# Andrew Blinn

*October 10, 2023*

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

## Goals

I want to make human engagement with algebraic abstractions more continuous, tangible, & joyous.
I study programming language theory to explain, explore, & engineer fluidly 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 Griffis, Cyrus Omar**
A unifying architecture for programming assistants addressing integrative design challenges.
The 📎 Paperclip Calculus formalizes assistant suggestion sensibility

**Total Type Error Localization and Recovery with Holes** · **POPL** · *2024*
**Eric Zhao, Raef Maroof, Anand Dukkipati, Andrew Blinn, Zoe Pan, Cyrus Omar**
Unlike typical type systems, the *marked lambda calculus* offers a comprehensive 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**
Structured editors struggle to preserve syntax without compromising text-like editing. Our approach guides temporary disassembly via *syntactic obligations* that, once discharged, guarantee proper reassembly

**Tylr - A Tiny Tile-based Structure Editor** · **TyDe Workshop** · *2022*
**David Moon, Andrew Blinn, Cyrus Omar**
Tylr marries hierarchically structured and linear editing paradigms via a novel destructuring mechanism

**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

## School

**University of Michigan** · **Ph.D Candidate, Computer Science** · *Current*
Building intelligent multi-modal programming environments combining LLMs, responsive 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 Fructure, a prototype structured interaction editor for on edit-time term-rewriting
· Recorded Talk · Fructure 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, Ann Arbor, Empire Builder*

2023: MWPLS, Local First Unconf, Fission TrainJam, Strange Loop, Gradient Retreat, Causal Islands
2022, 2021, 2020: 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 Fructure, 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, 2018$^2$, 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.