

Analyse Syntaxique – TD3

Exercice 1. On considère la grammaire algébrique $G = (X, V, A, R)$ avec $X = \{a, b\}$, $V = \{A, S\}$ et $R = \{A \rightarrow aSb \mid aA, S \rightarrow aaS \mid aa\}$. Donner un automate reconnaissant $L(G)$.

Exercice 2. On considère la grammaire algébrique $G = (X, V, A, P)$ avec $X = \{a, b\}$, $V = \{A\}$ et $P = \{A \rightarrow aAb \mid AA \mid bAa \mid \epsilon\}$.

Question 1. *Tout mot de $L(G)$ commençant par a se termine-t-il par b ?*

Question 2. *Donner une dérivation dans G permettant d'engendrer le mot $aabbbaab$ à partir de l'axiome A .*

Question 3. *Que vaut le langage $L(G)$? Justifier.*

Exercice 3. Pour chacun des langages suivants, donner une grammaire algébrique qui l'engendre :

- a^*b
- $\{a^n b^p \mid n > p \geq 0\}$
- $\{a^n b^p \mid n \neq p\}$
- $\{a^n b^p c^n \mid n \geq 0, p \geq 0\}$
- $\{u \in (a + b)^* \mid u \text{ est un palindrome}\}$

Exercice 4. Donner une grammaire algébrique engendrant toutes les expressions rationnelles n'utilisant pas le mot vide sur $\{a, b\}$. L'alphabet terminal doit être $\{a, b, (,), +, *\}$ (Le symbole $+$ désigne l'union de deux langages).

Exercice 5. La syntaxe d'une commande est décrite comme suit : une commande est composée de 4 parties dont seule la première est obligatoire.

- la première partie est constituée du nom de la commande (un identificateur de commande).
- la deuxième partie est composée du nom de la variable (un identificateur) sur laquelle va s'appliquer la commande.
- la troisième partie correspond à la liste des paramètres : c'est une suite de paramètres (qui sont ici des caractères) séparés par des virgules, encadrée par des accolades.
- la dernière partie correspond à la partie option. Elle est constituée par une liste d'options séparées par des virgules, encadrée par des crochets. Une option est représentée soit par un caractère, soit par un ensemble ayant la syntaxe de l'ensemble des paramètres.

Question 4. *Ecrire une grammaire décrivant la syntaxe d'une commande, i.e. définir l'alphabet terminal, l'alphabet des variables, l'axiome et les règles de production. Le langage engendré est-il reconnaissable ?*

Exercice 6. On considère des arbres dont les nœuds sont étiquetés par des symboles. A chaque symbole, est associé un entier positif ou nul : son arité. Un arbre sera correctement étiqueté si, pour chaque noeud de l'arbre, le nombre de ses successeurs directs est égal à l'arité de son étiquette. Dans l'exercice, les symboles considérés seront g d'arité 2, h d'arité 1 et c d'arité 0. Un arbre t sera identifié à sa notation linéaire, i.e. l'ensemble des arbres correctement étiquetés est engendré par la grammaire $G = (\{g, h, c, (,), \cdot\}, \{A\}, A, P)$ avec $P = \{A \rightarrow g(A, A) \mid h(A) \mid c\}$. Par exemple, $g(h(c), g(c, h(h(c))))$ et $g(g(h(c), c), h(h(c)))$ représentent bien des arbres correctement étiquetés sur cet alphabet tandis que ni $g(c)$ ni $h(h(c), g)$ ni $c(c, c)$ ne sont corrects.

Question 5. Dire quelles sont les expressions correctes parmi les expressions suivantes. Justifier.

1. $g(c, h(c))$
2. $h(g(c, c), c)$
3. $h(h(h(g(c, c))))$
4. $g(h(c), h(h(c)))$
5. $g(g(c, c), g(c, c), c)$

Question 6. Définir une grammaire qui engendre l'ensemble des arbres correctement étiquetés qui contiennent un nombre pair de c .

Question 7. Définir une grammaire qui engendre l'ensemble des arbres correctement étiquetés qui contiennent au moins un g .

La hauteur d'un arbre est définie comme étant la longueur de la plus grande branche de l'arbre ; par exemple, c est de hauteur nulle, la hauteur de $g(g(h(c), c), h(h(c)))$ est 3, et $g(h(c), g(c, h(h(c))))$ a pour hauteur 4.

Question 8. Définir une grammaire qui engendre l'ensemble des arbres correctement étiquetés de hauteur supérieure ou égale à 2.

Exercice 7.

Question 9. Donner une grammaire algébrique engendrant les expressions booléennes construites sur l'alphabet terminal $\{V, F, \wedge, \vee, \neg, (,)\}$.

Question 10. Même question pour les expressions booléennes dont la valeur de vérité est vraie.

Exercice 8. On considère la grammaire définie par l'ensemble de règles suivant, extrait de la grammaire du langage C :

$$\begin{array}{ll} \text{expression} & \rightarrow \text{unaire} \\ \text{unaire} & \rightarrow * \text{ unaire} \mid \text{postfixe} \\ \text{postfixe} & \rightarrow \text{postfixe} ++ \mid \text{primaire} \\ \text{primaire} & \rightarrow \text{i} \mid (\text{ expression}) \end{array}$$

Question 11. Donnez l'arbre de dérivation pour le mot $*i++$.

Question 12. En déduire si dans le langage C, l'expression $*i++$ est équivalente à l'expression $(*i)++$ ou à l'expression $*(i++)$.

Question 13. Donnez les arbres de dérivation des expressions $(*i)++$ et $*(i++)$.