TD4 14 février 2018

Langages Formels

Exercice 1 (Grammaires)
Montrer que les langages suivants sont algébriques :

1. $\{\ u\widetilde{u}\ :\ u \in \{a,b\}^*\ \}$ où $\widetilde{u}$ est l'image miroir
2. $\{\ a^n b^m\ :\ n \neq m,\ n,m \in \mathbb{N}\ \}$
3. $\{\ a^n b^p c^q\ :\ n,q \geq 0,\ p \geq n+q\ \}$
4. $\{\ a^n b^p\ :\ 0 \leq n,\ n \leq p \leq 2n\ \}$

Exercice 2 (Langages de Dyck)
Soit $\Sigma_n = \{\ a_i\ :\ 1 \leq i \leq n\ \} \cup \{\ \overline{a}_i\ :\ 1 \leq i \leq n\ \}$ un alphabet vu comme n paires de parenthèses. Soit $G_n = (N_n, \Sigma_n, P_n, S)$ la grammaire définie par $S \to a_1 S \overline{a}_1 S \mid \ldots \mid a_n S \overline{a}_n S \mid \epsilon$. Le langage $D_n^* = \mathcal{L}_{G_n}(S)$ est appelé langage de Dyck sur n paires de parenthèses.

1. Montrer que
$$D_1^* = \{\ w \in \Sigma_1^*\ :\ |w|_{a_1} = |w|_{\overline{a}_1} \text{ et } |v|_{a_1} \geq |v|_{\overline{a}_1} \text{ pour tout } v \leq w\ \}.$$
2. On considère le système de réécriture $R_n = (\Sigma_n, P_n')$ dont les règles sont $\{\ (a_i\overline{a}_i, \epsilon)\ :\ 1 \leq i \leq n\ \}$. Montrer que
$$D_n^* = \{\ w \in \Sigma_n^*\ :\ w \to_{R_n}^* \epsilon\ \}.$$
3. Soit Γ un alphabet disjoint de Σ_n, $\Sigma = \Sigma_n \cup \Gamma$, et $L \subset \Sigma^*$ un langage. On définit la clôture $\mathrm{clot}(L) = \{v \in \Sigma^* \mid \exists w \in L,\ w \xrightarrow{*} v \text{ dans } R_n\}$. Montrer que si L est reconnaissable, alors $\mathrm{clot}(L)$ l'est aussi.
On définit la réduction $\mathrm{red}(L) = \{v \in \mathrm{clot}(L) \mid v \not\to \text{ dans } R_n\}$. Montrer que si L est reconnaissable, alors $\mathrm{red}(L)$ l'est aussi.

Exercice 3 (Propriétés de fermeture)
Montrer que la famille des langages algébriques est fermée par :

1. Concaténation ;

2. Itération ;
3. Union ;
4. Image miroir ;
5. substitution algébrique [1] et morphisme.

Exercice 4 (Problème du vide)
Montrer que le problème du vide d'une grammaire algébrique est Ptime-dur, par réduction de Horn-SAT.

Exercice 5 (Langages rationnels et grammaires linéaires)
Soit $G = (N, \Sigma, P, S)$ une grammaire algébrique. On dit que G est

— *linéaire* quand $P \subseteq N \times (\Sigma^* \cup \Sigma^* N \Sigma^*)$
— *linéaire gauche* quand $P \subseteq N \times (\Sigma^* \cup N \Sigma^*)$
— *linéaire droite* quand $P \subseteq N \times (\Sigma^* \cup \Sigma^* N)$.

Montrer qu'un langage est rationnel ssi il peut être engendré par une grammaire linéaire gauche (ou droite).

Exercice 6 (Grammaires ambiguës)

1. Montrer que la grammaire suivante est ambiguë :

$$\begin{array}{lcl} S & \rightarrow & \mathsf{if}\ c\ \mathsf{then}\ S\ \mathsf{else}\ S \\ S & \rightarrow & \mathsf{if}\ c\ \mathsf{then}\ S \\ S & \rightarrow & a \end{array}$$

2. Voici un extrait de la syntaxe d'Algol 60, qui évite l'ambiguïté ci-dessus grâce à la distinction entre ⟨*unconditional statement*⟩ et

1. Une substitution $\sigma : \Sigma \to 2^{\Sigma'^*}$ telle que pour tout $a \in \Sigma$, $\sigma(a)$ est algébrique.

$\langle$*conditional statement*$\rangle$:

$$
\begin{aligned}
\langle \textit{statement} \rangle &\to \langle \textit{unconditional statement} \rangle \\
&\mid \langle \textit{conditional statement} \rangle \\
\langle \textit{unconditional statement} \rangle &\to \langle \textit{for statement} \rangle \\
\\
\langle \textit{conditional statement} \rangle &\to \langle \textit{if statement} \rangle \\
&\mid \langle \textit{if statement} \rangle \ \texttt{else} \ \langle \textit{statement} \rangle \\
\langle \textit{if statement} \rangle &\to \langle \textit{if clause} \rangle \ \langle \textit{unconditional statement} \rangle \\
\langle \textit{if clause} \rangle &\to \texttt{if} \ \langle \textit{boolean expression} \rangle \ \texttt{then} \\
\\
\langle \textit{for statement} \rangle &\to \langle \textit{for clause} \rangle \ \langle \textit{statement} \rangle \\
\langle \textit{for clause} \rangle &\to \texttt{for} \ \langle \textit{variable} \rangle \ := \ \langle \textit{for list} \rangle \ \texttt{do}
\end{aligned}
$$

Montrer que cette grammaire reste néanmoins ambiguë.

3. Donner une grammaire non ambiguë équivalente.