Université Mouloud MAMMERI de Tizi-Ouzou
Faculté de génie électrique et informatique
Département d'informatique

Année universitaire : 2013/2014
2ième année licence – Informatique
module : Théorie des langages

Epreuve de Moyenne Durée

le 18/6/2014 – Durée 1h 30mn – documents non autorisés

EXERCICE **1** : (***5 pts***)

Une girouette est un instrument indiquant le sens du vent. On considère qu'il y a quatre directions possibles : est, sud, ouest et nord. On suppose aussi que l'aiguille de la girouette, indiquant le sens, tourne d'un quart de cercle à la fois ; soit dans le sens des aiguilles d'une montre (*sens **a***), soit dans le sens opposé (*sens **b***). On supposera que la direction initiale indiquée par la girouette est l'est.

Soit L = ensemble des mouvements de l'aiguille qui se terminent à la position de départ.

1) Les mots suivants sont ils dans L ? il s'agit de : *aababb, ababb, abaaaa, bbabb*. ***(2 pts)***
2) Caractériser le langage L. ***(1,5 pts)***
3) Trouver une grammaire qui génère L. ***(1,5 pts)***

EXERCICE **2** : (***4 pts***)

Trouver des grammaires qui engendrent les langages suivants :

1) $L_1 = \{ a^n (ab)^n \ / \ n \geq 0 \}$; ***(2 pts)***
2) $L_2 = \{ w \in \{0, 1\}^* \ / $ la chaîne w représente un nombre entier divisible par 4 } ; ***(2 pts)***

EXERCICE **3** : (***6 pts***)

Soit L_1 le langage des mots de $\{a, b\}^*$ tel que dans chaque mot w de L_1, toutes les sous-chaînes de «a» consécutifs sont de longueurs ≤ 2 ; et le langage $L_2 = \{aaa, aba\}$.

1) Construire un automate d'états finis simple qui accepte L_1. ***(1,5 pts)***
2) Construire un automate d'états finis simple qui accepte L_2. ***(1,5 pts)***
3) Construire un automate d'états finis simple qui accepte $L_1 \cup L_2$. ***(1,5 pts)***
4) Rendre l'automate de 3) déterministe, s'il ne l'est pas. ***(1,5 pts)***

EXERCICE **4** : (***5 pts***)

1) En utilisant les dérivées, vérifier si les langages suivants sont réguliers :
   1-a) $L_1 = \{ a^n.b^{2m} / n \geq 1, m \geq 0 \}$; ***(1,5 pts)***
   1-b) $L_2 = \{ w.w^R / w \in \{a, b\}^* \}$. ***(1,5 pts)***
2) En utilisant les dérivées, construire un automate d'états finis correspondant à l'expression régulière : $(1.1^*.0.0^*.1)^*.0.1^*$ ***(2 pts)***

Bon courage !

Bref corrigé : (EMD de ThL – L2 informatique – 2013/2014)

EX.**1** :

1) Les mots suivants sont dans L : *aababb*, *abaaaa*
les mots qui ne sont pas dans L : *ababb*, *bbabb*

2) L peut être caractérisé comme suit : $L = \{ w \in \{a, b\}^* \ / \ ||w|_a - |w|_b| \equiv 0\ [4] \}$, ou :
$L = \{ w \in \{a, b\}^* \ / \ \exists k$ tel que $|w|_a = |w|_b + 4k$ ou $|w|_b = |w|_a + 4k \}$

3) Une grammaire pour L : G = ({a, b}, {S, A, B}, P, S)
P : $S \to ABS \mid AAAAS \mid BBBBS \mid \varepsilon$
$AB \to BA$
$BA \to AB$
$A \to a$
$B \to b$

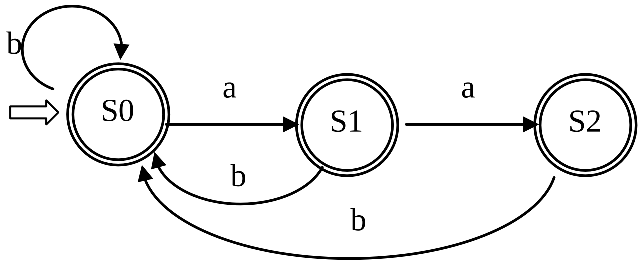
EX.**2** :

1) Une grammaire pour L_1 : $G_1 = (\{a, b\}, \{S\}, P_1, S)$
P_1 : $S \to aSab \mid \varepsilon$

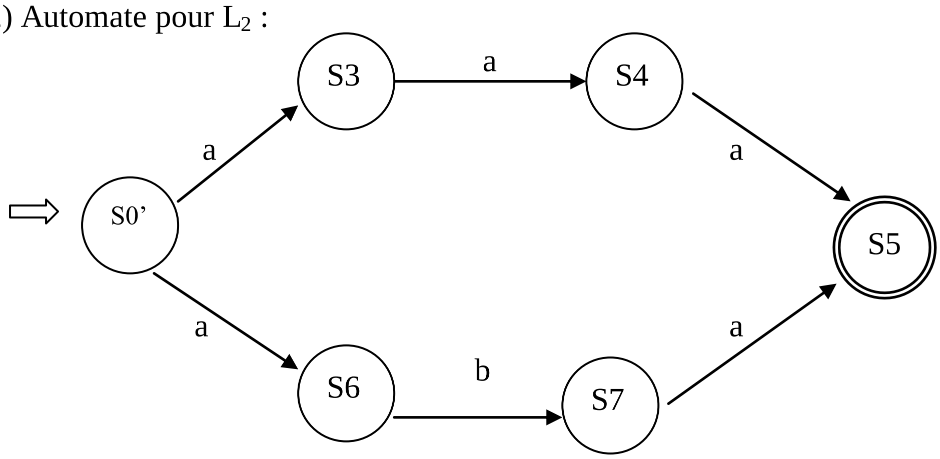
2) Une grammaire pour L_2 : $G_2 = (\{0, 1\}, \{S, A\}, P_2, S)$
P_2 : $S \to 0 \mid A$
$A \to 0A \mid 1A \mid 00$

EX. **3** :

1) Automate pour L_1 :

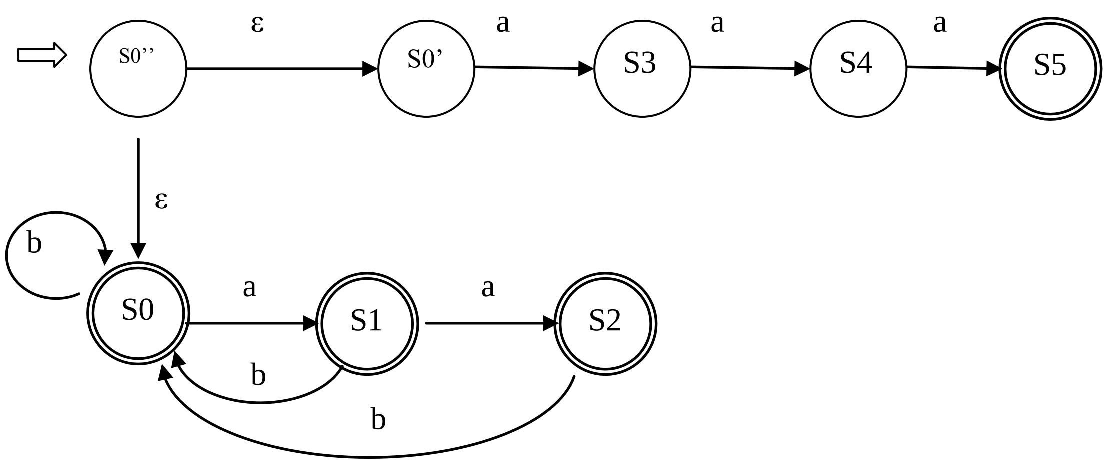


2) Automate pour L_2 :

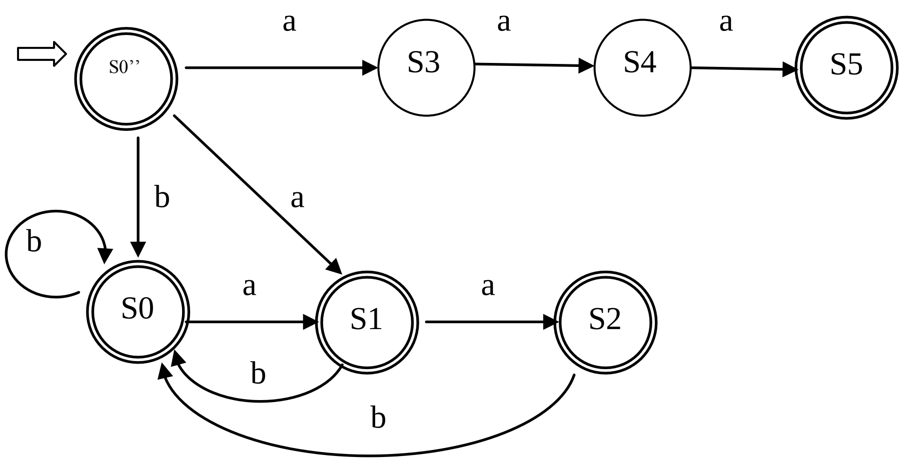


3) Puisque aba $\in$ L_1, alors $L_1 \cup L_2 = L_1 \cup \{aaa\}$

Automate semi généralisé :



Après élimination des ε-règles, on obtient :



4) L’automate de 3) n’est pas déterministe : à partir de S0’’ et avec la même lettre «a», on peut aller dans deux états différents (S1 et S3).

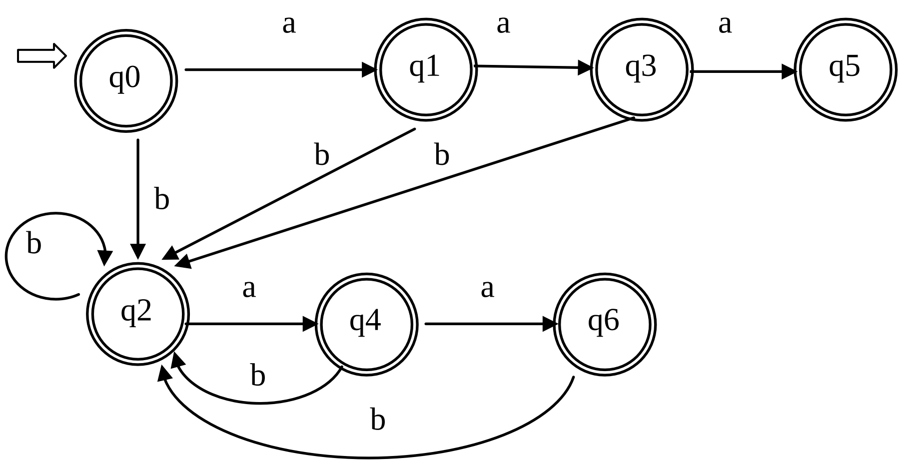
Déterminisation de l’automate de 3) :

Construction de la table de transition de l’automate déterministe :

	a	b
<S0’’> = q0	<S1,S3> (q1)	<S0> (q2)
<S1,S3> = q1	<S2,S4> (q3)	<S0> (q2)
<S0> = q2	<S1> (q4)	<S0> (q2)
<S2,S4> = q3	<S5> (q5)	<S0> (q2)
<S1> = q4	<S2> (q6)	<S0> (q2)
<S5> = q5	/	/
<S2> = q6	/	<S0> (q2)

les états soulignés sont des états finaux (ils le sont tous !).

Automate déterministe :



EX. **4** :

1)

1-a) Soit S0 = $L_1 = \{ a^n.b^{2m} / n \geq 1, m \geq 0 \}$. Ce langage est régulier, car ses dérivées par rapport aux mots de {a, b}* sont finies :

S0 || a = $\{ a^n.b^{2m} / n \geq 0, m \geq 0 \}$ = S1

S0 || b = ø

S1 || a = S1

S1 || b = $\{ b^{2m-1} / m \geq 1 \}$ = S2

S2 || a = ø

S2 || b = $\{ b^{2m} / m \geq 0 \}$ = S3

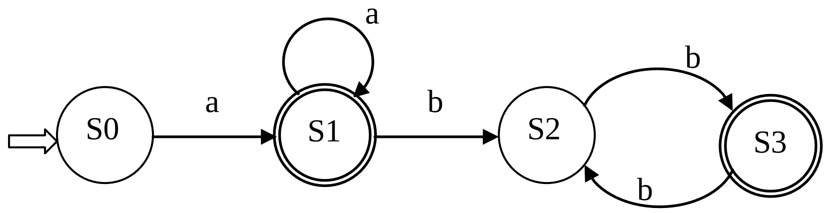
S3 || a = ø

S3 || b = S2

Après S3, on n'obtient plus de nouveaux états.

Remarque : l'automate d'états finis qui accepte L_1 est le suivant :

(S1 et S3 sont les seuls états finaux car ils contiennent ε)



1-b) Le langage $L_2 = \{ w.w^R / w \in \{a, b\}^* \}$ n'est pas régulier.

Démonstration par l'absurde : supposons que L_2 est régulier, il existe alors deux mots w_1 et w_2 tels que : $w_1 \neq w_2$ et $L_2 \| w_1 = L_2 \| w_2$.

On a :

$L_2 \| w_1 = \{ u.u^R.w_1^R / u \in \{a, b\}^* \}$ et $L_2 \| w_2 = \{ u.u^R.w_2^R / u \in \{a, b\}^* \}$

et donc : $w_2.(L_2 \| w_1) = w_2.(L_2 \| w_2) = \{ w_2.u.u^R.w_1^R / u \in \{a, b\}^* \} = L_2$

d'où : $(w_2.u)^R = u^R.w_1^R$, or $(w_2.u)^R = u^R.w_2^R$; il s'en suit que : $w_1^R = w_2^R$

et ainsi $w_1 = w_2$: contradiction.

2) Soit S0 = (1.1*.0.0*.1)*.0.1* ; calculons les dérivées de S0, pour cela posons α = (1.1*.0.0*.1)*

S0 || 0 = (α || 0).0.1* ∪ (0.1*) || 0 = ((1.1*.0.0*.1) || 0).α.0.1* ∪ 1* = ø ∪ 1* = 1* = S1 (S1 est final)

S0 || 1 = (α || 1).0.1* ∪ (0.1*) || 1 = ((1.1*.0.0*.1) || 1).α.0.1* = (1*.0.0*.1).α.0.1* = S2 (S2 non final)

S1 || 0 = ø

S1 || 1 = 1* = S1

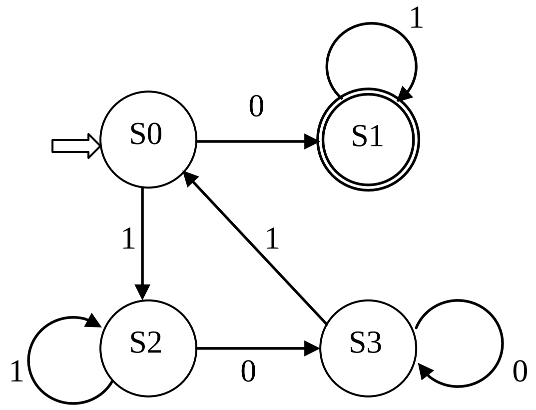
S2 || 0 = ((1*.0.0*.1) || 0).α.0.1* = 0*.1.α.0.1* = S3 (S3 non final)

S2 || 1 = ((1*.0.0*.1) || 1).α.0.1* = 1*.0.0*.1.α.0.1* = S2

S3 || 0 = (0*.1.α.0.1*) || 0 = S3

S3 || 1 = (0*.1.α.0.1*) || 1 = α.0.1* = S0

D'où l'automate :



---------------------- Fin du corrigé de l'EMD de ThL – L2 informatique – 2013/2014 ----------------------