

VEERA SIVARAJAN

✉ sveera.2001@gmail.com

🌐 linkedin.com/in/veerasivarajan

🐙 github.com/veera-sivarajan

Education

University of Massachusetts Amherst

Amherst, MA

B.S. Computer Science

Sep. 2019 - Dec. 2023

- Relevant Courses: Compilers (CS 410), Functional Programming (CS 220), Introduction to Algorithms (CS 311), Statistics I (STATS 515), Artificial Intelligence (CS 383), Operating Systems (CS 377)
- Activities: UMass Table Tennis Club, UMass Running Club

Experience

Center for Youth Engagement

May 2021 – May 2023

IOS Developer

Amherst, MA

- Developed interactive books using SwiftUI to help incarcerated students visualize the concepts easily.
- Designed and implemented a fail-safe technique to recover from application caching issues.
- Improved UI by implementing popular IOS design patterns.

Recurse Center

May 2022 – Aug. 2022

Participated in a self-directed educational retreat for programmers

Brooklyn, NY

- Deepened knowledge in programming language design and compiler implementation.
- Pair programmed with others and created a Emacs major mode for SerenityOS' Jakt programming language.
- Gave presentations about bytecode compiler and virtual machine implementation techniques.

Projects

Boa | Contributed to an open-source Javascript engine in *Rust*

- Fixed bugs to improve compliance with ECMAScript specification.
- Added tests to test262 repository to check for malformed expressions.
- Learned about compiler design and compiler testing techniques.

Boba | Compiler for a static and implicitly typed language to x86 assembly in *Rust*

- Designed the syntax and semantics of the language based on Rust and C.
- Implemented a lexer, a recursive descent LL(1) parser, a type inference and code generation mechanism.
- Optimized code by implementing constant propagation and constant evaluation on an IR in static single assignment form.

Bessy | Stack based bytecode interpreter in *Rust*

- Implemented an operator-precedence parser, a virtual machine and a mark-and-sweep garbage collector.
- Added a web interface for the compiler using WebAssembly and Javascript.
- Improved performance by interning all strings and removing unnecessary deep copies.

Lang0 | Tree-walk interpreter in *C++20*

- Implemented static analysis passes to check for unused and uninitialized variables.
- Designed and implemented anonymous functions and dynamic arrays.
- Used smart pointers and other modern C++ features to build a clean and sound interpreter.

Skills & Interests

Languages: Rust, C++, C, x86, Python, Java, Swift

Technologies: LLVM, MLIR, GDB, WebAssembly, Git, Bash

Interests: Systems software, Compiler development, Technical writing

Activities

PLISS | Programming Language Implementation Summer School

Oct. 2022

- Attended lectures on programming language design, implementation and testing techniques.
- Learned about static type checking for dynamic languages, proving compiler correctness and code optimization techniques.
- Interacted with researchers and explored the state of the art in programming language research.

PLDI | Volunteer for a Conference on Programming Language Design and Implementation

May 2022

- Moderated panel discussion for workshops.
- Provided technical assistance for virtual attendees.