Pierce Darragh

pierce.darragh@gmail.com
https://pdarragh.github.io

Education

In-Progress **University of Maryland**
PhD in Computer Science (Programming Languages).
Advised by David Van Horn.

2018 **University of Utah**
MS in Computer Science.
BS in Computer Science, Minor in Linguistics.

Teaching

As Instructor

Spring 2022 **CMSC 388X: Introduction to Programming Language Theory**
https://www.cs.umd.edu/class/spring2022/cmsc388X/

I developed a new undergraduate course to teach undergraduate students basic concepts in programming language theory. On one day of the week we covered content roughly lifted from *Types and Programming Languages*, and on the other day of the week we had discussions on assigned papers. Covered topics included:

- Syntactic theory (e.g., BNF grammars, metafunctions).
- Structural induction over syntax for constructing proofs.
- Reduction and typing relations via small-step operational semantics.
- The lambda calculus.
- Extending the lambda calculus with types and recursion.

To select the papers we read, students formed small groups and each group had to choose a paper from a pre-approved list to read and present for discussion with the class. Student feedback for the class was overwhelmingly positive.

As Graduate Teaching Assistant

Fall 2022–Present **CMSC 430: Compilers**
https://www.cs.umd.edu/class/spring2025/cmsc430/

This course teaches students how to implement compilers in Racket for languages of increasing complexity, targeting the x86 assembly language.

Over the past eight semesters, I have worked with various course instructors (Professors David Van Horn, Jose Manuel Calderon Trilla, Milijana Surbatovich, and Anwar Mamat), to improve the course. My work has included:

- Creating new assignments and modifying existing ones.
- Writing and grading midterm exams.
- Developing new instructional material (lecture notes, quizzes, etc.).
- Implementing new automated grading infrastructure.
- Analyzing course data to inform subsequent decisions and discussions.

I have also been working on the a86 Assembly Interpreter (see Selected Projects), which I intend to use for this class to guide students' debugging efforts in a more systematic and course-specific manner.

2023–2024 **Excellence in Teaching Award**

I was anonymously nominated for — and subsequently selected to win — a departmental Excellence in Teaching Award for the 2023–2024 academic year due to my efforts in TAing CMSC 430: Compilers (above). The department chooses ∼5 recipients for these awards each year, selected from among all staff and faculty.

Spring 2023 **CMSC 433: Programming Paradigms**

This course was loosely based on the previous semester's CMSC 488B (below), which taught students how to use Haskell by thinking lazily and functionally.

Spring 2022 **CMSC 488B: Advanced Functional Programming**

This course taught students how to program in Haskell, including discussions of basic category theory and the use of QuickCheck.

Fall 2021 **CMSC 330: Programming Languages**

This required undergraduate course taught students about programming in Ruby, OCaml, and Rust. It also introduced students to concepts in basic programming language theory, including the lambda calculus, operational semantics, type-checking, parsing, and so on.

Research

Publications

GPCE 2023 **Generating Conforming Programs With Xsmith.**

Authors: William Gallard Hatch, Pierce Darragh, Sorawee Porncharoenwase, Guy Watson, and Eric Eide.

Date: October 2023.

Venue: International Conference on Generative Programming: Concepts & Experiences 2023.

URL: pdarragh.github.io/p/gpce23

Synopsis: Xsmith is a domain-specific language for implementing fuzzers that operate in the style of Csmith, implemented in Racket. We provide implementations for a handful of languages and report on bugs identified in some of their implementations.

BRM 2021 **SweetPea: A standard language for factorial experimental design.**

Authors: Sebastian Musslick, Anastasia Cherkaev, Ben Draut, Ahsan Sajjad Butt, Pierce Darragh, Vivek Srikumar, Matthew Flatt, and Jonathan D Cohen.

Date: April 2021.

Venue: Behavior Research Methods, volume 54, issue 2.

URL: pdarragh.github.io/p/sweetpea

Synopsis: We introduce SweetPea, a domain-specific language for specifying factorial experimental designs, implemented in Python. Although built with the field of psychology in mind, SweetPea can be used for most factorial experiments.

Scheme 2020 **Clotho: A Racket Library for Parametric Randomness.**

Authors: Pierce Darragh, William Gallard Hatch, and Eric Eide.
Date: August 2020.
Venue: Scheme and Functional Programming Workshop 2020.
URL: pdarragh.github.io/p/scheme20
Synopsis: Clotho is a Racket library that implements *parametric randomness*, a style of (pseudo)random generation where external manipulations of recorded sampling events correspond to discrete changes in the structure of the output. It was built as part of the implementation of Xsmith.

ICFP 2020 **Parsing with Zippers (Functional Pearl).**

Authors: Pierce Darragh and Michael D. Adams.
Date: August 2020.
Venue: PACMPL, volume 4, issue ICFP.
URL: pdarragh.github.io/p/icfp20
Synopsis: Parsing with Derivatives is a known technique for implementing a parser with an elegant theory, but which suffers from poor performance. Parsing with Zippers is built upon the same theory of parsing, but featuring a deviation in the mode of traversal of the input that produces a significant speedup.

Presentations

RacketCon 2020 **Clotho: A Racket Library for Parametric Randomness.**
I was invited to give this talk again after presenting at the Scheme Workshop.
Scheme 2020 **Clotho: A Racket Library for Parametric Randomness.**
ICFP 2020 **Parsing with Zippers (Functional Pearl).**

Selected Projects

In-Progress **a86 Assembly Interpreter** github.com/cmsc430/a86-interpreter

CMSC 430 has students implement compilers in Racket targeting a restricted subset of the x86-64 assembly language, which we call "a86." I am implementing a stepable, time-traveling a86 interpreter with helpful, course-tailored implementation details to improve the student debugging experience.

Similar to efforts like Python Tutor and Learn-OCaml, I intend to extend the capabilities of this interpreter to provide automated course-specific feedback for students, such as hints for specific assignments based on heuristics we use effectively as instructors already. Additionally, I am developing a mechanism to synthesize information about student submissions to our automated grading platform to support instructor feedback in a setting with hundreds of students and too-few instructors.

2020–2021 **SweetPea** sweetpea-org.github.io

A domain-specific language built for the declarative specification of randomized experimental designs. I rewrote the back-end processing system and revised the front-end API.

2019–2020 **Xsmith** www.flux.utah.edu/project/xsmith
A generic fuzzer generator, built in the spirit of Csmith but implemented as a domain-specific language in Racket. I implemented the Python fuzzer specification and its necessary internal components, and also developed a new Racket library (named Clotho) to improve Xsmith's capabilities for exploring state spaces.

Professional Experience

2021–Present **University of Maryland**, Graduate Teaching Assistant
Assisted in the instruction and execution of various courses during my PhD.

2020–2021 **SweetPea Research Group, University of Utah**, Research Associate
Rewrote implementation and expended functionality of SweetPea under the direction of Matthew Flatt and in coordination with a team at Princeton University. This project culminated in a publication.

2019–2020 **Flux Research Group, University of Utah**, Research Associate
Developed new features for Xsmith under the direction of Eric Eide, including support for alternate type systems and a random program generator for Python. Also developed a new library, Clotho, to enable repeatable complex random generation simulation. This work resulted in two publications, one each for Xsmith and Clotho.

2018–2019 **U-Combinator Research Group, University of Utah**, Research Associate
Worked with Michael Adams on various projects as an extension of research that had been started as an undergraduate. This work resulted in the publication of Parsing with Zippers.

Summer 2017 **Apple, Inc.**, Software Engineer Intern
Designed, built, and presented a secure framework for automatically creating proxy servers intended for use in internal penetration testing.

Academic Awards

2023–2024	Excellence in Teaching, University of Maryland Department of Computer Science.
2021–Present	Dean's Fellowship, University of Maryland.
2012–2016	National Merit Scholarship, sponsored by E∗TRADE.
2012	Merit Scholarship with Presidential Honors, University of Utah.

Non-Academic Service and Leadership

2020–2022	Moderator, /r/ProgrammingLanguages Discord server.
2020–2021	Community manager, Jean Yang's `#PLTalk` Twitch stream and Discord server.
2014–2017	Web administrator, University of Utah Club Swim Team.
2014–2015	Men's team captain, University of Utah Club Swim Team.