Feuille de TD n° 1 (séance 2) : `ocamllex`

Exercice 1 :

1. Écrire un fichier `entiers.mll` permettant de lire des entiers séparés par des espaces, des sauts de ligne ou des caractères de tabulation. Ce fichier contiendra la définition d'une règle `entier_suivant` produisant un entier.
2. Écrire une fonction Caml `list_from_string : string -> int list` renvoyant la liste des entiers contenus dans la chaîne passée en paramètre, en utilisant la fonction `entier_suivant` produite par le fichier `entiers.mll`

Exercice 2 :

1. Écrire une description en `ocamllex` pour des nombres décimaux avec les options de notation suivantes :
   - le nombre peut ou non être précédé d'un signe + ou `-` ;
   - le premier chiffre ne peut être un 0 que s'il est immédiatement suivi d'un point ;
   - le symbole de séparation entre parties entière et fractionnaire est le point ;
   - s'il y a un point, le dernier chiffre de la partie fractionnaire ne peut être un 0.
2. Construire un automate non déterministe reconnaissant les chaînes représentant des nombres décimaux.
3. Construire un automate déterministe.

Exercice 3 :

1. Écrire un fichier `expr.mll` permettant de lire des expressions comprenant des entiers et les opérateurs `+` et `*`. Ce fichier comprendra la définition d'un type `token` et d'une règle `next_token` produisant un lexème (entier, addition, multiplication).
2. Écrire la définition d'un type `expression` permettant de représenter des expressions arithmétiques avec les opérateurs d'addition et de multiplication.
3. Écrire une fonction `eval : expression -> int` qui évalue l'expression passée en paramètre.