteration, the user inspects the alarm with the highest confidence and labels its ground truth, and the approach recomputes the confidences of all alarms given the feedback. This project is funded by DARPA (Defense Advanced Research Project Agency, USA). (<http://petablox.org>)

- ▶ **ASPIRE: Transformations for Reducing Software Complexity** 2017 – present
I have been working on the ASPIRE project for program reduction. The complexity and bloat of today's software have led to decreased performance and increased security vulnerabilities. The ASPIRE project aims for a fully automated reduction process that can handle legacy source and binary code, and scale to large, complex programs. The resulting subsetted code should preserve properties of the original code, be maintainable and extensible, and provide improved performance and security. I focus on semantic-aware program reduction techniques. This project is funded by ONR (The Office of Naval Research, USA). (<http://aspire.cis.upenn.edu>)
- ▶ **Sparrow: a static analyzer for C program** 2011 – present
I have been a core developer of Sparrow and mainly involved in its relational analysis engine. Sparrow is a state-of-the-art static analyzer that aims to verify the absence of fatal bugs in C source. In particular, I have been developing techniques for cost-effective relational analysis, context-sensitive analysis, unsound analysis enabled by semantic-based pre-analysis and machine learning. Sparrow is now open-source and available via GitHub. This project was funded by NRF (National Research Foundation of Korea). (<http://www.github.com/ropas/sparrow>)
- ▶ **Inferbo: Infer-based buffer overrun analyzer** 2016 – 2017
I have been a core developer of Inferbo that is a precise and scalable buffer-overrun analyzer based on the Facebook Infer analyzer. Inferbo scales to a large and fast-moving codebase like Facebook thanks to Infer's modular analysis engine. Modular analysis separately analyzes each sub-part (e.g., procedure) of a large program and composes the whole analysis result using the partial information. Despite its efficiency, modular analysis has previously been used for checking relatively simple or inductively-defined properties. To achieve a modular buffer-overrun analyzer which requires sophisticated numerical reasoning, we designed a symbolic abstract domain and procedure summary by observing some common buffer-overrun issues in Facebook's codebase. Inferbo is now merged in Facebook Infer and available via the Infer GitHub. This project was funded by Facebook. (<https://github.com/facebook/infer>)
- ▶ **Selective X-sensitive Analysis** 2013 – 2017
I have been working on the selective X-sensitive analysis framework and mainly developing selectively relational analysis. Selective X-sensitive analysis applies certain sensitivity X (e.g. context, flow, or relational analysis) only when and where doing so is likely to improve the precision of the main analysis. The challenge is to estimate the impact of X on the main analysis's precision. To this end, we have developed 1) impact pre-analyses that are based on the abstract interpretation framework, and 2) machine learning techniques that learn the behaviors of the impact pre-analyses. In the project, I mainly designed an impact pre-analysis and machine learning techniques for the octagon relational analysis. This project was funded by NRF (National Research Foundation of Korea).
- ▶ **Global Sparse Analysis Framework** 2011 – 2012
I joined the sparse analysis project and designed the sparse interval analysis engine part. Our sparse analysis framework provides a general method for achieving global static analyzers that are precise, sound, yet also scalable. Based on the framework, we have derived a sparse version of Sparrow which is 175x more scalable than the baseline in terms of lines of code and scales to a million lines of C programs. In the project, I participated in designing a pre-analysis for the

interval analysis and implementing the sparse interval analysis. This project was funded by NRF (National Research Foundation of Korea). (<http://ropas.snu.ac.kr/sparseanalysis>)

Publications

1. Interactive Program Reasoning Using Bayesian Inference.
Mukund Raghothaman, Sulekha Kulkarni, **Kihong Heo**, and Mayur Naik.
In Programming Language Design and Implementation (PLDI'18), 2018.
2. Accelerating Search-Based Program Synthesis Using Learned Probabilistic Models.
Woosuk Lee, **Kihong Heo**, Rajeev Alur, and Mayur Naik.
In Programming Language Design and Implementation (PLDI'18), 2018.
3. Adapting Static Analysis via Learning with Bayesian Optimization.
Kihong Heo, Hakjoo Oh, Hongseok Yang, and Kwangkeun Yi.
ACM Transactions on Programming Languages and Systems (to appear), 2018.
4. Learning Analysis Strategies for Octagon and Context Sensitivity from Labeled Data Generated by Static Analyses.
Kihong Heo, Hakjoo Oh, and Hongseok Yang.
Formal Methods in System Design (to appear), 2018.
5. Automatically generating features for learning program analysis heuristics for C-like languages.
Kwonsoo Chae, Hakjoo Oh, **Kihong Heo**, and Hongseok Yang.
PACMPL, 1(OOPSLA), 101:1–101:25, 2017.
6. Machine-Learning-Guided Selectively Unsound Static Analysis.
Kihong Heo, Hakjoo Oh, and Kwangkeun Yi.
In International Conference on Software Engineering (ICSE'17), 2017.
7. Selective Conjunction of Context-sensitivity and Octagon Domain toward Scalable and Precise Global Static Analysis.
Kihong Heo, Hakjoo Oh, and Kwangkeun Yi.
Software—Practice & Experience, 47(11), 1677–1705, 2017.
8. Sound Non-Statistical Clustering of Static Analysis Alarms.
Woosuk Lee, Wonchan Lee, Dongok Kang, **Kihong Heo**, Hakjoo Oh, and Kwangkeun Yi.
ACM Transactions on Programming Languages and Systems, 39(4), 16:1–16:35, 2017.
9. Learning a Variable-Clustering Strategy for Octagon from Labeled Data Generated by a Static Analysis.
Kihong Heo, Hakjoo Oh, and Hongseok Yang.
In International Static Analysis Symposium (SAS'16), 2016.
10. Selective X-Sensitive Analysis Guided by Impact Pre-Analysis.
Hakjoo Oh, Wonchan Lee, **Kihong Heo**, Hongseok Yang, and Kwangkeun Yi.
ACM Transactions on Programming Languages and Systems, 38(2), 6:1–6:45, 2016.
11. Widening with Thresholds via Binary Search.
Sol Kim, **Kihong Heo**, Hakjoo Oh, and Kwangkeun Yi.
Software—Practice & Experience, 46(10), 1317–1328, 2016.

12. Selective Context-sensitivity Guided by Impact Pre-analysis.
Hakjoo Oh, Wonchan Lee, **Kihong Heo**, Hongseok Yang, and Kwangkeun Yi.
In Programming Language Design and Implementation (PLDI'14), 2014.
13. Global Sparse Analysis Framework.
Hakjoo Oh, **Kihong Heo**, Wonchan Lee, Woosuk Lee, Daejun Park, Jeehoon Kang, and Kwangkeun Yi.
ACM Transactions on Programming Languages and Systems, 36(3), 8:1–8:44, 2014.
14. A Sparse Evaluation Technique for Detailed Semantic Analyses.
Yoonseok Ko, **Kihong Heo**, and Hakjoo Oh.
Computer Languages, Systems & Structures, 40(3-4), 99–111, 2014.
15. Design and Implementation of Sparse Global Analyses for C-like Languages.
Hakjoo Oh, **Kihong Heo**, Wonchan Lee, Woosuk Lee, and Kwangkeun Yi.
In Programming Language Design and Implementation (PLDI'12), 2012.

Software

I have contributed to the following open-source software:

- ▶ Sparrow: A Static Analyzer for C Program
<http://www.github.com/ropas/sparrow>
- ▶ Petablox: Declarative Program Analysis Framework for Big Code
<https://github.com/petablox-project/petablox>
- ▶ Infer: A Static Analyzer for Java, C, C++, and Objective-C
<http://www.github.com/facebook/infer>

Talks

- ▶ Interactive Alarm Ranking System using Bayesian Inference
Invited talk, Korea University. 01/04/2018
- ▶ Machine-Learning-Guided Selectively Unsound Static Analysis
Invited talk, Naver. 06/26/2017
- ▶ Machine-Learning-Guided Selectively Unsound Static Analysis
Paper presentation, ICSE 2017. 05/26/2017
- ▶ Inferbo: Infer-based buffer-overflow analyzer
Invited talk, Korea University. 04/14/2017
- ▶ Inferbo: Infer-based buffer-overflow analyzer
Invited talk, KAIST. 03/24/2017
- ▶ Selectively Sensitive Static Analysis by Impact Pre-analysis and Machine Learning
Invited talk, Codemind. 02/20/2017
- ▶ Learning a Variable-Clustering Strategy for Octagon from Labeled Data Generated by a Static Analysis
Paper presentation, SAS 2016. 09/08/2016

Teaching Experience

- ▶ **Teaching Assistant** : SNU 4541.664 Program Analysis (grad) Spring 2010
- ▶ **Teaching Assistant** : SNU 4190.210 Programming Languages Spring 2009

References

Kwangkeun Yi

Professor
Dept. of Computer Science and Engineering
Seoul National University
Email: kwang@ropas.snu.ac.kr

Hakjoo Oh

Assistant Professor
Dept. of Computer Science and Engineering
Korea University
Email: hakjoo_oh@korea.ac.kr

Hongseok Yang

Professor
School of Computing
KAIST
Email: hongseok.yang@kaist.ac.kr

Mayur Naik

Associate Professor
Dept. of Computer and Information Science
University of Pennsylvania
Email: mhnaik@cis.upenn.edu

Last updated: May 12, 2018
<http://www.cis.upenn.edu/~kheo>