UBMA - Faculté des sciences d'ingénieur
Département d'informatique - 2$^{ème}$ année LMD

Date : juin 2009
Durée : 1h45

**N.B : TOUTE RÉPONSE DOIT ÊTRE JUSTIFIÉE**

# EXAMEN : THÉORIE DES LANGAGES

## EXERCICE 1 (3+1 PTS) : LES AUTOMATES À PILE

Soit $(X = \{a, b\}, \Pi = \{A, B, \triangleright\}, Q = \{0, 1, 2, 3, 4, 5\}, initial = 0, F = \{3\}, \delta)$ un automate à pile tel que $\delta$ est donnée par :

$\delta(0, a, \triangleright) = (1, A\triangleright)$ $\delta(2, b, A) = (2, \varepsilon)$ $\delta(4, b, B) = (4, BB)$ $\delta(5, \varepsilon, \triangleright) = (3, \triangleright)$
$\delta(1, a, A) = (1, AA)$ $\delta(2, \varepsilon, \triangleright) = (3, \triangleright)$ $\delta(4, a, B) = (5, \varepsilon)$
$\delta(1, b, A) = (2, \varepsilon)$ $\delta(0, b, \triangleright) = (4, B\triangleright)$ $\delta(5, a, B) = (5, \varepsilon)$

1. Dire si les mots suivants sont reconnus : $ba, aba, aabb$.
2. Quel est le langage reconnu par cet automate ? Que faut-il rajouter à l'automate pour que $\varepsilon$ soit reconnu ?

## EXERCICE 2 (4 PTS) : CALCUL AVEC LES AEF

Dans le système de numération à base 3, les entiers sont écrits en utilisant les chiffres 0, 1 et 2. Ainsi, le nombre $(0112)_3$ (en base 3), par exemple, est l'équivalent de $2 \times 3^0 + 1 \times 3^1 + 1 \times 3^2 + 0 \times 3^3 = 14$ en décimal.

1. donner un AEF reconnaissant tous les nombres entiers, multiples de 2, écrits en base 3 ;
2. <u>déduire</u> alors une propriété des nombres entiers, multiples de 2, écrits en base 3.

## EXERCICE 3 (6 PTS) : LES GRAMMAIRES ET LES LANGAGES RÉGULIERS

Soit la grammaire suivante : $G = (\{[\,,\,]\,,\,:\,,\,a\}, \{S, T\}, S, \{S \to [T], T \to a | a : T\})$ générant le langage $L(G)$ (les terminaux sont [, ], : et $a$).

1. Donner trois mots dérivés à partir de G (donner les chaînes de dérivation).
2. Donner le type de cette grammaire.
3. Quel est le langage généré par G ? Montrer qu'il est régulier.
4. Donner un AEF qui reconnaît $L(G)$.
5. Donner une grammaire régulière à gauche générant $L(G)$.

## EXERCICE 4 (7+1 PTS) : LES PRÉFIXES ET LES AEF

1. Soit $L_1 = b^*ab^*$. Donner un automate déterministe et minimal reconnaissant les mots de $L_1$.
2. Quel est le langage $pref(L_1)$ (le langage des préfixes de $L_1$) ? Donner un AEF déterministe et minimal qui le reconnaît.
3. A la lumière de la première et la deuxième questions, dire comment utiliser l'automate d'un langage régulier $L$ pour obtenir celui de $pref(L)$. Expliquer votre proposition.
4. La transformation ne fonctionne pas si l'automate comporte des états improductifs. Dire pourquoi (**rappel** : un état est dit *improductif* s'il n'existe aucun chemin entre cet état et un état final de l'automate).
5. Donner une condition sur l'automate d'un langage régulier $L$ pour que $L = pref(L)$.

# CORRIGE DE L'EXAMEN DE TL : JUIN 2009

**Exercice 1 ( pts):**

1-

$(0,b,\triangleright) \rightarrow (4,a,B\triangleright) \rightarrow (5,\varepsilon,\triangleright) \rightarrow (3,\varepsilon,\triangleright)$, plus aucune transition possible. Le mot est accepte car il a été entièrement lu et l'état 3 est final ( ).

$(0,a,\triangleright) \rightarrow (1,b,A\triangleright) \rightarrow (2,a,\triangleright)$, plus aucune transition possible. Le mot est rejeté car il n'a pas pu être lu entièrement ( ).

$(0,a,\triangleright) \rightarrow (1,a,A\triangleright) \rightarrow (1,b,AA\triangleright) \rightarrow (2,b,A\triangleright) \rightarrow (2,\varepsilon,\triangleright) \rightarrow (3,\varepsilon,\triangleright)$, *plus aucune transition* possible. Le mot est accepté car il a été entièrement lu et l'état 3 est final ( ).

2- Le langage reconnu est $\{a^nb^n, n>0\}+\{b^na^n, n>0\}$ ( )

Pour reconnaître $\varepsilon$, il faut rajouter : $\delta(0, \varepsilon,\triangleright)=(3, \triangleright)$ ( )

**Exercice 2 ( pts):**

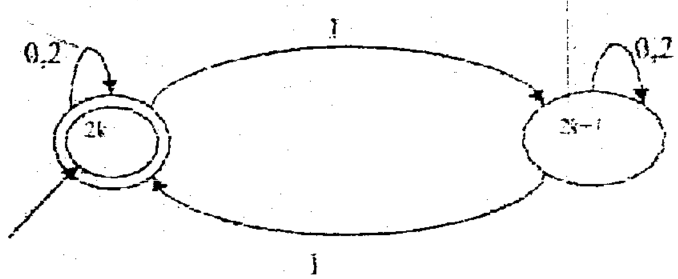
1- Les nombres entiers en base 3 s'écrivent à l'aide de 0, 1 et 2. Si on prend un nombre entier *n* écrit en base 3, alors l'insertion d'un chiffre à droite de *n* correspond aux opérations suivantes ( ):

| Insertion d'un | Résultat |
|---|---|
| 0 | 3n |
| 1 | 3n+1 |
| 2 | 3n+2 |

Les nombres entiers peuvent être pairs impairs, ils possèdent alors les formes 2k ou 2k+1.
Selon la forme, on obtient alors le tableau suivant :

| *Forme* | 0 | 1 | 2 |
|---|---|---|---|
| 2k | 6k=2k' | 6k+1=2k'+1 | 6k+2=2k' |
| 2k+1 | 6k+3=2k'+1 | 6k+4=2k' | 6k+5=2k'+1 |

On obtient alors l'automate suivant ( ):




2- D'après l'automate, les nombres entiers multiples de 2 écrits en base 3 possèdent un *nombre pair de 1* ( ). *Une autre description est possible : un nombre entier en base 3 est* multiple de 2 si la somme de ses chiffres est paire.

**Exercice 3 ( ):**

1- les mots sont [a], [a : a], [a : a : a]

S→[T]→[a]

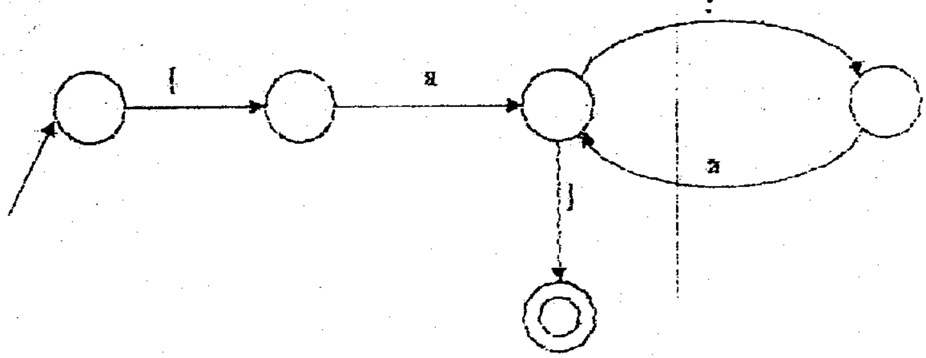
S→[T]→[a : T]→[a : a]

S→[T]→[a : T]→[a : a : T]→[a : a : a] ( )

2-Cette grammaire est hors-contexte ( ).

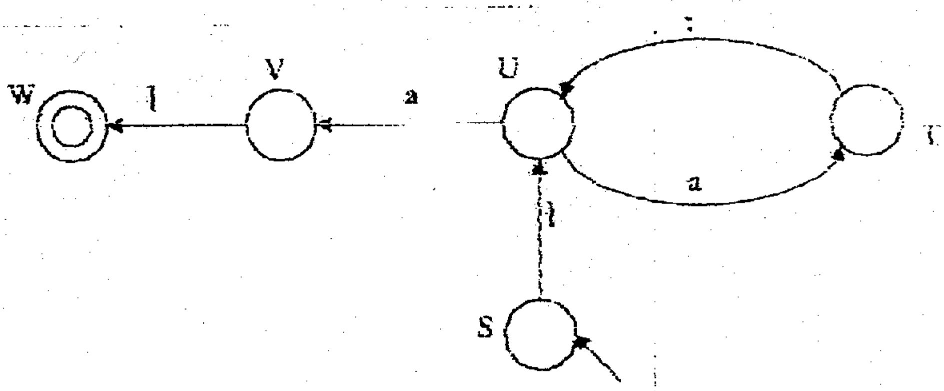
3- Le langage généré par cet automate est : [a(:a)*]. C'est un langage régulier vu qu'il existe une expression régulière qui le dénote ( ).

4- ( )




5-

On construit d'abord l'automate du langage miroir de L(G) :




La grammaire régulière à droite du langage miroir de L(G) est :

({[,], :,a},{S,T,U,V,W},S,{S→]U, T→ :U, U→aT|aV, V→[W, W→ε}) ainsi la grammaire régulière gauche qui génère L(G) est :

({[,], :,a},{S,T,U,V,W},S,{S→U], T→ U :, U→Ta|Va, V→W[, W→ε}) ( )

**Exercice 4 ( ):**

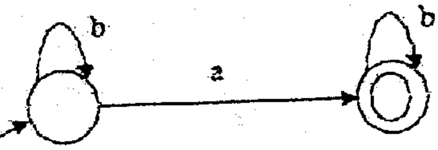
1- L1=b*ab*

L1 || a= (b* || a).ab* | f(b*) (ab* || a)=b*=L2

L1 || b= (b* || b).ab* | f(b*) (ab* || b)=b*ab*=L1

$L2 \| a = \emptyset$
$L2 \| b = b^* = L2$

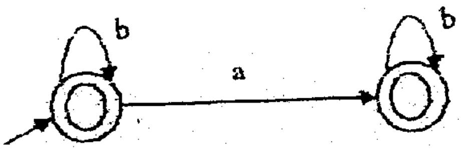
On obtient alors l'automate suivant ( ):




2- $L=\{b^n a b^m, n,m>=0\}$ → $pref(L)=\{b^n, n>=0\} + \{b^n a, n>=0\} + \{b^n a b^m, n,m>=0\}$
→ $pref(L) = b^* \mid b^*ab^*$ ( )

$L3 = b^* \mid b^*ab^*$
$L3 \| a = (b^* \| a) \mid (b^* \| a).ab^* \mid f(b^*) (ab^* \| a) = b^* = L4$
$L3 \| b = (b^* \| b) \mid (b^* \| b).ab^* \mid f(b^*) (ab^* \| b) = b^* \mid b^*ab^* = L3$
$L4 \| a = \emptyset$
$L4 \| b = b^* = L4$

On obtient alors l'automate suivant ( ) :




3- Pour obtenir l'automate de pref(L) à partir de celui de L, tous les états de l'automate de L deviennent des états finaux dans l'automate de pref(L) ( ).

Soit L un langage régulier reconnu par l'automate A. Soit w un mot accepté par A et u un de ses préfixes. Après avoir lu w, l'automate se trouve dans un état final de A. Mais, si on lit seulement une partie de w (le mot u), l'automate se trouve dans un état non final de A. Ainsi, si u doit être reconnu, alors tous les états de A doivent être des états finaux ( ).

4- La justification peut se faire en donnant un contre-exemple (un automate avec des états improductifs dont la transformation ne fonctionne pas).
La bonne justification est cependant la suivante : soit un automate A reconnaissant un langage L et comportant un état improductif *q*. Si l'automate de pref(L) est obtenu en transformant les états de A en des états finaux, alors l'état *q* devient final. Soit u un mot permettant d'atteindre l'état *q* (c'est donc un préfixe d'un mot de L)→on peut trouver un mot v tel que *uv* est reconnu par A (*u* est le préfixe de *uv*). Donc, il existe un chemin entre *q* et un des états finaux de A→ *q* n'est pas improductif. Contradiction. CQFD ( ).

5- D'après la méthode proposée, il est évident que L=pref(L) si tous les états de L sont finaux (après transformation de l'automate de L, on obtient la même chose). Evidemment, il faut que l'automate de L ne comporte pas d'états improductifs. ( )

it's over this time
hear mesagess
Rest you from my mind,
see your smile
coz its over...!
Coming Back and close the door