



**SORBONNE
UNIVERSITÉ**

Rapport du projet

Evaluateur-Typeur de Lambda-Calcul

Saïd Mohammad ZUHAIR (21204924)

November 2024

1. Introduction

Ce rapport présente le projet d'évaluation et de typage de lambda-calcul. J'ai réalisé ce projet avec le langage ocaml, un utilisant une librairie de test qui s'appelle `alcotest` et le build tool `dune`.

2. Structure du projet

Le projet est structuré comme suit:

- `bin/`
 - `eval.ml` (binaire qui prend un fichier en entrée et affiche le résultat de l'évaluation)
 - `type.ml` (binaire qui prend un fichier en entrée et affiche le type de l'expression)
- `lib/`
 - `ast.ml` (définition de l'arbre syntaxique abstrait)
 - `eval.ml` (évaluateur)
 - `type.ml` (typeur)
 - `parser.mly` (parseur)
 - `lexer.mll` (lexer)
- `test/`
 - `lib/` (module utilitaire pour les tests)
 - `prog/` (liste de programmes de tester le parseur et le lexer)
 - `eval2.ml` (tests de la section 2)
 - `eval4.ml` (tests d'évaluation de la section 4)
 - `eval5.ml` (tests d'évaluation de la section 5)
 - `type3.ml` (tests de la section 3)
 - `type4.ml` (tests de typage de la section 4)
 - `type5.ml` (tests de typage de la section 5)

3. Évaluateur

- J'ai réussi à implémenter l'évaluateur jusqu'à la section 5.
- L'évaluateur implémente une stratégie d'évaluation **left-to-right call-by-value**.
- La mémoire est représentée par une liste d'associations d'un entier avec des pterm.

4. Typeur

- J'ai réussi à implémenter le typeur jusqu'à la section 5.2 (il manque la section 5.3).
- Les equations de types sont représentées par une liste de `(ptype * ptype)`.
- J'ai notamment passé beaucoup de temps à implémenter la règle `fixe`.

5. Parseur/Lexer

- J'ai implémenté un parseur et un lexer pour tous les constructeurs de l'évaluateur.
- J'ai fait le choix de construire un `fixe` que par un `let` qui est représenté par `let rec` dans le parseur.
- J'ai aussi implémenté un `fun` (lambda) qui peut prendre plusieurs arguments.
- Je n'ai pas eu le temps de faire des tests propre pour le parseur et le lexer mais j'ai essayé de lancer les programmes dans le dossier `test/prog/` et ils ont tous donné le résultat attendu.

6. Execution

- Pour exécuter en mode evaluation, il faut executer `dune exec bin/eval.exe <fichier>`
- Pour exécuter en mode typage, il faut executer `dune exec bin/type.exe <fichier>`
- Pour exécuter les tests, il faut executer `dune test`
- Pour exécuter les tests d'une section, il faut executer `dune exec test/<fichier>.exe`