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

Exercice 4 (Suite):

Les grammaires qui génèrent les langages suivants :

7) L'ensemble des mots de passe de sécurité faible, qui sont formés que des lettres ou que des chiffres.

Exemple : L = {a, 7, aa, 2020, 04, thl, usthb, 16, … }

G=<T, N, S, P> où T={0,..9, a,…z, A, …, Z}

N={{<Password>, <SuiteLettres>, <SuiteChiffres>, <Lettre>, <Chiffre>} et

S= <Password> et P =

<Password>→ <SuiteLettres> / <SuiteChiffres> /*soit des lettres, soit des chiffres*/

<SuiteLettres>→ <Lettre> <SuiteLettres> / <Lettre> /*une ou plusieurs lettres*/

<SuiteChiffres>→ <Chiffre> <SuiteChiffres> / <Chiffre> /*un ou plusieurs chiffres*/

<Lettre>→a / b /…../ z / A / B /…../ Z /* les différentes possibilités de lettres*/

<Chiffre>→0 / 1 /…. / 9 /*les différentes possibilités de chiffres*/

8) L'ensemble des mots de passe de sécurité moyenne, qui comportent au moins une lettre **et** au moins un chiffre mais aucun caractère spécial.

Exemple : L = {a7, 9aa, 20z20, s400, thl2020, usthb16alger, … }

G=<T, N, S, P> où T={0,..9, a,…z, A, …, Z}

N={<Password>, <Suite>, <Chiffre>, <Lettre> } et S= <Password>

L'ensemble des productions =

<Password> →<Suite> <Chiffre> <Suite> <Lettre> <Suite> /

<Suite> <Lettre> <Suite> <Chiffre> <Suite

/* comporte un chiffre puis une lettre ou inversement et une suite aléatoire avant, après et entre ces deux symboles */

<Suite>→<Lettre> <Suite> / <Chiffre> <Suite> / 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

<Chiffre> → 0 / 1 / … / 9 /* les différentes possibilités du chiffre */

<Lettre>→ a / b /…../ z / A / B /…../ Z /* les différentes possibilités de la lettre */

Remarques :

Le non terminal <Suite> génère la partie aléatoire (combinaison entre lettres et/ou chiffres) qui se trouve avant, après ou entre la lettre et le chiffre.

Exercice 5 :

Il faut remarquer que chaque mot correspondant à un message est composé de deux parties :

- Première partie : l'information utile
- Deuxième partie : bit de parité

Exemples de messages valides (respectant la parité)

L={00, 11, 000, 101, 10111, 10100, 110101, ….}

Exemples de messages altérés (ne respectant pas la parité)

L={01, 10, 001, 100, 10110, 10101, 110100, ….}

Finalement, l'ensemble des mots respectant la parité est le langage des mots ayant un nombre pair de 1. Sa grammaire est :

G=<T, N, S, P> où T={0,1} N={S,P, I} et les productions :

S→ 0P / 1I /* 0 ou 1 est le plus petit message qu'on lui ajoutera le bit de parité*/

P→ 0P / 1I / 0 /* P représente le nombre pair de 1 */

I→ 0I / 1P/ 1 /* I représente le nombre impair de 1 */

Exercice 6 :

On donne les grammaires pour les langages suivants :

1) $L_1 = \{(ab)^n\, a^{2p}\, (ba)^m \,/\, n, p\geq 0 \text{ et } m\geq 1\}$

Exemples : L_1={ba, abba, aaba, ababaaaaaba, …., **ab…ab aa…aa ba…ba**, … }

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

Il faut remarquer que les mots sont composés de trois parties <u>**indépendantes**</u> : les **ab**, les **aa**, les **ba**

Donc, première grammaire pour L_1 est la suivante :

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

S→ ABC /* les trois parties du mot */

A→ abA / ε /* la première partie : une suite aléatoire de ab. La sortie par ε */

B→ aaB / ε /* la deuxième partie : une suite aléatoire de aa. La sortie par ε */

C→ baC / ba /*la troisième partie : une suite aléatoire de ba. Au moins un ba ($m\geq1$)*/

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

Une deuxième grammaire pour L_1 est la suivante :

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

S→ abS / A /* on commence par générer la suite des **ab***/

A→ aaA / B /* dans une deuxième étape on génère la suite des **aa** */

B→ baB / ba /* dans la dernière, on génère la suite des **ba** avec minimum un ba*/

Cette grammaire est de type 3. Donc c'est elle qui va être retenue.

Remarque : dans la première grammaire, les trois parties des mots ont été générées indépendamment l'une de l'autre (en parallèle). Dans la deuxième grammaire, les trois parties ont été générées dans l'ordre (en séquentiel).

2) $L_2 = \{a^{2i+3}\, b^{2j+2} \;/\; i, j\geq 0\}$

Exemples : L_2 = {aaabb, aaaaabb, aaabbbb, aaaaabbbb, …., **aa…aa aaa bb…bb bb**, … }

Il faut remarquer que les mots sont composés de deux parties <u>**indépendantes**</u> : les **aa (suivis de aaa)**, les **bb (suivis de bb)**
Donc, première grammaire pour L_2 est la suivante :

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

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

S→ AB /* les deux parties du mot */

A→ aaA / aaa /* la première partie : une suite aléatoire de aa qui se termine par aaa*/

B→ bbB / bb /* la deuxième partie : une suite aléatoire de bb qui se termine par bb*/

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

Une deuxième grammaire pour L_2 est la suivante :

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

S→ aaS / aaaB /* on commence par générer la suite des **aa** avec au moins **aaa***/

B→ bbB / bb /* on génère la suite de **bb** avec au moins **bb** */

Cette grammaire est de type 3. Donc c'est elle qui va être retenue.