

Yunho Kim

Ph. D in Computer Science
Department of Computer Science
Hanyang University

yunhokim@hanyang.ac.kr
614 H305 (ITBT), Hanyang University
222 Wangsimni-ro, Seongdong-gu, Seoul, South Korea

Education

Feb 2009-Feb 2017	Ph. D in Computer Science, KAIST (advisor: Prof. Moonzoo Kim) Thesis: Automated Unit Test Generation with Realistic Unit Context Synthesis for Low False Alarms
Mar 2007-Feb 2009	M. S. in Computer Science, KAIST (advisor: Prof. Moonzoo Kim) <ul style="list-style-type: none">• Integrated Master and Ph. D program in computer science
Mar 2003-Feb 2007	B. S. in Computer Science, KAIST
Mar 2001-Feb 2003	Seoul Science High School

Experience

Sep 2020-Present	Assistant Professor, Department of Computer Science, Hanyang University
Mar 2018-Aug 2020	Research Assistant Professor, School of Computing, KAIST
Feb 2017-Feb 2018	Postdoctoral Researcher, Software Testing and Verification(SWTV) group, KAIST (advisor: Prof. Moonzoo Kim)

Research Interests

My research interests span automated software testing and debugging through program analysis and formal verification techniques especially for safety critical embedded software.

- **Automated Software Unit/System Test Generation using Concolic Testing and Mutation Analysis**
- **Effective Fault Localization using Mutation Analysis and Machine Learning**
- **Applying Automated Testing and Debugging Techniques to Safety Critical Embedded SW in Industries**

Publications (Google scholar citations:789, h-index: 16)

• **Refereed International Journal Articles**

- [1] **Y. Kim** and S. Hong, Learning-based Mutant Reduction using Fine-grained Mutation Operators, Software Testing, Verification & Reliability (STVR), 2019 (Under review)
- [2] **Y. Kim** and S. Hong, DeMiner: Test Generation for High Test Coverage through Mutant Exploration, Software Testing, Verification & Reliability (STVR), 2019
- [3] **Y. Kim**, D. Lee, J. Baek, and M. Kim, MAESTRO: Automated Test Generation Framework for High Test Coverage and Reduced Human Effort in Automotive Industry, Information and Software Technology (IST), 2019
- [4] **Y. Kim**, S. Mun, S. Yoo, and M. Kim, Precise Learn-to-Rank Fault Localization using Dynamic and Static Features of Target Programs, ACM Transactions on Software Engineering and Methodology (TOSEM), 28(4), 2019
- [5] S. Hong, T. Kwak, B. Lee, Y. Jeon, B. Ko, **Y. Kim**, M. Kim, MUSEUM: Debugging Real-World Multilingual Programs Using Mutation Analysis, Information and Software Technology (IST), vol 82, pages 80-95, Feb 2017
- [6] Z. Xu, **Y. Kim**, M. Kim, M. Cohen, and G. Rothermel, Directed Test Suite Augmentation: An Empirical Investigation, Journal of Software Testing, Verification and Reliability (STVR), volume 25, issue 2, pages 77-114, March 2015

- [7] M. Kim, **Y. Kim**, and Y. Choi, Concolic Testing of the Multi-sector Read Operation for Flash Storage Platform Software, Formal Aspects of Computing (FACJ), vol 24, no 2, 2012
- [8] M. Kim, **Y. Kim**, and H. Kim, A Comparative Study of Software Model Checkers as Unit Testing Tools: An Industrial Case Study, IEEE Transactions on Software Engineering (TSE), vol 37, no 2, March 2011

• **Refereed International Conference Papers**

- [1] **Y. Kim**, S. Hong and M. Kim, Target-driven Compositional Concolic Testing with Function Summary Refinement for Effective Bug Detection, ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), Aug 26-30, 2019 (**acceptance rate:24.4%**)
- [2] **Y. Kim**, D. Lee, J. Baek, and M. Kim, Concolic Testing for High Test Coverage and Reduced Human Effort in Automotive Industry, Intl. Conf. on Software Engineering (ICSE) Software Engineering In Practice (SEIP) track, May 25-31, 2019 (**acceptance rate:25%**)
- [3] **Y. Kim**, Y. Choi, and M. Kim, Precise Concolic Unit Testing of C Programs with Alarm Filtering Using Symbolic Calling Contexts, International Conference on Software Engineering (ICSE), 2018 (**acceptance rate:21%**)
- [4] **Y. Kim**, S. Hong, B. Ko, D. Phan, and M. Kim, Invasive Software Testing: Mutating Target Programs to Diversify Test Exploration for High Test Coverage, IEEE International Conference on Software Testing, Verification and Validation (ICST), April 9-13, 2018 (**Distinguished paper award**) (acceptance rate: 25.2%)
- [5] L. Phan, **Y. Kim**, M. Kim, MUSIC: Mutation Analysis Tool with High Configurability and Extensibility, Mutation Workshop, 2018
- [6] S. Hong, B. Lee, T. Kwak, Y. Jeon, B. Ko, **Y. Kim**, M. Kim, Mutation Based Fault Localization for Real-World Multilingual Programs, IEEE/ACM International Conference on Automated Software Engineering (ASE), Nov 9-13, 2015 (**acceptance rate: 19%**)
- [7] **Y. Kim** and M. Kim, SAT-based Bounded Software Model Checking for Embedded Software: A Case Study, Asia-Pacific Software Engineering Conference (APSEC), Dec 1-4 2014 (acceptance rate: 30%)
- [8] **Y. Kim**, Z. Xu, M. Kim, M. Cohen, and G. Rothermel, Hybrid Directed Test Suite Augmentation: An Interleaving Framework, IEEE International Conference on Software Testing, Verification and Validation (ICST), March 31-April 4, 2014 (acceptance rate: 28%)
- [9] S. Moon, **Y. Kim**, M. Kim, S. Yoo, Ask the Mutants: Mutating Faulty Programs for Fault Localization, IEEE International Conference on Software Testing, Verification and Validation (ICST), March 31-April 4, 2014 (acceptance rate: 28%)
- [10] **Y. Kim**, Y. Kim, T. Kim, G. Lee, Y. Jang, and M. Kim, Automated Unit Testing of Large Industrial Embedded Software using Concolic Testing, IEEE/ACM Automated Software Engineering (ASE) Experience track, Nov 11-15, 2013 (**acceptance rate:23%**)
- [11] **Y. Kim**, M. Kim, Y. Kim, and Y. Jang, Industrial Application of Concolic Testing Approach: A Case Study on libxif by Using CREST-BV and KLEE, Intl. Conf. on Software Engineering (ICSE), Software Engineering in Practice (SEIP) track, June 2-9, 2012 (**acceptance ratio:19%**)
- [12] M. Kim, **Y. Kim**, and G. Rothermel, A Scalable Distributed Concolic Testing Approach: An Empirical Evaluation, IEEE International Conference on Software Testing, Verification, and Validation (ICST), April 17-21, 2012 (acceptance ratio: 27%)
- [13] M. Kim, **Y. Kim** and Y. Jang, Industrial Application of Concolic Testing on Embedded Software: Case Studies, IEEE International Conference on Software Testing, Verification and Validation (ICST) Industrial track, April 17-21, 2012, **nominated as a best paper (acceptance ratio: 23%)**

- [14] Z. Xu, **Y. Kim**, M. Kim and G. Rothermel, A Hybrid Directed Test Suite Augmentation Technique, IEEE Intl. Symp. on Software Reliability Engineering (ISSRE), Hiroshima, Japan, Nov 29-Dec 2 2011 (**acceptance ratio: 25%**)
- [15] M. Kim and **Y. Kim**, Automated Analysis of Industrial Embedded Software, Automated Technology for Verification and Analysis (ATVA), Taipei, Taiwan, Oct 11-14 2011 (invited paper)
- [16] **Y. Kim** and M. Kim, SCORE: a Scalable Concolic Testing Tool for Reliable Embedded Software, ACM SIGSOFT Foundation of Software Engineering (FSE) Tool demonstration track, pages 420-423, Szeged, Hungary, Sep 5-9 2011 (acceptance ratio: 47%)
- [17] **Y. Kim**, M. Kim and Y. Jang, Concolic Testing on Embedded Software - Case Studies on Mobile Platform Programs, ACM SIGSOFT Foundation of Software Engineering (FSE) Industrial track, Sep 5-9, 2011
- [18] Z. Xu, **Y. Kim**, M. Kim, G. Rothermel, and M. Cohen, Directed Test Suite Augmentation: Techniques and Tradeoffs, ACM SIGSOFT Foundation of Software Engineering (FSE), pages 257-266, Santa Fe, New Mexico, USA, Nov 7-11 2010 (**acceptance ratio: 20%**)
- [19] **Y. Kim**, M. Kim, N. Dang, Scalable Distributed Concolic Testing: a Case Study on a Flash Storage Platform, Grand Challenge in Verified Software Track @ Intl. Conf. on Theoretical Aspects of Computing (ICTAC) (LNCS 6255), pages 199-213, Natal, Brazil, Sep 1-3 2010
- [20] M. Kim and **Y. Kim**, Concolic Testing of the Multi-sector Read Operation for Flash Memory File System, Grand Challenge in Verified Software Track @ Brazilian Symposium on Formal Methods (SBMF), pages 251-265, Gramado, Brazil, Aug 19-21 2009 (LNCS 5902)
- [21] M. Kim, **Y. Kim** and H. Kim, Unit Testing of Flash Memory Device Driver through a SAT-based Model Checker, IEEE/ACM Automated Software Engineering (ASE), pages 198-207, L'Aquila, Italy, Sep 15-19 2008 (**acceptance ratio: 12%**)
- [22] M. Kim, Y. Choi, **Y. Kim** and H. Kim, Formal Verification of a Flash Memory Device Driver - an Experience Report, Spin Workshop (LNCS 5156), pages 144-159, LA, USA, August 10-12 2008
- [23] M. Kim, **Y. Kim**, Y. Choi, and H. Kim, Pre-testing Flash Device Driver through Model Checking Techniques, IEEE Intl. Conf. on Software Testing, Verification and Validation (ICST), pages 475-484, Lillehammer, Norway, April 9-11 2008 (**acceptance ratio: 20%**)

- **Refereed Domestic Journal Articles**

- [1] K. Park, H. Song, J. Lee, G. Jo, **Y. Kim**, and M. Kim, Automatic Test Case Generation through Concolic Testing to Improve SW Quality of Defense Weapon System, Journal of KIISE: Software and Applications, Vol. 46, No. 9, Nov 2019
- [2] H. Kim, **Y. Kim**, and M. Kim, Improving Applicability and Usability of a Concolic Testing Tool CROWN, Journal of KIISE: Software and Applications, Vol. 45, No. 10, Oct 2018
- [3] H. Lim, **Y. Kim**, and M. Kim, Automated Capturing and Replaying Unit Inputs of C Programs from System Executions through Static and Dynamic Analysis, Journal of KIISE: Software and Applications, Vol. 45, No. 10, Oct 2018
- [4] **Y. Kim**, H. Kim, W. Yang, and M. Kim, A Comparative Study of C Program Mutation Tools for Effective Mutation Analysis: A Case Study of Proteum and Milu, Journal of KIISE: Software and Applications, Vol. 45, No. 4, Apr 2018
- [5] **Y. Kim** and M. Kim, Automated Unit-test Generation for Detecting Vulnerabilities of Android Kernel Modules, Journal of KIISE: Software and Applications, Vol. 44, No. 2, Feb 2017

- [6] Y. Jeon, **Y. Kim**, S. Hong, and M. Kim, Mutagen4J: Effective Mutant Generation Framework for Java Programs, Journal of KIISE: Software and Applications, Vol. 43, No. 9, Sep 2016
- [7] **Y. Kim**, T. Kim, M. Kim, H. Lee, H. Jang, and M. Park, Effective Integer Promotion Bug Detection Technique for Embedded Software, Journal of KIISE: Software and Applications, Journal of KIISE: Software and Applications, Vol. 43, No. 6, Jun 2016
- [8] S. Moon, **Y. Kim**, M. Kim, FEAST: An Enhanced Fault Localization Technique using Probability of Test Cases Executing Faults, Journal of KIISE: Software and Applications, Vol 40, No. 10, Oct 2013
- [9] **Y. Kim**, Y. Park, M. Kim, A Comparative Case Study on Static Program Analysis Tools, Journal of KIISE: Computing Practices and Letters, Vol. 19, No. 8, Aug 2013
- [10] **Y. Kim**, M. Kim, Y. Jang, CREST-BV: An Improved Concolic Testing Technique Supporting Bitwise Operations for Embedded Software, Journal of KIISE: Software and Applications, Vol. 40, No. 2, Feb 2013
- [11] Y. Kim, M. Kim, **Y. Kim**, E. Jung, Comparison of Search Strategies of KLEE Concolic Testing Tool, Vol. 18, No. 4, Apr 2012

• Refereed Domestic Conference Papers

- [1] K. Park, J. Lee, H. Song K. Cho, **Y. Kim**, and M. Kim, Concolic Testing to Improve SW Quality of Defense Weapon System, Korea Software Congress (KSC), Dec 19-21, 2018 (**Distinguished best paper award**)
- [2] S. Hong, **Y. Kim**, M. Kim, S. Yoon, H. Jung, and S. Park, AtomicitySanitizer: Effective Runtime Atomicity Violation Detector for Multithreaded C Programs, Korea Software Congress (KSC), Dec 19-21, 2018
- [3] A. Lee, H. Kim, **Y. Kim**, and M. Kim, Improvement of Concolic Testing Effectiveness by Supporting Bitfield Symbolic Variable, Korea Computer Congress (KCC), Jun 20-22, 2018
- [4] H. Kim, **Y. Kim**, and M. Kim, Improving Applicability and User Interface of CREST, Korea Conference on Software Engineering (KCSE), Jan 29-31, 2018 (**Best paper award**)
- [5] H. Lim, **Y. Kim**, and M. Kim, Automated Capturing & Replaying Dynamic Unit Inputs of C Programs from System Executions, Korea Software Congress (KSC), Dec 20-22, 2017 (**Distinguished best paper award**)
- [6] P. Loc, B. Ko, **Y. Kim**, and M. Kim, COMUT: A Configurable Mutant Generation Tool for C programs for effective and efficient mutation analysis, Korea Software Congress (KSC), Dec 20-22, 2017 (**Best paper award**)
- [7] **Y. Kim**, H. Kim, W. Yang, and M. Kim, A Comparative Study of C Program Mutation Tools for Effective Mutation Analysis: A Case Study of Proteum and Milu, Winter Korea Computer Congress (KCC), Dec 21-23, 2016 (**Best paper award**)
- [8] **Y. Kim** and M. Kim, Automated Unit-test Generation for Detecting Vulnerabilities of Android Kernel Modules, Korea Computer Congress (KCC), Jun 29-Jul 1, 2016 (**Best paper award**)
- [9] **Y. Kim** and M. Kim, Efficient Dynamic Symbolic Execution Search Strategy using Input Coverage, Korea Conference on Software Engineering (KCSE), Jan 27-29, 2016 (**Best short paper award**)
- [10] Y. Park, **Y. Kim**, J. Cho, and M. Kim, Effective Concolic Testing using a Symbolic Library, Korea Conference on Software Engineering (KCSE), Feb 12-14, 2014 (**Best short paper award**)
- [11] S. Mun, **Y. Kim**, M. Kim, Improved Fault Localization Technique using Weighted Test Cases, Korea Conference on Software Engineering (KCSE), Jan 30 - Feb 1, 2013 (**Distinguished best short paper award**)
- [12] Y. Park, **Y. Kim**, and M. Kim, A Comparative Study of Static Analysis Tools: A Case Study on libexif by Using Coverity and Sparrow, Korea Computer Congress (KCC), Nov 23-24, 2012

- [13] **Y. Kim**, M. Kim, and Y. Jang, CREST-BV: An Improved Concolic Testing Technique with Bitwise Operations Support for Embedded Software, Korea Computer Congress(KCC), June 27-19, 2012 (**Best paper award**)
- [14] Duc Bui Hoang, **Y. Kim**, and M. Kim, A Case Study of the Application of Dynamic Symbolic Execution to Real-World Binary Programs, Korea Conference on Software Engineering (KCSE), Feb 8-10, 2012
- [15] Y. Kim, **Y. Kim**, and M. Kim, Case Study on Testing with KLEE Concolic Testing Tool, Korea Computer Congress (KCC), Nov 25-26, 2011
- [16] **Y. Kim** and M. Kim, Comparison of Test-case Generation Techniques based on Dynamic Symbolic Execution and Genetic Algorithm, Korea Conference on Software Engineering (KCSE), Feb 9-11, 2011
- [17] **Y. Kim** and M. Kim, Comparison of Test-case Generation Tools based on Dynamic Symbolic Execution, Koea Conference on Software Engineering (KCSE), Feb 8-10, 2010
- [18] M. Kim, **Y. Kim**, and H. Kim, Formal Verification of a Flash Memory Device Driver through Model Checking, Korea Conference on Software Engineering (KCSE), Feb 20-22, 2008

Awards and Scholarships

1. IEEE Conference on Software Testing, Validation and Verification (ICST), **Distinguished paper award**, 2018
 - **Y. Kim**, S. Hong, B. Ko, L. Phan and M. Kim, Invasive Software Testing: Mutating Target Programs to Diversify Test Exploration for High Test Coverage
2. Korea Software Congress (KSC), **Distinguished best paper award**, 2018
 - K. Park, J. Lee, H. Song K. Cho, **Y. Kim**, and M. Kim, Concolic Testing to Improve SW Quality of Defense Weapon System
3. Korea Conference on Software Engineering (KCSE), **Best paper award**, 2018
 - H. Kim, Y. Kim, and M. Kim, Improving Applicability and User Interface of CREST
4. Korea Software Congress (KSC), **Distinguished best paper award**, 2017
 - H. Lim, **Y. Kim**, and M. Kim, Automated Capturing & Replaying Dynamic Unit Inputs of C Programs from System Executions
5. Korea Software Congress (KSC), **Best paper award**, 2017
 - D. Phan, B. Ko, **Y. Kim**, and M. Kim, COMUT: A Configurable Mutant Generation Tool for C programs for effective and efficient mutation analysis
6. KAIST School of Computing, **Best Ph.D's dissertation award**, 2017
 - **Y. Kim**, Automated Unit Test Generation with Realistic Unit Context Synthesis for Low False Alarms
7. Samsung HumanTech Paper Competition, **Bronze award**, 2017
 - **Y. Kim**, Effective Concolic Unit Testing based on Realistic Unit Context Synthesis
8. Winter Korea Computer Congress (KCC), **Best paper award**, 2016
 - **Y. Kim**, H. Kim, W. Yang, and M. Kim, A Comparative Study of C Program Mutation Tools for Effective Mutation Analysis: A Case Study of Proteum and Milu
9. Korea Computer Congress (KCC), **Best paper award**, 2016
 - **Y. Kim** and M. Kim, Automated Unit-test Generation for Detecting Vulnerabilities of Android Kernel Modules
10. Korean Institute of Information Scientists and Engineers, 35th Student Research Paper Competition (graduate student track), **Second prize**, Apr 2016
 - **Y. Kim**, Effective Automated Concolic Unit Testing based on Realistic Unit Context Synthesis
11. Korea Conference on Software Engineering (KCSE), **Best paper award (short paper)**, 2016
 - **Y. Kim**, and M. Kim, Efficient Dynamic Symbolic Execution Search Strategy using Input Coverage
12. Samsung HumanTech Paper Competition, **Bronze award**, 2014
 - S. Moon and **Y. Kim**, MUSE: Precise Fault Localization based on Program Mutants
13. Korea Conference on Software Engineering (KCSE), **Best paper award (short paper)**, 2014
 - Y. Park, **Y. Kim**, J. Cho, and M. Kim, Effective Concolic Testing using a Symbolic Library

14. Korean Institute of Information Scientists and Engineers, 32nd Student Research Paper Competition (graduate student track), **Grand prize**, Apr 2013
 - S. Moon and **Y. Kim**, FEAST: Coverage-based Fault Localization with Fault Weights on Test Cases
15. Korea Computer Congress (KCC), **Best paper award**, 2012
 - **Y. Kim**, M. Kim, and Y. Jang, CREST-BV: An Improved Concolic Testing Technique with Bitwise Operations Support for Embedded Software
16. Korea Presidential Science Scholarship, Mar 2003 to Feb 2007

Research Funding

- **Funding from Korean Government Agencies**

1. Automated SW Testing using AI-based Mutation Analysis for High Bug Detection Ability, **Individual Basic Science & Engineering Research Program supported by national Research Foundation of Korea (NRF)**, 2020-2023 (KRW 300,000,000)
2. Automated SW Unit Testing based on Unit Correlation Metrics for Accurate Bug Detection, **Individual Basic Science & Engineering Research Program supported by national Research Foundation of Korea (NRF)**, 2017-2020 (KRW 150,000,000)

Patents

1. Co-inventor, Application No. 10-2018-0052680 in Korea, Testing Method and Apparatus of Target Function Included in Target Program, May 2018
2. Co-inventor, Application No. 10-2018-0053140 in Korea, Testing Method and Apparatus of Target Program using Mutated Program, May 2018
3. Co-inventor, Application No. 10-2018-0053145 in Korea, Generating Method and Apparatus of Mutant Programs, Which is Flexible and Highly Scalable, May 2018
4. Co-inventor, Patent No. 10-1227024-0000 in Korea, Method of distributed scalable concolic testing for software reliability, Jan 2010

Software

1. CROWN: Concolic testing for Real-wOrld softWare aNalysis (C++ and OCaml)
 - CROWN is a lightweight instrumentation-based concolic testing tool to generate concrete test inputs for real-world C programs. It supports complex C features such as bitwise operators, floating point arithmetic, bitfields and so on. SW. CROWN consists of 13,060 LoC (12,349 lines of C++ and 711 lines of Ocaml code).
 - Available upon request
2. MUSIC: MUtation analySIs tool with High Configurability and Extensibility (C++)
 - MUSIC is an open-source mutation analysis tool for C/C++ programs. MUSIC supports high configurability which allows a user to specify code locations to be mutated and mutation operators to be applied as well as control the number of generated mutants. In addition, MUSIC is highly extensible so that users can implement their own mutation operator in 10~20 LoC C++ code. MUSIC consists of 30,350 lines of C++ code.
 - Available at <https://github.com/swtv-kaist/MUSIC>
3. SCORE: Scalable COncolic testing for REliable Software (C++)
 - SCORE is a distributed concolic testing for C programs. It aims to generate a large number of test cases very fast, each of which explores a unique execution path of a target C program. SCORE implements a distributed concolic

testing algorithm that can utilize a large number of computing nodes in a scalable manner with low communication overhead among distributed nodes. SCORE consists of 2,817 lines of C++ code.

- Available at <http://swtv.kaist.ac.kr/tools/score>

Industry Engineer Training

1. Concolic Testing Theory and Practice with CROWN, Hyundai Mobis, May 3, 2019

-I taught the theory of automated testing techniques focusing on concolic testing and industrial case studies of concolic testing for detecting bugs in embedded SW. Also, I give a tutorial on using a concolic testing tool, CROWN, to achieve high test coverage.

2. SW Engineer Certification for C++, Hyundai Mobis and Korea Productivity Center, Aug 1 - Oct 31, 2018

- With Korea Productivity Center, I made evaluation criteria and three programming exams for the certification of C++ programming skills of SW engineers in Hyundai Mobis. Also, I gave concrete feedbacks to examinees to improve their C++ skills further.

3. Tutorial of Clang/LLVM C/C++ Front-End Framework, Hyundai Mobis, Nov 14-15, 2017

- I gave 2-days intensive course of program analysis using Clang/LLVM C/C++ analysis framework. I taught the basic of static/dynamic program analysis, compiler theory focusing on the front-end, and the Clang/LLVM framework.

4. Automated Software Testing for Software Design Engineer in Test (SDET), LG Electronics, Jun 29-30, 2017

- I gave 2-days intensive course of the automated software testing and formal verification techniques. I taught the theory and practice of automated testing techniques focusing on concolic testing and software model checking techniques.

Professional Activities

• **International Conference Organizer**

1. Proceedings chair of IEEE/ACM International Conference on Automation of Software Test (AST) 2020

• **International Conference Program Committees**

1. International Conference on Software Engineering (ICSE) Software Engineering in Practice (SEIP) track, 2020
2. International Conference on Software Engineering (ICSE) New Ideas and Emerging Results (NIER) track, 2020
3. International Conference on Program Comprehension (ICPC) Tool Demonstration Track, 2020
4. Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE), 2019
5. International Conference on Software Testing, Verification and Validation (ICST), 2018-2019
6. International Conference on Formal Aspects of Component Software (FACS), 2018
7. Asia-Pacific Software Engineering Conference (APSEC), 2018

• **International Journal Reviewers**

1. ACM Transactions on Software Engineering and Methodology (TOSEM), 2019
2. IEEE Transactions on Software Engineering (TSE), 2019-2020
3. Journal of Software: Testing, Verification and Reliability (STVR), 2019

4. Information and Software Technology (IST), 2018

• **International Conference Sub-Reviewers**

1. International Conference on Software Engineering (ICSE), 2014-2017, 2019
2. International Symposium on Software Testing and Analysis (ISSTA), 2014, 2017-2018
3. IEEE/ACM Automated Software Engineering (ASE), 2017
4. International Conference on Software Testing, Verification and Validation (ICST), 2015-2017
5. IEEE International Symposium on Software Reliability Engineering (ISSRE), 2014-2017
6. Automated Technology for Verification and Analysis (ATVA), 2007, 2009, 2013-2014, 2016
7. Asia-Pacific Software Engineering Conference (APSEC), 2014
8. Verified Software: Theories, Tools and Experiments, (VSTTE) 2014
9. IEEE/ACM Automated Software Engineering (ASE) tool track, 2009, 2013
10. International Conference on Quality Software (QSIC), 2011
11. IEEE Computer Software and Applications Conference (COMPSAC), 2010
12. International Conference on Runtime Verification (RV), 2009
13. Adaptive and Reconfigurable Embedded Systems (APRES), 2008

Invited Talks

• **International Invited Talks**

1. Target-driven Compositional Concolic Testing, Workshop on Testing, Analysis, and Verification of Cyber-Physical Systems and Internet of Things (TAV-CPS/IoT), Beijing, China, Jul 15, 2019
2. Directed Compositional Concolic Testing with Function Summary Refinement for Crashing Input Generation, Workshop on Finding Software Errors before They Find You, Stellenbosch, South Africa, Oct 15, 2018

• **Domestic Invited Talks**

1. Concolic Unit Testing, National Security Research Institute, Mar 9, 2020
2. Concolic Unit Testing, Korea Conference on Software Engineering Tutorial, Feb 3, 2020
3. Precise Learn-to-Rank Fault Localization using Dynamic and Static Features of Target Programs, The 3rd Workshop on Intelligent Automation Techniques for Verification and Debugging of Fullstack Multilingual Software, Jan 30, 2020
4. Directed Compositional Concolic Testing with Function Summary Refinement for Crashing Input Generation, The 2nd Workshop on Intelligent Automation Techniques for Verification and Debugging of Fullstack Multilingual Software, Jan 19, 2019
5. Precise Concolic Unit Testing of C Programs with Alarm Filtering Using Symbolic Calling Contexts, KIISE SIGPL Winter School, Feb 19, 2018
6. CONBRIO: Precise Automated Unit Testing, The 1st Workshop on Intelligent Automation Techniques for Verification and Debugging of Fullstack Multilingual Software, Jan 17, 2018
7. Automated Unit Test Generation with Realistic Unit Context Synthesis for Low False Alarms, The 2nd Winter School on Secure Software Development, Jan 27 – Jan 29, 2016
8. Effect Dynamic Symbolic Execution Search Strategy using Input Coverage, The 1st Winter School on Secure Software Development, Feb 2 – Feb 4, 2015

9. SAT-based Bounded Software Model Checking for Industrial Embedded Software: A Case Study, The 2nd Workshop on Secure Software Development, Feb 20 – 22, 2014
10. Automated Unit Testing of Large Industrial Embedded Software using Concolic Testing, The 1st Workshop on Secure Software Development, Nov 22, 2013
11. Automated Unit Testing of Large-Scale Embedded SW using Concolic Testing, The 9th Workshop on Software Analysis for Error-free Computing, 2013
12. Hybrid Approach of Genetic Algorithm and Dynamic Symbolic Execution for Effect Test Generation, The 6th Workshop on Software Analysis for Error-free Computing, Jun 25 – Jun 27, 2011
13. Concolic Testing for Real-world Application: A Case Study on S Company, The 5th Workshop on Software Analysis for Error-free Computing, Jan 7 – Jan 9, 2011
14. Smart Phone SW Testing using Concolic Testing: A Case Study on Busybox, The 4th Workshop on Software Analysis for Error-free Computing, Aug 25 – Aug 28, 2010
15. Comparison between SAT and SMT as a Software Analysis Engine, The 2nd Workshop on Software Analysis for Error-free Computing, Jul 9 - Jul 11, 2009
16. Comparison between SAT and SMT as a Software Analysis Engine, The 1st Workshop on Software Analysis for Error-free Computing, Nov 21 - Nov 22, 2008

Research Experiences

• Principal Investigator for Government-Funded Project

1. Principal investigator, Automated SW Testing using AI-based Mutation Analysis for High Bug Detection Ability, National Research Foundation of Korea (NRF), 2020-Present
2. Principal investigator, Automated SW Unit Testing based on Unit Correlation Metrics for Accurate Bug Detection, National Research Foundation of Korea (NRF), 2017-2020

• Researcher for Government-Funded Projects

1. Researcher, Dynamic and Static SW Data-driven Automated SW Fault Detection and Localization, National Research Foundation of Korea (NRF), 2019-Present
2. Researcher, Intelligent Automation Techniques for Verification and Debugging of Fullstack Multilingual Software, National Research Foundation of Korea (NRF), 2017-Present
3. Researcher, Automated SW Testing and Debugging Techniques for Improving SW Quality, National Research Foundation of Korea (NRF), 2017-2019

• Researcher for Industry-Funded Projects

1. Researcher, Application of Concolic Testing for SW Testing Productivity Improvement, Mando, 2019
2. Researcher, Dynamic Concurrency Bug Detector for Windows Multi-threaded Programs, Samsung Electronics, 2018
3. Researcher, Requirement-based Dynamic Reliability Test Development, LIG Nex1, 2018
4. Researcher, Effective Fault Localization using Mutation Analysis, Samsung Electronics, 2017-Present
5. Researcher, Automated Dynamic Test Case Generation using Path Search Algorithms, Hyundai Mobis, 2017-2019

- **Research Assistant for Government-Funded Projects**

1. Research assistant, Automated SW Testing and Debugging Techniques for Improving SW Quality, National Research Foundation of Korea (NRF), 2016-2017
2. Research assistant, Behavioural Coverage for Effective Software Testing, National Research Foundation of Korea (NRF), 2014-2016 (Joint Research Project with Prof. Willem Visser (ACM Distinguished Member, PC co-chair of ICSE 2016) at Stellenbosch University, South Africa)
3. Research assistant, Development of Mobile S/W Security Testing Tools for Detecting New Vulnerabilities of Android, Institute for Information and communications Technology Promotion (IITP), 2013-2016
4. Research assistant, Research and Development of Dual Operating System Architecture with High-Reliable RTOS and High-Performance OS, Institute for Information and communications Technology Promotion (IITP), 2012-2017
5. Research assistant, Practical Dynamic Symbolic Execution for Binary Programs, National Security Research Institute (NSRI), 2011
6. Research assistant, Improved Automated Test Case Generation through Parallelized Concolic Testing Technique, National Research Foundation of Korea (NRF), 2010-2011
7. Research assistant, Improving Embedded Software Reliability using SAT Solver, National Research Foundation of Korea (NRF), 2008-2009

- **Research Assistant for Industry-Funded Project**

1. Research assistant, Effective Fault Localization using Mutation Analysis, Samsung Electronics, 2015-2017
2. Research assistant, Applying Concolic Testing for Improving Reliability of Automotive Software, Hyundai Motor Company, 2015-2017
3. Research assistant, Applying Concolic Testing for Embedded Software Testing Automation, LG Electronics, 2013-2015
4. Research assistant, Automated Unit Test Generation Framework by using Concolic Testing, Samsung Electronics, 2010-2014
5. Research assistant, Formal Verification of Flash Memory Device Driver, Samsung Electronics, 2007-2008

Research Visits

- **International Research Visits**

1. Prof. Willem Visser (ACM Distinguished Member, PC co-chair of ICSE 2016) Stellenbosch University, Jun 13 – Jun 18, 2016
 - Discussing and developing the optimization techniques for faster input counting for symbolic execution paths based on the compiler optimization techniques such as constant propagation and variable elimination.
2. Prof. Willem Visser, Stellenbosch University, Apr 18 – Apr 25, 2015
 - Discussing and developing the effective concolic search strategy using input counting of symbolic execution paths. The core idea was to guide concolic search to cover the execution paths with fewer inputs first to detect hard-to-find bugs.
3. Prof. Gregg Rothmel (ACM Distinguished Scientist, General Chair of ICSE 2020), University of Nebraska-Lincoln, Nov 9 – Nov 24, 2011
 - Discussing and developing a hybrid approach of concolic testing and genetic algorithm (GA) to achieve high test coverage. The core idea was to identify which test requirement is easy-to-achieve or hard-to-achieve for

concolic testing or GA and to select/apply one of them to achieve a given test requirement faster

Extra Activities

- President of GON Hacking/Security Group of KAIST (2005)
 - ◆ <http://gon.kaist.ac.kr/>