Correction Série 3

Cours : Théorie de langages et compilation

Filière/Classe : 2ème ING

Sections : A, B, C &D

Exercice 1

Considérons la grammaire G_1 dont les productions sont les suivantes :

$S \rightarrow SaSaSbS|SaSbSaS|SbSaSaS|\varepsilon$

1. La grammaire est-elle régulière ? Justifier.
2. Trouver $L(G_1)$.
3. Donner une dérivation à gauche de la séquence abaaab.
4. La grammaire est-elle ambigüe ? Justifier.

Correction :

1. La grammaire est non régulière puisque le côté droit des règles ne respecte pas la règle des grammaires régulières : ∊Terminal ou xF tel que x Terminal et F NonTerminal, ou ε. Voici un contre-exemple : $S \rightarrow SaSaSbS$.
2. $L(G_1)=\{w\in\{a,b\}^* \mid |a| = 2|b| \}$.
3. S⇒SaSaSbS⇒aSaSbS⇒aSbSaSaSaSbS⇒abSaSaSaSbS⇒abaSaSaSbS⇒abaaSaSbS⇒abaaaSbS⇒abaaabS⇒abaaab
4. Oui, puisqu'il y a 2 dérivations pour abaaabaab :

   Dérivation1 :
   S⇒SaSaSbS⇒SaSaSbSaSaSbS⇒SaSaSbSaSaSbS⇒aSaSbSaSaSbS⇒aSbSaSaSaSbSaSaSbS⇒abSaSaSaSbSaSaSbS⇒abaSaSaSbSaSaSbS⇒abaaSaSbSaSaSbS⇒abaaaSbSaSaSbS⇒abaaabSaSaSbS⇒abaaabaSaSbS⇒abaaabaaSbS⇒abaaabaabS⇒abaaabaab

   Dérivation2 :
   S⇒SaSaSbS⇒aSaSbS⇒aSbSaSaSaSbS⇒abSaSaSaSbS⇒abaSaSaSbS⇒abaSaSaSbSaSaSbS⇒abaaSaSbSaSaSbS⇒abaaaSbSaSaSbS⇒abaaabSaSaSbS⇒abaaabaSaSbS⇒abaaabaaSbS⇒abaaabaabS⇒abaaabaab

Exercice 2

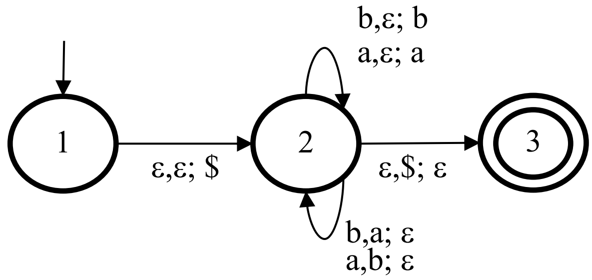
Pour chacun des langages suivants :

a. $L_1 = \{u \in \{a,b\}^* : \ |u|_a = |u|_b\}$
b. $L_2 = \{w\overline{w} : w \in \{a,b\}^* \ et \ \overline{w} \ est \ le \ mot \ miroir \ de \ w \}$
c. $L_3 = \{a^m b^n c^m : n \geq 0 \ et \ m > 0\}$
d. $L_4 = \{a^n b^m c^m d^{2n} : n \geq 0 \ et \ m \geq 0\}$

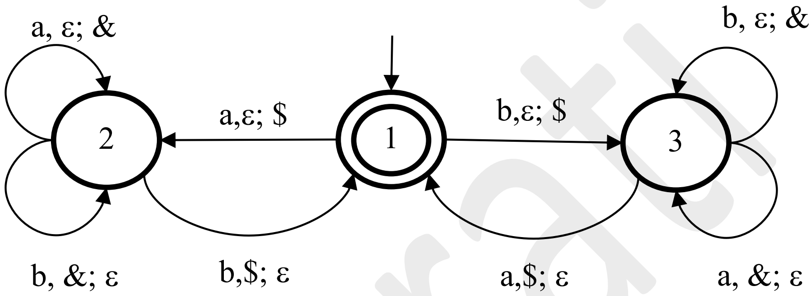
1. Donner un automate à pile

Correction

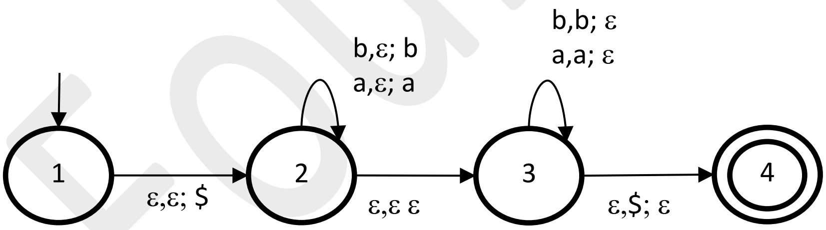
a. $L_1 = \{u \in \{a,b\}^* : \ |u|_a = |u|_b\}$



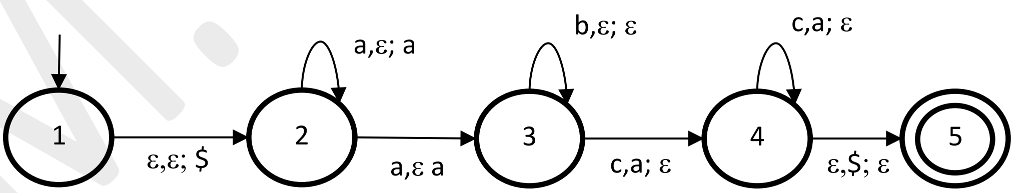
Déterministe



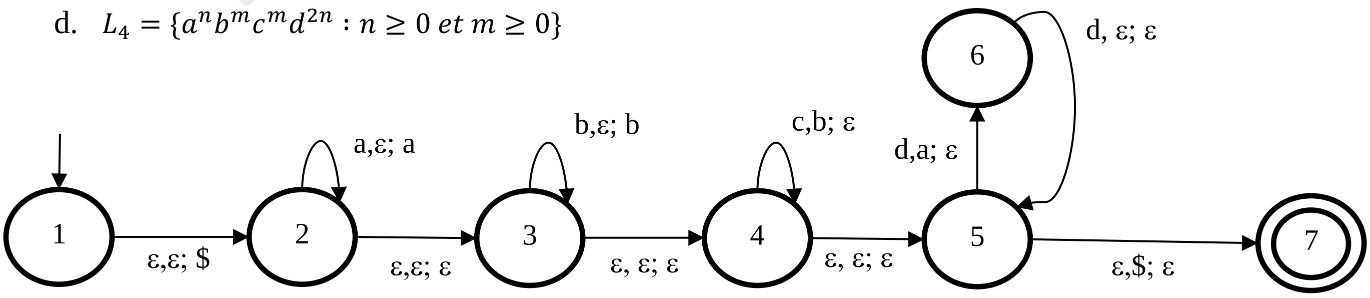
b. $L_2 = \{w\overline{w} : w \in \{a,b\}^* \ et \ \overline{w} \ est \ le \ mot \ miroir \ de \ w\}$



c. $L_3 = \{a^m b^n c^m : n \geq 0 \ et \ m > 0\}$



d. $L_4 = \{a^n b^m c^m d^{2n} : n \geq 0 \ et \ m \geq 0\}$



2. Donner une grammaire qui le génère.

Correction :

a. $L_1 = \{u \in \{a,b\}^* : |u|_a = |u|_b\}$

$S \rightarrow aSbS \mid bSaS \mid \varepsilon$

b. $L_2 = \{w\overline{w} : w \in \{a,b\}^* \; et \; \overline{w} \; est \; le \; mot \; miroir \; de \; w\}$

$S \rightarrow aSa \mid bSb \mid \varepsilon$

Sachez que si le langage était $\{wx\overline{w} : w \in \{a,b\}^* \; et \; \overline{w} \; est \; le \; mot \; miroir \; de \; w \; et \; x = a \; ou \; x = b\}$, la grammaire serait : $S \rightarrow aSa \mid bSb \mid a \mid b \mid \varepsilon$

c. $L_3 = \{a^m b^n c^m : n \geq 0 \; et \; m > 0\}$

$S \rightarrow aSc \mid aBc$
$B \rightarrow bB \mid \varepsilon$

e. $L_4 = \{a^n b^m c^m d^{2n} : n \geq 0 \; et \; m \geq 0\}$

$S \rightarrow aSdd \mid B$

$B \rightarrow bBc \mid \varepsilon$

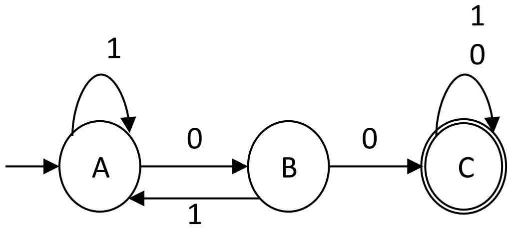
Exercice 3 :

Créer un automate pour chacune de ces expressions régulières puis donner la grammaire correspondant à chaque automate

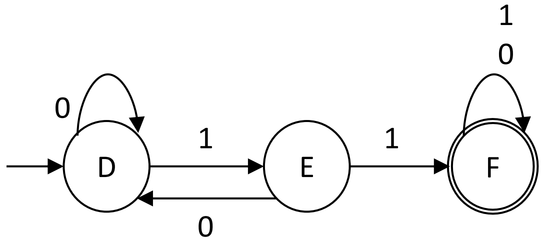
- $1^* + 10^*1^*0$
- $\{w \mid w \in \{0, 1\}^*$ et w contient la sous-chaîne 00 ou 11$\}$
- $\{w \mid w \in \{0, 1\}^*$ et w ne se termine pas par 01$\}$

Correction :

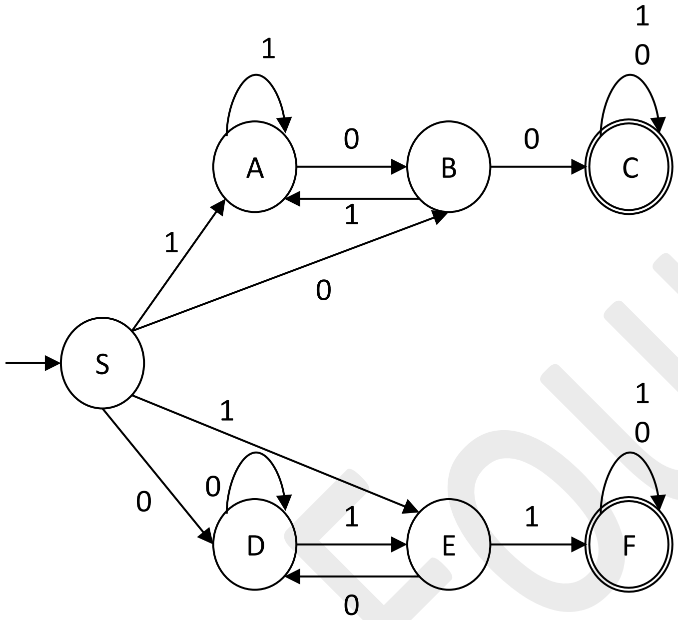
Vous pouvez construire l'automate qui contient 00 :



Ensuite construire l'automate qui contient 11 :



Ensuite vous n'avez que faire l'union :



La grammaire :

S→1A|0B|1E|0D

A→1A|0B

B→1A|0C

C→1C|0C|ε

D→1E|0D

E→1F|0D

F→1F|0F|ε

Remarque :

Ainsi vous obtenez une grammaire régulière et c'est normal car le langage est régulier. Mais faites attention, si le langage n'est pas régulier c'est qu'il n'existe pas d'automate fini qui le reconnait mais plutôt un automate à pile et donc pour trouver la grammaire, il faut trouver la grammaire pour chaque langage sans passer par l'automate car on ne peut pas (du moins on l'a pas fait en classe) passer de l'automate à pile à la grammaire mais plutôt de la grammaire à l'automate à pile. Une fois que la grammaire trouvée (de tête), vous appliquer l'union de grammaires hors contextes qu'on a vu en cours.

Exercice 4

Considérons les grammaires suivantes :

Grammaire G_2:

$S \to SaAb|bBaS|\varepsilon$
$A \to bAa|\varepsilon$
$B \to aBb|\varepsilon$

Grammaire G_3:

S → ST | SSb | TS | a
T → Sa | Tb |ε

Grammaire G_4:

S →S ∨ T | T
T →T ∧ F | F
F → ¬ F| (S)| x | y

1. De quel(s) type(s) sont ces grammaires.
2. La grammaire G_2 est-elle sous forme normale binaire de Chomsky ? Si non donner une grammaire sous forme normale binaire de Chomsky qui génère le même langage.
3. Montrer que la grammaire G_2 est ambiguë.
4. Éliminer la récursivité à gauche dans chacune de ces grammaires.
5. Factoriser à gauche chacune de ces grammaires
6. Donner la grammaire générant $L(G_2)^*$.
7. Donner la grammaire générant $L(G_4) \cup L(G_3)$.
8. Construire un automate à pile acceptant par état final pour la grammaire G_3, le transformer en un automate à pile acceptant par pile vide.
9. Construire un automate à pile acceptant par pile vide pour la grammaire G_4, le transformer en un automate à pile acceptant par état final.

Correction :

1. Ces grammaires sont hors contexte, car elles ne respectent pas les conditions des grammaires régulières mais respectent les règles de grammaires HC (voir cours).
2. Pour mettre cette grammaire sous forme normale binaire de Chomsky, il faut éliminer les règles-ε. Cependant, comme elle génère ε, il faut avant tout ajouter un nouveau symbole S' qui sera le nouveau axiome de la grammaire et qui sera le seul à générer ε. Ainsi, les symboles générateurs d'epsilon sont U={S,A,B} (cet ensemble doit être calculé avant l'ajout de S'). Nous obtenons donc la nouvelle grammaire :

$S' \to S|\varepsilon$
$S \to SaAb|bBaS|\varepsilon$
$A \to bAa|\varepsilon$
$B \to aBb|\varepsilon$

Nous ajoutons donc les règles suivantes après élimination d'un ou plusieurs symboles de U (les règles en fluo sont les règles d'origine, et celles en-dessous sont celles générées en éliminant un ou plusieurs symboles de U).

$S' \to S$	$S \to SaAb$	$S \to bBaS$	$A \to bAa$	$B \to aBb$
$S' \to \varepsilon$	$S \to aAb$	$S \to baS$	$A \to ba$	$B \to ab$
	$S \to Sab$	$S \to bBa$		
	$S \to ab$	$S \to ba$		

Nous ajoutons deux symboles non terminaux X et Y et les règles $X \to a$ et $Y \to b$.

$S' \to S$	$S \to SXAY$	$S \to YBXS$	$A \to YAX$	$B \to XBY$
$S' \to \varepsilon$	$S \to XAY$	$S \to YXS$	$A \to YX$	$B \to XY$
	$S \to SXY$	$S \to YBX$	$X \to a$	
	$S \to XY$	$S \to YX$	$Y \to b$	

Nous ramenons le côté droit des règles à deux non terminaux ou un terminal.

$S' \to S$ $S' \to \varepsilon$	$S \to SR_1$ $R_1 \to XR_2$ $R_2 \to AY$	$S \to YR_4$ $R_4 \to BR_5$ $R_5 \to XS$	$A \to YR_6$ $R_6 \to AX$	$B \to XR_7$ $R_7 \to BY$
	$S \to XR_2$	$S \to YR_5$	$A \to YX$	$B \to XY$
	$S \to SR_3$ $R_3 \to XY$	$S \to YR_5$ $R_5 \to BX$	$X \to a$	
	$S \to XY$	$S \to YX$	$Y \to b$	

Nous éliminons les règles *Non-Terminal*→ *Non-Terminal* ($S' \to S$) par propagation.

$\cancel{S' \to S}$ $S' \to SR_1$ $S' \to XR_2$	$S \to SR_1$ $R_1 \to XR_2$ $R_2 \to AY$	$S \to YR_4$ $R_4 \to BR_5$ $R_5 \to XS$	$A \to YR_6$ $R_6 \to AX$	$B \to XR_7$ $R_7 \to BY$
$S' \to SR_3$ $S' \to XY$	$S \to XR_2$	$S \to YR_5$	$A \to YX$	$B \to XY$
$S' \to YR_4$ $S' \to YR_5$ $S' \to YX$	$S \to SR_3$ $R_3 \to XY$	$S \to YR_5$ $R_5 \to BX$	$X \to a$	
$S' \to \varepsilon$	$S \to XY$	$S \to YX$	$Y \to b$	

Les règles définitives.

$S' \to SR_1$	$S \to SR_1$	$S \to YR_4$	$A \to YR_6$	$B \to XR_7$

$S' \to XR_2$	$R_1 \to XR_2$	$R_4 \to BR_5$	$R_6 \to AX$	$R_7 \to BY$
	$R_2 \to AY$	$R_5 \to XS$		
$S' \to SR_3$	$S \to XR_2$	$S \to YR_5$	$A \to YX$	$B \to XY$
$S' \to XY$				
$S' \to YR_4$	$S \to SR_3$	$S \to YR_5$	$X \to a$	
$S' \to YR_5$	$R_3 \to XY$	$R_5 \to BX$		
$S' \to YX$				
$S' \to \varepsilon$	$S \to XY$	$S \to YX$	$Y \to b$	

3. La grammaire est ambiguë, voici deux dérivations pour la même séquence :
   $S \Longrightarrow SaAb \Longrightarrow bBaSaAb \overset{*}{\Rightarrow} baab$ (A, B et S sont remplacés par ε).
   $S \Rightarrow bBaS \Rightarrow bBaSaAb \overset{*}{\Rightarrow} baab$ (A, B et S sont remplacés par ε).

4. Elimination de la récursivité à gauche des grammaires récursives.

Grammaire G_2 :

$S \to SaAb|bBaS|\varepsilon$
$A \to bAa|\varepsilon$
$B \to aBb|\varepsilon$

La grammaire est récursive à gauche $\boldsymbol{S} \to \boldsymbol{S}aAb$.

Nous commençons par éliminer les règles epsilon (nous l'avons déjà fait) :

$S' \to S|\varepsilon$
$S \to SaAb|aAb|Sab|ab|bBaS|baS|bBa|ba$
$A \to bAa|ba$
$B \to aBb|ab$

Nous choisissons l'ordre de traitement : S'(1), S(2), A(3), B(4)
i=1 (S')
j=0 : *boucle vide*
Élimination de la récursivité immédiate sur S' : ***rien à faire***.

i=2 (S)
j=1 (S') : *pas de règles* $S \to S'\alpha$ donc ***rien à faire***.
Élimination de la récursivité immédiate sur S :
$S \to SaAb|aAb|Sab|ab|bBaS|baS|bBa|ba$
Deviennent
$S \to aAbS''|abS''|bBaSS''|baSS''|bBaS''|baS''$
$S'' \to aAbS''|abS''|\varepsilon$

i=3 (A)
j=1 (S') : *pas de règles* $A \to S'\alpha$ donc ***rien à faire***
j=2 (S) : *pas de règles* $A \to S\alpha$ donc ***rien à faire***.
Élimination de la récursivité immédiate sur A : ***rien à faire***.

i=4 (B)
j=1 (S') : *pas de règles* $B \to S'\alpha$ donc ***rien à faire***
j=2 (S) : *pas de règles* $B \to S\alpha$ donc ***rien à faire***
j=3 (A) : *pas de règles* $B \to A\alpha$ donc ***rien à faire***.
Élimination de la récursivité immédiate sur B : ***rien à faire***.

Donc la grammaire G_2 finale est :
$S' \to S|\varepsilon$
$S \to aAbS''|abS''|bBaSS''|baSS''|bBaS''|baS''$
$S'' \to aAbS''|abS''|\varepsilon$
$A \to bAa|ba$
$B \to aBb|ab$

Grammaire G_3:

$S \to ST|SSb|TS|a$
$T \to Sa|Tb|\varepsilon$

Cette grammaire est récursive à gauche (directe (en jaune) et indirecte (en vert)) :

Nous commençons par éliminer les règles epsilon :

L'ensemble des symboles générateurs d'epsilon U={T}

$S \to ST$	$S \to SSb$	$S \to TS$	$S \to a$	T → Sa	T → Tb
$S \to S$		$S \to S$			T → b

La nouvelle grammaire devient (les règles $S \to S$ sont éliminées) :

S → ST | SSb | TS | a
T → Sa | Tb | b

Nous choisissons l'ordre de traitement : S(1), T(2)
i=1 (S)
j=0 : *boucle vide*
Élimination de la récursivité immédiate sur S :
S → TSS' | aS'
S' → TS' | SbS'|ε
T → Sa | Tb | b

i=2 (T)
j=1 (S) : *règles de la forme* $T \to S\alpha$: T → Sa *devient*

T → TSS'a | aS'a

Nouvelles règles :

S → TSS' | aS'
S' → TS' | SbS'|ε
T → TSS'a | aS'a | Tb |b

Élimination de la récursivité immédiate sur T, T → TSS'a | aS'a | Tb |b *deviennent* :

T → aS'aT' | *bT'*
T' → SS'aT' | *bT'*|ε

Grammaire G_3 finale :
Donc la grammaire G_3 finale est :
S → TSS' | aS'
S' → TS' | SbS'|ε
T → aS'aT' | Tb |bT'
T' → SS'aT' | bT'|ε

Grammaire G_4:

S → S ∨ T | T
T → T ∧ F | F
F → ¬F | (S) | x | y

La grammaire G4 n'a pas de productions-ε, ni de cycles. Elle est récursive à gauche (**S** → **S** ∨ T) et **T** → **T** ∧ F.

Nous choisissons l'ordre de traitement : S(1), T(2), F(3)

i=1 (S)
j=0 : *boucle vide*
Élimination de la récursivité immédiate sur S : S → S ∨ T | T *devienne*nt :
S → TS'
S' → ∨ TS'|ε
T → T ∧ F | F
F → ¬F | (S) | x | y

i=2 (T)
j=1 (S) : *règles de la forme T → Sα* : ***rien à faire***.
Élimination de la récursivité immédiate sur T : T → T ∧ F | F *deviennent* :
T → FT'
T' →∧ FT' | ε
S → TS'
S' → ∨ TS'|ε
F → ¬F | (S) | x | y

i=3 (F)
j=1 (S) : *rien à faire*
j=2 (T) : *rien à faire*

Nouvelle grammaire G4 :
T $\rightarrow$ FT′
T′ $\rightarrow\wedge$ FT′ | ε
S $\rightarrow$ TS’
S’ $\rightarrow\vee$ TS′|ε
F $\rightarrow\neg$F | (S) | x | y

5. Factorisation à gauche.

Factorisation de G2 :
$S' \rightarrow S|\varepsilon$
$S \rightarrow aAbS''|abS''|bBaSS''|baSS''|bBaS''|baS''$
$S'' \rightarrow aAbS''|abS''|\varepsilon$
$A \rightarrow bAa|ba$
$B \rightarrow aBb|ab$

Factorisation au niveau de S :
$S \rightarrow aC|bD$
$C \rightarrow AbS''|bS''$
$D \rightarrow BaSS''|aSS''|BaS''|aS''$
Fatorisation au niveau de D :
$D \rightarrow BaE|aF$
$E \rightarrow SS''|S''$
$F \rightarrow SS'|S''$
Nouvelles règles de S :
$S \rightarrow aC|bD$
$C \rightarrow AbS''|bS''$
$D \rightarrow BaE|aF$
$E \rightarrow SS''|S''$
$F \rightarrow SS'|S''$

Factorisation au niveau de S’’ :
$S'' \rightarrow aG|\varepsilon$
$G \rightarrow AbS''|bS''$

Factorisation au niveau de A :
$A \rightarrow bH$
$H \rightarrow Aa|a$

Factorisation au niveau de B :
$B \rightarrow aI$
$I \rightarrow Bb|b$

Nouvelles grammaire G2 :
$S \rightarrow aC|bD$
$C \rightarrow AbS''|bS''$

$D \rightarrow BaE|aF$
$E \rightarrow SS''|S''$
$F \rightarrow SS'|S''$
$S'' \rightarrow aG|\varepsilon$
$G \rightarrow AbS''|bS''$
$A \rightarrow bH$
$H \rightarrow Aa|a$
$B \rightarrow aI$
$I \rightarrow Bb|b$

Factorisation de G_3 :

S → TSS' | aS'
S' → TS' | SbS'|ε
T → aS'aT' | Tb |bT'
T' → SS'aT' | bT'|ε

La grammaire est déjà factorisée à gauche.

6. Donner la grammaire générant $L(G_2)^*$.

Grammaire G_2 :

$S \rightarrow SaAb|bBaS|\varepsilon$
$A \rightarrow bAa|\varepsilon$
$B \rightarrow aBb|\varepsilon$

Grammaire G_2* :

$S' \rightarrow SS'|\varepsilon$
$S \rightarrow SaAb|bBaS|\varepsilon$
$A \rightarrow bAa|\varepsilon$
$B \rightarrow aBb|\varepsilon$

7. Donner la grammaire générant $L(G_4) \cup L(G_3)$

Grammaire G_3 :

$S \rightarrow ST|SSb|TS|a$
$T \rightarrow Sa|Tb|\varepsilon$

Grammaire G_4:

S → S ∨ T | T
T → T ∧ F | F
F → ¬F | (S) | x | y

Grammaire $G_3 \cup G_4$:

Comme les deux grammaires ont les mêmes symboles non terminaux, il faut renommer les symboles S et T dans G_3. Ainsi, les productions de deviennent G_3 :

$S' \rightarrow S'T'|S'S'b|T'S'|a$
$T' \rightarrow S'a|T'b|\varepsilon$

Les productions de $G_3 \cup G_4$ sont :
$S'' \rightarrow S|S'$
$S' \rightarrow S'T'|S'S'b|T'S'|a$
$T' \rightarrow S'a|T'b|\varepsilon$
$S \rightarrow S \vee T \mid T$
$T \rightarrow T \wedge F \mid F$
$F \rightarrow \neg F \mid (S) \mid x \mid y$

8. Construire un automate à pile acceptant par état final pour la grammaire *G*₃, le transformer en un automate à pile acceptant par pile vide.

Cet automate a été corrigé en cours, il suffit de revoir le cours. D'une manière générale, l'automate contient toujours le même nombre d'état quel que soit la grammaire. Comme la transformation donne toujours un automate qui vide sa pile, pour le transformer en un automate qui accepte par pile vide, il suffit que l'automate n'ait aucun état final.