

LICENCE Sciences et Technologie – 3^{ème} année
Contrôle terminal
EI62 - Langages et Compilation
Durée de l'épreuve : 2 heures

Documents autorisés: Polycopié et notes personnelles
Le barème est indicatif.

A. Description de langages (6 points)

On s'intéresse dans cet exercice au langage L sur l'alphabet $\{0,1\}$ constitué des mots **ne contenant pas** la séquence 0 1 1.

Par exemple 0 0 0 1 0 0 1 0 est dans L mais pas 0 0 0 1 0 1 1 0 1 0.

- 1) Ecrire un automate A1 reconnaissant L. Votre automate est-il déterministe ou non déterministe ? est-il complet ? justifiez votre réponse.
- 2) En déduire un automate A2 reconnaissant les mots **contenant** la séquence 0 1 1. **Indication impérative** : vous décrierez le **procédé** permettant de **passer de A1 à A2**.
- 3) Ecrire une grammaire engendrant L. Quelle propriété caractéristique possède votre grammaire ?
- 4) Transformez cette grammaire en grammaire attribuée calculant le nombre de 1 (format « théorique » du cours ou ANTLR).

B. ANALYSE SYNTAXIQUE (8 points)

B1. Analyse ascendante LR

On donne la grammaire G1 :

$S \rightarrow a S c \mid B$

$B \rightarrow b B \mid c$

Sur le modèle vu en cours, appliquez l'analyse LR (non déterministe, en choisissant la bonne règle) sur la chaîne : $w = abbcc$.

B2. Analyse LL

Calculez la fonction PREMIER pour G1, ainsi que la table d'analyse. G1 est-elle LL(1) ? justifiez votre réponse. Pourquoi n'a-t-on pas besoin de la fonction SUIVANT ?

Sur le modèle vu en cours, appliquez l'analyse LL sur la chaîne : $w = abbcc$.

B3. Analyse LL

Soit la grammaire G2 :

$S \rightarrow a S c \mid B$

$B \rightarrow b B \mid b$

G2 a-t-elle une chance d'être LL(1) ? pourquoi ?

Opérez une transformation susceptible de la rendre LL(1). Soit G3 la nouvelle grammaire. Calculez la table d'analyse LL(1). Appliquez l'analyse LL sur la chaîne : w = abbcc.

C. COMPILATION (6 points)

Expliquez le principe général de la compilation. Vous préciserez notamment les notions d'analyse lexicale, d'analyse syntaxique, et le principe de la génération de code.

A titre d'exemple, vous donnerez les arbres d'analyse et de syntaxe abstraite pour l'instruction suivante :

if (x<0) y = 0 - x else y = x

Pour le fragment de grammaire suivant (variante simplifiée de celle du cours, en syntaxe « théorique ») :

instruction →

IDENT '=' expression

| 'if' '(' condition ')' instruction 'else' instruction

| ...

expression → ((ENTIER | IDENT) ('+' | '-') expression) | ENTIER | IDENT

condition → expression ('<' | '>' | '==') expression

CORRIGE

A. Automate A1 (AFD non complet)

	A	B	C
0	B	B	B
1	A	C	-

Init = {A} Fin = {A,B,C}

A2 : on complete A1 par un puits, qui devient le seul état acceptant.

Grammaire (régulière) :

$A \rightarrow 0 B \mid 1 A \mid \text{eps}$

$B \rightarrow 0 B \mid 1 C \mid \text{eps}$

$C \rightarrow 0 B \mid \text{eps}$

Attribuée :

$A \rightarrow 0 B$

$A.N := B1.N$

$A \rightarrow 1 A$

$A.N = A1.N + 1$

$A \rightarrow \text{eps}$

$A.N = 0$

$B \rightarrow 0 B$

$B.N = B1.N$

$B \rightarrow 1 C$

$B.N = C1.N + 1$

$B \rightarrow \text{eps}$

$B.N = 0$

$C \rightarrow 0 B$

$C.N = B1.N$

$C \rightarrow \text{eps}$

$C.N = 0$

B.

B1. G1 :

$S \rightarrow a S c \mid B$

$B \rightarrow b B \mid c$

Sur le modèle vu en cours, appliquez l'analyse LR (non déterministe, en choisissant la bonne règle) à la chaîne : $w = \text{abbcc}$.

\$	abbcc\$	Décaler
\$a	bbcc\$	D
\$ab	bcc\$	D
\$abb	cc\$	D
\$abb <u>c</u>	c\$	R ($B \rightarrow c$)
\$abb <u>B</u>	c\$	R ($B \rightarrow bB$)
\$ab <u>B</u>	c\$	R ($B \rightarrow bB$)
\$a <u>B</u>	c\$	R ($S \rightarrow B$)
\$a <u>S</u>	c\$	D
\$a <u>Sc</u>	\$	R ($S \rightarrow aSc$)
\$ <u>S</u>	\$	SUCCES

B2. Analyse LL

Calculez les fonctions PREMIER et SUIVANT pour G1, ainsi que la table d'analyse. G1 est-elle LL(1) ? justifiez votre réponse.

Sur le modèle vu en cours, appliquez l'analyse LL à la chaîne : $w = abbcc$.

PREMIER

S \rightarrow a b c

B \rightarrow b c

Table d'analyse

Grammar
 $S \rightarrow a S c$
 $\mid B.$
 $B \rightarrow b B$
 $\mid c.$

	\$	c	b	a
S		$S \rightarrow B$	$S \rightarrow B$	$S \rightarrow a S c$
B		$B \rightarrow c$	$B \rightarrow b B$	

Return home to [enter a new grammar.](#)

S\$	abbcc\$	$S \rightarrow aSc$
aSc\$	abbcc\$	D
Sc\$	bbcc\$	$S \rightarrow B$
Bc\$	bbcc\$	$B \rightarrow bB$
bBc\$	bbcc\$	D
Bc\$	bcc\$	$B \rightarrow bB$
bBc\$	bcc\$	D
Bc\$	cc\$	$B \rightarrow c$
cc\$	cc\$	D
c\$	c\$	D
\$	\$	SUCCES

B3. Analyse LL

Soit la grammaire G2 :

$S \rightarrow a S c \mid B$

$B \rightarrow b B \mid b$

G2 a-t-elle une chance d'être LL(1) ? pourquoi ?

Non : non déterminisme des règles su B en présence d'un b.

Factorisation

$S \rightarrow a S c \mid B$

$B \rightarrow b C$

$C \rightarrow B \mid \text{eps}$

	PREM	SUIV
S	a b	\$ c
B	b	\$ c
C	b	\$

c

Grammar
 $S \rightarrow a S c$
 $\quad | B.$
 $B \rightarrow b C.$
 $C \rightarrow B$
 $\quad | .$

	\$	b	c	a
S		$S \rightarrow B$		$S \rightarrow a S c$
B		$B \rightarrow b C$		
C	$C \rightarrow \epsilon$	$C \rightarrow B$	$C \rightarrow \epsilon$	

[Return home to enter a new grammar.](#)

S\$	abbcc\$	$S \rightarrow aSc$
aSc\$	abbcc\$	D
Sc\$	bbcc\$	$S \rightarrow B$
Bc\$	bbcc\$	$B \rightarrow bC$
bCc\$	bbcc\$	D
Cc\$	bcc\$	$C \rightarrow B$
Bc\$	bcc\$	$B \rightarrow bC$
bCc\$	bcc\$	D
Cc\$	cc\$	$C \rightarrow \epsilon$
c\$	cc\$	D
\$	c\$	ECHEC : expected end of file, found 'c'

C. if ($x < 0$) $y = 0 - x$ else $y = x$

instruction \rightarrow

IDENT '=' expression

| 'if' '(' condition ')' instruction 'else' instruction

| ...

expression \rightarrow ((ENTIER | IDENT) ('+' | '-') expression) | ENTIER | IDENT

condition \rightarrow expression ('<' | '>' | '==') expression

Arbre d'analyse :

instruction('if',

```

('(', condition(expression(IDENT('x')), '<', expression(ENTIER('0'))), '(',
instruction(IDENT('y'), '=', expression(ENTIER('0')), '- ', expression(IDENT('x')))),
instruction(IDENT('y'), '=', expression(IDENT('x')))) )

```

AST : quelque chose comme : $\text{if}(<(x,0), =(y, -(0,x)) , =(y,x))$