FIL A3 UV Sensibilisation à la recherche - 2017-2018
Titre : Inférence de type à la ML en Java
Encadrant : Rémi Douence

De nombreux langages de programmation utilisent la notion de typage statique.

Le système de type de Hindley Milner est un classique sur lequel repose de nombreux systèmes de type. Il permet de typer des programmes fonctionnels d'ordre supérieur et prend en compte le polymorphisme. Son algorithme d'inférence lui permet d'inventer le type le plus général pour un programme donné.

L'article original qui décrit cet algorithme date de 1982 et n'est pas facile à lire. Aussi on lui préférera un texte plus didactique : le chapitre 7 de Write You A Haskell (http://dev.stephendiehl.com/fun/). Ce document contient le code Haskell de l'algorithme d'inférence de type ainsi que des explications et exemples.

L'objectif de ce projet est de programmer cet algorithme en Java. Votre programme devra prendre en entrée l'arbre syntaxique d'une lambda-expression et retourner le type de cette expression, ou une erreur si l'expression n'est pas typable.

Vous devrez donc dans un premier temps lire le chapitre, comprendre l'algorithme et les exemples puis programmer en Java un code équivalent à celui du chapitre. Le chapitre contient deux algorithmes différents, vous implanterez la seconde version basée sur des contraintes qui est plus simple à comprendre.

La grammaires des programmes à typer est définie par six cas :

```
E ::= x  |  \x->E  |  E E  |  let x = E in E | 1 | True
```

Vous pourrez rajouter les autres cas afin de typer des fonctions comme factorielle ou fibonnaci.