TD3 : Transformation de grammaires

29 septembre 2015

1. Soit le langage L_1 sur le vocabulaire $V = \{l', homme, ours, qui, a, vu\}$ formé de l'ensemble des phrases finies de la forme *l'homme qui a vu l'ours*, *l'homme qui a vu l'homme qui a vu l'ours*, *l'homme qui a vu l'homme qui a vu ... qui a vu l'ours*.
   Donner une grammaire algébrique (*context-free*) engendrant L_1.

2. Transformer en forme normale de Greibach la grammaire suivante (qui est déjà non récursive gauche) : $S \to Aa \mid b$
   $A \to bdC \mid cC$
   $C \to cC \mid adC \mid c \mid ad$

   Solution: La grammaire de départ est déjà non récursive gauche. On obtient l'ordre partiel $S < A$. Arbitrairement on le rend total par exemple avec $S < A < C$.
   La règle $S \to Aa$ est changée en $S \to bdCa|cCa$ La suite est triviale : il ne reste plus de règles dont la partie droite commence par un non terminal. Les terminaux en position non initiale de partie droite sont changés en non terminaux équivalents (i.e. produisant le terminal équivalent seul : $W \to w$).
   1 $S \to bDCA' \mid cCA' \mid b$
   $D \to d$
   $A' \to a$
   2 $A \to bDC \mid cC$
   3 $C \to cC \mid aDC \mid c \mid aD$

3. Appliquer l'algorithme de dé-récursivation (gauche) vu en cours à la grammaire suivante, grammaire de la liste :
   $S \to (\ L\ ) \mid a$
   $L \to L, S \mid S$

   Solution: On a une seule règle récursive gauche (directe) et aucune indirecte. À la place du couple règle récursive gauche $L \to L, S$ plus règle de terminaison de la récursion $L \to S$, on commence par dériver la terminaison par S avec la nouvelle règle $L \to SL'$ puis c'est L' qui capte la récursion $L' \to, SL'$ ou bien termine $L' \to \varepsilon$.

4. Soit la grammaire suivante : $S \longrightarrow aSbS \mid bSaS \mid \varepsilon$.
   (a) Donner l'algorithme permettant de rendre ε-libre une grammaire algébrique quelconque.

   Solution:
   1. On construit $\mathcal{A} = \{A \in V / A \xrightarrow{+} \varepsilon\}$.
   2. On supprime toutes les règles $A \to \varepsilon$.

3. Pour toute règle $B \longrightarrow \alpha_0 A_0 \alpha_1 A_1 ... \alpha_n$ où les A_i appartiennent à l'ensemble $\mathcal{A}$, on remplace la règle par toutes les règles $B \longrightarrow \sigma(\alpha_0 A_0 \alpha_1 A_1 ... \alpha_n)$
où $\sigma(m)$ est l'ensemble de toutes les combinaisons où les A_i sont remplacés ou non par ε.
4. Si $S \in \mathcal{A}$, ajouter S' dans V, et $S' \longrightarrow S \mid \varepsilon$ dans P, et remplacer l'axiome S par S'.

(b) Appliquer l'algorithme à la grammaire donnée ci-dessus.

Solution:
$$\begin{array}{lcl} S' & \rightarrow & S \mid \varepsilon \\ S & \rightarrow & aSbS \mid bSaS \mid ab \mid ba \mid aSb \mid abS \mid bSa \mid baS \end{array}$$

5. Transformer la grammaire suivante en grammaire sans règles singulières :
$S \rightarrow AB \mid A \; ; \; A \rightarrow aB \mid bA \mid aSb \; ; \; B \rightarrow S \mid b$

Solution: $S \rightarrow AB \mid aB \mid bA \mid aSb \; ; \; A \rightarrow aB \mid bA \mid aSb \; ; \; B \rightarrow AB \mid aB \mid bA \mid aSb \mid b$

6. Enlever la récursivité gauche de la grammaire suivante :
$E \rightarrow Ea \mid b$

Solution:
$$\begin{array}{lcl} E & \rightarrow & bE' \\ E' & \rightarrow & aE' \mid \varepsilon \end{array}$$

7. Mettre sous forme normale de Chomsky la grammaire définie par les règles de production suivantes :
$S \rightarrow AB \mid aS \mid a$
$A \rightarrow Ab \mid \varepsilon$
$B \rightarrow AS$

Solution: Il faut commencer par rendre la grammaire ε-libre, ce qui est assez simple dans ce cas particulier, puisque seul A est susceptible de s'effacer :

	$S' \rightarrow S \mid \varepsilon$	$S' \rightarrow S \mid \varepsilon$
$S \rightarrow AB \mid aS \mid a$	$S \rightarrow AB \mid (\varepsilon)B \mid aS \mid a$	$S \rightarrow AB \mid B \mid aS \mid a$
$A \rightarrow Ab \mid \varepsilon$	$A \rightarrow Ab \mid (\varepsilon)b$	$A \rightarrow Ab \mid b$
$B \rightarrow AS$	$B \rightarrow AS \mid (\varepsilon)S$	$B \rightarrow AS \mid S$

Il faut ensuite débarrasser la grammaire des productions singulières, qui sont exclues dans une grammaire quadratique.
L'algorithme général est le suivant :

1. Partir d'une grammaire ε-libre.
2. Pour tout $A \in V$, construire $N_A = \{B \in V \; / \; A \stackrel{*}{\longrightarrow} B\}$
3. Soit P l'ensemble initial de règles ; on construit P' le nouvel ensemble de productions de la manière suivante :
   (a) Ajouter dans P' toutes les règles non singulières de P.
   (b) Pour toute règle $B \longrightarrow \alpha$ de P' (non singulière par hypothèse), pour tout $A \in V$ tel que $B \in N_A$, ajouter $A \longrightarrow \alpha$ à P'.

L'application systématique de l'algorithme à notre grammaire donne :

$S' \to S \mid \varepsilon$	$N_S = \{B, S\}$	$S \to AB$ non sing. ; et $S \in N_B$ donc on ajoute $B \to AB$
$S \to AB \mid aS \mid a$	$N_A = \emptyset$	$S \to aS$ non sing. ; et $S \in N_B$ donc on ajoute $B \to aS$
$A \to Ab \mid b$	$N_B = \{B, S\}$	$S \to a$ non sing. ; et $S \in N_B$ donc on ajoute $B \to a$
$B \to AS$		$B \to AS$ non sing. ; et $B \in N_S$ donc on ajoute $S \to AS$

Cela donne la grammaire suivante, qu'il est alors simple de rendre quadratique :

$S' \to S \mid \varepsilon$	$S' \to S \mid \varepsilon$
$S \to AS \mid AB \mid aS \mid a$	$S \to AS \mid AB \mid XS \mid a$
$A \to Ab \mid b$	$X \to a$
$B \to AS \mid AB \mid aS \mid a$	$A \to AY \mid b$
	$Y \to b$
	$B \to AS \mid AB \mid XS \mid a$

Il y a deux cycles imbriqués dans la grammaire ε-libre ($S \xrightarrow{*} B \xrightarrow{*} S$), ce qui explique qu'*in fine* S et B engendrent le même langage. Cela suggère une simplication de la grammaire, mais ce n'était pas demandé ici.

À propos de la simplication : dans le cas particulier de cette grammaire, il aurait été possible de se contenter de la grammaire :

$S' \to S \mid \varepsilon$	... mais cela tient au fait que B n'est atteint que par S, dans la règle
$S \to AS \mid AB \mid aS \mid a$	$A \to AB$, et que par conséquent tout ce qu'on perd en supprimant
$A \to Ab \mid b$	la dérivation $B \to S$ est « rattrapé » par la dérivation $S \to AS$.
$B \to AS$	

8. Proposer une grammaire en forme de Greibach équivalente à la grammaire suivante :

$S \longrightarrow aSB \mid BA$
$A \longrightarrow Sbc \mid Ac$
$B \longrightarrow bSB \mid b$

Solution: $S \longrightarrow aSB \mid BA$
$A \longrightarrow Sbc \mid Ac$
$B \longrightarrow bSB \mid b$

Pas d'ε-production, pas de récursivité indirecte, pas de productions singulère. Il faut juste traiter la récursivité directe de A (version de dérécursivation sans ε). Cela donne :

$S \longrightarrow aSB \mid BA$
$A \longrightarrow Sbc \mid SbcA'$
$A' \longrightarrow cA' \mid c$
$B \longrightarrow bSB \mid b$

Résultat final après suppression de la récursivité gauche (directe) :

$X \longrightarrow b$
$Y \longrightarrow c$
$S \longrightarrow aSB \mid bSBA \mid bA$
$A \longrightarrow aSBXY \mid bSBAXY \mid bXY \mid aSBXYA' \mid bSBAXYA' \mid bXYA'$
$A' \longrightarrow cA' \mid c$
$B \longrightarrow bSB \mid b$