**Exercice 1 :**

On donne les grammaires pour les langages suivants :

**7)** L7={am bn / m ≠ n et m, n ≥0 }

**Exemples :** L7 = {a, b, aaaab, abbb, aaa, bbbb, aaabbbbbb, …}

On remarque que : **m ≠ n ⇔ m>n ou m<n** donc on se retrouve dans le cas de deux conditions combinées par un **ou**.

**Condition 1)** m > n **⇔** m= n+k avec k>0**.**

Dans ce cas : am bn **=** ak an bn avec n≥0 et k>0 (k, n sont indépendants)

Donc la grammaire est :

G1=<T1, N1, S1, P1> où T1={a,b} N1={S1, A1, B1} P1 :

**S1🡪 A1B1** /\* les deux parties du mot \*/

**A1🡪 aA1 / a** /\* la première partie : une suite aléatoire de a. La sortie para (k>0)\*/

**B1🡪 aB1b / ɛ** /\* la deuxième partie : la séquence anbn. La sortie parɛ \*/

**Condition 2)** m < n ⇔n= m+j avec j>0**.**

Dans ce cas : am bn **=** am bm bj  avec m≥0 et j>0 (m, j sont indépendants)

Donc la grammaire est :

G2=<T2, N2, S2, P2> où T2={a,b} N2={S2, A2, B2} P2 :

**S2🡪 A2B2** /\*les deux parties du mot\*/

**A2🡪 aA2b/ ɛ** /\*la première partie : la séquence ambm. La sortie parɛ \*/

**B2🡪 bB2 / b** /\*la deuxième partie : une suite aléatoire de b. La sortie parb (j>0)\*/

Ainsi, on obtient la grammaire globale, en ajoutant juste un nouvel axiome S et une nouvelle règle S🡪S1 / S2 et en maintenant toutes les productions de G1 et de G2.

Ainsi, la grammaire qui génère L6 est la suivante :

G=<T, N, S, P> où T={a,b,c} N=N1∪N2∪{S}, P=P1∪P2∪ {S🡪S1 / S2}

Cette grammaire est de type 2.

**9)** L9={w∈{a,b,c}\* / ⎢w⎟a + ⎢w⎟c est divisible par 3}

**Exemples :** L9 = {ɛ, bb, aabc, cabbc, aabbba, cbbcc, accbaabbabb, … }

Les mots de L9 sont composés des mots où le nombre de **a** plus le nombre de **c** est un multiple de 3 (3p). Aucune condition sur le nombre de **b**. ɛ appartient au langage car ⎢ɛ⎟a + ⎢ɛ⎟c=0

**Remarque :**

Dans un mot quelconque w : ⎢w⎟a+⎢w⎟c= 3p **ou** ⎢w⎟a+⎢w⎟c=3p+1 **ou** ⎢w⎟a+⎢w⎟c=3p+2.

Donc, une grammaire pour L9 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 **a** plus **c** multiple de 3 (3p)

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

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

L’ensemble des productions P est :

**S🡪 bS / aA / cA / ɛ /\*** Si on génère **b**, le nombre de **a** et **c** ne change pas. Si on génère **a** ou **c**, il devient 3p+1. On peut s’arrêter à n’importe quel moment \*/

**A🡪 bA / aB / cB /\*** Si on génère **b**, le nombre de **a** et **c** ne change pas. Si on génère **a** ou **c** , il devient 3p+2. \*/

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

Cette grammaire est de type 3.

**10)** L10={w∈{0, 1}\* / w est divisible par 3}

**Exemples :** L10 = {0, 11, 011, 110, 000, 0011, 1001, 001001, 1100, … }

D’une manière générale, si w∈{0, 1}\* alors :

w0=2\*w (ajouter 0 à droite revient à multiplier w par 2. Exp : si w=1 alors w0=10 qui est 2)

w1=2\*w+1 (ajouter 1 à droite revient à multiplier w par 2 et d’ajouter 1. Exp : si w=1 alors w1=11 qui est 3)

Pour notre exercice, un mot quelconque en binaire est soit un multiple de 3 (reste de sa division sur 3 est 0), soit un multiple de 3 plus 1 (reste de sa division sur 3 est 1), soit un multiple de 3 plus 2 (reste de sa division sur 3 est 2).

Donc, une grammaire pour L10 est la suivante :

G=<T, N, S, P> où T={0,1} N={S, Z, U, D}

**Où S :** c’est l’axiome.

**Z :** représente les nombres binaires multiple de 3 (reste = **Z**éro).

**U :** représente les nombres binaires multiple de 3 plus 1 (reste = **U**n).

**D :** représente les nombres binaires multiple de 3 plus 2 (reste = **D**eux).

L’ensemble des productions P est :

**S🡪 0Z / 1U /\*** Si on génère **0**, on obtient le nombre 0 qui est multiple de 3. Si on génère **1**, on obtient qui est multiple de 3 plus 1\*/

**Z🡪 0Z / 1U / ɛ /\*** Si on génère **0**, le nombre reste toujours multiple de 3. Si on génère **1** il devient multiple de 3 plus 1. On ne peut s’arrêter qu’ici \*/

**U🡪 0D / 1Z /\*** Si on génère **0**, le nombre devient multiple de 3 plus 2. Si on génère **1** il devient multiple de 3 \*/

**D🡪 0U / 1D /\*** Si on génère **0**, le nombre devient multiple de 3 plus 1. Si on génère **1** il devient multiple de 3 plus 2 \*/

Cette grammaire est de type 3.

**Remarque :** Une autre solution serait de considérer **Z** comme étant l’axiome. Dans ce cas, il faut prévoir deux sorties : une à partir de Z par 0 et l’autre à partir de U par 1 (il faut ajouter les règles Z🡪0 et U🡪1)

**11)** L11={wuwR / w∈{a, b}\* et u∈{c}\* }

**Exemples :** L11 = {ɛ, c, ccc, aca, abbcbba, aacaa, bbbcbbb, baabcbaab, … }

Les mots de L11 sont composés comme suit : à gauche on trouve un mot quelconque, à droite son miroir et au milieu une suite aléatoire de c.

Donc, une grammaire pour L11 est la suivante :

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

**S🡪 aSa / bSb/ C** /\* chaque **a** à gauche (ou **b**), lui correspond un **a** à droite (ou **b**)\*/

**C🡪 cC / ɛ /**\* pour générer la suite aléatoire de **c** qui joue le rôle de séparateur\*/

Cette grammaire est de type 2.