TD langages rationnels — Corrigé

David A. Madore

2 décembre 2016

INF105

Git : `ba099a7 Fri Dec 2 10:10:47 2016 +0100`

Exercice 1.

Soit $\Sigma = \{0, 1\}$. On appelle *mot binaire* un mot sur l'alphabet Σ, et mot binaire *normalisé* un mot binaire qui *soit* commence par 1, *soit* est exactement égal à 0.

(1) Montrer que le langage $L_n = \{0, 1, 10, 11, 100, 101, \ldots\}$ des mots binaires normalisés est rationnel en exhibant directement une expression rationnelle qui le dénote, et montrer qu'il est reconnaissable en exhibant directement un automate fini qui le reconnaît.

(2) On définit la *valeur numérique* d'un mot binaire $x_{n-1} \cdots x_0$ comme $\sum_{i=0}^{n-1} x_i 2^i$ (où x_i vaut 0 ou 1 et est numéroté de 0 pour le chiffre le plus à droite à $n - 1$ pour le plus à gauche) ; la valeur numérique du mot vide ε est 0.

Parmi les langages suivants, certains sont rationnels. Dire lesquels et justifier brièvement pourquoi ils le sont (on ne demande pas de justifier pourquoi ceux qui ne sont pas rationnels ne le sont pas) :

(a) le langage L_a des mots binaires dont la valeur numérique est paire,

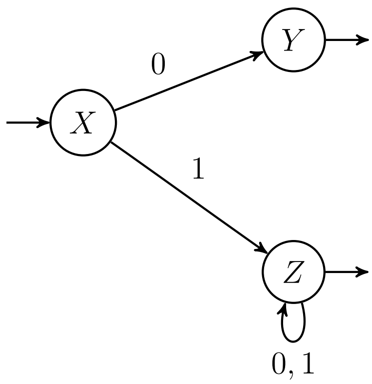
(b) le langage L_b des mots binaires *normalisés* dont la valeur numérique est paire,

(c) le langage L_c des mots binaires dont la valeur numérique est multiple de 3 (indication : selon que n est congru à 0, 1 ou 2 modulo 3, et selon que x vaut 0 ou 1, à quoi est congru $2n + x$ modulo 3 ?),

(d) le langage L_d des mots binaires dont la valeur numérique est un nombre premier,

(e) le langage L_e des mots binaires dont la valeur numérique est une puissance de 2, i.e., de la forme 2^i pour $i \in \mathbb{N}$,

<u>*Corrigé.*</u> (1) On peut écrire $L_n = L_{0|1(0|1)*}$, langage dénoté par l'expression rationnelle $0|1(0|1)*$. Ce langage est reconnu, par exemple, par le DFAI suivant :

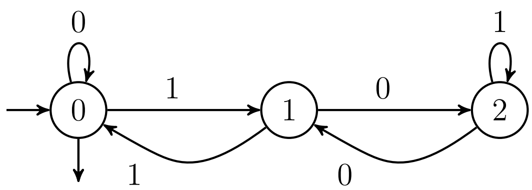


(2) (a) Le langage L_a est rationnel car il s'agit du langage des mots binaires qui soit sont le mot vide soit finissent par 0 : il est dénoté par l'expression rationnelle $\underline{\varepsilon}|(0|1){*}0$. (b) On a $L_b = L_a \cap L_n$ et on a vu que L_a et L_n sont rationnels, donc L_b l'est aussi (on peut aussi exhiber une expression rationnelle qui le dénote : $0|1(0|1){*}0$).

(c) Ajouter un 0 ou un 1 à la fin d'un mot binaire de valeur numérique n le transforme en un mot de valeur numérique $2n + x$ où x est le chiffre affixé. Considérons les six combinaisons entre les trois cas possibles de la valeur numérique n modulo 3 et les deux cas possibles de la valeur de x :

$n \equiv ?\pmod 3$	$x = ?$	$2n + x \equiv ?\pmod 3$
0	0	0
0	1	1
1	0	2
1	1	0
2	0	1
2	1	2

Ceci définit un DFA dont les trois états correspondent aux trois valeurs possibles de n modulo 3, la transition $n \to n'$ étiquetée par x correspond au passage de n à $2n + x$ modulo 3, c'est-à-dire :



(On a marqué l'état 0 comme initial car le mot vide a une valeur numérique congrue à 0 modulo 3, et seul 0 comme final car on veut reconnaître les multiples de 3.)

(d) Le langage L_d n'est pas rationnel (on pourrait le démontrer à l'aide du lemme de pompage, mais ce n'est pas très facile).

(e) Le langage L_e est rationnel car il s'agit du langage dénoté par l'expression rationnelle $0{*}10{*}$. ✓

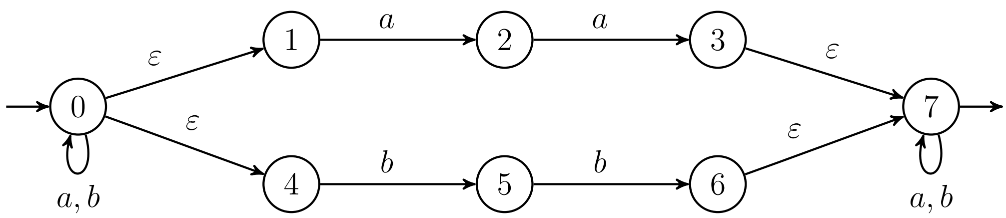
Exercice 2.

Soit $\Sigma = \{a\}$. Montrer que le langage $L = \{a^2, a^3, a^5, a^7, a^{11}, a^{13} \ldots\}$ constitué des mots ayant un nombre *premier* de a, n'est pas rationnel.

Corrigé. Supposons par l'absurde que L soit rationnel. D'après le lemme de pompage, il existe un certain k tel que tout mot de L de longueur $\geq k$ se factorise sous la forme uvw avec (i) $|v| \geq 1$, (ii) $|uv| \leq k$ et (iii) $uv^iw \in L$ pour tout $i \geq 0$. Soit p un nombre premier supérieur ou égal à k (qui existe car l'ensemble des nombres premiers est infini) : le mot $a^p \in L$ admet une factorisation comme on vient de dire. Posons $|u| =: m$ et $|v| =: n$, si bien que $|w| = p - m - n$. On a alors $n \geq 1$ d'après (i), et $|uv^iw| = m + in + (p - m - n) = p + (i - 1)n$ est premier pour tout $i \geq 0$ d'après (iii). En particulier pour $i = p + 1$ on voit que $p + pn = p(n + 1)$ est premier, ce qui contredit le fait qu'il s'agit d'un multiple non-trivial ($n + 1 \geq 2$) de p. ✓

Exercice 3.

On considère l'automate suivant :



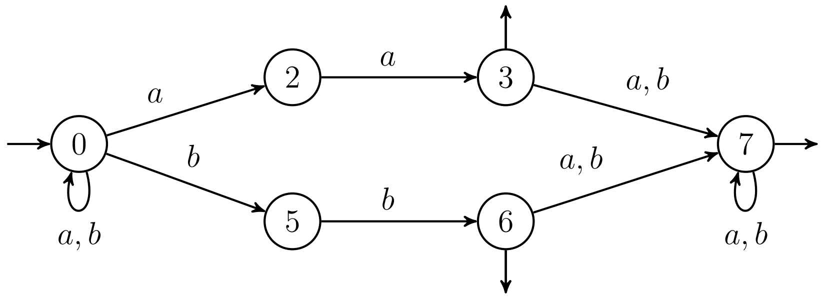
(0) Décrire brièvement le langage accepté par l'automate en question.

(1) Cet automate est-il déterministe ? Si non, le déterminiser.

(2) Minimiser l'automate déterminisé (on doit trouver un DFA ayant quatre états). Décrire brièvement la signification de ces quatre états, de façon à vérifier qu'il accepte le même langage que décrit en (0).

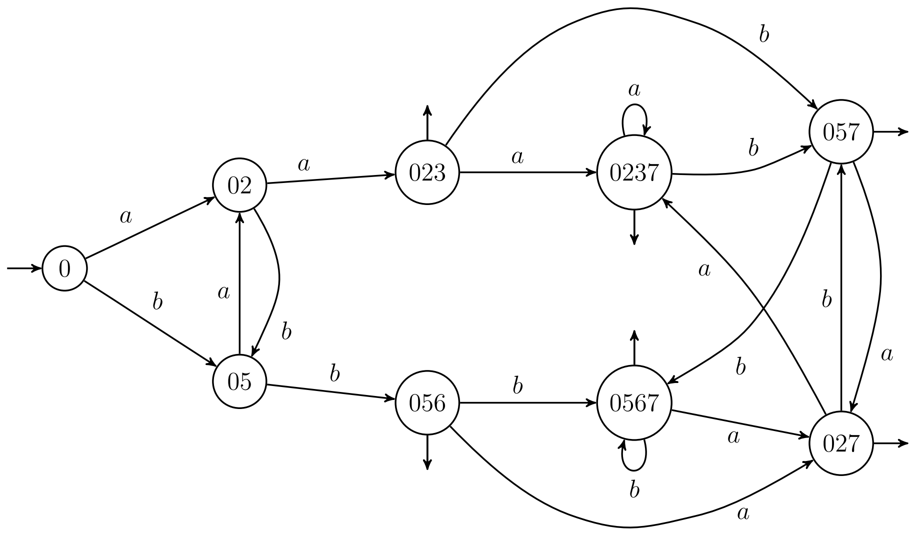
Corrigé. (0) L'automate proposé accepte les mots ayant soit deux a consécutifs (en passant par le chemin $0 \to 1 \to 2 \to 3 \to 7$) soit deux b consécutifs (en passant par le chemin $0 \to 4 \to 5 \to 6 \to 7$).

(1) L'automate ayant des ε-transitions, il ne peut pas être déterministe : on a affaire à un εNFA. Avant de le déterminiser, on élimine ses ε-transitions : la ε-fermeture de 0 est $\{0, 1, 4\}$, celle de 3 est $\{3, 7\}$ et celle de 6 est $\{6, 7\}$; on est amené au NFA suivant :



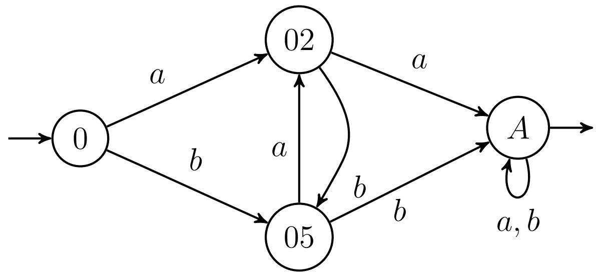
(Les états 1 et 4 étant inaccessibles, ils ont été retirés. Il ne faut pas oublier que 3 et 6 sont finaux puisqu'ils ont l'état final 7 dans leur ε-fermeture.)

On peut maintenant procéder à la déterminisation. Pour abréger les noms des états, on note, par exemple, 023 pour $\{0, 2, 3\}$. En construisant de proche en proche, on obtient le DFA suivant :



(À titre d'exemple, la transition étiquetée b partant de l'état 0237 conduit à l'état 057 car les transitions étiquetées b dans le NFA précédent et partant des états parmi $\{0, 2, 3, 7\}$ sont $0 \to 0$, $0 \to 5$, $3 \to 7$ et $7 \to 7$. Tous les états contenant l'un des symboles $3, 6, 7$ sont finaux.)

(2) On a affaire à un DFA (sous-entendu : *complet*) dont tous les états sont accessibles, on peut donc appliquer directement l'algorithme de Moore. Une première partition sépare les états finaux, soit $023, 0237, 027, 056, 0567, 057$ des non-finaux, soit $0, 02, 05$. L'étape suivante distingue 02 parce que sa a-transition conduit à une classe différente que celles de 0 et 05, et 05 parce que sa b-transition conduit à une classe différente de 0 et 02. Les étapes suivantes ne changent rien. Finalement, on arrive à un automate à quatre états :



où A représente la classe de tous les états finaux de l'automate précédent.

La signification des quatre états est la suivante : l'état 0 signifie que l'automate n'a encore rien lu, l'état 02 signifie que l'automate vient de lire un a, le 05 signifie qu'il vient de lire un b, le A signifie qu'il a lu deux a consécutifs ou bien deux b consécutifs. Sur cette description, il est clair que l'automate accepte les mots contenant deux a consécutifs ou bien deux b consécutifs. ✓