Université Alger 1-FS-Dept MI **THL** **LI/S3/2016-2017**

CONTROLE CONTINU N°1 **durée=35mn**

Documents: non autorisés **Groupe : 05** **date : 04/12/2016**

<u>Questions :</u>

1- Soit la grammaire suivante :

$S \rightarrow AB$
$A \rightarrow aAb \mid \varepsilon$
$B \rightarrow aBb \mid \varepsilon$

- Quel est le type de cette grammaire ?
- Quel est le langage généré par cette grammaire ?
- Donner les arbres de dérivation pour : **ab, abb, abab, aabbaabb**
- Cette grammaire est-elle ambigüe ? justifier votre réponse.

2- Soit V={a,b}.
- Donner une grammaire de type 3 générant V^+.

3- Trouver le langage L_i engendré par chacune des grammaires suivantes :
- G_1 : $S \rightarrow a \mid b \mid aSb$
- G_2 : $S \rightarrow \varepsilon \mid bBa$; $Ba \rightarrow baT$; $baT \rightarrow baaS$
- En déduire $L_1.L_2$

4- Représenter l'automate donné par la table de transition suivante :

	0	1
→E0	E1,E2	---
E1	---	E0
E2	E3	E2
E3	E4	----
*E4	---	---

- Cet automate est-il déterministe ? justifier ?

Bon courage

Université Alger 1-FS-Dept MI **THL** **LI/S3/2016-2017**

CONTROLE CONTINU N°1 **durée =35mn**

Documents: non autorisés **Groupe : 08** **date : 04/12/2016**

<u>Questions :</u>

1- Soit la grammaire suivante :

$S \rightarrow AB \mid C$
$A \rightarrow aAb \mid \varepsilon$
$B \rightarrow cBd \mid \varepsilon$
$C \rightarrow aCd \mid aDd$
$D \rightarrow bDc \mid \varepsilon$

- Donner le type de cette grammaire.
- Quel est le langage généré par cette grammaire ?
- Montrer que cette grammaire est ambigüe.

2- Soit V={c,d} ;
- Donner une grammaire de type 3 générant V^*.

3- Trouver les grammaires qui engendrent les langages suivants :
- $L_1 = \{a^n b^{n-1}\ ou\ \ a^{n-1} b^n \mid n \geq 1\}$
- $L_2 = \{a^n b^n \mid n \geq 0\}$
- En déduire $L_1.L_2$

4- Représenter l'automate donné par la table de transition suivante :

	a	b
→E0	E1	E0
E1	---	E2
*E2	E1	E2

- Cet automate est-il déterministe ? justifier ***?***

Bon courage

Corrigé du contrôle THL

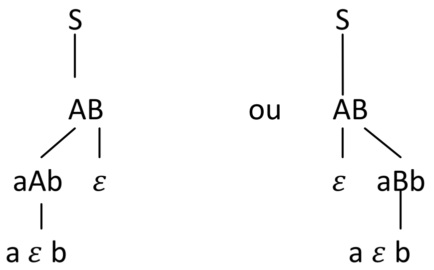
Sujet 1 (du groupe 5)

Réponses :

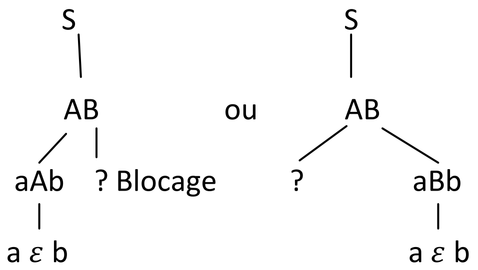
R1 :

1- La grammaire est de type 2
2- Le langage généré par cette grammaire est : $L=\{a^n b^n a^m b^m \;/\; n,m >=0\}$
3- Arbre de dérivation :

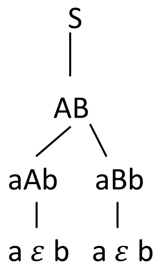
- **ab (2 arbres)**



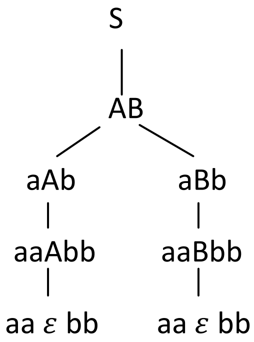
- **abb**



- **abab**



- **aabbaabb**



4- la grammaire est ambigüe car pour le mot **ab** , on a deux arbres de dérivation

R2 :

- donnons une grammaire de type 3 générant V^+, avec $V=\{0,1\}$
  $V^+=\{ w \in \{0,1\}^* \} - \{ \varepsilon\}$
  Soit S l'axiome de cette grammaire :
  S→ 0S | 1S |1 |0

R3 :

Trouvons les langages engendrés par chacune des grammaires suivantes :

- G_1 : $S_1 \to a$ (Règle 1)
  $S_1 \to b$ (Régle 2)
  $S_1 \to aS_1b$ (Règle 3)

$S_1 \xrightarrow{3} aS_1b \xrightarrow{3} aaS_1bb \xrightarrow{1} aaabb$ ou $S_1 \xrightarrow{3} aS_1b \xrightarrow{3} aaS_1bb \xrightarrow{2} aabbb$

$S_1 \xrightarrow{1} a$ ou $S_1 \xrightarrow{2} b$

Donc le langage généré par cette grammaire : $L(G_1)=\{ a^n b^{n-1}$ ou $a^{n-1}b^n \;|\; n>=1\}$

- $G_2 : S_2 \xrightarrow{1\ et\ 2} \varepsilon \mid bBa\ ; Ba \xrightarrow{3} baT\ ; baT \xrightarrow{4} baaS_2$

$$S_2 \xrightarrow{2} bBa \xrightarrow{3} bbaT \xrightarrow{4} baaS_2 \xrightarrow{1} baa$$

$$S_2 \xrightarrow{1} \varepsilon$$

$$baaS_2 \xrightarrow{1} baabBa \xrightarrow{3} baabbaT \xrightarrow{4} baabaaS_2 \xrightarrow{1} baabaa$$

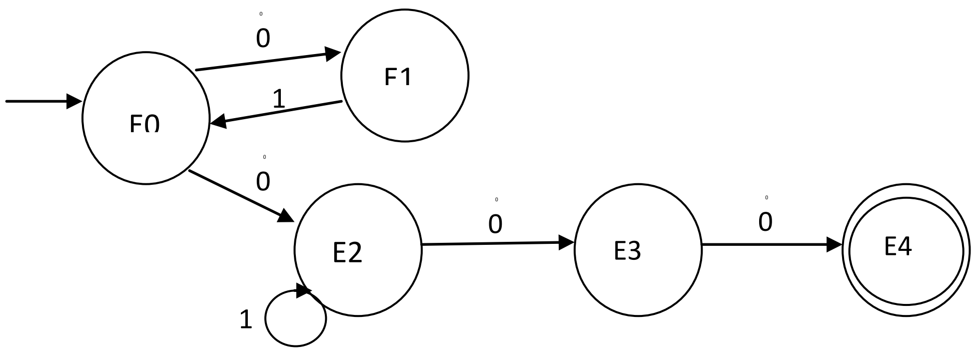
......

Donc le langage engendré par cette grammaire est : $L(G_2)=\{ (baa)^n \mid n>=0\}$

- En déduire $L_1.L_2$:
  Soit S l'axiome de la grammaire G qui engendre $L_1.L_2$:
  Il suffit d'ajouter une règle d'axiome S et l'axiome S_1 (resp S_2) de la grammaire G_1 (resp G_2) devient un non-terminal :
  Les règles de production de G sont les suivantes:
  $S \to S_1S_2$
  $S_1 \to a \mid b \mid a\, S_1b$
  $S_2 \to \varepsilon \mid bBa$
  $Ba \to baT$
  $baT \to baaS_2$

R4 :

L'automate correspondant à la table de transition est :



L'automate n'est pas déterministe car il ya 2 arcs étiquetés par 0 partant de l'état E0 (la transition par 0 n'est pas unique)

Sujet 2 (du groupe 8)

Réponses :

R1 :

1- La grammaire est de type 2
2- Le langage généré par cette grammaire est : $L=\{a^nb^nc^md^m$ avec n,m >=0 ou $a^kb^lc^ld^k$ avec k>l>=0}
3- La grammaire est ambigüe car le mot abcd possède deux arbres de dérivation

abcd

S→AB→aAbB→abB→abcBd→abcd

Ou

S→C→aDd→abDcd→abcd

R2:

- donnons une grammaire de type 3 générant V^*, avec V={0,1}

  $V^* = V^+ + \{ \varepsilon\}$

  Soit S l'axiome de cette grammaire :

  S→ 0S | 1S | $\boldsymbol{\varepsilon}$

R3 :

Trouvons les grammaires qui engendrent les langages suivants :

- $L_1 = \{a^n b^{n-1}\ ou\ \ a^{n-1}b^n\ |n \geq 1\}$

  $G_1 : S_1 \rightarrow a \mid b \mid aS_1b$

- $L_2 = \{a^n b^n\ |n \geq 0\}$

  $G_2 : S_2 \rightarrow\ aS_2b \mid \varepsilon$

- En déduire $L_1.L_2$

  Soit S l'axiome de la grammaire G qui engendre $L_1.L_2$:

  Il suffit d'ajouter une règle d'axiome S et l'axiome S_1 (resp S_2) de la grammaire G_1 (resp G_2) devient un non-terminal :

  Les règles de production de G sont les suivantes:

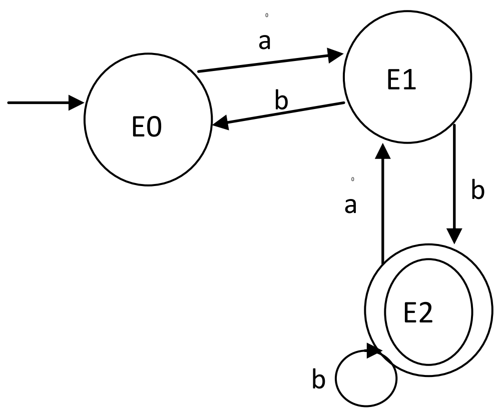
  $S \rightarrow S_1S_2$

  $S_1 \rightarrow a \mid b\ \mid\ a\,S_1b$

  $S_2 \rightarrow\ a\,S_2b\ \mid\ \varepsilon$

R4 :

L'automate correspondant à la table de transition est :



L'automate n'est pas déterministe car il ya 2 arcs étiquetés par b partant de l'état E1 (la transition par b n'est pas unique)

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Alger1 – Benyoucef Benkhadda
Faculté des Sciences
Département Mathématiques Informatique

Corrigé de l'examen THL

Exercice 1 (cours) (3 pts)

Transformer la grammaire suivante en FNC :

- G_1({a,b}, {S,A,B,C,D}, S, P) avec
  P= { S→ABCD
  A→aA | ε
  B→Ab
  C→ A | aaB
  D→ aA |B }

Pour pouvoir transformer une grammaire en sa FNC, il faut que cette grammaire soit réduite et propre **(0.5)**

1- Grammaire réduite : c'est une grammaire qui contient que des symboles non terminaux utiles, il faut éliminer les symboles inaccessibles et improductifs
La grammaire G_1 est **réduite** car tous ses symboles non terminaux sont utiles. **(0.25)**

2- Grammaire propre : c'est une grammaire sans cycle et libre- ε. La grammaire G_1 est sans cycle mais non libre- ε , donc il faut la transformer en une grammaire libre- ε. **(0.25)**

3- Éliminer les règles de productions Immédiates (A→ ε) : **(0.5)**

- On peut donc transformer la grammaire en éliminant toutes les règles de production immédiates et en modifiant les autres règles, on obtient la grammaire G' dont
  P_1= { S→ABCD | BCD | ABD | BD
  A→aA | a
  B→Ab | b
  C→A | aaB
  D→ aA | a |B }

4- Eliminer les règles de production unitaires (C→A ; D→B) en apportant les modifications sur P_1 suivantes : On obtient un nouveau ensemble de production : **(0.5)**
P_2={ S→ABCD | BCD | ABD | BD
A→aA | a
B→Ab | b
C→aA | a | aaB
D→ aA | a |Ab | b}

5- Modifier les règles qui contiennent des symboles non terminaux et des symboles terminaux :
La grammaire est définie sur l'alphabet {a,b} donc on ajoute 2 symboles non terminaux , ainsi que les règles de production E→a et F→b , puis on peut modifier les autres règles en conséquence : **(0.5)**
P_3 = { S→ABCD | BCD | ABD |BD
A→EA | a
B→AF | b
C→EA | a | EEB
D→ EA | a |AF | b
E→a
F→b }

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Alger1 – Benyoucef Benkhadda
Faculté des Sciences
Département Mathématiques Informatique

Corrigé de l'examen THL

6- Appliquer le cas3 de l'algorithme de transformation en FNC :
On obtient la grammaire dont l'ensemble des règles de production est le suivant :

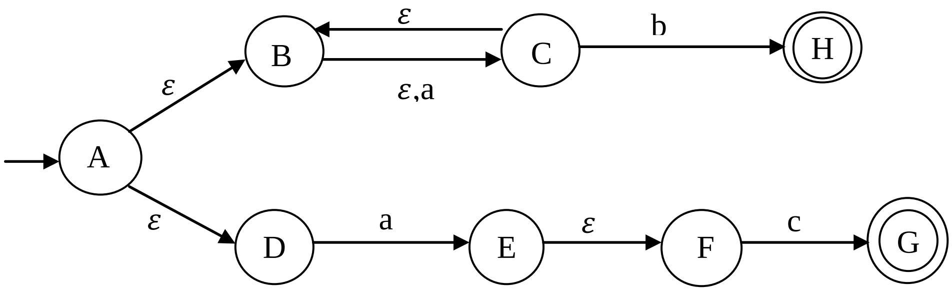
P_4= { S → AX_1 | BX_2 | AX_3 | BD
A → EA | ab
B → AF | b
C → EA | a | EX_4
D → EA | a | AF | b
E → a
F → b
X_1 → BX_2
X_2 → CD
X_3 → BD
X_4 → EB} la grammaire est maintenant sous FNC

(0.5)

Exercice 2 (5pts)

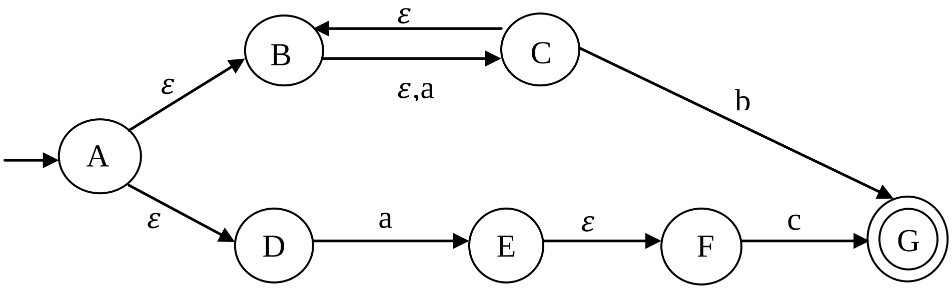
Soit l'expression suivante : $a^*.b + a.c$ définie sur l'alphabet X={*a*, *b*, *c*}.

1- Donnez l'automate T de Thompson associé.

T=


On peut modifier l'automate en gardant un seul état final

(1.25)

T=


T est un automate non déterministe avec des ε-transitions

2- Donnez un automate A non déterministe et sans ε-transitions correspondant.

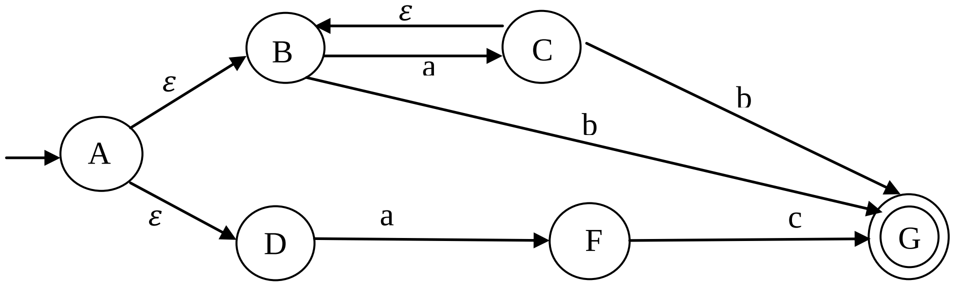
➢ Elimination des ε-transitions

$B \xrightarrow{\varepsilon} C \xrightarrow{b} G \quad donc \quad B \xrightarrow{b} G$ **(0.25)**

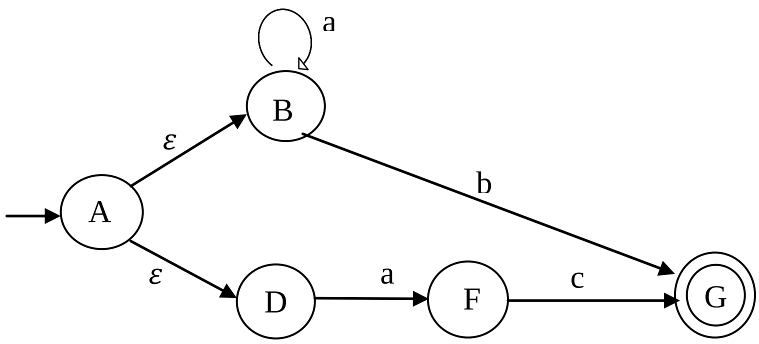
Corrigé de l'examen THL

$D \overset{a}{\rightarrow} E \overset{\varepsilon}{\rightarrow} F \quad donc \quad D \overset{a}{\rightarrow} F \quad ou\ bien\ choisir\ un\ autre\ chemin\ \ E \overset{\varepsilon}{\rightarrow} F \overset{c}{\rightarrow} G \quad donc \quad E \overset{c}{\rightarrow} G$

Eliminer l'état E ou l'état F **(0.25)**



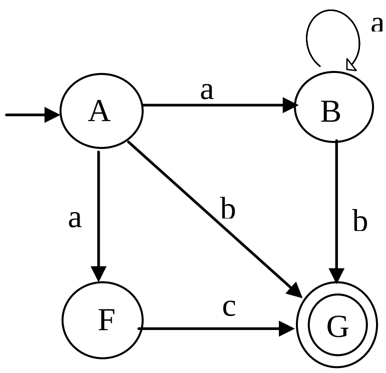
$B \overset{a}{\rightarrow} C \overset{\varepsilon}{\rightarrow} B \quad donc \quad B \overset{a}{\rightarrow} B$ **(0.25)**



$A \overset{\varepsilon}{\rightarrow} B \overset{a}{\rightarrow} B \quad donc \quad A \overset{a}{\rightarrow} B \quad et \quad A \overset{\varepsilon}{\rightarrow} B \overset{b}{\rightarrow} G \quad donc \quad A \overset{b}{\rightarrow} G$ **(0.25)**

$A \overset{\varepsilon}{\rightarrow} D \overset{a}{\rightarrow} F \quad donc \quad A \overset{a}{\rightarrow} F$ **(0.25)**

Donc l'automate A non déterministe et sans ε-transitions est comme suit : (0.5)



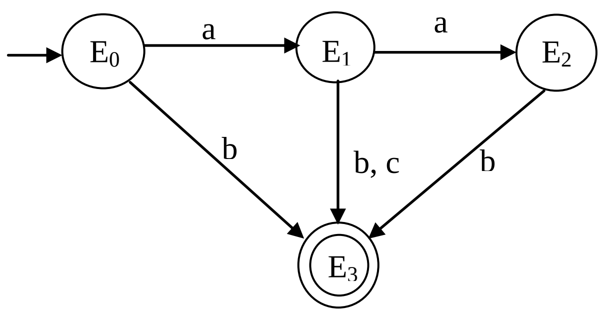
Corrigé de l'examen THL

3- Donnez l'automate du langage complémentaire qu'elle représente

Pour pouvoir construire le complémentaire d'un automate il faut que ce dernier soit déterministe et complet

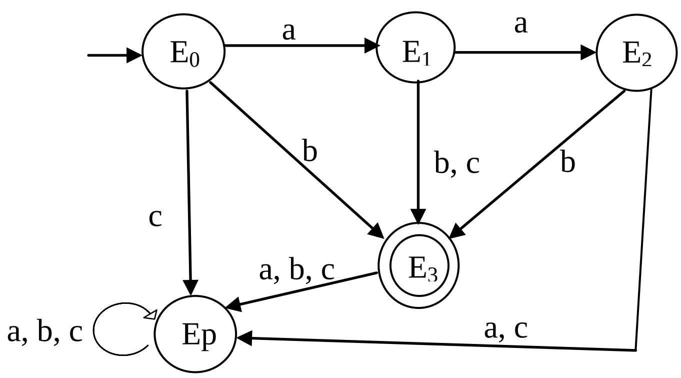
1- Déterminisation de l'automate A : (0.5)

	A	**B**	**c**
→{A} =E_0	{B, F}	{G}	--
{B, F}=E_1	{B}	{G}	{G}
***{G}=E_3**	--	--	--
{B}=E_2	{B}	{G}	--

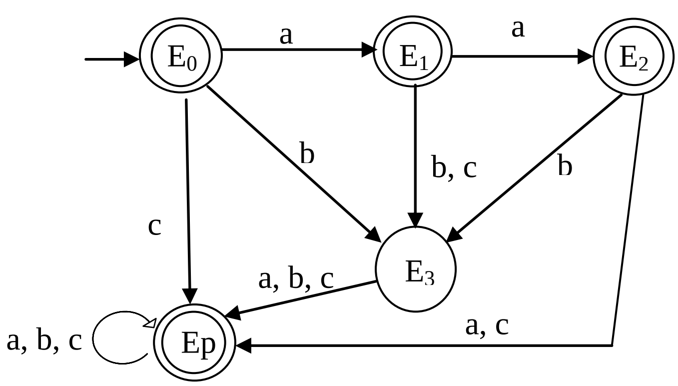


(0.5)

2- Compléter l'automate : (0.5)



Dans l'automate complémentaire de A déterministe (A_det), les états finaux de l'automate A_det deviennent simples et les états simples deviennent finaux.



(0.5)

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Alger1 – Benyoucef Benkhadda
Faculté des Sciences
Département Mathématiques Informatique

Corrigé de l'examen THL

Exercice 3 (6pts)

Soit G({a,b},{S, A, B}, S, P) avec P : { *S→AbB (1)*
A→aA (2)
A→ε (3)
B→aB (4)
B→bB (5)
B→ε (6) }

1-Dérivations gauches : (1.5) (0.5 pour chaque mot)

- ***$W_1=a^2bab=aabab$***

$S \overset{(1)}{\rightarrow} AbB \overset{(2)}{\rightarrow} aAbB \overset{(2)}{\rightarrow} aaAbB \overset{(3)}{\rightarrow} aabB \overset{(4)}{\rightarrow} aabaB \overset{(5)}{\rightarrow} aababB \overset{(6)}{\rightarrow} aabab$; Mot accepté.

- ***$W_2=ba^2b=baab$***

$S \overset{(1)}{\rightarrow} AbB \overset{(3)}{\rightarrow} bB \overset{(4)}{\rightarrow} baB \overset{(4)}{\rightarrow} baaB \overset{(5)}{\rightarrow} baabB \overset{(6)}{\rightarrow} baab$; Mot accepté.

- ***$W_3=a^3b^2=aaabb$***

$S \overset{(1)}{\rightarrow} AbB \overset{(2)}{\rightarrow} aAbB \overset{(2)}{\rightarrow} aaAbB \overset{(2)}{\rightarrow} aaaAbB \overset{(3)}{\rightarrow} aaabB \overset{(5)}{\rightarrow} aaabbB \overset{(6)}{\rightarrow} aaabb$; Mot accepté.

2- Dérivations droites : (1.5) (0.5 pour chaque mot)

- ***$W_1=a^2bab=aabab$***

$S \overset{(1)}{\rightarrow} AbB \overset{(4)}{\rightarrow} AbaB \overset{(5)}{\rightarrow} AbabB \overset{(6)}{\rightarrow} Abab \overset{(2)}{\rightarrow} aAbab \overset{(2)}{\rightarrow} aaAbab \overset{(3)}{\rightarrow} aabab$; Mot acceptée

- ***$W_2=ba^2b=baab$***

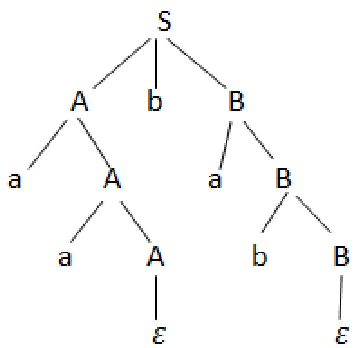
$S \overset{(1)}{\rightarrow} AbB \overset{(4)}{\rightarrow} AbaB \overset{(4)}{\rightarrow} AbaaB \overset{(5)}{\rightarrow} AbaabB \overset{(6)}{\rightarrow} Abaab \overset{(3)}{\rightarrow} baab$

- ***$W_3=a^3b^2=aaabb$***

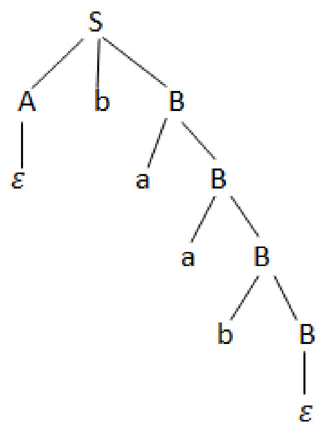
$S \overset{(1)}{\rightarrow} AbB \overset{(5)}{\rightarrow} AbbB \overset{(6)}{\rightarrow} Abb \overset{(2)}{\rightarrow} aAbb \overset{(2)}{\rightarrow} aaAbb \overset{(2)}{\rightarrow} aaaAbb \overset{(3)}{\rightarrow} aaabb$

3- Arbre syntaxique : (1.5) (0.5 pour chaque arbre)

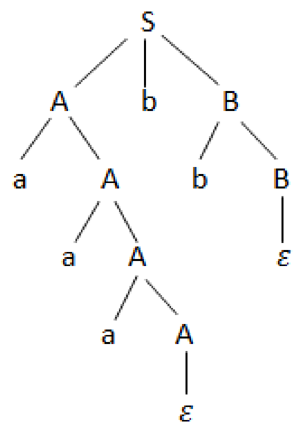
- ***$W_1=a^2bab=aabab$***



$W_2=ba^2b=baab$



$W_3=a^3b^2=aaabb$



Quelque soit la dérivation gauche ou droite, la structure de l'arbre de dérivation est la même.

Corrigé de l'examen THL

4- Le langage engendré par la grammaire G est : *L(G)= {w Є {a, b}*/ w=* $a^i b a^k b^j$ *; i, j, k >=0}* **(1)**

5- On remarque que pour tout w_i, il n'est pas possible de trouver deux arbres de dérivation différents, par conséquent la grammaire n'est pas ambigüe. **(0.5)**

Exercice 4 (6 pts)

1- Trouver les grammaires qui engendrent les langages suivants :

- $L_1 = \{a^i b^j \mid i > j \geq 0\}$
- $L_2 = \{a^i b^j \mid j > i \geq 0\}$

- La grammaire qui engendre le langage L_1 peut être dénotée avec les règles de production suivantes :
  A→aAb | aA | a **(1)**

- La grammaire qui engendre le langage L_2 peut être dénotée avec les règles de production suivantes :
  B→aBb | aB | a **(1)**

2- En déduire la grammaire de $L = L_1 + L_2$

La grammaire qui engendre L est donnée par les règles de production antérieures, plus la règle suivante :

S → A | B

Donc la grammaire G qui engendre L est définie comme suit :

G({a,b},{S,A,B}, S, P) avec P= { S → A | B
A→ aAb | aA | a
B→ aBb | bB |b }

(1)

3- Construisons l'automate à pile (AP) reconnaissant le langage L :

Les instructions (transitions) d'AP sont données dans la table suivante : **(2)**

On suppose que l'automate commence avec une pile contenant le symbole #

Représentation 1	Représentation 2	Signification
$\delta(q_0, a, \#) = (q_1, a\#)$	$\#q_0a \rightarrow \#aq_1$	Empilement du 1er a (i≠0)
$\delta(q_1, a, a) = (q_1, aa)$	$aq_1a \rightarrow aaq_1$	Empilement de tous les a
$\delta(q_1, \varepsilon, a) = (q_2, \varepsilon)$	$aq_1 \rightarrow q_2$	Dépiler le dernier a empilé (j=0 ; pas de b ou j<i)
$\delta(q_2, \varepsilon, a) = (q_2, \varepsilon)$	$aq_2 \rightarrow q_2$	Dépiler tous les a (vider la pile des a) ; q_2 état final (j=0)
$\delta(q_1, b, a) = (q_1, \varepsilon)$	$aq_1b \rightarrow q_1$	Dépiler a en lecture d'un b (j≠0)
$\delta(q_0, b, \#) = (q_3, \#)$	$\#q_0b \rightarrow \#q_3$	Lecture de b (i=0 ; pas de a) sans changement de l'état de la pile (pile = #) (j>i)
$\delta(q_3, b, \#) = (q_3, \#)$	$\#q_3b \rightarrow \#q_3$	Lecture de tous les b ; q_3 état final
$\delta(q_1, b, \#) = (q_3, \#)$	$\#q_1b \rightarrow \#q_3$	Le cas où i ≠0 et j>i ; lecture de tous les b qui restent ; pile =# ; q_3 état final
$\delta(q_3, \varepsilon, \#) = (q_3, \#)$	$\#q_3 \rightarrow \#q_3$	q_3 : état final servant à accepter les mots qui ont plus de b
$\delta(q_2, \varepsilon, \#) = (q_2, \#)$	$\#q_2 \rightarrow \#q_2$	q_2 : état final servant à accepter les mots qui ont plus de a

Corrigé de l'examen THL

Représentation graphique : (1)