Polyadic Approximations and Space Complexity. I started developing the theory of polyadic approximations in my LICS 2012 paper (nowadays it also goes by the name "*rigid Taylor expansion*"). Apart from its deep connections with intersection types (POPL 2018) and its applications to concurrent process calculi (POPL 2019), this has interesting complexity applications, exemplified by *parsimonious logic* (a new paradigm for implicit implicit computational complexity) and a "programming-language-theoretic" proof of the Cook-Levin theorem (NP-completeness of SAT). The latter suggests the following "equation":

$$\frac{\text{linear/affine } \lambda\text{-terms}}{\text{higher-order programs}} = \frac{\text{Boolean circuits}}{\text{Turing machines}}$$

The part of Boris's Ph.D. thesis that he is developing with me is supposed to build, following the above analogy and standard complexity-theoretic lore (Borodin 1977), a type system for a "realistic" higher-order programming language (say, call-by-value PCF) which is able to express properties not just about execution time but also about space. In particular, this should lead to a machine-independent definition of space complexity classes directly on higher-order programs. I had started working on this also with Gabriele Vanoni (Bologna), but Covid-19 interrupted it.

Foundations of Differentiable Programming. A couple of years ago I started a collaboration with Michele Pagani (IRIF) and my former Ph.D. student Aloïs Brunel (Deepomatic) on *differentiable programming*. We introduced a program transformation, based on linear logic, for computing gradients via a higher-order generalization of so-called *backpropagation*, thus unveiling its logical structure and proving for the first time its correctness in a higher-order programming language (POPL 2020). In presence of conditionals, automatic differentiation (AD) methods may introduce errors, which intuitively occur at the "cusps" of if-then-else. The AD community has been aware of this issue for a long time (Speelpenning 1980). In the case of straight-line programs with conditionals and gotos, it was known that the problem is negligible, in the sense that the set of points on which AD fails is of measure zero (Joss 1976). With Pagani, we proved that this is the case for PCF too. Although expected, this result was unknown and its proof is far from trivial (polyadic approximations actually play a role in it!). This is currently submitted to POPL 2021.

Complexity of Cut-Elimination for Subsystems of MLL. This is an ongoing joint work with Anupam Das (Birmingham), Noam Zeilberger (LIX) and Tito. We are studying the complexity of the normalization problem (given two terms, are they β-equivalent?) for subclasses of linear λ-terms (namely *planar*, *bridgeless* or both). In the plain linear case, the normalization problem is known to be P-complete (Mairson and Terui 2003). We proved that it remains P-complete for planar terms. We are close to proving P-completeness for bridgeless terms too. For planar-bridgeless, we don't know yet.

A Functorial Approach to Structural Complexity. This is the thing I am spending most of my time on right now, and it is the most speculative. A couple of years ago, while learning about *descriptive complexity* (Immerman 1999), I realized that if you reformulate it in terms of categorical logic, it looks super nice. In particular, the notion of *quantifier-free reduction* of descriptive complexity organizes itself very well around a category **Bool** whose objects are certain logical theories and whose morphisms are logical interpretations between them. This category is equivalent to a category of *finitary Boolean categories* (Carboni, Lack, Walters 1993) and (isomorphism classes of) finite-product-and-finite-coproduct-preserving functors.

At this point, enters "functorial geometry", which I've been learning from the nLab and notes of a Master course by Bertrand Toën (Toulouse). In there, he presents a notion of *geometric context* as a site $\mathcal{C}$ (category + Grothendieck coverage) plus a family of arrows of $\mathcal{C}$ satisfying certain axioms, and defines *abstract varieties* as sheaves on $\mathcal{C}$ which are "locally representable". He then shows how, for suitable choices of geometric contexts, different well-known notions of variety arise (*e.g.* topological, smooth). In particular, if one takes $\mathcal{C} = \mathbf{CRing}_{\mathrm{fp}}^{\mathrm{op}}$ (the opposite of the category of finitely presentable commutative rings) with the so-called "Zarisky coverage", the abstract varieties are just *schemes* (the central object of algebraic geometry). Now, if you take $\mathcal{C} = \mathbf{Bool}^{\mathrm{op}}$ with a suitable coverage, the abstract varieties you get seem to be very interesting from a complexity viewpoint! For example, things like the Cook-Levin theorem look a lot like theorems about the existence of *moduli spaces*, a fundamental concept in geometry. To put it very boldly (who cares, no-one outside our group will read this!) *structural complexity theory seems to be algebraic geometry with Boolean categories in place of commutative rings*.

So, since the beginning of the lockdown, we've organized a long-distance working group with Tom Hirschowitz (Chambéry), who is interested in functorial geometry because he's studying synthetic differential geometry (which is a form of it) and Carlos Simpson (Nice), who is an algebraic geometer (and former advisor of Toën—it's a small world!), during which we've been trying (Tom and I) to learn a bit about sheaf cohomology, which seems to be the next thing people do when they have a notion of variety like Toën's. All this while I try to understand bits and pieces of algebraic geometry and dig deeper into categorical logic. As you can imagine, I haven't gone very far with cohomology, I'm still stuck with the basics... so I guess I'll be doing this for a *long* time to come!