

Exercice 0

15 points

S'inscrire au cours et dans un groupe avant le 29 octobre 2004 à minuit, comme expliqué sur la page web du cours.

Exercice 1 (BNF)

9 points

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

1. Tous les palindromes sur l'alphabet $\{a, b\}$. Un palindrome est un mot symétrique, possiblement vide, comme *abaaba*.
2. Tous les mots sur l'alphabet $\{a\}$ dont la longueur est un nombre premier strictement inférieur à 10.
3. Tous les mots sur l'alphabet $\{a, b\}$ de la forme $a^m b^n$ avec $m \geq 0$, $n \geq 0$ et $m + n$ pair.
4. Tous les mots sur l'alphabet $\{a, b\}$ qui ne contiennent pas le sous-mot *aba*.

Si vous utilisez plusieurs symboles non-terminaux, indiquez clairement lequel est le symbole initial.

Exercice 2 (langages réguliers)

7 points

1. Décrivez en une phrase le langage défini sur l'alphabet $\{a, b\}$ qui est produit par la grammaire suivante. S est le symbole initial.

$$\begin{aligned} S &= \epsilon \\ S &= a S b S \\ S &= b S a S \end{aligned}$$

Le langage décrit par cette grammaire est-il régulier ?

2. Parmi les langages de l'exercice 1, lesquels sont réguliers ? Pour ceux qui le sont, prouvez-le.

Exercice 3 (analyseur lexical)

7 points

La grammaire ci-dessous définit un langage régulier sur l'alphabet $\{a, b, c\}$. Son symbole initial est S . Donnez une seule règle EBNF non récursive définissant le même langage, puis écrivez une fonction Java qui retourne vrai si et seulement si un mot donné appartient à ce langage.

$$\begin{aligned} S &= Tc \\ T &= \epsilon \mid aT \mid bT \end{aligned}$$

Dans le code de la fonction utilisez la variable `ch` pour obtenir le caractère courant et la fonction `next()` pour faire passer `ch` au caractère suivant. La fin du mot est atteinte lorsque `ch` vaut EOF. Remarque : vous ne devez pas redéfinir la fonction `next()`.

Exercice 4 (EBNF)

7 points

Trouvez une grammaire EBNF non récursive qui spécifie la syntaxe des identificateurs du langage Scala dont la spécification est la suivante.

1. Un identificateur Scala est un mot non vide composé :
 - de lettres,
 - de chiffres,
 - du caractère `'_'`, et
 - de caractères spéciaux (par ex. `'+' , '=').`
2. Un identificateur ne peut pas commencer par un chiffre.
3. Dans un identificateur, un caractère spécial ne peut être suivi que par un caractère spécial et ne peut être précédé que par le caractère `'_'` ou par un caractère spécial.

Voici quelques exemples d'identificateurs Scala :

```
–  
a_b_c  
ab2  
ab_2  
a__b  
a_  
+  
a_+ =
```