

Soit $\mathcal{G} = \langle \mathcal{A}, \mathcal{V}, \mathcal{P} \rangle$ une grammaire algébrique avec \mathcal{A} l'alphabet des symboles terminaux, \mathcal{V} l'ensemble des variables (*i.e.* les symboles non-terminaux) et \mathcal{P} les règles de production.

La grammaire \mathcal{G} est *pré-réduite* si :

- (1) aucun non-terminal n'engendre le langage vide : $\forall T \in \mathcal{V}, L_{\mathcal{G}}(T) \neq \emptyset$.

La grammaire \mathcal{G} est *réduite vis-à-vis d'un non-terminal* $T \in \mathcal{V}$ si elle est pré-réduite et si :

- (2) tous les non-terminaux peuvent apparaître par dérivation à partir de T :

$$\forall S \in \mathcal{V}, \exists u_1, u_2 \in (\mathcal{A} \cup \mathcal{V})^* \text{ vérifiant } u_1 S u_2 \in \hat{L}_{\mathcal{G}}(T)$$

Pour réduire \mathcal{G} , on peut définir :

- (1) $U_0 = \mathcal{A}$, $U_i = U_{i-1} \cup \{S \in \mathcal{V} : \exists m \in U_{i-1}^*, S \rightarrow m \in \mathcal{P}\}$ et $U = (\bigcup_{i \geq 0} U_i) \setminus \mathcal{A}$ l'ensemble des variables *productives* ;
 (2) $W_0 = \{T\}$, $W_i = W_{i-1} \cup \{S \in \mathcal{V} : \exists S$

2 Réduire vis-à-vis de S_0 , puis rendre propre les grammaires suivantes :

$$1. \begin{cases} S_0 \rightarrow S_1 S_2 \mid S_3 S_4 \mid S_5 \\ S_1 \rightarrow S_1 S_1 \mid S_1 S_4 \\ S_2 \rightarrow a S_2 \mid S_3 \\ S_3 \rightarrow S_2 \mid S_4 \mid S_5 a S_3 \\ S_4 \rightarrow c S_4 \mid \varepsilon \\ S_5 \rightarrow S_4 \mid b \end{cases} \quad 2. \begin{cases} S_0 \rightarrow S_0 S_1 \mid S_1 \mid S_2 \\ S_1 \rightarrow b S_1 \mid a S_3 S_5 \mid \varepsilon \\ S_2 \rightarrow b S_2 \mid a b S_4 \mid \varepsilon \\ S_3 \rightarrow b S_1 \mid S_2 S_3 \\ S_4 \rightarrow a S_5 S_2 \mid b S_1 S_6 S_7 \\ S_5 \rightarrow a S_5 \mid b S_5 S_6 \\ S_6 \rightarrow a S_6 b \mid a S_4 \\ S_7 \rightarrow S_5 S_2 \mid a S_7 S_6 \end{cases} \quad 3. \begin{cases} S_0 \rightarrow S_1 \mid S_0 a S_2 \mid b S_2 S_3 \mid \varepsilon \\ S_1 \rightarrow S_1 b S_0 \mid S_3 a S_1 \\ S_2 \rightarrow S_0 b S_0 \mid S_2 S_3 a \mid a S_1 S_3 \\ S_3 \rightarrow S_1 S_2 \mid S_3 b S_2 \mid a S_0 b S_3 \end{cases}$$

La grammaire $\mathcal{G} = \langle \mathcal{A}, \mathcal{V}, \mathcal{P} \rangle$ est sous forme normale de Chomsky (ou sous forme normale quadratique) si on a $\mathcal{P} \subseteq \mathcal{V} \times \mathcal{V}\mathcal{V} \cup \mathcal{V} \times \mathcal{A}$.

À partir d'une grammaire \mathcal{G} propre, décomposer les règles de productions ayant strictement plus de deux symboles : remplacer les règles du type $A \rightarrow u_1 u_2 \cdots u_k$ avec $u_i \in \mathcal{V} \cup \mathcal{A}$ et $k \geq 3$ par $A \rightarrow u_1 A_1, A_1 \rightarrow u_2 A_2, \dots, A_{k-2} \rightarrow u_{k-1} u_k$ où les A_i sont de nouvelles variables.

On obtient alors une grammaire sous forme normale de Chomsky en introduisant une nouvelle variable T_a avec la règle $T_a \rightarrow a$ pour chaque lettre terminale a et en remplaçant par T_a chaque occurrence de a dans tout membre droit de longueur 2.

3 Mettre les grammaires suivantes sous forme normale de Chomsky :

$$1. \begin{cases} S \rightarrow aS \mid Sb \mid c \\ S \rightarrow A \\ A \rightarrow b \mid c \end{cases} \quad 2. \begin{cases} S \rightarrow A \\ A \rightarrow b \mid c \end{cases} \quad 3. \begin{cases} S \rightarrow abcdef \\ S \rightarrow aS \mid Ab \\ A \rightarrow cB \\ B \rightarrow dA \mid e \end{cases} \quad 4. \begin{cases} S \rightarrow abcdef \\ S \rightarrow aS \mid Ab \\ A \rightarrow cB \\ B \rightarrow dA \mid e \end{cases} \quad 5. \begin{cases} S \rightarrow aSAb \mid bSS \mid d \\ A \rightarrow cASaA \mid bcd \end{cases} \quad 6. \begin{cases} S \rightarrow x \mid lSr \mid SpS \mid SmS \end{cases}$$

La grammaire $\mathcal{G} = \langle \mathcal{A}, \mathcal{V}, \mathcal{P} \rangle$ est sous forme normale de Greibach (resp. presque-Greibach) si on a $\mathcal{P} \subseteq \mathcal{V} \times \mathcal{A}\mathcal{V}^*$ (resp. $\mathcal{P} \subseteq \mathcal{V} \times \mathcal{A}(\mathcal{A} \cup \mathcal{V})^*$).

Pour $\mathcal{V} = \{S_1, \dots, S_r\}$, on pose $\mathcal{V}' = \{S'_1, \dots, S'_r\}$. Pour \mathcal{G} propre, on construit une suite $\mathcal{G} = \mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{2r+1}$ de grammaires équivalentes (avec \mathcal{G}_{2r+1} presque-Greibach) :

- \mathcal{G}_{2i} s'obtient en remplaçant dans \mathcal{G}_{2i-1} les règles $S_i \rightarrow S_i m_1 \mid \dots \mid S_i m_k \mid w_1 \mid \dots \mid w_p$ par

$$\begin{cases} S_i \rightarrow w_1 S'_i \mid \dots \mid w_p S'_i \mid w_1 \mid \dots \mid w_p \\ S'_i \rightarrow m_1 S'_i \mid \dots \mid m_k S'_i \mid m_1 \mid \dots \mid m_k \end{cases}$$

- \mathcal{G}_{2i+1} s'obtient en remplaçant dans toute règle de \mathcal{G}_{2i} de la forme $S \rightarrow S_j m$ avec $j \leq i$, cette occurrence de S_j par les membres droits issus de S dans \mathcal{G}_{2i} .

On obtient alors une grammaire sous forme normale de Greibach en introduisant une nouvelle variable T_a avec la règle $T_a \rightarrow a$ pour chaque lettre terminale a et en remplaçant par T_a chaque occurrence de a non initiale d'un membre droit.

4 Mettre les grammaires suivantes sous forme normale de Greibach :

$$1. \begin{cases} S_1 \rightarrow S_2 S_3 \\ S_2 \rightarrow S_1 S_2 \mid a \\ S_3 \rightarrow b \end{cases} \quad 2. \begin{cases} S_1 \rightarrow S_2 S_3 \\ S_2 \rightarrow S_1 S_2 \mid a \\ S_3 \rightarrow S_3 S_1 \mid b \end{cases} \quad 3. \begin{cases} S \rightarrow ST \mid a \\ T \rightarrow TS \mid b \end{cases} \quad 4. \begin{cases} S \rightarrow SaT \mid TT \mid b \\ T \rightarrow Td \mid T Sa \mid aS \mid c \end{cases} \quad 5. \begin{cases} S_1 \rightarrow S_1 S_1 \mid S_1 S_2 \mid S_3 S_2 \mid S_2 b \\ S_2 \rightarrow S_2 S_1 \mid S_1 S_2 \mid S_3 S_3 \mid S_1 a \\ S_3 \rightarrow S_1 S_3 \mid S_2 S_3 \mid S_3 S_1 \mid a \end{cases}$$