

Contrôle continu - Compilation (L3 Info)

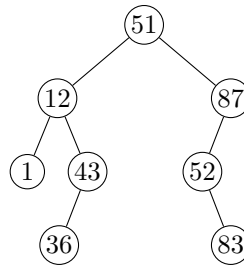
Durée : 2h - documents interdits

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 Arbres binaires de recherche

Un arbre binaire de recherche (ABR) est un arbre dont chaque nœud possède au plus deux fils, appelés fils droit et fils gauche. Chaque nœud n possède de plus une clef qui est telle que chaque nœud du sous-arbre gauche de n a une clef inférieure ou égale à celle de n , et que chaque nœud du sous-arbre droit de n possède une clef supérieure ou égale à celle de n . Voici un exemple d'ABR composé de huit nœuds.



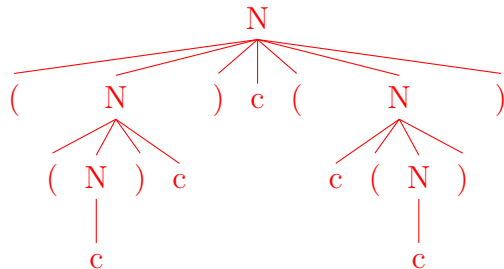
Un ABR peut être représenté sous la forme d'une chaîne parenthésée. L'arbre ci-dessus est représenté par la chaîne suivante : $((1)12((36)43))51((52(83))87)$

De telles chaînes peuvent être générées par la grammaire G suivante :

$$\begin{aligned}
 A &\rightarrow N \\
 N &\rightarrow (N)c(N) \\
 N &\rightarrow (N)c \\
 N &\rightarrow c(N) \\
 N &\rightarrow c
 \end{aligned}$$

On considère que les clefs, matérialisées par le symbole c sont les terminaux de la grammaire. On peut accéder à la valeur d'une clef grâce à l'attribut v , on écrit : $c.v$.

Q1.1 Dessiner l'arbre de dérivation de l'arbre $((c)c)c(c(c))$



Q1.2 Tout nœud d'un ABR possède un *niveau*. Le niveau de la racine vaut 0, le niveau des fils de la racine vaut 1 et ainsi de suite. On souhaite représenter le niveau de chaque nœud d'un ABR à l'aide de l'attribut n . Ecrire une grammaire attribuée, fondée sur la grammaire G qui permet de calculer le niveau des nœuds d'un ABR. L'attribut n est-il hérité ou synthétisé?

$$\begin{array}{l|l}
 A \rightarrow N & N.n = 0 \\
 N \rightarrow (N)c(N) & N_1.n = N_2.v = N.n + 1 \\
 N \rightarrow (N)c & N_1.n = N.n + 1 \\
 N \rightarrow c(N) & N_1.n = N.n + 1 \\
 N \rightarrow c &
 \end{array}$$

Q1.3 La *hauteur* d'un arbre est égale au niveau de son nœud de niveau le plus élevé. Ecrire une grammaire attribuée, fondée sur la grammaire G , qui permet de calculer l'attribut h , qui est la hauteur de l'ABR. L'attribut h est-il hérité ou synthétisé ?

$$\begin{array}{l|l}
 A \rightarrow N & A.h = N.h \\
 N \rightarrow (N)c(N) & N.h = \max(N_1.h, N_2.h) \\
 N \rightarrow (N)c & N.h = N_1.h \\
 N \rightarrow c(N) & N.h = N_1.h \\
 N \rightarrow c & N.h = N.n
 \end{array}$$

ou bien

$$\begin{array}{l|l}
 A \rightarrow N & A.h = N.h \\
 N \rightarrow (N)c(N) & N.h = \max(N_1.h, N_2.h) + 1 \\
 N \rightarrow (N)c & N.h = N_1.h + 1 \\
 N \rightarrow c(N) & N.h = N_1.h + 1 \\
 N \rightarrow c & N.h = 0
 \end{array}$$

Q1.4 On définit pour chaque nœud n les attributs min et max qui désignent respectivement la plus petite clef et la plus grande clef des nœuds appartenant au sous arbre de racine n . Ecrire une grammaire attribuée, fondée sur la grammaire G qui permet de calculer les attributs min et max des nœuds d'un ABR. Ces attributs sont-ils hérités ou synthétisés ?

$$\begin{array}{l|ll}
 A \rightarrow N & A.min = N.min & A.max = N.max \\
 N \rightarrow (N)c(N) & N.min = N_1.min & N.max = N_2.max \\
 N \rightarrow (N)c & N.min = N_1.min & N.max = c.v \\
 N \rightarrow c(N) & N.min = c.v & N.max = N_1.max \\
 N \rightarrow c & N.min = c.v & N.max = c.v
 \end{array}$$

Q1.5 un ABR est correct s'il vérifie la propriété fondamentale des ABR : étant donné un nœud n , chaque nœud du sous-arbre gauche de n a une clef inférieure ou égale à celle de n , et chaque nœud du sous-arbre droit de n possède une clef supérieure ou égale à celle de n . On définit l'attribut $corr$ qui vaut *vrai* si cette propriété est respectée et *faux* sinon. Ecrire une grammaire attribuée, fondée sur la grammaire G , qui permet de calculer l'attribut $corr$, est-il hérité ou synthétisé ?

$$\begin{array}{l|l}
 A \rightarrow N & A.corr = N.corr \\
 N \rightarrow (N)c(N) & N.corr = (c.v \geq N_1.min) \wedge (c.v \leq N_2.max) \wedge N_1.corr \wedge N_2.corr \\
 N \rightarrow (N)c & N.corr = (c.v \geq N_1.max) \wedge N_1.corr \\
 N \rightarrow c(N) & N.corr = (c.v \leq N_1.min) \wedge N_2.corr \\
 N \rightarrow c & N.corr = vrai
 \end{array}$$

On simplifie un peu la grammaire des ABR pour donner la grammaire G' :

$$\begin{array}{l} 0 \quad A \rightarrow N \\ 1 \quad N \rightarrow (N)c(N) \\ 2 \quad N \rightarrow \varepsilon \end{array}$$

On souhaite construire un analyseur LR pour la grammaire G' .

Q1.6 Calculer les suivants des symboles A et N .

$$SUIV(A) = \{\$, SUIV(N) = \{), \$\}$$

Q1.7 Construire l'automate LR de G' (il contient 9 états, plus l'état d'acceptation)

$$\begin{array}{ll} I_0 = \{S \rightarrow \cdot N, N \rightarrow \cdot (N)c(N), N \rightarrow \cdot\} & I_1 = \{S \rightarrow N \cdot\} \\ I_2 = \{N \rightarrow (\cdot N)v(N), N \rightarrow \cdot (N)c(N)\} & I_3 = \{N \rightarrow (N \cdot)c(N)\} \\ I_4 = \{N \rightarrow (N) \cdot c(N)\} & I_5 = \{N \rightarrow (N)c \cdot (N)\} \\ I_6 = \{N \rightarrow (N)c \cdot (N), N \rightarrow \cdot (N)c(N), N \rightarrow \cdot\} & I_7 = \{N \rightarrow (N)c(N) \cdot\} \\ I_8 = \{N \rightarrow (N)c(N) \cdot\} & \end{array}$$

$$\begin{array}{ll} \delta(I_0, N) & = I_1 \\ \delta(I_0, () & = I_2 \\ \delta(I_1, \$) & = acc \\ \delta(I_2, () & = I_2 \\ \delta(I_2, N) & = I_3 \\ \delta(I_3,)) & = I_4 \\ \delta(I_4, c) & = I_5 \\ \delta(I_5, () & = I_6 \\ \delta(I_6, N) & = I_7 \\ \delta(I_6, () & = I_2 \\ \delta(I_7,)) & = I_8 \end{array}$$

Q1.8 Construire la table SLR à partir de l'automate

	()	c	\$	N
0	d2	r2		r2	1
1				acc	
2	d2	r2		r2	3
3		d4			
4			d5		
5	d6				
6	d2	r2		r2	7
7		d8			
8		r1		r1	

Q1.9 Faire les huit premières étapes de l'analyse du mot $()c()$ à l'aide de la table SLR.

0	\$0	()c()\$	d2
2	\$02)c()\$	r2
3	\$023)c()\$	d4
4	\$0234	c()\$	d5
5	\$02345	()\$	d6
6	\$023456)\$	r2
7	\$0234567)\$	d8
8	\$02345678	\$	r1
0	\$0	\$	r2
1	\$01	\$	acc

2 Conflits

Q2.1 Décrire de manière concise et précise ce qu'est un conflit réduction/réduction et ce qu'est un conflit décalage/réduction.

Un conflit réduction/réduction apparaît lorsque l'analyseur peut réduire la pile avec deux règles différentes et qu'il n'a pas moyen de choisir laquelle des deux règles utiliser. Un conflit décalage/réduction apparaît lorsque l'analyseur peut réduire à l'aide d'une règle ou décaler le prochain symbole et qu'il n'a pas moyen de choisir laquelle de ces deux opérations il faut effectuer.

Soit la grammaire augmentée G suivante :

$$\begin{aligned}
 0 \quad S &\rightarrow A \\
 1 \quad A &\rightarrow Bce \\
 2 \quad A &\rightarrow Ccd \\
 3 \quad B &\rightarrow a \\
 4 \quad C &\rightarrow a
 \end{aligned}$$

Q2.2 Quel est le langage généré par G ? G est-elle ambiguë? pourquoi?

$L(G) = \{ace, acd\}$ G n'est pas ambiguë car à chacun de ces deux mots elle ne peut associer qu'un arbre

Q2.3 Calculez les suivants des symboles S, A, B, C

$$SUIV(S) = \{\$, \}, SUIV(A) = \{\$, \}, SUIV(B) = \{c\}, SUIV(C) = \{c\}$$

Q2.4 A partir de l'automate LR de G , représenté ci-dessous, construire la table SLR. Les transitions de l'automate sont représentées par la fonction de transition δ . $\delta(1, c) = 2$ nous indique qu'il y a une transition sortante de l'état 1, étiquetée par c et menant à l'état 2.

$$\begin{aligned}
 I_0 &= \{S \rightarrow \cdot A, A \rightarrow \cdot Bce, A \rightarrow \cdot Ccd, B \rightarrow \cdot a, C \rightarrow \cdot a\} & I_1 &= \{S \rightarrow A \cdot\} & I_2 &= \{B \rightarrow a \cdot, C \rightarrow a \cdot\} \\
 I_3 &= \{A \rightarrow B \cdot ce\} & I_4 &= \{A \rightarrow Bc \cdot e\} & I_5 &= \{A \rightarrow Bce \cdot\} \\
 I_6 &= \{A \rightarrow C \cdot cd\} & I_7 &= \{A \rightarrow Cc \cdot d\} & I_8 &= \{A \rightarrow Ccd \cdot\} \\
 \delta(I_0, A) &= I_1 & \delta(I_0, a) &= I_2 & \delta(I_0, B) &= I_3 & \delta(I_0, C) &= I_6 & \delta(I_1, \$) &= acc \\
 \delta(I_3, c) &= I_4 & \delta(I_4, e) &= I_5 & \delta(I_6, c) &= I_7 & \delta(I_7, d) &= I_8
 \end{aligned}$$

	<i>a</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>\$</i>	<i>A</i>	<i>B</i>	<i>C</i>
0	2					1	3	6
1					<i>acc</i>			
2		<i>r3r4</i>						
3		<i>d4</i>						
4				<i>d5</i>				
5					<i>r1</i>			
6		<i>d7</i>						
7			<i>d8</i>					
8					<i>r2</i>			

Q2.5 La grammaire G est-elle LR ? si ce n'est pas le cas, dites quel type de conflit elle présente et quelles règles sont concernées.

La grammaire G n'est pas LR, elle présente un conflit de réduction réduction entre les règles 3 et 6 lorsque l'analyseur est dans l'état 2

Q2.6 On modifie la grammaire en espérant que la nouvelle grammaire G' soit LR :

$$\begin{array}{l}
 0 \quad S \rightarrow A \\
 1 \quad A \rightarrow Bce \\
 2 \quad A \rightarrow acd \\
 3 \quad B \rightarrow a
 \end{array}$$

Sans faire la table SLR mais en vous aidant des questions précédentes, dites si G' est LR. Si ce n'est pas le cas, dites quel type de conflit elle présente et quelles règles sont concernées.

La grammaire G' n'est pas LR, elle présente un conflit de décalage réduction. Lorsque la pile contient le symbole a , et que le prochain symbole est c , l'analyseur peut soit faire un décalage soit réduire par la règle 3

Q2.7 Ecrire une grammaire LR G'' , telle que $L(G'') = L(G)$.

$$\begin{array}{l}
 0 \quad S \rightarrow A \\
 1 \quad A \rightarrow ace \\
 2 \quad A \rightarrow acd
 \end{array}$$