Internship proposal:
Formal analysis of legislation through the Catala DSL

Advisor:
Denis Merigoux
Team Prosecco – Inria
denis.merigoux@inria.fr

Head of department:
Karthikeyan Barghavan
Team Prosecco – Inria
karthikeyan.barghavan@inria.fr

Location of the internship: Paris, France

Context The growing complexity of legislative texts like the French tax code or the more recent European General Data Protection Regulation makes it harder and harder not only to businesses and persons to comply with the law, but also for legislators themselves to reform and craft new legislation compatible with the existing texts. Moreover, some pieces of legislation specify computations and processes that should be defined in a rigorous way: what amount of taxes you should pay according to your revenues, for how long a user's data can be stored and who can access it. Statistical learning techniques can be used to infer some knowledge from the text of the law [10], but they do not provide a systematic and high-assurance way of analysing it [8]. Rather, we prefer to leverage the insights of Kowalski et al. [9] or more recently Lawsky [4, 3], and propose a more formal approach to the problem. Catala [5] is a DSL suitable for the specification of computation-heavy legislation in a literate programming style. See `https://catala-lang.org/` for a presentation of the language.

Goals While the core semantics of Catala have already been formalized, the goal of the internship would be to connect the existing Catala compiler to existing formal verification tools. Many properties could be proven about the legislation, ranging from internal coherence (decrees comply with laws) to insights about impact of reforms. See [6] for a proof of concept for the French socio-fiscal system marginal rate. Picking the right tool to prove those properties will be the challenge of the internship, with choices ranging from abstract interpretation (for instance using MOPSA [7]) or SMT solving (for instance using Z3 [2]) to deductive verification (for instance using Coq [1] of F* [11]).

The design process of the tooling will be done in collaboration with law academics from Université Panthéon-Sorbonne as well as the Chicago Priztker School of Law, as interaction with the users of the system is crucial to guide the research priorities. The running case study for this work would be the implementation of the French family benefits.

Qualifications This internship of 6 months or less would be hosted by the team Prosecco at Inria Paris. **This internship will probably not lead to a PhD offer in the Prosecco team, although the topic is raising academic attention in France**. The preferred qualifications for the student at the beginning of the internship would be:

- fluency with the OCaml programming language;
- experience at writing a compiler for a (preferably functional) language;
- some experience with at least one of the formal methods techniques described above;
- motivation for interacting with potential users of the language (law people).

References

[1] Yves Bertot and Pierre Castéran. *Interactive theorem proving and program development: Coq'Art: the calculus of inductive constructions*. Springer Science & Business Media, 2013.

[2] Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[3] Sarah B. Lawsky. Formalizing the Code. *Tax Law Review*, 70(377), 2017.

[4] Sarah B. Lawsky. A Logic for Statutes. *Florida Tax Review*, 2018.

[5] Denis Merigoux and Liane Huttner. Catala: Moving Towards the Future of Legal Expert Systems. working paper or preprint, September 2020.

[6] Denis Merigoux, Raphaël Monat, and Christophe Gaie. Étude formelle de l'implémentation du code des impôts. In *31ème Journées Francophones des Langages Applicatifs*, Gruissan, France, January 2020.

[7] A. Miné, A. Ouadjaout, and M. Journault. Design of a modular platform for static analysis. In *Proc. of TAPAS'18*, LNCS, page 4, 28 Aug. 2018.

[8] Marcos A Pertierra, Sarah Lawsky, Erik Hemberg, and Una-May O'Reilly. Towards formalizing statute law as default logic through automatic semantic parsing. In *ASAIL@ ICAIL*, 2017.

[9] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory. The british nationality act as a logic program. *Commun. ACM*, 29(5):370–386, May 1986.

[10] Harry Surden. Machine learning and law. *Wash. L. Rev.*, 89:87, 2014.

[11] Nikhil Swamy, Catalin Hritcu, Chantal Keller, Aseem Rastogi, Antoine Delignat-Lavaud, Simon Forest, Karthikeyan Bhargavan, Cédric Fournet, Pierre-Yves Strub, Markulf Kohlweiss, Jean-Karim Zinzindohoué, and Santiago Zanella-Béguelin. Dependent types and multi-monadic effects in F*. In *43rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pages 256–270. ACM, January 2016.