ISEN
ALL IS DIGITAL!
LILLE

yncréa

CIR 3
Théorie des langages – DS – 06/11/2018
Pas de documents autorisés

Exercice 1 :

Ecrire les commandes permettant de retrouver dans un fichier « texte.txt » :

- Le mot « bonjour ». grep -E "bonjour" texte.txt
- Le mot « bonjour » sans respecter la casse. grep -E "(B|b)(O|o)(N|n)(J|j)(O|o)(U|u)(R|r)" texte.txt
- Toutes les formes du verbe « réussir » conjugué au présent de l'indicatif.
  grep -E "(réuss)((i)|(issons)|(issez)|(issent))" texte.txt

Exercice 2 :

Ecrire une grammaire sur l'alphabet **{a,b,c}** permettant de générer les mots du langage :

$$L = \{ a^n b^m c^k \mid m > n+k \}$$

S = ABC
A = aB | ε
B = Bb | ε
C = Bc | ε

Exercice 3 :

Construire un automate sur l'alphabet ***{a,b,c}*** ne contenant aucune paire de lettres adjacentes identiques (ex. L'automate ne reconnait pas les mots : abbc, bcacc ou baabcc).

Combien y a-t-il de mots de longueur 3 dans ce langage ?

Exercice 4 :

Construire un automate sur l'alphabet ***{0,1}*** qui permet de vérifier si une chaîne binaire, de longueur quelconque, contient le motif **10110011.** L'automate accepte la chaine binaire au premier motif rencontré.

Exercice 5 :

Soit le langage L sur l'alphabet ***{a,b}*** regroupant tous les mots ayant un nombre pair de ***a*** et ne contenant pas le motif ***bb*** (les ***b*** ne doivent pas se coller).

- Décrire le langage ***L*** par une grammaire régulière.
- Décrire le langage ***L*** par un automate minimal déterministe.

Exercice 6 :

Nous souhaitons concevoir un automate sur l'alphabet **{a,b}** reconnaissant les mots qui contiennent une sous-séquence « aa » ou « bb ».

- Montrez que l'automate n'est pas déterministe en justifiant avec un exemple.
- Transformez cet automate en automate déterministe en suivant la méthode vue en cours :