

Contrôle continu - Compilation (L3 Info) Durée : 2h - documents interdits

Conventions

- Axiome : symbole non terminal de la partie gauche de la première production de la grammaire
- Symboles non terminaux : lettres *MAJUSCULES ITALIQUES*
- Symboles terminaux : lettres minuscules ou caractères spéciaux simples

1 Vrai ou faux (3 pts)

Pour chacune des affirmations suivantes dites si elle est vraie ou fausse en justifiant brièvement et de façon convaincante votre réponse (seules les réponses accompagnées d'une justification seront prises en compte).

Q1.1 Les grammaires *LR* ne sont pas ambiguës. **Vrai** - si une grammaire est ambiguë, alors il existe au moins un mot qui possède plus d'un arbre de dérivation. Lorsque l'on réalisera l'analyse syntaxique de ce mot, il arrivera un moment où on sera confronté à un conflit et c'est en choisissant chacune des alternatives de ce conflit que l'on pourra créer des arbres de dérivation différents.

Q1.2 La grammaire suivante est *LR(1)* : $\{S \rightarrow X|Y, X \rightarrow a, Y \rightarrow a\}$ **Faux** - Cette grammaire n'est pas *LR(1)* car elle est ambiguë. Elle associe deux arbres de dérivation au mot a .

Q1.3 La grammaire suivante est *LR(1)* : $\{S \rightarrow X b|Y c, X \rightarrow a, Y \rightarrow a\}$ **Vrai** - Lors de l'analyse par décalage réduction à l'aide de cette grammaire, on peut se trouver avec un a au sommet de la pile. Ce a peut être réduit à l'aide de la règle $X \rightarrow a$ ou $Y \rightarrow a$. Il suffit de regarder le prochain symbole du mot à analyser pour résoudre le conflit. Si c'est un b alors il faut réduire à l'aide de la règle $X \rightarrow a$ alors que si c'est un c , il faut réduire à l'aide de la règle $Y \rightarrow a$.

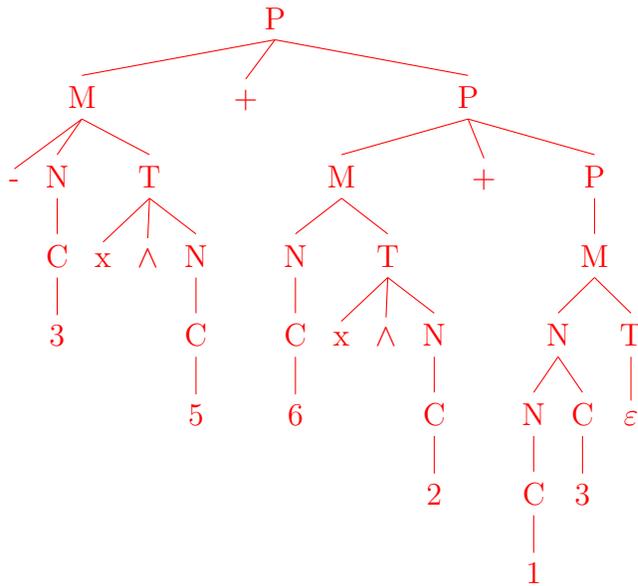
2 Grammaires attribuées (6 pts)

Soit la grammaire G suivante :

$$\begin{aligned}P &\rightarrow M + P \mid M \\M &\rightarrow NT \\M &\rightarrow -NT \\T &\rightarrow x \wedge N \mid \varepsilon \\N &\rightarrow NC \mid C \\C &\rightarrow 0 \mid 1 \mid \dots \mid 9\end{aligned}$$

G permet de générer des polynômes d'une variable, notée x . Dans ces polynômes, l'opération puissance est représentée par le symbole \wedge (x^2 s'écrit donc $x \wedge 2$).

Q2.1 Ecrire l'arbre de dérivation du polynôme $-3x \wedge 5 + 6x \wedge 2 + 13$



Q2.2 On appelle degré d'un polynôme P le degré le plus élevé de ses termes (le degré du polynôme $-3x \wedge 5 + 6x \wedge 2 + 13$, par exemple, est 5). Ecrire une grammaire attribuée à partir de G qui définit un attribut d permettant de calculer le degré du polynôme.

$P \rightarrow M + P$	$P.d = \max(M.d, P_1.d)$
$P \rightarrow M$	$P.d = M.d$
$M \rightarrow NT$	$M.d = T.d$
$M \rightarrow -NT$	$M.d = T.d$
$T \rightarrow x \wedge N$	$T.d = N.d$
$T \rightarrow \varepsilon$	$T.d = 0$
$N \rightarrow NC$	$N.d = N_1.d * 10 + C.d$
$N \rightarrow C$	$N.d = C.d$
$C \rightarrow 0 \mid 1 \mid \dots \mid 9$	$C.d = C.val$

Q2.3 On souhaite calculer la valeur d'un polynôme pour une valeur donnée de la variable x . Pour cela on définit les attributs vx et vp . vx est la valeur de x et vp est la valeur du polynôme. Indiquer pour ces deux attributs s'ils sont hérités ou synthétisés et écrire une grammaire attribuée à partir de G permettant de calculer leurs valeurs.

- vx est un attribut hérité
- vp est un attribut synthétisé

$P \rightarrow M + P$	$M.vx = P_1.vx = P.vx$	$P.vp = M.vp + P_1.vp$
$P \rightarrow M$	$M.vx = P.vx$	$P.vp = M.vp$
$M \rightarrow NT$	$N.vx = T.vx = M.vx$	$M.vp = N.vp * T.vp$
$M \rightarrow -NT$	$N.vx = T.vx = M.vx$	$M.vp = -1 * N.vp * T.vp$
$T \rightarrow x \wedge N$		$T.vp = T.vx * N.vp$
$T \rightarrow \varepsilon$		$T.vp = 1$
$N \rightarrow NC$		$N.vp = N_1.vp * 10 + C.vp$
$N \rightarrow C$		$N.vp = C.vp$
$C \rightarrow 0 \mid 1 \mid \dots \mid 9$		$C.vp = C.val$

Q2.4 Etant donné un polynôme P , on souhaite construire le polynôme P' , qui est le polynôme dérivé de P .

Pour cela on rappelle les règles de dérivation suivantes :

- $(f + g)' = f' + g'$
- $(c \times f)' = c \times f'$
- $(c)' = 0$
- $(x^c)' = \begin{cases} c \times x^{c-1} & \text{si } c > 0 \\ 0 & \text{sinon.} \end{cases}$

Où, c est une constante, x la variable et f et g des fonctions. Si on applique ces règles à l'exemple précédent, on obtient $-15x \wedge 4 + 12x \wedge 1$.

Ecrire une grammaire attribuée à partir de G qui définit un attribut δ permettant de construire la dérivée d'un polynôme sous la forme d'un chaîne de caractères.

$P \rightarrow M + P$	$P.\delta = M.\delta \parallel + \parallel P_1.\delta$
$P \rightarrow M$	$P.\delta = M.\delta$
$M \rightarrow NT$	$M.\delta = N \parallel * \parallel T.\delta$
$M \rightarrow -NT$	$M.\delta = -N \parallel * \parallel T.\delta$
$T \rightarrow x \wedge N$	$T.\delta = N.\delta \parallel * x \wedge \parallel N.\delta - 1$
$T \rightarrow \varepsilon$	$T.\delta = 0$
$N \rightarrow NC$	$N.\delta = N_1.\delta * 10 + C.\delta$
$N \rightarrow C$	$N.\delta = C.\delta$
$C \rightarrow 0 \mid 1 \mid \dots \mid 9$	$C.\delta = C.val$

3 PREMIER, SUIVANT, dérivation, ambiguïté (4 pts)

Soit la grammaire G_1 :

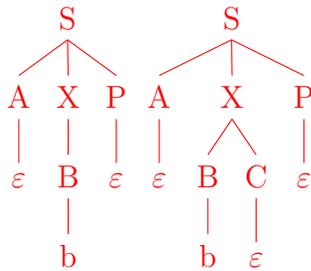
S	\rightarrow	$A X D$
A	\rightarrow	$\mathbf{a} \mid \varepsilon$
X	\rightarrow	$B C \mid B$
B	\rightarrow	$\mathbf{b} \mid \varepsilon$
C	\rightarrow	$\mathbf{c} \mid \varepsilon$
D	\rightarrow	$\mathbf{d} \mid \varepsilon$

Q3.1 Calculez les ensembles PREMIER et SUIVANT pour les symboles non terminaux de G_1 .

	PREMIER	SUIVANT
S	ε a b c d	\perp
A	ε a	\perp b c d
X	ε b c	\perp d
B	ε b	\perp c d
C	ε c	\perp d
D	ε d	\perp

Q3.2 Démontrez que la grammaire G_1 est ambiguë.

Il existe deux arbres de dérivations possible pour le mot b :



4 Création de la table SLR (5 pts)

Soit la grammaire G_2 qui est une version simplifiée de la grammaire des polynômes :

$$\begin{aligned}
 S &\rightarrow P \\
 P &\rightarrow M + P \\
 P &\rightarrow M \\
 M &\rightarrow NT \\
 T &\rightarrow x \wedge N \\
 N &\rightarrow Nc \\
 N &\rightarrow c
 \end{aligned}$$

Q4.1 Construisez l'automate $LR(0)$ de G_2 (il y a 12 états).

$$\begin{aligned}
 I_0 &= \{S \rightarrow \bullet P, P \rightarrow \bullet M + P, P \rightarrow \bullet M, M \rightarrow \bullet NT, N \rightarrow \bullet Nc, N \rightarrow \bullet c\} \\
 ALLER_{\rightarrow}(I_0, P) &= \{S \rightarrow P \bullet\} = I_1 \\
 ALLER_{\rightarrow}(I_0, M) &= \{P \rightarrow M \bullet + P, P \rightarrow M \bullet\} = I_2 \\
 ALLER_{\rightarrow}(I_0, N) &= \{M \rightarrow N \bullet T, N \rightarrow N \bullet c, T \rightarrow \bullet x \wedge N\} = I_3 \\
 ALLER_{\rightarrow}(I_0, c) &= \{N \rightarrow c \bullet\} = I_4 \\
 ALLER_{\rightarrow}(I_2, +) &= \{P \rightarrow M + \bullet P, P \rightarrow \bullet M + P, P \rightarrow \bullet M, M \rightarrow \bullet NT, N \rightarrow \bullet Nc, N \rightarrow \bullet c\} = I_5 \\
 ALLER_{\rightarrow}(I_3, T) &= \{M \rightarrow NT \bullet\} = I_6 \\
 ALLER_{\rightarrow}(I_3, c) &= \{N \rightarrow Nc \bullet\} = I_7 \\
 ALLER_{\rightarrow}(I_3, x) &= \{T \rightarrow x \bullet \wedge N\} = I_8 \\
 ALLER_{\rightarrow}(I_5, P) &= \{P \rightarrow M + P \bullet\} = I_9 \\
 ALLER_{\rightarrow}(I_5, M) &= I_2 \\
 ALLER_{\rightarrow}(I_5, N) &= I_3 \\
 ALLER_{\rightarrow}(I_5, c) &= I_4 \\
 ALLER_{\rightarrow}(I_8, \wedge) &= \{T \rightarrow x \wedge \bullet N, N \rightarrow \bullet Nc, N \rightarrow \bullet c\} = I_{10}
 \end{aligned}$$

- 1 $E \rightarrow E + T$
- 2 $E \rightarrow T$
- 3 $T \rightarrow T * F$
- 4 $T \rightarrow F$
- 5 $F \rightarrow (E)$
- 6 $F \rightarrow a$

Etat	Action					Aller à		
	a	+	*	()	\$	E	T	F
0	d5			d4		1	2	3
1		d6			acc			
2		r2	d7		r2			
3		r4	r4		r4			
4	d5			d4		8	2	3
5		r6	r6		r6			
6	d5			d4			9	3
7	d5			d4				10
8		d6			d11			
9		r1	d7		r1			
10		r3	r3		r3			
11		r5	r5		r5			

Q5.1 Simulez au moins les dix premières étapes de l'analyse LR du mot $(a + a) * a$. Les configurations sont représentées par le triplet (α, w, y) , où α est la pile avec le sommet à droite, w est la partie de la bande de lecture qui commence au caractère sous la tête de lecture (ce qui reste à analyser), et y est la séquence de symboles de sortie (numéros des productions appliquées lors des réductions).

- $(\perp 0, (a+a)*a\perp, d4)$
- $(\perp 0 4, a+a)*a\perp, d5)$
- $(\perp 0 4 5, +a)*a\perp, r6)$
- $(\perp 0 4 3, +a)*a\perp, r4)$
- $(\perp 0 4 2, +a)*a\perp, r2)$
- $(\perp 0 4 8, +a)*a\perp, d6)$
- $(\perp 0 4 8 6, a)*a\perp, d5)$
- $(\perp 0 4 8 6 5,)*a\perp, r6)$
- $(\perp 0 4 8 6 3,)*a\perp, r4)$
- $(\perp 0 4 8 6 9,)*a\perp, r3)$
- $(\perp 0 4 2,)*a\perp, r2)$
- $(\perp 0 4 8,)*a\perp, d11)$
- $(\perp 0 4 8 11, *a\perp, r5)$
- $(\perp 0 3, *a\perp, r4)$
- $(\perp 0 2, *a\perp, d7)$
- $(\perp 0 2 7, a\perp, d5)$
- $(\perp 0 2 7 5, \perp, r6)$
- $(\perp 0 2 7 5 10, \perp, r3)$
- $(\perp 0 2, \perp, r2)$
- $(\perp 0 1, \perp, acc)$

RAPPEL : construction table SLR

1. Construire $C = \{I_0, I_1, \dots, I_n\}$ la collection d'ensemble d'articles LR(0) pour G'
2. L'état i est construit à partir de I_i . Les actions d'analyse syntaxique pour l'état i sont déterminées comme suit :
 - (a) Si $A \rightarrow \alpha \bullet a\beta$ est dans I_i et si $ALLER_A(I_i, a) = I_j$, alors $ACTION[i, a] = dj$. Dans ce cas, a doit être un terminal.

(b) Si $A \rightarrow \alpha \bullet$ est dans I_i , alors $\text{ACTION}[i, a] = r_j$ où j est le numéro de la règle $A \rightarrow \alpha$ pour tout $a \in \text{SUIVANT}(A)$, à l'exception de S'

(c) Si $S' \rightarrow S \bullet$ est dans I_i , alors $\text{ACTION}[i, \$] = acc$

Si un conflit entre différentes actions résulte de ces règles, la grammaire n'est pas SLR.

3. Les transitions de transfert $\text{GOTO}[i, A]$ pour l'état i sont construites pour tout non terminal A comme suit : si $\text{ALLER_A}(I_i, A) = I_j$ alors $\text{GOTO}[i, A] = j$
4. Toutes les entrées non remplies par les règles 2 et 3 sont positionnées à **err** (cellules vides)
5. L'état initial est celui construit à partir de l'ensemble d'items contenant $S' \rightarrow \bullet S$