

Exercice 0

12 points

S'inscrire au cours et dans un groupe avant le 3 novembre à minuit, comme expliqué sur la page web du cours.

Exercice 1 (BNF)

6 points

Trouvez une grammaire non-contextuelle au format BNF pour décrire l'ensemble des mots sur l'alphabet $\{a, b\}$ qui contiennent un nombre impair de a (indiquez clairement le symbole initial).

Donnez également un automate fini déterministe qui reconnaît ce langage et une expression régulière qui le désigne.

Exercice 2 (Nombres binaires)

6 points

Trouvez une grammaire non-contextuelle au format EBNF pour les langages suivants :

1. Les nombres binaires strictement supérieurs à 1001 (9 en décimal).
2. Les nombres binaires multiples de 4.

Pour chacune des solutions, on ne considère que les nombres dont le chiffre de poids fort est 1.

Exercice 3 (Analyseur lexical)

6 points

La grammaire EBNF ci-dessous définit un langage régulier sur l'alphabet $\{0, 1, \dots, 9, .\}$. Son symbole initial est `float`. Transformez cette grammaire en une grammaire EBNF qui soit analysable par la gauche.

`digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"`

`float = {digit} "." digit {digit}
| digit {digit} "." {digit}`

Écrivez ensuite une fonction Scala qui termine sans erreur si et seulement si un mot donné appartient à ce langage.

Dans le code de la fonction utilisez la variable `char` pour obtenir le caractère courant, la fonction `nextChar` pour faire passer `char` au caractère suivant et la fonction `error` de Scala pour signaler une erreur. La fin du mot est atteinte lorsque `char` vaut `charEOF`.

Remarque : vous ne devez pas redéfinir la fonction `nextChar`.