Thomas Porter

Research Statement

September 3, 2023

My main research interest can be summarized as improving human-computer interaction in the domain of programming and theorem proving. I am guided by a vision of an automated, interactive curriculum for the formal and formalized sciences. In this document I present a description of my guiding vision, the framework within which I conceptualize research related to improving human-computer interaction for proofs and programs, and finally the specific research questions I am interested in pursuing next. This order is logical in the sense of passing from the general and motivating to the specific and motivated, but perhaps the final section is the most relevant, as it describes my concrete personal intentions, so you may wish to jump there.

Vision

My research goals are motivated and guided by a vision of an automated, interactive curriculum in the formal and formalized sciences. The vision is derived from the following desirable properties of a learning system:

- Educational effectiveness
- Universal and eternal accessibility
- Ease and enjoyment of use
- Breadth, depth, and unity of content

By my reasoning, these principles suggest, if not nearly entail, the following system features:

- Use of a **dependent type theory** as the language of expression
- A **database of items**: formalized definitions, propositions, and exercises
- A transparent and **semantics-aware interface** between the user and the system
- A collection of powerful development and **paedogogical tools** enabled by the above items
- A **navigation system** capable of guiding a user through the content

Not only are these features close to necessary to achieve the stated goals, but I think they are not far from sufficient. I believe in the power of well-crafted exercises to inevitably impress upon a student practically any lesson. Additional material such as diagrams and natural language explanations would be helpful, and can indeed be added, but the more content is internalized into the formalism, the more powerful the automated tools for navigation and tutelage.

Research Framework

I view research in pursuit of improved human-computer interfaces for proofs and programs as mainly comprising the three areas, each of which depend on each other for feasibility. The first is making visible the hidden; bringing computation that is normally hidden or delayed into a visible interface and the present, enabling interpretability by the user. Examples include the following:

- Parsing is externalized to structure editing
- Implicit polymorphism is externalized to automated explicit polymorphic application
- Refinement types are externalized to accompanying proofs
- Coercive subtyping is externalized to inserted injections
- Type classes are externalized to accompanying records

- Tactics are externalized to commands resulting in visible rewrites, or object-language terms
- Monadic error handing and other reified errors fit this category

The second is formalizing the informal; bringing human user activities from outside the formalism into the formalism, enabling interpretability by the computer. Examples include the following:

- Top down development is internalized to holes, as in Agda, Idris, and Hazel
- Symbol reuse and ad-hoc polymorphism are internalized to type classes
- Reasoning about program correctness is internalized to dependent type systems
- Intermixing natural language and formalism is internalized in literate programming

The third is building tools for interaction that would not be possible without the above unifications. Examples include the following:

- The above cases of implicit polymorphism, refinement types, subtypes, and type classes are first externalized to visible syntactic constructs, then automated using semantic aware tools
- Error explanation, type debugging
- Hint suggestion for educational theorem proving
- General programming and proving assistants
- Context and library search
- Proof reuse and recovery
- Curriculum compilation and navigation
- Natural language of formalism

Personal Research Goals

My first and foremost research goal is to develop a gapless, semantics-aware editor for a dependently typed language. In particular, this could take the form of extending the core calculus of Hazel with dependent types, drawing from the literature on the bidirectionally typed and gradual calculus of inductive constructions.

Building atop the core language, I would like to work on developing an intelligent assistant for the editor. This has two main branches: developing an assistant system that suggests maximally helpful edit actions in arbitrary situations, and developing a tutoring system that, perhaps with access to an example solution hidden from the user, suggests hints of controlled magnitude. The first could take the form of extending the Hazel assistant to reason with the newly implemented dependent type system, drawing from the field of automatic theorem proving. The second would be an extension of the ideas represented in the CMU Proof Tutor powered by Automated Proof Search (AProS) from first order logic and set theory to dependent type theory.

I am also interested in general research questions related to developing human-centered interactive theorem proving systems. These include editing and interface features such as making visible the effects of tactics, semantic search, proof reuse and repair, enriched typography of formalism, and natural language generation from formalism. They also include language features, such as pattern-matching for quotient types and a type theoretic encoding of “structures up to symmetry" that I call gauge types.