

EXAMEN - Compilation - SMI- S6

Durée 2 H

Année : 2018/2019

Exercice n°1 :

Soit la grammaire d'expressions arithmétiques définie par les productions suivantes :

$$\begin{aligned} S &\mapsto E ; S \mid \epsilon \\ E &\mapsto E + T \mid E - T \mid T \\ T &\mapsto T * F \mid T / F \mid F \\ F &\mapsto \text{const} \mid (E) \end{aligned}$$

Les terminaux de la grammaire sont : $\{ ; + - * / \text{const} () \}$

- 1) Donner les dérivations les plus à gauche pour la chaîne : $3 ; 2/3*(1-3);$
- 2) Cette grammaire est-elle récursive à gauche ? Si oui, éliminer cette récursivité.

Exercice n°2 :

Soit la grammaire G1 avec attributs et actions sémantiques

Soit l'expression EXP : $7*9 + 15$

- 1) Dessiner l'arbre abstrait de l'expression EXP
- 2) Dessiner l'arbre décoré de l'expression EXP
- 3) A l'aide d'une pile, évaluer de l'expression EXP

Règles de production	actions sémantiques
$S \rightarrow E\$$	Write (E.valeur)
$E \rightarrow E1 + T$	$E.valeur := E1.valeur + T.valeur$
$E \rightarrow T$	$E.valeur := T.valeur$
$T \rightarrow T1 * F$	$T.valeur := T1.valeur * F.valeur$
$T \rightarrow F$	$T.valeur := F.valeur$
$F \rightarrow (E)$	$F.valeur := E.valeur$
$F \rightarrow I$	$F.valeur := I.valeur$
$I \rightarrow I1 \text{ digit}$	$I.valeur := I1.valeur * 10 + \text{digit.valeur}$
$I \rightarrow \text{digit}$	$I.valeur := \text{digit.valeur}$

Exercice n°3 :

- 1) Pour analyser syntaxiquement d'un mot, on se base sur une table d'analyse. Donner les étapes à suivre pour concevoir cette table à partir d'une grammaire.

2) Soit de la table d'analyse suivante correspondant à la grammaire ci-dessous :

$$\begin{cases} E \rightarrow TE' \\ E' \rightarrow +TE' | \epsilon \\ T \rightarrow FT' \\ T' \rightarrow *FT' | \epsilon \\ F \rightarrow (E) | nb \end{cases}$$

Les erreurs affichent les messages suivants :

ERR1 : « il manque un opérande »

ERR2 : « expression vide »

ERR3 : « la fin prématurée du programme. »

ERR4 : « il manque une parenthèse droite »

ERR5 : « symbole inattendu »

	nb	+	*	()	\$
E	E→TE'	ERR1	ERR1	E→TE'	ERR2	ERR3
E'		E'→+TE'			E'→ε	E'→ε
T	T→FT'	ERR1	ERR1	T→FT'	ERR1	ERR3
T'		T'→ε	T'→*FT'		T'→ε	T'→ε
F	F→nb	ERR1	ERR1	F→(E)	ERR1	ERR3
nb	Dépil					
+		Dépil				
*			Dépil			
(Dépil		
)	ERR4	ERR4	ERR4	ERR4	Dépil	ERR4
\$	ERR5	ERR5	ERR5	ERR5	ERR5	Accepte

A l'aide de cette table d'analyse et de piles, analyser les deux mots : 4*5+8 et ((5*9) et déduire l'arbre syntaxique si le mot est accepté sinon indiquer le message de l'erreur affiché.

Exercice n°4 : Donner le code à trois adresses généré pour la séquence de programme suivante:

```
A := 2 ;
For B := 5 to 10
{
  If A > B and B > 7 then C[B] := C[B] + A ;
  A := A + 2
}
```

Exercice n°5 : Peut-on effectuer l'optimisation suivante ? justifiez votre réponse.

a := d - b	a := d - b
b := c * a	b := c * a
c := d - b	c := a
d := c * a	d := c * a

Exercice n°6 : Optimisez le code suivant en appliquant une séquence de transformations. Préciser pour chaque étape quels sont les types d'optimisations qui ont été appliquées.

```
t1 = t0
t2 = 4
t3 = t1 * t1
t4 = t2 + 2
t5 = t0 ^ 2 // t0 puissance 2
t6 = t5 + t3
t7 = t4 * t6
```