

Notes du module de grammaires lexicalisées

Hugo Mougard

1 Grammaires lexicalisées

Une grammaire lexicalisée est une grammaire dont toutes les règles contiennent au moins un terminal.

2 Forme normale de Greibach

Concerne les langages propres (sans ϵ) dont les grammaires (propres elles aussi, ou réduites) sont sous la forme

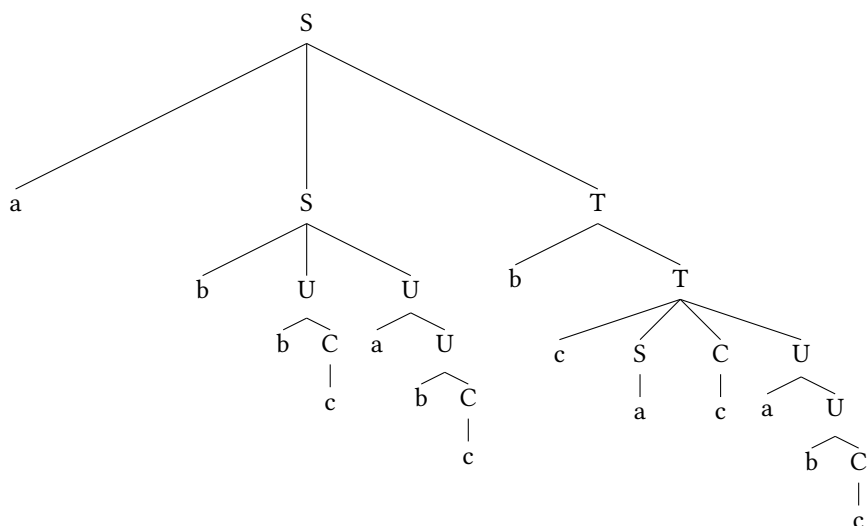
$$V \rightarrow XV^*$$

où X est terminal et V non-terminal.

Par exemple, voici une grammaire lexicale :

$$\begin{aligned} S &\rightarrow aST \mid bUU \mid a \\ T &\rightarrow bT \mid cSCU \mid cTU \\ U &\rightarrow aU \mid bC \\ C &\rightarrow c \end{aligned}$$

Et l'arbre de parsing de la phrase `abbcabcabcacabc` en utilisant cette grammaire :



On peut aussi « inverser » la grammaire pour la centrer sur les terminaux :

$a \rightarrow S$
 $aST \rightarrow S$
 $aU \rightarrow U$
 $bUU \rightarrow S$
 $bT \rightarrow T$
 $bC \rightarrow U$
 $cSCU \rightarrow T$
 $cTU \rightarrow T$
 $c \rightarrow C$

2.1 Traduction en automate à piles :

En utilisant la grammaire inversée ci-dessus, on peut produire le tableau ci-dessous :

	S	T	U	C
a	ϵ		U	
	ST			
b	UU	T	C	
c	SCU			ϵ
	TU			

Le tableau se lit, par exemple pour la case (a, S) : on peut dépiler S si on lit a et qu'on empile ϵ ou T puis S.

Exemple d'exécution :

bande : abb, pile S
 → lit a
 → bande : bb, dépile S, empile ST, pile ST
 → lit b
 → bande : b, dépile S, empile UU, pile UUT
 → lit b
 → bande : #, dépile U, empile C, pile CUT

2.2 Transformation en forme normale de Greibach :

Tout d'abord, il faut noter quels sont les problèmes :

- les règles produisant ϵ ne sont pas tolérées (on vise une grammaire propre)
- les règles récursives à gauche ne sont pas tolérées

Voici maintenant l'algorithme pour transformer les règles récursives à gauche en règles récursives à droite :

Pour tout V_i non terminal, on introduit V_i' puis, on transforme tout règle de la forme :

$$V_j \rightarrow V_i m_1 \mid V_i m_2 \mid \dots \mid V_i m_p \mid w_1 \mid w_2 \mid \dots \mid w_q$$

en

$$V_j \rightarrow w_1 V_i' \mid w_2 V_i' \mid \dots \mid w_q V_i'$$

et

$$V_i' \rightarrow m_1 V_i' \mid m_2 V_i' \mid \dots \mid m_p V_i' \mid m_1 \mid m_2 \mid \dots \mid m_p$$

Exemple :

$A \rightarrow BC$
 $B \rightarrow AB \mid a$
 $C \rightarrow AC \mid b$