TD 8 : Grammaires algébriques

Exercice 1 (Complexité)**.** On définit la *taille* d'une grammaire algébrique $G = \langle N, \Sigma, P, S\rangle$ comme

$$|G| = \max(\sum_{A\to\alpha\in P} |A\alpha|, |N \cup \Sigma|) \ .$$

1. Une grammaire algébrique est sous *forme quadratique* si pour toute règle $A \to \alpha$ de P, $|\alpha| \leq 2$. Montrer que, pour toute grammaire algébrique G, on peut construire une grammaire algébrique G' équivalente sous forme quadratique en temps $O(|G|)$.
2. Un symbole non terminal A de N est *utile* s'il existe une dérivation de la forme

   $$S \Rightarrow^* \alpha A\beta \Rightarrow^* w$$

   avec α et β des chaînes de symboles dans $(N \uplus \Sigma)^*$ et w un chaîne de symboles dans Σ^*.

   Montrer que pour toute grammaire algébrique G, on peut calculer l'ensemble de ses symboles non terminaux utiles en temps $O(|G|)$. En déduire un algorithme pour construire une grammaire *réduite* équivalente à G en temps linéaire.
3. En déduire une borne de complexité du calcul d'une grammaire algébrique équivalente où les règles $A \to \alpha$ de P vérifient $|\alpha| \geq 1$ (sauf potentiellement pour une règle $S \to \varepsilon$, mais alors on demande aussi $\alpha \in (\Sigma \cup N\backslash\{S\})^*$).
4. Quelle borne de complexité pouvez-vous donner pour la suppression des règles de la forme $A \to B$ avec A et B dans N ? Pour la mise sous forme normale de Chomsky ?

Exercice 2 (Théorème de Chomsky et Schützenberger)**.** Le théorème dit qu'un langage L est algébrique si et seulement s'il existe un entier n, un langage rationnel R et un morphisme π tels que $L = \pi(D_n^* \cap R)$, où D_n^* dénote l'ensemble des mots bien parenthésés sur un alphabet à n paires de parenthèses.

L'intuition derrière ce théorème est que l'on peut séparer les aspects de structure (le parenthésage, c'est-à-dire la structure d'arbre) et de contrôle (le langage rationnel) d'un langage algébrique – ce dont on verra une autre interprétation avec les automates à pile.

1. Soit G une grammaire algébrique sur Σ. Proposer une grammaire algébrique G', qui explicite la structure des dérivations de G au moyen d'un alphabet Σ_n de n sortes de parenthèses, telle que

   $$L(G') \subseteq D_n^* \text{ et } L(G) = \pi(L(G'))$$

   avec π une projection de $\Sigma_n^* \to \Sigma^*$.

2. Il faut maintenant trouver un langage rationnel R tel que

$$L(G') = D_n^* \cap R \ .$$

   En fait, il existe même un langage *local* (cf. feuille de TD 1, exercice 5) qui remplit ce rôle. Proposer un tel langage.

3. Montrer qu'il existe un morphisme μ de $\Sigma_n \to \Sigma_2$ tel que

$$D_n^* = \mu^{-1}(D_2^*) \ .$$

   En déduire une autre formulation du théorème.

Exercice 3 (Intersection avec un langage reconnaissable et algorithme de Cocke, Kasami et Younger)**.**

1. Soit G une grammaire algébrique et A un automate fini. Contruire G' algébrique telle que $L(G') = L(G) \cap L(A)$.
2. Modifier la construction pour calculer *au vol* si $L(G') = \emptyset$.
3. Quelle est la complexité de la construction d'une grammaire algébrique pour le langage $L(G) \cap \{w\}$?

   Un problème avec cet algorithme est qu'il génère en général beaucoup de règles inutiles. On voudrait filtrer la génération des règles de la grammaire pour n'ajouter une règle avec un non terminal de la forme $[q, A, q']$ (où q et q' sont des états de l'automate reconnaissant w, et A un non terminal de G) que s'il existe réellement un facteur u de w dans $L_G(A) \cap L_{q,q'}$.

   Proposer un algorithme avec cet invariant. Quelle borne de complexité obtenez-vous ?