Yuting Wang

汪宇霆

Shanghai Jiao Tong University
Room 201, IEEE Building No.1
No.800 Dongchuan Road, Minhang District
Shanghai 200240, China
Email: `yuting.wang@sjtu.edu.cn`
Website: `http://jhc.sjtu.edu.cn/~yutingwang`

I am a Tenure-Track Associate Professor at the John Hopcroft Center for Computer Science, Shanghai Jiao Tong University.

Research Interests

My research interests are broadly in the area of formal verification of software systems. Within this context, I am interested in developing programming, specification and reasoning formalisms, in constructing programming languages and frameworks that implement these formalisms and in applying the formalisms and their implementations to verify software artifacts.

Education

- *University of Minnesota, Twin Cities*, Minneapolis, MN, USA
  Ph.D. in Computer Science and Engineering, September 2011 - December 2016
  Thesis: *A Higher-Order Abstract Syntax Approach to the Verified Compilation of Functional Programs*
  Advisor: Gopalan Nadathur
- *University of Connecticut*, Storrs, CT, USA
  M.S. in Computer Science and Engineering, August 2011
  Thesis: *AMIBE: an Imperative Programming Language with First Class Continuations*
  Advisor: Laurent Michel
- *Shanghai Jiao Tong University*, Shanghai, China
  M.S. in Power System and its Automation, March 2009
- *Shanghai Jiao Tong University*, Shanghai, China
  B.S. in Electric Engineering and Automation, June 2006

Work Experience

- Tenure-Track Associate Professor at the John Hopcroft Center for Computer Science, Shanghai Jiao Tong University, 2020 - Present.
- Post doctoral researcher supervised by Zhong Shao on the NSF Expeditions-In-Computing project *DeepSpec* (`https://deepspec.org`), Yale University, 2016 - 2019. Work supervised by Zhong Shao.
- Research assistant on an NSF funded project entitled “Reasoning about Specifications of Computations”, University of Minnesota, 2011 - 2016. Work supervised by Gopalan Nadathur.

- Research internship, INRIA Saclay, Palaiseau, France, Summer 2014. Work supervised by Kaustuv Chaudhuri.
- Research internship, INRIA Saclay, Palaiseau, France, Summer 2012. Work supervised by Kaustuv Chaudhuri and Dale Miller.
- Research assistant on the Short-Term Load Forecasting project, University of Connecticut, 2009 - 2011. Work supervised by Laurent Michel and Peter Luh.

Teaching and Tutoring

- CS1604: Principles and Methods of Program Design, SJTU, Spring 2022, 2023.
- AI3606: Mathematical Logic, SJTU, Fall 2021, 2022, 2023, 20.
- CS171: Programming Principles and Methods, SJTU, Fall 2020.
- Teaching assistant for CSCI-4011: Formal Languages and Automata Theory, University of Minnesota, Fall 2015. Lecturer: Gopalan Nadathur.
- Tutoring: Dawn Michaelson (undergraduate at the U. of Minnesota, Fall 2016), Andrew Wu (undergraduate at the Davidson College, now Ph.D. student at Stanford University, Fall 2017).

Publications (*Corresponding authors are marked with asterisk*)

- Ling Zhang, Yuting Wang*, Jinhua Wu, Jérémie Koenig, and Zhong Shao. *Fully Composable and Adequate Verified Compilation with Direct Refinements between Open Modules*. Proceedings of the ACM on Programming Languages (PACMPL), 8(POPL), pages 72:1-72:31, 2024.
- Jinhua Wu, Yuting Wang*, Meng Sun, Xiangzhe Xu, and Yichen Song. *Towards a Framework for Developing Verified Assemblers for the ELF Format*. Proceedings of the 21st Asian Symposium on Programming Languages and Systems (APLAS), pages 205-223, 2023.
- Siyu Liu and Yuting Wang*. *Verified Transformation of Continuation-Passing Style into Static Single Assignment Form*. Proceedings of the 17th International Symposium on Theoretical Aspects of Software Engineering (TASE), pages 20-37, 2023.
- Yuting Wang, Ling Zhang, Zhong Shao, and Jérémie Koenig. *Verified Compilation of C Programs with a Nominal Memory Model*. Proceedings of the ACM on Programming Languages (PACMPL), 6(POPL), pages 25:1-25:31, 2022.
- Xiangzhe Xu, Jinhua Wu, Yuting Wang*, Zhenguo Yin, and Pengfei Li. *Automatic Generation and Validation of Instruction Encoders and Decoders*. Proceedings of the 33rd International Conference on Computer-Aided Verification (CAV), pages 728-751, 2021.
- Yuting Wang, Xiangzhe Xu, Pierre Wilke, and Zhong Shao. *CompCertELF: Verified Separate Compilation of C Programs into ELF Object Files*. Proceedings of the ACM on Programming Languages (PACMPL), 4(OOPSLA), pages 197:1-197:28, 2020.
- Yuting Wang, Pierre Wilke, and Zhong Shao. *An Abstract Stack Based Approach to Verified Compositional Compilation to Machine Code*. Proceedings of the ACM on Programming Languages (PACMPL), 3(POPL), pages 62:1-62:30, 2019.

- Gopalan Nadathur and Yuting Wang. *Schematic Polymorphism in the Abella Proof Assistant*. Proceedings of the 20th International Symposium on Principles and Practice of Declarative Programming (PPDP), pages 15:1-15:13, 2018.
- Yuting Wang and Gopalan Nadathur*. *A Higher-Order Abstract Syntax Approach to Verified Transformations on Functional Programs*. Proceedings of the 25th European Symposium on Programming (ESOP), pages 752-779, 2016.
- Yuting Wang and Kaustuv Chaudhuri. *A Proof-Theoretic Characterization of Independence in Type Theory*. Proceedings of the 13th International Conference on Typed Lambda Calculi and Applications (TLCA), pages 332-346, 2015.
- David Baelde, Kaustuv Chaudhuri, Andrew Gacek, Dale Miller, Gopalan Nadathur, Alwen Tiu, and Yuting Wang. *Abella: A System for Reasoning about Relational Specifications*. Journal of Formalized Reasoning, 7(2), 2014.
- Yuting Wang, Kaustuv Chaudhuri, Andrew Gacek, and Gopalan Nadathur. *Reasoning about Higher-Order Relational Specifications*. Proceedings of the 15th ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming (PPDP), pages 157-168, 2013.
- Yuting Wang and Gopalan Nadathur. *Towards Extracting Explicit Proofs from Totality Checking in Twelf*. Proceedings of the 8th ACM SIGPLAN International Workshop on Logical Frameworks and Metalanguages: Theory and Practice (LFMTP), pages 55-66, 2013.
- Che Guan, Peter Luh, Laurent Michel, Yuting Wang, and Peter Friedland. *Very Short-Term Load Forecasting: Wavelet Neural Networks with Data Pre-Filtering*. IEEE Transactions on Power Systems, 28(1):30-41, 2013.

Software

- *Abella*: `http://abella-prover.org`

  An interactive theorem-prover that is noteworthy for its support of higher-order abstract syntax and for the two-level logic approach to reasoning about formal specifications. This system represents joint work with other researchers at the University of Minnesota and at INRIA, Saclay, France. My contributions to the system were to build a complete treatment of an expressive specification logic called the logic of higher-order hereditary Harrop formulas, to co-develop a methodology for using this enhancement in complex reasoning tasks (collaboration with Kaustuv Chaudhuri) and to co-design and implement a form of polymorphism called *schematic polymorphism* to support polymorphic reasoning (collaboration with Gopalan Nadathur).
- *Stack-Aware CompCert*: `https://certikos.github.io/compcertmc/`

  The state-of-the-art verified compiler for C is CompCert, whose verified compilation chain outputs abstract assembly code in Coq. Further compilation to machine code is handled by an unverified external toolchain. Collaborating with Pierre Wilke and Zhong Shao, I designed and implemented an extension to CompCert that supports verified compilation to machine code. The key idea is to instrument the memory model of CompCert with an explicit notion of stack (hence making CompCert "Stack-Aware"), so that the stack consumption is preserved by compilation. This enables a finite stack in CompCert that, together with the finite code and data, can be mapped to the finite memory space on real machines.

Stack-Aware CompCert also supports a form of compositional compilation known as *contextual compilation* for composing verified abstraction layers. Contextual compilation provides the basis for compositionality of kernel modules in the verified operating system CertiKOS (`https://www.cs.yale.edu/flint/certikos/`).

- *AMIBE*: `http://digitalcommons.uconn.edu/gs_theses/142/`

  AMIBE is an imperative programming language that supports first class continuations that I co-designed with Laurent Michel. As part of this work, I implemented a compiler for AMIBE in C++. This system demonstrates how to develop efficient *constraint programming* languages by exploiting rich optimizations provided by modern compiler infrastructures such as LLVM.

- *VSTLF*: `https://github.com/ldmbouge/vstlf`

  Very Short Term Load Forecasting (VSTLF) is a system for forecasting electric power load in short terms (from minutes to hours) by using neural networks. I was the main programmer for the VSTLF project from 2009 to 2011. VSTLF is written in Java.

Talks

- Verified Compilation of C Programs with a Nominal Memory Model. POPL 2022. Online, January, 2022.
- CompCertELF: Verified Separate Compilation of C Programs into ELF Object Files. SPLASH-OOPSLA 2020. Online, November, 2020.
- Stack-Aware CompCert: Verified Compositional Compilation of C Programs into Machine Code. The DeepSpec @ PLDI 2019 workshop. Phoenix, USA, July 2019.
- Compilation Support for End-to-End Formal Verification of Software Systems. The 4th SJTU Future Information Technology International Forum for Young Scholars (SIFYS). Shanghai Jiao Tong University, Shanghai, China, June 2019.
- Stack-Aware CompCert: Verified Compilation of C Programs into Machine Code. Seminar in John Hopcroft Center. Shanghai Jiao Tong University, Shanghai, China, February 2019.
- Schematic Polymorphism in the Abella Proof Assistant. The 20th International Symposium on Principles and Practice of Declarative Programming. Frankfurt am Main, Germany, September 2018.
- A Framework for Verified Compilation of Functional Programs. Seminar in Prof. Zhong Shao's group (invited talk). Yale University, USA, March 2016.
- A Proof-Theoretic Characterization of Independence in Type Theory. The 13th International Conference on Typed Lambda Calculi and Applications. University of Warsaw, Poland, July 2015.
- Verified Transformations of Functional Programs. Midwest Verification Day 2014. University of Missouri, USA, October 2014.
- Verified Functional Program Transformations Using Higher-Order Abstract Syntax. Parsifal Seminar. INRIA Saclay, France, June 2014.

- Towards Extracting Explicit Proofs from Totality Checking in Twelf. The 8th ACM SIGPLAN International Workshop on Logical Frameworks and Metalanguages: Theory and Practice. Boston, USA, September 2013.
- The Abella Approach to Specifying and Reasoning about Formal Systems. Midwest Verification Day 2012. University of Kansas, USA, September 2012.
- New Developments with the Abella System. Workshop on Abella and Bedwyr. Ecole Polytechnique, France, July 2012.
- AMIBE: an Imperative Programming Language with First Class Continuations. Midwest Verification Day 2011. University of Minnesota, USA, September 2011.

Academic Service

- Program Committee:
  - Certified Programs and Proofs, 2023
  - International Conference on Formal Techniques for Distributed Objects, Components, and Systems, 2023
  - Logical Frameworks and Meta-Languages: Theory and Practice, 2018
- Artifact Evaluation Committee:
  - International Conference on Functional Programming (ICFP), 2019 and 2020

Awards and Affiliations

- ACM Member, since 2013.
- CCF Group on Formal Methods, since 2020
- Doctoral Dissertation Fellowship, Graduate School, University of Minnesota, 2014-15.
- Travel Award, NSF-INRIA grant entitled “Research Experience for US Students at INRIA” (REUSSI), 2012.
- Graduate Fellowship, Department of Computer Science and Engineering, University of Minnesota, 2011-12.
- Guanghua Scholarship for M.S., Shanghai Jiao Tong University, 2008.
- Excellent Graduate of Shanghai Jiao Tong University, 2006.
- Academic Excellence Scholarship, Shanghai Jiao Tong University, 2003 and 2004.