Module Langages Formels
TD 5 : Grammaires et Langages Algébriques

Exercice 1 `There is a fire at the insurance agency.`
Quels sont les langages engendrés par les productions suivantes ?

1. $G_1 : \left\{ \begin{array}{lll} S \rightarrow aSBC \mid aBC & bB \rightarrow bb & CB \rightarrow BC \\ bC \rightarrow bc & aB \rightarrow ab & cC \rightarrow cc \end{array} \right.$

2. $G_2 : \left\{ \begin{array}{lll} S \rightarrow CD & Ab \rightarrow bA & C \rightarrow aCA \mid bCB \\ Ba \rightarrow aB & AD \rightarrow aD & Bb \rightarrow bB \\ BD \rightarrow bD & C \rightarrow \epsilon & Aa \rightarrow aA \\ & D \rightarrow \epsilon & \end{array} \right.$

3. $G_3 : \quad S \rightarrow aS \mid aSbS \mid \epsilon$

Exercice 2 `John has a long mustache.`
Donner des grammaires engendrant les langages suivants.

$$\{a^i b^j c^k,\ i > j\} \qquad \{a^i b^j c^k,\ i \neq j\} \qquad \{a^{2^n},\ n \geq 0\} \qquad \{a^{n^2},\ n \geq 0\}$$

Exercice 3 `Un Langage intrinsèquement ambigu.`
On considère le langage $L = \left\{ a^l b^m c^n \middle| l = m \vee m = n \right\}$.

3.1. Montrer que ce langage est algébrique.

On rappelle le lemme d'OGDEN.

> **Lemme (Ogden)** :
> Soit L un langage algébrique. Il existe un entier N tel que pour tout mot $z \in L$ dans lequel on marque au moins N positions distinctes, il est possible de décomposer z sous la forme $z = uxvyw$ avec
> – x ou y contient au moins une position marquée,
> – xvy contient au plus N positions marquées,
> – pour tout $i \geq 0$, $ux^i vy^i w \in L$.

3.2. Soit G une grammaire reconnaissant L. Montrer qu'il existe $u \in L$ tel qu'il existe deux arbres de dérivation de G disctincts menant à u. Conclure.

► *On pourra considérer les mots $a^N b^N c^{N+N!}$ et $a^{N+N!} b^N c^N$ pour un N bien choisi.*

Exercice 4 `Automates et nombres (suite)`
Nous allons revenir à la notion de transducteur abordée au TD précédent, et continuer à explorer son application aux calculs sur les entiers. Commençons par la définir plus précisément.

Définition :
Soit A un alphabet d'entrée et B un alphabet de sortie. Un transducteur *séquentiel* (gauche) est un automate fini à 2 bandes $\mathcal{A} = (Q, A \times B^*, i, Q, \delta)$ tel que :
- la projection sur la bande d'entrée $((Q, A, i, Q, \delta))$ est un automate fini déterministe ;
- tout état est terminal.

Les étiquettes des flèches d'un transducteur séquentiel sont des couples de $A \times B^*$. Si partant d'un état q on lit la lettre $a \in A$ sur l'entrée pour arriver dans l'état p, on écrit en sortie $v \in B^*$, tel que $q \xrightarrow{(a,v)} p$.

La relation de mots (finis) $R \subseteq A^* \times B^*$ est *réalisée par* $\mathcal{A}$ si R est l'ensemble des étiquettes des chemins finis commençant dans l'état i. En d'autres termes, un couple (f, g) est reconnu par $\mathcal{A}$ s'il existe un état q tel que $i \xrightarrow{(f,g)} q$.

La fonction φ est réalisable par un transducteur fini si la relation $(x, \varphi(x))$ est réalisable par un transducteur fini.

Un transducteur séquentiel droit lit et écrit les mots de droite à gauche.

Un transducteur *sous-séquentiel droit* est un transducteur séquentiel droit qui admet un mode de reconnaissance tordu qui utilise une fonction dite *terminale* $w : Q \to B^*$. On dira alors que le couple (f, g) est reconnu par $\mathcal{A}$ s'il existe un chemin $i \xrightarrow{f/g''} q$ tel que $g = g'g''$ et $w(q) = g'$.

4.1. Donner un transducteur séquentiel qui efface les 0 en tête des mots (sur $\{0, 1\}$).

4.2. Donner un transducteur sous-séquentiel droit qui ajoute 1 à un entier en base 2 (pour l'alphabet d'entrée, prendre $\{0, 1, \square\}$ avec $\square$ qui marque la fin de l'entier).

4.3. Donner un transducteur sous-séquentiel droit qui additionne deux entiers en base 2 (pour l'alphabet d'entrée, prendre $\{0, 1, \square\}^2 \to$ le couple $(10, 3)$ s'écrira $(\square 1010, \square\square\square 11)$ et correspondra au mot $(\square, \square)(1, \square)(0, \square)(1, 1)(0, 1)$).

4.4. Donner un transducteur sous-séquentiel droit pour la soustraction en base 2 (pour l'alphabet d'entrée, prendre $\{0, 1, \square\}^2$).

4.5. Donner un automate déterministe reconnaissant les mots finissant par le motif *abbab*, puis un transducteur séquentiel gauche supprimant toutes les occurrences de *abbab* dans un mot.