Paddings rationnels

Travail 02

Ceci est une proposition de correction. Il existe d'autres chemins pour arriver au même résultat.

Partie 1 : logarithme d'un langage

Soit $\mathcal{A}$ un alphabet fini et $L \subseteq \mathcal{A}^*$ un langage. On définit le logarithme de L :

$$LOG(L) = \{u \in \mathcal{A}^* \mid \exists v \in \mathcal{A}^*, |v| = 2^{|u|} \text{ et } uv \in L\}$$

L'objectif est de montrer que si L est régulier, alors $LOG(L)$ aussi.

Soit $\mathfrak{A} = (\mathcal{A}, Q, i, F, \delta)$ un automate fini déterministe reconnaissant L. On pose $\mathcal{G} = (\mathcal{P}(Q))^Q$. On définit, pour $g, h \in \mathcal{G}$, l'opération :

$$g \circ h = q \mapsto \bigcup_{q' \in h(q)} g(q')$$

On pose $g_0 \in \mathcal{G}$ la fonction $g_0 : q \mapsto \{\delta(q, a) \mid a \in \mathcal{A}\}$.

On définit enfin l'automate déterministe $\mathfrak{B}$:

— l'alphabet est toujours $\mathcal{A}$;
— ses états sont $Q' = Q \times \mathcal{G}$;
— son état initial est $i' = (i, g_0)$;
— la fonction de transition est $\Delta : ((q, g), a) \mapsto (\delta(q, a), g \circ g)$;
— les états finaux sont $F' = \{(q, g) \mid F \cap g(q) \neq \varnothing\}$.

Exercice 1

Donner la définition « en français » de $\mathcal{G}$, puis déterminer son cardinal.

Solution.

$\mathcal{G}$ est l'ensemble des fonctions de Q dans $\mathcal{P}(Q)$. Son cardinal est $|\mathcal{P}(Q)|^{|Q|} = \left(2^{|Q|}\right)^{|Q|} = 2^{|Q|^2}$.

Exercice 2

On définit, pour $n \in \mathbb{N}$, la propriété suivante :

$$P(n) : \forall x \in \mathcal{A}^n, \Delta^*(i', x) = (q, g) \iff \delta^*(i, x) = q \text{ et } \forall r \in Q, g(r) = \{s \in Q \mid \exists m \in \mathcal{A}^{2^n}, \delta^*(r, m) = s\}$$

Moins formellement, $\Delta^*(i', x)$ renvoie $\delta^*(i, x)$ ainsi que la fonction qui, à un état, associe les états atteignables après lecture d'un mot de longueur $2^{|x|}$.

Montrer, pour tout $n \in \mathbb{N}$, la propriété $P(n)$.

On pourra, si on le souhaite, utiliser la notation $\delta^*(r, \mathcal{A}^K) = \{s \in Q \mid \exists m \in \mathcal{A}^K, \delta^*(r, m) = s\}$ pour $K \in \mathbb{N}$.

Solution.

On procède par récurrence sur n.
Initialisation : pour $n = 0$: on remarque que $\mathcal{A}^0 = \{\varepsilon\}$, donc il reste à vérifier la propriété pour seulement le mot ε.
D'un côté, $\Delta^*(i', \varepsilon) = i' = (i, g_0)$. On a bien $\delta^*(i, \varepsilon) = i$; puis pour tout $r \in Q, g_0(r) = \{\delta(r, a) \mid a \in \mathcal{A}\}$

ce qui correspond bien aux états accessibles depuis r en lisant un mot de longueur $2^0 = 1$.
La réciproque est assurée par le caractère déterministe de $\mathfrak{B}$.
Hérédité : on suppose disposer que $P(n)$ pour un certain rang n. Soit $x \in \mathcal{A}^{n+1}$; on écrit $x = ay$ avec $a \in \mathcal{A}$ et $y \in \mathcal{A}^n$.
D'abord, $\Delta^*(i', x) = \Delta(\Delta^*(i', y), a)$; soit $\Delta^*(i', x) = (g, q)$. Alors $\Delta^*(i', x) = \Delta((g, q), a) = (\delta(q, a), g \circ g)$. Par hypothèse de récurrence, on sait que $q = \delta^*(i, y)$; donc $\delta(q, a) = \delta(\delta^*(i, y), a) = \delta^*(i, ya) = \delta^*(i, x)$. De plus, toujours par hypothèse de récurrence, on sait que pour tout $r \in Q$, $g(r) = \delta^*(r, \mathcal{A}^{2^{|y|}})$ (voir la notation proposée par l'énoncé). Donc :

$$\begin{aligned} g \circ g(r) &= \bigcup_{q' \in g(r)} g(q') \\ &= \bigcup_{q' \in g(r)} \delta^*(q', \mathcal{A}^{2^{|y|}}) \end{aligned}$$

Or, si $q' \in g(r)$, il existe $m \in \mathcal{A}^{2^{|y|}}$ tel que $\delta^*(r, m) = q'$; si $s \in \delta^*(q', \mathcal{A}^{2^{|y|}})$, il existe $w \in \mathcal{A}^{2^{|y|}}$ tel que $\delta^*(q', w) = s$; donc $\delta^*(r, mw) = s$; et $|mw| = |m| + |w| = 2^{|y|} + 2^{|y|} = 2^{|y|+1} = 2^{|x|}$. Tout cela permet de conclure que pour tout $r \in Q, g \circ g(r) \subseteq \delta^*(r, \mathcal{A}^{2^{|x|}})$; pour l'inclusion réciproque, soit $\tilde{m} \in \mathcal{A}^{2^{|x|}} = \mathcal{A}^{|y|} \cdot \mathcal{A}^{|y|}$, on peut découper $\tilde{m}$ en $\tilde{m} = m \cdot w$ avec $m, w \in \mathcal{A}^{2^{|y|}}$; donc $\delta^*(r, \tilde{m}) \in \delta^*(\delta^*(r, m), w) \subseteq g \circ g(r)$.
La réciproque est assurée par le caractère déterministe de $\mathfrak{B}$.
Donc $P(n+1)$ est vraie.

Exercice 3

Montrer que $\mathfrak{B}$ reconnaît $LOG(L)$.

Solution.

Soit $x \in \mathcal{L}(\mathfrak{B})$. Notons $(q, g) = \Delta^*(i, x)$; alors $(q, g) \in F'$. Par l'exercice précédent, on sait que $q = \delta^*(i, x)$ et que $g : r \mapsto \delta^*(r, \mathcal{A}^{2^{|x|}})$. Donc $g(q) = \delta^*(\delta^*(i, x), \mathcal{A}^{2^{|x|}})$; on sait par F' que $g(q) \cap F \neq \varnothing$. Donc il existe $y \in \mathcal{A}^{2^{|x|}}$ tel que $\delta^*(\delta^*(i, x), y) \in F$; donc $\delta^*(i, xy) \in F$. Donc $xy \in \mathcal{L}(\mathfrak{A}) = L$, et $|y| = 2^{|x|}$; donc $x \in LOG(L)$.
Réciproquement, soit $x \in LOG(L)$. Alors il existe $y \in \mathcal{A}^{2^{|x|}}$ tel que $xy \in L$. Donc si $\Delta^*(i', x) = (q, g)$, $q = \delta^*(i, x)$ et $g : r \mapsto \delta^*(r, \mathcal{A}^{2^{|x|}})$. Or, $\delta^*(i, xy) = \delta^*(\delta^*(i, x), y) = \delta^*(q, y) \in F$ (car $xy \in L$) ; donc comme $\delta^*(q, y) \in \delta^*(q, \mathcal{A}^{2^{|x|}}) \subseteq g(q)$, $g(q) \cap F \neq \varnothing$; donc $(q, g) \in F'$, et donc $x \in \mathcal{L}(\mathfrak{B})$.

Partie 2 : généralisation à d'autres fonctions

De manière générale, pour $f : \mathbb{N} \mapsto \mathbb{N}$ et un langage L, on définit :

$$PAD_f(L) = \{u \in \mathcal{A}^* \mid \exists v \in \mathcal{A}^*, |v| = f(|u|) \text{ et } uv \in L\}$$

On note alors $FP = \{f : \mathbb{N} \to \mathbb{N} \mid \forall L \subseteq \mathcal{A}^*, L \text{ rationnel} \implies PAD_f(L) \text{ rationnel}\}$: ce sont les fonctions qui préservent la rationnalité des langages par l'opération de remplissage PAD.

Exercice 4

1. Justifier que $e : n \mapsto 2^n \in FP$.
2. Montrer que $id_{\mathbb{N}} : n \mapsto n \in FP$.
3. Montrer que $c : n \mapsto n^2 \in FP$.
   Indication : on posera $Q' = Q \times \mathcal{G} \times \mathcal{G}$, et on remarquera que $(n+1)^2 = n^2 + 2n + 1$.
4. Justifier succinctement que pour tout $k \in \mathbb{N}$, $f_k : n \mapsto n^k \in FP$.

5. Montrer que $fibo : n \mapsto fibo(n) \in FP$ (avec $fibo(0) = 0, fibo(1) = 1$ et $fibo(n+2) = fibo(n+1) + fibo(n)$).
6. Justifier que $FP \neq \mathbb{N}^{\mathbb{N}}$.

Solution.

1. On remarque que pour un langage L,

$$PAD_e(L) = \{u \in \mathcal{A}^* \mid \exists v \in \mathcal{A}^*, |v| = 2^{|u|} \text{ et } uv \in L\} = LOG(L)$$

   On vient de démontrer que si L est rationnel, alors $LOG(L)$ est aussi rationnel; donc $e \in FP$.
2. Pour un automate déterministe $\mathfrak{A} = (\mathcal{A}, Q, i, F, \delta)$, on définit un nouvel automate $\mathfrak{B}$ de la manière suivante :
   — d'abord, on pose $g_{id} : q \mapsto \{q\}$ (on souligne que $g_{id} \neq g_0$ (a priori));
   — l'alphabet de $\mathfrak{B}$ est toujours $\mathcal{A}$;
   — ses états sont $Q' = Q \times \mathcal{G}$;
   — son état initial est $i' = (i, g_{id})$;
   — la fonction de transition est $\Delta : ((q, g), a) \mapsto (\delta(q, a), g \circ g_0)$;
   — les états finaux sont $F' = \{(q, g) \mid F \cap g(q) \neq \varnothing\}$.

   Par rapport à ce qu'on a fait partie 1, seul l'état initial et la fonction de transition ont changé. Alors il est possible de montrer pour tout $n \in \mathbb{N}$ l'équivalent de la propriété $P(n)$ dans le cas présent, qui deviendrait :

$$\forall x \in \mathcal{A}^n, \Delta^*(i', x) = (q, g) \iff \delta^*(i, x) = q \text{ et } \forall r \in Q, g(r) = \delta^*(r, \mathcal{A}^{|x|})$$

   En fait, dans un état $(q, g) \in Q'$, g sert de compteur pour retenir la longueur du mot lu.
3. Pour un automate déterministe $\mathfrak{A} = (\mathcal{A}, Q, i, F, \delta)$, on définit un nouvel automate $\mathfrak{B}$ de la manière suivante :
   — l'alphabet de $\mathfrak{B}$ est toujours $\mathcal{A}$;
   — ses états sont $Q' = Q \times \mathcal{G} \times \mathcal{G}$;
   — son état initial est $i' = (i, g_{id}, g_{id})$;
   — la fonction de transition est $\Delta : ((q, g, h), a) \mapsto (\delta(q, a), g \circ g_0, h \circ g \circ g \circ g_0)$;
   — les états finaux sont $F' = \{(q, g, h) \mid F \cap h(q) \neq \varnothing\}$.

   Alors il est possible de montrer pour tout $n \in \mathbb{N}$ l'équivalent de la propriété $P(n)$ dans le cas présent, qui deviendrait :

$$\forall x \in \mathcal{A}^n, \Delta^*(i', x) = (q, g, h) \iff \delta^*(i, x) = q \text{ et } \forall r \in Q, g(r) = \delta^*(r, \mathcal{A}^{|x|}) \text{ et } h(r) = \delta^*(r, \mathcal{A}^{|x|^2})$$

   En fait, dans un état $(q, g, h) \in Q'$, g sert à compter n et h à compter n^2 (avec n la longueur du mot lu).
4. On peut généraliser le processus précédent; si on peut créer un automate pour f_k, il suffit de créer un état compteur pour toute puissance $j \leqslant k$; et par le binôme de Newton, toute puissance peut s'écrire comme une combinaison linéaire à coefficients entiers naturels des puissances précédentes. La taille de l'automate explose, cependant.
5. Pour un automate déterministe $\mathfrak{A} = (\mathcal{A}, Q, i, F, \delta)$, on définit un nouvel automate $\mathfrak{B}$ de la manière suivante :
   — l'alphabet de $\mathfrak{B}$ est toujours $\mathcal{A}$;
   — ses états sont $Q' = Q \times \mathcal{G} \times \mathcal{G}$;
   — son état initial est $i' = (i, g_{id}, g_0)$;
   — la fonction de transition est $\Delta : ((q, g, h), a) \mapsto (\delta(q, a), h, g \circ h)$;

— les états finaux sont $F' = \{(q, g, h) \mid F \cap g(q) \neq \varnothing\}$.

Alors il est possible de montrer pour tout $n \in \mathbb{N}$ l'équivalent de la propriété $P(n)$ dans le cas présent, qui deviendrait :

$$\Delta^*(i', x) = (q, g, h) \iff \delta^*(i, x) = q \text{ et } \forall r \in Q, g(r) = \delta^*(r, \mathcal{A}^{fibo(|x|)}) \text{ et } h(r) = \delta^*(r, \mathcal{A}^{fibo(|x|+1)})$$

L'idée est la même, on utilise des compteurs.

Remarque. On en déduit que pour toute suite récurrente linéaire à coefficients entiers positifs, la suite en question est dans FP.

6. On définit la fonction suivante :

$$f : n \mapsto \begin{cases} 0 \text{ si } n \text{ est un carré impair parfait} \\ 1 \text{ si } n \text{ est un impair qui n'est pas un carré parfait} \\ 0 \text{ si } n \text{ est pair} \end{cases}$$

Remarquons que f est bien définie sur $\mathbb{N}$. Plus encore, on observe que pour tout $n \in \mathbb{N}$:

— si n est pair, $n + f(n)$ est pair ;
— si n est un impair non carré, $n + f(n)$ est pair ;
— si n est un impair carré, $n + f(n)$ est impair.

Autrement écrit, $n + f(n)$ est impair ssi n est un carré impair.

Posons $L = (a + b) \cdot ((a + b)^2)^*$: c'est l'ensemble des mots de longueur impaire. Montrons alors que $PAD_f(L) = \{u \in \mathcal{A}^* \mid |u| \text{ est un carré impair}\}$.

Si u est de longueur un carré impair, il suffit de choisir $v = \varepsilon$; alors $uv \in L$ (et on a bien $|v| = f(|u|)$).

Si u est de longueur paire, $f(|u|) = O$; la seule possibilité pour v serait ε, mais uv serait de longueur paire et donc $uv \notin L$.

Si u est de longueur un impair non carré, alors $f(|u|) = 1$; donc $|u| + f(|u|)$ est pair, donc, peu importe le choix de v tel que $|v| = f(|u|)$, $uv \notin L$.

Avec le lemme de l'étoile, on peut montrer que $\{u \in \mathcal{A}^* \mid |u| \text{ est un carré impair}\}$ est irrationnel ; donc L est rationnel, mais $PAD_f(L)$ est irrationnel, donc $f \notin FP$.

Partie 3 : considérations algorithmiques

Exercice 5

Soit $\mathfrak{A} = (\mathcal{A}, Q, i, F, \delta)$ un automate fini.

1. Donner la taille d'un automate $\mathfrak{B}$ reconnaissant $LOG(\mathcal{L}(\mathfrak{A}))$.
2. Proposer un algorithme en pseudo-code permettant de construire $\mathfrak{B}$ à partir de $\mathfrak{A}$.
3. Établir la complexité temporelle de votre algorithme.

Solution.

1. En observant la partie 1, on remarque que $|Q'| = |Q| \times 2^{|Q|^2}$; la fonction de transition Δ, qu'on implémente généralement comme une matrice, est alors de taille $|Q'|^2 = |Q|^2 \times 2^{2 \times |Q|^2}$.
2. On commence par parler de l'implémentation des éléments de $\mathcal{G}$: comme il s'agit d'une fonction de Q dans $\mathcal{P}(Q)$, on encodera ces éléments comme des matrices, de sorte que `g.(q)` est un tableau de booléens vérifiant `g.(q).(q')` ssi $g' \in g(q)$. On procède ici par un parcours en profondeur ; on pourrait faire un parcours en largeur.

$\Delta \leftarrow$ un dictionnaire vide
$S \leftarrow$ une liste vide
construire g_0
$P \leftarrow$ une pile vide
empiler (i, g_0)
tant que P *est non vide* **:**
    dépiler (q, g)
    calculer $g \circ g$
    empiler tous les couples $(\delta(q, a), g \circ g)$ et les ajouter à S (s'ils ne l'ont pas encore été)
    ajouter à Δ les cases $\Delta[(q, g), a] = (\delta(q, a), g \circ g)$
renvoyer Δ et S

3. Pour construire g_0, il s'agit de lire δ, qui est plus ou moins une simple liste de taille $|\mathcal{A}| \times |Q|$; g_0 est de taille $|Q|^2$, donc cette construction est en $O(|Q| \times (|\mathcal{A}| + |Q|))$.

   Au pire, notre parcours en profondeur fera autant d'itérations de la boucle « tant que » qu'il n'y a d'arêtes dans le graphe sous-jacent ; donc au plus $O(|Q|^2 \times 2^{2\times|Q|^2})$ itérations.

   Reste à calculer chaque itération de la boucle « tant que ». La principale difficulté est le calcul de $g \circ g$: il s'agit de $|Q|$ unions de tableaux de booléens de taille $|Q|$. Chaque union est en $O(|Q|)$; donc au total, on a besoin de $O(|Q|^2)$ opérations.

   On obtient ainsi une complexité en $O(|Q| \times (|\mathcal{A}| + |Q|) + |Q|^4 \times 2^{2\times|Q|^2})$. Le premier terme, sauf alphabet complètement dément, sera négligeable : on peut raisonnablement attendre une complexité en $O(|Q|^4 \times 2^{2\times|Q|^2})$. C'est beaucoup.