LI HUANG

◇ Webpage: huangl223.github.io/li
◇ Email: li.huang@constructor.org

RESEARCH INTERESTS

- Software engineering, formal methods, AI4SE
- Software testing, automated test generation
- Automatic program repair

TEACHING INTERESTS

- Software engineering, software verification, software testing, formal methods

EDUCATION

- PhD, Software Engineering
  - Constructor Institute of Technology & Constructor University
  - November 2020 - June 2025
  - Thesis: *Straddling the Border between Tests and Proofs*
  - Advisor: Prof. Bertrand Meyer
- Master, Software Engineering
  - School of Data and Computer Science, Sun Yat-sen University
  - September 2017 - July 2019
  - Thesis: *Tool Supported Verification of (Non)-functional Requirements in Cyber Physical Systems Using Simulink Design Verifier*
  - GPA: 94/100
  - Advisors: Prof. Eun-Young Kang, Prof. Zibin Zheng
- Bachelor, Software Engineering
  - School of Data and Computer Science, Sun Yat-sen University
  - September 2013 - July 2017
  - GPA: 3.9/5.0
  - Bachelor's Thesis: *Tool Supported Verification and Validation of Automotive Systems*
  - Thesis Advisor: Prof. Eun-Young Kang

PROFESSIONAL EXPERIENCE

- Postdoc researcher, Chair of Software Engineering, Constructor Institute of Technology
  - January 2025 - Present
  - Research on combining formal verification techniques to improve the quality of AI-generated code.
- Lecturer, School of Electronic and Information Engineering, Beibu Gulf University
  - July 2020 - December 2020

    - Teaching courses in software engineering and computer science, covering a range of topics such as programming languages and software architecture.
- Research Engineer, China CEPREI Research Institute
    - March 2020 - July 2020
    - Verification of safety and security properties for Vehicular Ad-hoc Networks (VANET)
    - Testing of security properties for embedded integrated circuits.
- Research Assistant, Sun Yat-sen University
    - July 2019 - February 2020
    - Formal verification and validation of cyber physical systems (CPS).
    - Formal analysis of functional and timing constraints of CPS using stochastic Bip[1].

TEACHING EXPERIENCE

- *Algorithms and Data Structures*, Constructor Institute of Technology, lecturer, Spring 2025.
- *Software Construction, Software Architecture, and Software Engineering*, Constructor Institute of Technology, teaching assistant, Fall 2022, 2023, 2024.
- *Advances in Software Engineering*, Constructor Institute of Technology, teaching assistant, Fall 2023, 2024.
- *Introduction to C++ Programming*, Beibu Gulf University, School of Electronics and Information, lecturer, Fall 2020
- *Fundamentals of Computer Systems*, Beibu Gulf University, School of Chemistry, School of Electronics and Information, lecturer, Fall 2020
- *Computer Language and Implementation*, Sun Yat-sen University, School of Data and Computer Science, Software Engineering Institute, teaching assistant, Spring 2017, 2018.
- *Computer Language and Implementation*, Sun Yat-sen University, School of Data and Computer Science, Software Engineering Institute, teaching assistant, Spring 2017, 2018.
- *Introduction to Cyber-Physical Systems: Intelligent Vehicle Software Design*, Sun Yat-sen University, School of Data and Computer Science, Software Engineering Institute, teaching assistant, Spring 2017, 2018.
- *Introduction to Real-Time Systems*, Sun Yat-sen University, School of Data and Computer Science, Software Engineering Institute, teaching assistant, Fall 2017, 2018.

STUDENT SUPERVISION

- Mohit Kumar Basak, Master thesis, *Optimization of test cases generated by Proof2Test for simple test cases in Eiffel*, Chair of Software Engineering, Constructor Institute of Technology.
- Wenchen Lai, Bachelor thesis, *Formal analysis of CPS using Uppaal-SMC*, Sun Yat-sen University, School of Data and Computer Science, Software Engineering Institute.

AWARD

- IEEE Real-Time Systems Symposium (RTSS), Hong Kong, Student Travel Grant (2019).
- European Joint Conferences on Theory and Practice of Software (ETAPS), Prague, Czech Republic, Student Scholarship (2019)

[1] http://www-verimag.imag.fr/Rigorous-Design-of-Component-Based.html

- Formal Methods in Computer-Aided Design (FMCAD), University of Texas, Austin, USA, Student Forum Travel Award (2018)
- Chinese National Endeavor Scholarship (3 times, 2013 - 2016)
  - Awarded by Chinese Government (top 5%)
- Excellent Student Scholarship (3 times, top 20% based on GPA, 2013 - 2016)
- Excellent Team in Ke Teng Cup Software Creativity Competition (top 5 teams, 2014)

CONFERENCE PROGRAM COMMITTEES AND JOURNAL REVIEWER

- Program Committee, Int. Conference on Testing Software and Systems (ICTSS), 2025
- Program Committee, Int. Workshop on Frontiers in Software Engineering Education (FISEE), 2023
- Program Committee, Int. Workshop on Software Faults & Software Hardware Interaction Faults (IWSF & SHIFT), 2023, 2024, 2025
- Reviewer, Springer Nature Computer Science (SN-CS)

PUBLICATIONS

1. **Li Huang**, Ilgiz Mustafin, Marco Piccioni, Alessandro Schena, Reto Weber, and Bertrand Meyer. “Do AI models help produce verified bug fixes?.” arXiv preprint arXiv:2507.15822 (2025).
2. **Li Huang**, Bertrand Meyer, and Reto Weber. “Loop unrolling: formal definition and application to testing.” To appear at the *International Conference on Testing Software and Systems (ICTSS)*, available at arXiv:2502.15535, 2025.
3. **Li Huang**, Sophie Ebersold, Alexander Kogtenkov, Alexandr Naumchev, Bertrand Meyer, Yinling Liu, Aliyu Alege. “Lessons from Formally Verified Deployed Software Systems.”, To appear at *ACM Computing Surveys*, available at arxiv.org/abs/2301.02206, 2025.
4. **Li Huang**, Bertrand Meyer, Manuel Oriol. “Seeding Contradiction: a Fast Method for Generating Full-coverage Test Suites”. *Special Issue: Testing Software and Systems: theory and applications, Springer Nature Computer Science (SNCS)*, 6(1), 41, 2024
5. **Li Huang**, Bertrand Meyer, Manuel Oriol. ”Is MCDC Really Better? Lessons from Combining Tests and Proofs”. In the *International Conference on Tests and Proofs (TAP)*, pp. 25-44. 2024.
6. **Li Huang**, Bertrand Meyer, Ilgiz Mustafin, Manuel Oriol. “Execution-Free Program Repair”. In the *Companion Proceedings of the International Conference on the Foundations of Software Engineering (FSE-IVR)*, 2024.
7. Bertrand Meyer, Viktoryia Kananchuk, **Li Huang**. “BUGFIX: towards a common language and framework for the Automatic Program Repair community”. In the *International Workshop on Automated Program Repair*, pp. 9-13. 2024.
8. **Li Huang**, Bertrand Meyer. “A Failed Proof Can Yield a Useful Test.” *Software Testing, Verification and Reliability (STVR)*, 33(7), 2023.
9. **Li Huang**, Bertrand Meyer, Manuel Oriol. “Seeding Contradiction: a Fast Method for Generating Full-coverage Test Suites.” In *IFIP International Conference on Testing Software and Systems (ICTSS)*, pp. 52-70, 2023

10. **Li Huang**, Bertrand Meyer and Manuel Oriol. "Improving Counterexample Quality from Failed Program Verification." In the *International Symposium on Software Reliability Engineering Workshop (ISSRE-W)*, Charlotte, North Carolina, USA, 2022.

11. **Li Huang** and Eun-Young Kang. "Work-In-Progress: Formal Analysis of Hybrid-Dynamic Timing Behaviors in Cyber-Physical Systems." In the *The IEEE Real-Time Systems Symposium-Brief Presentation (RTSS-BP)*, Hong Kong, China, December, 2019.

12. **Li Huang**, Tian Liang and Eun-Young Kang. "Formal Verification of Dynamic and Stochastic Behaviors for Automotive Systems." In the *International Conference on Engineering of Complex Computer Systems (ICECCS)*, Guangzhou, China, November, 2019.

13. **Li Huang**, Tian Liang and Eun-Young Kang. "Tool-Supported Analysis of Dynamic and Stochastic Behaviors in Cyber-Physical Systems." In the *International Conference on Software Quality, Reliability, and Security (QRS)*, University of Sofia, Sofia, Bulgaria, July, 2019.

14. **Li Huang** and Eun-Young Kang. "Formal Verification of Safety & Security Related Timing Constraints for A Cooperative Automotive System." In the *European Joint Conferences on Theory and Practice of Software (ETAPS-FASE)*, pp. 210-227, Springer, Prague, Czech Republic, April, 2019.

15. **Li Huang** and Eun-Young Kang. "SMT-based Probabilistic Analysis of Timing Constraints in Cyber-Physical Systems." In the *Formal Methods in Computer-Aided Design (FMCAD) Student Forum*, University of Texas, Austin, USA, October, 2018.

16. Eun-Young Kang and **Li Huang**. "Probabilistic Analysis of Timing Constraints in Autonomous Automotive Systems using Simulink Design Verifier." In the *International Symposium on Dependable Software Engineering Theories, Tools and Applications (SETTA)*, pp. 170-186, Springer, Beijing, China, September 2018.

17. Eun-Young Kang, Dongrui Mu, and **Li Huang**. "Probabilistic Verification of Timing Constraints in Automotive Systems using UPPAAL-SMC." In the *International Conference on Integrated Formal Methods (IFM)*, pp. 236-254, Springer, Maynooth, Ireland, September 2018.

18. Eun-Young Kang, **Li Huang**, and Dongrui Mu. "Formal Verification of Energy and Timed Requirements for a Cooperative Automotive System." In the *ACM/SIGAPP Symposium On Applied Computing in Software Engineering (SAC)*, pp. 1492-1499, ACM, Pau, France, April 2018.

19. Eun-Young Kang, Dongrui Mu, **Li Huang**, and Qianqing Lan. "Verification and Validation of a Cyber-Physical System in the Automotive Domain." In *International Conference on Software Quality, Reliability and Security (QRS)*, pp. 326-333, IEEE, Prague, Czech Republic, July 2017.

20. Eun-Young Kang, Dongrui Mu, **Li Huang** and Qianqing Lan. "Model-Based Analysis of Timing and Energy Constraints in an Autonomous Vehicle System." In *International Conference on Software Quality, Reliability and Security (QRS)*, pp. 525-532, IEEE, Prague, Czech Republic, July 2017.

TECHNICAL PATENTS

- 034-22-US-001, *System and method for generating failing tests from failed proofs*, 2022
- 034-23-US-012, *Seeding contradiction as a fast method for generating full-coverage test suites*, 2023
- 034-23-US-018, *System and method for repairing computer programs automatically without execution*, 2024