

# TESLA ICE ZHANG

✉ ice1000kotlin@foxmail.com · 🌐 ice1000 · 🌐 ice1000.org · in ice1000

## EDUCATION

---

**The Pennsylvania State University**, PA, US

08/2018 – Present

Major: Computer Science, Anticipated Graduation Date: 06/2022, GPA 3.20/4.00

## WORK EXPERIENCE

---

**JetBrains Research**, Remote

01/2020 – Present

(HoTT and Dependent Types) Theorem Prover Arend development Intern

- Used features like composite build and buildSrc to reduce build time and improve automation.
- Designed and implemented syntax sugars and IDE features, such as the `___` syntax, subexpr visitor, etc.

**PingCAP Inc.**, Remote

08/2018 – 08/2019

(Distributed Storage Systems) TiKV Intern - Ecosystem Team

- Improved many TiKV-relevant libraries, like optimizing the performance of grpcio, adding new features to procinfo.
- Helped to migrate the Protocol-Buffer library used by TiKV and its Raft implementation.

**Sourcebrella Inc.**, Shenzhen, China

02/2018 – 07/2018

(Static Analysis) Compiler Front-End, IDE Plugin Dev Intern

- Created IntelliJ/CLion/Eclipse plugin for the Pinpoint analyzer. Co-worked on the SonarQube plugin.
- Created a multi-threading cross Java/Kotlin source code indexer which can index Hadoop within 4 minutes.

## PERSONAL PROJECTS

---

**Voile**

<https://github.com/owo-lang/voile-rs>

Experimental dependently-typed language with kind-based row-polymorphism

- Supports dependent types, meta variable unification, and non-dependent row-polymorphic record and variant.
- Supports cumulative universe levels and uses a universe lifting operation to get rid of  $\text{Set}\omega$ .
- Implements CLI parser/source code parser and interactive REPL with the help of Rust language's ecosystem.

**IntelliJ Pest**

<https://github.com/pest-parser/intellij-pest>

A Pest grammar language plugin for IDEs based on the IntelliJ Platform

- 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.
- Supports exporting the highlighted user code to HTML format.

## SKILLS

---

- **Program Language: 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 (in random order).
- **Compiler:** understand various program representations such as Control Flow Graphs and A-Normal Form.
- **Kotlin/Java: 4 years** of experience, familiar with JNI, Gradle, and Swing, understand Contract DSL and Kotlin coroutines, did some code analysis with Kotlin's compiler.
- **Type System:** understand Martin-Löf Type Theory, coinduction and Cubical Type Theory, currently studying Homotopy Type Theory, familiar with Idris, Agda (**2 years** of experience, contributor), F\*, Arend and some Coq.
- **IDE Tooling: 3 years** of experience, familiar with the IntelliJ Platform infrastructure (created Julia, DTLC, Pest and many other plugins), also have experience with Eclipse/SonarQube plugin development.
- **Development Tool:** can adapt to any editors/OSs, usually use JetBrains IDEs and Emacs under Ubuntu.

## MISCELLANEOUS

---

- Some profile links (please use a PDF reader with web link support): Bintray profile (for publishing JVM libraries), Crates.io profile (for publishing Rust libraries), IntelliJ Plugin developer profile, Research Statement
- Languages: English - fluent (TOEFL 100), Chinese - native speaker
- Opensource Contributions: <https://ice1000.org/opensource-contributions> contributed to agda, KaTeX, shields.io, raft-rs, grpc-rs, intellij-solidity, intellij-haskell, intellij-rust, rust-analyzer and other projects
- StackOverflow: <https://tinyurl.com/y5cmw3dz> 4500+ reputations, also active on other StackExchange sites
- Get the latest version of this resume: <https://tinyurl.com/y8xdlfug>
- Get the complete version of this resume: <https://tinyurl.com/y2v59t36>
- **1 dan** on CodeWars, ranked #31 on the whole site, primarily in Haskell, Agda and Idris
- Love making friends