

Zhongxing Yu

Research Interests

My general research interests are in programming language, software engineering, and machine learning, in particular I am excited about working at the intersection between them. The goal of my research is to tackle important, pressing, and difficult problems via principled, elegant, and practical solutions. My current research program has a particular focus on the areas of program synthesis and program repair, aiming to make programming easier and programs more robust.

Education

2010/9 - 2016/2 **Beihang University**, Beijing, China

Ph.D. Automation Science

Advisor, Prof. Chenggang Bai

Thesis: Research on Spectrum-based Software Fault Localization Techniques

2006/9 - 2010/7 **Beihang University**, Beijing, China

B.S. Automation Science

Thesis: Simulation of Aircraft Instrument

Work Experience

2018/3 - Now **KTH Royal Institute of Technology**, Sweden (*KTH is the largest and oldest technical university in Sweden, and one of the leading technical universities in Europe*)

Postdoctoral Research Fellow

Collaborator, Prof. Martin Monperrus

2016/8 - 2018/2 **Inria**, France (*Inria is a French national research institution focusing on computer science and applied mathematics, and is among the top research institutes on computer science worldwide*)

Postdoctoral Research Fellow

Collaborator, Prof. Martin Monperrus

Publications *Please click paper titles for linked PDFs*

In Submission

POPL 2020 **Zhongxing Yu**, Matias Martinez, Tegawende f.Bissyande, and Martin Monperrus. *Learning the Relation between Code Features and Code Transforms with Structured Prediction*, <https://arxiv.org/abs/1907.09282>, Under Review at 47th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2020), CCF-A.

TOSEM Thomas Durieux, **Zhongxing Yu**, Youssef Hamadi, Martin Monperrus. *Production-Driven Patch Generation and Validation*, Major Revision at ACM Transactions on Software Engineering and Methodology (TOSEM), CCF-A.

JSS Benjamin Danglot, Oscar Vera-Perez, **Zhongxing Yu**, Andy Zaidman, Martin Monperrus, and Benoit Baudry. *A Snowballing Literature Study on Test Amplification*, <https://arxiv.org/abs/1705.10692>, Minor Revision at Journal of Systems and Software (JSS), CCF-B.

Refereed Conference/Journal Papers

- TSE 2019 **Zhongxing Yu**, Chenggang Bai, Lionel Seinturier, and Martin Monperrus. *Characterizing the Usage, Evolution, and Impact of Java Annotations in Practice*, in IEEE Transactions on Software Engineering (TSE), Online First, April 2019, CCF-A.
- EmSE 2019 **Zhongxing Yu**, Matias Martinez, Benjamin Danglot, Thomas Durieux, and Martin Monperrus. *Alleviating Patch Overfitting with Automatic Test Generation: A Study of Feasibility and Effectiveness for the Nopol Repair System*, in Empirical Software Engineering Journal (EmSE), vol. 24, no. 1, pp. 33–67, 2019-02, *Invited to ICSE 2019 as part of the Journal First Paper Track*, CCF-B.
- ICST 2018 Thomas Durieux, Youssef Hamadi, **Zhongxing Yu**, Benoit Baudry, and Martin Monperrus. *Exhaustive Exploration of the Failure-oblivious Computing Search Space*, in Proceedings of the 11th IEEE Conference on Software Testing, Validation and Verification (ICST'18), pp. 139–149, Vasteras, Sweden, CCF-C, (acceptance rate 25.2%=30/119).
- ICSE 2018-SEIP Simon Urli, **Zhongxing Yu**, Lionel Seinturier, and Martin Monperrus. *How to Design a Program Repair Bot? Insights from the Repairator Project*, in Proceedings of the 40th ACM/IEEE International Conference on Software Engineering, Track Software Engineering in Practice (ICSE-SEIP'18), pp. 95–104, Gothenburg, Sweden, CCF-A, (acceptance rate 23.6%=31/131).
- ICSE 2015 **Zhongxing Yu**, Chenggang Bai, and Kai-Yuan Cai. *Does the failing test execute a single or multiple faults? An approach to classifying failing tests*, in Proceedings of the 37th International Conference on Software Engineering (ICSE'15), pp. 924–935, Florence, Italy, CCF-A, (acceptance rate 18.5%=84/452).
- IST 2013 **Zhongxing Yu**, Chenggang Bai, and Kai-Yuan Cai. *Mutation-oriented test data augmentation for GUI software fault localization*, in Information and Software Technology Journal (IST), vol. 55, no. 12, pp. 2076–2098, 2013-12, CCF-B.
- QSIC 2012 Jing Feng, Bei-Bei Yin, Kai-Yuan Cai and **Zhongxing Yu**. *3-Way GUI Test Cases Generation Based on Event-Wise Partitioning*, in Proceedings of the 12th International Conference on Quality Software (QSIC'12, currently known as QRS), pp. 89–97, Xi'an, China, CCF-C, (acceptance rate 17.6%=13/74).
- HASE 2011 **Zhongxing Yu**, Hai Hu, Chenggang Bai, Kai-Yuan Cai, and W. Eric Wong. *GUI Software Fault Localization Using N-gram Analysis*, in Proceedings of the IEEE 13th International Symposium on High-Assurance Systems Engineering (HASE'11), pp. 325–332, Boca Raton, FL, USA.

Technical/Project Report

Zhongxing Yu, Matias Martinez, Benjamin Danglot, Thomas Durieux, and Martin Monperrus. *Test Case Generation for Program Repair: A Study of Feasibility and Effectiveness*, 2017.

Thomas Durieux, Benjamin Danglot, **Zhongxing Yu**, Matias Martinez, Simon Urli, Martin Monperrus. *The patches of the Nopol automatic repair system on the bugs of Defects4J version 1.1.0*, 2017.

Research Experience

Projects as the Lead Contributor

- 2018/3 - Now
(at KTH) **Structural Code Transform Prediction.** Designed and implemented the first approach for structurally predicting code transforms at the level of AST nodes using conditional random fields (work under review at POPL 2020). The approach captures the probability distribution over the source code edit space during program evolution by learning from “Big Code”, and has important implications for both program synthesis and program repair. The methodology underlying this work can be extended for other programming language and software engineering tasks, and this is my favourite work up to now.

- 2017/8-2018/9 (at Inria and KTH) **Large-scale Empirical Study about Java Annotations.** Conducted the first large-scale empirical study about Java annotations on 1,094 notable open-source projects hosted on GitHub (TSE 2019). The study generates 10 novel and important empirical findings about Java annotation usage, annotation evolution, and annotation impact, and the implications of these findings will benefit developers, researchers, tool builders, and language or library designers involved in annotation engineering.
- 2016/8-2017/8 (at Inria) **Defeating Overfitting in Program Repair.** Performed the first deep analysis of the overfitting problem in program repair and proposed an approach called UnsatGuided for alleviating the overfitting problem for synthesis-based repair techniques by making use of automatic test case generation techniques (EmSE 2019).
- 2013/4-2014/8 (at Beiahnng University) **Failing Test Classification.** Designed and implemented the first approach for classifying failing tests into single-fault-execution and multiple-fault-execution ones (ICSE 2015). By feeding only the identified single-fault-execution failing tests as inputs to debugging techniques (e.g., fault localization and program repair), the approach alleviates the unrealistic single-fault failure assumption typically adopted by these debugging techniques.
- 2010/10-2013/1 (at Beiahnng University) **GUI Software Fault Localization.** Designed and implemented techniques to conduct effective fault localization for GUI software, including one technique which uses n -gram analysis to do fault localization (HASE 2011) and another technique which augments existing tests with additional tests generated by mutation to improve fault localization (IST 2013).

Projects with Major Contributions

- 2017/3-2019/1 (at Inria and KTH) **Survey on Test Amplification.** Helped collect and analyze literature about test amplification, and contributed significantly to the organization and writing about the first comprehensive survey on test amplification (work subject to minor revision at JSS).
- 2017/6-2018/2 (at Inria) **Production-Driven Program Repair.** Helped design and implement Itzal, the first system for Java that performs patch generation for uncaught exceptions *in production* (work subject to major revision at TOSEM).
- 2017/6-2017/9 (at Inria) **Program Repair Bot.** Helped design and implement Repairnator, the first repair bot that constantly monitors test failures, reproduces bugs, and runs program repair tools against each reproduced bug (ICSE 2018-SEIP).
- 2017/3-2017/8 (at Inria) **Exploration of Failure-Oblivious Computing Behavior.** Helped design and conduct the first experiment that analyzes the size and diversity of the failure-oblivious computing behaviors (ICST 2018).

Skills

Languages	Java, C, C++, C#, Python, Assembly Language, JavaScript, Matlab, PHP
Tools & Libraries	Code Parser [Spoon, JavaParser], SMT Solver [Z3], AST Comparator [Gumtree, ChangeDistiller], Probabilistic Graphical Model [OpenMarkov, GRMM, PyStruct]
Systems	Linux, Windows, Unix, DOS, Mac

Languages

Mandarin	Native language
English	Fluent (speaking, reading, writing)
French	Beginner
Swedish	Beginner

Selected Talks

- ICSE'19-Journal First (preview) Alleviating Patch Overfitting with Automatic Test Generation: A Study of Feasibility and Effectiveness for the Nopol Repair System
2019/05/20, Stockholm, Sweden
- KTH Seminar Structural Code Transform Prediction
2019/04/10, Stockholm, Sweden
- KTH Seminar Characterizing the Usage, Evolution, and Impact of Java Annotations in Practice
2019/02/12, Stockholm, Sweden
- Inria Spirals Seminar Automatic Test Case Generation for Alleviating Overfitting Problem in Automatic Program Repair
2016/10/17, Lille, France
- ICSE'15 Does the Failing Test Execute a Single or Multiple Faults? An Approach to Classifying Failing Tests
2015/05/25, Florence, Italy
- QSIC'12 3-Way GUI Test Cases Generation Based on Event-Wise Partitioning
2012/08/28, Xi'an, China

Services

- Reviewer ACM Transactions on Software Engineering and Methodology (TOSEM): 2019
IET Software Journal: 2017, 2018
- Sub-Reviewer ACM Transactions on Software Engineering and Methodology (TOSEM): 2015, 2017, 2018
IEEE Transactions on Software Engineering (TSE): 2014, 2016, 2017
Empirical Software Engineering Journal (EmSE): 2016
Information and Software Technology Journal (IST): 2012, 2015
Journal of Systems and Software (JSS): 2016
ACM/IEEE International Conference on Software Engineering (ICSE): 2017
ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE): 2018
IEEE Annual International Computers, Software and Applications Conference (COMPSAC): 2012, 2015
IEEE International Conference on Quality Software (QSIC): 2013, 2014

Teaching Experiences

At Beihang University, I have been the teaching assistant for the following courses:
Software Reliability Engineering, Prof. Chenggang Bai, Winter 2013, Postgraduate Course.
Computer Software Technology, Prof. Chenggang Bai, Spring 2012, Undergraduate Course.

Awards and Honors

- 2018 Postgraduate Scholarship (KTH Royal Institute of Technology 2018-2020)
2016 Postgraduate Scholarship (INRIA 2016-2018)
2015 Research Travel Grant for ICSE 2015, Beihang University
2015 Outstanding Research Achievement Award, Beihang University
2014 Outstanding Research Achievement Award, Beihang University

References

Prof. Martin Monperrus

*KTH Royal Institute of Technology
Division of Theoretical Computer Science
School of Electrical Engineering and Computer Science
Lindstedtsvagen 3, 10044 Stockholm, Sweden
martin.monperrus@csc.kth.se
(+46) 08-790-65-40*

Prof. Chenggang Bai

*Beihang University
School of Automation Science
E919, New Main Building, No. 37 Xueyuan
Road, Haidian District, Beijing, P.R. China
bcg@buaa.edu.cn
(+86) 010-82339672*

Prof. Lionel Seinturier

*Inria Lille - Nord Europe/University of Lille
Faculty of Science and Technology
40 Avenue Bureau B318 Parc Scientifique de la
Haute Borne 59650 Villeneuve d'Ascq France
Lionel.Seinturier@inria.fr
(+33) (0)3-59-35-87-76*