

Second Year Review

William Troiani

February 13, 2023

1 Introduction

What is a proof? Thinking deeply about this question can and has lead to the establishment of formal connections between some aspects of mathematical logic and some aspects of theoretical computer science. For example, the Curry-Howard Correspondence can be understood as an equivalence between categories, one of which is “computational”, and the other which is “logical”. We take a moment to explain this equivalence, assuming familiarity with the basic constructs of the simply typed λ -calculus, and intuitionistic sequence calculus (with soul connective implication, denoted \supset), for more details the interested reader can consult [1].

Let Q be a set of variables, where a variable $x : \tau$ consists of a **name** x and a **type** τ (formally, one begins with a countably infinite set of variable names \mathcal{V} and a countably infinite set of types (again, formal symbols) Φ . For each type $\rho \in \Phi$ we take a copy of \mathcal{V} and denote this copy \mathcal{V}_ρ . These are the **typed variables**. A “variable” means a “typed variable”). The λ -calculus consists of **terms** which are built inductively using three **term formation rules**, see [1, Appendix A] but recall that each term has an associated type which is either an **atomic type** ρ or an **arrow type** $\rho \longrightarrow \tau$.

We define a category \mathcal{L}_Q whose objects are elements of the set $\Phi \amalg \{\mathbf{1}\}$ where $\mathbf{1}$ is a symbol we formally adjoin. A morphism $\tau \longrightarrow \sigma$ where $\tau, \sigma \in \Phi$ is an equivalence class of terms t of type $\tau \longrightarrow \sigma$, and a morphism $\mathbf{1} \longrightarrow \tau$ is an equivalence class of terms t of type τ . The equivalence relation which defines these equivalence relations are given by three **term reduction** rules: **α -conversion** [1, Appendix A, page 62] **β -reduction** [2, Definition 1.3.2], and **η -expansion** [2, Definition 1.3.9]. These reductions provide the *dynamic*

content of the simply typed λ -calculus. The category \mathcal{L}_Q is the “computer science” component of the Curry-Howard equivalence.

On the other hand, given a sequence \mathbf{q} of named variables, we define a category $\mathcal{S}_{\mathbf{q}}$ to consist of intuitionistic sequence calculus proofs. More precisely, the objects are a list of variables (which should be taken to be the same set of variables used to define \mathcal{L}_Q) along with a formal element $\mathbf{1}$ adjoined. A morphism $\rho \longrightarrow \tau$ in $\mathcal{L}_{\mathbf{q}}$ is an equivalence class of proofs of $\rho \supset \tau$ (ie, the proposition ρ implies the proposition τ) under the hypotheses \mathbf{q} and a morphism $\mathbf{1} \longrightarrow \rho$ is an equivalence class of proofs of the ρ under hypotheses \mathbf{q} . The appropriate equivalence relation here is much more difficult to explain and justification for the generators of this relation was part of the contribution of the paper [1].¹

Thus we have described two categories, one of which involves morphisms which arise naturally in theoretical computer science, and the other involves morphisms which arise naturally in formal logic. That these two categories are equivalent [1, Theorem 4.15] is a remarkable result which provides motivation to consider these two fields from the perspective of the other.

It is the author’s opinion that the most interesting part of this equivalence is the relationship between the *dynamics* of both systems. How can we understand these dynamics mathematically? This question leads to the first of two projects which provide the work which I have done during the second year of my PhD. This project will be crudely referred to as “the French project” as it was carried out in France under the supervision of Thomas Seiller, and indeed is very “French” in nature; there are many researchers in France who are interested in the dynamics of the aforementioned systems. For similar reasons, I refer to the project I have done mostly in collaboration with Murfet as “the Australian project”.

2 The French project

How can we understand the dynamics of the λ -calculus? We make a technical note here that we will now consider the **untyped** λ -calculus (where earlier we considered the *simply typed* λ -calculus). We have particular interest in the terms of these two systems, of which those in the simply typed λ -calculus

¹To the authoritative reader: this paper is work which was done by Daniel Murfet and myself *before* I started my PhD. The main reason why this has been mentioned here is to provide appropriate context for what comes next.

provide a subset of those in the untyped λ -calculus. An example of a term which exists in the untyped λ -calculus but not in the simply typed λ -calculus is the fixed point

$$\Delta := (\lambda x.xx)(\lambda x.xx) \tag{1}$$

We notice that this term reduces under β -reduction to itself, which obscures the separation between “data” and “functions of data”. The blurring of this line is a key feature of the untyped λ -calculus (after all, in the simply typed λ -calculus, the distinction between terms of atomic type and terms of arrow type provides a sharp distinction between data and functions of data).

It was Girard’s idea [3] to consider each term in the untyped λ -calculus as a *normal functor*, that is, a functor $F : \underline{\text{Set}}^A \rightarrow \underline{\text{Set}}^B$ (where A, B are sets, ie, discrete small categories) where F preserves wide pullbacks and filtered colimits. A crucial part of this paper is the establishment between three different criteria which are equivalent to F being normal. See [3, Theorem 2.8] but note that there is a typo in the statement, part (ii) should read “ F is isomorphic to an analytic functor”.

Using this Theorem, Girard defines a denotational model of the λ -calculus. Whilst doing so, he notices the presence of extra structure in the model than what appears in the syntax (the λ -calculus), and so extends the syntax to a new system which is now known as *linear logic*. Indeed, this logic is central in the intersection between theoretical computer science and mathematical logic, and features connectives which would be interesting to have rich mathematical models of. We return to this point later.

I read this paper very thoroughly, and I have written notes for more the main Theorem and the model. ²

While reading this paper, it occurred to me that the model is over complicated and features ideas which could be made to be more transparent if we worked in a more concrete setting. Thomas and myself, along with another researcher Morgan Rogers, have worked on “deategorifying” Girard’s model and recently have made progress on this, and are currently writing this up for publication. This new model is partially written in Section 7 of [4]. This new model is one of the most significant contributions I have been involved in throughout this

²These notes do not appear on my webpage at the moment because I am still working on them, but I have attached them as a separate document so that the authoritative reader can observe the work I have done. This document is 43 pages long, and features extensions of lemmas and theorems originally stated in [3], for example, Section 3 of my notes works with functor categories $\underline{\text{Set}}^A$ where A is a small category, where in [3] on the case where A is a small discrete category is considered.

PhD, and was discovered during my second year. I am largely indebted to the help of Seiller and Rogers, but I point out that a refinement to the definition of the App function ([4, Definition 7.2.1], in particular, the adjunction of Definition [4, Definition 7.1.1]), many of the core proofs used in this section and all of the proofs of the other sections, as well as the idea to search for such a model in the first place, were ideas of mine influenced by Girard’s paper [3].

3 The Australian Project

In Section 2 we described a *denotational* model of the untyped λ -calculus. What makes this a *denotational* model is the fact that terms which are equivalent under $\alpha\beta\eta$ -equivalence are modelled as the exact same function. That is, the model is unaware of the dynamics. Generally speaking, we have a good understanding of the conversion rules as static objects, but how can we understand these rules as *dynamic* objects?

This question, influenced by the ideas of, again, Girard, but in a different context [5], lead Murfet to discover *cut-systems*, which provide a refinement to horizontal composition in the bicategory of Landau-Ginzburg models (as well as other contexts). Murfet and myself believe that cut-systems should provide a model for linear logic, and moreover, a model which rather than interpreting proofs which are equivalent to each other as the *same* object, instead interprets them as different objects which are related to each other via the dynamics available to any cut-system (ie, the splitting of idempotents).

The Landau-Ginzburg category can be observed through many different lenses. One of these lenses is that of Quantum Error Correction Codes, considered as interactive physical systems. Understanding the aforementioned connectives of linear logic inside this setting is a core goal of this PhD. Murfet and myself are continuing to work towards this goal, but currently we have found a model for a sub-fragment of linear logic using Quantum Error Correction Codes, which is currently being written up for publication, and a model for this same sub-fragment using coordinate rings, which was discovered, written up, and submitted during my second year. See [7].

This model is interesting because it relates two distinct *and pre-existing* dynamical structures in mathematics. One of them is Buchberger’s Algorithm, which calculates Gröbner bases from generating sets of ideals in certain polynomial rings, and cut-elimination, which corresponds under the Curry-

Howard correspondence to β -reduction (and so may be thought of as a dynamics of computation).

Whilst working on this paper, I noticed that cut-elimination can in fact be thought of in two different ways. One could consider an instance of the cut-rule in a proof and *reduce* this, by commuting it with another deduction rule, hence reducing the distance in the proof between this cut rule and an axiom rule, or one could consider eliminating all instances of the cut-rule simultaneously in one, big *normalisation* step. The standard presentation of Buchberger's Algorithm, given in say [8], is capable of performing normalisation, whereas a variation on Buchberger's Algorithm is needed in order to perform *reduction*.

4 Other academic work

This year I ran two online seminars and one online course at *metauni*, an online community where volunteer researchers present seminars to participants. The first of these was a proof of a key result used in Gödel's First Incompleteness Theorem, that the primitive recursive functions are representable. The notes for these are available on my webpage [9].

The second of these was on the Ax-Grothendieck Theorem, which states that any injective polynomial $f : \mathbb{C}^n \longrightarrow \mathbb{C}^n$ is surjective. This result is interesting due to the proof rather than the result, as the proof uses model theory techniques. This seminar was an opportunity for participants to see how formal logic can be used to prove mathematical statements (ie, logic is more than just philosophical musings).

The course was an introduction to category theory, notes for these can be found on my webpage too [11].

References

- [1] Murfet, Troiani. *The Gentzen-Mints-Zucker Duality*. <https://arxiv.org/abs/2008.10131>
- [2] Sorensen. *Urzyczyn Lectures on the Curry-Howard Isomorphism* View series: Studies in Logic and the Foundations of Mathematics July 4 2006.

- [3] Girard. *Normal functors, power series and lambda-calculus*. Lecture Notes in Computer Science, 62, Springer, Berlin (1978), pp. 72-89
- [4] Rogers. Seiller. Troiani. *Analytic functors and λ -calculus*. To appear.
- [5] Girard. *I Interpretation of system F*. Studies in Logic and the Foundations of Mathematics Volume 127, 1989, Pages 221-260
- [6] Murfet. *The cut operation on matrix factorisations*. <https://arxiv.org/abs/1402.4541>
- [7] Murfet. Troiani. *Elimination and cut-elimination in multiplicative linear logic* <https://arxiv.org/abs/2207.10871>
- [8] Cox. Little. O'Shea. *Ideals, Varieties, and Algorithms*. Springer Cham, 17 October 2016
- [9] Troiani. *Gödel's First Incompleteness Theorem*, <https://williamtroiani.github.io/Notes/FirstIncompletenessTheorem.tex.pdf>
- [10] Troiani. *Ax-Grothendieck Theorem*. <https://williamtroiani.github.io/Notes/ax-grothendieck.pdf>
- [11] Troiani. *Shadows of Computation*. <https://williamtroiani.github.io/Teaching.html>