

# Compte rendu projet scala

Sujet 1 : interpréteur pour le lambda-calcul

## Les structures de données :

Pour représenter les expressions de lambda calculs, deux choix étaient disponibles, l'utilisation de variables nommées et l'utilisation d'indices de Bruijn. Pour cela, j'ai utilisé des structures personnalisées de type enum, une pour chaque représentation.

### 1. Les expressions utilisant les variables nommées

Afin de représenter les expressions utilisant les variables nommées, j'ai mis en place trois types de termes :

- Les variables, représentées par leur nom (qui est une simple chaîne de caractères)
- Les abstractions, qui sont représentées par le nom de la variable qu'elle remplace, ainsi que par l'expression qu'elle donne
- Les applications, qui sont représentées par deux expressions, celle à gauche de l'application, et celle à droite.

Les expressions utilisant les variables nommées ne sont utilisées que pour l'affichage et la construction de lambdas termes, et ne sont donc pas utilisés pour les étapes de réduction et pour le calcul.

### 2. Les expressions utilisant les indices de Bruijn

Afin de représenter les expressions utilisant les indices de Bruijn, j'ai dû mettre en place quatre types de termes :

- Les variables libres, représentées par leur nom (qui est une chaîne de caractère)
- Les variables liées, qui sont représentées par leur indice en rapport à leur abstraction
- Les abstractions, qui sont représentées par le résultat de l'abstraction
- Les applications, qui sont représentées par deux expressions, celle à gauche de l'application, et celle à droite.

Les expressions utilisant les indices de Bruijn sont utilisées pour les Beta-réductions et donc pour toutes les opérations d'évaluation de l'expression.

## Les opérations sur les types d'expressions :

Différentes opérations ont été mises en place sur les expressions utilisant les variables nommées et sur les expressions utilisant les indices de Bruijn :

- La transformation en chaîne de caractère
- La possibilité de savoir si une expression est en forme normale
- La possibilité de passer des variables nommées aux indices de Bruijn et vice-versa
- Une étape d'évaluation paresseuse
- Une étape d'évaluation stricte
- Une évaluation paresseuse totale
- Une évaluation stricte totale
- Plusieurs fonctionnalités de test qui sont utilisées dans la classe AppTest.

Ces différentes fonctionnalités sont expliquées en profondeur dans le code de App.scala grâce aux scaladocs. Elles sont également utilisées dans son main ainsi que dans les nombreux tests de AppTest.