Automates & Langages

Septembre 2020

nicolas.fayard@dauphine.eu

Fermeture (Automates *DFA*)

Prouvez que si L est régulier (= S'il existe un DFA pour L) alors $\overline{L}$ est régulier ($w \in \overline{L}$ si $w \notin L$).

Fermeture (Automates *DFA*)

Prouvez que si L est régulier (= S'il existe un DFA pour L) alors $\overline{L}$ est régulier ($w \in \overline{L}$ si $w \notin L$).
Soit A l'automate représentant L : $A = (Q, \Sigma, \delta, s, F)$
On peut construire l'automate $\overline{A}$ de $\overline{L}$: $\overline{A} = (Q, \Sigma, \delta, s, Q \setminus F)$

Fermeture (Automates *DFA*)

Prouvez que si L est régulier (= S'il existe un DFA pour L) alors $\overline{L}$ est régulier ($w \in \overline{L}$ si $w \notin L$).
Soit A l'automate représentant L : $A = (Q, \Sigma, \delta, s, F)$
On peut construire l'automate $\overline{A}$ de $\overline{L}$: $\overline{A} = (Q, \Sigma, \delta, s, Q \setminus F)$

Prouvez que si L_1 et L_2 sont régulier alors $L_1 \cap L_2$ est régulier.

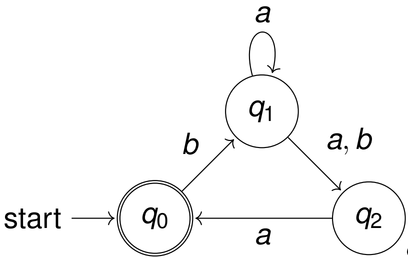
Fermeture (Automates *DFA*)

Prouvez que si L est régulier (= S'il existe un DFA pour L) alors $\overline{L}$ est régulier ($w \in \overline{L}$ si $w \notin L$).
Soit A l'automate représentant L : $A = (Q, \Sigma, \delta, s, F)$
On peut construire l'automate $\overline{A}$ de $\overline{L}$: $\overline{A} = (Q, \Sigma, \delta, s, Q \setminus F)$

Prouvez que si L_1 et L_2 sont régulier alors $L_1 \cap L_2$ est régulier.
Soit $A^1 = (Q^1, \Sigma^1, \delta^1, s^1, F^1)$ et $A^2 = (Q^2, \Sigma^2, \delta^2, s^2, F^2)$
On a $A^1 \cap A^2 = (Q, \Sigma, \delta, s, F)$ avec
$Q = \{(q_1, q_2), \forall q^1 \in Q^1, q^2 \in Q^2\}$
$\Sigma = \Sigma^1 + \Sigma^2$
$\delta := \delta((q^1, q^2), a) = (\delta^1(q^1, a), \delta^2(q^2, a))$
$s = (s^1, s^2)$
$F = (q^1, q^2) | q^1 \in Q^1 \setminus F^1, q^2 \in Q^2 \setminus F^2$

Automate non-Déterministe

$A = (Q, \Sigma, \delta, s, Q \setminus F)$
Déterministe $\delta : Q \times \Sigma \rightarrow Q$ (vers un état).
Non-Déterministe $\delta : Q \times \Sigma \rightarrow 2^Q$ (Ensemble d'états).
Exemple :



$\delta =$

	a	b
q_0	$\{\}$	$\{q_1\}$
q_1	$\{q_1, q_2\}$	$\{q_2\}$
q_2	$\{q_0\}$	$\{\}$

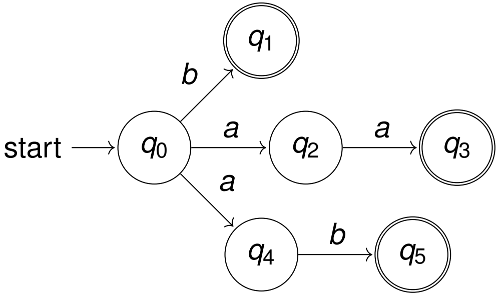
baa ✓, $baaaa$ ✓, a $\times$, ba $\times$.

Automate non-Déterministe - Exercice

1) $L = \{ab, aa, b\}$

Automate non-Déterministe - Exercice

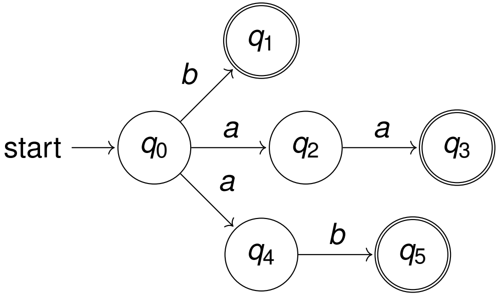
1) $L = \{ab, aa, b\}$



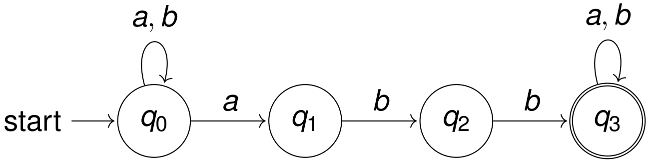
2) $L = \{w | w$ contient le facteur $abb\}$

Automate non-Déterministe - Exercice

1) $L = \{ab, aa, b\}$



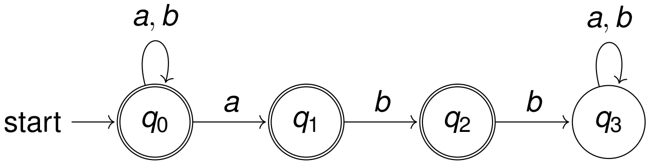
2) $L = \{w | w \text{ contient le facteur } abb\}$



Automate non-Déterministe - Exercice

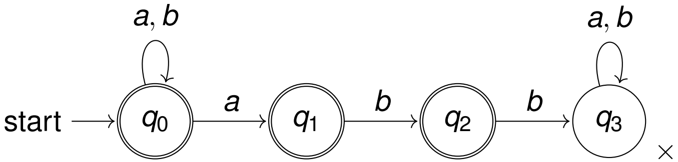
3) $L = \{w | w$ ne contient contient pas le sous-mot $abb\}$

3) $L = \{w | w$ ne contient contient pas le sous-mot $abb\}$



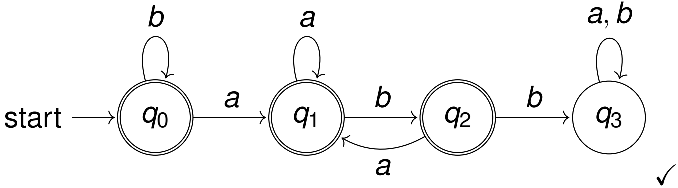
Automate non-Déterministe - Exercice

3) $L = \{w | w$ ne contient contient pas le sous-mot $abb\}$



La fermeture par complémentarité n'est pas toujours vraie pour les automates non- déterministe.

Impossible de le faire avec un automate non-déterministe.

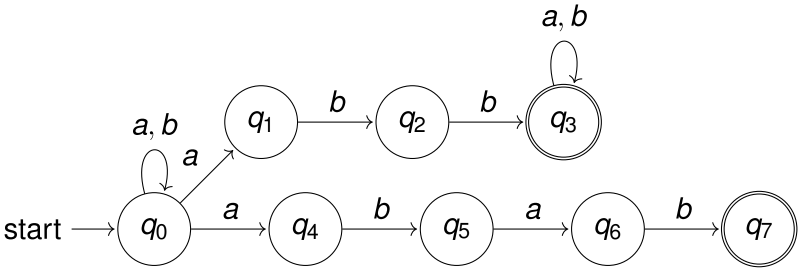


4) $L = \{w | w \text{ contient } abb \text{ ou } abab\}$

4) $L = \{w | w \text{ contient } abb \text{ ou } abab\}$

4) $L = \{w | w \text{ contient } abb \text{ ou } abab\}$

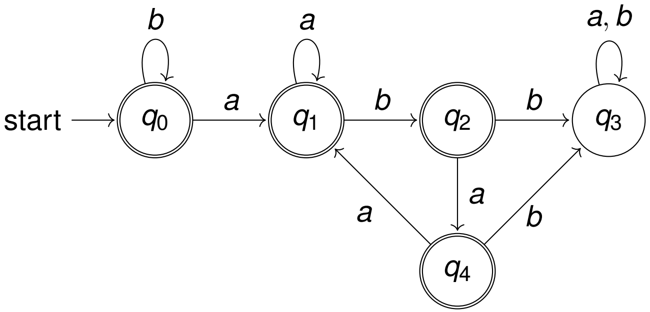
Les *NFA* sont fermés par l'union



Automate non-Déterministe - Exercice

5) $L = \{w | w$ ne contient ni *abb*, ni *abab*$\}$

5) $L = \{w|w$ ne contient ni abb, ni $abab\}$

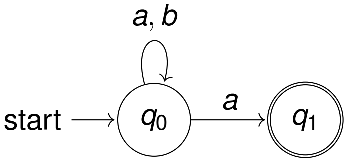


Automate non-Déterministe - Exercice

Donnez un *NFA* et un *DFA* pour les langages suivants :
$L_0 = \{w |$ le caractère à la position $|w|$ est un $a\}$

Donnez un *NFA* et un *DFA* pour les langages suivants :
$L_0 = \{w|$ le caractère à la position $|w|$ est un $a\}$

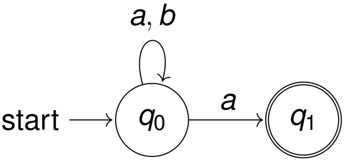
NFA :



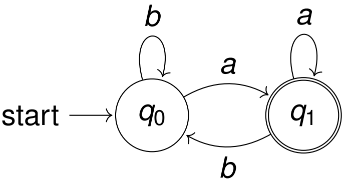
Automate non-Déterministe - Exercice

Donnez un *NFA* et un *DFA* pour les langages suivants :
$L_0 = \{w|$ le caractère à la position $|w|$ est un $a\}$

NFA :



DFA :

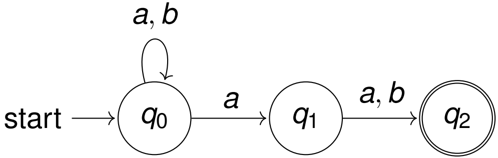


Automate non-Déterministe - Exercice

$L_1 = \{w|$ le caractère à la position $|w| - 1$ est un $a\}$

$L_1 = \{w |$ le caractère à la position $|w| - 1$ est un $a\}$

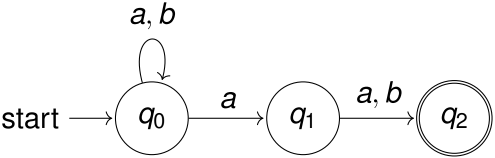
NFA :



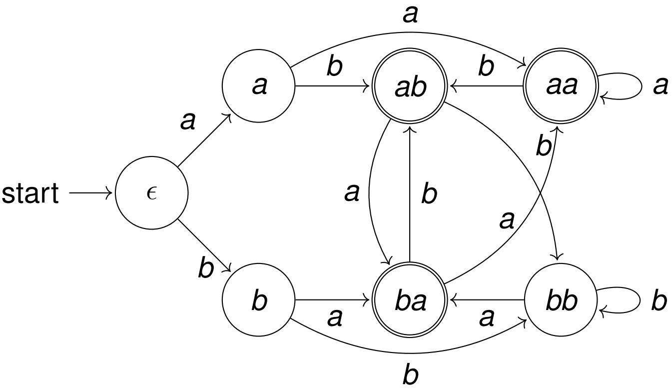
Automate non-Déterministe - Exercice

$L_1 = \{w|$ le caractère à la position $|w| - 1$ est un $a\}$

NFA :



DFA :

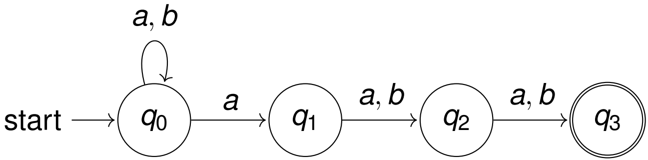


Automate non-Déterministe - Exercice

$L_2 = \{w |$ le caractère à la position $|w| - 2$ est un $a\}$

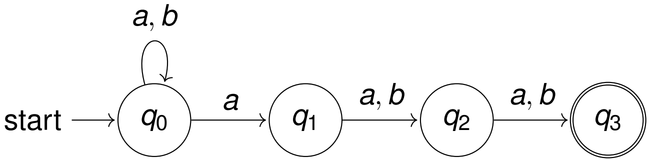
$L_2 = \{w |$ le caractère à la position $|w| - 2$ est un $a\}$

NFA :

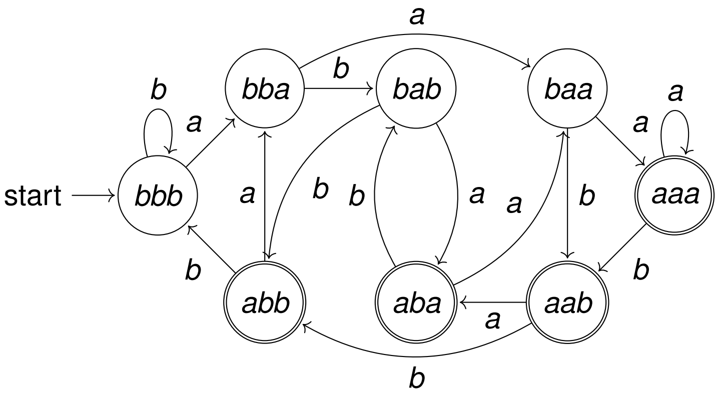


$L_2 = \{w |$ le caractère à la position $|w| - 2$ est un $a\}$

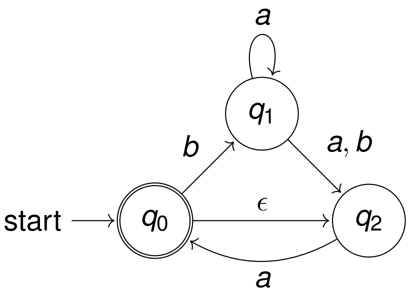
NFA :



DFA :



NFA_ϵ

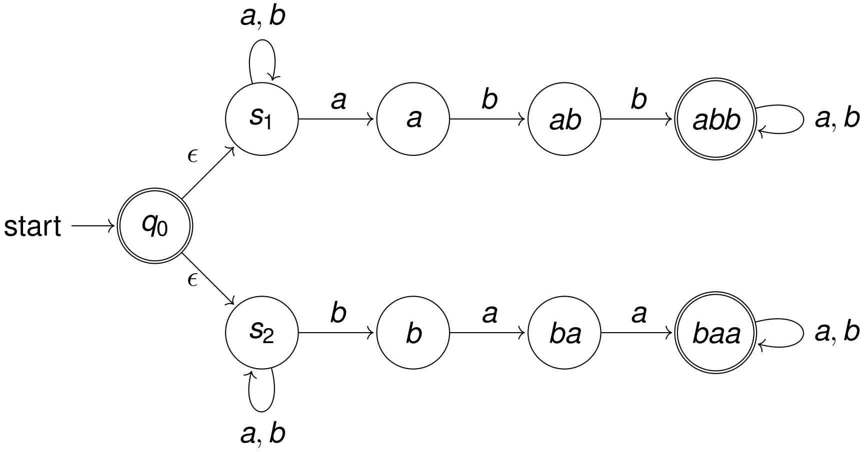


a ✓ , ab ×, $abba$ ✓

NFA_ϵ - Exercice

$L = \{w | w \text{ contient } abb \text{ ou } bba\}$

$L = \{w | w \text{ contient } abb \text{ ou } bba\}$



NFA_ϵ - Fermeture par Union

Montrons que soit deux NFA_ϵ, M_1 et M_2, on peut construire un NFA_ϵ $M = M_1 \cup M_2$.

NFA_ϵ - Fermeture par Union

Montrons que soit deux NFA_ϵ, M_1 et M_2, on peut construire un NFA_ϵ $M = M_1 \cup M_2$.

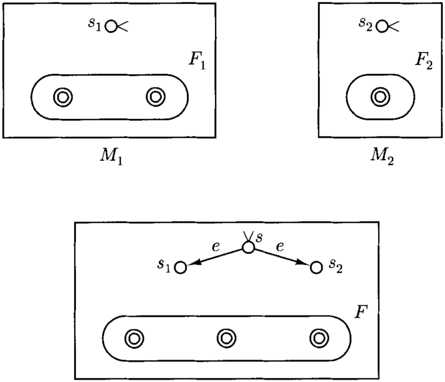


FIGURE – Caption

NFA_ϵ - Fermeture par Union

Montrons que soit deux NFA_ϵ, M_1 et M_2, on peut construire un NFA_ϵ $M = M_1 \cup M_2$.

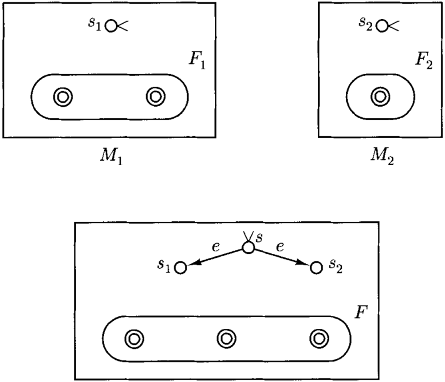


FIGURE – Caption

$$M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$$
$$M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$$
$$M = (Q, \Sigma, \delta, s, F)$$

Avec :
s le nouveau état initial,
$Q = s \cup Q_1 \cup Q_2$,
$F = F_1 \cup F_2$,
$\delta = \delta_1 \cup \delta_2 \cup \{(s, \epsilon, s_1), (s, \epsilon, s_2)\}$.

NFA_ϵ - Fermeture par Concaténation

Montrons que soit deux NFA_ϵ, M_1 et M_2, on peut construire un NFA_ϵ $M = M_1 M_2$.

Montrons que soit deux NFA_ϵ, M_1 et M_2, on peut construire un NFA_ϵ $M = M_1 M_2$.

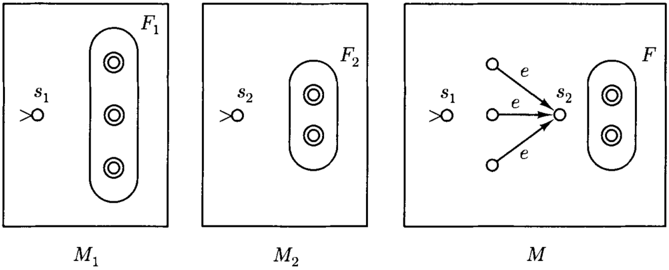


FIGURE – Lewis, H. R., Papadimitriou, C. H. (1998). Elements of the Theory of Computation. ACM SIGACT News.

$M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$
$M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$
$M = (Q, \Sigma, \delta, s_1, F_2)$

Avec :
s le nouveau état initial,
$Q = Q_1 \cup Q_2$,
$\delta = \delta_1 \cup \delta_2 \cup \{(f, \epsilon, s_1), \forall f \in F_1\}$.

NFA_ϵ - Fermeture par Etoile de Kleene

Montrons que soit deux NFA_ϵ, M_1 et M_2, on peut construire un NFA_ϵ $M = M_1^*$.

NFA_ϵ - Fermeture par Etoile de Kleene

Montrons que soit deux NFA_ϵ, M_1 et M_2, on peut construire un NFA_ϵ $M = M_1^*$.

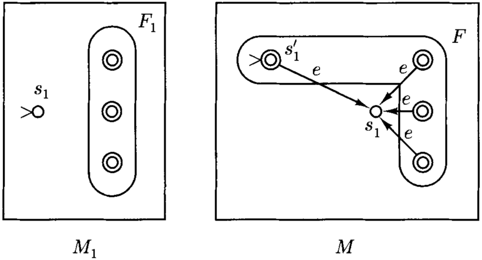


FIGURE – Lewis, H. R., Papadimitriou, C. H. (1998). Elements of the Theory of Computation. ACM SIGACT News.

NFA_ϵ - Fermeture par Etoile de Kleene

Montrons que soit deux NFA_ϵ, M_1 et M_2, on peut construire un NFA_ϵ $M = M_1^*$.

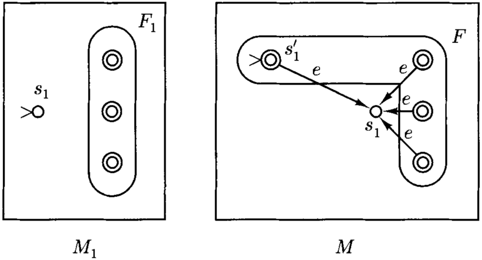


FIGURE – Lewis, H. R., Papadimitriou, C. H. (1998). Elements of the Theory of Computation. ACM SIGACT News.

$M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$
$M = (Q, \Sigma, \delta, s, F)$

Avec :
s le nouveau état initial,
$Q = s \cup Q_1$,
$F = s \cup F_1$,
$\delta = \delta_1 \cup \{(f, \epsilon, s_1), \forall f \in F\}$.

Montrons que soit deux NFA_ϵ, M_1 et M_2, on peut construire un NFA_ϵ $M = I(M_1)$.

NFA_ϵ - Fermeture par inverse

Montrons que soit deux NFA_ϵ, M_1 et M_2, on peut construire un NFA_ϵ $M = I(M_1)$.

$M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$
$I(M_1) = (Q, \Sigma, \delta, s, F)$

Avec :
s le nouveau état initial,
$Q = s \cup Q_1$,
$F = s_1$,
$\delta = \{(k_2, a, k_1), \forall(k_1, a, k_2) \in \delta_1\} \cup \{(s, \epsilon, f), \forall f \in F_1\}$

Minimisation

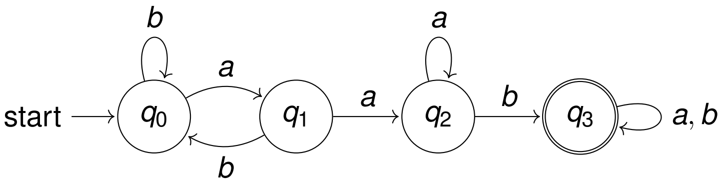
Definition (Equivalence des mots)

Pour un langage L, deux mots $w_1, w_2 \in \Sigma^*$ sont dit :

- Indistinguables (noté $w_1 \sim w_2$) si et seulement si $\forall w' \in \Sigma^*, w_1 w' \Leftrightarrow w_2 w'$. w_1 et w_2 appartienne à la même classe d'équivalence.
- Distinguables (noté $w_1 \nsim w_2$) Si et seulement si $\exists w' \in \Sigma^*, (w_1 w' \in L \cap w_2 w' \notin L) \vee (w_1 w' \notin L \cap w_2 w' \in L)$.

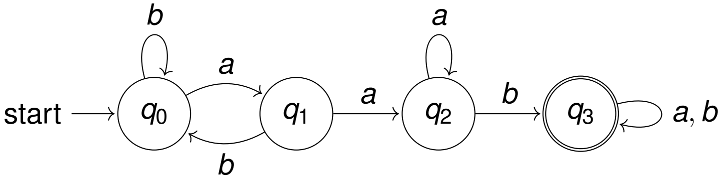
$\forall L$ le nombre d'état minimum d'un automate est égal au nombre de classe d'équivalence de L.

$L = \{w|w \text{ contient } aab\}$. Est ce que l'automate ci-dessous est de taille minimal ?



Minimisation - Exercice

$L = \{w | w \text{ contient } aab\}$. Est ce que l'automate ci-dessous est de taille minimal ?

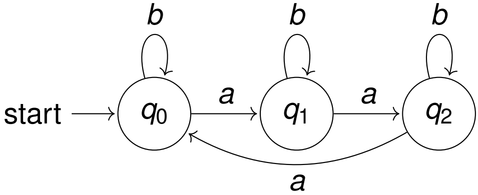


	$w_2 = a$	$w_3 = aa$	$w_4 = aab$
$w_1 = \epsilon$	$w_1 ab \notin L$ $w_2 ab \in L$	$w_1 b \notin L$ $w_3 b \in L$	$w_1 \epsilon \notin L$ $w_4 \epsilon \in L$
$w_2 = a$		$w_2 b \notin L$ $w_3 b \in L$	$w_2 \epsilon \notin L$ $w_4 \epsilon \in L$
$w_3 = aa$			$w_3 \epsilon \notin L$ $w_4 \epsilon \in L$

Oui.

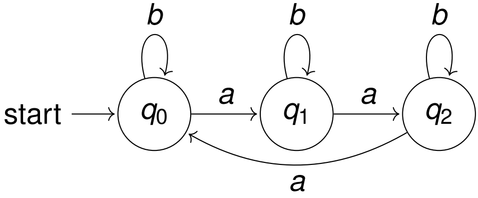
Minimisation - Exercice

$L = \{w | \text{le nombre de } a \text{ est divisible par } 3\}$. Est ce que l'automate ci-dessous est de taille minimal ?



Minimisation - Exercice

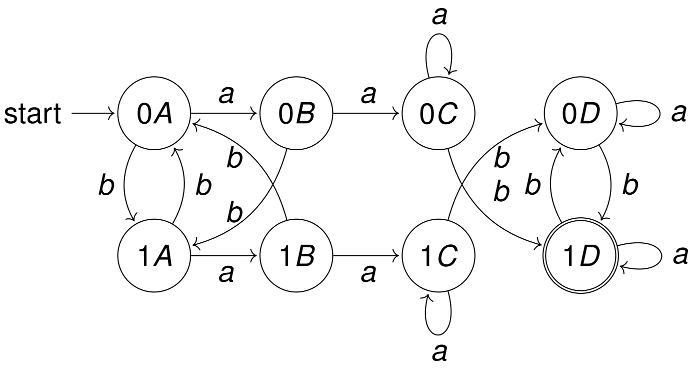
$L = \{w | \text{le nombre de } a \text{ est divisible par } 3\}$. Est ce que l'automate ci-dessous est de taille minimal ?



	$w_2 = a$	$w_3 = aa$
$w_1 = \epsilon$	$w_1\epsilon \in L$ $w_2\epsilon \notin L$	$w_1\epsilon \in L$ $w_3\epsilon \notin L$
$w_2 = a$		$w_2 a \notin L$ $w_3 a \in L$

Oui.

$L = \{w|w$ contient le sous-mot *abb* et le nombre de *b* est impaire$\}$. Est ce que l'automate ci-dessous est de taille minimal ?



Frame Title

	$w_2 = a$	$w_3 = aa$	$w_4 = aab$	$w_5 = b$	$w_6 = ba$	$w_7 = baa$
$w_1 = \epsilon$	$w_1 ab \notin L$ $w_2 ab \in L$	$w_1 b \notin L$ $w_3 b \in L$	$w_1 \epsilon \notin L$ $w_4 \epsilon \in L$	$w_1 aab \in L$ $w_5 aab \notin L$	$w_1 aab \in L$ $w_6 aab \notin L$	$w_1 aab \in L$ $w_7 aab \notin L$
$w_2 = a$		$w_2 b \notin L$ $w_3 b \in L$	$w_2 \epsilon \notin L$ $w_4 \epsilon \in L$	$w_2 aab \in L$ $w_5 aab \notin L$	$w_2 aab \in L$ $w_6 aab \notin L$	$w_2 aab \in L$ $w_7 aab \notin L$
$w_3 = aa$			$w_3 \epsilon \notin L$ $w_4 \epsilon \in L$	$w_3 b \in L$ $w_5 b \notin L$	$w_3 b \in L$ $w_6 b \notin L$	$w_3 b \in L$ $w_7 b \notin L$
$w_4 = aab$				$w_4 \epsilon \in L$ $w_5 \epsilon \notin L$	$w_4 \epsilon \in L$ $w_6 \epsilon \notin L$	$w_4 \epsilon \in L$ $w_7 \epsilon \notin L$
$w_5 = b$					$w_5 abb \notin L$ $w_6 abb \in L$	$w_5 bb \notin L$ $w_7 bb \in L$
$w_6 = ba$						$w_6 bb \notin L$ $w_7 bb \in L$