USTHB
Faculté d'Electronique & Informatique
Département d'Informatique
L2-ACAD (Sections A & C)

Année 2019/2020

Théorie des Langages

Solutions des Exercices de la Série 1

Exercice 6 :

On donne les grammaires pour les langages suivants :

3) $L_3=\{a^i b^j \ / \ i \geq j+1, j\geq 0\}$

Exemples : L_3 = {a, aab, aaab, aaaa, aaabb, aaaaab, …., }

Ici les deux parties du mot sont dépendantes : on ne peut pas générer les a et les b séparément.

$i \geq j+1 \Leftrightarrow i= j+k+1 \ / \ k\geq 0$

dans ce cas : $a^i b^j = a^{k+1} a^j b^j$ (par remplacement de i)

Donc, une grammaire pour L_3 est la suivante :

G=<T, N, S, P> où T={a,b} N={S, A, B} P :

S→ AB

A→ aA / a /* la partie des **a** seuls avec au moins un **a***/

B→ aBb / ε /* la partie $\mathbf{a^j b^j}$ qui est de la forme $\mathbf{a^n b^n}$*/

Cette grammaire n'est pas de type 3 à cause de la première et la quatrième règle.
Elle est de type 2 car toutes les règles sont de la forme $A\rightarrow\alpha$, $A\in N$ et $\alpha \in (T\cup N)^*$

Remarque : Puisque $i= j+k+1$, on peut écrire $a^i b^j = a^j a^{k+1} b^j$ avec $j,k\geq 0$ et dans ce cas on peut avoir une autre grammaire pour ce langage qui est la suivante :

G=<T, N, S, P> où T={a,b} N={S} P :

S→ aS / aSb / a /*avec chaque **b** à droite il y a un **a** à gauche mais on peut avoir des **a** supplémentaires à gauche. Minimum un a*/

4) $L_4=\{c^n w \ / \ w\in\{a,b\}^* \text{ et } |w| = n \text{ et } n\geq 0\}$

Exemples : L_4 = { ε, ca, cb, ccaa, ccba, cccaba, cccbba…., **ccc…(a∪b) (a∪b) (a∪b)**, … }
Les mots de L_4 sont composés de **n** occurrences de la lettre **c** suivis d'un mot de longueur **n** (composé de **n** lettres, chacune peut être **a** ou **b**).

Donc, une grammaire pour L_4 est la suivante :

G=<T, N, S, P> où T={a,b,c} N={S} P :

S→ cSa / cSb / ε

USTHB Année 2019/2020
Faculté d'Electronique & Informatique
Département d'Informatique
L2-ACAD (Sections A & C)

Théorie des Langages
Solutions des Exercices de la Série 1

Pour chaque **c** à gauche, il y a un **a** ou un **b** à droite. La sortie se fait par **ε.**
Cette grammaire est de type 2.

5) $L_5 = \{a^{2m} b^{2n} c^{2p} / 2m+n+1 = p, n\geq 1 \text{ et } m,p\geq 0 \}$
Exemples : L_5={bbcccc, aabbcccccccc, … }

D'après la condition, on remplace **p** par **2m+n+1** dans la forme des mots et on obtient :
$\mathbf{a^{2m} b^{2n} c^{2p} = a^{2m} b^{2n} c^2 c^{2n} c^{4m}}$ avec $n\geq 1$ et $m\geq 0$
Dans la nouvelle forme des mots, les correspondances sont les suivantes :
Pour 2 **a** à gauche (extérieur), ça leur correspond 4 **c** à droite : $\mathbf{\underline{a^{2m}}\, b^{2n}\, c^2\, c^{2n}\, \underline{c^{4m}}}$
Pour 2 **b** à gauche (intérieur), ça leur correspond 2 **c** à droite : $\mathbf{a^{2m}\, \underline{b^{2n}}\, c^2\, \underline{c^{2n}}\, c^{4m}}$
Il y a 2 **c** au milieu.

Donc, une grammaire pour L_4 est la suivante :

G=<T, N, S, P> où T={a,b,c} N={S, A} P :

S→ aaScccc / A /* pour 2 **a** à gauche, ça correspond 4 **c** à droite*/
A→ bbAcc / bbcccc /* pour 2 **b** à gauche, ça correspond 2 **c** à droite. La sortie par **bbcccc** car au moins n=1 et donc p=2*/

Cette grammaire est de type 2.

6) $L_6=\{a^m b^n c^p / m>n \text{ ou } 2n\leq p \text{ et } m,n,p\geq 0 \}$

Exemples : L_5 = {a, acc, aabcc, bcc, aabbcccc, …}

Ici, il y a deux conditions combinées par un **ou**. Donc, on peut voir L_6 comme étant l'union de deux langages : le premier respectant la première condition et le deuxième respectant la deuxième condition.
On va étudier les deux conditions séparément :

Condition 1) $m > n \Leftrightarrow m= n+k$ avec $k >0$.
Dans ce cas : $a^m b^n c^p = a^k a^n b^n c^p$ avec $n,p\geq 0$ et $k >0$ (k, n, p sont indépendants)
Donc la grammaire est :
$G_1=<T_1, N_1, S_1, P_1>$ où T_1={a,b,c} $N_1=\{S_1, A_1, B_1, C_1\}$ P_1 :

S_1→ $A_1B_1C_1$ /* les trois parties du mot */

A_1→ aA_1 / a /* la première partie : une suite aléatoire de a. La sortie par a (k>0)*/

B_1→ aB_1b / ε /* la deuxième partie : la séquence a^nb^n. La sortie par ε */

C_1→ cC_1 / ε /* la troisième partie : une suite aléatoire de c. La sortie par ε */

USTHB Année 2019/2020
Faculté d'Electronique & Informatique
Département d'Informatique
L2-ACAD (Sections A & C)

Théorie des Langages
Solutions des Exercices de la Série 1

Condition 2) $p \geq 2n \Leftrightarrow p = 2n+j$ avec $j \geq 0$.
Dans ce cas : $a^m b^n c^p \Leftrightarrow a^m b^n c^{2n} c^j$ avec m, n, $j \geq 0$ (m, n, j sont indépendants)
Donc la grammaire est :
$G_2 = <T_2, N_2, S_2, P_2>$ où $T_2=\{a,b,c\}$ $N_2=\{S_2, A_2, B_2, C_2\}$ P_2 :

$\mathbf{S_2 \rightarrow A_2B_2C_2}$ /* les trois parties du mot */

$\mathbf{A_2 \rightarrow aA_2 / \varepsilon}$ /* la première partie : une suite aléatoire de a. La sortie par ε */

$\mathbf{B_2 \rightarrow bB_2cc / \varepsilon}$ /* la deuxième partie : la séquence a^nb^n. La sortie par ε */

$\mathbf{C_2 \rightarrow cC_2 / \varepsilon}$ /* la troisième partie : une suite aléatoire de c. La sortie par ε */

Ainsi, on obtient la grammaire globale, en ajoutant juste un nouvel axiome S et une nouvelle règle $S \rightarrow S_1 / S_2$ et en maintenant toutes les productions de G_1 et de G_2.

Ainsi, la grammaire qui génère L_6 est la suivante :

$G=<T, N, S, P>$ où $T=\{a,b,c\}$ $N=N_1 \cup N_2 \cup \{S\}$, $P=P_1 \cup P_2 \cup \{S \rightarrow S_1 / S_2\}$

Remarques : les non terminaux de la grammaire globale, sont ceux de la première, plus ceux de la deuxième, plus S (nouvel axiome). S_1 et S_2 ne sont plus axiomes. L'ensemble des productions de la grammaire globale est composé des productions de la première grammaire, plus celles de la deuxième en leur ajoutant la règle $S \rightarrow S_1 / S_2$.

Cette grammaire est de type 2.

8) $L_8=\{w \in \{a,b,c\}^* \ / \ |w|_c=3p+1, \quad p \geq 0\}$

Exemples : L_8 = { c, cb, acab, cccc, bacaacabbcbcaa, cccbcbaaaa, … }
Les mots de L_8 sont composés des mots où le nombre de **c** est un multiple de 3 plus 1 (3p+1).
Aucune condition sur le nombre de **a** ou le nombre de **b**.

Remarque : d'une manière générale, le nombre de **c** dans un mot quelconque peut être : soit un multiple de 3, soit un multiple de 3 plus 1, soit un multiple de 3 plus 2. Notons qu'un nombre multiple de 3 plus 3 est multiple de 3.

On peut écrire $|w|_c \equiv r$ [3] qui se lit le nombre de **c** est congrue à **r** modulo 3 avec $r \in \{0, 1, 2\}$.
Donc, on peut répartir les mots de X* en 3 classes d'équivalences (3p, 3p+1 et 3p+2).

Si on ajoute une lettre **a** ou **b** à un mot, le nombre de **c** reste inchangé mais si on ajoute un **c**, il augmente de 1 (modulo 3), comme illustré dans le tableau suivant :

USTHB
Faculté d'Electronique & Informatique
Département d'Informatique
L2-ACAD (Sections A & C) Année 2019/2020

Théorie des Langages
Solutions des Exercices de la Série 1

$\|w\|_c$\lettre	a	b	c
$\|w\|_c$ = 3p	3p	3p	3p+1
$\|w\|_c$ = 3p+1	3p+1	3p+1	3p+2
$\|w\|_c$ = 3p+2	3p+2	3p+2	3p

Donc, une grammaire pour L_8 est la suivante :

G=<T, N, S, P> où T={a,b,c} N={S, A, B}

S : représente les mots qui ont le nombre de **c** multiple de 3 (3p)

A : représente les mots qui ont le nombre de **c** multiple de 3 plus 1 (3p+1).

B : représente les mots qui ont le nombre de **c** multiple de 3 plus 2 (3p+2).

L'ensemble des productions P est :

S→ aS / bS / cA /* Si on génère **a** ou **b**, le nombre de **c** ne change pas. Si on génère **c**, il devient 3p+1*/

A→ aA / bA / cB / ε /* Si on génère **a** ou **b**, le nombre de **c** ne change pas. Si on génère **c**, il devient 3p+2. Ici, on peut s'arrêter à n'importe quel moment*/

B→ aB / bB / cS /* Si on génère **a** ou **b**, le nombre de **c** ne change pas. Si on génère **c**, il redevient 3p*/

Cette grammaire est de type 3.