# Tesla Zhang

ice1000kotlin@gmail.com · https://ice1000.org · https://github.com/ice1000

## Education

B.S. in Computer Science at **The Pennsylvania State University**, PA, US — Aug, 2018 – Dec, 2022
Minor in Mathematics, GPA 3.28/4.00, selected courses: Math 435, Cmpsc 450, Math 427, Math 429, Math 437

Ph.D. in Computer Science at **Carnegie Mellon University**, PA, US — Aug, 2023 – Present
Advisor: Stephanie Balzer, selected courses: 15-836, 15-791

## Work Experience

**JetBrains Research**, Remote — Jan, 2020 – Dec, 2020
*HoTT and Dependent Types*, Interactive Theorem Prover Development
- Improved the language/IDE, such as sections, hygiene macros, `Fin` type with elaborative subtyping, semantic highlighting, etc.
- Created a debugger for inspecting bidirectional type-checking and REPL in both CLI and IntelliJ IDEA.

**PLCT Lab**, Remote — Dec, 2020 – Present
*Implementation of Dependent Types*, Opensource Maintainer

**RisingWave Labs**, Remote — Jul, 2022 – Jul, 2023
*Streaming Database*, Developer Intern
- Implemented a pretty printing framework for trees with smart line fitting and Unicode art. Integrated into SQL explain.

**Sourcebrella Inc.**, Shenzhen, China — Feb, 2018 – Jul, 2018
*Static Analysis*, Compiler Frontend, IDE Plugin Development

**PingCAP Inc.**, Remote — Aug, 2018 – Aug, 2019
*Distributed Storage Systems*, TiKV Intern – Ecosystem Team

## Related Projects

**Aya Prover**, Practical Implementation of Dependent Types (role: project leader) — aya-prover/aya-dev
- Supports dependent types, dependent pattern matching with confluence check for overlapping clauses, higher inductive types, GADTs (paper published), hierarchial universes, cubical type theory features, and implicit arguments.
- Supports visualization of the type checking traces and exporting elaboration result to HTML or LaTeX. Supports LSP in VSCode. Binaries releases are built with jlink and GraalVM native-image.

**IntelliJ Pest**, Pest language plugin for IntelliJ Platform — pest-parser/intellij-pest
- Semantic-based highlighting, completion, navigation, definition extraction/inlining, and Rust plugin integration.
- Provides live preview – test grammar files by dynamically highlighting user code according to the grammar on the fly. These highlighted code could be exported to HTML.

## Skills

- Programming Languages: multilingual (not limited to any specific language), especially experienced in Java Kotlin Rust C# Agda Haskell Arend, comfortable with Dart C C++ F# F★ Idris Perl MATLAB (in random order).
- Compiler: understand various program representations such as CFG, ANF, (P)HOAS, etc. and normalization by evaluation.
- Kotlin/Java: **8 years of experience**, familiar with JNI, JPMS, Gradle, Kotlin coroutines, and Swing.
- Type Theory: understand Martin-Löf type theory, coinduction, HoTT, and Cubical, familiar with Idris, Agda (**3 years** of experience, contributor), Arend and some Lean/F★/Coq.
- IDE Tooling: **4 years of experience**, familiar with the IntelliJ Platform infrastructure (created Julia, DTLC, Pest, etc.), also have experience with Eclipse/SonarQube/VSCode plugin development.
- Tools: editor-agnostic, have experience with team tools like YouTrack, Jira, GitHub, BitBucket, Slack, JetBrains Space and more.

## Misc

- Profile links (please use a PDF reader with hyperlink support): Crates.io, IntelliJ Marketplace
- Languages: English - fluent (TOEFL 100), Chinese - native speaker
- Open-source contributions: https://ice1000.org/opensource-contributions, contributed to **agda, Arend, libgdx, jacoco, KaTeX, shields.io, grpc-rs, intellij-solidity, intellij-haskell, intellij-rust, TeXiFy-IDEA, rust-analyzer** and other projects
- StackOverflow: 6000+ reputation, also active on Proof Assistants (5000+ reputation) and other StackExchange sites
- Latest revision of this resume: one-page version https://tinyurl.com/y8xdlfug, complete version: https://tinyurl.com/y2v59t36
- **1 dan** on CodeWars, ranked #111 on the whole site (Top 0.020%), primarily in Haskell, Agda, and Idris

## Publications & Preprints

[1] T. Zhang, "A Simpler Encoding of Indexed Types," in *Proceedings of the 6th ACM SIGPLAN International Workshop on Type-Driven Development*, in TyDe '21. Republic of Korea: ACM, 2021. doi: 10.1145/3471875.3472991.