Automates Avancés
Travaux Dirigés n°4

Une *grammaire hors-contexte* est un quadruplet $G = (V, T, P, S)$, où V et T sont deux ensembles finis disjoints (appelés respectivement variables (ou non-terminaux) et terminaux), P est un ensemble de productions de la forme $A \to \alpha$, où $A \in V$ et $\alpha \in (V \cup T)^*$ et S est le symbole de départ. Soit $\Sigma = V \cup T$.

Nous définissons deux relations $\Rightarrow$ et $\Rightarrow_G^*$ entre suites de Σ^* comme suit: Si $A \to \beta$ est une production de P et $\alpha, \gamma \in \Sigma^*$, $\alpha A \gamma \Rightarrow \alpha\beta\gamma$. La suite $\alpha A\gamma$ est un *prédécesseur immédiat* de $\alpha\beta\gamma$ et $\alpha\beta\gamma$ est un *successeur immédiat* de $\alpha A\gamma$. La relation $\Rightarrow^*$ est alors la fermeture réflexive et transitive de $\Rightarrow$. Si $\alpha \Rightarrow^* \beta$, alors α est un prédécesseur de β et β est un successeur de α. Étant donné $L \subseteq \Sigma^*$, nous définissons $pre(L) = \{\alpha \in \Sigma^* \mid \exists \beta \in L \text{ avec } \alpha \Rightarrow \beta\}$ et $post(L) = \{\beta \in \Sigma^* \mid \exists \alpha \in L \text{ avec } \alpha \Rightarrow \beta\}$. $pre^i(L)$ est inductivement défini par $pre^0(L) = L$ et $pre^{i+1}(L) = pre(pre^i(L))$. Enfin, nous définissons $pre^*(L) = \bigcup_{i \geq 0} pre^i(L)$ ce qui est le même que $\{\alpha \in \Sigma^* \mid \exists \beta \in L \text{ avec } \alpha \Rightarrow^* \beta\}$. De même, $post^*(L) = \{\beta \in \Sigma^* \mid \exists \alpha \in L \text{ avec } \alpha \Rightarrow^* \beta\}$. Le langage généré par G est donné par $L(G) = post^*(\{S\}) \cap T^*$.

▸ Exercice 1.
Soit G donnée par les productions $S \to ASB \mid \epsilon$, $A \to a$ et $B \to b$.

- Quelle est le langage généré par G ?
- Quel est l'image de Parikh de $L(G)$? Donnez un automate fini avec le même image de Parikh.
- Donnez $post^*(S) \cap V^*$. Est-ce que c'est régulier ?

Un algorithme simple pour calculer $pre^*(L)$ d'un langage régulier donné par un automate M est donné ci-dessous. Il est basé sur la saturation de la relation de transition d'un automate non-déterministe $M = (Q, \Sigma, \delta, q_0, F)$, où $\delta \subseteq Q \times \Sigma \times Q$. Nous définissons la relation de transition étendue $\widehat{\delta} : (Q \times \Sigma^*) \to 2^Q$ par: $\widehat{\delta}(q, \epsilon) = \{q\}$, $\widehat{\delta}(q, a) = \{q' \mid (q, a, q') \in \delta\}$ et $\widehat{\delta}(q, wa) = \{p \mid p \in \widehat{\delta}(r, a) \text{ pour un } r \in \widehat{\delta}(q, w)\}$

Entrée: $G = (V, T, P, S), M = (Q, \Sigma, \delta, q_0, F)$, **Sortie**: δ_{pre^*}
$rel \leftarrow \delta$
Répéter
  pour $q, q' \in Q, A \to \beta \in P$ **faire**
    si $q' \in \widehat{rel}(q, \beta)$ **alors** $rel \leftarrow rel \cup \{(q, A, q')\}$
jusqu'à ce que rel ne change plus
retourner rel

▸ Exercice 2.
Considérez la grammaire de l'exercice 1.

- Calculer $pre^*(\{aab\})$. Est-ce que $w \in L(G)$?

- Calculer $pre^*(\{aabb\})$. Est-ce que $w \in L(G)$?
- Est-ce qu'on peut donner un algorithme pour calculer $post^*(L)$ qui est similaire à celui pour pre^* ?

▸ Exercice 3.

Soit $G = (V, T, P, S)$ une grammaire hors-contexte. Une variable $X \in V$ est *utile* s'il y a une dérivation $S \Rightarrow^* \alpha X \beta \Rightarrow^* w$ pour $\alpha, \beta \in (V \cup T)^*$ et $w \in T^*$. Une variable $X \in V$ est *effaçable* s'il y a une dérivation $X \Rightarrow^* \epsilon$.

- Donnez un algorithme basé sur pre^* pour calculer les variables utiles d'une grammaire.
- Donnez un algorithme basé sur pre^* pour calculer les variables effaçables d'une grammaire.

▸ Exercice 4.

Considérez la grammaire suivante:

$$\begin{array}{l} S \rightarrow AB \mid CA \mid BD \\ B \rightarrow BC \mid AB \\ A \rightarrow a \\ C \rightarrow aB \mid b \\ D \rightarrow d \end{array}$$

- Calculez les variables utiles en utilisant l'algorithme de l'exercice 3.

Considérez la grammaire suivante:

$$\begin{array}{l} S \rightarrow aA \mid bB \\ A \rightarrow BC \mid BD \mid EE \mid BDE \\ B \rightarrow \epsilon \mid aB \mid D \\ C \rightarrow cD \mid dE \\ D \rightarrow c \mid dE \mid BB \\ E \rightarrow \epsilon \mid aE \mid Eb \end{array}$$

- Calculez les variables effaçables en utilisant l'algorithme de l'exercice 3.

▸ Exercice 5.

Le problème de l'appartenance pour une grammaire est de déterminer si un mot $w \in T^*$ est généré par une grammaire G.

- Donnez un algorithme basé sur pre^* pour le problème de l'appartenance.
- Donnez un algorithme basé sur pre^* pour tester si le langage généré par une grammaire G est inclu dans un langage régulier L. Appliquez cet algorithme pour tester si le langage de la grammaire de l'exercice 1 est inclu dans le langage a^*b^* (dans le langage $(ab)^*$).

▸ Exercice 6.

Soit M un automate à pile donné par $M = (Q, \Sigma, \Gamma, q_0, Z_0, F, \delta)$ avec $Q = \{q_0, q_1\}$, $\Sigma = \{a, b\}$, $\Gamma = \{A, Z_0\}$, $F = \{q_2\}$ et

$$\begin{array}{l} \delta(q_0, a, Z_0) = \{(q_0, AZ_0)\}, \delta(q_0, b, A) = \{(q_1, AAA)\} \\ \delta(q_0, \epsilon, Z_0) = \{(q_1, Z_0)\}, \delta(q_1, b, A) = \{(q_1, \epsilon)\}, \delta(q_1, \epsilon, Z_0) = \{(q_2, \epsilon)\} \end{array}$$

- Quel est le langage généré par M ?
- Calculer $pre^*(q_2)$.