

# Kuang-Chen Lu

Malden, MA 02148 | [kuang-chen\\_lu@brown.edu](mailto:kuang-chen_lu@brown.edu) |  Taiwan (R.O.C.) Citizen

Expecting to graduate in May 2025 with a doctoral degree in Computer Science. While pursuing my PhD, I have been developing software used by people worldwide, conducting user studies, analyzing quantitative data, and formalizing programming languages. I am passionate about applying my skills to solve real-world problems and further expanding my skill set.

## Education

---

**PhD, Computer Science** Sept. 2020 – May 2025  
**Brown University**, Providence, RI  
Advisor: Shriram Krishnamurthi  
Research Area(s): Programming Languages, Formal Methods, Computing Education

**MS, Computer Science** Sept. 2018 – May 2020  
**Indiana University**, Bloomington, IN  
Advisor: Jeremy Siek  
Research Area(s): Programming Languages (Gradual Typing, Relational Programming)

**BS, Bioinformatics** Sept. 2014 – May 2018  
**Shanghai Jiao Tong University**, Shanghai, China  
Advisor: Chaochun Wei  
Research Area(s): Bioinformatics (Sequence analysis)

## Software and Systems

---

**SMoL Tutor**: <https://smol-tutor.xyz> (Source code private to prevent cheating; available on request)  
A tutoring system built for teaching misconceptions about the behavior of conventional programming languages. It has been used at institutions both domestic and abroad and by independent users worldwide.

**Stacker**: <https://smol-tutor.xyz/stacker/> (GitHub: <https://github.com/LuKuangChen/stacker-2023>)  
A visualization tool built for explaining how conventional programming languages behave.

**Brown Benchmark for Table Types (B2T2)**: <https://github.com/brownplt/B2T2>  
A benchmark for programming languages/systems that process tabular data. It assembles table operators that are common in major tabular programming frameworks: R Tidyverse, Python pandas, Julia, LINQ, and SQL.

**SMoL Misinterpreters**: <https://github.com/LuKuangChen/smol-misinterpreters>  
A collection of interpreters, each of which represent a misconception about programming language behavior.

**RPAN: Rice Pan-genome Browser**: <https://cgm.sjtu.edu.cn/3kricedb/visualization>  
Visualize the pan-genome of ~3000 rice species. My major contribution is the phylogenetic tree visualizer on the left and its connection with the gene browser on the right.

**InstaModel**: <https://github.com/brownplt/insta-model>  
A model of the semantics of Static Python, a gradually typed Python supported by Meta's variant of CPython (cinder). The model is shown to match cinder by producing the same results on cinder's test suite.

**BUGS VSCode Support**: <https://github.com/LuKuangChen/vscode-language-bugs>  
A VSCode extension for the language BUGS (Bayesian inference Using Gibbs Sampling). It implements the Language Service Protocol.

**Equivalent of Cast Calculus**: <https://github.com/LuKuangChen/Equivalence-of-Cast-Calculi>  
A framework for formally proving the equivalence of cast calculi (e.g., coercions, hypercoercions, and threesomes).

## Skills

---

**General Computing**: Python, Rust, OCaml, Java, Julia, Racket, C, TypeScript, ReScript

**Domain-specific Computing**: Quantitative Data Processing (R, MATLAB, Google Sheets), Web (React, Elm, Google Apps Script), Tabular/Relational Programming (SQL, REL, Prolog, miniKanren, LINQ), SVG, (La)TeX, PyTorch, Proof Assistants (Coq, Isabelle, Agda), Formal Methods (Alloy/Forge, PLT Redex), UI/UX (Figma)

**CS Subfields**: Natural Language Processing, Machine Learning, Human-Computer Interaction, Bioinformatics

**Other Fields**: Biology, Statistics, Math, Cognitive Science, Psychology

**Languages**: English (Fluent), Mandarin (Native), Taiwanese Hokkien (Native), Cantonese (Native)

## Work Experience

---

### R&D Intern

May 2023 – Aug. 2023

### Compiler/Language Team, Relational AI

- Prevented the compiler from executing ill-formed higher-order relations.
- Compared several ways of defining variadic functions in Julia on their readability and performance.
- Refactored several units of the compiler so that variadic functions are always defined in the best way. The refactoring improves the compiler's performance by several magnitudes on large programs.
- Proposed a solution (liked and accepted by the team) to a bug in the handling of generic relations.
- Fixed a bug in the handling of some recursive generic relations.
- Created a tutorial for the rest of the company on how generic relations work.

## Teaching Experience

---

- Teaching Assistant for [CSCI 1730 Programming Languages](#): Fall 2021, Fall 2022, Fall 2023, Fall 2024
- Teaching Assistant for [CSCI 1260 Compilers](#): Spring 2021
- Teaching Assistant for [C311/B521 Programming Languages](#): Spring 2019, Fall 2019

## Publications

---

- **Lu, Kuang-Chen**, and Shriram Krishnamurthi. "Identifying and Correcting Programming Language Behavior Misconceptions." OOPSLA, 2024.
- **Lu, Kuang-Chen**, Shriram Krishnamurthi, Kathi Fisler, and Ethel Tshukudu. "What Happens When Students Switch (Functional) Languages (Experience Report)." *Proceedings of the ACM on Programming Languages* 7, no. ICFP (2023): 796-812.
- **Lu, Kuang-Chen**, Ben Greenman, Carl Meyer, Dino Viehland, Aniket Panse, and Shriram Krishnamurthi. "Gradual Soundness: Lessons from Static Python." *The Art, Science, and Engineering of Programming* 7, no. 1 (2022).
- **Lu, Kuang-Chen**, Ben Greenman, and Shriram Krishnamurthi. "Types for Tables: A Language Design Benchmark." *The Art, Science, and Engineering of Programming* 6, no. 2 (2022).
- **Lu, Kuang-Chen**. "Equivalence of Cast Representations in Gradual Typing." Master's thesis, Indiana University, 2020.
- **Lu, Kuang-Chen**, Jeremy G. Siek, and Andre Kuhlenschmidt. "Hypercoercions and a framework for equivalence of cast calculi." In *Workshop on Gradual Typing*. 2020.
- Ma, Weixi, **Lu, Kuang-Chen**, and Daniel P. Friedman. "Higher-order Logic Programming with  $\lambda$ Kanren." In *Proceedings of the 2020 miniKanren and Relational Programming Workshop*.
- **Lu, Kuang-Chen**, Weixi Ma, and Daniel P. Friedman. "Towards a miniKanren with fair search strategies." In *Proceedings of the 2019 miniKanren and Relational Programming Workshop*, pp. 1-15. 2019.
- Hu, Zhiqiang, Chen Sun, **Kuang-chen Lu**, Xixia Chu, Yue Zhao, Jinyuan Lu, Jianxin Shi, and Chaochun Wei. "EUPAN enables pan-genome studies of a large number of eukaryotic genomes." *Bioinformatics* 33, no. 15 (2017): 2408-2409.
- Sun, Chen, Zhiqiang Hu, Tianqing Zheng, **Kuangchen Lu**, Yue Zhao, Wensheng Wang, Jianxin Shi et al. "RPAN: rice pan-genome browser for~ 3000 rice genomes." *Nucleic acids research* 45, no. 2 (2017): 597-605.

## Awards

---

- The paper "Types for Tables" was [awarded the Editor's Choice in 2022 at the Programming Journal](#)
- The paper "Identifying and Correcting Programming Language Behavior Misconceptions" was [awarded a Distinguished Paper Award in 2024 at OOPSLA 2024](#)