Université Sultan Moulay Slimane

Faculté Polydisciplinaire de Khouribga

Examen (module: Compilation)

SMI S5 – 2019 – 2020

Durée: 1h 30min

Questions de cours:

1. Citer le rôle principale d'un analyseur syntaxique.
2. L'opérationnalisation des analyseurs syntaxiques repose sur les grammaires, donner la définition formelle d'une grammaire.

Exercice 1

Soient l'alphabet ∑={a,b,c} et la table de transition suivante :

Δ	**a**	**b**	**c**
0	1		
1			1,2
2		2	

e_0={0} T={2}

1. Les mots ac, acc, accbc , acccccbbb et acccbbbc appartiennent –ils au langage reconnu par l'automate
2. Donner l'automate correspondant et déterminiser le .
3. Proposer une expression régulière qui correspond au langage reconnu par l'automate précédent.

Exercice 2 :

On considère la syntaxe suivante:

Syntaxe	**Exemple1**	**Exemple2**	**Exemple3**	**Exemple4**
Id = valeur Id = Id + Id Id = Id + valeur	Code a=10	Code a = b + c	Code a=b+20	Code a=b*c+d+e

Partie I :

On propose la grammaire suivante:

S = id = T E
T = id | valeur
E = + T E | *T E |ε

1. Donner l'ensemble V_N et l'ensemble V_T.
2. Donner les premiers et les suivants de chaque symbole non terminal.
3. Élaborer la table d'analyse LL de cette grammaire.
4. Analyser la phrase **x = y + 4 + z** et Donnez l'arbre de dérivation associé. A noter que x,y et z sont des identificateurs.

Partie II :

Cette partie traitera une grammaire permettant de produire la syntaxe d'une fonction sous Python.

On considère la syntaxe suivante:

Syntaxe	**Exemple1**	**Exemple2**	**Exemple3**
def fonction(liste_paramètres): instr où instr présente l'instruction de la partie I	Code	Code	Code
	def fct1(x): x=x+x*x	def fct2(x,y): x=x+2*y	def fct1(x,y,z): x=x+2*y+z
	Résultat de l'exécution	Résultat de l'exécution	Résultat de l'exécution

	>>> x=1
	>>> fct1(x)
	>>>x
	2
		>>> x=1
		>>> y=2
		>>> fct2(x,y)
		>>>x
		5
		>>>y
		2
			>>> x=1
			>>> y=2
			>>> z=3
			>>> fct3(x,y,z)
			>>>x
			8
			>>>y
			2
			>>>z
			3

1. Proposer une grammaire pour la syntaxe des fonctions.
   Indication: pensez à utiliser la partie I;
2. Donner l'ensemble V_N et l'ensemble V_T. de la nouvelle grammaire
3. Donner les premiers et les suivants de chaque symbole non terminal.
4. Élaborer la table d'analyse LL de cette grammaire.

Université Sultan Moulay Slimane

Faculté Polydisciplinaire de Khouribga

Corrigé Examen (module: Compilation)

SMI S5 – 2019 – 2020

Durée: 1h 30min

Questions de cours:

1. Citer le rôle principale d'un analyseur syntaxique.

Le rôle principale d'un AL est reconnaître la juxtaposition des mots, pour dire vers la fin que c'est une phrase correcte ou incorrecte

2. L'opérationnalisation des analyseurs syntaxiques repose sur les grammaires, donner la définition formelle d'une grammaire.

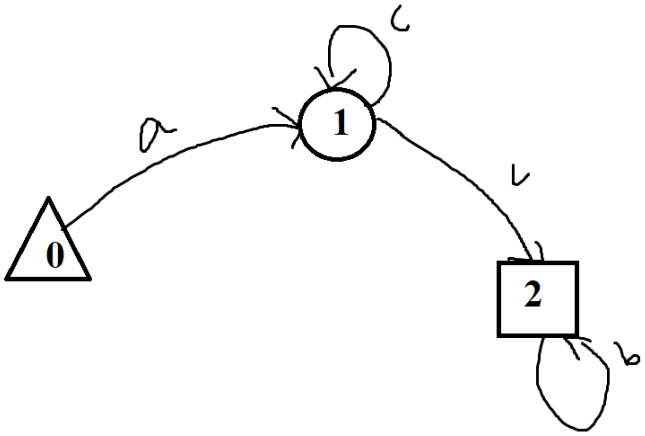
<S,Vn,Vt,P> où S signifie l'axiome, Vn l'ens des symboles non terminaux , Vt l'ens des symboles terminaux, et P l'ens des règles de productions

Exercice 1

Soient l'alphabet ∑={a,b,c} et la table de transition suivante :

Δ	a	b	c
0	1		
1			1,2
2		2	

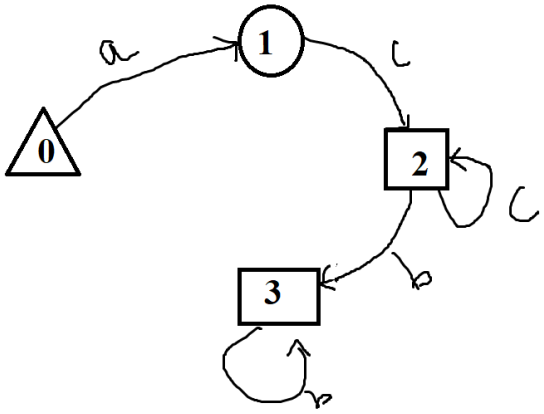
e_0={0} T={2}



1. Les mots ac, acc, accbc , acccccbbb et acccbbbc appartiennent –ils au langage reconnu par l'automate
   Les mots ac, accbc, acccbbbc, n'appartiennent pas à ce langage
   Les mots acc, acccccbbb appartiennent à ce langage

2. Donner l'automate correspondant et déterminiser le .

	a	b	c
{0}	1		
{1}			{1,2}
{1,2}		2	{1,2}
2		2	



3. Proposer une expression régulière qui correspond au langage reconnu par l'automate précédent.

1 $L_0=aL_1$
2 $L_1=cL_2$
3 $L_2=c^*| bL_3$
4 $L_3=b^*$
3 et 4 donnent $L_2=c^*|b+$ si on utilise (2) $L_1=c+|c+b+$ et d'après (1) $L_1=ac+ |a\ c+\ b+$
$L_0=ac+|ac+b+$

Exercice 2 :
On considère la syntaxe suivante:

Syntaxe	Exemple1	Exemple2	Exemple3	Exemple4
Id = valeur Id = Id + Id Id = Id + valeur	Code	Code	Code	Code
	a=10	a = b + c	a=b+20	a=b*c+d+e

Partie I :
On propose la grammaire suivante:

S = id = T E
T = id | valeur
E = + T E | *T E |ε

1. Donner l'ensemble V_N et l'ensemble V_T.

V_N={ S, T , E} V_T={ id, = , valeur, + , *} S représente l'axiome

2. Donner les premiers et les suivants de chaque symbole non terminal.

	Premiers	Suivants
S	id	$
T	id , valeur	+,*, $
E	+, * , ε	$

3. Élaborer la table d'analyse LL de cette grammaire.

1 S = id = T E

2 T = id

3 T = valeur

4 E = + T E

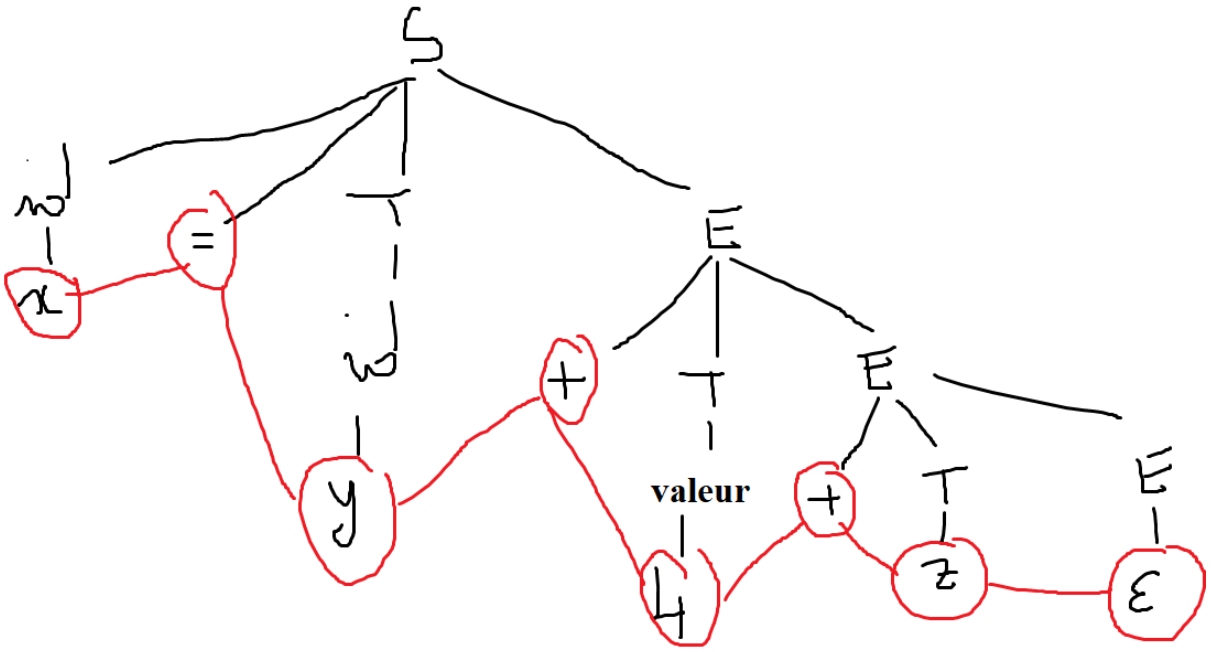
5 E = *T E

6 E = ε

	id	+	*	Valeur	=	$
S	1					
E		4	5			6
T	2			3		

Pile	Entrée	Sortie
$S	~~x =~~ **y + 4 + z$**	id = T E
$ET~~=id~~	**y + 4 + z$**	id
$E ~~id~~	~~**y**~~ **+ 4 + z$**	+TE
$ET+	**+ 4 + z$**	valeur
$E	+ z $	+TE
$ET	z$	id
$E	$	6
$	$	**Accepté**

4. Analyser la phrase **x = y + 4 + z** et Donnez l'arbre de dérivation associé. A noter que x,y et z sont des identificateurs.



Partie II :

Cette partie traitera une grammaire permettant de produire la syntaxe d'une fonction sous Python.
On considère la syntaxe suivante:

Syntaxe	**Exemple1**	**Exemple2**	**Exemple3**
def fonction(liste_paramètres): instr où instr présente l'instruction de la partie I	Code	Code	Code
	def fct1(x): x=x+x*x	def fct2(x,y): x=x+2*y	def fct1(x,y,z): x=x+2*y+z
	Résultat de l'exécution	Résultat de l'exécution	Résultat de l'exécution

	>>> x=1
	>>> fct1(x)
	>>>x
	2
		>>> x=1
		>>> y=2
		>>> fct2(x,y)
		>>>x
		5
		>>>y
		2
			>>> x=1
			>>> y=2
			>>> z=3
			>>> fct3(x,y,z)
			>>>x
			8
			>>>y
			2
			>>>z
			3

1. Proposer une grammaire pour la syntaxe des fonctions.
   Indication: pensez à utiliser la partie I;
   S' → def id (LA) : S
   LA → id | LA , id
   S → id = T E
   T → id | valeur
   E → + T E | *T E |ε

2. Donner l'ensemble V_N et l'ensemble V_T. de la nouvelle grammaire

V_N={ S', LA, S, T , E} V_T={ def, (,) , : , , , id, = , valeur, + , *} S représente l'axiome

5. Donner les premiers et les suivants de chaque symbole non terminal.

	Premiers	Suivants
S'	def	$
LA	id	) , ,
S	id	$
T	id , valeur	+,*, $
E	+, * , ε	$

3. Donner les premiers et les suivants de chaque symbole non terminal.
4. Élaborer la table d'analyse LL de cette grammaire.

1. S' → def id (LA) : S
2. LA → id
3. LA → LA , id
4. S → id = T E
5. T → id
6. T → valeur
7. E → + T E
8. E → *T E
9. E → ε

	id	+	*	Valeur	=	def	(	)	:	,	$
S'						1					
LA	2,3										
S	4										
E		7	8								9
T	5			6							

On remarque que la case [LA, id] contient deux règles alors dans ce cas on parle d'ambiguïté