TD2 25 janvier 2017

Langages Formels

Exercice 1 (Construction d'automates, ou pas)
Pour chacun des langages suivants sur $\Sigma = \{0, 1\}$, dire s'il est régulier (reconnaissable par un automate fini) ou pas. Justifier en donnant un automate ou un argument.

1. $\{\ u\ :\ |u|_0 = |u|_1\ \}$ où $|u|_a$ est le nombre d'occurrences de a dans u ;
2. $\{\ 0^p 1^q\ :\ p \geq q\ \}$;
3. $\{\ 0^p 1^q\ :\ p \geq q,\ q \leq 1337\ \}$;
4. $\{\ u\ :\ \overline{u}^2 = 0 \mod 3\}$ où $\overline{u}^2$ est l'interprétation de u en binaire, avec le bit de poids fort à droite.

Exercice 2 (Dérivées partielles, automate d'Antimirov)
La dérivée partielle est aux expressions rationnelles ce que le résiduel est aux langages : on va définir une opération $\partial_a(E)$ qui va correspondre (modulo interprétation) à $a^{-1}\mathcal{L}(E)$.

Formellement, pour une expression E et une lettre a, on définit l'*ensemble d'expressions* $\partial_a(E)$ comme suit, où l'opération de concaténation est naturellement étendue sur les ensembles d'expressions :

$$
\begin{array}{rcl}
\partial_a(\underline{\emptyset}) & = & \partial_a(\underline{\epsilon}) \ = \ \emptyset \\
\partial_a(\underline{b}) & = & \left\{ \begin{array}{ll} \emptyset & \text{si } a \neq b \\ \{\underline{\epsilon}\} & \text{sinon} \end{array} \right. \\
\partial_a(E + E') & = & \partial_a(E) \cup \partial_a(E') \\
\partial_a(E^*) & = & \partial_a(E) \cdot \{E^*\} \\
\partial_a(E \cdot E') & = & \left\{ \begin{array}{ll} \partial_a(E) \cdot \{E'\} & \text{si } \epsilon \notin E \\ (\partial_a(E) \cdot \{E'\}) \cup \partial_a(E') & \text{sinon} \end{array} \right.
\end{array}
$$

On définit ensuite $\partial_w(E)$ pour un mot w quelconque en posant $\partial_\epsilon(E) = \{E\}$ et $\partial_{wa}(E) = \partial_a(\partial_w(E))$, où $\partial_w(S) = \bigcup_{E \in S} \partial_w(E)$ quand S est un ensemble d'expressions.

1. Calculer les dérivées partielles de $(ab+b)^*ba$ par a et b.
2. Montrer que $\mathcal{L}(\partial_w(E)) = w^{-1}\mathcal{L}(E)$.
3. Utiliser cette construction pour traduire une expression rationnelle en un automate acceptant le même langage, en ne s'inquiétant pas de la finitude de cet automate pour l'instant.
4. On définit l'ensemble des suffixes propres d'un mot :

$$\mathrm{Suf}(w) = \{\ v \in \Sigma^+\ :\ \exists u, w = uv\ \}$$

   Montrer les résultats suivants pour $w \in \Sigma^+$:

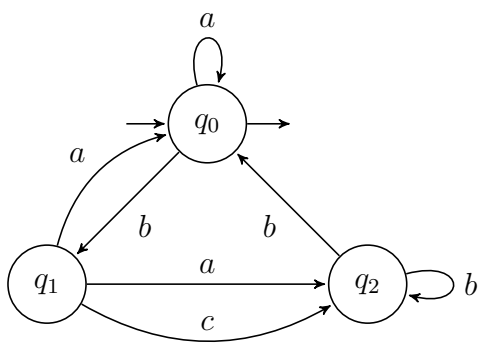
$$\begin{aligned}
\partial_w(E+E') &= \partial_w(E) \cup \partial_w(E') \\
\partial_w(E \cdot E') &\subseteq (\partial_w(E) \cdot E') \cup \bigcup_{v \in \mathrm{Suf}(w)} \partial_v(E') \\
\partial_w(E^*) &\subseteq \bigcup_{v \in \mathrm{Suf}(w)} \partial_v(E) \cdot E^*
\end{aligned}$$

5. Soit $\|E\|$ le nombre d'occurrences de lettres de Σ dans E. Montrer que l'ensemble des dérivées partielles différentes de E contient au plus $\|E\| + 1$ éléments.

Exercice 3 (Algorithme de Brzozowski-McCluskey)
L'objectif de cet exercice est de traduire un automate fini en une expression rationnelle. Nous allons procéder par transformations successives de l'automate, en utilisant une généralisation du type d'automate considéré : la relation de transition sera un sous-ensemble de $Q \times 2^{\Sigma^*} \times Q$. Une exécution d'un tel automate reconnait la concaténation des langages des transitions, et le langage reconnu par l'automate est l'union de ces exécutions.

1. Montrer que tout automate généralisé est équivalent à un automate généralisé pour lequel il existe exactement une transition entre chaque paire d'états : $q' \in \delta(q, L)$ et $q' \in \delta(q, L')$ implique $L = L'$.
2. Soit un automate généralisé $\mathcal{A}$ avec un unique état initial i, un unique état final f. Soit $q \in Q_{\mathcal{A}}$, $q \notin \{i, f\}$. Montrer qu'il existe un automate équivalent à $\mathcal{A}$ avec pour ensemble d'états $Q_{\mathcal{A}} \setminus \{q\}$.
3. En déduire que si L est reconnu par un automate généralisé $\mathcal{A}$, alors L appartient à la clôture rationnelle des étiquettes des transitions de $\mathcal{A}$.

4. Appliquer la construction pour calculer une expression rationnelle correspondant à l'automate suivant :



5. On considère l'alphabet $\Sigma_n = \{1, \ldots, n\} \times \{1, \ldots, n\}$, et on définit :

$$L = \{\ (a_1, a_2)(a_2, a_3) \ldots (a_m, a_{m+1})\ :\ m \geq 1,\ a_1 = 1,\ a_{m+1} = n\ \}$$

   (a) Donner un automate de taille quadratique reconnaissant L.

   (b) Quelle est la taille de l'expression obtenue par la construction précédente sur cet automate ?

Exercice 4 (Langage local, automate de Glushkov)

1. Une expression rationnelle E sur Σ est *linéaire* si chaque symbole de Σ apparaît au plus une fois dans E.

   Montrer que tout langage rationnel sur Σ est le résultat de l'application d'un morphisme alphabétique $\Delta \to \Sigma$ au langage d'une expression rationnelle linéaire sur Δ.

2. Soit L un langage sur Σ. On définit les ensembles

$$\begin{aligned}
P(L) &= \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\} && \text{(1-préfixes)}\\
S(L) &= \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\} && \text{(1-suffixes)}\\
F(L) &= \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\} && \text{(2-facteurs)}\\
N(L) &= \Sigma^2 \backslash F(L) && \text{(2-non facteurs)}
\end{aligned}$$

   Soit E une expression rationnelle telle que $\mathcal{L}(E) = L$. Donner un algorithme pour calculer $P(L)$, $S(L)$ et $F(L)$ à partir de E.

3. Un langage L sur Σ est *local* si

$$L\backslash\{\varepsilon\} = (P(L)\Sigma^* \cap \Sigma^* S(L))\backslash\Sigma^* N(L)\Sigma^* \ .$$

   Un automate fini déterministe est *local* si, pour chaque symbole a de Σ, il y a au plus un état accessible par une transition sur a : $|\{q' \in Q \mid \exists q \in Q, (q, a, q') \in \delta\}| =\leq 1$. Il est *standard* si son état initial n'a aucune transition entrante.

   Montrer que les trois propositions suivantes sont équivalentes :

   (a) L est un langage local.

   (b) L est reconnu par un automate local standard.

   (c) L est reconnu par un automate local.

4. Les langages locaux sont clos par plusieurs opérations. Montrer que

   (a) si L est local, alors L^* est local.

   (b) si L_1 et L_2 sont des langages locaux sur des aphabets disjoints, alors $L_1 \cup L_2$ et $L_1 \cdot L_2$ sont des langages locaux.

   Montrer qu'une expression linéaire représente un langage local.

5. En déduire un algorithme de contruction d'un automate équivalent à une expression rationnelle. Quelle est la taille de l'automate obtenu ?

6. Appliquer cet algorithme à l'expression rationnelle $(ab + b)^*ba$.

7. Soit $\Sigma_n = \{1, \ldots, n\}$ un alphabet et L l'ensemble des sous-mots de $1 \cdots n$. Donner une expression rationnelle pour L, et construire son automate de Glushkov.