Option Informatique en Spé MP et MP*
DS du mardi 7 décembre 2004 : le corrigé

Problème 1 : automates finis pondérés

Question 1 Soient i et f deux objets distincts, n'appartenant pas à Q. Notons $Q' = Q \cup \{i,f\}$ et δ' la réunion de δ et des deux ensembles suivants :

- δ_1 est l'ensemble des transitions (i, x, q) telles qu'il existe une transition (j, x, q) de $\mathcal{A}$ avec $j \in I$;
- δ_2 est l'ensemble des transitions (q, x, f) telles qu'il existe une transition (q, x, g) de $\mathcal{A}$ avec $g \in F$.

Remarquons que l'intersection de ces deux ensembles peut ne pas être vide, s'il existe une transition menant d'un état de I à un état de F ; ceci revient à dire que $\mathcal{A}$ reconnaît au moins un mot de longueur 1. Définissons alors $\mathcal{A}' = (Q', \delta', \{i\}, \{f\})$. Il est clair que les propriétés concernant i et f sont vérifiées. Il reste à montrer que $\mathcal{A}'$ reconnaît le même langage que $\mathcal{A}$.

• Soit $x \in \Sigma$; x est reconnu par $\mathcal{A}$ ssi il existe une transition (q, x, q') de $\mathcal{A}$ telle que $q \in I$ et $q' \in F$; mais ceci revient à dire que $\mathcal{A}'$ possède la transition (i, x, f) ou encore que x est reconnu par $\mathcal{A}'$.

• Soit u de longueur $n \geqslant 2$, reconnu par $\mathcal{A}$. u est l'étiquette d'un calcul réussi de $\mathcal{A}$, qui passe par les états $q_0, q_1, \ldots, q_{n-1}, q_n$ avec $q_0 \in I$ et $q_n \in F$. Alors u est l'étiquette d'un calcul de $\mathcal{A}'$ qui passe successivement par les états $i, q_1, \ldots, q_{n-1}, f$. Ce calcul est donc réussi : par suite, u est reconnu par $\mathcal{A}'$.

• Soit u de longueur $n \geqslant 2$, reconnu par $\mathcal{A}'$. u est l'étiquette d'un calcul réussi $i, q_1, \ldots, q_{n-1}, f$. De par la construction de $\mathcal{A}'$, il existe $q_0 \in I$ et $q_n \in F$ tels que les transitions (q_0, u_1, q_1) et (q_{n-1}, u_n, q_n) appartiennent à δ. Donc u est l'étiquette d'un calcul de $\mathcal{A}$ qui passe successivement par les états $q_0, q_1, \ldots, q_{n-1}, q_n$. Ce calcul est donc réussi : par suite, u est reconnu par $\mathcal{A}$.

Question 2 Un calcul de $\mathcal{A}_{\mathrm{eq}}$ de longueur non nulle est de l'une des deux formes suivantes : $1 \xrightarrow{a^n} 1$ avec $n > 0$; ou $1 \xrightarrow{a^n} 1 \xrightarrow{b} 2 \xrightarrow{b^q}$ avec $n \geqslant 0$ et $q \geqslant 0$. Les calculs réussis sont ceux du deuxième type, de poids total nul ; ceci équivaut à l'égalité $p = q + 1$. Donc les étiquettes des calculs réussis sont les mots de la forme $a^n b^n$ avec $n > 0$.

Question 3 Les mots de M_{diff} sont de trois types :

- ceux de la forme a^p avec $p > 0$;
- ceux de la forme $a^p b^q$ avec $p > q > 0$;
- ceux de la forme b^q avec $q > 0$;
- ceux de la forme $a^p b^q$ avec $q > p > 0$.

Nous allons montrer que M_{diff} est le langage reconnu par le $\mathbb{Z}$-automate de la figure 1 :

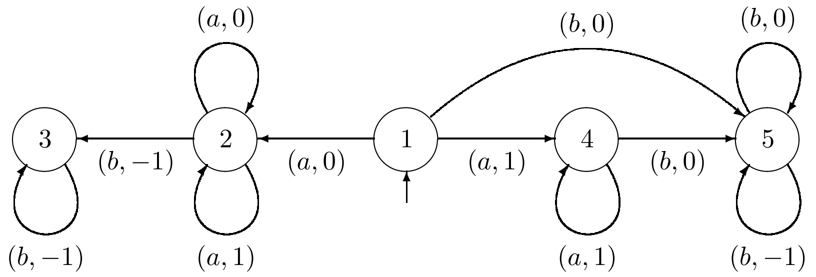

Figure 1: un $\mathbb{Z}$-automate reconnaissant M_{diff}

Justifions cette affirmation. Les calculs de M_{diff} ont une longueur au moins égale à 1 ; nous pouvons les séparer en quatre types, selon les états traversés :

- un calcul qui se termine dans l'état 2 est réussi ssi il a pour étiquette a^p avec $p > 0$ et n'emprunte pas la boucle $2 \xrightarrow{(a,1)} 2$;
- un calcul qui se termine dans l'état 3 est réussi ssi il a une étiquette de la forme $a^{q+n}b^q$ où $n > 0$ est le nombre de transitions étiquetées par le couple $(a, 0)$;
- un calcul qui ne passe pas par l'état 4 se termine dans l'état 5 est réussi ssi il a une étiquette de la forme b^q avec $q > 0$ et n'emprunte pas la boucle $5 \xrightarrow{(b,-1)} 5$;
- un calcul qui passe par l'état 4 et se termine dans l'état 5 est réussi ssi il a une étiquette de la forme $a^p b^{p+n+1}$ où $n \geqslant 0$ est le nombre de passages dans la boucle $5 \xrightarrow{(b,0)} 5$.

Ceci termine la preuve.

▶ La figure 2 présente une solution proposée par une étudiante du lycée Fénelon ; l'automate proposé n'a que quatre états.

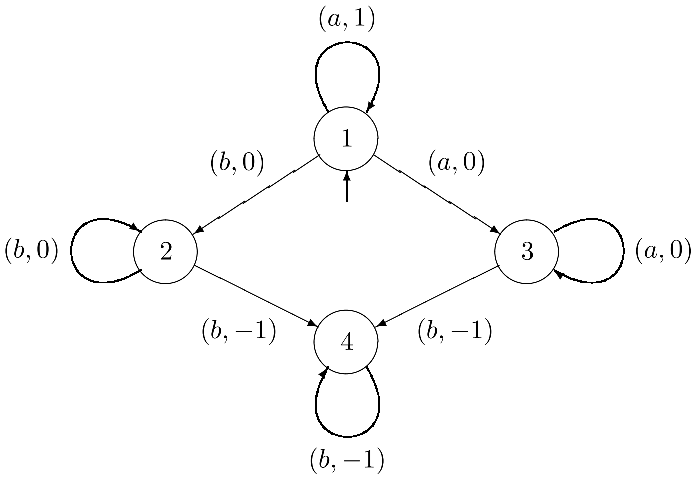

Figure 2: un $\mathbb{Z}$-automate à quatre états reconnaissant M_{diff}

Question 4 D'après le résultat de la question 1, nous pouvons supposer $I = \{i\}$ et $F = \{f\}$. Définissons $\mathcal{B} = (Q, \delta', i)$ où δ' est la partie de $Q \times \{e, g, g^{-1}\}$ définie par les règles suivantes :

- si $q' \neq f$, alors $(i, x, g, q') \in \delta'$ ssi $(i, x, q') \in \delta$;
- si $q \neq i$ et $q' \neq f$, alors $(q, x, e, q') \in \delta'$ ssi $(q, x, q') \in \delta$;
- si $q \neq i$, alors $(q, x, g^{-1}, f) \in \delta'$ ssi $(q, x, f) \in \delta$;
- $(i, x, e, f) \in \delta'$ ssi $(i, x, f) \in \delta$.

Les calculs réussis de $\mathcal{B}$ sont de deux sortes : ceux de longueur 1, réduits à une transition $i \xrightarrow{(x,e)} f$; et ceux de longueur supérieure à 1, composés d'une transition $i \xrightarrow{(x,g)} q$, suivie de zéro, une ou plusieurs transitions de poids e, suivies d'une transition $q' \xrightarrow{(y,g^{-1})} f$. Il est clair que l'ensemble des étiquettes de ces calculs est identique à l'ensemble des étiquettes de calculs réussis de $\mathcal{A}$.

Question 5 $\mathcal{A} = (Q, \delta, i)$. Définissons successivement :

- $Q' = Q \times G$;
- $i' = (i, e)$ où e est le neutre de G ;
- $F' = Q \times \{e\}$;
- enfin $\big((q, g), x, (q', g')\big)$ appartient à δ' ssi $(q, x, g^{-1}g', q')$ est une transition de δ.

Notons $\mathcal{A}' = (Q', \delta', i', F')$. Le calcul $i \xrightarrow{(u_1,g_1)} q_1 \ldots \xrightarrow{(u_n,g_n)} q_n$ de $\mathcal{A}$ est réussi ssi $g_1 \ldots g_n = e$; mais $(i, e) \xrightarrow{u_1} (q_1, g1) \xrightarrow{u_2} (q_2, g_2) \ldots \xrightarrow{u_n} (q_n, g_n)$ est un calcul réussi de $\mathcal{A}'$ ssi $g_n = e$. Ces deux calculs ont même étiquette. Ceci montre que $\mathcal{A}'$ reconnaît le même langage que $\mathcal{A}$.

Question 6 Soient $\mathcal{A}_1 = (Q_1, \delta_1, i_1)$ et $\mathcal{A}_2 = (Q_2, \delta_2, i_2)$ deux $\mathbb{Z}$-automates qui reconnaissent respectivement L_1 et L_2. Nous pouvons supposer Q_1 et Q_2 disjoints. Soit $i \notin Q_1 \cup Q_2$. Notons $Q = Q_1 \cup Q_2 \cup \{i\}$; δ est la réunion de δ_1, δ_2 et des ensembles de transitions δ_3 et δ_4 définis par les règles suivantess :

- $(i, x, g, q') \in \delta_3$ ssi $(i_1, x, g, q') \in \delta_1$;
- $(i, x, g, q') \in \delta_4$ ssi $(i_1, x, g, q') \in \delta_2$;

Le $\mathbb{Z}$-automate $\mathcal{A} = (Q, \delta, i)$ reconnaît $L_1 \cup L_2$ (preuve non détaillée ici).

Question 7 Appliquons le résultat de la question 4, en prenant comme groupe $(\mathbb{Z}, +)$ et $g = +1$: ainsi, R est reconnu par un $\mathbb{Z}$-automate $\mathcal{A}_R = (Q_R, \delta_R, i_R)$, déduit d'un automate fini non déterministe $\mathcal{A}_r = (Q_R, \gamma, i_R, f_R)$ réduit. D'autre part, L est reconnu par un $\mathbb{Z}$-automate $\mathcal{A}_L = (Q_L, \delta_L, i_L)$; nous pouvons supposer Q_R et Q_L disjoints. Construisons un $\mathbb{Z}$-automate $\mathcal{A} = (Q, \delta, i)$ reconnaissant $R \cdot L$. Son ensemble d'états est $Q = Q_R \cup Q_L$; son état initial est i_R ; enfin, sa table de transition δ est obtenue en ajoutant à la réunion de δ_L et δ_R chaque transition (f_R, x, g, q') telle que (i_L, x, g, q') appartienne à δ_L. Soient γ_R et γ_L des calculs réussis de $\mathcal{A}_R$ et $\mathcal{A}_L$ respectivement ; notons $\widehat{\gamma_R}$ le calcul de $\mathcal{A}$ déduit de γ_R en remplaçant la première transition (i_L, x, g, q') par (f_R, x, g, q') ; alors $\gamma_L \widehat{\gamma_R}$ est un calcul réussi de $\mathcal{A}$. Réciproquement, tout calcul réussi de $\mathcal{A}$ passe par un ou plusieurs états de $\mathcal{A}_R$, le dernier étant nécessairement f_R ; puis par plusieurs états états de $\mathcal{A}_L$, le premier étant q'. Notons γ_1 la partie de ce calcul qui s'achève en f_R, et γ_2 celle qui commence en f_R. Il est clair que γ_1 est un calcul réussi de $\mathcal{A}_L$; et que, si l'on remplace f_R par i_L dans γ_2, on obtient un calcul réussi de $\mathcal{A}_L$.

Question 8 Appliquons le résultat de la question 4, en prenant comme groupe $(\mathbb{Z}, +)$ et $g = +1$: ainsi, R est reconnu par un $\mathbb{Z}$-automate $\mathcal{A}_R = (Q_R, \delta_R, i_R)$. D'autre part, L est reconnu par un $\mathbb{Z}$-automate $\mathcal{A}_L = (Q_L, \delta_L, i_L)$; quitte à multiplier le poids de chaque transition par 2, nous pouvons supposer que les poids des transitions de $\mathcal{A}_L$ sont pairs. Construisons un $\mathbb{Z}$-automate $\mathcal{A} = (Q, \delta, i)$ reconnaissant $R \cap L$. Son ensemble d'états est $Q = Q_R \times Q_L$; son état initial est (i_R, i_L) ; sa table de transition δ est définie comme suit : $\big((q_R, q_L), x, g, (q'_R, q'_L)\big)$ appartient à δ ssi il existe une transition (q_R, x, g_R, q'_R) de $\mathcal{A}_R$ et une transition (q_L, x, g_L, q'_L) de $\mathcal{A}_L$ telles que $g = g_R + g_L$. Il est clair qu'un calcul de $\mathcal{A}$ est réussi ssi les deux calculs obtenus par projection sur les deux automates sont réussis ; la clé est que les poids des transitions de $\mathcal{A}_R$ n'interfèrent pas avec les poids des transitions de $\mathcal{A}_L$ (preuve non détaillée ici).

Question 9 $R_1 = \{c\}^*$ et $R_2 = \{a\}^*$ répondent à la question.

Question 10 Ni ab, ni bc ne peut apparaître dans q ; donc $q = a^j$ ou $q = b^j$ ou $q = c^j$ avec $j > 0$. Mais dans chaque cas le mot pq^2r n'est pas dans L : en effet, dans $u = a^n b^n c^n$, un ou deux des trois facteurs a^n, b^n et c^n verront leur longueur augmenter, et le troisième ne sera pas modifié.

Question 11 Comme précédemment, ni ab, ni bc ne peuvent apparaître dans q, donc les lettre de q sont toutes égales ; même raisonnement pour s. Clairement, le mot pq^2rs^2t n'est pas dans L.

Question 12 Si le $\mathbb{Z}$-langage reconnu par $\mathcal{A}$ est infini, alors ce langage contient au moins un mot u de longueur au moins n. Il est l'étiquette d'un calcul réussi qui passe par les états $q_0, q_1, \ldots, q_p$ avec $p \geqslant n$. Le principe des tiroirs affirme que deux de ces états sont confondus ; donc il existe des indices j et k tels que $0 \leqslant j < k \leqslant p$ et $q_j = q_k$. Ceci met en évidence le cycle $q_j \to q_{j+1} \to \cdots \to q_k$ dans le graphe de $\mathcal{A}$.

Question 13 Considérons un calcul réussi de $\mathcal{A}$ contenant un cycle γ de poids nul. L'étiquette de ce calcul est de la forme pqr, où $q \neq \varepsilon$ est l'étiquette du cycle. Mais alors le mot pq^kr appartient à L pour tout $k \in \mathbb{N}$, ce qui contredit le résultat de la question 10.

► Nous dirons qu'un calcul de $\mathcal{A}$ est *élémentaire* s'il ne comporte aucun cycle. Nous dirons qu'un cycle de $\mathcal{A}$ est *élémentaire* si les états autres que celui dans lequel commence et se termine le cycle ne sont traversés qu'une fois.

Question 14 Supposons que tous les cycles apparaissant dans des calculs réussis de $\mathcal{A}$ soient de poids strictement négatif. L'ensemble des calculs élémentaires (réussis ou non) est fini ; donc l'ensemble des poids de ces calculs est une partie bornée de $\mathbb{Z}$; soit $k > 0$ majorant la valeur absolue de tout élément de cet ensemble. L'ensemble des cycles élémentaires est également fini ; or aucun ne peut apparaître itéré plus de k fois. Donc il n'existe qu'un nombre fini de calculs réussis. Ceci contredit l'existence de cycles, prouvée à la question 12.

Question 15 À chaque calcul réussi Γ de $\mathcal{A}$, nous pouvons associer un calcul élémentaire γ, en supprimant au besoin un ou plusiers cycles : la longueur du calcul étant finie, ceci se fait en un nombre fini d'étapes ; nous dirons que γ est un *squelette* de Γ. Il n'existe qu'un nombre fini de calculs élémentaires ; considérons ceux qui traversent au moins un état de $\mathcal{A}$ par lequel passe un cycle. Soient γ un tel calcul et π son poids ; supposons que les cycles qui passent par un état traversé par γ sont tous de poids strictement positif ; alors, le nombre de calculs réussis de $\mathcal{A}$ dont γ est un squelette est fini. Du coup, le langage reconnu par $\mathcal{A}$ est fini : ceci est contradictoire.

Question 16 Soient γ_1 et γ_2 de poids respectifs $\pi_1 > 0$ et $\pi_2 < 0$. Alors ${\gamma_1}^{-\pi_2}$ et ${\gamma_2}^{\pi_1}$ sont deux cycles de poids opposés.

Question 17 Supposons que ces deux cycles sont imbriqués, par exemple $j \leqslant k \leqslant n \leqslant m$. Notons γ_1 le cycle $q_j \to q_{j+1} \to \cdot \to q_{m-1} \to q_m$, π_1 son poids, γ_2 le cycle $q_k \to q_{k+1} \to \cdots \to q_{n-1} \to q_n = q_k$ et π_2 son poids. Notons γ le cycle $q_n \to q_{n+1} \to \cdots \to q_{m-1} \to q_m = q_j \to q_{j+1} \to \cdots \to q_{n-1} \to q_n = q_k$ (obtenu en court-circuitant γ_2 lors d'un parcours de γ_1) et π son poids. Si $\pi = 0$, c'est fini ; sinon, γ_2 et γ sont deux cycles dont les poids sont de signes opposés ; et $|\gamma| + |\gamma_2| < |\gamma_1| + |\gamma_2|$.

• Supposons que ces deux cycles sont enchevêtrés, par exemple $j \leqslant k \leqslant m \leqslant n$. Le raisonnement est le même qu'à la question précédente, en notant cette fois γ le cycle $q_m \to q_{m_1} \to \cdots \to q_{n-1} \to q_n = q_k \to q_{k+1} \to \cdots \to q_{j-1} \to q_j = q_m$.

• Nous concluons en notant qu'au bout d'un nombre fini d'étapes, nous tombons sur deux cycles dont l'un est de poids nul, ou deux cycles dont les poids sont de signes opposés et qui ne sont ni imbriqués, ni enchevêtrés.

Question 18

Question 19 Soit $\mathcal{A}$ un $\mathbb{Z}$-automate ; trois situations peuvent se présenter :

- ou bien aucun calcul réussi de $\mathcal{A}$ ne contient de cycle ; le langage reconnu par $\mathcal{A}$ est fini, ce ne peut être M ;
- ou bien il existe un calcul réussi de $\mathcal{A}$ contenant au moins un cycle de poids nul ; mais alors, d'après la question 13, $\mathcal{A}$ ne peut reconnaître M ;
- ou bien il existe un calcul réussi de $\mathcal{A}$ contenant contenant deux cycles γ_1 et γ_2 dont les poids sont de signes opposés. D'après la question 17, nous pouvons supposer que ces deux cycles ne sont ni imbriqués, ni enchevêtrés. L'étiquette de ce calcul est alors de la forme $pqrst$, où $q \neq \varepsilon$ est l'étiquette de γ_1 et $s \neq \varepsilon$ celle de γ_2. Notons π_1 et π_2 leurs poids respectifs ; alors pour tout $k \in \mathbb{N}$, le mot $pq^{1+k\pi_2}rs^{1+k\pi_1}t$ est dans L ; mais alors, d'après le résultat de la question 11, $\mathcal{A}$ ne peut reconnaître M.

Dans tous les cas, nous aboutissons à une contradiction. Donc M n'est pas un $\mathbb{Z}$-langage. **OUF !**

▶ Dans l'article *Languages accepted by integer weighted finite automatas*, dont est inspiré ce sujet, les auteurs adoptent une démarche différente : ils se proposent de montrer que la famille des $\mathbb{Z}$-langages n'est pas stable par concaténation. Pour ce faire, ils étudient le langage $L_{\text{eq}} \cdot L_{\text{eq}}$; hélas, leur preuve est fausse ...

Question 20 La réponse est négative : notant $\overline{L}$ le complémentaire de L, nous avons $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$. La famille des $\mathbb{Z}$-langages est close par réunion ; si elle était close par complémentation, elle le serait aussi par intersection, ce qui n'est pas.

Problème 2 : longueur minimale d'un mot d'un langage rationnel

Question 1 Ceci a été vu en TD :

- $\lambda(\emptyset) = \mathbf{v}$;
- $\lambda(\varepsilon) = \mathbf{f}$;
- $\lambda(x) = \mathbf{f}$ pour toute lettre x ;
- $\lambda(e + e') = \lambda(e)$ et $\lambda(e')$;
- $\lambda(e \cdot e') = \lambda(e)$ ou $\lambda(e')$;
- $\lambda(e^*) = \mathbf{f}$.

Question 2 Il suffit de transcrire la spécification précédente :

```
let rec lambda = function
 | Vide -> true
 | Epsilon | Lettre _ -> false
 | Union(e,e') -> lambda e & lambda e'
 | Produit(e,e') -> lambda e || lambda e'
 | Etoile _ -> false ;;
```

Question 3 Le travail est plus compliqué qu'à la question 1 :

- $\mu(\emptyset) = -\infty$;
- $\mu(\varepsilon) = 0$;
- $\mu(x) = 1$ pour toute lettre x ;
- $\mu(e + e') = \mu(e)$ si $\mu(e') = -\infty$;
- $\mu(e + e') = \mu(e')$ si $\mu(e) = -\infty$;
- $\mu(e + e') = \min\big(\mu(e), \mu(e')\big)$ dans les autres cas ;
- $\mu(e \cdot e') = -\infty$ si $\mu(e) = -\infty$ ou $\mu(e') = -\infty$;
- $\mu(e \cdot e') = \mu(e) + \mu(e')$ sinon ;
- $\mu(e^*) = 0$.

Question 4 Ici encore, on transcrit la spécification algébrique. Deux remarques :

- utiliser λ pour tester si le langage décrit par une sous-expression est vide serait une erreur, puisqu'il faudrait parcourir deux fois chaque sous-arbre, dans le pire des cas ; ceci mènerait à un coût exponentiel en la taille de l'expression ;
- attention aux filtrages imbriqués : il faut parenthéser tout `match ... with` qui est à l'intérieur d'un autre `match ... with`.

```
let rec mu = function
 | Vide -> None
 | Epsilon ->  Some 0
 | Lettre _ -> Some 1
 | Union(e,e') -> ( match (mu e,mu e') with
   | None,None -> None
   | None,Some x' -> Some x'
   | Some x,None -> Some x
   | Some x,Some x' -> Some(min x x')
   )
 | Produit(e,e') -> ( match (mu e,mu e') with
   | None,_ | _,None -> None
   | Some x,Some x' -> Some(x+x')
   )
 | Etoile _ -> Some 0 ;;
```

Question 5 Notons E_0 l'ensemble des états initiaux ; puis, pour $n \in \mathbb{N}$, notons E_{n+1} la réunion de E_n et de l'ensemble des états que l'on peut atteindre par une transition partant d'un état appartenant à E_n. Cette suite croissante prend ses valeurs dans l'ensemble fini $\mathcal{P}(Q)$ donc elle stationne. On établit que si $E_{n+1} = E_n$, alors la suite stationne à partir du rang n (voire plus tôt). Si $|I| \geqslant 1$, la suite stationne au plus tard à partir du rang $|Q| - |I|$. Ceci nous fournit bien un algorithme.

Question 6 Construisons la suite $(E_n)_{n\in\mathbb{N}}$ en nous arrêtant au premier des deux événements suivants :

- E_n contient un état final : alors $\lambda(L) = \mathbf{f}$;
- $E_{n+1} = E_n$: alors $\lambda(L) = \mathbf{v}$.

Question 7 Même démarche, c'est juste la conclusion qui change :

- E_n contient un état final : alors $\mu(L) = n$;
- $E_{n+1} = E_n$: alors $\mu(L) = -\infty$.

Question 8 Nous supposons L non vide. Notons n_0 le rang à partir duquel la suite $(E_n)_{n\in\mathbb{N}}$ stationne : $n_0 = \mu(L)$. Choisissons $q_{n_0} \in E_{n_0}$: q_{n_0} est final, et il existe $q_{n_0-1} \in E_{n_0-1}$ et $u_{n_0} \in \Sigma$ tels que $q_{n-1} \xrightarrow{u_{n_0}} q_{n_0}$. Nous pouvons poursuivre ce parcours : pour chaque $k \in [\![1, n]\!]$, il existe $q_{k-1} \in E_{k-1}$ et $u_k \in \Sigma$ tels que $q_{k-1} \xrightarrow{u_k} q_k$. Notons $u = u_1u_2\ldots u_n$. Alors $q_0 \xrightarrow{x} q_n$, et q_0 est un état initial ; donc u est un mot de L de longueur minimale.

▶ Remarque : programmer le calcul de l'ensemble de mots de L de longueur $\mu(L)$ est un exercice intéressant ; le coût doit rester linéaire par rapport à la taille de l'expression rationnelle décrivant L.

Question 9 Notons $\mathcal{S}(\mathcal{A}, n)$ l'ensemble des mots de longueur n du langage reconnu par l'automate déterministe complet $\mathcal{A} = (Q, \delta, i, F)$. Deux remarques :

- $\mathcal{S}(\mathcal{A}, 0)$ est vide ssi $i \notin F$;
- $\mathcal{S}(\mathcal{A}, n+1)$ est la réunion des $|\Sigma|$ langages $\{x\} \cdot \mathcal{S}(\mathcal{A}_x, n)$ où $\mathcal{A}_x = (Q, \delta, i \cdot x, F)$.

Il suffit donc de calculer $\mu(L)$, puis de construire la suite des $\mathcal{S}(\mathcal{A}, n)$ pour $n \in [\![0, \mu(L)]\!]$. Le dernier terme de cette suite est le langage cherché. Le coût est au moins égal au nombre de mots à énumérer, il peut donc être exponentiel en $\mu(L)$.

FIN