

TD n° 1

1 Automates

1. Étant donné un automate \mathcal{A} , donner un automate \mathcal{A}_* tel que $(L_{\mathcal{A}})^* = L_{\mathcal{A}_*}$

Réponse: On suppose un automate $\mathcal{A} = (Q, \Sigma, \delta, I, F)$. Une construction simple consiste à :

- ajouter les ensembles d'états initiaux à F , ainsi, le mot vide est reconnu.
- soit l'ensemble $P = \{(p, x) \mid x \in \Sigma, \delta(p, x) \cap F \neq \emptyset\}$ (i.e. les prédécesseurs d'un état final, associé au symbole qui permet d'aller dans cet état final). Plus exactement, $\mathcal{A}_* = (Q, \Sigma, \delta', I, I \cup F)$, avec

$$\delta'(p, x) \begin{cases} \delta(p, x) \cup I & \forall (p, x) \in P \\ \delta(p, x) & \text{sinon} \end{cases}$$

2. Montrons que $(a|b)^* = (a^*b^*)^*$.

Réponse: On a deux approches. La première est d'utiliser la sémantique des expressions régulières et de montrer la double implication. La seconde est de construire pour chaque expression un automate, de le déterminer et de le minimiser.

L'autre approche utilise les expressions régulières.

\supseteq Ce sens est simple en remarquant que $(a|b)^* = \Sigma^*$. On a automatiquement que $(a^*b^*)^* \subseteq \Sigma^*$.

\subseteq Supposons que w est reconnu par $(a|b)^*$, montrons qu'il est reconnu par $(a^*b^*)^*$ par induction sur $n = |w|$.

— Cas de base, $n = 0$, c'est à dire $w = \epsilon$. On a bien que $\epsilon \in (a^*b^*)^*$ (à cause de l'étoile de Kleene).

— Supposons $|w| > 0$. Alors w est de la forme $x \cdot w'$, avec $|w'| = n - 1$. Comme $w = x \cdot w' \in (a|b)^*$, on a que $x \cdot w' \in (a|b) \cdot (a|b)^*$ (définition de $*$). Par hypothèse d'induction, $w' \in (a^*b^*)^*$. Avec $x \in (a|b)$, on a bien que $x \in (a^*b^*)$ (soit $x = a$ et on a $a \in a^1b^0$, soit $x = b$ et on a $b \in a^0b^1$). Donc $w = x \cdot w' \in (a^*b^*) \cdot (a^*b^*)^*$ donc $w \in (a^*b^*)^*$.

Pour le deuxième cas on peut aussi procéder par équivalence d'expressions régulières.

— $a \subseteq (a^*b^*)$ (attention, ici a n'est pas un mot mais une expression régulière (qui ne dénote que le mot de une lettre a)).

— $b \subseteq (a^*b^*)$

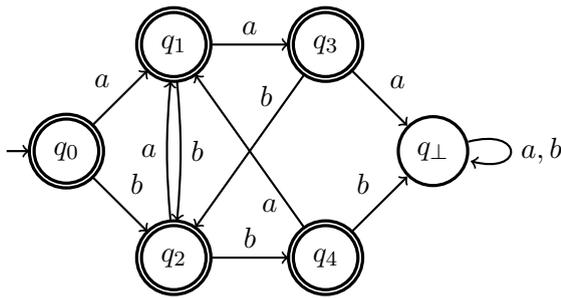
— donc $a|b \subseteq (a^*b^*)$

— donc $(a|b)^* \subseteq (a^*b^*)^*$

Le dernier pas repose sur le fait que $A \subseteq B \Rightarrow A^* \subseteq B^*$ (ce qui se montre simplement par induction sur la taille d'un mot $w \in A^*$). L'intérêt étant que l'on peut montrer cette propriété une fois pour toute.

3. Donner un DFA sur $\Sigma = \{a, b\}$ reconnaissant tous les mots n'ayant pas plus de deux occurrences consécutives de la même lettre :

Réponse: On propose l'automate ci-dessous (complet) :



q_0 permet d'accepter le mot vide (qui est un mot du langage). q_1 et q_2 représentent le fait que l'on vient de lire un a (après un b ou au début du mot) ou un b (après un a ou au début du mot). Les états q_3 et q_4 représentent le fait que l'on a lu deux a de suite ou deux b de suites. Depuis ces états si on continue de lire la même lettre, on passe dans un état puis, si on lit une lettre différente, « on croise » pour aller dans l'état mémorisant le fait qu'on a lu au moins une de ces lettres.

4. On définit le dédoublement d'un mot par : $dd(\epsilon) = \epsilon$ et $\forall x \in \Sigma : dd(v \cdot x) = dd(v) \cdot xx$. Exemple : sur $\Sigma = \{a, b\}$, $dd(aba)$ est $aabbbaa$. Supposons que L est un langage rationnel. Montrer que $dd(L) = \{dd(v) \mid v \in L\}$ est rationnel.

Réponse: Si L est rationnel, alors il est reconnaissable. Il existe donc un DFA minimal $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ reconnaissant L . On considère l'automate $\mathcal{A}' = (Q', \Sigma, \delta', q_0, F')$, avec :

- $Q' = Q \cup \{q'_i \mid q_i \in Q\}$. On appelle q' le doublon de l'état q .
- $\delta'(q, x) = q'$ (on envoie chaque état vers son doublon)
- $\delta'(q', x) = \delta(q, x)$
- $F' = \{q' \mid q \in F\} \cup (\{q_0\} \cap F)$ (la dernière composante permet d'ajouter q_0 dans F' s'il était dans F , i.e. si le mot vide était reconnu par \mathcal{A}).

Considérons un mot $v \in L$. Montrons que $dd(v)$ est reconnu par \mathcal{A}' . Si v est reconnu par \mathcal{A} , il existe un run q_0, \dots, q_k pour $v = x_1, \dots, x_n$. Considérons le mot $dd(v) = x_1, x_1, \dots, x_n, x_n$. Le run $q_0, q'_0, \dots, q_n, q'_n$ est acceptant pour ce mot dans \mathcal{A}' . En effet, $\delta'(q_0, x_1) = q'_0$ (par définition de δ'), puis $\delta'(q'_0, x_1) = \delta(q_0, x_1) = q_1$. Par induction sur n on montre que l'on arrive ainsi en q'_n qui est un état acceptant de \mathcal{A}' .

5. Étant donné un automate \mathcal{A} donner un automate (non déterministe) reconnaissant $L_{\mathcal{A}}^R$ le langage miroir de $L_{\mathcal{A}}$.

Réponse: Soit $\mathcal{A} = (Q, \Sigma, \delta, I, F)$. On pose $\mathcal{A}^R = (Q, \Sigma, \delta^R, F, I)$, avec

$$\delta^R(q, x) = \{q' \mid q \in \delta(q', x)\}$$

On montre que cet automate reconnaît le langage miroir $L_{\mathcal{A}}$. Supposons que $w \in L_{\mathcal{A}}$. Alors il existe un run q_{i_0}, \dots, q_{i_n} reconnaissant $w = x_1 \dots x_n$. Considérons la séquence d'états : q_{i_n}, \dots, q_{i_0} . Celle-ci est un run acceptant pour $\mathcal{A}^R : w^R = x_n \dots x_1$

- $q_{i_n} \in F$ (car c'est un run acceptant pour \mathcal{A}), qui est l'ensemble des états initiaux de \mathcal{A}^R .
- $q_{i_0} \in I$ (car c'est un run pour \mathcal{A}), qui est l'ensemble des états acceptants de \mathcal{A}^R .
- $q_{i_{k-1}} \in \delta^R(q_{i_k}, x_k)$, car $q_{i_k} \in \delta(q_{i_{k-1}}, x_k)$

Par un argument symétrique, on remarque aussi que s'il existe un run acceptant dans \mathcal{A}^R , pour un mot $u = y_1 \dots y_m$ alors on peut construire un run pour le mot $y_m \dots y_1$ dans \mathcal{A} .

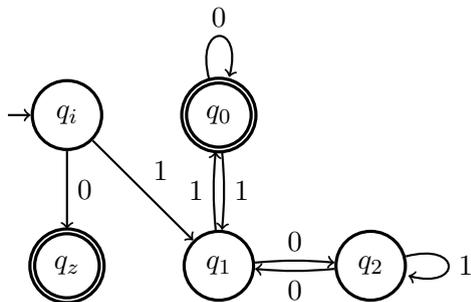
6. On considère l'alphabet $\Sigma = \{0, 1\}$. On considère le langage L_2 comme l'ensemble des mots binaires représentant un multiple de deux (sans zéro non significatif). Ce langage est-il reconnaissable? Même question pour L_3 l'ensemble des mots binaires représentant un multiple de 3 (sans zéro non significatif). Ce langage est-il reconnaissable? Qu'en est-il du langage L_6 des mots binaires représentant un multiple de 6?

Indication pour L_3 on pourra remarquer que pour tout nombre, son reste dans la division par 3 est 0 (multiple de 3), 1 ou 2. Pour chacun de ces cas, on pourra examiner ce que signifie rajouter un 0 ou un 1 à la fin de ce nombre.

Réponse:

- Le langage L_2 peut être décrit simplement par une expression régulière : $0 | 1(0|1)^*0$
- On peut représenter le langage L_3 par un DFA. On aura un état représentant par reste possible. On remarque que si on a lu une suite de chiffre, alors lire un zero ensuite revient à le multiplier par deux et lire un 1 revient à la multiplier par deux et ajouter 1. Si un nombre n est divisible par 3 alors on peut écrire $n = 3k$. Lire un 1 après ce nombre en binaire revient à faire $:3k * 2 + 1 = 6k + 1$, donc le reste devient 1. On a donc les cas :
 - $3k \times 2 = 6k$, reste 0
 - $3k \times 2 + 1 = 6k + 1$, reste 1
 - $(3k + 1) \times 2 = 6k + 2$, reste 2
 - $(3k + 1) \times 2 + 1 = 6k + 3$, reste 0
 - $(3k + 2) \times 2 = 6k + 4 = 3(2k + 1) + 1$, reste 1
 - $(3k + 2) \times 2 + 1 = 6k + 5 = 3(2k + 1) + 2$, reste 2

On obtient alors l'automate :



- Le langage L_6 des multiples de 6 reconnaissable, car c'est l'intersection de L_3 et L_6 .

7. On appelle langage de Dyck un l'ensemble D des mots bien parenthésés sur un alphabet $\{(,)\}$. Par exemple, le mot $((()())(())())$ est bien parenthésé. On peut définir formellement cette propriété. Un mot w est bien parenthésé si :
- pour tout préfixe u de w , le nombre de $)$ dans u est inférieur au nombre de $($
 - il y a autant de $($ que de $)$ dans le mot
- Montrer que D n'est pas un langage rationnel.

Réponse: On utilise le lemme de la pompe. Supposons D rationnel. On sait qu'il existe $p \geq 1$ tel que pour tout $w = xyz$ de longueur au moins p (avec les bonnes conditions), $xy^n z$ est dans D pour tout n . En particulier $|xy| \leq p$ et $y \geq 1$.

Considérons le mot w constitué de p parenthèses ouvrantes suivies de p parenthèses fermantes. On pose $x = \epsilon, y = ({}^p, z =)^p$. Ce mot est de longueur au moins p . On a bien $|xy| = p$ et $|y| \geq 1$. Considérons le mot $xy^2 z$. Ce mot contient $2 \times p$ (et seulement p). Comme $p \geq 1, xy^2 z$ contient strictement plus de parenthèses ouvrantes que de fermantes, donc il n'est pas dans D . Contradiction, D n'est pas rationnel.