

Zhan Shi

✉ darkflames@pm.me | github.com/dark-flames | t.me/Dark_flames

Education

Kyoto University

Master of Informatics in Communication and Computer Engineering

Advisor: Atsushi Igarashi

Apr, 2022 - Mar, 2024

Kyoto, Japan

Kumamoto University

B.Eng. in Computer Science

Oct, 2019 - March, 2022

Kumamoto, Japan

Shandong University

B.Eng. in IoT Engineering

Sep, 2017 - June, 2021

Qingdao, Shandong, China

Research Interests

Dependent Type Theory:

- Implementation of dependently typed programming languages, focusing on dependent elaboration and dependent pattern matching.
 - Exploring semantic models and methods for dependent type theories, including categorical semantics, normalization-by-evaluation, and logical relations.
 - Improving the usability of dependently typed programming languages to extend the power of dependent types into general programming.
 - Investigating the integration of dependent types with other type systems, such as gradual typing and effect systems.
-

Work Experience

Luogu, Shanghai, China

Backend Team Leader, Remote, Part-Time

[luogu.com](https://www.luogu.com)

Jul, 2017 - Apr, 2023

- Led and participated in the backend development of the biggest online-judge platform in China.
 - Designed and developed a back-end framework in PHP with dependency injection and container compilation.
 - Designed and led the development of several curial middlewares, including a distributed asynchronous task worker in Rust, a WebSocket server in Python.
 - While continuously introducing new features, progressively optimized and refactored the existing codebase, enabling scalability from tens of thousands to over a million users, and supported an annual judgment volume of fifty million.
-

Research Experience

A Cast Calculus for Implementing Gradual Dependent Types, Master's Thesis

Jan, 2023 - Feb, 2024

Kyoto University, advised by Prof. Atsushi Igarashi and Prof. Taro Sekiyama

Kyoto, Japan

- Proposed a novel approach to implementing gradual dependent types soundly and efficiently for introducing dependent types into general-purpose programming languages.
 - Introduced a cast calculus as the core language base on dependent pattern matching and pattern unification.
 - Implemented a prototype of the cast calculus in Haskell.
 - Presented a short paper[1] at SRC@SPLASH 2023, giving a talk and a poster presentation.
-

OpenSource Contributions

Aya Prover, Practical implementation of a dependent type system

[aya-dev](https://github.com/aya-dev)

- Overhauled records to support dependent types.
 - Helped with some bugs and refactorings in primitive definitions.
-

Personal Projects





yukino, A type-driven and high-performance ORM framework in Rust

[yukino-dev](https://github.com/yukino-dev)

- Derives SQL operations from simple Rust code based on a monadic structure.
- Provides a functional query builder that delegates its type-checking to the type system of Rust (makes heavy use of type-level computation).
- The abstraction is type-safe but zero-cost, ensures both efficiency and type safety.

toy-dt-cpp, A toy dependently typed language implementation in C++

[top-dt-cpp](https://github.com/top-dt-cpp)

quote-data , A tokenization library for procedural macros in Rust	 quote-data
annotation-rs , Compile-time annotation parser for Rust	 annotation-rs
derivation-resolver , Derivation tree resolver for STLC and System F in Rust	 derivation-resolver
riscv-cpu , Assignment project, a pipelined RISC-V CPU in Verilog	 RISC-V-CPU

Skills

- **Programming Languages:**
 - Proficient in both Object-Oriented Programming and Functional Programming, experienced in meta-programming. Able to work effectively with any programming language.
 - Highly proficient in Agda, C/C++, Haskell, PHP, Python, Rust.
 - Comfortable with Coq, Java, JavaScript, OCaml, TypeScript.
- **Type Theory:** familiar with variants dependent type theories, their semantics, and relevant proofs, interested in normalization-by-evaluation, gradual typing, and effect systems, and also have experience with formal verification.
- **Compiler:** understand variants compiler architectures, familiar with compiler frontend, especially in type checking. Also have experience in parser generator and DSL design.
- **Web Development: 7 years** of experience in full-stack web development:
 - Backend: proficient in various backend frameworks in PHP, Python, and Rust, including Symfony, Laravel, Django, and Actix. Familiar with message queues and asynchronous programming. Also experienced in backend framework design.
 - Frontend: experienced with Vue.
 - Database: knowledgeable in MySQL, PostgreSQL, Redis, and Elasticsearch with a strong background in database design and optimization.
 - DevOps: skilled in Docker, proficient in CI/CD practices, and experienced with cloud services.
- **Development Tools:** can adapt to any editors/OSs, usually use JetBrains IDEs and VSCode under Ubuntu, familiar with git and docker, and have experience with team collaboration tools like GitHub and Slack.
- **Languages:** Chinese: native, English: advanced, Japanese: intermediate

Honors and Awards

National Olympiad in Informatics in Provinces , First Prize	Nov, 2015
Competitive programming contest held by China Computer Federation	Shandong, China

Publications & Talks

- [1] Z. Shi, “Partial Gradual Dependent Type Theory,” in *Companion Proceedings of the 2023 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity*, in SPLASH 2023. Cascais, Portugal: Association for Computing Machinery, 2023, pp. 22–24. doi: [10.1145/3618305.3623594](https://doi.org/10.1145/3618305.3623594).