ENSIIE - Cours Langages et Systèmes Formels 1

$\boxed{\textbf{TD 2}}$

Frédéric Blanqui

Exercice 39 Soit G la grammaire définie par les règles suivantes:
$S \to ACA$
$A \to aAaD|B|C$
$B \to bB|b$
$C \to cS|\varepsilon$
$D \to \varepsilon$

(a) Calculez l'ensemble des non-terminaux effaçables.

(b) Donnez une grammaire équivalente sans règles de la forme $X \to \varepsilon$ sauf $S \to \varepsilon$ si $\varepsilon \in L(G)$.

(c) Eliminez les règles de la forme $X \to Y$ où X, Y sont des non-terminaux.

(d) En déduire une grammaire propre équivalente.

Solution.

(a) C et D sont effaçables car $C \to \varepsilon$ et $D \to \varepsilon$ sont des règles de G. Donc A est effaçable également car $A \to C \in G$ et C est effaçable. Donc S est effaçable également car $S \to ACA$ et A et C sont effaçables. B n'est pas effaçable car tout mot généré à partir de B commence par b. Donc, les non-terminaux effaçables sont $\{C, D, A, S\}$.

(b) Pour chaque règle $X \to u$, on rajoute toutes les règles $X \to u'$ où u' est obtenu en effaçant certains non-terminaux effaçables (si u a n non-terminaux effaçables, cela génère 2^n règles), et on efface toutes les règles de la forme $X \to \varepsilon$. On obtient G_1':

$S \to ACA|CA|AA|AC|A|C$
$A \to aAaD|B|C|aaD|aAa|aa$
$B \to bB|b$
$C \to cS|c$

G_1' n'est pas tout-à-fait équivalente à G car $\varepsilon \in L(G)$ et $\varepsilon \notin L(G_1')$. Pour obtenir une grammaire équivalente, il faut rajouter la règle $S \to \varepsilon$.

Remarque: D est improductif dans G_1'; on pourrait donc l'éliminer également.

(c) Pour chaque non-terminal X, on calcule $\mathcal{R}(X) = \{Y \mid X \to^* Y\} - \{X\}$, l'ensemble des non-terminaux différents de X vers lesquels X se réécrit, par itération.

	A	B	C	D	S
1	BC				AC
2	BC				ABC

Pour chaque non-terminal X, réduit $Y \in \mathcal{R}(X) - \{X\}$, règle $Y \to u$, on rajoute la règle $X \to u$:

$S \to ACA|CA|AA|AC|A|C|aAaD|B|C|aaD|aAa|aa|bB|b|cS|c$
$A \to aAaD|B|C|aaD|aAa|aa|bB|b|cS|c$
$B \to bB|b$
$C \to cS|c$

Enfin, on élimine toutes les règles unaires, obtenant ainsi G_2':

$S \to ACA|CA|AA|AC|aAaD|aaD|aAa|aa|bB|b|cS|c$
$A \to aAaD|aaD|aAa|aa|bB|b|cS|c$
$B \to bB|b$
$C \to cS|c$

G est équivalent à $G_2' \cup \{S \to \varepsilon\}$.

(d) Pour obtenir une grammaire propre, il faut encore que le non-terminal de départ n'apparaisse dans aucun membre droit de règle. Pour cela, on prend un nouveau non-terminal de départ T et on rajoute les règles $T \to \varepsilon$ et, pour chaque règle $S \to u$, la règle $T \to u$:

$T \to \varepsilon|ACA|CA|AA|AC|aAaD|aaD|aAa|aa|bB|b|cS|c$
$S \to ACA|CA|AA|AC|aAaD|aaD|aAa|aa|bB|b|cS|c$
$A \to aAaD|aaD|aAa|aa|bB|b|cS|c$
$B \to bB|b$
$C \to cS|c$ ■

Exercice 40 Nous allons calculer une grammaire réduite équivalente à:
$S \to AC|BS|B$
$A \to aA|aF$
$B \to CF|b$
$C \to cC|D$
$D \to aD|BD|C$
$E \to aA|BSA$
$F \to bB|b$

(a) Calculez l'ensemble des non-terminaux productifs et éliminez les règles contenant un non-terminal improductif.

(b) Calculez les non-terminaux accessibles et éliminez les règles contenant un non-terminal inaccessible.

Solution.

(a) Les non-terminaux sont A, B, C, D, E, F, S. B est productif car $B \to b$. F est productif car $F \to b$. Donc S est productif car $S \to B$, et A est productif car $A \to aF$. Donc E est productif car $E \to aA$. Par contre, C et D sont improductifs. En éliminant les règles contenant l'un d'eux, on obtient:

$S \to BS|B$
$A \to aA|aF$
$B \to b$
$E \to aA|BSA$
$F \to bB|b$

(b) A partir de S, on peut avoir B. A partir de B, aucun autre non-terminal. Donc, seuls S et B sont accessible. A, E, F sont inaccessibles. En éliminant les règles contenant l'un d'eux, on obtient:

$S \to BS|B$
$B \to b$

Exercice 11 Une grammaire $(\Sigma, \mathcal{N}, \mathcal{R}, S)$ est en forme normale de Chomsky (FNC) si elle est propre et que toute règle autre que $S \to \varepsilon$ est de la forme $X \to a$ ou $X \to YZ$ avec $a \in \Sigma$ et $Y, Z \in \mathcal{N}$.

(a) Montrer que toute grammaire propre peut être mise en FNC.

(b) Mettre la grammaire suivante en FNC:

$S \to SaB|aB$
$B \to bB|\varepsilon$

Solution.

(a) Pour chaque lettre $a \in \Sigma$, on rajoute un nouveau non-terminal X_a et la règle $X_a \to a$. On obtient ainsi un ensemble $\mathcal{R}_1$ de règles. Dans chaque règle de $\mathcal{R}_1$ de la forme $X \to x_1 \dots x_n$ où $n \geq 2$ et $x_i \in \Sigma \cup \mathcal{N}$, on remplace chaque $x_i \in \Sigma$ par X_{x_i}. On obtient ainsi un ensemble $\mathcal{R}_2$ de règles. Enfin, dans $\mathcal{R}_2$, on remplace chaque règle de la forme $X \to X_1 \dots X_n$ avec $n \geq 3$ par $\{X \to X_1Y_1\} \cup \{Y_i \to X_{i+1}Y_{i+1} \mid i \in \{1, \dots, n-3\}\} \cup \{Y_{n-2} \to X_{n-1}X_n\}$ où $Y_1, \dots, Y_{n-2}$ sont de nouveaux symboles. Soit G' la grammaire ainsi obtenue.

Une grammaire G est propre si elle ne contient aucune règle de la forme $X \to Y$ et, soit $\varepsilon \notin L(G)$ et G ne contient aucune règle de la forme $X \to \varepsilon$, soit $\varepsilon \in L(G)$ et $S \to \varepsilon$ est la seule règle de cette forme et aucun membre droit de règle ne contient S. Les transformations précédentes n'introduisent aucune règle de la forme $X \to Y$, $X \to \varepsilon$ ou $X \to r$ avec S dans r. Donc, G' est propre.

Reste à montrer que G et G' sont équivalentes. Nous avons $L(G) \subseteq L(G')$ car $\to_G \subseteq \to_{G'}^+$. Enfin, on peut montrer que, pour tout k et $u \in \Sigma^*$, si $S \to_{G'}^k u$, alors $S \to_G^* u$, par induction sur k.

(b) On commence par construire une grammaire propre équivalente. Pour cela, on commence par calculer les non-terminaux effaçables (B), rajouter les règles qu'on obtient en effaçant les non-terminaux effaçables, et éliminer les ε-règles:

$S \to SaB|Sa|aB|a$
$B \to bB|b$

Il n'y a pas de règles unaires à éliminer. Enfin, comme S a plusieurs occurrences, on prend un nouveau symbole de départ T avec les règles:

$T \to SaB|Sa|aB|a$
$S \to SaB|Sa|aB|a$
$B \to bB|b$

La grammaire obtenue est propre. Maintenant, on applique la méthode précédente:

$T \to SU|SX_a|X_aB|a$
$U \to X_aB$
$S \to SU|SX_a|X_aB|a$
$B \to X_bB|b$
$X_a \to a$
$X_b \to b$ ■

Exercice 38 Soit $\Sigma = \{(,)\}$ l'alphabet dont les lettres sont la parenthèse ouvrante et la parenthèse fermante. Soit A le langage engendré par les règles $S \to (S) \mid SS \mid \varepsilon$, et soit B le langage engendré par les règles $S \to (S)S \mid \varepsilon$.

(a) Montrez que $B \subseteq A$.

(b) Montrez que B est stable par concaténation: si $u, v \in B$ alors $uv \in B$.

(c) En déduire que $A \subseteq B$.

Solution.

(a) B est généré par deux règles: $S \to_B \varepsilon$ et $S \to_B (S)S$. La première règle est aussi une règle de A. La seconde peut-être simulé par l'application de plusieurs règles de A: $S \to_A SS \to_A (S)S$. Autrement dit, toute $\to_B$ dérivation peut-être traduite en une $\to_A$ dérivation. Ainsi, $B \subseteq A$.

(b) Soient $u, v \in B$. Alors, $S \to_B^* u$ et $S \to_B^* v$. Raisonnons par récurrence sur la longueur $n \geq 1$ de la dérivation de S à u. Si $n = 1$ alors $u = \varepsilon$ et, en effet, $uv = v \in B$. Sinon, $S \to_B (S)S \to_B^* u$. Donc, $u = (x)y$ avec $S \to_B^* x$ et $S \to_B^* y$. Par hypothèse de récurrence sur y, $S \to_B^* yv$. Donc, $S \to_B^* uv$.

(c) Montrons que, si $S \to_A^n u$, alors $S \to_B^* u$ par récurrence sur n. Si $n = 0$, c'est immédiat. On raisonne ensuite par cas sur la première règle appliquée:

- $S \to_A \varepsilon$. Immédiat.
- $S \to_A (S)$. Alors, $u = (x)$ avec $S \to^{n-1} x$. Par hypothèse de récurrence, $S \to_B^* x$. Donc, $u \in B$ puisque $S \to_B (S)S \to_B^* (x)S \to_B (x)\varepsilon = u$.
- $S \to_A SS$. Alors, $u = xy$ avec $S \to_A^p x$, $S \to_A^q y$ et $p + q = n - 1$. Par hypothèse de récurrence, $S \to_B^* x$ et $S \to_B^* y$. Donc, $xy \in B$ car B est clos par concaténation. ■

Exercice 41 Montrez que $L = \{(a^m b^n)^2 \mid m, n \in \mathbb{N}\}$ n'est pas algébrique.

Exercice 44 Trouvez un automate reconnaissant le langage 0^*1^*.

Exercice 43 (a) Donnez un automate reconnaissant les entiers relatifs (en base 10) avec, au début éventuellement, un signe + pour les entiers positifs ou nuls et des zéros inutiles. Ainsi, par exemple, +02 doit être accepté.

(b) Donnez un automate reconnaissant les nombres en virgule flottante avec exposants (e.g. +020.13e+4).

Exercice 6 L'interprétation dans $\mathbb{N}$ d'un mot u sur l'alphabet $\Sigma = \{0, 1\}$, $\overline{u}$, est définie par récurrence sur la longueur de u ainsi: $\overline{\varepsilon} = 0$, $\overline{u0} = 2\overline{u}$ et $\overline{u1} = 2\overline{u} + 1$. Montrer que $\{u \in \Sigma^* \mid \exists k, \overline{u} = 3k\}$ est reconnaissable par un automate.