

# Zhan Shi

✉ [darkflames@pm.me](mailto:darkflames@pm.me) |  [github.com/dark-flames](https://github.com/dark-flames) |  [t.me/Dark\\_flames](https://t.me/Dark_flames) |  [dark-flames.com](https://dark-flames.com)

---

## Education

### Kyoto University

**Master of Informatics** in Commucation and Computer Engineering

Apr, 2022 - Mar, 2024

Kyoto, Japan

Advisor: Atsushi Igarashi

### 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

- Implementation of dependently typed programming languages, focusing on features such as pattern matching.
  - Semantic models of type theories, especially categorical semantics, leverage categorical perspectives to grasp the essence of new language features quickly.
  - Semantics-based approaches for programming languages, such as normalization-by-evaluation and logical relations.
  - Dependent version of practical type systems, such as gradual typing and effect systems.
- 

## 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.
- 

## Work Experience

**Luogu**, Shanghai, China

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


Backend Team Leader, Remote, Part-Time

Jul, 2017 - Apr, 2023

- Led and participated in the backend development of the biggest online-judge platform in China.
  - Designed and developed a backend framework in PHP with dependency injection and container compilation.
  - Optimized and refactored the existing codebase progressively while continuously introducing new features, enabling scalability from tens of thousands to over a million users, and supporting an annual judgment volume of fifty million.
- 

## OpenSource Contributions


**Aya Prover**, Practical implementation of a dependent type system

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

- 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://yukino-dev.github.io)

- Derived SQL operations from simple Rust code based on a monadic structure.
- Developed a functional query builder that delegates its type-checking to the type system of Rust.
- Provided a zero-cost abstraction that ensures both efficiency and type safety.


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

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


**quote-data**, A tokenization library for procedural macros in Rust

 [quote-data](https://quote-data.github.io)

**annotation-rs**, Compile-time annotation parser for Rust

 [annotation-rs](https://annotation-rs.github.io)

**derivation-resolver**, Derivation tree resolver for STLC and System F in Rust

 [derivation-resolver](https://derivation-resolver.github.io)

---

## Skills

- **Programming Languages:** Not limited to any specific language, especially experienced in Agda, C/C++, Haskell, PHP, Python, Rust, comfortable with Coq, Java, JavaScript, OCaml, TypeScript.
- **Type Theory:** familiar with various dependent type theories and their semantics, and have experience with formal verification.
- **Compiler:** understand various 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, proficient in backend development, databases, and DevOps practices.
- **Languages:** Chinese: native, English: advanced, Japanese: intermediate

---

## 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).