Let us take a look at CompCert's SimplExpr module. In the last part of MPRI 2.4, we explain that this is a monadic program, allowing Xavier Leroy to (safely) access an effectful gensym operation in an otherwise pure programming language.

Now, let us look at its associated correctness proof and, in particular, the supporting specification lemmas. In MPRI 2.36.1, we learn that the "separation properties" alluded to in the specification file are indeed the ingredients necessary to setup a separation logic.

However, it does not take a semester of MPRI 2.7.2 to notice that these structures (monads for programming, separation logic for reasoning) intertwine in the proof: the proof carries over the purely functional representation of the monadic program, thus managing the monadic, logical and sub-structural invariants all at once. Automating such a proof is a daunting task, and rightfully so!

This PhD project aims at combining concepts and techniques from these three domains in order to:

1. Streamline the implementation and proof of the SimplExpr module. A measure of success will be given by the length of the resulting proof as well as its resilience to changes;

2. Generalize the approach to the other monads used in CompCert. A measure of success will be given by the ease with which new use cases are supported;

3. Develop a formal understanding of separation logics for monadic programs. A concrete result of this work would be a generic library for specifying monads and their logics as well as offering support for automated reasoning.