# Option Informatique en Spé MP et MP*
# Automate des sous-mot : le corrigé

***Question 1*** • Si $z = 1$, alors $u = a^n$ possède exactement $n + 1$ sous-mots (autant que de facteurs). Si $z = n$, alors $u$ possède $2^n$ sous-mots : en effet, la fonction qui à une partie $\mathcal{I}$ de $[\![1, n]\!]$ associe le sous-mot obtenu en gardant les $u_i$ tels que $i \in \mathcal{I}$ est clairement bijective.

***Question 2*** • Notons $q_0, \dots, q_n$ les états ; il y a $n$ transitions de la forme $(q_{i-1}, u_i, q_i)$ et $n$ transitions de la forme $(q_{i-1}, \varepsilon, q_i)$. Le calcul associé au sous-mot $v$ de $u$ passe par les transitions du premier type qui correspondent aux lettres conservées, et par les transitions instantanées qui correspondent aux lettres effacées.

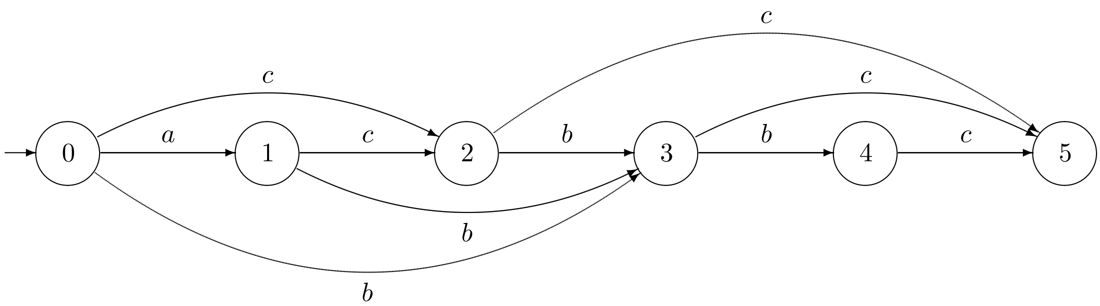
***Question 3*** • Énumérons les sous-mots de *abbc* par longueur croissante et, à longueur égale, par ordre lexicographique : $\varepsilon$, *a*, *b*, *c*, *ab*, *ac*, *bb*, *bc*, *abb*, *abc*, *bbc* et *abbc*.

Énumérons maintenant les étiquettes des calculs réussis de l'automate proposé (dont chaque état est final). Pour changer, nous choisirons les transitions de haut en bas ... Mots de longueur 0 : $\varepsilon$ ; de longueur 1 : *b*, *a*, *c* ; de longueur 2 : *bc*, *bb*, *ab*, *ac* ; de longueur 3 : *bbc*, *abc*, *abb* ; de longueur 4 : *abbc*. Nous obtenons bien le même ensemble.

***Question 4*** • La tableau $v[1..z]$ est initalisé avec des zéros. La gestion de l'ensemble des états est banale, et n'est donc pas explicitée.

- $j \leftarrow 1$ ; $u_j = a$ ; $i$ décrivant l'intervalle $[\![v[u_j], j-1]\!] = [\![0, 0]\!]$, on ajoute la transition $(q_0, a, q_1)$ ; $v[1] \leftarrow 1$ ;
- $j \leftarrow 2$ ; $u_j = c$ ; $i$ décrivant l'intervalle $[\![v[u_j], j-1]\!] = [\![0, 1]\!]$, on ajoute les transitions $(q_0, c, q_2)$ et $(q_1, c, q_2)$ ; $v[2] \leftarrow 2$ ;
- $j \leftarrow 3$ ; $u_j = b$ ; $i$ décrivant l'intervalle $[\![v[u_j], j-1]\!] = [\![0, 2]\!]$, on ajoute les transitions $(q_0, b, q_3)$, $(q_1, b, q_3)$ et $(q_2, b, q_3)$ ; $v[3] \leftarrow 3$ ;
- $j \leftarrow 4$ ; $u_j = b$ ; $i$ décrivant l'intervalle $[\![v[u_j], j-1]\!] = [\![3, 3]\!]$, on ajoute la transition $(q_3, b, q_4)$ ; $v[4] \leftarrow 4$ ;
- $j \leftarrow 5$ ; $u_j = c$ ; $i$ décrivant l'intervalle $[\![v[u_j], j-1]\!] = [\![2, 4]\!]$, on ajoute les transitions $(q_2, c, q_5)$, $(q_3, c, q_5)$ et $(q_4, c, q_5)$ ; $v[5] \leftarrow 5$.

Et voici l'automate :




► Nous noterons $v_j[x]$ la valeur de $v[x]$ au moment où l'algorithme aborde la $j$-ième étape ; nous aurons donc $v_{j+1}[u_j] = j$ et $v_{j+1}[x] = v_j[x]$ pour tout $k \neq j$.

***Question 5*** • Nous avons besoin d'un résultat préliminaire. Soit $x \in \Sigma$. Montrons que pour $j \in [\![1, z]\!]$ et $i \geqslant v_j[x]$, aucune transition étiquetée $x$ n'émane de $q_i$. Le résultat est clair pour $j = 0$ puisqu'à ce moment l'automate en cours de construction ne possède aucune transition. Supposons le résultat acquis à l'entrée de la $j$-ième étape de l'algorithme ; des transitions étiquetée $x$ ne seront ajoutées au cours de cette étape que si $x = u_j$, mais dans ce cas elles émaneront des états $q_{v_j[u_j]}$ à $q_{j-1}$. Donc, à l'entrée de la $(j+1)$-ième étape, aucune transition étiquetée $x$ n'émanera de $q_i$, pour $i \geqslant j = v_{j+1}[u_j]$. Par récurrence, l'assertion est vraie pour tout $j \in [\![0, n]\!]$.

• Fixons $i \in [\![0, n]\!]$ et $x \in \Sigma$ et considérons une transition partant de $q_i$. Notons $u_j$ l'étiquette de cette transition et $q_j$ l'état auquel elle mène. Par construction, $v_j[u_j] \leqslant i < j$. Lorsque l'algorithme aborde la $j$-ième étape, aucune transition étiquetée $u_j$ n'émane de $q_{v_j[u_j]}$, et à plus forte raison de $q_i$. Après cette $j$-ième étape, $q_i$ ne sera plus pris en compte, même si l'on retrouve une occurrence de $u_j$. Donc il n'existe pas d'autre transition partant de $q_i$ et étiquetée $u_j$ : ceci prouve que $\mathcal{A}_u$ est déterministe.

***Question 6*** • L'assertion est vraie au démarrage de l'algorithme. Plaçons-nous au début de la $j$-ième étape ; il est clair que les $v[x]$ avec $x \neq u_j$ ne sont pas modifiés au cours de cette étape. Par hypothèse, $v_j[u_j]$ est le plus petit numéro d'un état dont n'émane aucune transition étiquetée $u_j$ ; et nous avons vu à la question 5 qu'aucune transition étiquetée $u_j$ n'émane de $q_i$, pour $i \geqslant v_j[u_j]$. Au cours de cette étape, des transitions étiquetées $u_j$ sont ajoutées, mais les numéros des états dont elles partent sont exactement les entiers $v_j[j]$ à $j-1$ ; donc, à la fin de cette étape, $v_{j+1}[u_j] = j$ est le plus petit numéro d'un état dont n'émane aucune transition étiquetée $u_j$.

***Question 7*** Preuve du sens direct : la transition $q_i \xrightarrow{x} q_j$ est ajoutée à la $j$-ième étape de l'algorithme : nécessairement, $x = u_j$ et $v_j[j] \leqslant i < j$. À ce moment, aucune transition étiquetée $x$ n'émane de $q_k$ pour $k \geqslant v_j[j]$ (d'après le résultat préliminaire établi à la question 5) et donc en particulier pour $v_j[j] \leqslant k < j$. Donc $j$ est le plus petit $k > i$ tel que $u_k = x$.

Preuve de la réciproque : fixons $i$ et $x$. Supposons qu'il existe $k > i$ tel que $u_k = x$ ; notons $j$ le plus petit indice $k$ vérifiant ceci. Pour $i < k < j$, nous avons $u_k \neq x$, donc il n'existe pas de transition $q_i \xrightarrow{x} q_k$ ; donc $v_j[u_j] = v_j[x] \leqslant i$. Ainsi $v_j[u_j] \leqslant i < j$ ; de par le fonctionnement de l'algorithme, une transition $q_i \xrightarrow{x} q_j$ sera ajoutée à la $j$-ième étape.

***Question 8*** • Clairement, $\varepsilon$ est reconnu par $\mathcal{A}_u$. Soit $t = t_1 \dots t_p$ un sous-mot de $u$ autre que $\varepsilon$ ; il existe une injection croissante $s$ de $[\![1,p]\!]$ dans $[\![1,n]\!]$ telle que $t_i = u_{s(i)}$ pour tout $i \in [\![1,p]\!]$. Définissons $\sigma : [\![1,p]\!] \mapsto [\![1,n]\!]$ comme suit : $\sigma(1)$ est le plus petit $i \geqslant 1$ tel que $t_1 = u_i$ ; puis, pour $k \in [\![2,p]\!]$, $\sigma(k)$ est le plus petit $i \geqslant \sigma(k-1)$ tel que $t_k = u_i$. Nous définissons ainsi une injection croissante $\sigma$ telle que $t_i = u_{\sigma(i)}$ pour tout $i \in [\![1,p]\!]$. D'après le résultat de Q7, $q_{\sigma(k-1)} \cdot t_k = q_{\sigma(k)}$ pour tout $k \in [\![1,p]\!]$. Donc $q_0 \xrightarrow{t_1} q_{\sigma(1)} \longrightarrow \cdots \xrightarrow{t_i} q_{\sigma(i)} \cdots \xrightarrow{t_p} q_{\sigma(p)}$. Ceci montre que $t$ est l'étiquette d'un calcul réussi de $\mathcal{A}_u$, donc $t$ est reconnu par $\mathcal{A}_u$.

Remarque : parmi les injections croissantes de $[\![1,p]\!]$ dans $[\![1,n]\!]$ qui répondent à la question, $\sigma$ est minimale pour l'ordre lexicographique. Notons également que cette preuve n'utilise pas de raisonnement par récurrence.

***Question 9*** • $\varepsilon$ (qui est reconnu par $\mathcal{A}_u$) est un sous-mot de $u$. Soit $t = t_1 \dots t_p$ reconnu par $\mathcal{A}_u$ ; il existe une suite $q_{i_0} = q_0, q_{i_1}, \dots, q_{i_p}$ d'états telle que $q_{i_0} \xrightarrow{t_1} q_{i_1} \longrightarrow \cdots \xrightarrow{t_p} q_{i_p}$. De par le fonctionnement, $i_{k+1} > i_k$ et $t_k = u_{i_k}$. Alors $\sigma : k \in [\![1,p]\!] \mapsto i_k$ est une injection croissante de $[\![1,p]\!]$ dans $[\![1,n]\!]$ telle que $t_i = u_{\sigma(i)}$ pour tout $i \in [\![1,p]\!]$ : donc $t$ est un sous-mot de $u$.

Remarque : ici encore, la preuve n'utilise pas de raisonnement par récurrence.

***Question 10*** • L'automate étant déterministe, il part de $q_i$ au plus $z$ transitions. D'autre part, le résultat de la question 7 implique que, pour $i$ fixé, la fonction $x \mapsto q_i \cdot x$ est injective ; et, par définition, elle prend ses valeurs dans l'ensemble des $q_k$ où $k$ décrit $[\![i+1,n]\!]$. Par conséquent, si $i \geqslant n-z$, il part au plus $n-i$ transitions de $q_i$.

***Question 11*** • Par définition, $z \leqslant n$. D'où la majoration du nombre de transitions de $\mathcal{A}_u$ :

$$\begin{aligned}\sum_{0 \leqslant i \leqslant n} \min(z, n-i) &= \sum_{0 \leqslant i \leqslant n-z} z + \sum_{n-z < i \leqslant n} (n-i) = (n-z+1)z + \sum_{0 \leqslant i < z} i \\ &= (n-z+1)z + \frac{z(z-1)}{2} = \frac{2(n-z+1)z + z(z-1)}{2} = \frac{z(2n+1-z)}{2}\end{aligned}$$

***Question 12*** • Les mots pour lesquels ce majorant est atteint sont les mots de longueur $\ell \geqslant z$, dont les $z$ dernières lettres sont distinctes : dans ce cas, il partira exactement $n-i$ transitions de l'état $q_{n-i}$.

## Références bibliographiques

▶ Le point de départ de ce sujet est l'article de Zdeněk Troníček et Bořivoj Melichar intitulé *Directed Acyclic Subsequence Graph* (Prague Stringology Club Workshop'98).

**FIN**