-Département d'Informatique Licence académique 2ième année -

# -THEORIE DES LANGAGES -

Série 01 : Automates d'états finis

~ Selmani ~

AFD : automate d'états finis déterministe
AFND : automate d'états finis non déterministe

* EXERCICE 01 :

Soit l'automate A = {{$q_0$, $q_1$, $q_2$}, {∈, 0,1}, ∂, $q_0$, {$q_1$}} avec ∂ donné par la table de transition :

| ∂ | 0 | 1 | ∈ |
|---|---|---|---|
| $q_0$ | {$q_0$, $q_1$} | {$q_1$, $q_2$} | {$q_2$} |
| $q_1$ | Ø | {$q_1$} | Ø |
| $q_2$ | {$q_2$} | {$q_1$} | Ø |

- Cet automate est il déterministe ?
- Les mots 001, 0*11*, et 0*110 sont-ils acceptés par A ?
- Quel est le langage accepté par A ?

* EXERCICE 02 :

Construire *directement* des AFDs pour les langages suivants :

$L_1$ = {x ∈Σ* / x se termine par la chaine 01}
$L_2$ = {x ∈Σ* / x contient la sous chaine 101}

* EXERCICE 03 :

Pour chacune des expressions régulières suivantes construire un automate qui reconnait le langage correspondant.

$r_1 = 0^*1^*$     $r_2 = 0^* + 11^*$     $r_3 = 0^*(01+10)^*$     $r_4 = (01^*)^*$

* EXERCICE 04 :

Pour chacun des automates des figures 1 à 4 décrire le langage reconnu en donnant une expression régulière qui le caractérise.

* EXERCICE 05 :

Convertir les AFNDs des langages décrits par $r_3$ et $r_4$ de l'exercice 03 en des AFDs équivalents.

* EXERCICE 06 :

Construire des automates définissant les langages suivants. Donnez des expressions régulières qui représentent ces langages. Dire dans chaque cas si l'automate est déterministe ou pas.

1. L'ensemble des mots qui se terminent par 11.
2. L'ensemble des mots de longueur impaire.
3. L'ensemble des mots de longueur paire.
4. L'ensemble des mots qui commencent et se terminent par 0.
5. L'ensemble des mots où le nombre d'occurrences des 1 est multiple de 3.
6. L'ensemble des mots qui contiennent la chaine 010 au moins une fois.

* EXERCICE 07 :

Construire un AFD M tel que L(M) ={x ∈Σ* / x ne contient pas la sous chaine 101}

* EXERCICE 08 :

On considère les langages suivants :

$L_1$ = {x ∈Σ* / x contient la sous chaine 00}
$L_2$ = {x ∈Σ* / x contient la sous chaine 11}

1). Construire l'automate qui accepte le langage complémentaire de $L_1$ défini par $\Sigma^* - L_1$.
2) Construire des ANFDs pour les langages $L_1 \cup L_2$, $L_1L_2$, $L_1^*$
3) Construire un AFD qui reconnait le langage $L_1 \cap L_2$.

* EXERCICE 09 :

Pour chacun des automates des figures 5 et 6 donnez l'automate minimal équivalent.
Montrez que l'AFD du langage $L_2$ de l'exercice 02 est un automate minimal.

# THEORIE DES LANGAGE

## SERIE 01

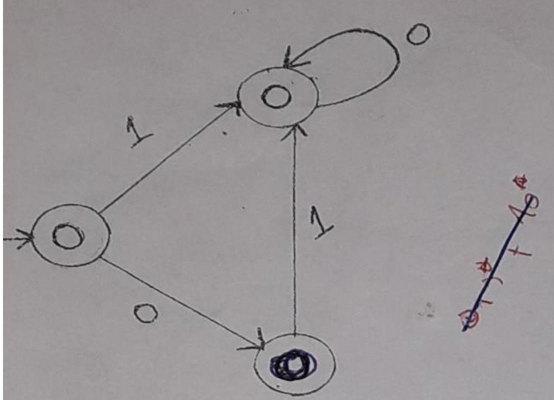



fig 1

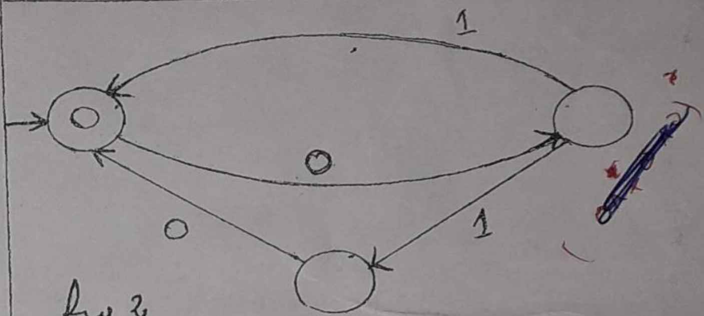



fig 2

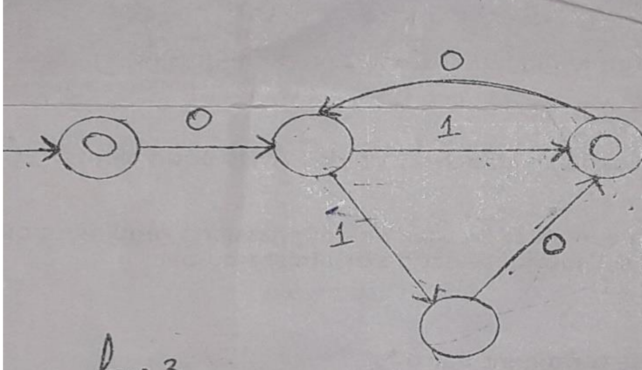



fig 3

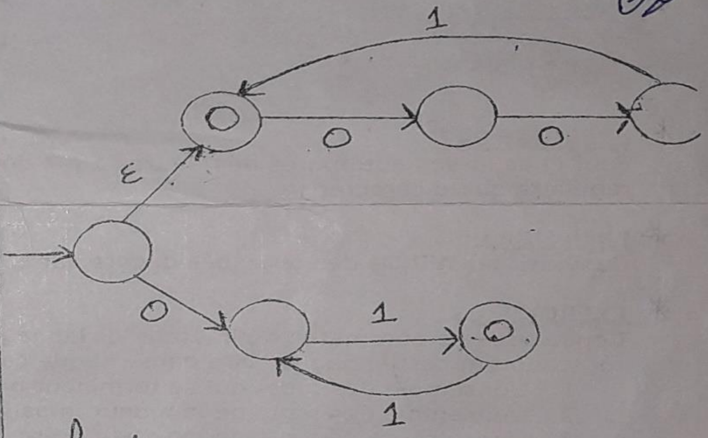



fig 4

fig 5

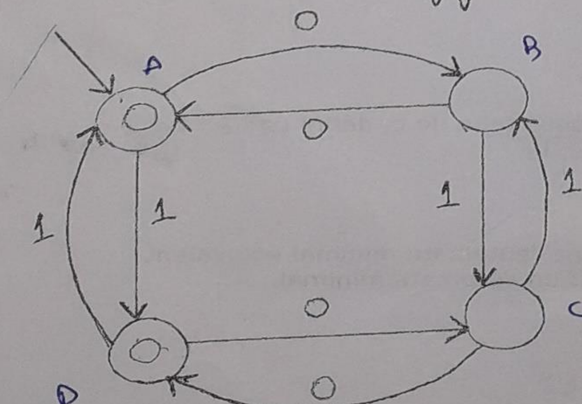
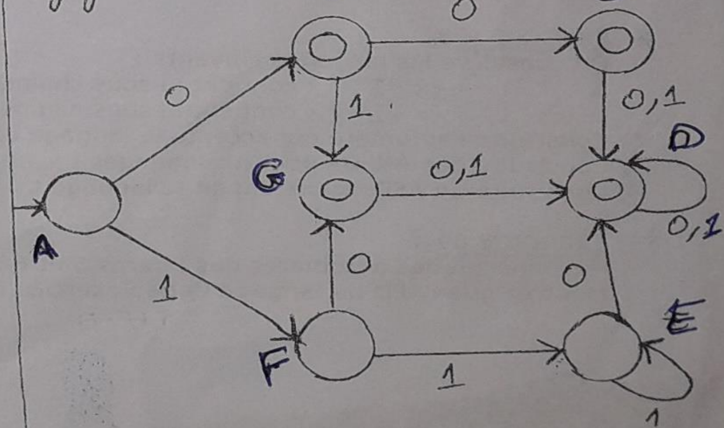



fig 6



Département d'Informatique Faculté des Sciences—UFA Sétif-1

# -THEORIE DES LANGAGES-

## Série 02

*Grammaires Hors Contexte et Automates à Pile*

**EXERCICE 01**

Soit la grammaire G = ({E}, {+, *, (, ), id}, P, E) donnée par les productions :

E → E + E / E * E / (E) / id où id représente les identificateurs.

1. Donnez les arbres de dérivations pour les mots (id+id)*id et id*(id+id)+id
2. Cette grammaire est-elle ambiguë ?

**EXERCICE 02**

Soit la grammaire G donnée par G = ({S}, { (, ) }, P, S)

S → SS / (S) /ε

1. Montrez en dessinant les arbres de dérivation que les mots (()()) et (()(()())) appartiennent au langage généré par G.
2. Pouvez-vous décrire le langage généré par G ?

**EXERCICE 03**

Soit la grammaire G = ({S}, {0, 1}, P, S) où S → S1S/0

1. Montrez que cette grammaire est ambiguë (elle admet au moins deux dérivations distinctes pour le même mot).
2. Quel est le langage généré par G ?
3. Proposez une grammaire non ambigüe qui génère le même langage.

**EXERCICE 04**

Simplifiez les grammaires suivantes :

$$\begin{cases} S \to bBAa/bB/aB \\ A \to aAbB \\ B \to SBaS/ aC \\ C \to SBA / ab \end{cases} \qquad \begin{cases} S \to aBa \\ B \to Sb / bCC / DaB \\ C \to ab / DD \\ D \to aDB \\ E \to aC \end{cases}$$

**EXERCICE 05**

Construire des grammaires régulières qui génèrent les langages :

$L_1 = (01)^*$ $L_2 = (01)^*11$ $L_3 = 0^* + (0+11)^*$ $L_4 = 0^*1^*$

**EXERCICE 06**

Construire des grammaires hors contexte qui génèrent les langages :

$L_0 = \{0^n110^n / n \geq 0\}$

$L_1 = \{wcw^t / w \in (0+1)^*\}$

$L_2 = \{0^n1^{2n} / n \geq 0\}$

$L_3 = \{0^n1^m / n \leq m \leq 2n\}$

$L_4 = L_2L_3$ $L_5 = L2 \cup L3$

$L_6 = \{0^n1^n0^m / n \geq 0 ; m \geq 0\}$

Selma

**EXERCICE 07**

Soit la grammaire G donnée par les productions : $G = (\{S,A,B\},\{a,b\},P,S)$

$S \rightarrow bA \,/\, aB$
$A \rightarrow bAA \,/\, aS \,/\, a$
$B \rightarrow aBB \,/\, bS \,/\, b$

1. Montrez que le mot aabbab appartient à L(G).
2. Pouvez-vous décrire le langage généré par G.
3. Mettre la grammaire G sous la forme normale de Chomsky

**EXERCICE 08**

Soit la grammaire donnée par les productions : $A \rightarrow Aab/Aba/a/b$

1. Donnez une grammaire équivalente non récursive à gauche.
2. Quel est le langage généré par ces productions ?

**EXERCICE 09**

Soit la grammaire $G = (\{S,A\},\{a,b\},P,S)$

$S \rightarrow AA \,/\, a$
$A \rightarrow SS \,/\, b$

1. Donnez une grammaire équivalente sous la forme normale de Greibach.

**EXERCICE 10**

Soit la grammaire $G = (\{S, B, C\}, \{a, b, c\}, P, S)$

$S \rightarrow aSBC \,/\, abC$
$CB \rightarrow BC$
$bB \rightarrow bb$
$bC \rightarrow bc$
$cC \rightarrow cc$

1. De quel type est cette grammaire ?
2. Montrez que les mots $a^2b^2c^2$ et $a^3b^3c^3$ sont générés par G.
3. Quel est le langage généré par cette grammaire ?

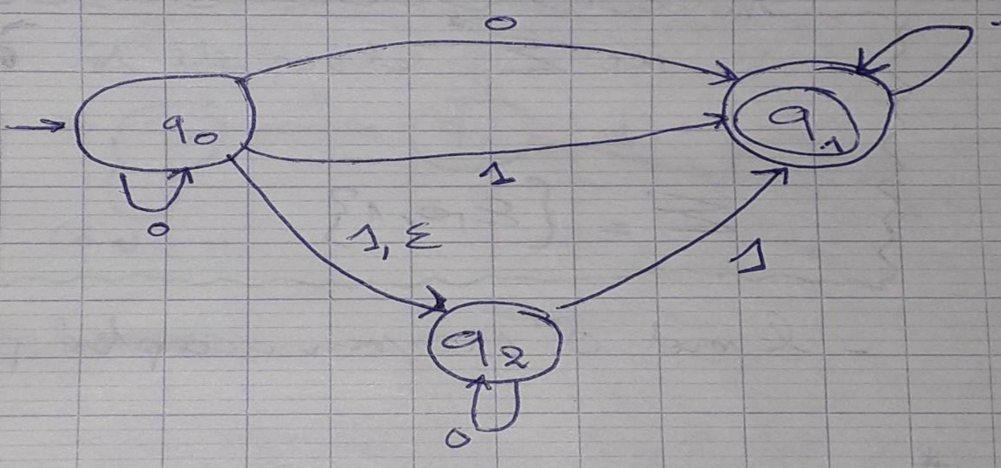
**EXERCICE 11**

1. Construire des automates à pile déterministes qui acceptent les langages $L_1$, $L_2$ et $L_6$ de l'exercice 06.
2. Construire un automate à pile qui accepte le langage $\{w \in (a+b)^* \,/\, |w|_a = |w|_b\}$. Donnez la suite des configurations de l'automate pour le mot aabbabab.

# Série n° 01

## exo1 :

$A = \{ \{q_0, q_1, q_2\}, \{\epsilon, 0, 1\}, \delta, q_0, \{q_1\} \}$




- Cét automate est non déterminate car il exist la transition null ($\varepsilon$)
- il exist au moins une transition à partir d'un etat $q_0$ sur le symbol 0 (2 transition)

- le mot 001 est accepté par A
le chemain : $q_0 \to q_0 \to q_1$

$0^* = \{\varepsilon, 0, 00, 000, 0000, \dots\}$

- le mot $0^* 1 1^*$ est accepté par A
le chemin : $q_0 \to q_1 \to q_1$

notion :

$x \in \Sigma^*$ est accepté si $\delta(q_0, x) = \{q_1\}$

état initial — état final

$\Sigma = \{\varepsilon, 0, 1\}$

- le mot $0^* 110$ non accepté par A

*

$L(A) = \{ x \in \Sigma^* / \delta(q_0, x) \in F \}$

tel que F est l'ensemble des états finaux

$0, 1 \iff (0+1)$
$1, \varepsilon \iff (1+\varepsilon)$

* le langage accepte par A

$L(A) = 0^* (0+1) 1^* +$

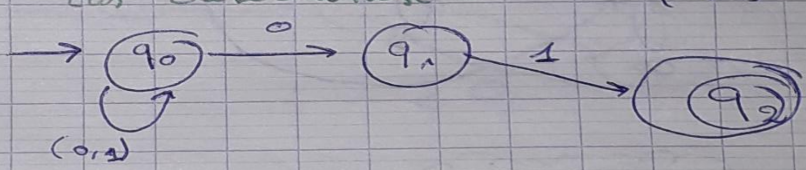
$0^* (1+\varepsilon) 0^* . 1 1^*$

exo n° 2

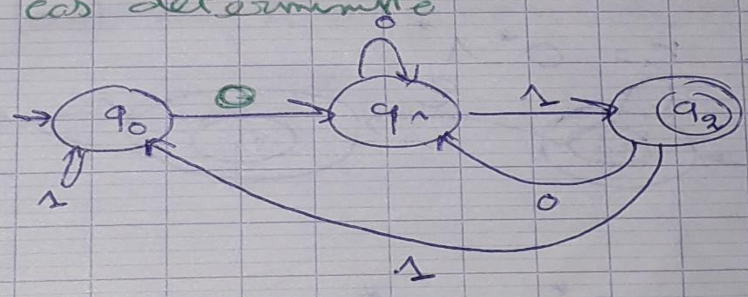
* $L_1 = \{ x \in \Sigma^* / x$ se termine par la chaine $01 \}$

~~$L_2 = \{ x \in \Sigma^* / x$ contient~~

cas non déterminisé $(0+1)^* 0 1$

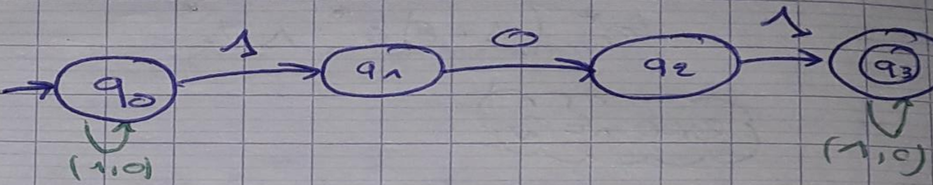


cas determinisé

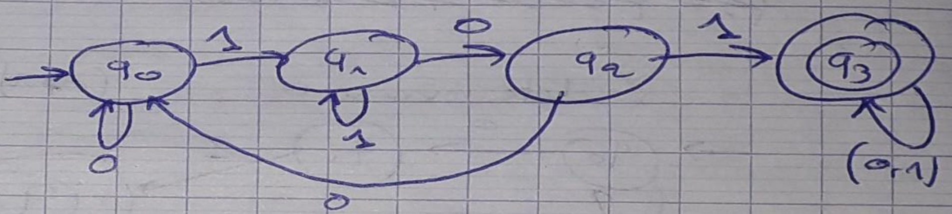


* $L_2 = \{ x \in \Sigma^* / x$ contient la sous chaine $101 \}$

cas non deterministe :

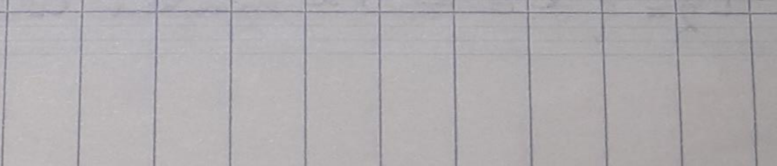



cas determinise :

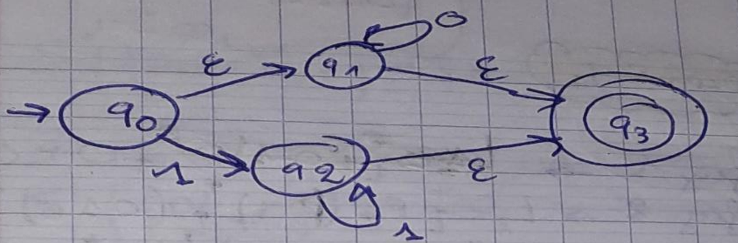



exercice n° 3

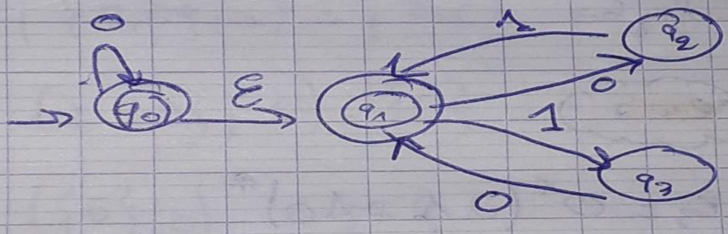
- $r_1 = 0^* 1^*$




- $r_2 = 0^* + 1 1^*$

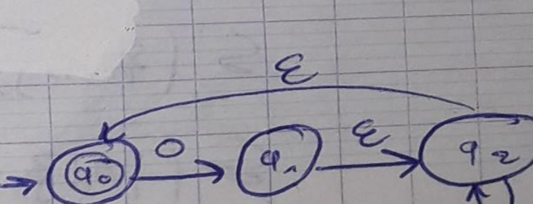



$- r_3 = 0^* (01 + 10)^*$




$(01)^* = \{ \varepsilon, 01, 0101, 010101, \ldots \}$

$- r_4 = (0\ 1^*)^*$



$$\boxed{(L_1+L_2)^* = (L_1^* \cdot L_2^*)^*}$$

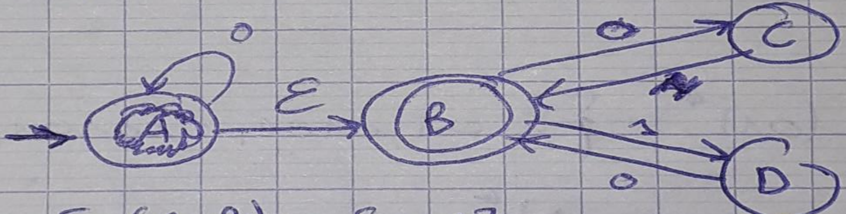
اختصار

## exo n°: 4

- fig 1 $\rightarrow$ $L_1 = \{ \varepsilon + 10^* + 0 + 010^* \}$
- fig 2 $\rightarrow$ $L_2 = \{ \varepsilon + ((01)^* \cdot (010)^*)^* \} \Rightarrow (01+010)^*$
- fig 3 $\rightarrow$ $L_3 = \{ \varepsilon + 0(1+10) ((01)^* \cdot (010)^*)^* \Rightarrow (01+010)^*$
- fig 4 $\rightarrow$ $L_4 = \{ \varepsilon + 01 \cdot (11)^* + \varepsilon (001)^* \}$
- fig 5 $\rightarrow$
- fig 6 $\rightarrow$

## exo 5

$r_3 = 0^* (01 + 10)^*$




$\delta(A, B) = \{A, B\}$

$\delta(A, 1) = \emptyset$

$\delta(\{A, B\}, 0) = \{A, B, C\}$

$\delta(\{A, B\}, 1) = \{D\}$

$\delta(\{A, B, C\}, 0) = \{A, B, C\}$

$\delta(\{A, B, C\}, 1) = \{B, D\}$

$\delta(\{D\}, 0) = \{B\}$

$\delta(\{D\}, 1) = \emptyset$

$\delta(\{B, D\}, 0) = \{B, C\}$

$\delta(\{B, D\}, 1) = \{D\}$

$\delta(\{B\}, 0) = \{C\}$

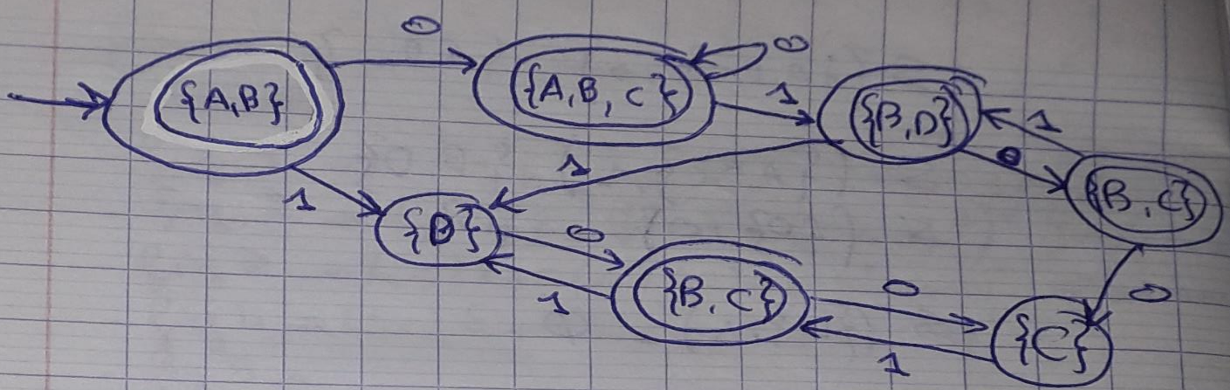
$\delta(\{B\}, 1) = \{D\}$

$\delta(\{B, C\}, 0) = \{C\}$

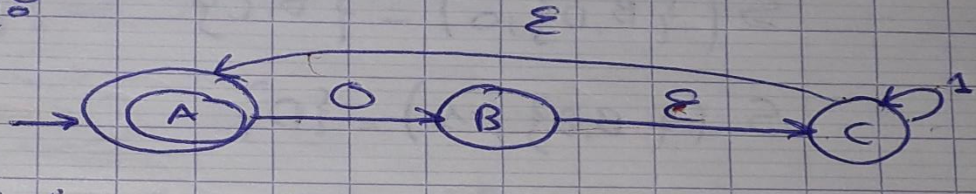
$\delta(\{B, C\}, 1) = \{B, D\}$

$\delta(\{C\}, 0) = \emptyset$

$\delta(\{C\}, 1) = \{B\}$




* $r_4$:




$\delta(\{A\},0) = \{B\}$

$\delta(\{A\},1) = \emptyset$

$\delta(\{B\},0) = \emptyset$

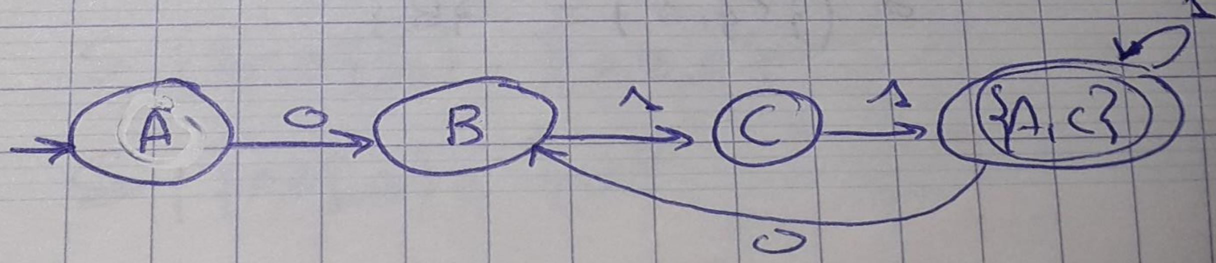
$\delta(\{B\},1) = \{C\}$

$\delta(\{C\},0) = \emptyset$

$\delta(\{C\},1) = \{A,C\}$

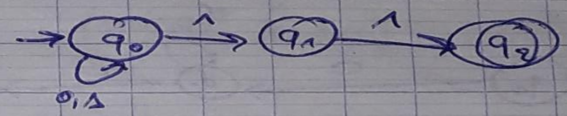
$\delta(\{A,C\},0) = \{B\}$

$\delta(\{A,C\},1) = \{A,C\}$



Exo 6 :

1).




$(0+1)^* 11$

le cas : est automate non determiste

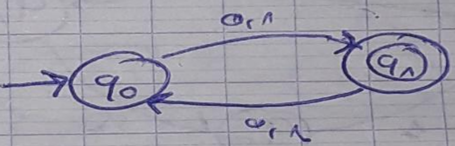
2).

$\Sigma = \{0,1\}$

$|0011| = 4$

$|011| = 3$

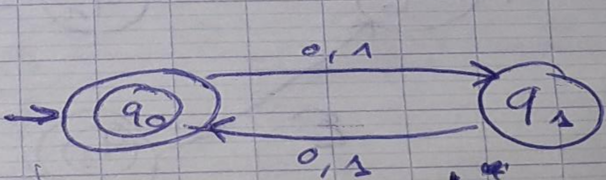
$L_2 = \{ x \in \Sigma^*, |x| = 2k+1 , k \geq 0 \}$




$= ((0+1).(0+1).(0+1))^*$

le cas : est automate deterministe

3.




$L_3 = \{ ((0+1).(0+1))^* + \varepsilon \}$

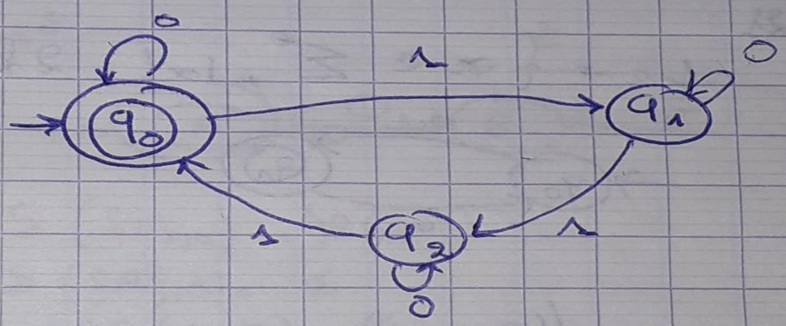
automate est deterministe

4-


$L_2 = 0(0+1)^* 0$

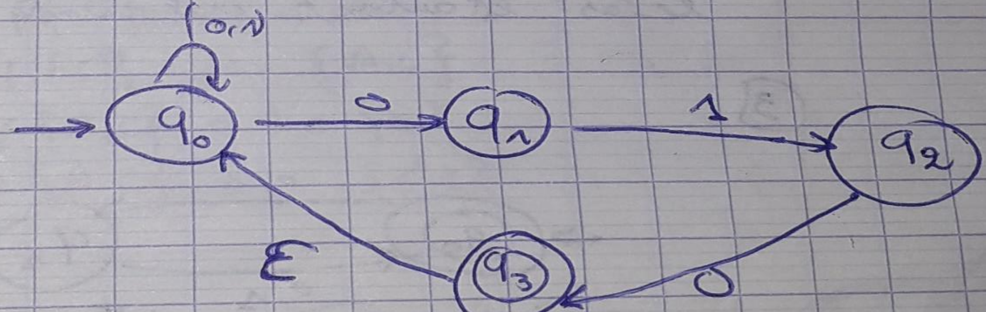
AFND

5




$L_5 = (0^* 1 0^* 1 0^* 1)^*$

6

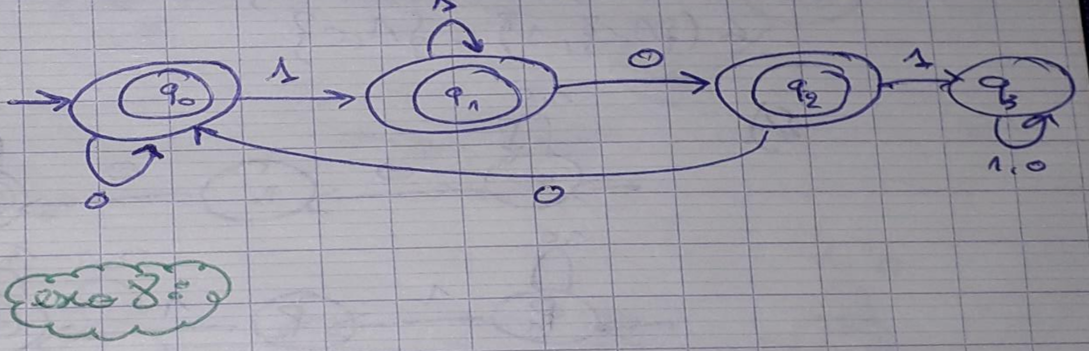


~~$L_6 = (0+1)^* \cdot (010)$~~

$L_6 = (0+1)^* (010) (0+1)^* (01)^* (0+1)^*$

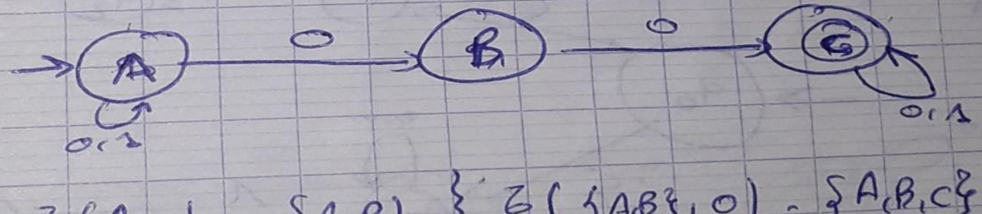
exo 7 :

$L(M) = \{ x \in \Sigma^* \mid x$ ne contient pas la sous chaîne $101 \}$



exo 8 :

1)- $L_1 = \{ x \in \Sigma^* \mid x$ contient la sous chaîne $00 \}$

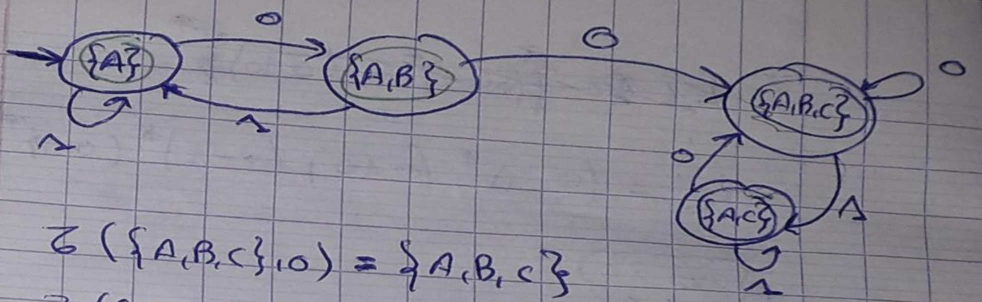


$\delta(A, 0) = \{A, B\}$

$\delta(A, 1) = \{A\}$

$\delta(\{A, B\}, 0) = \{A, B, C\}$

$\delta(\{A, B\}, 1) = \{A\}$



complémentaire ->

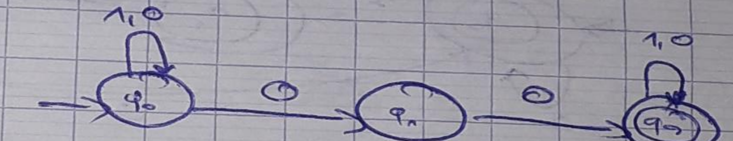
$\delta(\{A,B,C\},0) = \{A,B,C\}$

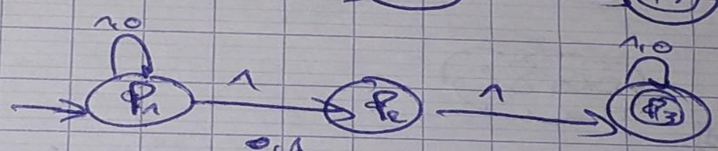
$\delta(\{A,B,C\},1) = \{A,C\}$

$\delta(\{A,C\},0) = \{A,B,C\}$

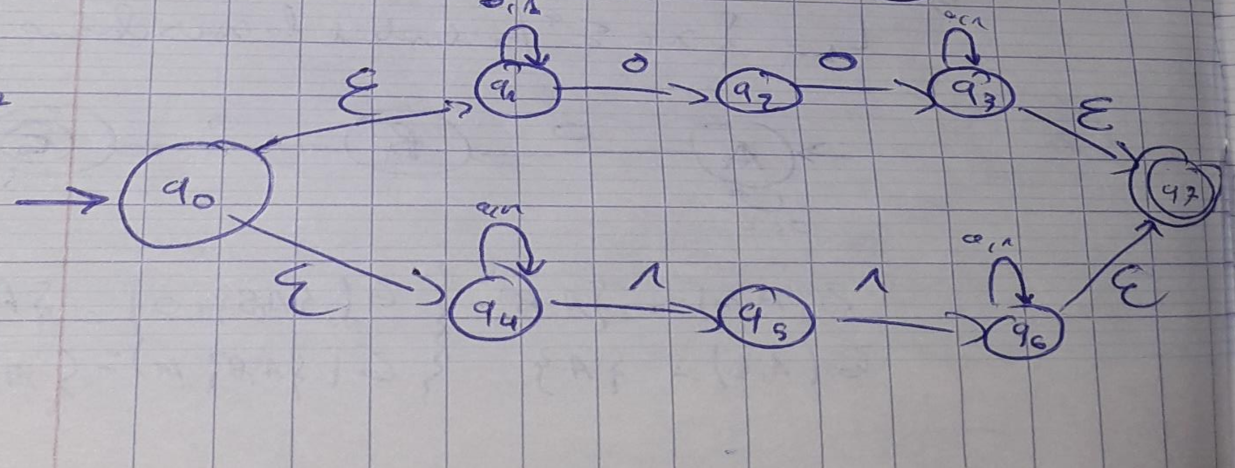
$\delta(\{A,C\},1) = \{A,C\}$

2)

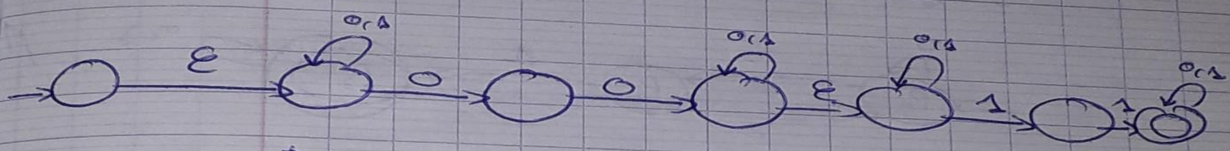




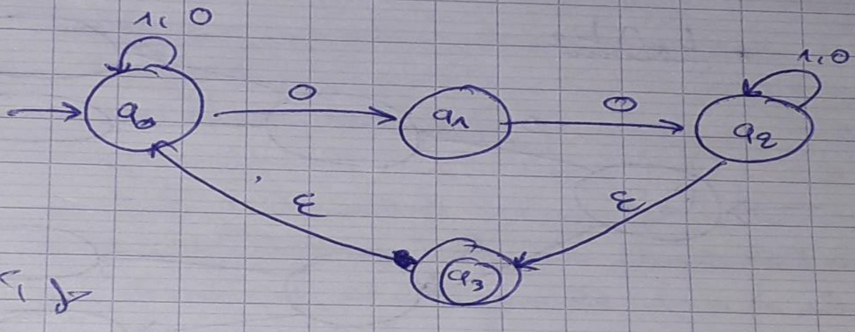
$L_1 \cup L_2$



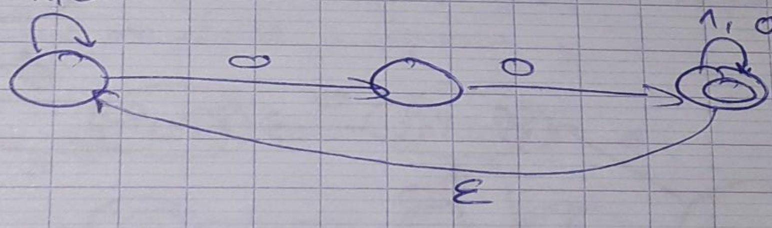
* $L_1 L_2$



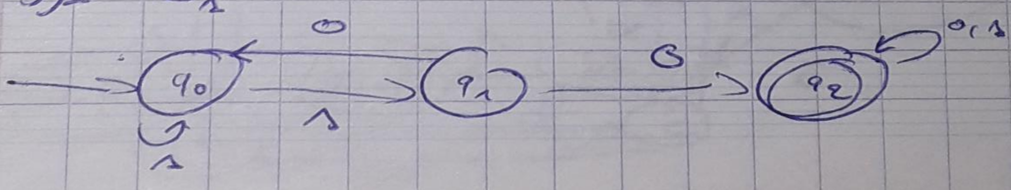

. $L^*$



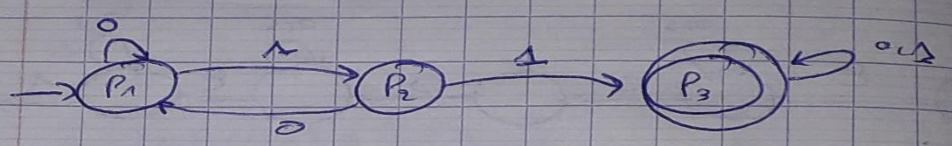

حل آخر :



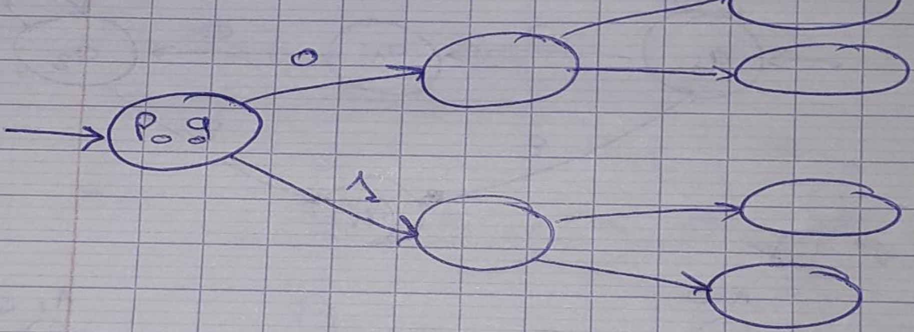

3)- $L_1$

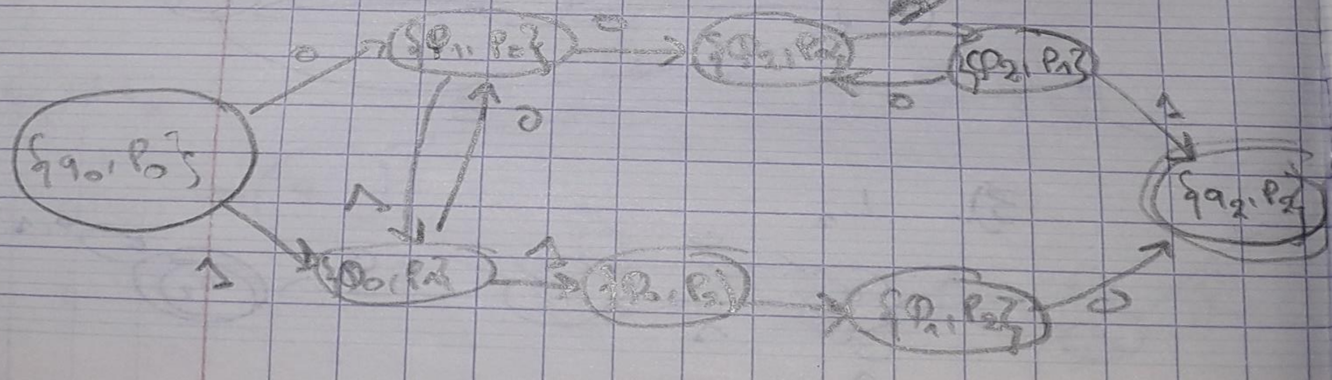


- $L_2$ (non deterministe)




* $L_1 \cap L_2$



exo 9 :

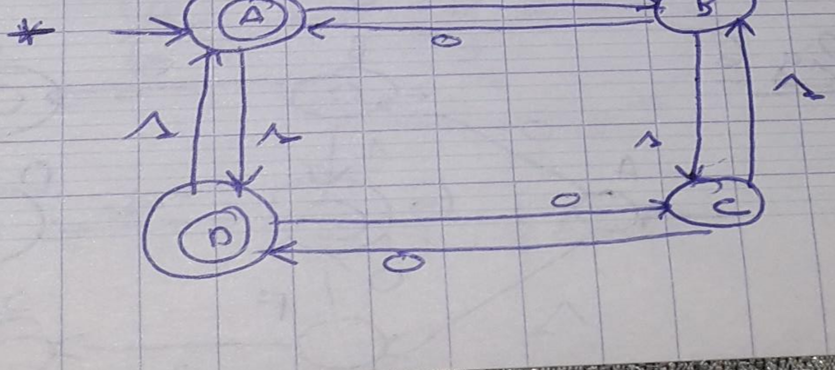
Rappel :

l'algorithe de Minimisation :

1- faire deux classes : A contenant les etats finaux et B : contenant les etats non finaux

2) S'il existe une symbol a et deux etats $q_i$ et $q_j$ d'une meme classe tel que $\delta(q_i, a)$ et $\delta(q_j, a)$ n'appartiennent pas à la meme classe alors créés une nouvelle classe et séparer $q_i$ et $q_j$. On laisse dans la meme classe tout les etats qui donnent un etat d'arrivée dans la même classe.

3). Recommencer l'étape 2 jusqu'à ce qu'il n'y plus de classe à séparer



$\{A, D\} \xleftarrow{0} \{B, C\}$

$A \equiv D$

$A \equiv D$ ✓

$\delta(A, 0) \overset{?}{=} \delta(D, 0)$

$\delta(A, 1) \overset{?}{=} \delta(D, 1)$

$B \equiv C$

$A \equiv A$

$A \equiv D$

$\{A, D\}$

$B \equiv C$

$B \equiv C$ ✓

$\delta(B, 0) = \delta(C, 0)$

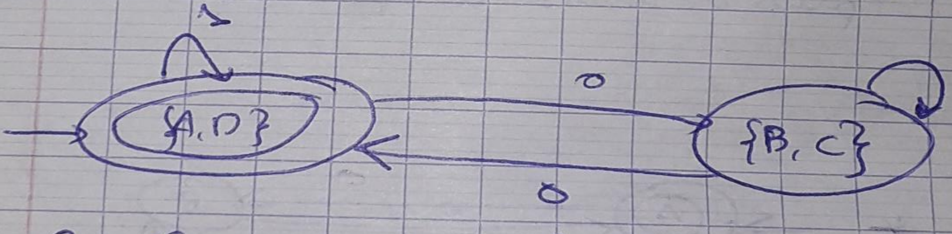
$\delta(B, 1) = \delta(C, 1) =$

$D \equiv A$

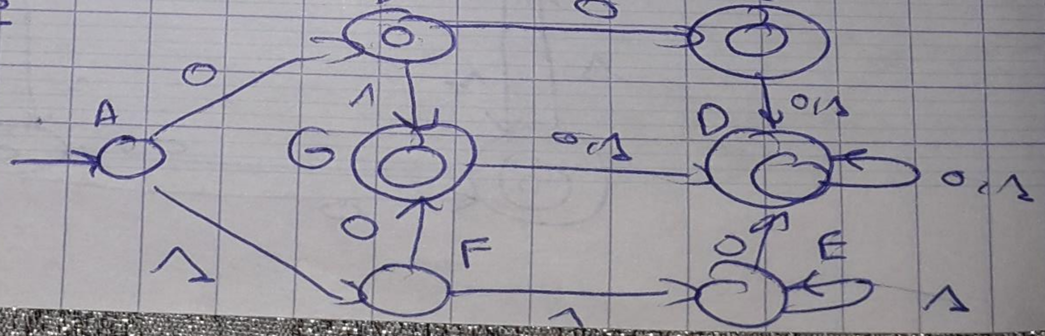
$B \equiv C$

$\boxed{B \equiv C}$ ✓

$\{B, C\}$




- fig 6

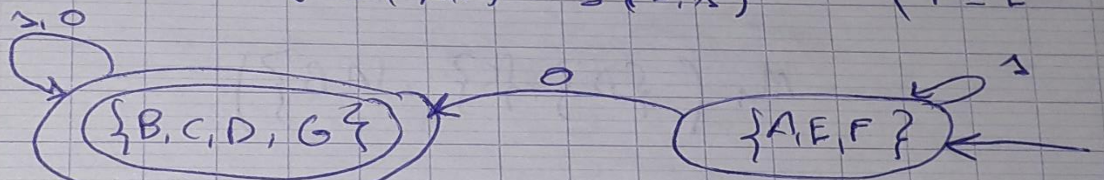


$M_0 = (\{B, C, D, G\}, \{A, E, F\})$

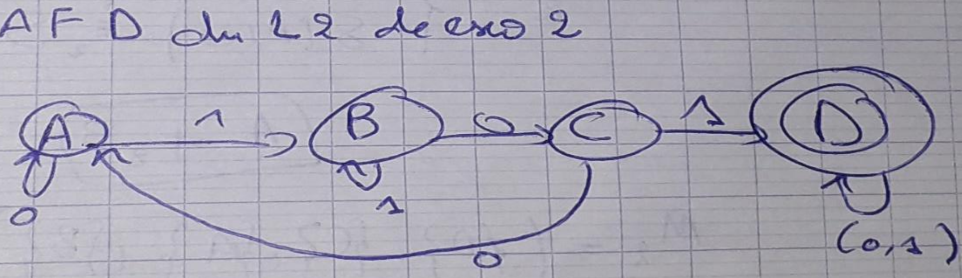
$B \equiv C$ $\quad \begin{cases} B \equiv C \\ \delta(B,0) = \delta(C,0) \\ \delta(B,1) = \delta(C,1) \end{cases}$ $\quad$ $C \equiv D$, $G \equiv D$

$D \equiv G$ $\quad \begin{cases} \delta(D,0) = \delta(G,0) \\ \delta(D,1) = \delta(G,1) \end{cases} \Rightarrow \} \checkmark$

$A \equiv E$ $\quad \begin{cases} \delta(A,0) = \delta(E,0) \\ \delta(A,1) = \delta(E,1) \end{cases} \Rightarrow \begin{cases} B \equiv D \checkmark \\ F \equiv E \end{cases}$




* AFD du L2 de exo 2




$M_0 = (\{D\} ; \{A, B, C\})$

$A \overset{?}{\equiv}_1 B \begin{cases} A \equiv_0 B \\ S(A,0) = S(B,0) \\ S(A,1) = S(B,1) \end{cases}$ $A = C$ ✓ $B = B$ ✓

$$\boxed{A \equiv B}$$

$B \overset{?}{\equiv}_1 C \begin{cases} B \equiv_0 C \\ S(B,0) = S(C,0) \\ S(B,1) = S(C,1) \end{cases}$ $C = A$ ✓ $B \neq D$ ✗

$$B \neq C$$

$M_1 \cdot (\{D\}, \{C\}, \{AB\})$

$A \equiv_2 B \begin{cases} A \equiv_1 B \\ S(A,0) \overset{?}{\equiv}_2 S(B,0) \end{cases}$ / $A = C$ ✗

$$\boxed{A \neq_2 B}$$

$M_2 = (\{D\}, \{C\}, \{A\}, \{B\})$

cet deterministe est minimal.

# Série n° 2

## exo n° 1

$G = (\{E\}, \{+, *, (, ), id\}, P, E)$

Non-terminaux

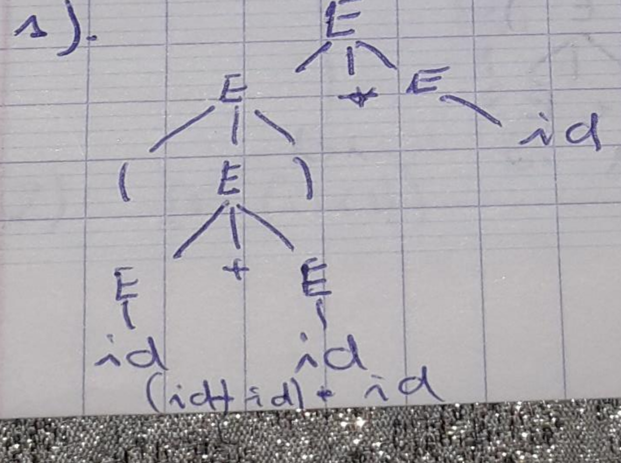
terminaux

les règle de production

les symbole de départ (l'axiome)

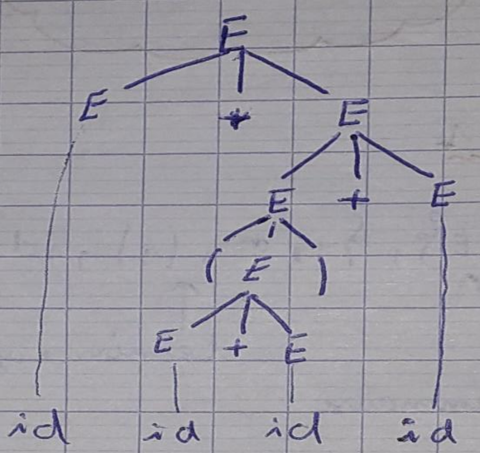
$$\begin{cases} E \rightarrow E + E \\ E \rightarrow E * E \\ E \rightarrow (E) \\ E \rightarrow id \end{cases}$$

1).



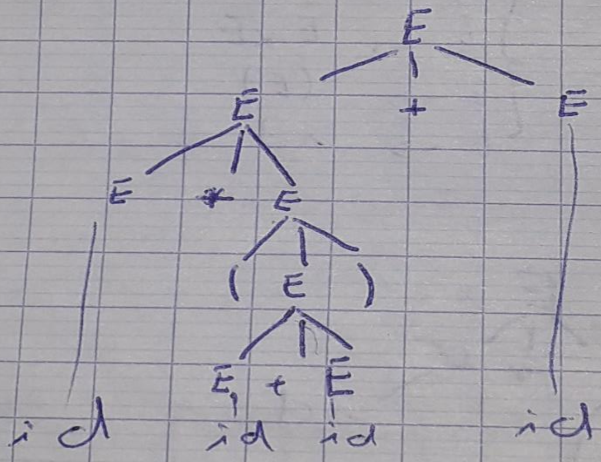

(id+id)*id

id * (id+id)+id





حل آخر



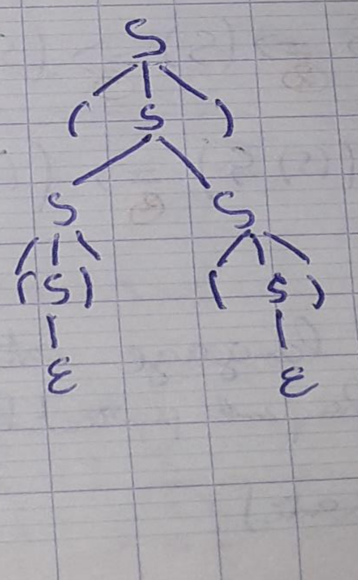
2) cette gramaire est ambiguë car elle admet aux moins deux dérivation distinctes pour le même mot

exe n° 2

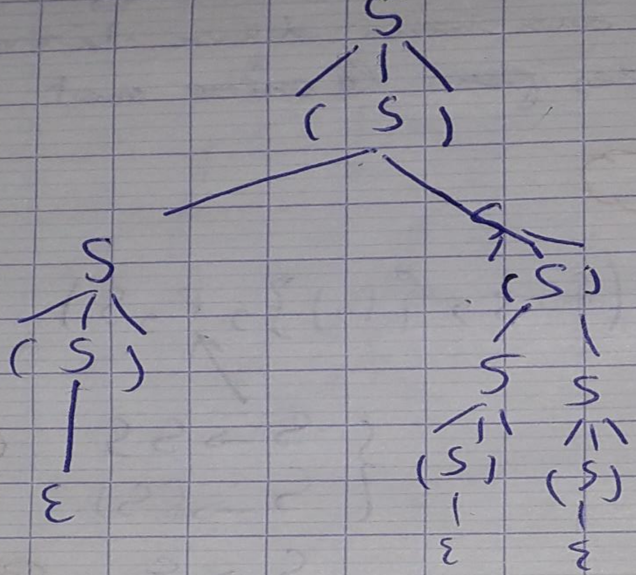
$$G = (\{S\}, \{(,)\}, P, S)$$

$$\begin{cases} S \rightarrow SS & ① \text{ règle} \\ S \rightarrow (S) & ② \text{ règle} \\ S \rightarrow \varepsilon & ③ \text{ règle} \end{cases}$$

1) $(()())$




2) $(()(()()))$



2)- $L(G) = \{ w \in T^*, \ S \overset{*}{\Rightarrow} w \}$

$S \xRightarrow[2]{} (S) \xRightarrow[1]{} (SS) \xRightarrow[2]{} ((S)S) \xRightarrow[2]{} ((S)(S)) \xRightarrow[3]{} (()(S)) \xRightarrow[3]{} (()())$

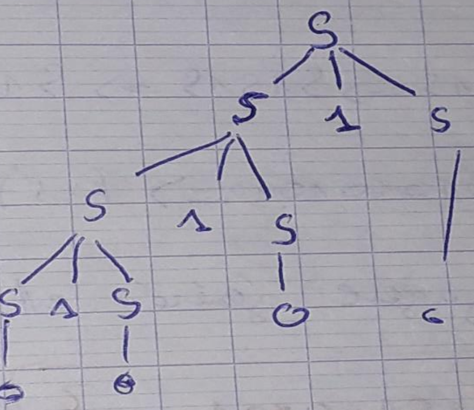
* ce langage est un langagee bien parantèse (chaque parantèse ouvrat à un parantèse fermate)

exo 3 :

1) $G = (\{S\}, \{0,1\}, P, S)$

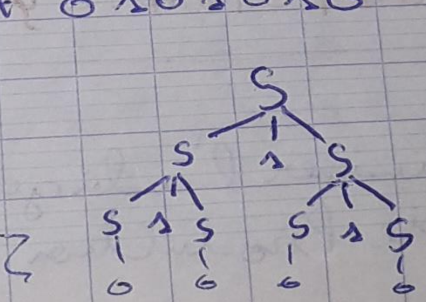
$$\begin{cases} S \to S1S \\ S \to 0 \end{cases}$$

dérivation 1



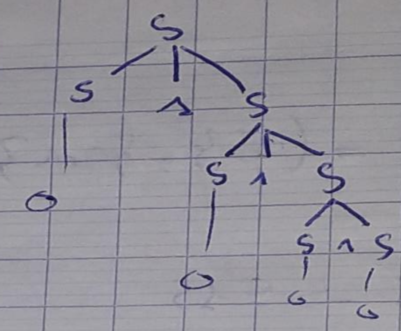
le mot 0101010

dérivation 2



le mot 0101010

dérivation 3




le mot 0 1 0 1 0 1 0

2)

$S \Rightarrow S1S \Rightarrow S1S1S \Rightarrow S1S1S1S \Rightarrow S1 \ldots$

$\Rightarrow 0101010101010 \Rightarrow (01)^n 0 \; ; \; n > 0$

la langagere généré est : $(01)^n 0$ $(n > 0)$

$(01)^* = \{ \varepsilon + 01 + 0101 + \ldots \}$

$(01)^+ = \{ 01 + 0101 + \ldots \}$

3) Pour éliminer l'ambiguité il faut changer la règle de Production

$G' : S \rightarrow 01S/0$

ou bien $G' : S \rightarrow S10/0$

exo 4 :

- élimination des symbôle non productifs
  - $C \rightarrow ab$ => C est Productif
  - $B \rightarrow aC$ => B " "
  - $S \rightarrow bB$ => S " "

A est non Productif => on l'élimine

- élimination des symbôle inaccessible.
- S est par defoult accessible.
- $S \rightarrow bB$ => B est "

$B \rightarrow aC$ => C est inaccessible

$G'$

$S \rightarrow bB \mid bB \mid aB$

$B \rightarrow SBaS \mid ac$

$C \rightarrow ab$

- elimination des symboles improductif

$C \rightarrow ab$ C est productif
$E \rightarrow aC$ E 〃 〃
$B \rightarrow bCC$ B 〃 〃
$S \rightarrow aBa$ S 〃 〃

D est improductif $\Rightarrow$ On supprime D.

- elimine des symbole innacessible

S par défaut est accessible
$S \rightarrow aBa$ B est accissible
$B \rightarrow bCC$ C est 〃

E est inaccessible $\rightarrow$ E est supprimé

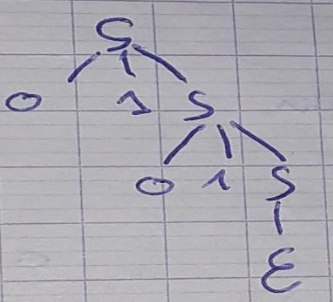
$$G' \begin{cases} S \rightarrow aBa \\ B \rightarrow bCC / Sb \\ C \rightarrow ab \end{cases}$$

Exo 5

- $L_1 = (01)^*$

$P = S \rightarrow 01S / \varepsilon$

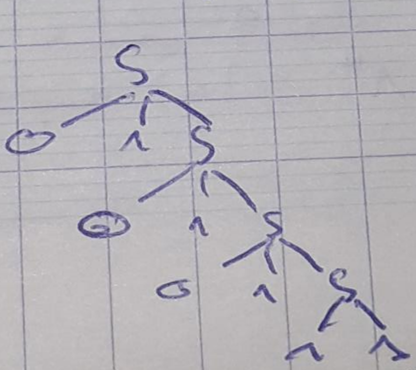
$G = (\{S\}, \{0,1\}, P, S)$




- $L_2 = (01)^* 11$

$P = S \longrightarrow 01S / 11$

$G = (\{S\}, \{0,1\}, P, S)$



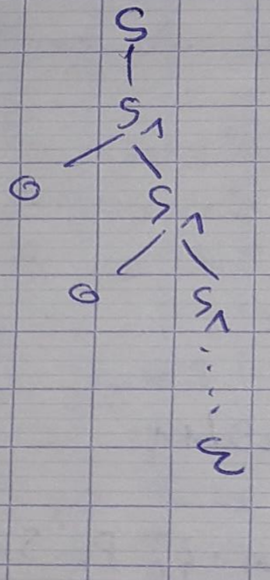
$L_3 = 0^* + (0+11)^*$

$S \rightarrow S_1 \mid S_2$

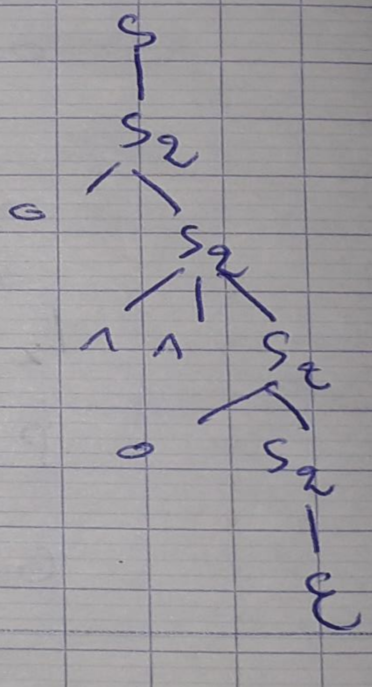
$S_1 \rightarrow 0S_1 \mid \varepsilon$

$S_2 \rightarrow 0\,S_2 \mid 11\,S_2 \mid \varepsilon$

$G = (\{S_1, S_2\}, \{0, 1\}, P, S).$



ou



0110

---

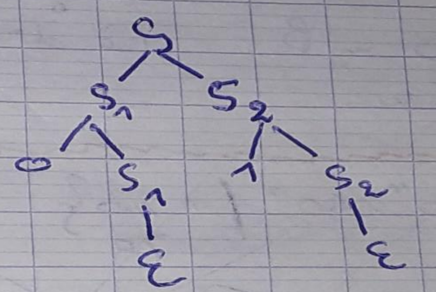
$L_4 = 0^* 1^*$

$S \rightarrow S_1 S_2$

$S_1 \rightarrow 0 S_1 \mid \varepsilon$

$S_2 \rightarrow 1 S_2 \mid \varepsilon$

$G = (\{S_1 S_2\}, \{0, 1\}, P, S)$




exo 6

classification des grammaires

1) gramaire sans restruction (type 0)

$\alpha \rightarrow \beta \left( \begin{array}{l} \alpha\beta \in (V,T)^* \\ \alpha \neq \Lambda \end{array} \right)$

2) grammaire sensible au context (type 1)

$\alpha \rightarrow \beta$ ou $\alpha, \beta \in (V,T)^*$

tel que $|\alpha| \leqslant |\beta|$

3) gramaire hors context (type 2)

$A \to \alpha$ où $A \in V$, $\alpha \in (V+T)^*$

4). grammaire regulière (type3)

$A \to B w$ où $A, B \in V$

$A \to w$ $\quad w \in T^*$

(à gauche)

$A \to w B$ (à droite)

$A \to w$

$L_0 = \{0^n 1 1 0^n / n \geq 0\}$

~~$L_0 = \{11, 0110, 001100, \dots\}$~~

~~$S \to 0 A 0$~~

~~$A \to 11$~~

$L_1 = \{w c w^t / w \in (0+1)^*\}$

$L_1 = \{100 c 001 ; 1101 c 1011 \dots\}$

$S \to 0S0 / 1S1 / c$

$S \to 0S0 \to 01S10 \to 011S110$

$\to 011c110$

$L'_0 = \{0^n 11 0^n , n \geqslant 0\}$

$S \rightarrow A11A$

$A \rightarrow 0A / \varepsilon$

* $L_2 = \{0^n 1^{2n} / n \geqslant 0\}$ $\boxed{P: S \rightarrow 0S11/\varepsilon}$

$L_2 = \{011/001111/000111111 \dots\}$

* $L_3 = \{0^n 1^m / n \leqslant m \leqslant 2n\}$

$S \rightarrow 0S11/0S1/\varepsilon$

* $L_4 = L_2 L_3$

$$\begin{cases} S \rightarrow S_1 S_2 \\ S_1 \rightarrow 0S_1 11/\varepsilon \\ S_2 \rightarrow 0S_2 11/0S_2 1/\varepsilon \end{cases}$$

* $L_5 = L_2 \cup L_3$

$$\begin{cases} S \rightarrow S_1/S_2 \\ S_1 \rightarrow 0S_1 11/\varepsilon \\ S_2 \rightarrow 0S_2 11/0S_2 1/\varepsilon \end{cases}$$

* $L_6 = \{0^n 1^n 0^m / n \geqslant 0 ; m \geqslant 0\}$

$$\begin{cases} S \to S_1 S_2 \\ S_1 \to 0 S_1 1 / \varepsilon \\ S_2 \to 0 S_2 / \varepsilon \end{cases}$$

exo 7 :

$G = (\{S, A, B\}, \{a, b\}, P, S)$

$S \to bA / aB$

$A \to bAA / aS / a$

$B \to aBB / bS / b$

1)- * $S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabB \Rightarrow aabbS \Rightarrow aabbaB \Rightarrow \boxed{aabbab}$

* $S \Rightarrow bA \Rightarrow bbAA \Rightarrow bbaSA \Rightarrow bbabAA \Rightarrow bbabaa$

2)- le langage généré par G contient autant de a que de b dans un ordre quelconque.

$$\begin{cases} A \to BC \\ A \to a \end{cases} \Big/ \begin{array}{l} A, B, C \in V^* \\ a \in T^* \end{array}$$

- 3) Mettre la gramaire sous (FNC)

$$\begin{cases} S \to bA / aB \\ A \to bAA / aS / a \\ B \to aBB / bS / b \end{cases} \to \begin{cases} S \to C_b A / C_a B \\ A \to C_b AA / C_a S / C_a \\ B \to C_a BB / C_b S / C_b \end{cases}$$

$C_a \to a$

$C_b \to b$

$$\Rightarrow \begin{cases} S \to C_b A / C_a B \\ A \to C_b D_1 / C_a S / C_a \\ D_1 \to AA \\ B \to C_b D_2 / C_b S / C_b \\ D_2 \to BB \\ C_a \to a \\ C_b \to b \end{cases}$$

$$\boxed{\begin{matrix} A \to A\alpha \\ A \to \beta \end{matrix} \quad \equiv \begin{cases} A \to \beta A' / \beta \\ A' \to \alpha A' / \alpha \end{cases}}$$

exo 8 :

1) $A \to A \underbrace{ab}_{\alpha_1} / A \underbrace{ba}_{\alpha_2} / \underset{\beta_1}{a} / \underset{\beta_2}{b}$

$$\begin{cases} A \to a A' / a / b A' / b \\ A' \to ab A' / ab / ba A' / ba \end{cases}$$

2) $L = (a+b) \cdot (ab+ba)^*$

exo 9 :

F.N.G :

1) tous les Production sous la F.N. Chomsky.
2) 〃 〃 〃 de P sont de la forme
$A \to a\alpha \quad A, \alpha \in V^*$
$a \in T$

*

$$G = (\{S, A\}, \{a, b\}, P, S)$$

$S \to AA \mid a$
$A \to SS \mid b$

on va renommer les variables.

1) $\begin{cases} A_1 \to A_2 A_2 \mid a \\ A_2 \to A_1 A_1 \mid b \end{cases}$

$$A_2 \to A_2 \underbrace{A_2 A_1}_{\alpha} \mid \underbrace{b}_{\beta_1} \mid \underbrace{a A_1}_{\beta_2}$$

$A_2 \to \beta_1 A' \mid \beta_1 \mid \beta_2 A' \mid \beta_2$
$A' \to \alpha A' \mid \alpha$

$$\begin{cases} A_2 \to bA_3 \mid b \mid aA_1A_3 \mid aA_1 \\ A_3 \to A_2 A_1 A_3 \mid A_2 A_1 \\ A_1 \to A_2 A_2 / a \end{cases}$$

exo 10 -

1) - le type de la gramaire type 1

$|\alpha| \leq |\beta|$

$S \to aSBC \;(1) \;/\; abC \;(2)$

$CB \to BC \;(3)$

$bB \to bb \;(4)$

$bC \to bc \;(5)$

$cC \to cc \;(6)$

2) - $S \to aSBC \to aabCBC \to aabBCC$

$\to a^2bbCC \to a^2bbcC \to a^2b^2cc \to a^2b^2c^2$

$S \overset{1}{\Rightarrow} aSBC \overset{1}{\Rightarrow} aaSBCBC \to$

$\overset{2}{\Rightarrow} aaabCBCBC \overset{3}{\Rightarrow} \underbrace{aaabBCCBC}$

$\overset{3}{\Rightarrow} aaabBCBCC \overset{4}{\Rightarrow} aaabbCBCC$

$\overset{3}{\Rightarrow} aaabbBCCC \overset{4}{\Rightarrow} aaabbbCCC \overset{5}{\Rightarrow} aaabbbcCC$

$\overset{6}{\Rightarrow} aaabbbccC \overset{6}{\Rightarrow} a^3b^3c^3$

3). $L = a^n b^n c^n$ $\quad n > 0$

$n \geqslant 1$

exo 11

## l'automate à pile :

définition : $A = (\Phi, \Sigma, \Gamma, S, q_0, Z_0, F)$

$\Phi$ : l'ensemble des etats.
$\Sigma$ : l'alphabet d'entré.
$\Gamma$ : " de la pile.
$S$ : fonction de transition
$q_0$ : l'état initial
$Z_0$ : le symbole de la pile vide.
$F$ : l'ensemble des état finaux

Movement :

$\Phi \times \Sigma \times \Gamma \longrightarrow \Phi \times \Gamma^*$

$S(P, a, R) = (q ; \gamma)$

1).- $L_1 = \{ w c w^t \mid w \in (0+1)^* \}$

$001 / c \, 1100$

$\delta(q_0, 0, Z_0) = (q_0, 0Z_0)$
$\delta(q_0, 0, 0) = (q_0, 00)$

$\delta(q_0, 1, 0) = (q_0, 100)$
$\delta(q_0, c, 1) = (q_1, 100)$
$\delta(q_1, 1, 1) = (q_1, \varepsilon)$
$\delta(q_1, 0, 0) = \cancel{(q_1, \varepsilon)} \; (q_1, \varepsilon_0)$
$\delta(q_1, \#, Z_0) = (q_0, Z_0)$

$\Rightarrow x \in (0+1) \quad y \in (0+1)$

$\delta(q_0, x, Z_0) = (q_0, xZ_0)$ empiler $x$
$\delta(q_0, y, x) = (q_0, yx)$ " $y$
$\delta(q_0, x, y) = (q_1, yx)$ rien à faire (changement d'état)
$\delta(q_1, y, y) = (q_1, \varepsilon)$ dépiler $x$
$\delta(q_1, \#, Z_0) = (q_f, Z_0)$ on accepte

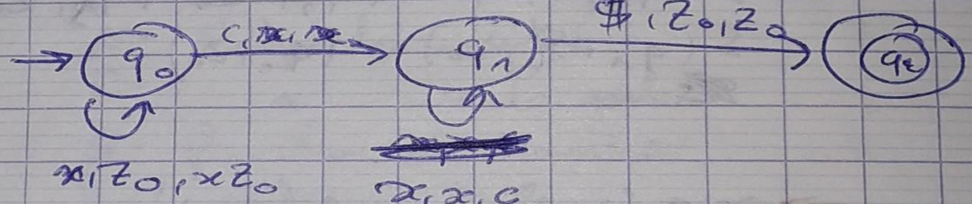
| |
|---|
| $y$ |
| $x$ |
| $Z_0$ |

00 0111 111

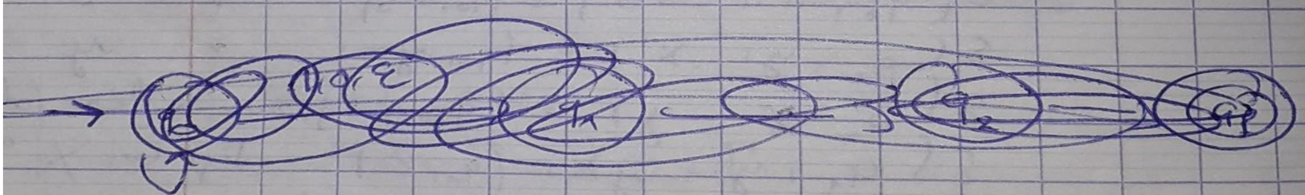
$\delta(q_0, 0, Z_0) = (q_0, 0Z_0)$ empiler le 1er 0
$\delta(q_0, 0, 0) = (q_0, 00)$ empiler le reste des 0
$\delta(q_0, 1, 0) = (q_1, \varepsilon)$ depiler le 0
$\delta(q_1, 1, 0) = (q_1, 0)$ la pile reste inchanger.
(changement d'etat)

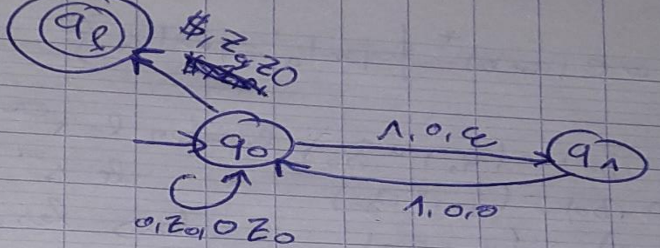
$\delta(q_1, \$, Z_0) = (q_2, Z_0)$ on accepte

$a \in (0+1)$




$x, y, yx\, Z_0$





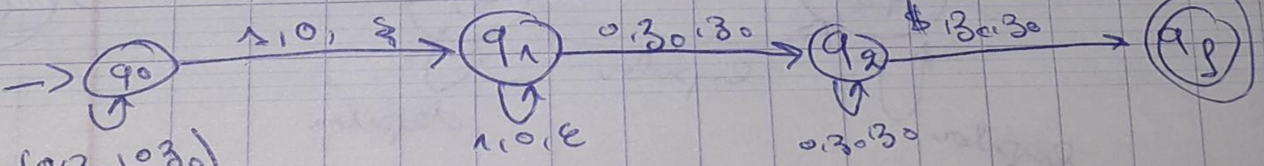

0.0.00

$L_6 =$

$\{0^n 1^m 0^n \mid n \geq 0, m \geq 0\}$

$\delta(q_0, 0, Z_0) = (q_0, 0Z_0)$ empiler le 1er 0
$\delta(q_0, 0, 0) = (q_0, 00)$ empiler tout les 0
$\delta(q_0, 1, 0) = (q_1, \varepsilon)$ dépiler 0
$\delta(q_1, 1, 0) = (q_1, \varepsilon)$ dépiler tout les 0
$\delta(q_1, 0, Z_0) = (q_2, Z_0)$ la pile reste inchager
$\delta(q_2, 0, Z_0) = (q_2, Z_0)$
$\delta(q_2, \$, Z_0) = (q_f, Z_0)$

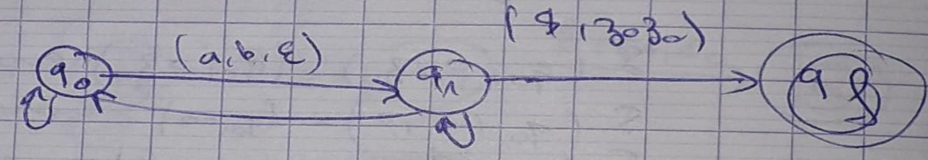
l'automate



$L = \{ w \in (a+b)^* \mid |w|_a = |w|_b \}$

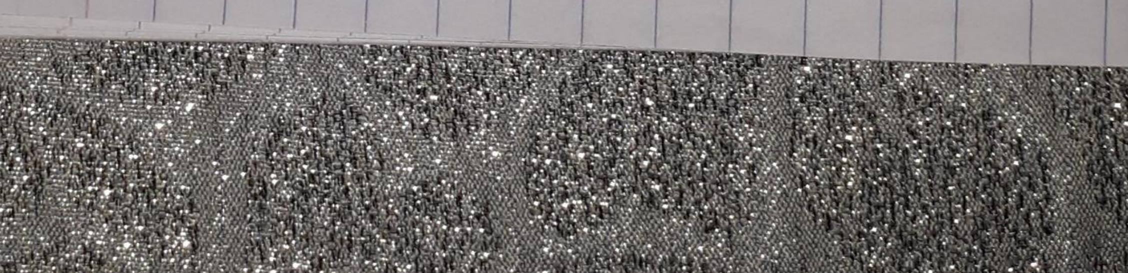
$\delta(q_0, a, Z_0) = (q_0, aZ_0)$ empiler le 1er a
$\delta(q_0, b, Z_0) = (q_0, bZ_0)$ " le 1er b
$\delta(q_0, a, a) = (q_0, aa)$ " tous les a
$\delta(q_0, b, a) = (q_1, \varepsilon)$ dépiler le a
$\delta(q_1, b, a) = (q_1, \varepsilon)$ dépiler tous les a
$\delta(q_1, \$, Z_0) = (q_f, Z_0)$ on accepte.

l'automate




$(a, Z_0, aZ_0)$ $(b, Z_0, bZ_0)$ $(b, a, \varepsilon)$
$(b, Z_0, bZ_0)$ $(a, a, aa)$ $(a, b, \varepsilon)$

$(a, a, aa)$ $(b, b, bb)$ $(b, b, bb)$



le mots aabbabab

$(q_0, aabbabab, Z_0) \vdash (q_0, abbabab, aZ_0)$

$\vdash (q_0, bbabab, aaZ_0) \vdash (q_1, babab, aZ_0)$

$\vdash (q_1, abab\ Z_0) \vdash (q_0, bab, aZ_0)$

$\vdash (q, ab, Z_0) \vdash (q_0, b, aZ_0) \vdash (q_0, \$, Z_0)$

$\vdash (q_f, \$, Z_0) \Rightarrow$ accepta

Département d'Informatique Licence académique $2^{ième}$ année

THEORIE DES LANGAGES : CONTROLE

Durée 1h30 documents non autorisés

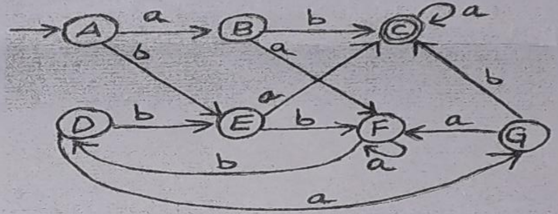
**Questions de cours (4 pts)** Selmani Skander

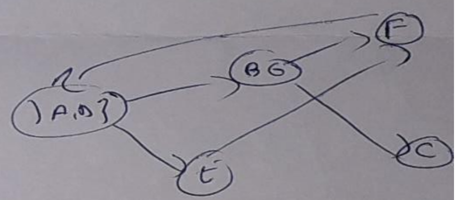
A) Pour chacune de ces affirmations, dire si elle est vraie ou fausse.
1. Tout langage régulier est infini. V
2. Tout langage non régulier est infini. F
3. Il y a une infinité de langages réguliers. F
4. Il y a une infinité de langages non réguliers. V

B) Il y a toujours exactement une seule réponse valable. Lorsque plusieurs réponses sont possibles, prendre la plus restrictive.
1. Le langage $\{a^n b^n \mid n \in N\}$ est :
   a) Fini b) Non régulier c) Reconnaissable par automate fini d) Vide
2. Quelle propriété de cette grammaire est vraie ?
   $S \longrightarrow aSc$
   $A \longrightarrow c$
   a) Linéaire à gauche b) Linéaire à droite c) Hors contexte d) Ambigüe
3. Quelle est la classe la plus stricte de la grammaire suivante ?
   S →inst ';' S
   S →inst ';'
   a) Régulière b) Hors contexte c) Contextuelle d) Sans restrictions

**EXERCICE 01 (4 pts)**

1) Donnez les éléments composant le quintuplet représentant l'automate suivant. (2pts)
2) Minimisez cet automate. (2pts)



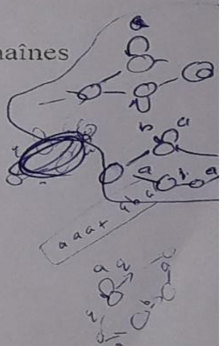



**EXERCICE 02 (5 pts)**

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 $\leq 2$ ; et le langage $L_2 = \{aaa, aba\}$.

3) Construire un automate d'états finis simple qui accepte $L_1$. (1,25pts)
4) Construire un automate d'états finis simple qui accepte $L_2$. (1,25pts)
5) Construire un automate d'états finis simple qui accepte $L_1 \cup L_2$. (1,25pts)
6) Rendre l'automate de [illegible]) déterministe, s'il ne l'est pas. (1,25pts)

5



**EXERCICE 03 (7 pts)**

Soit l'alphabet A = {a, b, c} et le langage :

$L = \{a^n b^p c^q \ / \ n,q >= 0, \ p >= (n+q)\}$.

1) Trouver une grammaire qui engendre ce langage et donner son type. (1,5+0,5=2pts)
2) Construire un automate à pile qui reconnait L avec n,q >= 0 et p=n+q. (2pts)
3) Expliquez son principe de fonctionnement. (1pt)
4) Vérifiez que l'automate prend en compte les cas où n=0, p=0 et q=0. (2pts)

Bon courage !