TD 4 de Langages Formels

MIM2, 2001-2002

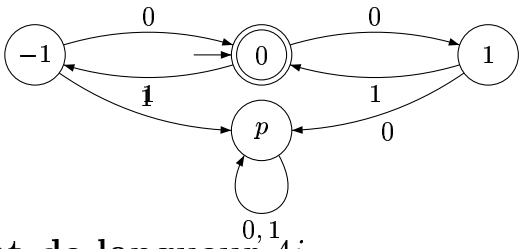
Laure Danthony

*Eléments de correction*

18 octobre 2001

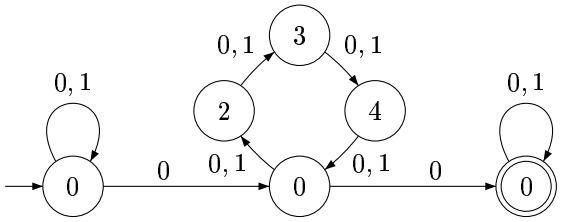
## 1 Même nb de 0 et de 1 et pb de préfixe

On vérifie facilement que l'automate suivant convient :




## 2 Sous mot de longueur $4i$

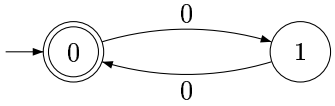
L'automate suivant :




reconnaît bien le langage demandé : on montre facilement qu'il reconnaît les mots de la forme $u0v0w$ avec $|v| = 4i$ et $u, v, w \in \{0, 1\}*$.

## 3 Langage rationnel ou non ?

a) Le langage est reconnu par l'automate :

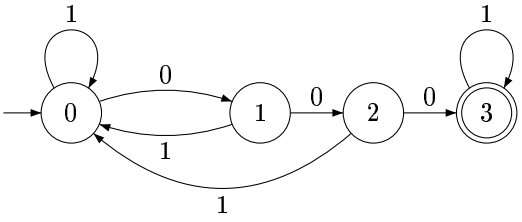


donc est rationnel.

b) Le lemme de l'étoile avec $u = 0^N 1^N 0^{2N}$ fournit, en pompant sur $0^N$, L non rationnel.

c) Voir le cours pour une preuve utilisant le lemme de l'étoile.

d) C'est le langage complémentaire du langage associé l'automate suivant : (qui reconnaît trois zéros consécutifs)




e) $L \cap (0^*1^*) = \{a^n b^b\}$ qui est non rationnel. Donc $L$ n'est pas rationnel.

f) C'est le langage dénoté par exemple par l'ER : $1(0+1)^*1 + 0(0+1)^*0$, donc il est rationnel

g) On peut utiliser le lemme de l'étoile sur $010^n110^n100$ et aucune découpe ne marche, ou bien la preuve qui suit qui est une simulation du lemme de l'étoile :
Le mot $(01)^n(10)^{n-1} \in L$ (avec $u = (01)^n$ et $v = 10$). Soit $\mathcal{M}$ un automate fini à $k$ états reconnaissant le langage $L$ rationnel. Dès que $n > k$, il existe $i, j$ entre 1 et $n$, distincts, vérifiant $\delta(q_0, (01)^i) = \delta(q_0, (01)^j)$. On pompe alors avec $(01)^{j-i}$. Le mot $(01)^{n+p}(10)^{n+1}$ $(p \geq 1)$ doit appartenir à L. Pour cela, il faut couper entre deux lettres consécutives identiques, donc il n'y a qu'un seul choix, c'est $u = (01)^{n+1}$, donc $u = (01)^{P+1}$, donc $v = \varepsilon$ et ça ne marche pas. contradiction.

# 4 Langages rationnels à partir d'un langage rationnel

a) Notons $|Q| = k$, $Q = q_1, \ldots q_k$. On réalise $2k$ copies de l'automate $\mathcal{M}$ reconnaissant $L$, que l'on dispose en deux colonnes. L'automate $\mathcal{M}'$ est alors construit de la façon suivante : l'état initial que l'on note $q_0$ pointe par $\varepsilon$-transition vers l'état $q_1$ du premier automate de la première colonne, vers $q_2$ du deuxième automate de la première colonne, .... Ensuite les états finaux du premier automate de la premiere colonne pointent par $\varepsilon$-transition vers l'état initial $q_1$ de son voisin le premier automate de la deuxième colonne ; les états finaux du 2° automate de la première colonne pointent par $\varepsilon$-transition vers l'état $q_1$ de son voisin de droite, .... Les états terminaux de $\mathcal{M}'$ seront $q_1$ du premier automate de la deuxième colonne, $q_2$ du deuxième automate de la deuxième colonne, .... On vérifie que c'est bien un automate reconnaissant $CYCLE(L)$.

b) $MAX(L)$ est reconnu par $\mathcal{M}' = (Q, \Sigma, \delta, q_0, F')$ avec $F' = \{q \in F, \forall u \in \Sigma^+, \delta(q, u) \notin F\}$.

c) $MIN(L)$ est reconnu par $\mathcal{M}' = (Q', \Sigma, \delta', q_0, F)$ avec $Q' = Q \cup \{p\}$, pour $q \in Q, \delta'(q,a) = \begin{cases} \delta(q,a) & \text{si } q \notin F \\ p & \text{sinon.} \end{cases}$, pour $a \in A, \delta'(p,a) = p$.

d) On marque terminaux tous les états qui pouvaient mener à un état terminal.

e) On considère un automate déterministe qui reconnaît $L$, on renverse les flèches et on rajoute un état initial qui pointe par epsilon-transition vers tous les anciens états terminaux, l'ancien état initial devient terminal.

# 5 Questions supplémentaires, à chercher

On suppose toujours que $L$ est rationnel.

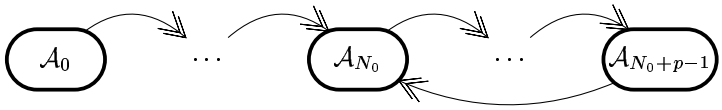
- Montrer que $\frac{1}{2}L = \{u \in \Sigma^*, \exists v \in \Sigma^*, |v| = |u| \text{ et } uv \in L\}$ est rationnel.
- Qu'en est-il du langage $\frac{2}{3}L = \{u \in \Sigma^*, \exists v, w \in \Sigma^*, |v| = |w| = |u| \text{ et } vuw \in L\}$ ?

*Pour la correction, je pompe allègrement le source du corrigé du DS 1 de l'option informatique MP/MP* du lycée du Parc : voir le site* `http://perso.wanadoo.fr/stephane.gonnord/`, section Informatique.

Les exemples pris pour illustrer les constructions le sont avec $L_1 = a^*b$ :

- Pour $\frac{1}{2}L$, ç'est un peu plus fin : après avoir lu un mot $u$ de taille $k$ dans $\mathcal{A}$, on aimerait savoir s'il existe un mot de taille $k$ reliant l'état dans lequel on est à un état final de $\mathcal{A}$. On pourrait donc faire des transitions dans des copies $(\mathcal{A}_k)_{k\in\mathbb{N}}$ de $\mathcal{A}$, en remplaçant la transition $(q_1, \alpha, q_2)$ de $\mathcal{A}$ par la transition $(q_1^k, \alpha, q_2^{k+1})$ de $\mathcal{A}^\infty$ (la réunion de tous les $\mathcal{A}_k$) pour tout $k \in \mathbb{N}$, soit schématiquement :

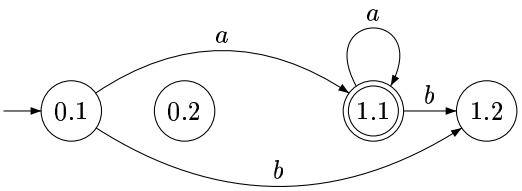

On déclarerait ensuite initial $q_i^0$, et acceptants dans cet "automate infini" les états $q^k$, pour chaque $q \in Q_k = \{q \in Q \mid \exists w \in A^k;\ \delta(q,w) \in F\}$.
Tout le problème est de "replier" cet automate. Pour cela, on note qu'il existe forcément $N_0, p \in \mathbb{N}$ avec $p > 0$ tels que $Q_{N_0} = Q_{N_0+p}$. Mais $Q_{k+1}$ est complètement défini à partir de $Q_k$ (il s'agit des états à l'origine d'une transition arrivant dans un état de $Q_k$). Ainsi, la suite $(Q_n)$ va être cyclique à partir de $N_0$. On peut donc replier notre automate, en ne prenant que les clones $\mathcal{A}_0, ..., \mathcal{A}_{N_0+p-1}$, et en dirigeant les transitions issues de $\mathcal{A}_{N_0+p-1}$ vers $\mathcal{A}_{N_0}$ :




Ainsi, dans ce macro-automate, en lisant $u$ depuis l'état $q_i^0$,on arrive dans

l'état $q^j$, avec $q = \delta(q_i, u)$ et $j$ tel que $Q_j = Q_{|u|}$. Cet automate va donc accepter exactement les $u$ de $\frac{1}{2}L$.

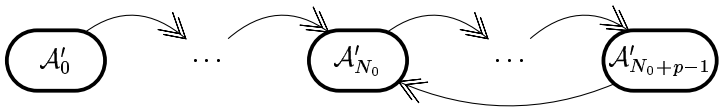
Pour $L_1$, on a $Q_0 = Q = \{2\}$, et $Q_k = \{1\}$ pour tout $k \geq 1$, donc le cycle est assez court, ce qui est heureux !




On a noté $i.j$ l'état $j$ de $\mathcal{A}_i$. Cet automate reconaît clairement $a^+$, ce qui est le résultat souhaité (Yes ...).

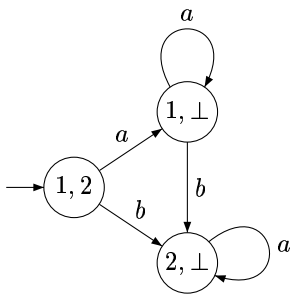
- Pour $\frac{2}{3}L$, ça devient monstrueux, puisqu'il faut remonter le temps en quelque sorte : en lisant $u$ dans notre automate à construire, on veut savoir s'il existe un état $q_1$ de $\mathcal{A}$ accessible en $|u|$ lettres, depuis lequel en lisant $u$, on arrive dans un état "$|u|$-coaccessible". La dernière condition n'est pas un problème. Pour la première, il y a deux problèmes : connaître les états "$|u|$-accessibles", et connaître les $\delta(q, u)$ depuis chacun de ces états.
On va résoudre le premier problème comme pour le problème de co-accessibilité. Pour le second, on va utiliser un automate qui garde en mémoire les mouvements dans $\mathcal{A}$ depuis chaque état (cf le problème de $\sqrt[N]{L}$)

  Pour $k \in \mathbb{N}$, on définit donc $I_k$ l'ensemble des états $k$-accessibles, c'est-à-dire de la forme $\delta(q_i, u)$ pour un certain $u \in A^k$. $I_{k+1}$ est l'ensemble des états accessibles depuis un état de $I_k$ en une transition. Ainsi, pour les mêmes arguments que précédemment, $(I_n)$ est cyclique, et même : $\big((I_k, Q_k)\big)_{k\in\mathbb{N}}$ est cyclique. On va noter $N_0$ et $p$ deux entiers (avec $p > 0$) tels que $(I_{N_0+p}, Q_{N_0+p}) = (I_{N_0}, Q_{N_0})$.
  Avec $Q = [\![1, n]\!]$, notons $\mathcal{A}' = \big(A, [\![1, n]\!]^n, (1, ..., n), \delta'\big)$ le macro-automate (sans état final) regroupant les mouvements dans $\mathcal{A}$ depuis tous les états, en posant $\delta'\big((i_1, ..., i_n), \alpha\big) = \big(\delta(i_1, \alpha), ..., \delta(i_n, \alpha)\big)$.
  On va considérer le macro-(macro-automate) :

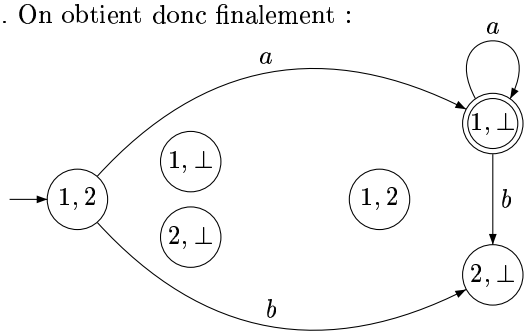



  Il n'y a plus qu'à déclarer initial $(1, ..., n)^0$, et acceptants les états $(i_1, ..., i_n)^k$ tels que il existe un état $q$ $k$-accessible depuis lequel, en lisant $u$, on arrive dans un état $k$-coaccessible, c'est-à-dire : il existe $j \in I_k$ tel que $i_j \in Q_k$.
  Tout est en place pour pouvoir affirmer la tête haute et sans sourciller que cet automate reconnaît clairement $\frac{2}{3}L$.

  Pour $L_1$, on a $I_0 = \{1\}$ et $I_k = \{1, 2\}$ pour tout $k \geq 1$. Par ailleurs le macro-automate est :




avec $\perp$ désignant le puits (ça fait vachement plus informaticien-logicien-pipologue). On obtient donc finalement :




Le langage reconnu est bien $a^+$ (yes yes!!!)