DM Langages Formels

À rendre le jeudi 14 mars

Soit Σ un alphabet fini. On considère une extension des expressions régulières avec l'opération d'inverse : soit $\mathrm{Reg}(\Sigma, {}^{-1})$ l'ensemble des expressions

$$e ::= 0 \mid 1 \mid a \mid e + e \mid e \cdot e \mid e^* \mid e^{-1}\,, \qquad \text{où } a \in \Sigma\,.$$

On note aussi $\mathrm{Reg}(\Sigma)$ l'ensemble des expressions qui n'utilisent pas ${}^{-1}$, i.e., des expressions régulières usuelles.

Une expression e est interprétée comme une *relation* sur un ensemble E. Plus précisément, étant donné un ensemble E et une interprétation $\sigma : \Sigma \to \mathcal{P}(E \times E)$ qui associe à chaque lettre une relation binaire sur E, on définit inductivement une extension $\sigma : \mathrm{Reg}(\Sigma, {}^{-1}) \to \mathcal{P}(E \times E)$ de σ aux expressions comme suit :

$$\begin{aligned}
\sigma(0) &= \emptyset \\
\sigma(1) &= \{(x,x) \mid x \in E\} \\
\sigma(a) &= \sigma(a) \\
\sigma(e+f) &= \sigma(e) \cup \sigma(f) \\
\sigma(e \cdot f) &= \sigma(e) \cdot \sigma(f) = \{(x,z) \in E \times E \mid \exists y, (x,y) \in \sigma(e) \wedge (y,z) \in \sigma(f)\} \\
\sigma(e^*) &= \sigma(e)^* = \{(x,y) \in E \times E \mid \exists n \geq 0, \exists x_0 = x, x_1, \ldots, x_n = y, \\
&\qquad\qquad\qquad\qquad \forall\, 0 \leq i < n, (x_i, x_{i+1}) \in \sigma(e)\} \\
\sigma(e^{-1}) &= \sigma(e)^{-1} = \{(x,y) \in E \times E \mid (y,x) \in \sigma(e)\}\,.
\end{aligned}$$

En particulier, pour un mot $u = a_0 a_1 \cdots a_{n-1} \in \Sigma^*$, $\sigma(u) = \sigma(a_0) \cdot \sigma(a_1) \cdots \sigma(a_{n-1})$.

On dit que deux expressions e et f sont équivalentes, noté $e \equiv f$, si pour tout ensemble E et pour toute interprétation $\sigma : \Sigma \to \mathcal{P}(E \times E)$, on a $\sigma(e) = \sigma(f)$. On note également $e \subseteq f$ si pour tous E et $\sigma : \Sigma \to \mathcal{P}(E \times E)$, on a $\sigma(e) \subseteq \sigma(f)$.

On s'intéresse au problème de décider si deux expressions sont équivalentes.

1 Expressions régulières

On considère d'abord le cas d'expressions régulières construites sans l'opérateur ${}^{-1}$. On définit le langage $L(e) \subseteq \Sigma^*$ d'une expression de manière usuelle :

$$L(0) = \emptyset \qquad L(1) = \{\varepsilon\} \qquad L(a) = \{a\}$$
$$L(e_1 + e_2) = L(e_1) \cup L(e_2) \qquad L(e_1 \cdot e_2) = L(e_1) \cdot L(e_2) \qquad L(e^*) = L(e)^*\,.$$

Question 1.

(a) Montrer que pour tout ensemble E et pour toute interprétation $\sigma : \Sigma \to \mathcal{P}(E \times E)$, pour toute expression régulière $e \in \mathrm{Reg}(\Sigma)$, on a

$$\sigma(e) = \bigcup_{v \in L(e)} \sigma(v)\,.$$

(b) Montrer que pour toutes expressions régulières $e, f \in \mathrm{Reg}(\Sigma)$, on a $e \equiv f$ si et seulement si $L(e) = L(f)$.

2 Expressions avec inverse

On dit qu'une expression $e \in \mathrm{Reg}(\Sigma, {}^{-1})$ est *normalisée* si l'utilisation de ${}^{-1}$ est restreinte aux expressions atomiques, i.e., si e est de la forme :

$$e ::= 0 \mid 1 \mid a \mid a^{-1} \mid e + e \mid e \cdot e \mid e^*\,, \qquad \text{où } a \in \Sigma\,.$$

Question 2. Montrer que pour toute expression $e \in \mathrm{Reg}(\Sigma, {}^{-1})$, il existe une expression normalisée $e' \in \mathrm{Reg}(\Sigma, {}^{-1})$ telle que $e \equiv e'$.

Question 3. On considère les expressions

$$e_1 = a \qquad e_2 = aa^{-1}a \qquad e_3 = bb^{-1}a$$

Pour tous $i, j \in \{1, 2, 3\}$, a-t-on $e_i \subseteq e_j$? Dans chaque cas, donner une preuve que l'inclusion est valide ou un contre-exemple.

On définit une copie $\overline{\Sigma}$ de Σ, qu'on suppose disjointe de Σ :

$$\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}\,.$$

Pour $a \in \Sigma$, on notera $\overline{(\overline{a})} = a$. Pour tous E et $\sigma : \Sigma \to \mathcal{P}(E \times E)$, on définit une extension $\overline{\sigma} : (\Sigma \cup \overline{\Sigma}) \to \mathcal{P}(E \times E)$ de σ à $\Sigma \cup \overline{\Sigma}$, en posant, pour tout $a \in \Sigma$,

$$\overline{\sigma}(a) = \sigma(a)\,, \quad \text{et} \quad \overline{\sigma}(\overline{a}) = (\sigma(a))^{-1}\,.$$

Pour toute expression $e \in \mathrm{Reg}(\Sigma, {}^{-1})$, on fixe une expression normalisée e' telle que $e \equiv e'$. On note $\tau(e) \in \mathrm{Reg}(\Sigma \cup \overline{\Sigma})$ l'expression régulière sur $\Sigma \cup \overline{\Sigma}$ obtenue en substituant toute occurence de a^{-1} dans e' par $\overline{a}$. Par exemple, $\tau((a + b^{-1})b(b^{-1}a^{-1})^*ab) = (a + \overline{b})b(\overline{b}\overline{a})^*ab$.

Question 4. Soit E un ensemble, et $\sigma : \Sigma \to \mathcal{P}(E \times E)$. Montrer que pour tout $e \in \mathrm{Reg}(\Sigma, {}^{-1})$,

$$\sigma(e) = \overline{\sigma}(\tau(e)) = \bigcup_{u \in L(\tau(e))} \overline{\sigma}(u)\,.$$

Question 5. Montrer que si $\tau(e) \equiv \tau(f)$ alors $e \equiv f$, mais que la réciproque n'est pas vraie : donner un exemple d'expressions e et f telles que $e \equiv f$, mais $\tau(e) \not\equiv \tau(f)$.

Soit $u = a_0 \dots a_{n-1} \in (\Sigma \cup \overline{\Sigma})^*$, où $a_i \in \Sigma \cup \overline{\Sigma}$ pour tout $0 \leq i < n$. On peut voir u comme un mot ou comme une expression sur $\Sigma \cup \overline{\Sigma}$, mais aussi comme un modèle dans lequel on pourra évaluer les expressions. Plus précisément, on associe à e l'ensemble $E_u = \{0, \dots, n\}$, et l'interprétation $\sigma_u : \Sigma \to \mathcal{P}(E_u \times E_u)$ définie par

$$\sigma_u(a) = \{(i, i+1) \mid 0 \leq i < n \wedge a_i = a\} \cup \{(i+1, i) \mid 0 \leq i < n \wedge a_i = \overline{a}\}\,.$$

Comme au dessus, on notera $\overline{\sigma_u} : \Sigma \cup \overline{\Sigma} \to \mathcal{P}(E_u \times E_u)$ son extension à $\Sigma \cup \overline{\Sigma}$: pour tout $a \in \Sigma$,

$$\begin{aligned}
\overline{\sigma_u}(a) &= \{(i, i+1) \mid 0 \leq i < n \wedge a_i = a\} \cup \{(i+1, i) \mid 0 \leq i < n \wedge a_i = \overline{a}\} \\
\overline{\sigma_u}(\overline{a}) &= \{(i, i+1) \mid 0 \leq i < n \wedge a_i = \overline{a}\} \cup \{(i+1, i) \mid 0 \leq i < n \wedge a_i = a = \overline{(\overline{a})}\}\,.
\end{aligned}$$

Question 6. Soit $u = ab\overline{b}\overline{a}ab$, et $e = (a + b^{-1})b(b^{-1}a^{-1})^*ab$. Que vaut $\sigma_u(e)$? Aucune justification n'est demandée.

Question 7. Soit $u, v \in (\Sigma \cup \overline{\Sigma})^*$. Montrer que les propositions suivantes sont équivalentes :

1. Pour tous E et $\sigma : \Sigma \to \mathcal{P}(E \times E)$, $\overline{\sigma}(u) \subseteq \overline{\sigma}(v)$.
2. $(0, |u|) \in \overline{\sigma_u}(v)$.

Question 8. Soit $e, f \in \mathrm{Reg}(\Sigma, {}^{-1})$. Montrer que les propositions suivantes sont équivalentes :

1. $e \equiv f$.
2. Pour tout $u \in (\Sigma \cup \overline{\Sigma})^*$, $\sigma_u(e) = \sigma_u(f)$.
3. Pour tout $u \in (\Sigma \cup \overline{\Sigma})^*$, $(0, |u|) \in \sigma_u(e)$ si et seulement si $(0, |u|) \in \sigma_u(f)$.

Un *automate à double sens* (ou *automate boustrophédon*) est un automate fini non-déterministe qui, à chaque transition, peut déplacer sa tête de lecture vers la droite ou vers la gauche. Avant une transition, la tête de lecture de l'automate se situe entre deux lettres, et, selon qu'il se déplace vers la gauche ou vers la droite, l'automate va lire la lettre située à sa gauche ou à sa droite. [1]

Plus formellement, un automate à double sens sur un alphabet Σ est un quadruplet $\mathcal{B} = (Q, T, I, F)$, où

— Q est un ensemble fini d'états,

— $I \subseteq Q$ est un ensemble d'états initiaux,

— $F \subseteq Q$ est un ensemble d'états acceptants,

— $T \subseteq Q \times \{\to, \leftarrow\} \times \Sigma \times Q$ est un ensemble de transitions.

Soit $u = a_0 \dots a_{n-1} \in \Sigma^*$. Un *calcul* de $\mathcal{B}$ sur u est une suite $(q_0, i_0), \dots, (q_k, i_k)$ telle que pour tout $0 \leq j \leq k$, $(q_j, i_j) \in Q \times \{0, \dots, n\}$, et si $j < k$:

— soit $i_{j+1} = i_j + 1$ et $(q_j, \to, a_{i_j}, q_{j+1}) \in T$,

— soit $i_{j+1} = i_j - 1$ et $(q_j, \leftarrow, a_{i_j - 1}, q_{j+1}) \in T$.

Un calcul de $\mathcal{B}$ est *acceptant* si $i_0 = 0$, $i_k = n$, $q_0 \in I$, et $q_k \in F$. Le langage accepté par $\mathcal{B}$, noté $L(\mathcal{B})$, est l'ensemble des mots $u \in \Sigma^*$ tels que $\mathcal{B}$ a un calcul acceptant sur u.

On admet temporairement le résultat suivant (prouvé dans la section 3) :

Théorème 1. *Pour tout automate à double sens $\mathcal{B}$ avec n états, on peut construire effectivement un automate déterministe classique $\mathcal{A}$ avec $2^{O(n^2)}$ états tel que $L(\mathcal{A}) = L(\mathcal{B})$. De plus, la construction de $\mathcal{A}$ est dans* PSPACE.

Pour tout $e \in \mathrm{Reg}(\Sigma, ^{-1})$, on note $\mathcal{A}_e = (Q_e, T_e, I_e, F_e)$ un automate fini sur $\Sigma \cup \overline{\Sigma}$ tel que $L(\mathcal{A}_e) = L(\tau(e))$. On lui associe un automate à double sens $\mathcal{B}_e = (Q_e, U_e, I_e, F_e)$ sur $\Sigma \cup \overline{\Sigma}$, défini par

$$U_e = \{(q, \to, a, q') \mid (q, a, q') \in T_e\} \cup \{(q, \leftarrow, \overline{a}, q') \mid (q, a, q') \in T_e\} .$$

Chaque transition de $\mathcal{A}_e$ sur une lettre $a \in \Sigma \cup \overline{\Sigma}$ donne donc lieu à deux transitions dans $\mathcal{B}_e$: une transition vers la droite sur a, et une transition vers la gauche sur $\overline{a}$.

1. Remarque : une autre définition, plus usuelle, consiste à considérer que la tête de lecture de l'automate est positionnée sur une lettre, et que la lettre lue ne dépend pas de la direction de déplacement (comme dans une machine de Turing). On peut aussi ajouter des marqueurs de début et de fin pour permettre à l'automate de tester s'il a atteint le début ou la fin du mot. Toutefois, toutes ces définitions ont le même pouvoir d'expression.

Question 9. Montrer que

$$L(\mathcal{B}_e) = \{u \in (\Sigma \cup \overline{\Sigma})^* \mid (0, |u|) \in \sigma_u(e)\} .$$

Question 10. Montrer que le problème de l'équivalence de deux expressions dans $\mathrm{Reg}(\Sigma, {}^{-1})$ est décidable. Quelle est sa complexité ?

3 Automates à double sens

On va maintenant prouver le Théorème 1.

Soit $\mathcal{B} = (Q, T, I, F)$ un automate à double-sens sur un alphabet Σ. Pour tout $u \in \Sigma^*$, on appelle

— $\lambda(u)$ l'ensemble des paires d'états $(q, q') \in Q \times Q$ telles qu'il existe un calcul de $\mathcal{B}$ sur u qui commence à gauche de u et finit à droite de u et va de l'état q à l'état q', i.e., un calcul de $\mathcal{B}$ sur u de la forme $(q_0, i_0), \ldots, (q_k, i_k)$ tel que $q_0 = q$, $i_0 = 0$, $q_k = q'$, et $i_k = |u|$; similairement,

— $\mu(u)$ l'ensemble des $(q, q') \in Q \times Q$ tels qu'il existe un calcul de $\mathcal{B}$ sur u de la forme $(q_0, i_0), \ldots, (q_k, i_k)$ avec $q_0 = q$, $i_0 = |u|$, $q_k = q'$, et $i_k = |u|$.

Pour $u, v \in \Sigma^*$, on note $u \sim v$ quand $(\lambda(u), \mu(u)) = (\lambda(v), \mu(v))$.

Question 11. Montrer que $\sim$ est une congruence à droite, c'est-à-dire une relation d'équivalence telle que pour tous $u, v, w \in \Sigma^*$, $u \sim v$ implique $uw \sim vw$.

Question 12. Prouver le Théorème 1. On pourra construire un automate dont les états sont des classes d'équivalence de $\sim$.