

Exercice 1

Soit X un alphabet et soient $a, b \in X$ et $w \in X^*$.

Montrer que si $wa = bw$ alors $a = b$.

Exercice 2

Soit X un alphabet et soient $x, y, w \in X^*$.

1. Vérifier si : $\text{Mir}(xy) = \text{Mir}(x)\text{Mir}(y)$

2. Montrer que $\text{Mir}(xy) = \text{Mir}(y)\text{Mir}(x)$

Exercice 3

Pour tout langage L défini sur un alphabet X , on définit la fonction suivante :

$$\text{Fact}(L) = \{u \mid \text{il existe un mot } v \in X^* \text{ tel que } uv \in L\}$$

Soient $L_a = \{a^n / n \geq 1\}$, $L_b = \{b^{2n} / n \geq 0\}$

1) Calculer $L_a.L_a$, $L_b.L_b$ et $L_a.L_b$.

2) Calculer $\text{Fact}(L_a)$, $\text{Fact}(L_b)$, $\text{Fact}(\text{Fact}(L_b))$ et $\text{Fact}(L_a.L_b)$.

Exercice 4

Donner les grammaires générant les langages suivants :

1. L'ensemble des nombres binaires.
2. L'ensemble des nombres binaires sans 0 inutiles en tête.
3. L'ensemble des nombres binaires de longueur paire.
4. Les nombres décimaux éventuellement signés n'ayant pas de 0 inutiles. Rappelons que la partie (optionnelle) après la virgule ne se termine pas par un 0.
5. L'ensemble des noms de variable (identificateurs) en Java. Un nom de variable en Java commence par une lettre alphabétique ou le caractère underscore (`_`) suivi par une suite quelconque de lettres alphabétiques, de chiffres et l'underscore.
6. L'ensemble des tableaux de caractères alphabétiques. Un tableau commence par `{` et se termine par `}` et les caractères sont séparés par virgule. Chaque caractère est compris entre deux quotes simples. Le tableau peut être vide.
7. L'ensemble des mots de passe de sécurité faible, qui sont formés que des lettres ou que des 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.

Exercice 5

On désire tester la conformité des messages reçus à un format prédéfini. Les messages sont des mots sur l'alphabet $\{0, 1\}$ munis d'un bit supplémentaire appelé bit de parité. Le bit de parité est le nombre de 1 que contient la partie du message qui le précède, calculé modulo 2. Le bit de parité crée une redondance susceptible de détecter les messages altérés.

- Donnez des mots correspondants à des messages respectant la parité et donc ils sont supposés non altérés.
- Donnez des mots correspondants à des messages « altérés » (ne respectant pas la parité).

On appelle L l'ensemble de tous les mots possibles respectant la parité.

Donner une grammaire qui génère le langage L .

Exercice 6

Donner les grammaires générant les langages suivants en donnant le type de la grammaire :

$$L1 = \{(ab)^n a^{2p} (ba)^m \mid n, p \geq 0 \text{ et } m \geq 1\}$$

$$\begin{aligned}
L_2 &= \{ a^{2i+3} b^{2j+2} / i, j \geq 0 \} \\
L_3 &= \{ a^i b^j / i \geq j+1 \} \\
L_4 &= \{ c^n w / w \in \{a, b\}^* \text{ et } |w|=n \} \\
L_5 &= \{ a^{2m} b^{2n} c^{2p} / 2m+n+1=p, n \geq 1 \text{ et } m, p \geq 0 \} \\
L_6 &= \{ a^m b^n c^p / m > n \text{ ou } 2n \leq p \} \\
L_7 &= \{ a^m b^n / m \neq n \} \\
L_8 &= \{ w \in \{a, b, c\}^* / |w|_c = 3p+1, p \geq 0 \} \\
L_9 &= \{ w \in \{a, b, c\}^* / |w|_a + |w|_c \text{ est divisible par } 3 \} \\
L_{10} &= \{ w \in \{0,1\}^* / w \text{ divisible par } 3 \} \\
L_{11} &= \{ wuw^R / w \in \{a, b\}^* \text{ et } u \in \{c\}^* \} \\
L_{12} &= \{ w \in \{a,b\}^* / |w|_a = |w|_b \}
\end{aligned}$$

Exercice 7

Pour chacune des grammaires suivantes, donner son type et le langage qu'elle génère.

1. $G_1 = (\{a, b\}, \{S, A\}, S, P_1)$ où P_1 est

$$S \rightarrow aS / aaA / bb \quad A \rightarrow bbA / \varepsilon$$

2. $G_2 = (\{a, b\}, \{S, A, B\}, S, P_2)$ où P_2 est

$$S \rightarrow aSab / abABab \quad A \rightarrow Ab / \varepsilon \quad B \rightarrow aaB / a$$

3. $G_3 = (\{a, b, c\}, \{S\}, S, P_3)$ où P_3 est

$$S \rightarrow aSa / aSb / c$$

4. $G_4 = (\{a, b\}, \{S, A, B\}, S, P_4)$ où P_4 est

$$S \rightarrow abS / abA \quad A \rightarrow aAB / \varepsilon \quad B \rightarrow bB / b$$

Exercice 8

Donner la grammaire engendrant des programmes du mini langage C de la forme :

```

#include<stdio.h>
int main(void)
{
    int age ;
    float salaire, smig;
    age=21;
    if(x>18)
    {
        printf(" Il est majeur ");
        salaire=20 000 ;
        if (salaire<smig) printf(" Le salaire est insuffisant ") ;
    }
    else {printf("Il est mineur ") ;}
}

```

Les seules instructions autorisées sont l'affectation, la conditionnelle et l'instruction d'impression **printf** simplifiée. L'alphabet terminal comporte : #include, stdio, void, main, <, >, int, float, printf, (,), {, }, if, else, =, " , ; , idf, entier, réel } où idf, entier et réel représentent respectivement un nom de variable, un entier et un réel.