

Zhiqi Chen

zhiqichen@prg.is.titech.ac.jp | Tokyo, Japan | [GitHub](#)

EDUCATION

Institute of Science Tokyo , Japan, M.Sc. in Mathematical and Computing Science	09/2024–present
• Focus: Algebraic effects, type and effect system, and functional programming	
• Thesis: <i>Interactive Programming with Algebraic Effect Handlers and Holes</i> [PDF]	10/2024–08/2026
Nanjing University , China	09/2019–06/2023
Bachelor of Science, major in Computer Science and Technology	GPA:4.32/5.0
• Honors: The People's Scholarship in China for two academic years	2019–2020, 2020–2021
• Teaching Assistant & Grader Structure and Interpretation of Computer Programs (SICP)	Fall 2020
• Thesis: <i>The Design and Implementation of Algebraic Effects in Programming Languages</i>	10/2022–06/2023

RESEARCH & PROJECTS

EffHole Language	<i>Master Thesis</i>	10/2024–08/2026
• Designed a programming language with safe combination of algebraic effects and holes		
• Established the meta-theory of the core calculus, including the main soundness theorem of the dynamic semantics, fully mechanized in Agda with 3000+ lines of code		
• Implemented a prototype language in Haskell and an interactive environment for the language as a vscode extension		
Homura Language	<i>Capstone Project</i>	10/2022–06/2023
• Implemented a functional programming language with algebraic effects and effect handlers		
• Implemented a type-and-effect system with full type inference based on constraint solving supported		
• Supported other features such as algebraic data types and recursive types and pattern matching		
AgdaHoare	<i>Individual Project</i>	01/2022
• Studied the Hoare Logic of a small declarative language in Agda		
• Discussed the small-step and big-step operational semantics, and proved the equivalence between these two semantics		
• Explored the Hoare Logic with embedded assertion logic, and analyzed the soundness of the small-step semantics		
Raytracer-rs	<i>Individual Project</i>	03/2022–06/2022
• Carried out a full implementation of a Physically Based Rendering (PBR) in Rust		
• Fulfilled the features including Whitted-style Ray Tracing, Microfacet BRDF Model, Multiple Importance sampling etc.		
FPGA-based Full Implementation of the CPU of MIPS Architecture	<i>Group Project</i>	12/2020
• Implemented a MIPS assembly code generator in Haskell by building a C-like eDSL		
• Developed a simple shell program and several user programs including a big integer calculator and Fibonacci sequence which ran successfully on our computer system		
• Yielded the whole program by my codegen which has 2314 lines of assembly code		

ACADEMIC COMMUNITY EXPERIENCE

ACM Student Research Competition at International Conference on Functional Programming 2025 (ICFP 2025)		10/2025
3rd place winner, for the research abstract titled <i>Towards Programming with Effect Handlers and Typed Holes</i>		
Poster presentation at the 27th Programming and Programming Language Workshop (PPL 2025)		03/2025
<i>Towards Live Effectful Programming with Typed Holes, Zhiqi Chen, Youyou Cong and Hidehiko Masuhara</i>		
PLaX Reading Group , Nanjing University	<i>Supervised by Dr. Xinyu Feng and Dr. Hongjin Liang</i>	02/2021-06/2023
• Read and discussed research papers of programming languages conferences including POPL, PLDI, ICFP etc. • Attended weekly group meetings, gave speeches on my review of two research papers • Designed posters and gave presentation on my capstone project on Algebraic Effect		
Huawei Programming Language Community Meta Programming Special Interest Group		Core Member 06/2021–present
• Participated in several internal meetings for the initial development of the community and the organizations of academic salons • Joined the Huawei-Edinburgh Coffee House, which were held by Huawei-Edinburgh Joint Lab		
Attended 2020 China Functional Programming Conference		

ADDITIONAL INFORMATION

• Programming Languages Competent: C++, Haskell, C, Python Familiar: Agda, Coq, Scheme, Scala, SQL, Ocaml, Pascal
• Formal Methods familiar with Agda and Coq, and methods of verification of program properties by them.
• Compiler have an experience of building eDSLs in Haskell, implementing compilers for a variant of Scheme, familiar with Parsec. I also have an experience of bootstrapping an interpreter for Forth language from machine code of x86.
• Architecture have a knowledge of FPGA development and assembly languages, and have an experience of developing an eDSL to generate assembly codes, implementing a tiny computer system.
• Development Tool have a knowledge of development tools such as make, git, Haskell stack, Rust cargo etc., using $\LaTeX$, Markdown to write documents. I am a daily Ubuntu user, and can work with Linux.
Languages: Chinese(native), English(TOEFL R30/L30/S22/W26), Japanese(JLPT N1 139/180)