5713768264
West Lafayette, IN
zhou956@purdue

Zhe Zhou

Homepage
GitHub
LinkedIn

Education

PhD of Computer Science **2018.8 – 2025.4 (expected)**
Purdue University, Advised by Prof. Suresh Jagannathan

Main courses: Programming Languages, Reasoning about Programs, Compiling and Programming Systems
Operating Systems, Pattern Recognition and Decision-Making Processes (GPA: 4.0)

Bachelor of Computer Science **2013.9 – 2017.7**
Peking University, Advised by Prof. Guangyu Sun

Work Experience

Full Time C++ Software Engineer **2017.7 – 2018.7**
Megvii *Beijing, China*

Applied Scientist Internship **2024.5 – 2024.9**
Amazon *Santa Clara, CA, USA*

Research Interest

refinement types, property-based testing, specification inference, program synthesis, distributed system

Skills&Languages

Mostly used:	Ocaml, Coq, Z3
Familiar with:	Dafny, SML, C, C++, Java, Python, Scala, Haskell, P

Publication

Data-Driven Abductive Inference of Library Specifications **OOPSLA'21**
***Zhe Zhou**, Robert Dickerson, Benjamin Delaware, and Suresh Jagannathan*
(**Distinguished Artifact**)

Covering All the Bases: Type-based Verification of Test Input Generators **PLDI'23**
***Zhe Zhou**, Ashish Mishra, Benjamin Delaware, and Suresh Jagannathan*
(**Distinguished Paper**)

A HAT Trick: Automatically Verifying Representation Invariants Using Symbolic Finite Automata **PLDI'24**
***Zhe Zhou**, Qianchuan Ye, Benjamin Delaware, and Suresh Jagannathan*

Derivative-Guided Symbolic Execution **POPL'25**
*Yongwei Yuan, **Zhe Zhou**, Julia Belyakova, and Suresh Jagannathan*
(**Conditional Accepted**)

Service

External Review Committee Member **OOPSLA'23**
Artifact Evaluation Committee Member **PLDI'23**

Project

Temporal Refinement Type System **PLDI'24, POPL'25, In progress**

Equip standard refinement type system with temporal specifications to verify effectful programs and distributed system.

Underapproximate Refinement Type System **PLDI'23, In progress**

Design a refinement type system that verifies the coverage property of the random test generator.

Data-driven Specifications Inference **OOPSLA'21**

Design a data-driven inference procedure which is guided by counterexamples to infer specifications of multiple the blackbox library APIs that are consistent with the given whitebox client code.

LLM for Proof Synthesis **In progress**

Combine Large Language Model (LLM) and symbolic approach in PL community to synthesize Coq proof.