



Proposition de stage: Gestion d'erreur avancée pour Catala

Emmanuel Chailloux, LIP6, emmanuel.chailloux@lip6.fr
Louis Gesbert, Prosecco – Inria, louis.gesbert@inria.fr
Denis Merigoux, Prosecco – Inria, denis.merigoux@inria.fr

Lieu du stage: Paris, France
Durée: approx. 6 mois
Niveau: M2 préféré / M1

Contexte Les messages d'erreurs du compilateur sont un élément clef de l'apprentissage et de l'interaction avec les langages de programmation. Un élément d'autant plus crucial lorsqu'il s'agit de DSL (*Domain Specific Language*, ou langage dédié) utilisant des abstractions spécifiques, et dont les utilisateurs n'ont pas toujours la programmation pour domaine premier.

Le langage dédié Catala [5] a pour objet la traduction de textes légaux, issus du droit fiscal ou de celui des prestations sociales par exemple, tout en respectant leur structure et en gardant une certaine clarté vis-à-vis d'un public de juristes. Il est étayé par une approche formelle inspirée par les travaux de Kowalski et al. [6] et plus récemment Lawsky [4, 3].

Objectifs Le but de ce stage est de développer l'outillage autour du langage Catala et d'assister l'équipe de développement du compilateur. Une première tâche sera de définir, en tant que paquet OCaml indépendant, une bibliothèque générique d'affichage d'erreurs claires et plaisantes à l'œil, en s'inspirant de ce qui est fait par la bibliothèque Rust Ariadne [1]. La-le stagiaire aura la charge d'étudier les différentes solutions d'affichage avancé sur le terminal utf-8 en OCaml (Format, utf8, ANSITerminal, lambda-term, etc.) ainsi que de définir l'API la plus appropriée pour solliciter l'affichage.

Dans le même temps, elle-il devra étudier les cas d'erreur du compilateur Catala, les uniformiser quand c'est nécessaire et les adapter à sa nouvelle API. Cela ira des messages d'erreurs de parsing (utilisant Menhir) aux erreurs de type, sans oublier les erreurs de syntaxe et autres alertes sur le code. Elle-il pourra être confronté-e directement aux utilisateurs pour étudier la qualité des messages produits.

Dans un second temps, des tâches plus avancées sur le compilateur lui-même sont envisagées, telles que l'agrégation d'erreurs en vue de retarder au maximum l'échec de la compilation, ou l'amélioration à la source de l'inférence de types afin de générer les messages les plus utiles possibles, dans l'esprit par exemple de [2].

Références

- [1] Joshua Barretto. Ariadne. <https://github.com/zesterer/ariadne>, 2022.
- [2] Arthur Charguéraud. Improving type error messages in OCaml. *Electronic Proceedings in Theoretical Computer Science*, 198 :80–97, dec 2015.
- [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, Nicolas Chataing, and Jonathan Protzenko. Catala : A programming language for the law. *Proc. ACM Program. Lang.*, 5(ICFP), August 2021.
- [6] 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.