CTD 4: Grammaires

BUT 2 – Automates et Langages – R4.A12

Nous avons précédemment donné une méthode de reconnaissance des mots appartenant à un langage (rationnel) à l'aide d'*automates*. Un automate fournit donc une description *analytique* d'un langage. Une autre façon de décrire un langage est de fournir une méthode de construction des mots de ce langage à l'aide d'une *grammaire*. Une grammaire est un ensemble de *règles* qui donnent donc une description *générative* d'un langage. Ces grammaires sont très souvent utilisées (sous une forme dite de *Backus-Naur* ou *BNF*) pour décrire des langages de programmation.

Définition 1 (Grammaire)**.** Une **grammaire** est un quadruplet $G = (V, \Sigma, \mathcal{P}, S)$ où

- V est un alphabet de symboles **non-terminaux** (ou **variables**) (vocabulaire) ;
- Σ est un alphabet de symboles **terminaux** ;
- $\mathcal{P} \subseteq (V \cup \Sigma)^+ \times (V \cup \Sigma)^*$ est un ensemble de **règles** ou **productions** de la forme $u \to v$.
- $S \in V$ est l'**axiome** (à partir duquel les mots sont générés).

Par exemple, une grammaire pour les mots représentant des expressions arithmétiques simples sur des chiffres : $G_0 = \{V, \Sigma, \mathcal{P}, S\}$ où $V = \{S, C\}$, $\Sigma = \{\texttt{+}, \texttt{-}, \texttt{/}, \texttt{*}, \texttt{(}, \texttt{)}, 0, \ldots, 9\}$ et $\mathcal{P} = \{S \to S\texttt{+}S, S \to S\texttt{-}S, S \to S\texttt{/}S, S \to S\texttt{*}S, S \to \texttt{(}S\texttt{)}, S \to C, C \to 0, \ldots, C \to 9\}$. On pourra écrire les règles de façon plus compacte ainsi :

$$\mathcal{P} = \left\{ \begin{array}{lcl} S & \to & S\texttt{+}S \mid S\texttt{-}S \mid S\texttt{/}S \mid S\texttt{*}S \mid \texttt{(}S\texttt{)} \mid C \\ C & \to & 0 \mid \ldots \mid 9 \end{array} \right\}$$

Définition 2 (Dérivation de mots)**.** Soit $G = (V, \Sigma, \mathcal{P}, S)$, $u \in (V \cup \Sigma)^+$ et $v \in (V \cup \Sigma)^*$, alors v peut être **dérivé** de u à partir de G, dénoté $u \underset{G}{\Longrightarrow} v$, ssi $\exists x, y\ V^*$ tels que :

- $u = xu'y$
- $v = xv'y$
- $u' \to v' \in \mathcal{P}$.

On note $\overset{*}{\underset{G}{\Longrightarrow}}$ la clôture transitive et réflexive de $\underset{G}{\Longrightarrow}$. On notera simplement $\Rightarrow$ quand la grammaire est sous-entendue.

Par exemple, à partir de la grammaire G_0, on peut dériver le mot `1+(2*3)` ainsi :

$$S \Rightarrow \underline{S}\texttt{+}S \Rightarrow \underline{C}\texttt{+}S \Rightarrow \texttt{1+}\underline{S} \Rightarrow \texttt{1+(}\underline{S}\texttt{)} \Rightarrow \texttt{1+(}\underline{S}\texttt{*}S\texttt{)} \Rightarrow \texttt{1+(}\underline{C}\texttt{*}S\texttt{)} \Rightarrow \texttt{1+(2*}\underline{S}\texttt{)} \Rightarrow \texttt{1+(2*}\underline{C}\texttt{)} \Rightarrow \texttt{1+(2*3)}$$

Définition 3 (Langage engendré)**.** Le **langage engendré** (ou généré) par une grammaire $G = (V, \Sigma, \mathcal{P}, S)$ est

$$\mathcal{L}(G) = \{w \mid S \overset{*}{\underset{G}{\Longrightarrow}} w\}$$

Définition 4 (Grammaire régulière)**.** Une grammaire est dite **régulière** si toutes ses règles sont de la forme $A \to wB$ ou $A \to w$ où $A, B \in V$ et $w \in \Sigma^*$.

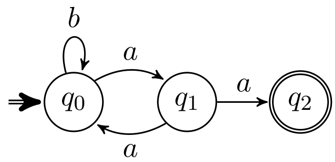
Théorème 1. Un langage est rationnel si et seulement s'il est généré par une grammaire régulière.

Exercice 1. On considère la grammaire régulière $G = (V, \Sigma, \mathcal{P}, S)$ où $V = \{S, A\}$, $\Sigma = \{a, b\}$ et

$$\mathcal{P} = \{S \to aA \mid bA \mid \epsilon, \quad A \to aS \mid bS\}$$

1. Donner une dérivation du mot $abba$.
2. Quel est le langage reconnu par cette grammaire ?
3. À partir de G, donner un automate qui reconnaît $\mathcal{L}(G)$.

Exercice 2. Donner une grammaire régulière qui engendre le langage reconnu par cet automate (non déterministe) :



Exercice 3. On considère l'alphabet $\Sigma_{\mathbb{B}} = \{0, 1\}$.

1. Donner une grammaire régulière qui engendre l'ensemble des mots qui sont des multiples de 2 en base 2.
2. Donner une grammaire régulière qui engendre l'ensemble des mots qui ont la même longueur que leur successeur en base 2.
3. Donner une grammaire régulière qui engendre l'ensemble des mots qui n'ont pas deux chiffres identiques successifs.

Exercice 4. Soient les quatre grammaires :

— $G_1 = (\{S, X, Y\}, \{a, b\}, \{S \to XabY, X \to bX|b, Y \to aY|bY|\epsilon\}, S)$
— $G_2 = (\{S\}, \{a, b\}, \{S \to aS|bS|a|b\}, S)$
— $G_3 = (\{S, X, Y\}, \{a, b\}, \{S \to X|Y, X \to \epsilon, Y \to aY|bY|a|b\}, S)$
— $G_4 = (\{S\}, \{a\}, \{S \to aS|\epsilon\}, S)$

et les quatre langages $L_1 = a^*$, $L_2 = (a + b)^*$, $L_3 = (a + b)^+$, $L_4 = b^*bab(a + b)^*$.

1. Pour chaque grammaire, dire s'il s'agit d'une grammaire régulière.
2. Attribuez à chaque grammaire son langage.

Exercice 5. Soit $\Sigma = \{a, b\}$. Soit L le langage des mots sur Σ dont la première et la dernière lettre sont distinctes.

1. Donnez une expression rationnelle qui représente le langage L.
2. Donnez un AFN qui reconnaît L.
3. Donnez une grammaire régulière qui engendre L.
4. Donnez une grammaire pour L qui ne comporte que deux variables (dont l'axiome).

Exercice 6. Sur l'alphabet $\Sigma_{\mathbb{B}}$, déterminez une grammaire régulière pour chacune des langages suivants :

1. $\{1^p 0^q : p > 0, q > 2\}$
2. $\{0^n 1^n | n \geq 0\}$

Exercice 7. Pour chacun des langages suivants, donner une grammaire (pas nécessairement régulière) qui l'engendre :

1. a^*b
2. $\{a^n b^p \mid n > p \geq 0\}$
3. $\{a^n b^p \mid n \neq p\}$
4. $\{a^n b^p c^n \mid n \geq 0, p \geq 0\}$
5. $\{u \in (a+b)^* \mid u \text{ est un palindrome}\}$