ANDREW BLINN

andrewblinn.com • me@andrewblinn.com • github.com/disconcision

RESEARCH INTERESTS

Programming Languages • Human-Computer Interaction • Computing Education

EDUCATION

University of Michigan • **Ph.D Student, Computer Science** • *September 2020 - Current*
Researching user interfaces for/as programming languages at Cyrus Omar's FP Lab.

University of Toronto • **H.B.Sc in Mathematics & Computer Science** • *May 2019*
Graduate-level coursework in abstract algebra, compilers, graphics & languages.
Coursework in algorithms, concurrency, differential geometry, operating systems & topology.
Built a Racket-based x86/C compiler for a λ-calculus-based language with macro system.

RESEARCH EXPERIENCE

Techniques in Variability-aware Data Structures with Marsha Chechik • *2018 - 2019*
Built & profiled Haskell data structures supporting variational analysis of software product lines.
Designed & built SpyShare, a Graphviz-based tool to visually inspect data sharing.
Created and modelled a system of GHC rewrite-rules using PLT Redex.
Project Report • Presentation Slides

Independent Study in Structured Editing in Racket with Gary Baumgartner • *Summer 2017*
Self-initiated study of existing refactoring, live programming & direct manipulation tooling.
Began work on Fructure, a Racket-based polyglot structure editor, and Containment Patterns,
which extend pattern matching to capture contexts as composable continuations.

CONFERENCES

Invited speaker at RacketCon • *2019* • *Salt Lake City*
Spoke about Fructure, a prototype structured editor focused on edit-time term-rewriting
Recorded Talk • Fructure Slides

Seat Filler • *Salt Lake City, Toronto, Eugene, St.Louis*
2019: Racket's How to Design Languages Summer School, Clojure North.
2018: Oregon Programming Languages Summer School, ICFP, Strange Loop, RacketCon

TEACHING

Course Development • *Summer 2018* • *University of Toronto*
Designed assignments and course materials for CSC324 - Principles of Programming Languages.
Specified and built Ductile, a toy language demonstrating exhaustive pattern matching on ADTs.
Implemented an algebraic stepper to illustrate continuations and non-determinism in Scheme.

Teaching Assistance • *University of Toronto*

Winter 2019	CSC324 Principles of Programming Languages
Fall 2018	CSC324 Principles of Programming Languages
Fall 2018	CSC104 Introduction to Computational Thinking
Winter 2018	CSC324 Principles of Programming Languages
Fall 2017	CSC324 Principles of Programming Languages

INDUSTRY EXPERIENCE

TODAQ Toronto • **Software Development in Clojure** • *May 2019 - August 2020*
On the back-end: Implementing and refining a new protocol for decentralized digital asset management based on a Merkel-trie-derived distributed data structure. On the front-end: Building user interfaces oriented around reifying distributed digital assets.