Département de Mathématiques, Informatique & Gestion
Sciences Mathématiques et Informatique (SMI/S5)
Compilation
Prof. : M. BENADDY
A.U:2019/2020

Corrigé de l’xamen de la session principale

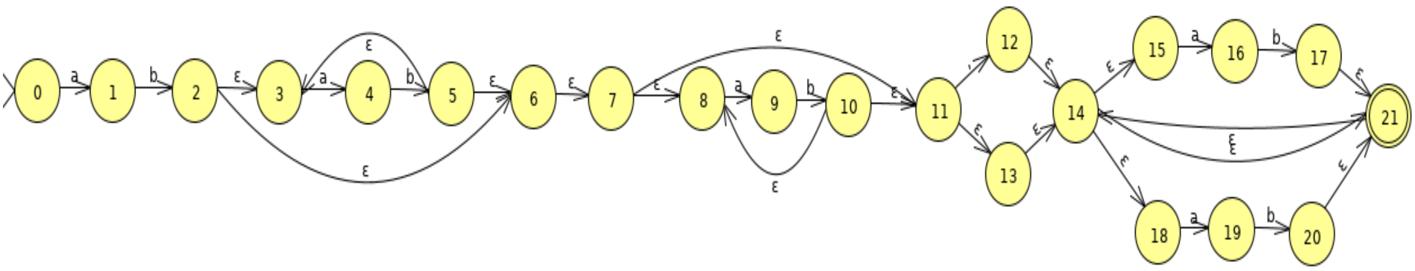
durée : 1h30

NB : Le barème est donné à titre indicatif

Exercice 1 (6pts) :

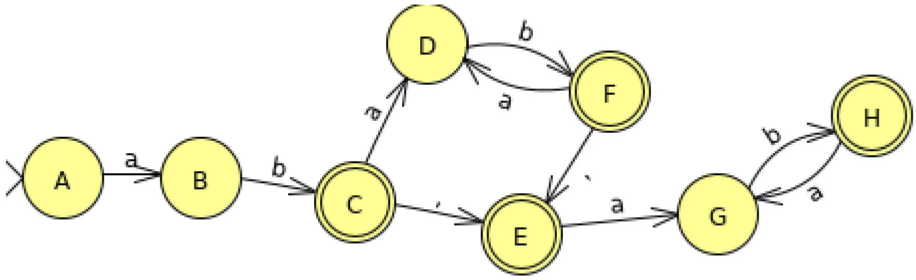
Soit l'expression régulière suivante :
((ab)+(ab)*),?(ab|ab)*

1. Quel est le langage dénoté par cette expression
   Ensemble des mots constitués de suite de ab.
2. Donner l'AFN correspondant à cette expression.



3. Transformer l'AFN de la question 2 en un AFD.

A=ε-f({0})={0}
Ɛ-f({T(A, a)})=Ɛ-f({1})={1}=B => Dtrans(A,a) = B
Ɛ-f({A, b}) = {Ø}
Ɛ-f({T(A, ,)}) = {Ø}
Ɛ-f({T(B, a)}) = {Ø}
Ɛ-f({T(B,b)}) = Ɛ-f({2})= {2,3,6,7,8,11,13,14,15,18,21} = C => Dtrans(B,b) = C
Ɛ-f({T(B, ,)}) = {Ø}
Ɛ-f({T(C,a)}) = Ɛ-f({4,9,16, 19})= {4,9,16,19} = D => Dtrans(C,a) = D
Ɛ-f({T(C, ,)}) = Ɛ-f({12})= {12,14,15,18,21} = E => Dtrans(C, ,) = E
Ɛ-f({T(C, b)}) = {Ø}
Ɛ-f({T(D, b)}) = Ɛ-f({5,10,17,20})= {3,5,6,7,8,10,11,13,14,15,17,18,20, 21} = F => Dtrans(D,b) = F
Ɛ-f({T(D, a)}) = {Ø}
Ɛ-f({T(D, ,)}) = {Ø}
Ɛ-f({T(E, a)}) = Ɛ-f({16, 19}) = {16, 19} = G => Dtrans(E,a) = G
Ɛ-f({T(E, b)}) = {Ø}
Ɛ-f({T(E, ,)}) = {Ø}
Ɛ-f({T(F, a)}) = Ɛ-f({4, 9, 16, 19}) = D => Dtrans(F, a) = D
Ɛ-f({T(F, b)}) = {Ø}
Ɛ-f({T(F, ,)}) = Ɛ-f({12}) = E => Dtrans(F, ,) = E
Ɛ-f({T(G, a)}) = {Ø}
Ɛ-f({T(G, b)}) = Ɛ-f({17, 20}) = {17, 21, 14, 15, 18, 20} = H => Dtrans(G, b) = H
Ɛ-f({T(G, ,)}) = {Ø}
Ɛ-f({T(H, a)}) = Ɛ-f({16, 19}) = G => Dtrans(H, a) = G
Ɛ-f({T(H, b)}) = {Ø}
Ɛ-f({T(H, ,)}) = {Ø}



4. Minimiser l'AFD résultant de la question 3 (donner la table des transitions et le graphe des états de l'AFD minimisé).
   Les états d'acceptation {C,E, F, H} et les états de non-acceptation {A,B,D,G}

   π={C,E, F, H} {A,B,D,G}
   On prend le groupe {C,E, F, H}
   sur le symbole a : T(C, a) = T(F, a) = {D} et T(E, a) = T (H, a) = {G} et sur les symboles b et , pas de distinction

   alors, π={C, F} {E, H} {A,B,D,G}
   on prend le groupe {C,F} sur le symbole a : T(C,a) = T(F,a) = {D} donc pas de distinction. Sur le symbole b : T(C, b)=T(F, b) = {Ø} donc pas de distinction. Sur le symboles . : T(C, ,)=T(F, ,) = {E} donc pas de distinction.

   π={C,F} {E, H} {A,B,D,G}
   on prend {E, H} sur le symbole a : T(E, a) = T(H, a)= {G} donc pas de distinction sur b : T(E, b) = T(H, b)= {Ø} donc pas de distinction. Sur le symbole , : y a pas de transition

   alors, π={C,F} {E, H} {A,B,D,G}
   on prend le groupe {A,B,D,G}
   sur le symbole a : T(A, a) = {B} $\subset$ {A,B,D,G}. Sur le symbole b : T(B,b) ={C} $\not\subseteq$ {A,B,D,G}, T(D,b) ={F} $\not\subseteq$ {A,B,D,G}, T(G, b) ={H} $\not\subseteq$ {A,B,D,G}. Sur le symbole, Pas de distinction et puisque sur le symbole b pas de distinction (car toutes les transitions quittent vers le même groupe {C,F} et seul l'état G quitte le groupe vers le groupe {E, H} alors.

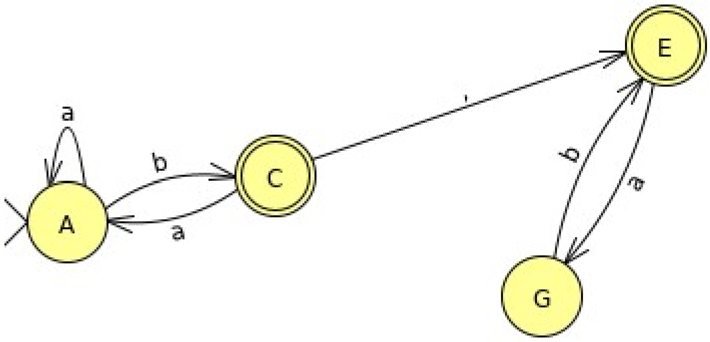
   alors, π={C,F} {E, H} {A,B,D} {G}
   on prend le groupe {A,B,D} sur le symboles symboles a,b, , pas de distinction donc

   π_f={C,F} {E, H} {A,B,D} {G}
   on prend {C} représentant de {C,F}, {E} représentant de {E, H} et {A} représentant de {A,B,D}

   π_f={C} {E} {A} {G}
   La table de transition de l'AFD minimal :

	a	**b**	**,**
A	A	C	
C	A		E
G		E	
E	G		

   Le graphe de transition :



Exercice 2 (12pts):

Soit la grammaire G avec les règles de production suivantes :

I → (E) | C | T
E → F
F → L | ε
C → id (A)
A → L | ε
L → L,E | E
T → n | t | f

1. Déterminer les non-terminaux et les terminaux de la grammaire G.
   N = {I, E, C, T, F, L, A} et T= {(,), id, , n, t, f}
2. Donner la table d'analyse prédictive de G.

	(	**)**	**,**	**f**	**id**	**n**	**t**	**$**
I	I→(E)			I→ T	I→ C	I→ T	I→ T	
E		E→ F	E→ F					
F		F→L|Ɛ	F→L|Ɛ					
C					C→ id (A)			
A		A→ L | Ɛ	A → L					
L		L→ E	L→L,E|E					
T				T→ f		T→ n	T→ t	

3. La grammaire G est elle LL(1), justifier ?
   Non car il y'a des entrées multiples dans la table d'analyse et la grammaire est récursive.
4. Calculer Début et Suivant pour les non terminaux de G.

	Premier	**Suivant**
I	{t, f, n, (, id}	{ $ }
E	{Ɛ, , }	{), , }
F	{Ɛ, , }	{), , }
C	{ id }	{ $ }
A	{Ɛ, , }	{) }
L	{Ɛ, , }	{), , }
T	{t, f, n}	{ $ }

5. Donner l'automate des items LR(0) canoniques pour G.

Etats	{items LR(0)}
fermeture({I' → I})=I0	**{I' → •I, I → •(E), I → •C, I → •T, C→ •id(A), T→ •n, T→ •t, T→ •f }**
Transition(I0,))= I1	**{F → •, L → •E, F → •L, L → •L,E, E→ •F, I→ (•E)}**
Transition(I0,C)= I2	**{I → C•}**
Transition(I0,I)= I3	**{I' → I•}**
Transition(I0,T)= I4	**{I → T•}**
Transition(I0,f)= I5	**{I → f•}**
Transition(I0,id)= I6	**{I → id•(A)}**
Transition(I0,n)= I7	**{T → n•}**
Transition(I0,t)= I8	**{T → t•}**
Transition(I1,E)= I9	**{I→ (E•), L→ E•}**
Transition(I1,F)= I10	**{E→ F•}**
Transition(I1,L)= I11	**{F → L•, L → L•,E}**
Transition(I6,()= I12	**{I → id(•A), F→ •, L → •E, F → •L, A→ •L, A→ •, L→ •L,E, E→ •F }**
Transition(I9,))= I13	**{I→ (E)•}**
Transition(I11,,)= I14	**{L → L,•E, F → •, L →•E, F →•L, L →•L,E, E →•F}**
Transition(I12,A)= I15	**{I → id(A•) }**
Transition(I12,E)= I16	**{ L → E• }**
Transition(I12,L)= I17	**{F → L•, A→ L•, L→ L•,E }**
Transition(I12,F)= I10	**{E→ F•}**
Transition(I14,E)= I18	**{L → L,E•, L →E•}**
Transition(I14,F)= I10	**{E→ F•}**
Transition(I14,L)= I11	**{F → L•, L → L•,E}**
Transition(I15,))= I19	**{I → id(A)•}**
Transition(I17,,)= I14	**{L → L,•E, F → •, L →•E, F →•L, L →•L,E, E →•F}**

6. Donner la table des actions et successeurs SLR de G.

	(	**)**	**,**	**f**	**id**	**n**	**t**	**$**	**A**	**C**	**E**	**F**	**I**	**L**	**T**
0	d1			d5	d6	d7	d8			2			3		4
1		r6	r6								9	10		11	
2								r2							
3								acc							
4								r3							

5								r14							
6	d12														
7								r12							
8								r13							
9		r11/ d13	r11												
10		r4	r4												
11		r5	r5/ d14												
12		r6/r9	r6						15		16	10		17	
13								r1							
14		r6	r6								18	10		11	
15		d19													
16		r11	r11												
17		r5/r8	r5/ d14												
18		r10/ r11	r10/ r11												
19								r7							

7. La grammaire G est-elle SLR ? Non car, il y'a des conflits dans les actions de la table d'analyse SLR0

Exercice 3 (2pts):

Ecrire un programme en Flex qui prend en paramétre un fichier et permet de reconnaître les noms de personnes ayant le format suivant : **N. PRENOM**

Exemple :
fichier ratexo3.l

```
%{
#include<stdio.h>
%}
nom [A-Z]
prenom [A-Z]+
%%
{nom}.{prenom} {printf("%s\n",yytext);}
.|\n {}
%%
int main (int argc, char *argv[]){
FILE *f;
if(argc>1){
	f=fopen(argv[1] , "r");
	if(!f){
		printf("fichier n est existe pas\n") ;
		exit(1) ;
}
```

```
else{
        yyin=f;
        yylex();
        fclose(f) ;
}       }
        return 0 ;
}
```

%%%%%%%%%%%%%%%%%%%%%%%
Exemple :

Compilation par Flex: Compilation gcc : Test avec le fichier test.txt	mohamed@benaddy-lap:~/cours/smis5_compile/examens/2019-2020$ flex ratexo3.l mohamed@benaddy-lap:~/cours/smis5_compile/examens/2019-2020$ gcc lex.yy.c mohamed@benaddy-lap:~/cours/smis5_compile/examens/2019-2020$ ./a.out test.txt B.MOHAMED M.AHMED A.ALI S.FARAH

Le fichier test.txt contient:
monsieur B.MOHAMED
monsieur M.AHMED
monsieur A.ALI
monsieur S.FARAH