

PIERCE DARRAGH

pierce.darragh@gmail.com
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.

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 Cherkayev, 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/cmssc430/a86-interpreter
 UMD uses a restricted subset of the x86-64 assembly language (called a86) for their undergraduate compilers course, which is implemented in Racket. I am implementing a step-able interpreter with helpful feedback to improve the student debugging experience.
 We also hope to expand the capabilities of the interpreter to use in our automated grading systems. I am looking at tools like Python Tutor and Learn-OCaml for inspiration.

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.

Teaching

AS INSTRUCTOR

Spring 2022 **CMSC 388X: Introduction to Programming Language Theory**
 I developed a new undergraduate course with a labmate. Select topics included:

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

Students also formed small groups, and each group selected one paper from a pre-approved list to read and present for discussion with the class.

AS GRADUATE TEACHING ASSISTANT

Fall 2022–Present	<p>CMSC 430: Compilers</p> <p>This course teaches students how to implement compilers in Racket for languages of increasing complexity, targeting the x86 assembly language.</p> <p>Over the past six semesters, I have been working with the course instructors, Professors David Van Horn and Jose Manuel Calderon Trilla, to improve the state of the course as it has exploded in popularity (enrollment is $\sim 3x$ what it was when I started TAing). This has involved rewriting and extending assignment autograders, developing new or alternate assignments, and helping write suitable exams, in addition to my regular TA responsibility of holding weekly office hours and being available online on Piazza or Discord to answer student questions.</p> <p>During this time, I have also been working on the a86 Assembly Interpreter (see Selected Projects above). I intend for this tool to be used as part of the class to help students understand assembly execution in a more direct way.</p>
Spring 2023	<p>CMSC 433: Programming Paradigms</p> <p>This course was loosely based on the previous semester’s CMSC 488B (below), which sought to teach students how to use Haskell by thinking lazily and functionally.</p>
Spring 2022	<p>CMSC 488B: Advanced Functional Programming</p> <p>This course taught students how to program in Haskell, including discussions of basic category theory and the use of QuickCheck.</p>
Fall 2021	<p>CMSC 330: Programming Languages</p> <p>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.</p>

Awards

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.

Experience

2020–2021	<p>SweetPea Research Group, University of Utah, Research Associate</p> <p>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.</p>
2019–2020	<p>Flux Research Group, University of Utah, Research Associate</p> <p>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.</p>

- 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.

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.