What is Manzonetto doing? Is he working enough?

Observational equivalences. In collaboration with Breuvart, Kerinec, and Barbarossa. See Barbarossa's sheet.

Call-by-value λ-calculus. The call-by-value λ-calculus has been introduced a long-time ago (Plotkin, 1975) its theory of program approximation is rather involved compared to its call-by-name version. This is a pity because such a paradigm is more similar to what is actually employed when implementing a functional programming language (usually, some form of call-by-need evaluation). The state of the art in 2004 is described in Paolini and Ronchi Della Rocca's book "The Parametric Lambda Calculus — A Metamodel for Computation." This research area has is knowing a second youth after the work by Carraro and Guerrieri, exhibiting the Linear-logic structure behind the calculus — on the syntactic side they proposed to use Regnier's σ-reductions to unblock some redexes previously considered as stuck, on the semantic side they defined relational CbV models, equivalently, non-idempotent intersection type systems. Recently, with Kerinec and Pagani we analyzed Ehrhard's CbV Taylor expansion and were able to propose for the first time a notion of CbV Böhm trees. With Pagani and Ronchi we introduced "lifted" relational intersection type systems for CbV λ-calculus enjoying adequacy. In collaboration with Emma and Ronchi, we are proving that in these models a characterization of solvability is at hand, and that type-inhabitation remains decidable (this is not trivial, as there is an idempotent type used to perform the lifting.)

Intrigila machines. It is well known that λ-calculus is Turing-complete, since all computable numerical functions can be equivalently written as λ-terms or as a Turing-machine program. However, Turing-machines are not well-suited for modelling the complexity of λ-calculus, namely its higher-order features. Together with Intrigila and Della Penna, we introduce *addressing machines* that are conceived for implementing λ-terms in the simplest ways. Every machine has an associated address. Every machine disposes of a finite number of registers (for storing addresses) an input-tape (for reading addresses) and a program (with 3 instructions: load the head of the tape in a register, apply two registers, call another machine whose address is store in a register). We show that this give rise to a rather syntactic model of λ-calculus whose theory is probably β-convertibility. Interestingly, for proving consistency we need a transfinite (but countable) induction. This is due to the presence in the system of a rule having countably many premises. For future works, we plan to enrich these machines with probabilistic and concurrent features. For comparisons with CuCh machines, call Stefano Guerrini.

Unrelated. Nicolas Münnic joined our group as a PhD student jointly supervised by Breuvart and myself. He will start by working on weighted relational semantics of PCF, then continue on Flavien's project.

Really unrelated. In 2017, Manzonetto embarked in the adventure of writing, jointly with Barendregt himself, a satellite book of the λ-calculus bible "The λ-calculus — Its syntax and semantics". The idea is to collect the solutions of all open problems proposed in the original book, present them in a uniform way, and make them more accessible. Currently, 250 pages, roughly 50% left to do.