

ANDREW BLINN

January 12, 2024

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

GOALS

I want to make engagement with algebraic abstractions more tangible & fun
I use programming language theory to explore, explain, & engineer fluid compositional interfaces


FOCUSES

Programming Languages + Environments + Education · Human-Computer Interaction

PAPERS

[An Integrative Human-Centered Architecture for Interactive Programming Assistants](#) · VLHCC · 2022

Andrew Blinn, David Moon, Eric Griffiths, Cyrus Omar

An architecture for programming assistants addressing integrative design challenges, including a formal framework for assistant suggestion sensibility:  [Paperclip Calculus](#)

[Total Type Error Localization and Recovery with Holes](#) · POPL · 2024

Eric Zhao, Raef Maroof, Anand Dukkupati, Andrew Blinn, Zoe Pan, Cyrus Omar

The *marked lambda calculus* is a formal account of total error localization and recovery, keeping editor semantic services fed even downstream of errors

[Gradual Structure Editing with Obligations](#) · VLHCC · 2023

David Moon, Andrew Blinn, Cyrus Omar

By foregrounding terms, structured editors compromise casual text-like editing. Finer-grained *syntactic obligations* allows looser edits while guaranteeing reassembly

[Filling typed holes with live GUIs](#) · PLDI · 2021

Cyrus Omar, David Moon, Andrew Blinn, Ian Voysey, Nick Collins, Ravi Chugh

Livelits allow users to fill program holes by directly manipulating user-defined GUIs embedded persistently into code, providing continuous graphical feedback

WORKSHOP PAPERS

[Toward a Live, Rich, Composable, and Collaborative Planetary Compute Engine](#) · PROPL · 2024

Alexander Bandukwala, Andrew Blinn, Cyrus Omar

We sketch a programming environment for planetary computing with climate science applications

[Tylr - A Tiny Tile-based Structure Editor](#) · TyDe · 2022

David Moon, Andrew Blinn, Cyrus Omar

Tylr combines hierarchical and linear editing paradigms via a novel destructuring mechanism

SCHOOL

[University of Michigan](#) · Ph.D Candidate, Computer Science · Now

Making smart, fun, multi-modal programming environments using LLMs, integrative design, and type-directed formal methods at Cyrus Omar's [FPLab](#)

[University of Michigan](#) · Master's of Science, Computer Science · 2023

Coursework in programming language theory, category theory, program synthesis, human-computer interaction, & on how people learn

[University of Toronto](#) · H.B.Sc, Math & Computer Science · 2019

Graduate coursework in abstract algebra, compilers, & graphics

INDUSTRY

[TODAQ Toronto](#) · Full-stack development in Clojure · 2019 - 2020

Built novel front-end interfaces to [sharpen the materiality of distributed digital assets](#). Implemented core back-end features for a decentralized digital asset management protocol

CONFERENCES

Invited speaker at RacketCon · 2019 · Salt Lake City

Introduced [Fructose](#), a prototype structured interaction editor for on edit-time term-rewriting
· [Recorded Talk](#) · [Fructose Slides](#)

Accepted speaker at Midwest PL Summit · 2023 · Ann Arbor

Progress report on Type-directed Prompt Construction for LLM-powered Code Completion
· [MWPLS Slides](#)

Accepted speaker at VL/HCC · 2022 · Rome

Presented an Integrative Human-Centered Architecture for Interactive Programming Assistants
· [Recorded Talk](#) · [VLHCC Slides](#)

Seat Filler · Rome, Chicago, Salt Lake City, Galiano, Toronto, Eugene, St.Louis, Empire Builder

2023: [MWPLS](#), [Local First Unconf](#), [Fission TrainJam](#), [Strange Loop](#), [Gradient Retreat](#), [Causal Islands](#)

2020 - 2022: [VL/HCC](#), [Gradient Retreat](#), [SPLASH/OOPSLA](#), [HATRA LIVE](#)

2019: [Racket's How to Design Languages Summer School](#), [Clojure North](#).

2018: [Oregon Programming Languages Summer School](#), [ICFP](#), [Strange Loop](#), [RacketCon](#)

PROJECTS

IDE Design, Implementation, Deployment, and Analytics with Cyrus Omar · 2020 - Current

Led a ground-up [rewrite of the Hazel IDE](#), extending David Moon's tylr restructuring engine into a full-fledged editor, language server, and educational tool which was deployed to 100 undergraduates.
Designed & implemented novel bidirectional type system features. [More about the Hazel project](#)

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: [Report](#) · [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 [Fructose](#), a Racket-based polyglot structure editor, and [Containment Patterns](#), which extend pattern matching to capture contexts as composable continuations.

TEACHING

Course Development · Summer 2022 · University of Michigan

Designed software infrastructure and assignments for Programming Language Theory (EECS490)
Led implementation on Hazel School Mode, an editor integration providing progressive live feedback to students. Features an Instructor Mode facilitating construction of new exercises

Course Development · Summer 2018 · University of Toronto

Designed assignments and course materials for Principles of Programming Languages (CSC324)
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 Michigan

2023, 2022, 2021 EECS490 · Programming Languages

Teaching Assistance · University of Toronto

2019, 2018², 2017 CSC324 · Principles of Programming Languages

2018 CSC104 · Introduction to Computational Thinking

MENTORSHIP

LLM Token-masking for Type-correct Code Completion · 2023 Summer - Current

Mentoring an undergraduate by guiding experiments around modifying local language models to ensure per-token generation respects semantic as well as syntactic invariants

PL x Deep Reinforcement Learning for Code Completion · 2021 Fall - Current

ML/PL collaboration: Mentoring two undergraduates, providing design and architecture guidance for a specialized version of the Hazel editor used for reinforcement-learning-based code completion research.

Interfaces for Live testing in online IDEs · 2021 - 2022

Mentored an undergraduate who helped implement live testing features of the Hazel language and IDE, successfully deployed to a class of 100 undergraduate students for a course in Fall 2022.