TD 4 – Grammaires algébriques

1 Grammaires algébriques (non contextuelles)

On rappelle qu'étant donné un mot $w \in \Sigma^*$, on définit le *mot renversé de w*, noté $w^{\mathcal{R}}$, comme étant $w^{\mathcal{R}} = \varepsilon$ si $w = \varepsilon$ et $w^{\mathcal{R}} = a_n \cdots a_1$ si $w = a_1 \cdots a_n$ pour $n \in \mathbb{N}_{>0}$.

Exercice 1. Donner une grammaire algébrique pour chacun des langages suivants (sans trop justifier).

1. $\{w \in \{a,b\}^* \mid w = w^{\mathcal{R}}\}$ (langage des palindromes sur $\{a,b\}$).
2. $\{w \in \{a,b\}^* \mid w \neq w^{\mathcal{R}}\}$ (langage des non palindromes sur $\{a,b\}$).
3. $\{w \in \{(,),0,1,\ldots,9,*,+\}^* \mid w$ correspond à une expression arithmétique sur $\mathbb{N}$ valide$\}$.
4. $\{a^i b^j c^k \mid i \neq j \vee j \neq k\}$.
5. $\{w \in \{a,b\}^* \mid |w|_a \geq |w|_b\}$.
6. $\{a^{n_0} b a^{n_1} b \cdots a^{n_k} b \mid k \in \mathbb{N} \wedge \exists j \in \mathbb{N}, n_j \neq j\}$.
7. $\{ww' \in \{a,b\}^* \mid |w| = |w'| \wedge w \neq w'\}$.

Solution 1.

1. $\mathcal{G} = (\{S\}, \{a,b\}, R, S)$ où R contient les règles :
$$S \to aSa \mid bSb \mid a \mid b \mid \varepsilon \ .$$
2. $\mathcal{G} = (\{S,T\}, \{a,b\}, R, S)$ où R contient les règles :
$$\begin{aligned} S &\to aSa \mid bSb \mid aTb \mid bTa \\ T &\to aTa \mid aTb \mid bTa \mid bTb \mid a \mid b \mid \varepsilon \ . \end{aligned}$$
3. $\mathcal{G} = (\{E, N, N'\}, \{(,),0,1,\ldots,9,*,+\}, R, E)$ où R contient les règles :
$$\begin{aligned} E &\to (E) \mid E * E \mid E + E \mid N \\ N &\to 0 \mid 1N' \mid 2N' \mid \cdots \mid 9N' \\ N' &\to 0N' \mid 1N' \mid \cdots \mid 9N' \mid \varepsilon \ . \end{aligned}$$
4. $\mathcal{G} = (\{A,B,C,D,G,S\}, \{a,b,c\}, R, S)$ où R contient les règles :
$$\begin{aligned} S &\to GC \mid AD \\ G &\to aGb \mid aA \mid bB \\ D &\to bDc \mid bB \mid cC \\ A &\to aA \mid \varepsilon \\ B &\to bB \mid \varepsilon \\ C &\to cC \mid \varepsilon \ . \end{aligned}$$
5. $\mathcal{G} = (\{S\}, \{a,b\}, R, S)$ où R contient les règles :
$$S \to aSbS \mid bSaS \mid aS \mid \varepsilon \ .$$

   Il est évident que tout mot $w \in \{a,b\}^*$ tel que $S \Rightarrow^* w$ vérifie $|w|_a \geq |w|_b$.
   Inversement, on montre par récurrence sur la longueur de w que tout mot $w \in \{a,b\}^*$ tel que $|w|_a \geq |w|_b$ vérifie $S \Rightarrow^* w$. Pour cela, on utilise le fait que pour tout mot $w \in \{a,b\}^*$ vérifiant que $|w|_a \geq |w|_b$, on a les deux cas suivants.

— Soit w commence par b et il existe donc un préfixe u de w tel que $|u|_a = |u|_b$ et u termine par a, d'où $w = uv = au'bv$ avec $u = au'b$, $|u'|_a = |u'|_b$ et $|v|_a = |w|_a - |u|_a \geq |w|_b - |u|_a = |w|_b - |u|_b = |v|_b$. On conclut par hypothèse de récurrence sur u' et v.
— Soit w commence par a et il y a alors deux sous-cas. Soit il existe un préfixe u de w tel que $|u|_a = |u|_b$ et u termine par b, et on procède de manière analogue au cas précédent. Soit un tel préfixe n'existe pas, et on peut soit directement conclure si w contient uniquement des a, soit w peut être décomposé comme $w = uv$ où la dernière lettre de u est un b, v contient uniquement des a et $|u|_a \geq |u|_b$. On a donc que $w = au'bv$ avec $|u'|_a \geq |u'|_b$ et $|v|_a \geq |v|_b$ et on peut conclure par hypothèse de récurrence sur u' et v.

6. $\mathcal{G} = (\{A, B, D, G, S\}, \{a, b\}, R, S)$ où R contient les règles :

$$\begin{aligned} S &\to GbD \\ G &\to BGa \mid BD \mid Aa \\ D &\to BD \mid \varepsilon \\ B &\to Ab \\ A &\to aA \mid \varepsilon \ . \end{aligned}$$

On observera ici qu'un mot $w \in \{a, b\}^*$ vérifie :
— $A \Rightarrow^* w$ si et seulement si $w \in \mathcal{L}(a^*)$;
— $B \Rightarrow^* w$ si et seulement si $w \in \mathcal{L}(a^*b)$;
— $D \Rightarrow^* w$ si et seulement si $w \in \mathcal{L}((a^*b)^*)$;
— $G \Rightarrow^* w$ si et seulement s'il existe $k, n \in \mathbb{N}, k \neq n$ tels que $w \in \mathcal{L}\big((a^*b)^k a^n\big)$.

7. $\mathcal{G} = (\{A, B, S\}, \{a, b\}, R, S)$ où R contient les règles :

$$\begin{aligned} S &\to AB \mid BA \\ A &\to aAa \mid aAb \mid bAa \mid bAb \mid a \\ B &\to aBa \mid aBb \mid bBa \mid bBb \mid b \ . \end{aligned}$$

Ceci vient du fait que pour tout alphabet Σ, on a

$$\{ww' \mid w, w' \in \Sigma^* n, |w| = \big|w'\big| \wedge w \neq w'\} = \bigcup_{\substack{k_1, k_2 \in \mathbb{N} \\ a_1, a_2 \in \Sigma, a_1 \neq a_2}} \Sigma^{k_1} a_1 \Sigma^{k_1} \Sigma^{k_2} a_2 \Sigma^{k_2} \ .$$

Exercice 2. Trouver une grammaire algébrique non ambiguë pour le langage

$$\{w \in \{(,), 0, 1, \ldots, 9, *, +\}^* \mid w \text{ correspond à une expression arithmétique sur } \mathbb{N} \text{ valide}\}$$

telle que dans chaque arbre de dérivation (« parse tree »), $*$ a priorité sur $+$.

Solution 2. $\mathcal{G} = (\{S, P, T, N, N'\}, \{(,), 0, 1, \ldots, 9, *, +\}, R, S)$ où R contient les règles :

$$\begin{aligned} S &\to P + S \mid P \\ P &\to T * P \mid T \\ T &\to (S) \mid N \\ N &\to 0 \mid 1N' \mid 2N' \mid \cdots \mid 9N' \\ N' &\to 0N' \mid 1N' \mid \cdots \mid 9N' \mid \varepsilon \ . \end{aligned}$$

Exercice 3. Soit $\mathcal{G} = (\{S\}, \{(,)\}, R, S)$ et $\mathcal{G}' = (\{B, R\}, \{(,)\}, R', B)$ des grammaires algébriques où R contient les règles

$$S \to SS \mid (S) \mid \varepsilon$$

et R' contient les règles

$$B \to (RB \mid \varepsilon$$
$$R \to) \mid (RR\ .$$

Montrer que $\mathcal{G}$ et $\mathcal{G}'$ engendrent le même langage, mais que l'une est ambiguë et pas l'autre.

Solution 3. Les deux engendrent le langage L des mots sur $\{(,)\}$ bien parenthésés. Pour montrer cela, on démontre, d'une part, par récurrence sur $|w|$, que pour tout $w \in L$, $S \Rightarrow^* w$ et $B \Rightarrow^* w$ et que pour tout $w = w'$) tel que $w' \in L$, $R \Rightarrow^* w$). D'autre part, on démontre, par récurrence sur la longueur des dérivations, que pour tout $w \in \{(,)\}^*$, si $S \Rightarrow^* w$, alors $w \in L$, si $B \Rightarrow^* w$, alors $w \in L$, et si $R \Rightarrow^* w$, alors $w = w'$) avec $w' \in L$.

On a que $\mathcal{G}$ est ambiguë, puisqu'il existe deux arbres de dérivation différents pour le mot ()()(). Au contraire, on peut montrer, par récurrence sur la longueur des dérivations, que pour tout $w \in \{(,)\}^*$, si $B \Rightarrow^* w$ ou $R \Rightarrow^* w$, alors il existe un unique arbre de dérivation pour w, en observant que tout mot $w \in L$ est soit vide, soit se décompose de manière unique comme $w = (w_1)w_2$ où $w_1, w_2 \in L$.

2 Langages algébriques (non contextuels) et langages rationnels

Étant donné une grammaire algébrique $\mathcal{G} = (V, \Sigma, R, S)$, on dit qu'elle est *linéaire droite* lorsque dans toute règle $X \to u$ de R, soit $u \in \Sigma^*$, soit $u = u'Y$ avec $u' \in \Sigma^*$ (le corps de chaque règle contient au plus une variable tout à droite).

Exercice 4. Montrer qu'un langage est rationnel si et seulement s'il est engendré par une grammaire linéaire droite.

Solution 4. Soit un AFD $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$. En supposant que $Q \cap \Sigma = \emptyset$, on construit la grammaire algébrique linéaire droite $\mathcal{G} = (Q, \Sigma, R, q_0)$ où R contient :

— pour tous $q \in Q$ et $a \in \Sigma$, la règle $q \to a\delta(q, a)$;

— pour tout $q \in F$, la règle $q \to \varepsilon$.

On a $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{G})$.

Soit $\mathcal{G} = (V, \Sigma, R, S)$ une grammaire algébrique linéaire droite. On construit l'AFN $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ avec, en supposant que $A \notin V$, $Q = (V \cup \{A\}) \times \Sigma^{\leq l}$ où $l \in \mathbb{N}$ est la longueur maximale du corps d'une règle de R (et $\Sigma^{\leq l}$ correspond à l'ensemble de tous les mots sur Σ de longueur au plus l), $q_0 = (S, \varepsilon)$, $F = \{(A, \varepsilon)\}$ et $\delta \colon Q \times (\Sigma \cup \{\varepsilon\}) \to \mathfrak{P}(Q)$ est telle que :

— pour tous $X \in V \cup \{A\}$, $w \in \Sigma^{\leq l}$, $|w| \geq 1$ et $a \in \Sigma$,

$$\delta((X, w), a) = \begin{cases} \{(X, u)\} & \text{si } w = au \text{ avec } u \in \Sigma^* \\ \emptyset & \text{sinon} \end{cases}$$

et $\delta((X, w), \varepsilon) = \emptyset$;

— pour tout $a \in \Sigma$, $\delta((A, \varepsilon), a) = \emptyset$ et $\delta((A, \varepsilon), \varepsilon) = \emptyset$;

— pour tous $X \in V$ et $a \in \Sigma$,

$$\delta((X, \varepsilon), a) = \{(Y, u) \mid Y \in V, u \in \Sigma^*, (X \to auY) \in R\} \cup \{(A, u) \mid u \in \Sigma^*, (X \to au) \in R\}$$

$$\text{et } \delta((X, \varepsilon), \varepsilon) = \begin{cases} \{(A, \varepsilon)\} & \text{si } (X \to \varepsilon) \in R \\ \emptyset & \text{sinon} \end{cases}.$$

Exercice 5. Soit la grammaire algébrique $\mathcal{G} = (\{S\}, \{a, b, c\}, R, S)$ où R contient les règles $S \to aSSb \mid c$. Montrer que tout langage rationnel inclus dans $\mathcal{L}(\mathcal{G})$ est fini.

Solution 5. Nous allons montrer que pour tout mot $w \in \{a,b\}^*$ tel que $S \Rightarrow^* w$, on a $|w|_a = |w|_b$ et que pour tout $n \in \mathbb{N}_{>0}$ vérifiant $|w| \geq 2^n - 1$, il existe un préfixe u de w tel que $|u|_a - |u|_b = n - 1$. Montrons-le par récurrence sur $|w|$.

Cas de base $|w| = 1$**.** Alors nécessairement $w = c$ et on conclut trivialement.

Récurrence. Soit $l \in \mathbb{N}, l \geq 2$ et supposons que pour tout mot $w \in \{a,b\}^*$ tel que $|w| < l$ et $S \Rightarrow^* w$, on a $|w|_a = |w|_b$ et que pour tout $n \in \mathbb{N}_{>0}$ vérifiant $|w| \geq 2^n - 1$, il existe un préfixe u de w tel que $|u|_a - |u|_b = n - 1$.

Soit $w \in \{a,b\}^*$ tel que $|w| = l$ et $S \Rightarrow^* w$. Alors w est nécessairement de la forme $w = aw_1w_2b$ où $S \Rightarrow^* w_1$ et $S \Rightarrow^* w_2$. Par hypothèse de récurrence, puisque $|w_1|_a = |w_1|_b$ et $|w_2|_a = |w_2|_b$, on a $|w|_a = |w|_b$.

Soit maintenant $n \in \mathbb{N}_{>0}$ tel que $|w| \geq 2^n - 1$. Si $n = 1$, on a que le préfixe $u = \varepsilon$ de w vérifie $|u|_a - |u|_b = 0$. Autrement, on a nécessairement que $|w_1| \geq 2^{n-1} - 1$ ou $|w_2| \geq 2^{n-1} - 1$. Si $|w_1| \geq 2^{n-1} - 1$, alors, par hypothèse de récurrence, il existe un préfixe u_1 de w_1 tel que $|u_1|_a - |u_1|_b = n - 2$; $u = au_1$ est donc un préfixe de w tel que $|u|_a - |u|_b = n - 1$. Si $|w_2| \geq 2^{n-1} - 1$, alors, par hypothèse de récurrence, il existe un préfixe u_2 de w_2 tel que $|u_2|_a - |u_2|_b = n - 2$; $u = aw_1u_2$ est donc un préfixe de w tel que $|u|_a - |u|_b = n - 1$ (puisque $|w_1|_a = |w_1|_b$).

Soit maintenant un langage $L \subseteq \mathcal{L}(\mathcal{G})$ infini. Il existe donc une suite $(w_n)_{n\geq 1}$ de mots de L telle que $|w_n| \geq 2^n - 1$ pour tout $n \in \mathbb{N}_{>0}$. Par ce que nous venons de montrer, pour chaque $n \in \mathbb{N}_{>0}$, on a donc qu'il existe une décomposition $w_n = u_nv_n$ de w_n telle que $|u_n|_a - |u_n|_b = |v_n|_b - |v_n|_a = n - 1$. Pour tous $n, m \in \mathbb{N}_{>0}, n \neq m$, on a donc que $u_n^{-1}L \neq u_m^{-1}L$, puisque $u_nv_n = w_n \in L$ alors que $u_mv_n \notin L$, en observant que $|u_m|_a + |v_n|_a - |u_m|_b - |v_n|_b = m - 1 - (n - 1) = m - n \neq 0$, ce qui prouve, par ce que nous venons de montrer, que u_mv_n ne peut appartenir à $\mathcal{L}(\mathcal{G})$ et donc à L. Par conséquent, L a un nombre infini de quotients à gauche, et ne peut donc être rationnel.

En conclusion, tout langage rationnel inclus dans $\mathcal{L}(\mathcal{G})$ doit être fini.

3 Forme normale de Chomsky

On rappelle qu'une grammaire algébrique $\mathcal{G} = (V, \Sigma, R, S)$ est en forme normale de Chomsky (FNC) quand toutes les règles de R sont de la forme :

$$A \to BC \quad \text{ou} \quad A \to a \quad \text{ou} \quad S \to \varepsilon$$

avec $A \in V$, $B, C \in V \setminus \{S\}$ et $a \in \Sigma$.

Exercice 6. Proposer une grammaire algébrique en FNC équivalente à la grammaire algébrique $\mathcal{G} = (\{B, C, S\}, \{a, b\}, R, S)$ où R contient les règles :

$$\begin{aligned} S &\to CSC \mid aB \\ C &\to B \mid S \\ B &\to b \mid \varepsilon \ . \end{aligned}$$

Solution 6. En appliquant l'algorithme du cours (et en regroupant les symboles redondants), on trouve la grammaire algébrique $\mathcal{G}' = (\{A, B, C, D, S, S_0\}, \{a, b\}, R', S_0)$ où R' contient les règles :

$$\begin{aligned} S_0 &\to CD \mid AB \mid a \mid SC \mid CS \\ S &\to CD \mid AB \mid a \mid SC \mid CS \\ C &\to b \mid CD \mid AB \mid a \mid SC \mid CS \\ D &\to SC \\ A &\to a \\ B &\to b \ . \end{aligned}$$

Exercice 7. Proposer un algorithme en temps polynomial (en la taille cumulée du mot et de la grammaire) qui reconnaît si un mot appartient au langage engendré par une grammaire algébrique en FNC.

Solution 7. Étant donné un alphabet Σ, soit $w \in \Sigma^*$ et $\mathcal{G} = (V, \Sigma, R, S)$ une grammaire algébrique en FNC.

Si $w = \varepsilon$, alors il suffit de vérifier si la règle $S \to \varepsilon$ appartient à R ou non.

Autrement, $w = a_1 \cdots a_n$ pour $n \in \mathbb{N}_{>0}$ et $a_1, \ldots, a_n \in \Sigma$, et donc pour tous $i, j \in [\![1, n]\!], i \leq j$, on va calculer l'ensemble $T_{i,j}$ des variables $X \in V$ telles que $X \Rightarrow^* a_i \cdots a_j$. Une fois ceci fait, il suffit de vérifier si S appartient à $T_{1,n}$ ou non.

Pour tous $i, j \in [\![1, n]\!], i \leq j$, on a

$$T_{i,j} = \begin{cases} \{X \in V \mid (X \to a_i) \in R\} & \text{si } i = j \\ \bigcup_{l \in [\![i, j-1]\!]} \{X \in V \mid (X \to X_1 X_2) \in R, X_1 \in T_{i,l}, X_2 \in T_{l+1,j}\} & \text{sinon} \end{cases}.$$

On voit donc assez facilement qu'on peut calculer les $T_{i,j}$ en temps polynomial en la taille cumulée de w et $\mathcal{G}$.