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

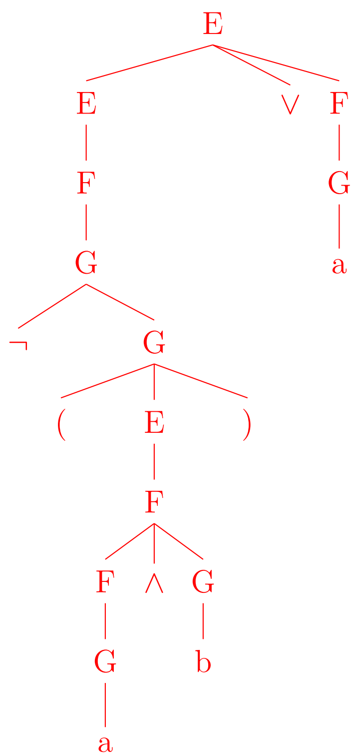
1 Grammaires attribuées *(7 pts)*

Soit la grammaire G suivante :

$$\begin{array}{lcl@{\qquad\qquad}lcl}
E & \rightarrow & E \vee F & G & \rightarrow & \neg G \\
E & \rightarrow & F & G & \rightarrow & (E) \\
F & \rightarrow & F \wedge G & G & \rightarrow & a \\
F & \rightarrow & G & G & \rightarrow & b
\end{array}$$

G permet de générer des formules de la logique propositionnelle construites avec les deux variables propositionnelles a et b.

Q1.1 Dessiner l'arbre de dérivation de la formule $\neg(a \wedge b) \vee a$



Q1.2 Ecrire une grammaire attribuée à partir de G qui définit un attribut pa qui vaut vrai si le nombre d'occurrences de la variable a dans une formule est pair. Dire si cet attribut est hérité ou synthétisé.

$$
\begin{array}{lcl|l}
E & \rightarrow & E \vee F & E.pa = (E_1.pa \wedge F.pa) \vee (\neg E_1.pa \wedge \neg F.pa) \\
E & \rightarrow & F & E.pa = F.pa \\
F & \rightarrow & F \wedge G & F.pa = (F_1.pa \wedge G.pa) \vee (\neg F_1.pa \wedge \neg G.pa) \\
F & \rightarrow & G & F.pa = G.pa \\
G & \rightarrow & \neg G & G.pa = G_1.pa \\
G & \rightarrow & (E) & G.pa = E.pa \\
G & \rightarrow & a & G.pa = faux \\
G & \rightarrow & b & G.pa = vrai
\end{array}
$$

Q1.3 On souhaite calculer la valeur d'une formule propositionnelle pour des valeurs données des variables a et b. Pour cela on définit les attributs va, vb et v. va est la valeur de a, vb celle de b et v est la valeur de la formule. 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.

— va et vb sont hérités

— v est synthétisé

$$
\begin{array}{lcl|l|l|}
E & \rightarrow & E \vee F & E_1.va = F.va = E.va & E.v = E_1.v \vee F.v \\
E & \rightarrow & F & F.va = E.va & E.v = F.v \\
F & \rightarrow & F \wedge G & F_1.va = G.va = F.va & F.v = F_1.v \wedge G.v \\
F & \rightarrow & G & G.va = F.va & F.v = G.v \\
G & \rightarrow & \neg G & G_1.va = G.va & G.v = \neg G_1.v \\
G & \rightarrow & (E) & E.va = G.va & G.v = E.v \\
G & \rightarrow & a & & G.v = G.va \\
G & \rightarrow & b & & G.v = G.vb
\end{array}
$$

Q1.4 On souhaite construire l'arbre abstrait d'une formule. Pour cela on définit les fonctions suivantes : OU(F1, F2), ET(F1,F2), NON(F), VAR(), qui construisent respectivement un nœud de type disjonction, conjonction, négation et variable. On définit de plus l'attribut sa. Etant donné un nœud n d'un arbre de dérivation, $n.sa$ est la racine de l'arbre abstrait qui correspond à n. Ecrire une grammaire attribuée à partir de G permettant de calculer la valeur de cet attribut.

$$
\begin{array}{lcl|l}
E & \rightarrow & E \vee F & E.sa = OU(E_1.sa, F.sa) \\
E & \rightarrow & F & E.sa = F.sa \\
F & \rightarrow & F \wedge G & F.sa = ET(F_1.sa, G.sa) \\
F & \rightarrow & G & F.sa = G.sa \\
G & \rightarrow & \neg G & G.sa = NON(G_1.sa) \\
G & \rightarrow & (E) & G.sa = E.sa \\
G & \rightarrow & a & G.sa = VAR() \\
G & \rightarrow & b & G.sa = VAR()
\end{array}
$$

2 Premier, Suivant, dérivation, ambiguïté *(4 pts)*

Soit la grammaire G_1 :

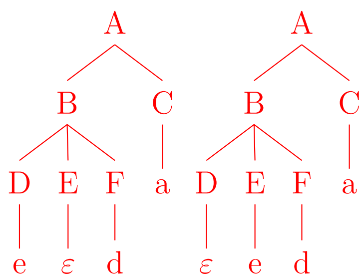
$$\begin{array}{lll@{\qquad\qquad}lll} A & \rightarrow & B\ C & D & \rightarrow & d \mid e \mid \varepsilon \\ B & \rightarrow & D\ E\ F \mid \varepsilon & E & \rightarrow & e \mid \varepsilon \\ C & \rightarrow & a\ C \mid a & F & \rightarrow & d \end{array}$$

Q2.1 Calculez les ensembles Premier et Suivant pour les symboles non terminaux de G_1.

	Premier	Suivant
A	e a d	\$
B	ε e d	a
C	a	\$
D	ε e d	e d
E	ε e	d
F	d	a

Q2.2 Démontrez que la grammaire G_1 est ambiguë.

Il existe deux arbres de dérivation possibles pour le mot `eda` :



3 Création de la table *SLR* *(6 pts)*

Soit la grammaire G_2 qui est une version simplifiée de la grammaire de formules propositionnelles :

$$\begin{array}{llll} 0 & E & \rightarrow & E \vee F \\ 1 & E & \rightarrow & F \\ 2 & F & \rightarrow & F \wedge G \\ 3 & F & \rightarrow & G \\ 4 & G & \rightarrow & \neg G \\ 5 & G & \rightarrow & a \\ 6 & G & \rightarrow & b \end{array}$$

Q3.1 Construisez l'automate $LR(0)$ de G_2 (il y a 12 états). Il n'est pas très facile d'arriver à une représentation graphique de l'automate qui soit claire. Si vous voulez, vous pouvez donner à part la définition de ses états : $I_0 = \{S \rightarrow \bullet E, \ldots\}\ldots$ et la fonction de transition sous la forme suivante $\delta(I_i, E) = I_j$, pour indiquer qu'il y a une transition étiquetée E entre les états I_i et I_j.

$I_0 : \{S \rightarrow \bullet E, E \rightarrow \bullet E \vee F, E \rightarrow \bullet F, F \rightarrow \bullet F \wedge G, F \rightarrow \bullet G, G \rightarrow \bullet a, G \rightarrow \bullet b, G \rightarrow \bullet \neg G\}$

$I_1 : \{G \rightarrow a \bullet\}$

$I_2 : \{G \rightarrow b \bullet\}$

$I_3 : \{G \rightarrow \neg \bullet G, G \rightarrow \bullet a, G \rightarrow \bullet b, G \rightarrow \bullet \neg G\}$

$I_4 : \{E \rightarrow E \bullet \vee F, S \rightarrow E \bullet\}$

$I_5 : \{E \rightarrow F \bullet, F \rightarrow F \bullet \wedge G\}$

$I_6 : \{F \rightarrow G \bullet\}$

$I_7 : \{G \to \neg G\bullet\}$
$I_8 : \{E \to E \vee \bullet F, F \to \bullet F \wedge G, F \to \bullet G, G \to \bullet a, G \to \bullet b, G \to \bullet\neg G\}$
$I_9 : \{F \to F \wedge \bullet G, G \to \bullet a, G \to \bullet b, G \to \bullet\neg G\}$
$I_{10} : \{E \to E \vee F\bullet, F \to F \bullet \wedge G\}$
$I_{11} : \{F \to F \wedge G\bullet\}$
$\delta(I_0, a) : I_1$
$\delta(I_0, b) : I_2$
$\delta(I_0, \neg) : I_3$
$\delta(I_0, E) : I_4$
$\delta(I_0, F) : I_5$
$\delta(I_0, G) : I_6$
$\delta(I_3, a) : I_1$
$\delta(I_3, b) : I_2$
$\delta(I_3, \neg) : I_3$
$\delta(I_3, G) : I_7$
$\delta(I_4, \vee) : I_8$
$\delta(I_5, \wedge) : I_9$
$\delta(I_8, a) : I_1$
$\delta(I_8, b) : I_2$
$\delta(I_8, \neg) : I_3$
$\delta(I_8, F) : I_{10}$
$\delta(I_8, G) : I_6$
$\delta(I_9, a) : I_1$
$\delta(I_9, b) : I_2$
$\delta(I_9, \neg) : I_3$
$\delta(I_9, G) : I_{11}$
$\delta(I_{10}, \wedge) : I_9$

Q3.2 Étant donné les ensembles Suivant de G_2 ci-dessous, construisez la table SLR de G_2
Suivant(E) = {\$, $\vee$}
Suivant(F) = {\$, $\vee$, $\wedge$}
Suivant(G) = {\$, $\vee$, $\wedge$}
où \$ représente le symbole de fond de pile et de fin de chaîne.

	ACTION						GOTO		
	a	b	$\wedge$	$\vee$	$\neg$	\$	E	F	G
0	d1	d2			d3		4	5	6
1			r5	r5		r5			
2						r6			
3	d1	d2			d3				7
4				d8		acc			
5			d9	r1		r1			
6			r3	r3		r3			
7			r4	r4		r4			
8	d1	d2			d3			10	6
9	d1	d2			d3				11
10			d9	r0		r0			
11			r2	r2		r2			

4 Analyse SLR *(3 pts)*

Soit la grammaire G_3 ci-dessous et sa table SLR correspondante :

1 $E \to E + F$
2 $E \to F$
3 $F \to F.G$
4 $F \to G$
5 $G \to G*$
6 $G \to \mathtt{a}$
7 $G \to \mathtt{b}$
8 $G \to (E)$

	ACTION								GOTO		
	a	b	(	)	+	.	$*$	\$	E	F	G
0	d2	d3	d1						4	5	6
1	d2	d3	d1						7	5	6
2				r6	r6	r6	r6	r6			
3				r7	r7	r7	r7	r7			
4					d8			acc			
5				r2	r2	d9		r2			
6				r4	r4	r4	d10	r4			
7				d11	d8						
8	d2	d3	d1							12	6
9	d2	d3	d1								13
10				r5	r5	r5	r5	r5			
11				r8	r8	r8	r8	r8			
12				r1	r1	d9		r1			
13				r3	r3	r3	r3	r3			

Q4.1 Simulez au moins les dix premières étapes de l'analyse LR du mot $a.(b * +a)$. 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).

(\$ 0 , `a.(b* + a)` \$, d2)
(\$ 0 2, `.(b* + a)` \$, r6)
(\$ 0 6, `.(b* + a)` \$, r4)
(\$ 0 5, `.(b* + a)` \$, d9)
(\$ 0 5 9 , `(b* + a)` \$, d1)
(\$ 0 5 9 1, `b* + a)` \$, d3)
(\$ 0 5 9 1 3, `* + a)` \$, r7)
(\$ 0 5 9 1 6, `* + a)` \$, d10)
(\$ 0 5 9 1 6 10, `+ a)` \$, r5)

(\$ 0 5 9 1 6, `+ a)` \$, r4)
(\$ 0 5 9 1 5, `+ a)` \$, r2)
(\$ 0 5 9 1 7, `+ a)` \$, d8)
(\$ 0 5 9 1 7 8, `a)` \$, d2)
(\$ 0 5 9 1 7 8 2, `)` \$, r6)
(\$ 0 5 9 1 6, `)` \$, r4)
(\$ 0 5 9 1 5, `)` \$, r2)
(\$ 0 5 9 1 7, `)` \$, d11)
(\$ 0 5 9 1 3, \$, r3)
(\$ 0 5, \$, r2)
(\$ 0 4, \$,acc)

RAPPEL : construction table SLR

1. Construire $C = \{I_0, I_1, \ldots, I_n\}$ 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 $A \to \alpha \bullet a\beta$ 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 $A \to \alpha\bullet$ est dans I_i, alors ACTION$[i, a] = rj$ où j est le numéro de la règle $A \to \alpha$ pour tout $a \in$ SUIVANT(A), à l'exception de S'
   (c) Si $S' \to S\bullet$ est dans I_i, alors ACTION$[i, \$] = acc$

   Si un conflit entre différentes actions résulte de ces règles, la grammaire n'est pas SLR.
3. Les transitions de transfert GOTO$[i, A]$ pour l'état i sont construites pour tout non terminal A comme suit : si ALLER_A$(I_i, A) = I_j$ alors 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 $S' \to \bullet S$