Elément de correction du TD 6 Math-Info

Année 2001-2002

1 Exercice-1

Soit L l'ensemble des mots sur $\{a, b\}$ contenant un b de plus que de a.

i). Montrer que pour $w \in L$ avec $w \neq b$, il existe des mots u et v dans L vérifiant une des égalités $w = auv$, $w = uav$, $w = uva$.

ii). En déduire que L est engendré par la grammaire de productions $S \to aSS|SaS|SSa|b$.

Correction :

i). Soit w un mot de L de longueur au moins 3. Trois cas sont possibles.

- ☞ Cas-1 : Le mot w commence par a, soit $w = aw'$. Alors le mot w' a deux b de plus que de a : il doit donc exister un préfixe u de w' qui a un b de plus que de a, autrement dit un préfixe u dans L. Alors, si w' est uv, v est aussi dans L, et on a $w = auv$.
- ☞ Cas-2 : Le mot w finit par a, soit $w = w'a$. Alors le mot w' a deux b de plus que de a, il a un préfixe u qui a un b de plus que de a, et, si w' est uv, alors on a $w = uva$, et u et v sont dans L.
- ☞ Cas-3 : Le mot w commence et finit par b. Alors on a $w = bw'b$, où w' contient un a de plus que de b. Soit u' le plus long préfixe de w' qui contient autant de a que de b : alors w' s'écrit $u'av'$, où v' aussi contient autant de a que de b. On a alors $w = uav$, où u est bu' et v est $v'b$, et u et v sont dans L par construction.

ii). On montre par induction sur la longueur d'une dérivation que, si S dérive u en p étapes, alors on a $|u|_b + |u|_S = |u|_a + 1$. Pour $p = 0$, on a $u = S$, et le résultat est vrai.

Sinon, soit v le mot précédent u dans une dérivation de u à partir de S : par hypothèse de récurrence, on a $|v|_b + |v|_S = |v|_a + 1$. Or, quelle que soit la production de G utilisée pour passer de v à u, soit elle ajoute un a et deux S, soit elle remplace un S par un b, donc l'égalité pour v entraîne l'égalité pour u. On a donc $L(G) \subseteq L$.

Inversement, on montre par induction sur la longueur de w que $w \in L$ entraîne $w \in L(G)$. Pour w de longueur 0 ou 1, l'implication est évidente. Sinon, d'après i)., il existe u et v dans L vérifiant $w = auv$, $w = uav$, ou $w = uva$. Par hypothèse d'induction, G dérive u et v à partir de S ; on obtient alors dans chaque cas une dérivation pour w, soit respectivement :

$S \vdash_G aSS \vdash_G^* auS \vdash_G^* auv = w$

$S \vdash_G SaS \vdash_G^* uaS \vdash_G^* uav = w$

$S \vdash_G SSa \vdash_G^* uSa \vdash_G^* uva = w$.

Donc on a $L \subseteq L(G)$ et donc finalement $L = L(G)$.

---

2 Exercice-2

Soit Σ un alphabet quelconque, et a une lettre fixée de Σ. Pour w un mot dans Σ^*, on note w^R le mot obtenu en recopiant w de droite à gauche. Construire une grammaire qui engendre le langage $\{waw^R; w \in \Sigma^*\}$.

Correction :
$S \to xSx|a$ avec $x \in \Sigma$.

---

3 Exercice-3

Considérer la grammaire G suivante :
$Axiome = S$
$N = \{S, A, B, C\}$
$T = \{a, b\}$
$P = \{S \to aAa, A \to Sb|bBB, B \to abb|aC, C \to aCA\}$
i). Metter la grammaire G sous forme normale de Chomsky.
ii). Transformer cette même grammaire G afin qu'elle soit sous forme normale de Greibach.

Correction :
i). Pour mettre la grammaire G sous forme normale de Chomsky, on introduit les variables X_a et X_b et les productions $X_a \to a$ et $X_b \to b$.
On traite ensuite les productions dont le membre de droite est de longueur supérieur à 2.
La grammaire G' est sous forme normale de Chomsky :
$S \to X_aY_1$
$X_a \to a$
$X_b \to b$
$Y_1 \to AX_a$
$B \to X_aY_2|X_aC$
$Y_2 \to X_bX_b$
$C \to X_aY_3$
$Y_3 \to CA$
$A \to SX_b|X_bY_4$
$Y_4 \to BB.$
Avec $N' = \{S, A, B, C, X_a, X_b, Y_1, Y_2, Y_3, Y_4\}$
ii). Pour mettre la grammaire G sous forme normale de Greibach, on applique l'algorithme vu en cours sur la grammaire G'.
On obtient la grammaire G'' :
$S \to aY_1$
$X_a \to a$
$X_b \to b$
$A \to aY_1X_b|bY_4$
$Y_1 \to aY_1X_bX_a|bY_4X_a$
$B \to aY_2|aC$
$Y_4 \to aY_2B|aCB.$
$Y_2 \to bX_b$

$C \rightarrow aY_3$
$Y_3 \rightarrow aY_3A$

---

4 Exercice-4

Donner des automates à piles reconnaissant les langages suivants :

☞ $L_1 = \{u \in \{a,b\}^*, |u|_a = |u|_b\}$;

☞ $L_2 = \{u \in \{a,b\}^*, |u|_a \geq |u|_b\}$;

☞ $L_3 = \{u \in \{a,b\}^*, |u|_a = 2|u|_b\}$;

Correction :

Pour L_1 : l'automate suivant reconnaît le langage par pile vide : $M = (q, \{a,b\}, \{a,b,\#\}, \delta, q, \#)$ avec :

$$\begin{array}{l}
\delta(q,a,\#) = (a, a\#) \\
\delta(q,b,\#) = (q, b\#) \\
\delta(q,a,a = (q, aa) \text{ et son symétrique en } b \\
\delta(q,b,a) = (q,\epsilon) \text{ et son symétrique en } b \\
\delta(q,\epsilon,\#) = (q,\epsilon)
\end{array}$$

Pour L_2 : idem sauf qu'on rajoute la règle $(q,\epsilon,a) = (q,\epsilon)$. L'automate n'est alors plus déterministe.

Pour L_3 : il faut empiler deux fois plus de b que de a. Alors on prend $Q = \{q_0, q_1\}$, $F = \emptyset$, $\Sigma = \{a,b\}$, $\Gamma = \{A,B\}$ et la fonction de transition est la suivante :

$$\begin{array}{l}
\delta(q_0,\epsilon,\gamma_0) = (q_0,\epsilon) \\
\delta(q_0,a,\gamma_0) = (q_0, A\gamma_0) \\
\delta(q_0,b,\gamma_0) = (q_0, BB\gamma_0) \\
\delta(q_0,a,A) = (q_0, AA) \\
\delta(q_0,b,B) = (q_0, BBB) \\
\delta(q_0,a,B) = (q_0,\epsilon) \\
\delta(q_0,b,A) = (q_1,\epsilon) \\
\delta(q_1,\epsilon,A) = (q_1, A) \\
\delta(q_1,\epsilon,\gamma_0) = (q_0, B\gamma_0)
\end{array}$$
