The diagrammatic calculus of a 2-category

Greta Coraglia[1] [0000−0002−9068−2345] and Ivan Di Liberti[2] [0000−0003−1787−4858]

[1] University of Genova, Genova, Italy
coraglia@dima.unige.it
[2] Stockholm University, Stockholm, Sweden
diliberti.math@gmail.com

Abstract. We introduce judgemental theories and their calculi as a general framework to present and study deductive systems. Judgements are computed as functors and rules as 2-cells. As an exemplification of the expressivity of our approach, we encode both dependent type theory and natural deduction as examples of judgemental theories. Our analysis sheds light on both the topics, providing a new point of view.

Keywords: categorical logic · deductive systems · dependent type theory · natural deduction · diagrammatic calculus.

The aim of this talk is to present a unified, category-based approach that accommodates diverse views on the topic of *deduction*. The effort required in order to do so turns out to be extremely fruitful, and in fact it can be used, for instance, to obtain novel results about the algebraic treatment of type constructors in dependent type theory.

One of the motivating examples is to give a theoretical framework in which the two following rules, which stand on very conceptually different grounds, can be compared.

$$\text{(Subs)}\ \frac{\Gamma \vdash a : A \qquad \Gamma.A \vdash B}{\Gamma \vdash B[a]} \qquad\qquad \text{(Cut)}\ \frac{x;\Gamma \vdash \phi \qquad x;\Gamma,\phi \vdash \psi}{x;\Gamma \vdash \psi}$$

One can traditionally be found in type theory [8], the other in proof theory [12]: despite their incredibly similar look, and the somehow parallel development of the respective theories in the same notational framework, there are some philosophical differences between the interpretation of the symbols above. Not only that, but the same "⊢" symbol seems to regard only statements of one kind `formula` in the case of (Cut), while it pertains to two - `term` and `type` - in that of (Subs).

Of course one could argue that these different points of view are mostly philosophical, and, in particular, the deep connection between proof theory and type theory has been studied for a while: its development falls under the paradigm that is mostly known as *propositions-as-types* [10], [13]. We believe that our theory bears witness to this and, in fact, gives it a categorical backbone.

Rebooting some ideas from [7], we develop what we call *judgemental theories*. Going back to the example of (Subs) and (Cut), we intuitively see how they both fit the same paradigm, in the sense that we could read both as instances of the following syntactic string of symbols

$$(\triangle)\ \frac{\heartsuit \ \vdash \blacksquare \qquad \square \vdash \clubsuit}{\heartsuit \ \vdash \spadesuit}$$

which we usually parse as: *by* $\triangle$, *given* $\heartsuit \ \vdash \blacksquare$ *and* $\square \vdash \clubsuit$ *we deduce* $\heartsuit \ \vdash \spadesuit$. Our theory allows for a coherent expression of all such strings of symbols, and shows how a suitable choice of *context* either produces (Subs) or (Cut): in the language of judgemental theories, in fact, we will show that they are coded as on the left and as on the right, respectively,

$$\begin{tikzcd} \dot{\mathbb{U}}.\Delta\Sigma\mathbb{U} \arrow[rr] \arrow[dr, bend right] & \arrow[d, phantom, "\swarrow"] & \dot{\mathbb{U}} \times \mathbb{U} \arrow[dl, bend left] \\ & \mathbb{U} & \end{tikzcd} \qquad \begin{tikzcd} \mathbb{E}d.dS\mathbb{E} \arrow[rr] \arrow[dr, bend right] & \arrow[d, phantom, "\swarrow"] & \mathbb{E}d.d\mathbb{E} \arrow[dl, bend left] \\ & \mathbb{F} & \end{tikzcd}$$

where $\mathbb{U}$ acts as a category for types, $\dot{\mathbb{U}}$ for terms, $\mathbb{F}$ for formulae, and $\mathbb{E}$ for pairs of formulae, in which the first entails the second. The other letters stand for functors regulating the relations in which all of these are.

We claim that, depending on the subject it pertains to, it is appropriate to interpret "'$\vdash$" with different categorical objects, in our case, functors (most of the time we will ask for them to be Grothendieck fibrations, though, so that substitution in properly dealt with): the domain of the functor acts as a universe for objects one wants to *judge on*, while the codomain collects contexts. We will say, for example, that a type A is in context Γ when the functor classifying types maps A to Γ. The functor classifying terms will need to be a different one - after all, terms are not types - although still on the same context category.

A rule in a deductive system, then, is an object that needs to relate two different functors, for example we need to be able to express some sort of typing rule: if a is a term in context Γ, then its type is in fact a type in context Γ. That is interpreted by lax commutative triangles, where one edge interprets the premise of a rule, one interprets the consequence, and the direction of the third edge describes the "direction" of the deduction. It is worth mentioning that the *lax* aspect of it all will be key in our theory, making it 2-dimensional on a categorical point of view.

Following this basic intuition, we show that, provided that one can combine rules in a coherent way, (e.g. we might be interested in considering pairs of judgements which are related in some sense, meaning that we want to be able to compute pullbacks of these functors classifying judgements) one can exhibit codes for all structural rules in both examples, and only starting from a few basic judgement classifiers.

While the framework of judgemental theories and the process of formalization of a given deductive system is purely syntactical - in the sense that we are not interested in what a given judgement or rule should *mean*, only in the symbols involved - one could wonder whether such an effort produces results close to categorical structures traditionally used as models. This is, in fact, the case.

- In the case of dependent types, we show how traditional categorical models ([2], [3], [5], [6], [11]) all fit into our paradigm. Moreover, properties that were considered *external*, such has having dependent sums for CE-systems [1], are *internalized* in our framework, so that one can quantitatively compare different models.
- In the case of natural deduction, we explicitly provide a proof of cut elimination (or normalization, see [9]) and in our setting it is particularly nice to see its interaction with the categorical infrastructure. Moreover, in this context, one can see how deduction trees turn into cones and cocones.

Although this introductory presentation seems to focus mainly on theories of dependent types and natural deduction, we hope to make it clear that we have chosen only these two specific frameworks as an example of the expressive power of this theory. In fact, we hope to soon cover modal logic, linear logic, and more.

This exposition is based on [4].

References

1. Ahrens, B., Emmenegger, J., North, P.R., Rijke, E.: B-systems and C-systems are equivalent. ArXiv e-prints (2021)
2. Awodey, S.: Natural models of homotopy type theory. Mathematical Structures in Computer Science **28**(2), 241âĂŞ286 (2018). https://doi.org/10.1017/S0960129516000268
3. Cartmell, J.: Generalised algebraic theories and contextual categories. Annals of Pure and Applied Logic **32**, 209–243 (1986). https://doi.org/https://doi.org/10.1016/0168-0072(86)90053-9, https://www.sciencedirect.com/science/article/pii/0168007286900539
4. Coraglia, G., Liberti, I.D.: Context, Judgement, Deduction. ArXiv e-prints (2021)
5. Dybjer, P.: Internal type theory. In: Berardi, S., Coppo, M. (eds.) Types for Proofs and Programs. pp. 120–134. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)
6. Jacobs, B.: Comprehension categories and the semantics of type dependency. Theoretical Computer Science **107**(2), 169–207 (1993). https://doi.org/https://doi.org/10.1016/0304-3975(93)90169-T, https://www.sciencedirect.com/science/article/pii/030439759390169T
7. Jacobs, B.: Categorical logic and type theory. Elsevier (1999)
8. Martin-Löf, P., Sambin, G.: Intuitionistic type theory, vol. 9. Bibliopolis Naples (1984)
9. Negri, S., Von Plato, J.: Structural proof theory. Cambridge university press (2008)
10. SÃÿrensen, M.H., Urzyczyin, P. (eds.): Lectures on the Curry-Howard Isomorphism, Studies in Logic and the Foundations of Mathematics, vol. 149. Elsevier (2006). https://doi.org/https://doi.org/10.1016/S0049-237X(06)80499-4, https://www.sciencedirect.com/science/article/pii/S0049237X06804994
11. Taylor, P.: Practical Foundations of Mathematics. No. v. 59 in Cambridge Studies in Advanced Mathematics, Cambridge University Press (1999), http://www.paultaylor.eu/prafm/
12. Troelstra, A.S., Schwichtenberg, H.: Basic proof theory. No. 43, Cambridge University Press (2000)
13. Wadler, P.: Propositions as types. Communications of the ACM **58**(12), 75–84 (2015)