

## Contrôle

**Date : 12/01/2017**

**Durée : 1h30m**

### Exercice 01 (04 pts)

1. Donner l'ER définissant le langage composé de tous les mots non vides, sur l'alphabet  $\{0,1\}$ , qui ne contiennent pas la sous chaîne **11**.
2. Donner l'ER définissant le langage composé de tous les mots non vides sur  $\{a, b\}$  qui contiennent un nombre de  $a$  divisible par **3**.
3. Donner l'ER acceptant les nombres entiers dont la valeur est strictement inférieure à **167** et ne commençant pas par des **0** inutiles.

### Exercice 02 (06 pts)

Nous considérerons dans la suite qu'un tableau en html est défini comme suit:

- Le tableau est encadré par les balises `<TABLE>` et `</TABLE>`.
- Chaque ligne est encadrée par les balises `<TR>` et `</TR>`
- Les cellules de valeur sont encadrées par les balises `<TD>` et `</TD>`

Exemple : le code html ci-dessous

```
<TABLE>
<TR>
<TD> Valeur 11 </TD>
<TD> Valeur 12 </TD>
<TD> Valeur 13 </TD>
</TR>
<TR>
<TD> Valeur 11 </TD>
<TD> Valeur 12 </TD>
<TD> Valeur 13 </TD>
</TR>
</TABLE>
```

génère le tableau suivant :

```
Valeur 11 Valeur 12 Valeur 13
Valeur 21 Valeur 22 Valeur 23
```

Certaines parties du fichier peuvent être des commentaires. Les commentaires html sont encadrés par `<!--` et `-->`.

Ecrivez un programme **Lex** qui prend en entrée un programme HTML que l'on supposera bien structuré et qui pour chaque tableau, écrit la phrase :

*"le tableau commence à la ligne  $i$  et contient  $k$  lignes"*

Où  $i$  est le numéro de la ligne du code HTML où commence le tableau et  $k$  est le nombre de lignes dans ce tableau (pour l'exemple précédent,  $k = 2$ ). Cependant vous ne devrez pas comptabiliser les lignes en commentaires, ni produire un affichage pour des tableaux qui seraient entièrement en commentaires.

### Exercice 03 (05 pts)

Soit  $G$  la grammaire suivante :

$S \rightarrow X \mid Yc$

$X \rightarrow Xa \mid \varepsilon$

$Y \rightarrow Yb \mid d$

1. Cette grammaire n'est pas LL(1) : pourquoi ?
2. Quel est le langage décrit par cette grammaire ?
3. Donner une grammaire  $G1$  qui décrit le même langage. Justifier en utilisant la table d'analyse que  $G1$  est une grammaire LL(1).

### Exercice 04 (05 pts)

$S \rightarrow Ab|Bb$   
 $A \rightarrow Ca|Cae|\epsilon$   
 $C \rightarrow dA$   
 $B \rightarrow f|g$

Construire la table SLR de G. G est-elle SLR(1) ? Si oui Simulez-en le fonctionnement sur la chaîne **ddaeab\$**.  
Sinon justifiez rigoureusement en énumérant tous les conflits et en précisant leur types (shift/reduce ou reduce/reduce)

---

---

## Corrigé type

**Fait le : 12/01/2017**

**Durée : 1h30m**

---

---

### Exercice 01 (04 pts)

1.  $(0|1)(01|0)^*$
2.  $bb^*(b^*ab^*ab^*ab^*)(b^*ab^*ab^*ab^*)^*$
3.  $[0-9][1-9][0-9]|1[0-5][0-9]|16[0-6]$

### Exercice 0 2 (06 pts)

```
%{ int i = 0, nbl=0,k= 0;
%}
%%
" <table> " {i=nbl ;}
" <tr> " {k++ ;}
" <\table> " {printf("le tableau commence à la ligne %d et contient %d lignes",i,k); k=0;}
\<!\-\-\-(.|\n)*\-\-\>{}
"\n" {nbl++ ;}
. {}
%%
int main(int argc, char *argv[])
{
yyin = fopen(argv[1], "r");
yylex();
fclose(yyin);
}
```

### Exercice 0 3 (05 pts)

Soit G la grammaire suivante :

$$S \rightarrow X \mid Yc$$

$$X \rightarrow Xa \mid \varepsilon$$

$$Y \rightarrow Yb \mid d$$

1. Cette grammaire n'est pas LL(1) : Parce qu'il existe deux cas de récursivité à gauche : **(0,5 pt)**

$$X \rightarrow Xa \mid \varepsilon$$

$$Y \rightarrow Yb \mid d$$

2. Cette grammaire décrit le langage :  $a^*|db^*c$  **(01 pt)**

3. Une grammaire G1 qui décrit le même langage. **(01,5 pts)**

$$S \rightarrow X \mid Yc$$

$$X \rightarrow aX \mid \varepsilon$$

$$Y \rightarrow dY'$$

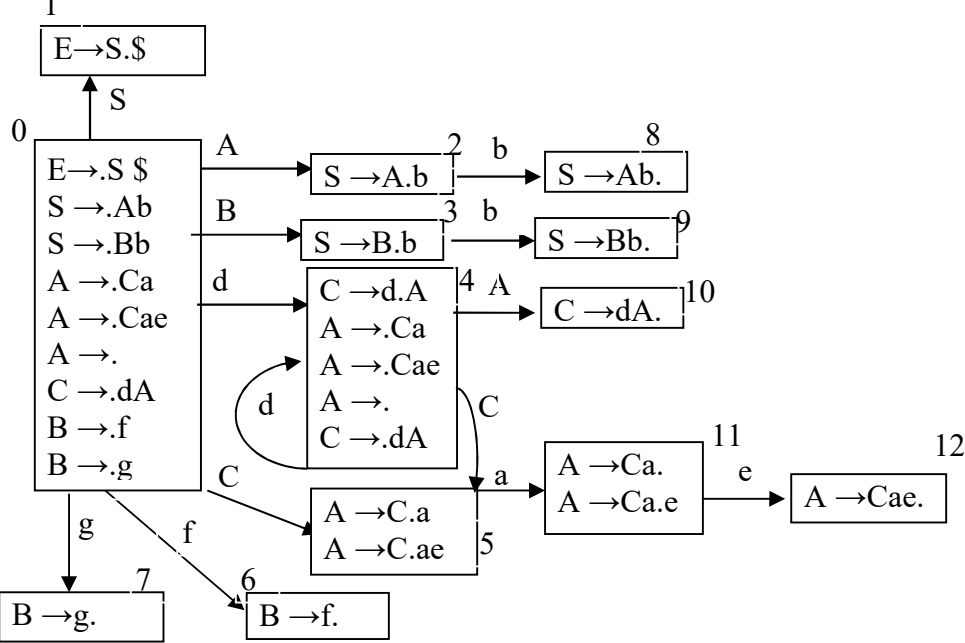
$$Y' \rightarrow bY' \mid \varepsilon$$

4. Justifier en utilisant la table d'analyse que G1 est une grammaire LL(1).

<b>(01 pt)</b>	Premiers	Suivants
S	a, d, $\varepsilon$	\$
X	a, $\varepsilon$	\$
Y	d	c
Y'	b, $\varepsilon$	c

<b>(01,5 pts)</b>	a	b	c	d	\$
S	$S \rightarrow X$			$S \rightarrow Yc$	$S \rightarrow X$
X	$X \rightarrow aX$				$X \rightarrow \varepsilon$
Y				$Y' \rightarrow dY'$	
Y'		$Y' \rightarrow bY'$	$Y' \rightarrow \varepsilon$		

Exercice 04 (05 pts) : Soit G la grammaire:  
Construction de l'automate LR pour G (1.5 pts).



(1pts).	Premiers	Suivants
S	b,d,f,g	\$
A	d, ε	b,a
B	f,g	b
C	d	a

SLR(1) ? oui il n’y a aucun cas de conflit (1,5 pts)

	a	b	d	e	f	g	\$		S	A	B	C
0	r5	r5	d4		d6	d7			1	2	3	5
1							Acc					
2		d8										
3		d9										
4	r5	r5	d4							10		5
5	d11											
6		r7										
7		r7										
8							r1					
9							r2					
10	r6											
11	r3	r3		d12								
12	r4	r4										

(1pt) Pile	Chaine	Action	Pile	Chaine	Action
\$0	ddaeab\$	d4	\$0C5a11	b\$	r3
\$0d4	daeab\$	d4	\$0A	b\$	2
\$0d4d4	aeab\$	r5	\$0A2	b\$	d8
\$0d4d4A	aeab\$	10	\$0A2b8	\$	r1
\$0d4 d4A10	aeab\$	r6	\$0S	\$	1
\$0 d4C	aeab\$	5	\$0S1	\$	Acc
\$0 d4C5	aeab\$	d11			
\$0 d4C5a11	eab\$	d12			
\$0 d4C5a11e12	ab\$	r4			
\$0 d4A	ab\$	10			
\$0 d4A10	ab\$	r6			
\$0C	ab\$	5			
\$0C5	ab\$	d11			

Université Mohamed Khider – Biskra.  
Faculté des Sciences Exactes et des  
Sciences de la Nature et de la Vie  
Département d'Informatique

Module: **Compilation**  
Niveau : **3<sup>ème</sup> Licence**  
Année universitaire: **2016/2017**  
**Documents non autorisés**

## Rattrapage

**Date : 04/03/2017**

**Durée : 1h30m**

### Exercice 01 (06 pts) :

- 1) Donner une expression régulière qui valide une forme simplifiée des adresses électroniques.

Définition simplifiée :

- une adresse électronique est constituée d'un champ ou plusieurs suivies d'un @ suivi d'un champ ou plusieurs.
- un champ est constitué d'un caractère ou plusieurs (lettre, chiffre, -, \_).

- 2) Un nombre binaire est constitué de 0 et de 1

A. Donner une expression régulière qui accepte les nombres binaires contenant au moins deux 0.

B. Donner une expression régulière qui accepte les nombres binaires contenant au plus trois 0.

- 3) Donner une expression régulière sur l'alphabet {a, b}, qui accepte tous les mots qui ne contiennent pas le facteur *bab* ;

### Exercice 02 (10 pts) :

Soit G la grammaire suivante :

$A \rightarrow da|acB$

$B \rightarrow abB|daA|Af$

1. G est-elle LL(1) ? justifier votre réponse. Sinon la transformer pour qu'elle le soit (grammaire G').
2. Construire la table d'analyse LL(1).
3. Construire l'automate LR pour G.
4. La grammaire G est-elle SLR(1) ? Justifier.

### Exercice 03 (04 pts) :

- 1) Ecrire un programme en lex qui supprime une suite d'espaces et les remplace par un seul espace.
- 2) Ecrire un programme en Flex qui compte le nombre de caractères, de mots et de lignes d'un texte en entrée.



## Corrigé type du Rattrapage

Fait le : 04/03/2014

Durée : 1h30m

### Exercice 01 (06 pts) :

1.  $([a-zA-Z0-9][\_\backslash])+@([a-zA-Z0-9][\_\backslash])+$
2.  $(0|1)^*0(0|1)^*0(0|1)^*$
3.  $1(0|\epsilon)1^*(0|\epsilon)1^*(0|\epsilon)1^*|01^*(0|\epsilon)1^*(0|\epsilon)1^*$
4.  $a^*(b^*((aa)a^*)^*)(a|\epsilon)$

### Exercice 03 (04 pts) :

- 1) Ecrire un programme en lex qui supprime une suite d'espaces et les remplace par un seul espace.

```
()* printf(" ");
%%
int main(int argc, char *argv[])
{yyin = fopen(argv[1], "r");
 yylex();
 fclose(yyin);}
```

- 2) Ecrire un programme en Flex qui compte le nombre de caractères, de mots et de lignes d'un texte % { int  
nbm = 0, nbl = 0, nbc = 0;

```
%}
mot [a-zA-Z]+
%%
{mot} nbm++; nbc+=yyleng;
\n nbl++; nbc++;
. nbc++;
%%
main()
{ yylex();
```

```
printf("Il y a %d voyelles, %d consonnes et %d punctuations.\n",nbm, nbc, nbl); }
```

### Exercice 02 (10 pts) :

Soit G la grammaire suivante :

$A \rightarrow da|acB$

$B \rightarrow abB|daA|Af$

G est-elle LL(1) ? justifier votre réponse. Non, il existe une factorisation indirecte dans la règle :

$B \rightarrow abB|daA|Af$  dont Premiers (A)  $\cap$  Premiers (B)  $\neq \emptyset$  **0.5pt**

Sinon la transformer pour qu'elle le soit (grammaire G').

$A \rightarrow da|acB$

$B \rightarrow abB|daA|(da|acB)f$

$B \rightarrow abB|daA|daf|acBf$

$B \rightarrow abB|acBf|daA|daf$

$B \rightarrow a(bB|cBf)|da(A|f)$

$B \rightarrow aB'|daB''$

$B' \rightarrow (bB|cBf)$

$B'' \rightarrow (A|f)$

**G' 1.5pts :**

$A \rightarrow da|acB$

$B \rightarrow aB'|daB''$

$B' \rightarrow bB|cBf$

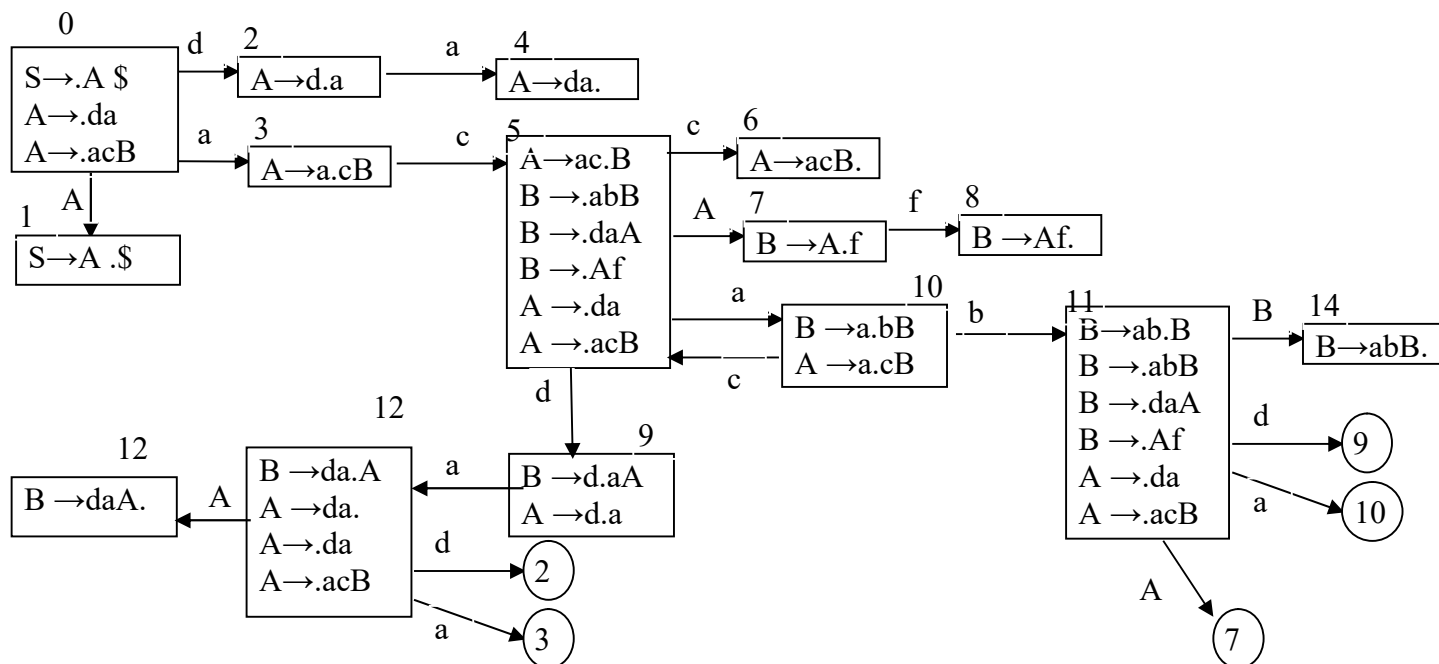
$B'' \rightarrow A|f$

1. Construire la table d'analyse LL(1).

01pt	Premiers	Suivants
A	d,a	\$,f
B	d,a	\$,f
B'	b,c	\$,f
B''	d,a,f	\$,f

2pts	a	b	c	d	f	\$
A	acB			da		
B	aB'			daB''		
B'		bB	cBf			
B''	A			A	f	

2. Construire l'automate LR pour G. (02.5 pts)



3. La grammaire G est-elle LR(0) ? Justifier.

SLR(1)	a	b	c	d	f	\$	(02.5 pts)	A	B
0	d3			d2				1	
1						Acc			
2	d4								
3			d5						
4					r1	r1			
5	d10							6	7
6					r2	r2			
7					d8				
8					r5	r5			
9	d12								
10		d11	d5						
11	d10			d9				7	14
12	d3			d2	r1	r1		13	
13					r4	r4			
14					r3	r3			

	Premiers	Suivants
A	a,d	\$,f
B	a,d	\$,f