TD 1 : Recherche de chaînes de caractères et expressions rationnelles

Correction

1 Recherche d'expressions rationnelles

Question 1 : Correction :

La démonstration est constructive, par récurrence sur la taille de l'expression rationnelle. (Il s'agit d'une autre démonstration que tout langage rationnel est reconnaissable.)

Si $e \equiv \epsilon$, l'automate $\mathcal{A}(e) = \langle \{i,t\}, i, t, \{(i,\epsilon,t)\}\rangle$ [1]. Cet automate est normalisé et a 2 états, donc $|Q| = 2 \leq 2|e|$.

Si $e \equiv a$, avec $a \in A$, l'automate $\mathcal{A}(e) = \langle \{i,t\}, i, t, \{(i,a,t)\}\rangle$. Cet automate est normalisé et a 2 états, donc $|Q| = 2 \leq 2|e|$.

Si $e \equiv e' + e''$ et $\mathcal{A}(e') = \langle Q', i', t', \delta'\rangle$ et $\mathcal{A}(e'') = \langle Q'', i'', t'', \delta''\rangle$ sont les automates construits par récurrence pour e' resp. e''. L'automate qui reconnaît e est $\mathcal{A}(e) = \langle Q' \cup Q'' \cup \{i,t\}, i, t, \delta' \cup \delta'' \cup \{(i,\epsilon,i'), (i,\epsilon,i''), (t',\epsilon,t), (t'',\epsilon,t)\}\rangle$. L'automate est normalisé et son nombre d'états est égal à $|Q'| + |Q''| + 4$ qui, en appliquant l'hypothèse d'induction est inférieur à $2|e'| + 2|e''| + 2 = 2|e|$.

Si $e \equiv e'.e''$ et $\mathcal{A}(e') = \langle Q', i', t', \delta'\rangle$ et $\mathcal{A}(e'') = \langle Q'', i'', t'', \delta''\rangle$ sont les automates construits par récurrence pour e' resp. e''. L'automate qui reconnaît e est $\mathcal{A}(e) = \langle (Q' \setminus \{t'\}) \cup Q'', i', t'', \delta'[t' \mapsto i''] \cup \delta''\rangle$ (l'état t' est remplacé par i''). L'automate est normalisé et son nombre d'états est égal à $|Q'| + |Q''| - 1$ qui, en appliquant l'hypothèse de récurrence est inférieur à $2|e'| + 2|e''| - 1 < 2|e|$.

Si $e \equiv e'^*$ et $\mathcal{A}(e') = \langle Q', i', t', \delta'\rangle$ est l'automate construit par récurrence pour e'. L'automate qui reconnaît e est $\mathcal{A}(e) = \langle Q' \cup \{i,t\}, i, t, \delta' \cup \{(i,\epsilon,i'), (t',\epsilon,t), (t',\epsilon,i'), (i,\epsilon,t)\}\rangle$. L'automate est normalisé et son nombre d'états est égal à $|Q'| + 2$ qui, en appliquant l'hypothèse de récurrence est inférieur à $2|e'| + 2 = 2|e|$.

Question 2 : Correction :

`Trans`(P, a) renvoie l'ensemble des états atteignables depuis P en ayant lu a.

La complexité en temps $O(N)$ est obtenue si les opérations d'initialisation d'un (sous-)ensemble d'états de Q est fait en $O(N)$, tandis que l'ajout d'un état à un sous-ensemble est effectué en temps constant. Ceci est possible en utilisant une représentation des ensembles par des vecteurs caractéristiques.

```
Algorithme Trans(P,a)
Entree: P ensemble d'états, a une lettre
Sortie: Ensemble d'états atteignables avec a depuis P
R := emptyset          % doit être en O(N)
pour q dans P faire    % O(N) tours
  s'il existe q' tq (q,a,q') est une transition % 2 fois car automate normalisé
  alors R = R U { q' } % doit être en temps constant
```

1. On utilise la notation pour un automate $\mathcal{A} = \langle Q, q_{init}, q_{final}, \delta\rangle$.

```
  finsi
finpour
retourner R
```

Question 3 : Correction :

Epsilon(P) renvoie l'ensemble des états atteignables depuis P en suivant des ϵ transitions.

Comme pour la fonction précédente, la complexité en temps $O(N)$ est obtenue si les opérations sur les ensembles d'états (par exemple, copie) sont possibles en $O(N)$ tandis que l'ajout et la suppression d'états sont des opérations en temps constant.

```
Algorithme Epsilon(P)
Entree: P ensemble d'états
Sortie: Ensemble d'états atteignables avec epsilon transitions depuis P
R := P;                % doit être en O(N)
T := P;                % doit être en O(N)
tant que T <> emptyset faire % O(N) tours
  enlève un q de T     % doit être en O(1)
  pour chaque (q, eps, q') faire % que 2, automate normalisé
    si q' absent de R alors % doit être en O(1)
      R := R U { q' }     % doit être en O(1)
      T := T U { q' }     % doit être en O(1)
    finsi
  finpour
fintantque
retourner R
```

Question 4 : Correction :

Occur_e(T) affiche les positions de T auxquelles sont reconnus des préfixes en $\mathcal{L}(e)$.

```
P := Epsilon({i});     % en O(N)
si t dans P alors      % en O(1)
  afficher 0
pour i en [1,|T|] faire % en O(|T|)
  P := Epsilon(Trans(P,T[i]))  % en O(N)
  si P est vide alors terminer % en O(N)
  si t en P alors afficher i   % en O(1)
fintpour
```

2 Automate des parties

Question 5 : Correction :

L'automate de $m+1$ états est $\mathcal{A} = \langle \{0,...,m\}, 0, m, \cup_{i=1}^{m}\{(i-1, P[i], i)\} \ \cup \ \cup_{a\in\Sigma}\{(0,a,0)\}$.

Les équations du langage $\mathcal{L}(i)$ sont vérifiées sur $\mathcal{A}$ en partant de $\mathcal{L}(m)$.

Question 6 : Correction :

La preuve de cette propriété est faite dans les notes de cours.

Question 7 : Correction :

Exemple : pour $P = ababcaba$, donc $m = 8$ et $\mathcal{A}$ a 9 états. Les états atteignables de $\mathcal{A}'$ sont des sous-ensembles de $\{0,...,8\}$ (les états de $\mathcal{A}$) tels que 0 est toujours présent (à cause de la boucle sur 0 en $\mathcal{A}$) et un seul état est final, celui qui contient m (en plus de 0). Plus précisément, il s'agit de l'ensemble $\{\{0\}, \{0,1\}, \{0,2\}, \{0,1,3\}, \{0,2,4\}, \{0,5\}, \{0,1,6\}, \{0,2,7\}, \{0,1,3,8\}\}$.

Soit Q' les états atteignables de $\mathcal{A}'$.

On démontre $Q' \subseteq \{\pi^*(k) \mid 0 \leq k \leq m\}$: Soit $q \in Q'$; on démontre que $q = \pi^*(max(q))$.

Si $q = \{0\}$ **:** alors $max(q) = 0$; comme $\pi^*(0) = \{0\}$, qed.

Si $q \supset \{0\}$ **:** et atteignable, donc pour tout $\ell \in q$ il existe un mot w t.q. le chemin de 0 à ℓ existe en A avec les transitions étiquetées par w. Dans ce cas, $P[1,\ell] \sqsupseteq w$ (w a le suffixe $P[1,\ell]$). Soit $k = max(q)$. Par définition de $\mathcal{A}$, le mot qui amène à k est $P[1,k]$. Par la propriété ci-dessus, pour tout $\ell < k$ en q, $P[1,\ell] \sqsupseteq P[1,k]$. D'après le point 2 ci-dessus, ceci est exactement $\pi^*(max(q))$, qed.

On démontre $Q' \supseteq \{\pi^*(k) \mid 0 \leq k \leq m\}$: Soit $k \in [0,m]$ et q un état de $\mathcal{A}'$ atteint en lisant $P[1,k]$. On démontre par récurrence sur k que $q = \pi^*(k)$:

Cas $k = 0$ **:** alors $q = \{0\} = \pi^*(0)$, qed.

Cas $0 < k \leq m$ **:** et la propriété est vraie pour $k' < k$. Donc :

$$q = \{\ell' \mid \exists \ell \in \pi^*(k-1), \ell' \in \delta(\ell, P[k])\}.$$

Comme l'état 0 a une boucle avec Σ en $\mathcal{A}$, 0 est en q. Donc

$$q = \{0\} \cup \{\ell + 1 \mid \exists \ell \in \pi^*(k-1), (\ell, P[k], \ell+1) \in \delta\}.$$

En utilisant la propriété de la question précédente, on obtient $q = \{0\} \cup \big(\pi^*(k) \setminus \{0\}\big)$, qed.

3 Période d'un mot

Question 8 : <u>Correction</u> : Démonstration que $Period(u) = |u| - \pi(|u|)$. On note $n = |u|$.

$$\begin{aligned}
Period(u) &= min\{0 < k \leq n \mid \forall i \in [1, n-k]\ u[i] = u[i+k]\} \quad \text{par def. periode} \\
&= min\{0 < k \leq n \mid u[1, n-k] = u[k+1, n]\} \quad \text{par reformulation} \\
&= min\{0 < n-k' \leq n \mid u[1,k'] = u[n-k'+1, n]\} \quad \text{changement } n-k = k' \\
&= n - max\{0 \leq k' < n \mid u[1,k'] \sqsupset u[1,|u|]\} \quad \text{par reformulation } = \\
&= n - \pi(n) \quad \text{définition } \pi
\end{aligned}$$

Question 9 : <u>Correction</u> : Démonstration que si p et q périodes de u et $p + q \leq |u|$ alors $pgcd(p,q)$ est période de u.

Si $p = q$, la démonstration est triviale.

Sinon, on suppose que $p < q$ et on démontre le lemme : $q - p$ est aussi une période. Soit $i \in [1, |u| - (q-p)]$.

Si $i \leq |u| - q$ **:** alors $u[i] = u[i+q]$ car q période. Comme $i + q - p \leq |u| - q + q - p$ alors $u[i+q-p] = u[i+q-p+p]$ car p période. Donc $u[i] = u[i+q] = u[i+q-p]$ donc $q-p$ période, qed.

Si $i > |u| - q$ **:** comme $i - p \geq 1$ (car $p + q < |u|$), alors $u[i-p] = u[i-p+p]$ car p période et $u[i-p] = u[i-p+q]$ car q période et $i - p \leq |u| - q$, donc $q - p$ période, qed.

La démonstration utilise le lemme ci-dessus et le fait que l'algorithme d'Euclide lent termine avec le calcul du pgcd pour obtenir le résultat.

Question 10 : <u>Correction</u> :

a. La démonstration est faite par récurrence.

$n = 5$ **:** $F_5[: -2] = aba$ et $F_4[: -2] = a$ et $F_3 = ab$, qed.

$n > 5$ **:** On suppose la propriété vraie pour tout $k \leq n$ et on le démontre pour $n+1$:

$$\begin{array}{rcl} F_{n+1}[: -2] & = & (F_n F_{n-1})[: -2] \\ & = & (F_{n-1} F_{n-2} F_{n-1})[: -2] \\ & = & F_{n-1}(F_{n-2} F_{n-1})[: -2] \\ & = & F_{n-1} F_n[: -2] \end{array}$$

b. Toujours par récurrence, on détaille que le cas d'hérédité.

$$\begin{array}{rcl} F_{n-1}^2 & = & F_{n-1} F_{n-1} \\ & = & F_{n-1} F_{n-2} F_{n-3} \;=\; F_n F_{n-3},\; qed. \\ F_{n-2}^3 & = & F_{n-2} F_{n-2} F_{n-2} \\ & = & F_{n-2} F_{n-3} F_{n-4} F_{n-2} \\ & = & \underbrace{F_{n-1} F_{n-4} F_{n-3}} F_{n-4} \end{array}$$

or $F_{n-1} F_{n-4} F_{n-3}[: -2]$ est $F_{n-1} F_{n-2}[: -2]$ qui est $F_n[: -2]$

c. Remarque : pour $n \geq 5$, $|F_{n-1}| < |F_{n-2}| \leq |F_n[: -2]|$.

$$\begin{array}{rcl} F_n[: -2] & = & F_{n-1} F_{n-2}[: -2] \\ & = & \underbrace{F_{n-2} F_{n-3}}_{|F_{n-1}|} F_{n-2}[: -2] \\ donc & & \forall i \in [1, |F_n[: -2]| - |F_{n-1}|],\; F_n[: -2][i] = F_n[: -2][i + |F_{n-1}|] \end{array}$$

La propriété (a) indique :

$$\begin{array}{rcl} F_n[: -2] & = & F_{n-2} F_{n-1}[: -2] \\ & = & \underbrace{F_{n-2}}_{|F_{n-2}|} (F_{n-2} F_{n-3})[: -2] \\ donc & & \forall i \in [1, |F_n[: -2]| - |F_{n-2}|],\; F_n[: -2][i] = F_n[: -2][i + |F_{n-2}|] \end{array}$$

Une autre propriété intéressante de $F_n[: -2]$: il est un palindrome.