

L3 – INF6ACT
Théorie des langages et compilation
durée 2h

Documents autorisés : notes personnelles, diapos du cours.

Chaque candidat doit, en début d'épreuve, porter son nom dans le coin de la copie réservé à cet usage; il le cachettera par collage après la signature de la feuille d'émargement. Sur chacune des copies intercalaires, il portera son numéro de place.

Rendre 2 copies séparées en notant bien le numéro de place :

- l'une qui traite l'exercice I (Automate fini) et l'exercice III (analyse LL)
- l'autre qui traite l'exercice II (Compilation)

Exercice I. Automate Fini

À rendre avec l'exercice III

On considère le langage L sur l'alphabet $\{a, b\}$ décrit par l'expression régulière

$$(aa + b)^*ab(bb)^*$$

Question 1. Donner les quatre mots de longueur 5 qui appartiennent au langage L.

Question 2. Donner trois mots de longueur 4 qui ne sont pas dans le langage L.

Question 3. Donner un automate fini (déterministe ou non) qui reconnaît le langage L.

Question 4. Donner une grammaire qui engendre le langage L.

Exercice II. Compilation

À rendre sur une copie séparée

On souhaite ajouter à notre calculette le support basique d'un type `rational` correspondant aux nombres rationnels. On les construira à l'aide de la syntaxe `(p,q)` pour désigner le nombre $\frac{p}{q}$. Ces nombres seront stockés dans la pile en utilisant deux entiers de la MVàP (un pour `p` et un pour `q`, `p` étant empilé avant `q`). On suppose l'existence des opérations `RMUL` qui multiplie deux rationnels et de `WRITER` qui affiche un rationnel.

Question 5. Donner le code MVàP pour le programme élémentaire suivant:

```
rational z = (42,110)
```

Question 6. Donner le résultat des opérations suivantes (on utilise les opérations usuelles sur les rationnels):

- `write((4,5)*(6,4)) ;`
- `write((1,2)+(1,3)) ;`
- `if ((1,3) < (1,2)) write(3) else write(2).`

Soit le code suivant (**Code MVàP EX1**) et le résultat de son assemblage

	Adr		Instruction	
PUSHI 7				
PUSHI 3				
JUMP Main	0		PUSHI	7
LABEL f	2		PUSHI	3
PUSHL -3	4		JUMP	35
PUSHI 4	6		PUSHL	-3
INF	8		PUSHI	4
JUMPF B1	10		INF	
PUSHI 0	11		JUMPF	26
PUSHL -3	13		PUSHI	0
PUSHI 2	15		PUSHL	-3
MUL	17		PUSHI	2
CALL f	19		MUL	
POP	20		CALL	6
STOREL -4	22		POP	
RETURN	23		STOREL	-4
LABEL B1	25		RETURN	
PUSHL -3	26		PUSHL	-3
PUSHI -1	28		PUSHI	-1
ADD	30		ADD	
STOREL -4	31		STOREL	-4
RETURN	33		RETURN	
RETURN	34		RETURN	
LABEL Main	35		PUSHI	0
PUSHI 0	37		PUSHI	3
PUSHI 3	39		CALL	6
CALL f	41		POP	
POP	42		PUSHI	1
PUSHI 1	44		PUSHG	0
PUSHG 0	46		PUSHG	1
PUSHG 1	48		RMUL	
RMUL	49		STOREG	1
STOREG 1	51		STOREG	0
STOREG 0	53		PUSHG	0
PUSHG 0	55		PUSHG	1
PUSHG 1	57		WRITER	
WRITER	58		HALT	
HALT				

Question 7. Compléter la trace d'exécution suivante.

pc			fp	pile
0	PUSHI	7	0	[] 0
2	PUSHI	3	0	[7] 1
4	JUMP	35	0	[7 3] 2
35	PUSHI	0	0	[7 3] 2
37	PUSHI	3	0	[7 3 0] 3
39	CALL	6	0	[7 3 0 3] 4
6	PUSHL	-3	6	[7 3 0 3 41 0] 6
8	PUSHI	4	6	[7 3 0 3 41 0 3] 7
10	INF		6	[7 3 0 3 41 0 3 4] 8
11	JUMPF	26	6	[7 3 0 3 41 0 1] 7
13	PUSHI	0	6	[7 3 0 3 41 0] 6
15	PUSHL	-3	6	[7 3 0 3 41 0 0] 7
...
...
22	POP		6	[7 3 0 3 41 0 5 6] 8
23	STOREL	-4	6	[7 3 0 3 41 0 5] 7
25	RETURN		6	[7 3 5 3 41 0] 6
41	POP		0	[7 3 5 3] 4
42	PUSHI	1	0	[7 3 5] 3
44	PUSHG	0	0	[7 3 5 1] 4
46	PUSHG	1	0	[7 3 5 1 7] 5
48	RMUL		0	[7 3 5 1 7 3] 6
49	STOREG	1	0	[7 3 35 3] 4
51	STOREG	0	0	[7 3 35] 3
53	PUSHG	0	0	[35 3] 2
55	PUSHG	1	0	[35 3 35] 3
57	WRITER		0	[35 3 35 3] 4
35	/ 3			
58	HALT		0	[35 3 35 3] 4

Question 8. Donner le code MVàP correspondant au programme suivant:

```
rational z = (4,5)
z = (6,5) * z
write(z)
```

On donne la grammaire suivante des expressions incluant un constructeur pour les rationnels :

```
expression returns [String code]
: ...
| '(' expression ',' expression ')' { // À Compléter }
: ...
;
```

Question 9. Indiquer sur votre copie les éléments à remplir dans la partie à compléter pour le constructeur.

Question 10. Montrer qu'il est possible de simuler le comportement de RMUL à l'aide des opérations de la MVàP sur les entiers. On supposera au besoin que l'adresse 0 est réservée et que l'on peut y stocker une valeur de façon temporaire.

Question 11. Indiquer quel programme calculette permet de générer le code **Code MVàP EX1** donné au début de l'exercice.

Exercice III. Analyse LL

À rendre avec l'exercice I

On considère la grammaire \mathcal{G} d'axiome E et de terminaux $\{ + , * , \text{nb} , (,) \}$ définie par :

$$\left\{ \begin{array}{l} E \rightarrow \text{nb} \mid (\text{OP E L}) \\ \text{OP} \rightarrow + \mid * \\ L \rightarrow \text{E X} \\ X \rightarrow L \mid \varepsilon \end{array} \right.$$

Question 12. Donner un arbre d'analyse pour le mot $(* \text{nb} \text{nb} \text{nb})$

Question 13. Déterminer les variables effaçables.

On dispose des tables Premier et Suivant de la grammaire \mathcal{G} :

Symbole	Premier	Suivant
E	nb , (\$, (, nb ,)
OP	+ , *	nb , (
L	nb , ()
X	nb , ()

Question 14. Indiquer de façon claire et précise :

14.a) comment l'ensemble Premier(E) est déterminé ?

14.b) comment l'ensemble Premier(L) est déterminé ?

14.c) comment l'ensemble Suivant(E) est obtenu ?

14.d) comment l'ensemble Suivant(X) est obtenu ?

Question 15.

15.a) Construire la table d'analyse LL(1) de la grammaire \mathcal{G} .

15.b) Pourquoi \mathcal{G} est-elle LL(1) ?

Question 16.

16.a) À l'aide de cette table, dérouler l'analyse LL(1) sur l'entrée

$(* \text{nb})$

16.b) De même, dérouler l'analyse LL(1) sur l'entrée

$(* \text{nb} \text{nb} \text{nb})$