

汪宇霆

上海交通大学

上海市闵行区东川路 800 号, 老行政楼 524 室, 邮编: 200240

电子邮箱: yuting.wang@sjtu.edu.cn

个人主页: <http://jhc.sjtu.edu.cn/~yutingwang>

本人现任上海交通大学计算机学院约翰·霍普克罗夫特计算机科学中心 (理论计算机研究所) 长聘教轨副教授。

研究方向

主要研究兴趣为程序设计语言理论 (尤其是 C、C++、Rust 等系统编程语言) 与系统软件 (编译器、操作系统等) 的形式化验证。在此背景下, 关注程序、规约与推理形式系统的设计, 构建实现这些形式系统的编程语言与框架, 并将这些形式系统及其实现用于构建和验证系统软件。

本人研究的一个核心主题是可组合性 (Compositionality), 即将大型系统分解为模块化组件、分别验证各组件、再组合验证结果、证明原系统性质的能力。近年来, 主要聚焦于以下三方面的组合性技术研究:

- 系统编程语言 (如 C、Rust) 编译器的形式化验证;
- 并发程序 (如无锁数据结构、并发原语) 的规约与验证;
- 高阶编程语言或逻辑系统的验证。

教育背景

- 明尼苏达大学双城分校 (美国), 计算机科学与工程, 博士, 2011.9 - 2016.12
博士论文: A Higher-Order Abstract Syntax Approach to the Verified Compilation of Functional Programs (面向函数式程序验证编译的高阶抽象语法方法)
导师: Gopalan Nadathur
- 康涅狄格大学 (美国), 计算机科学与工程, 硕士, 2009.8 - 2011.8
硕士论文: AMIBE: an Imperative Programming Language with First Class Continuations (AMIBE: 带一阶续延的命令式编程语言)
导师: Laurent Michel
- 上海交通大学, 电力系统及其自动化, 硕士, 2006.9 - 2009.3
- 上海交通大学, 电气工程及其自动化, 学士, 2002.9 - 2006.6

工作经历

- 上海交通大学计算机学院, 约翰·霍普克罗夫特计算机科学中心 (理论计算机研究所), 长聘教轨副教授, 2020 年—至今
- 耶鲁大学, 美国, 博士后研究员 (合作导师: Zhong Shao/邵中), 2016 年—2019 年
- INRIA Saclay, 法国, 研究实习 (合作导师: Kaustuv Chaudhuri), 2012 年夏、2014 年夏

教学经历

- 任课教师：程序设计原理与方法 (CS1604)，本科 IEEE 试点班，电子信息与电气工程学院，上海交通大学，2022—2026 年
- 任课教师：数理逻辑 (AI3606)，本科人工智能（卓越人才试点班），电子信息与电气工程学院，上海交通大学，2021—2024 年
- 任课教师：程序设计思想与方法 (CS171)，工科大平台，上海交通大学，2020 年
- 助教：形式语言与自动机理论 (CSCI-4011)，明尼苏达大学，2015 年（主讲教师：Gopalan Nadathur）
- 指导：
 - Haoze (Andrew) Wu，本科生，戴维森学院，2017 年（现为阿默斯特学院助理教授）
 - Dawn Michaelson，本科生，明尼苏达大学，2016 年

在读学生

- 毛逸舟，博士研究生，2025 年入学
- 倪亦澄，博士研究生，2024 年入学
- 吴进华，博士研究生，2022 年入学
- 姚栋文，硕士研究生，2024 年入学

毕业学生

- 张令，博士，2021—2026 年，华为费马实验室。博士论文：面向系统编程语言的可组合编译器验证研究。**入选 2026 年度中国计算机学会（CCF）博士学位论文激励计划 (形式化方法专委)**
- 刘思雨，硕士，2021—2024 年，腾讯。硕士论文：基于静态单赋值中间语言的函数式编译器验证方法
- 吴雨菲，本科生，2024—2025 年，摩根士丹利。毕设论文：基于形式语义的 C 语言解释器设计
- 梁亚伦，本科实习生，2024—2025 年，香港大学博士生（导师：Bruno C. d. S. Oliveira）
- 孙蒙，本科实习生，2022—2023 年，甲骨文
- 徐翔哲，本科实习生，2020—2021 年，普渡大学博士生（导师：Xiangyu Zhang）
- 周李韬，本科实习生，2020—2021 年，香港大学博士生（导师：Bruno C. d. S. Oliveira）
- 殷振国，硕士实习生，2020—2021 年，AMD

论文发表 (通讯作者以星号标注)

- Yicheng Ni, Yuting Wang*. *Foundational and Compositional Verification of Layered Concurrent Objects*. The 40th European Conference on Object-Oriented Programming (**ECOOP**), pages 21:1-21:31, 2026. **(CCF-B 类会议论文)**
- Dongwen Yao, Jinhua Wu, and Yuting Wang*. *Clight2Rust: Translating a Subset of C into Safe Rust with a Small Trusted Runtime*. The 20th International Symposium on Theoretical Aspects of Software Engineering (**TASE**), pages 55-72, 2026. **(CCF-C 类会议论文)**
- Ling Zhang, Yuting Wang*, Yalun Liang, and Zhong Shao. *CompCertOC: Verified Compositional Compilation of Multi-Threaded Programs with Shared Stacks*. Proceedings of the ACM on Programming Languages (PACMPL), 9(**PLDI**), pages 173:1-173:24, 2025. **(CCF-A 类会议论文, 以期刊形式发表)**
- Yu Zhang, Jérémie Koenig, Zhong Shao, and Yuting Wang. *Unifying Compositional Verification and Certified Compilation with a Three-Dimensional Refinement Algebra*. Proceedings of the ACM on Programming Languages (PACMPL), 9(**POPL**), pages 64:1-64:31, 2025. **(CCF-A 类会议论文, 以期刊形式发表)**
- Yicheng Ni, Yuting Wang*. *Generic Reasoning of the Locally Nameless Representation*. The 22nd Asian Symposium on Programming Languages and Systems (**APLAS**), pages 42-62, 2024. **(CCF-C 类会议论文)**
- 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. **(CCF-A 类会议论文, 以期刊形式发表)**
- Jinhua Wu, Yuting Wang*, Meng Sun, Xiangzhe Xu, and Yichen Song. *Towards a Framework for Developing Verified Assemblers for the ELF Format*. The 21st Asian Symposium on Programming Languages and Systems (**APLAS**), pages 205-223, 2023. **(CCF-C 类会议论文)**
- Siyu Liu and Yuting Wang*. *Verified Transformation of Continuation-Passing Style into Static Single Assignment Form*. The 17th International Symposium on Theoretical Aspects of Software Engineering (**TASE**), pages 20-37, 2023. **(CCF-C 类会议论文)**
- 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. **(CCF-A 类会议论文, 以期刊形式发表)**
- Xiangzhe Xu, Jinhua Wu, Yuting Wang*, Zhenguo Yin, and Pengfei Li. *Automatic Generation and Validation of Instruction Encoders and Decoders*. The 33rd International Conference on Computer-Aided Verification (**CAV**), pages 728-751, 2021. **(CCF-A 类会议论文)**
- 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. (**CCF-A 类会议论文，以期刊形式发表**)

- 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. (**CCF-A 类会议论文，以期刊形式发表**)
- 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*. The 25th European Symposium on Programming (**ESOP**), pages 752-779, 2016. (**CCF-B 类会议论文**)
- Yuting Wang and Kaustuv Chaudhuri. *A Proof-Theoretic Characterization of Independence in Type Theory*. 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*. 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*. 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.

软件系统

- **Stack-Aware CompCert** <https://github.com/yvting/compcertmc-website>

可生成机器码、经形式化验证的 C 语言编译器。作为首个经过形式化验证的商用优化编译器，CompCert(<https://compcert.org/>) 已被用于编译安全攸关的航空电子系统和核电站软件。为简化验证，CompCert 输出带有伪指令的抽象汇编代码，进一步编译为机器码的工作则由未经验证的外部工具链处理。本人与 Pierre Wilke、邵中 (Zhong Shao) 合作，设计并实现了对 CompCert 的这一扩展，使其支持到机器码的验证编译。其核心思想是为 CompCert 的内存模型加入显式的栈结构（从而使 CompCert 具备“栈感知”能力），并证明栈空间消耗在编译过程中得以保持。这使得编译到真实机器的有限栈内存成为可能。Stack-Aware CompCert 还支持一种称为上下文编译 (Contextual Compilation) 的组合编译方法，为形式验证操作系统 CertiKOS (<https://www.cs.yale.edu/flint/certikos/>) 中内核模块的组合性提供了基础支持。本人

还与徐翔哲合作，为 Stack-Aware CompCert 加入经验证的汇编器，使其生成 Linux 标准二进制文件格式（即 ELF 格式）的机器码，由此形成了 CompCertELF 编译器。

- **Nominal CompCert**

<https://github.com/SJTU-PLV/nominal-compcert-popl22-artifact>

CompCert 的轻量级扩展，支持对局部内存（例如线程私有栈内存）的模块化验证。CompCert 的一个代表性特性是其基于区块的内存模型，它在支持高层内存隔离的同时，允许比特级的底层操作。然而，该模型假设存在单一的全局内存空间，难以针对单个模块或线程的局部内存空间进行验证。本人提出了一种名义内存模型（Nominal Memory Model，采用名义技术（Nominal Techniques，由 Andrew Pitts 等人提出的对象命名理论），以支持将全局空间划分为相互隔离的局部区域。本人与张令用该名义内存模型扩展 CompCert，使得在 CompCert 中对内存的进行局部验证成为可能。

- **CompCertO with Direct Refinements**

<https://github.com/SJTU-PLV/direct-refinement-popl24-artifact>

同时支持异构程序（即用不同编程语言编写的程序）端到端编译和分离编译的形式验证编译器。经形式化验证的编译器如需用于实际生产，必须保证目标代码与源代码行为一致，即二者语义之间存在端到端的直接精化关系。为实现模块化程序的直接精化，传统方法将编译器内部工作机制暴露给用户，极大限制了其适用性。本人与张令发现，带有内存保护的 Kripke 关系可作为编译器各个阶段的统一语义接口，从而隐藏其内部细节。我们将该思想应用于 Zhong Shao 与 Jérémie Koenig 构建的 CompCertO 编译器，成功使得 CompCert 在支持异构模块的直接精化的同时，不暴露其内部工作机制。这使得针对带递归调用的复杂异构程序（如 C 与汇编之间互相调用）进行验证编译成为可能，这在此前极难实现。

- **CompCertOC**

<https://github.com/SJTU-PLV/compccertoc-pldi25-artifact>

CompCertO 的扩展，支持共享栈上数据的多线程程序。**CompCertOC 解决了可组合编译验证领域的一个长期公开难题，即如何将一个线程私有栈上分配的数据传递给另一线程，同时不破坏各线程编译的正确性。**由于通过参数传递和共享数据在并发程序中普遍存在，在 CompCertOC 之前开发的经验证编译器甚至无法支持教科书级别的多线程程序的分离编译。CompCertOC 在具备高级优化的工业级编译器 CompCert 基础上，支持异构、多线程、且相互调用的系统级模块（如 C 与汇编编写模块）进行组合编译验证，这使得其泛用性超越了已有的其他形式验证编译器。

- **RustCompCert**

<https://github.com/SJTU-PLV/CompCert/tree/rust-verified-compiler>

面向 Rust 编程语言的形式验证编译器前端。RustCompCert 以 Rust 核心子集 Rust-light 为输入，生成语义等价的 Clight 程序，后者再由 CompCert 编译为可执行代码。当前版本由本人与吴进华、余柳赬和孟令龙合作开发。该前端基于 CompCertO 技术，支持开放模块的编译验证，因此支持安全 Rust 代码与非安全代码（如 C 模块）的混合编译。Rust 的借用检查器（基于最新的 Polonius 算法）在 CompCert 抽象解释框架中单独实现，将借用检查器集成至前端的工作正在进行中，这将使 RustCompCert 成为完整可用的 Rust(子集) 形式化验证编译器。

- **Charon** <https://github.com/AeneasVerif/charon>

Eurydice <https://github.com/AeneasVerif/eurydice>

为构建从 Rust 到 C 的工业级编译链，本人研究团队为 Charon 和 Eurydice 做出了重要贡献。Charon 是一个工业级强度的 Rust 到 Low-Level Borrow Calculus（低级借用演算语言，简称 LLBC）的转译器，LLBC 是一种适用于 Rust 程序分析与验证的中间表示。Eurydice 负责将 LLBC 编译为 C。本工作的聚焦从多态 Rust 到 LLBC 的单态化翻译，并使得单态化 LLBC 能被翻译为 C 代码。针对 Charon，我们增加了对动态 Trait 对象的单态化和动态类型析构（drop）的支持；针对 Eurydice，我们增加了对动态大小类型（切片、trait 对象等）和虚表实例翻译的支持。**这使得单态化框架基本搭建完成，目前可以将 90% 以上 Rust 标准库翻译到 C 代码。**

- **Verified Layered Concurrent Objects**

<https://zenodo.org/records/19604812>

模块化并发程序验证框架，支持并发对象的分层组织和验证。该框架与倪亦澄和朱韬合作，在 Rocq 定理证明器中开发完成。核心思想是将并发对象表示为开放的标号迁移系统，并统一采用事件轨迹精化描述程序正确性。通过结合迁移系统的操作语义与的事件轨迹的指称语义，该框架同时支持常见的操作语义验证技术（模拟与归约），以及经验证层的垂直组合（分层精化）与水平组合（层接口扩展）。该框架已成功应用于验证并发原语（如自旋锁）以及复杂的无锁数据结构（如无锁并发有界队列）。

- **CSLED (Core Specification Language for Encoding and Decoding)**

<https://zenodo.org/records/8363543>

规约和自动生成经形式验证的机器指令编码器与解码器的框架。CSLED 是一种规约语言，用于形式化不同 ISA 的二进制级指令格式，是对 Norman Ramsey 开发的编码与解码规约语言 (SLED) 的形式化扩展。用户为特定 ISA（如 X86-64）编写 CSLED 规约，框架据此自动生成一对经形式化验证的指令编码器和解码器。CSLED 已成功应用于 CompCert 完整的 X86 和 RISC-V 汇编语言，为其生成经验证的指令编码器和解码器，进而用于构建 CompCert 的形式验证汇编器。CSLED 由本人与徐翔哲、吴进华、殷振国和李鹏飞合作开发完成。

- **Generic Locally Nameless Representation**

<https://zenodo.org/records/13924557>

基于局部无名表示 (Locally Nameless Representation, 简称 LNS)、对高阶编程语言进行泛型推理的框架。该框架与倪亦澄合作在 Rocq 中开发。通过定义关于高阶语言绑定结构的泛型操作及其公理，可在不对对象语言的语法作任何假设的情况下，描述和证明关于 LNS 的各种关键引理。该框架已成功应用于形式化证明 PCF 函数式语言中程序上下文等价与程序逻辑等价的一致性。该证明是著名程序设计语言教科书 Practical Foundations for Programming Languages 第 47 章纸上证明的形式化翻译。**在翻译过程中，我们发现了该书中紧致性证明 (Compactness) 的一处隐蔽错误，并提供了修正。该修正已被作者（卡内基梅隆大学 Robert Harper 教授）纳入勘误表** (<https://www.cs.cmu.edu/~rwh/pfpl/supplements/errata.pdf>)。

- **Abella** <http://abella-prover.org>

基于高阶证明论的新型交互式定理证明器，以其对高阶抽象语法 (Higher-Order Abstract Syntax) 的支持和用于形式规约推理的两级逻辑方法而著称。该系统是本人与明尼苏达大学及法国 INRIA Saclay 研究人员的合作成果。本人对该系统的贡献包括：实现了被称为高阶遗传 Harrop 公式逻辑 (Logic of Higher-Order Hereditary Harrop Formulas) 的规约逻辑，设计了在复杂推理任务中使用该增强功能的方法（与

Kaustuv Chaudhuri 合作)；设计并实现了称为模式多态 (Schematic Polymorphism) 的多态机制，使得 Abella 支持多态规约和证明（与 Gopalan Nadathur 合作）。

- **AMIBE** http://digitalcommons.uconn.edu/gs_theses/142/
支持一阶续延的命令式编程语言，由本人与 Laurent Michel 共同设计。本人在 C++ 中实现了 AMIBE 的编译器。该系统展示了如何利用现代编译器基础设施（如 LLVM）提供的丰富优化来开发高效的约束式编程语言 (Constraint Programming Languages)。
- **VSTLF** <https://github.com/ldmbouge/vstlf>
超短期负荷预测 (Very Short Term Load Forecasting, 简称 VSTLF) 是一个利用神经网络进行短期（从分钟到小时）电力负荷预测的系统。本人于 2009 年至 2011 年担任 VSTLF 项目的主程序员。VSTLF 使用 Java 编写。

获奖情况

- 明尼苏达大学研究生院博士论文奖学金，2014—2015 年
- NSF-INRIA 资助“美国学生在 INRIA 的研究经历” (REUSSI) 旅行奖，2012 年
- 明尼苏达大学计算机科学与工程系研究生奖学金，2011—2012 年
- 上海交通大学光华奖学金，2008 年
- 上海交通大学优秀毕业生，2006 年

学术兼职

- 上海交通大学，理论计算机科学研究所，2025 年起
- 第四届全国计算机科学技术名词编写委员会，理论计算机科学编写分委员会委员，2025 年起
- 中国计算机学会，形式化方法专业委员会执行委员，2021 年起
- ACM 会员，2013 年起

学术服务

- 学术会议程序委员会 (联合) 主席: TyDe(2025)
- 学术会议程序委员会委员: CCF PL(2026), POPL(2025), PriSC(2025), ML Family Workshop(2025), CAV(2025), APLAS(2024), CPP(2023), FORTE(2023), LFMTTP(2018)
- 《软件学报》专刊程序委员会委员: 形式化方法与应用 (2025), 约束求解与定理证明 (2022), 定理证明理论与应用 (2021)
- 工件评估委员会委员: ICFP(2020), ICFP(2019)
- 上海交大全国高考招生组成员 (江西省) (2020-2026)
- Moonbit 编程语言全球编程挑战赛评委 (2024)
- CCF《国际编译器技术论坛》讲者 (2025)
- 中国软件大会《编程语言前沿技术论坛》，《Rust 系统软件论坛》讲者 (2025)
- CCF 秀湖会议第十八期《面向大型基础软件的形式化验证技术》讲者 (2024)
- 中国软件大会《形式化方法教育论坛》讲者 (2024)

- 中国软件大会，《证明工程与安全编程论坛》技术论坛讲者 (2023)
- CCF 系统软件专委《编译器技术论坛》讲者 (2023)
- 中国计算机大会，《计算机软硬件系统安全性形式化验证》技术论坛讲者 (2022)
- 中国计算机大会，《安全程序设计语言》技术论坛讲者 (2021)
- 中国软件大会，《形式化方法青年学者论坛》讲者 (2021)
- 参与编写 CCF 学科发展报告《交互式定理证明及应用》(2023)
- 参与编写 CCF 学科发展报告《面向安全泛在计算的系统软件的研究进展与趋势》(2022)