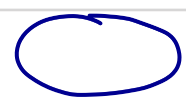
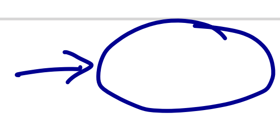
Automates et expressions rationnelles

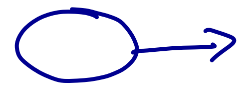
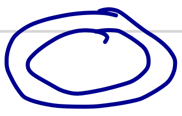
On se fixe un alphabet Σ, un ensemble fini de symboles.
Par exemple $\Sigma = \{a, b\}$

Déf. Un automate (fini) déterministe (AFD) est décrit par :
- un ensemble fini Q d'états
- un état initial $q_0 \in Q$
- un sous-ensemble $F \subseteq Q$ d'états finals
- une fonction de transition $\delta : Q \times \Sigma \to Q$

Représentation graphique

États 

État initial 

État final  ou 

Transition $q_1 \xrightarrow{a} q_2$ pour $\delta(q_1, a) = q_2$

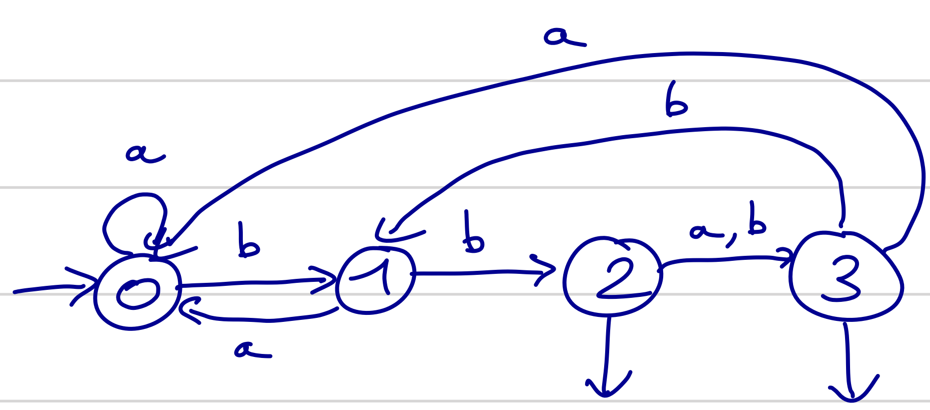
Exemple

$Q = \{0, 1, 2, 3\}$

$q_0 = 0$

$F = \{2, 3\}$

δ	a	b
0	0	1
1	0	2
2	3	3
3	0	1



Déf. Un mot u sur l'alphabet Σ est une suite finie de symboles de Σ, $u = u_0 \dots u_{n-1}$ avec $\forall\, 0 \leq i \leq n-1$, $u_i \in \Sigma$. La longueur d'un mot u, notée $|u|$, est le nombre de symboles n de u. On note par Σ^* l'ensemble de tous les mots sur Σ. On note ε le mot vide, c-à-d. l'unique mot t.q. $|\varepsilon| = 0$.

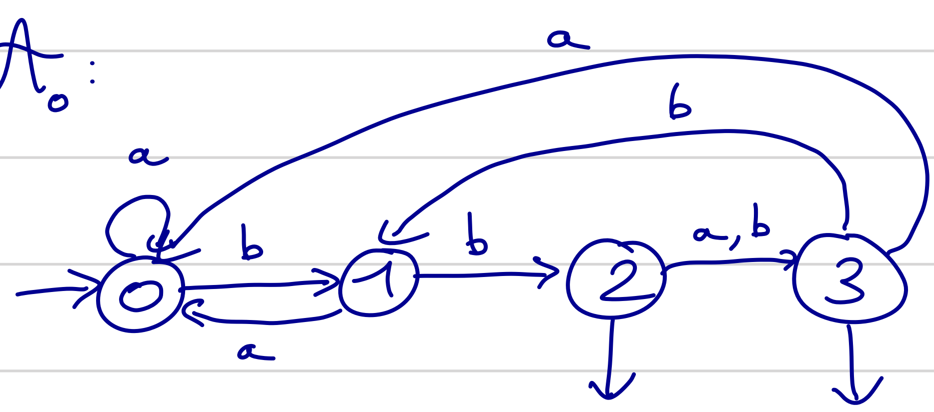
Déf. On dit qu'un AFD (Q, q_0, F, δ) <u>accepte</u> un mot $u \in \Sigma^*$ s'il existe une suite finie d'états $\in Q$ $q_0 \dots q_{|u|-1}$ t.q.

$$\begin{cases} \forall\ 0 \leq i \leq |u|-2 \quad \delta(q_i, u_i) = q_{i+1} \\ q_{|u|-1} \in F \end{cases}$$

On dit qu'un AFD accepte $\left\{\begin{array}{l}\text{le langage}\\ \text{l'ensemble de mots}\end{array}\right.$ $L \subseteq \Sigma^*$ s'il accepte tous les mots de L et n'accepte pas les mots $u \notin L$. On note $\mathcal{L}(\mathcal{A})$ le langage accepté par l'automate $\mathcal{A}$.

<u>Exemple</u>

$\mathcal{A}_0$:



$abb \in \mathcal{L}(\mathcal{A}_0)$ car $(0,0,1,2)$ satisfait les conditions d'acceptation.

$a \notin \mathcal{L}(\mathcal{A}_0)$ car $(0,0)$ est la séquence d'états correspondant aux transitions et $0 \notin F$.

$\underset{0\ 0\ 1\ 0\ 1\ 2\ 3\ 1\ 0}{ababbaba} \notin \mathcal{L}(\mathcal{A}_0)$

$\underset{0\ 0\ 1\ 2\ 3\ 1\ 2\ 3}{abbabbb} \in \mathcal{L}(\mathcal{A}_0)$

$\varepsilon \notin \mathcal{L}(\mathcal{A}_0)$ car $0 \notin F$.

<u>Complexité.</u> Etant donné un automate $\mathcal{A}$ représenté efficace ($O(1)$ pour accès à l'état initial, détermination si un état est final et obtention du résultat de fonction de transition) et un mot u, déterminer si $u \in \mathcal{L}(\mathcal{A})$ peut se faire en $O(|u|)$.

Déf. Un automate (fini) non-déterministe à transitions spontanées (ε-AFND) est décrit par :

- un ensemble <u>fini</u> Q d'états
- un ensemble $I \subseteq Q$ d'états initiaux
- un sous-ensemble $F \subseteq Q$ d'états finals
- une relation de transition $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$

qu'on peut aussi voir comme une fonction (partielle) de transition $\Delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$

<u>Exemple</u>

$Q = \{0, 1, 2, 3\}$

$I = \{0, 1\}$

$F = \{1, 2\}$

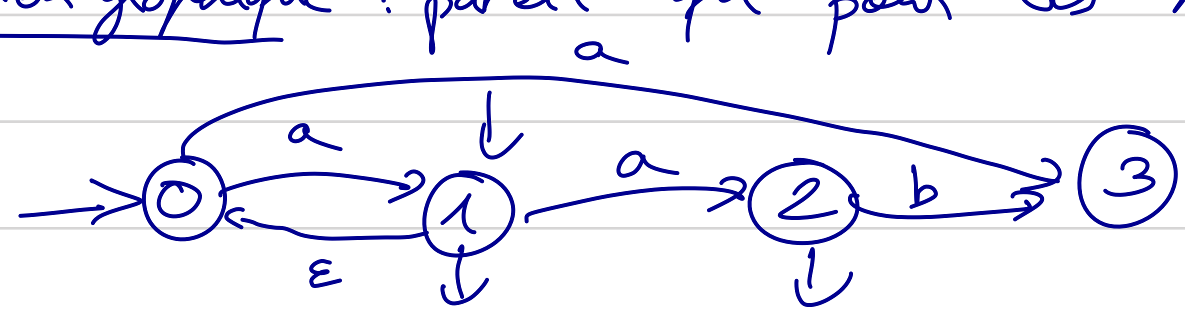
<u>Δ</u>

0, a, 1
0, a, 3
1, a, 2
2, b, 3
1, ε, 0

<u>ou</u>

	a	b	ε
0	$\{1,3\}$	$\emptyset$	$\emptyset$
1	$\{2\}$	$\emptyset$	$\{0\}$
2	$\emptyset$	$\{3\}$	$\emptyset$
3	$\emptyset$	$\emptyset$	$\emptyset$

<u>Représentation graphique</u> : pareil que pour les AFD



Déf. On dit qu'un ε-AFND (Q, I, F, Δ) <u>accepte</u> un mot $u \in \Sigma^*$ s'il existe une suite finie d'états $\in Q$ $q_0 \dots q_{|u|-1}$ t.q.

$$\begin{cases} \forall\, 0 \leq i \leq |u|-2 \\ q_0 \in I \\ q_{|u|-1} \in F \end{cases}$$

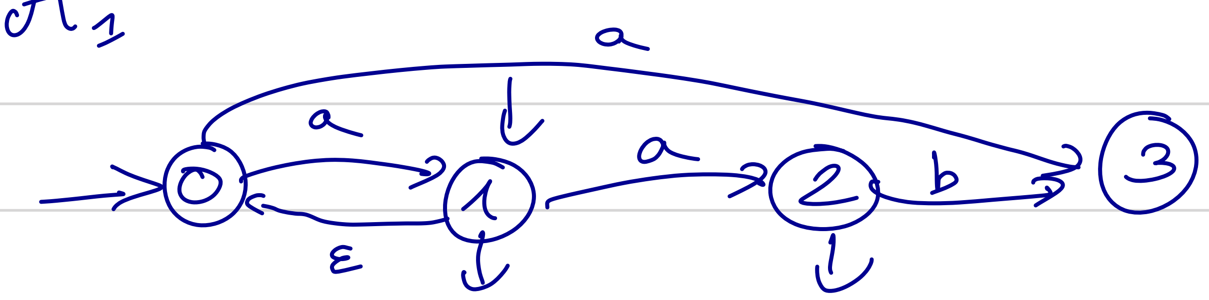
$q_i \xrightarrow{\varepsilon} \dots \xrightarrow{u_i} \dots \xrightarrow{\varepsilon} q_{i+1}$

Il existe une séquence finie d'états $(q'_0, \dots, q'_k)$ $\forall j, \exists \alpha \in \{u_i, \varepsilon\}$ $(q'_j, \alpha, q'_{j+1}) \in \Delta$

avec $q'_0 = q_i$, $q'_k = q_{i+1}$

et u_i n'apparaît qu'une seule fois

Ex. $\mathcal{A}_1$



$\varepsilon \in \mathcal{L}(\mathcal{A}_1)$

$aaa \in \mathcal{L}(\mathcal{A}_1)$

$aab \notin \mathcal{L}(\mathcal{A}_1)$

$0 \xrightarrow{a} 3$ bloqué
$0 \xrightarrow{a} 1 \xrightarrow{\varepsilon} 0 \xrightarrow{a} 3$ bloqué
$0 \xrightarrow{a} 1 \xrightarrow{\varepsilon} 0 \xrightarrow{a} 1 \xrightarrow{\varepsilon} 0$ bloqué
$0 \xrightarrow{a} 1 \xrightarrow{a} 2 \xrightarrow{b} 3 \notin F$

Prop. (déterminisation) : Pour tout ε-AFND $\mathcal{A}$, il existe un AFD $\mathcal{A}'$ t.q. $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$. Par ailleurs, $\mathcal{A}'$ peut se construire à partir de $\mathcal{A}$ en temps $\Theta(2^{|\mathcal{A}|} \times |\mathcal{A}|)$

Démonstration

On pose $\mathcal{A} = (Q, I, F, \Delta)$.

On définit $\mathcal{A}' = (\mathcal{P}(Q), I, F', \delta')$

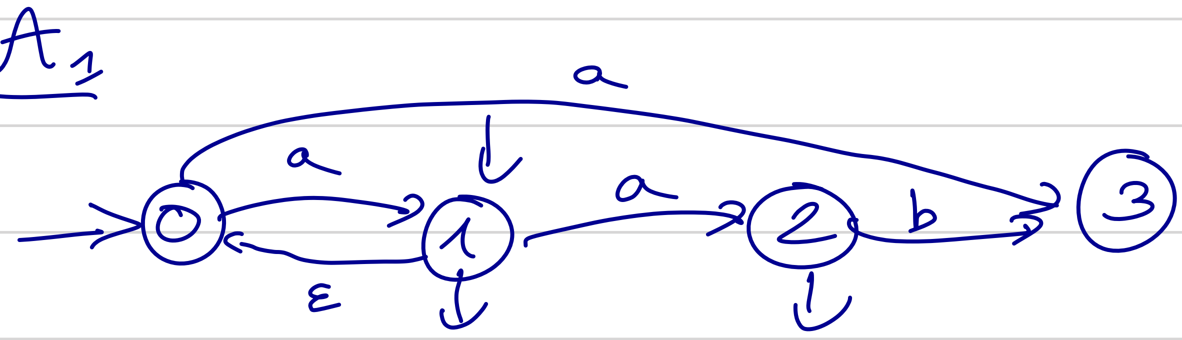
avec : $F' = \{X \subseteq Q \mid \exists q \in X$ t.q. soit $q \in F$ soit il existe une suite finie de transitions spontanées de q vers $q' \in F\}$

$$\underset{\in \mathcal{P}(Q)}{\delta'(X}, \underset{\in \Sigma}{\alpha)} = \{q \in Q \mid \exists q' \in X, \; q' \xrightarrow{\varepsilon} \cdots \xrightarrow{\alpha} \cdots \xrightarrow{\varepsilon} q\}.$$

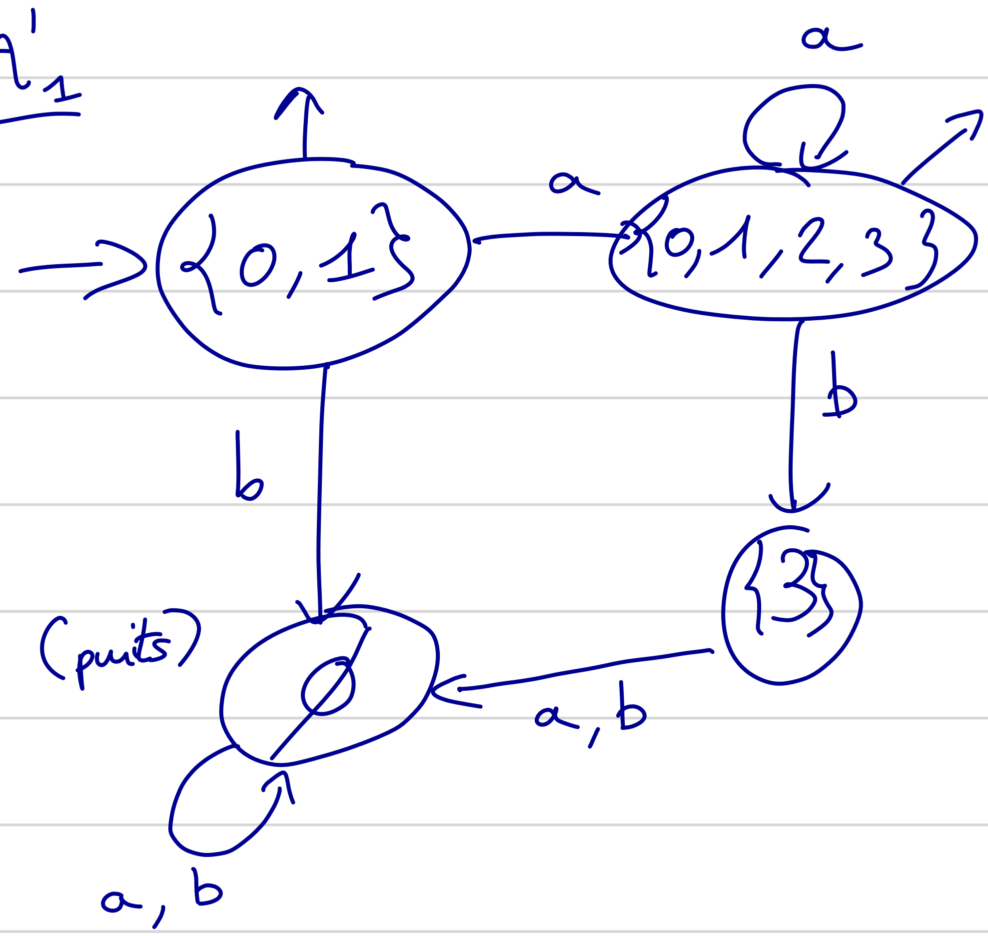
On peut vérifier que $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$. □

Exemple

$\mathcal{A}_1$



$\mathcal{A}'_1$



$\varepsilon \in \mathcal{L}(\mathcal{A}'_1)$

$aaa \in \mathcal{L}(\mathcal{A}'_1)$

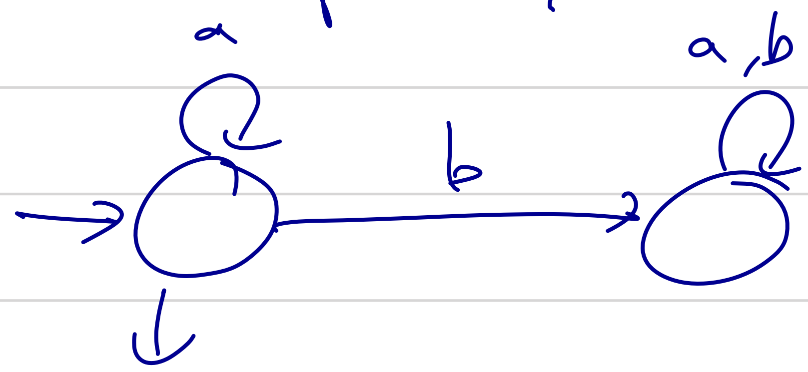
$aab \notin \mathcal{L}(\mathcal{A}'_1)$

$\mathcal{L}(\mathcal{A}'_1) = \mathcal{L}(\mathcal{A}_1)$

$= \{a^n \mid n \geq 0\}$

$\mathcal{A}''_1$

$\mathcal{A}'_1$ est en fait équivalent à l'automate :



Déf Une expression rationnelle (regular expression) [regex(p)] est une expression formelle définie sur un alphabet Σ de manière inductive :

- $\emptyset$ est une e.r.
- ε est une e.r.
- α est une e.r. si $\alpha \in \Sigma$

(ou, union) – $(e_1 \mid e_2)$ est une e.r. si e_1 et e_2 sont des e.r.

(concaténation) – $(e_1 e_2)$ est une e.r. ~~~~~~~~

(étoile de Kleene) – (e^*) est une e.r. si e est une e.r.

Ex. $\Sigma = \{a, b\}$

$\emptyset$, ε, a, b

$(a|b)$

(ab)

$(a|b)ab \equiv (((a|b))a)b$

$(ab)^*|a$

On omet souvent les parenthèses qd il n'y a pas d'ambiguïté.

Déf. Une e.r. e sur Σ dénote un langage sur Σ (un ensemble de mots sur Σ), noté $\mathcal{L}(e)$, défini inductivement par :

- $\mathcal{L}(\emptyset) = \emptyset$
- $\mathcal{L}(\varepsilon) = \{\varepsilon\}$
- $\mathcal{L}(\alpha) = \{\alpha\}$ où $\alpha \in \Sigma$
- $\mathcal{L}((e_1|e_2)) = \mathcal{L}(e_1) \cup \mathcal{L}(e_2)$
- $\mathcal{L}((e_1 e_2)) = \{u \in \Sigma^* \mid u = vw$ où $v \in \mathcal{L}(e_1)$ et $w \in \mathcal{L}(e_2)\}$
- $\mathcal{L}((e^*)) = \{u_1 u_2 \dots u_n \mid n \geq 0, \forall i\ u_i \in \mathcal{L}(e)\}$

$\mathcal{L}((a|b)) = \{a, b\}$

$\mathcal{L}((ab)) = \{ab\}$

$\mathcal{L}((a|b)ab) = \mathcal{L}((((a|b))a)b) = \{aab, bab\}$

$\mathcal{L}((ab)^*|a) = \{a, \varepsilon, ab, abab, ababab, abababab, \dots\}$

Langage des mots avec uniquement des a : a^*

Langage des mots se terminant par 2 a : $(a|b)^* aa$

Extensions syntaxiques

$\bullet \equiv (a_1 | \ldots | a_n)$ où $\Sigma = \{a_1, \ldots, a_n\}$

$[a_1a_2a_3] \equiv (a_1 | a_2 | a_3)$ où $a_1, a_2, a_3 \in \Sigma$

$[\hat{}\, a_1 a_2] \equiv (a_3 | \ldots | a_n)$ où $\Sigma = \{a_1, a_2, a_3, \ldots a_n\}$

$(\sim\sim)+ \equiv (\sim\sim)(\sim\sim)^*$

$(\sim\sim)? \equiv (\sim\sim \mid \varepsilon)$

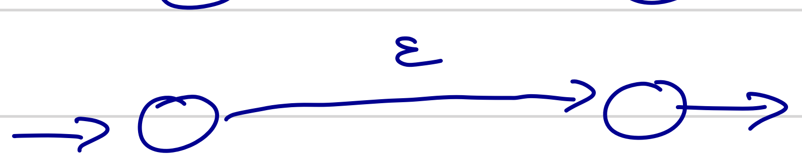
On préfixe les caractères spéciaux par un \.

Thm. : Pour toute e. r. e, il existe un ε-AFND $\mathcal{A}$ t. q. $\mathcal{L}(e) = \mathcal{L}(\mathcal{A})$. De plus cet automate $\mathcal{A}$ peut être calculé en temps linéaire en l'e. r. (et il a exactement 2 fois plus d'états qu'il n'y a de symboles hors parenthèses dans e).

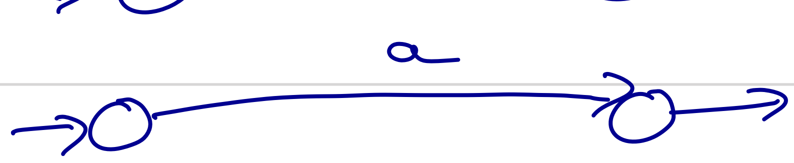
Preuve : Construction de Thompson

$\emptyset$

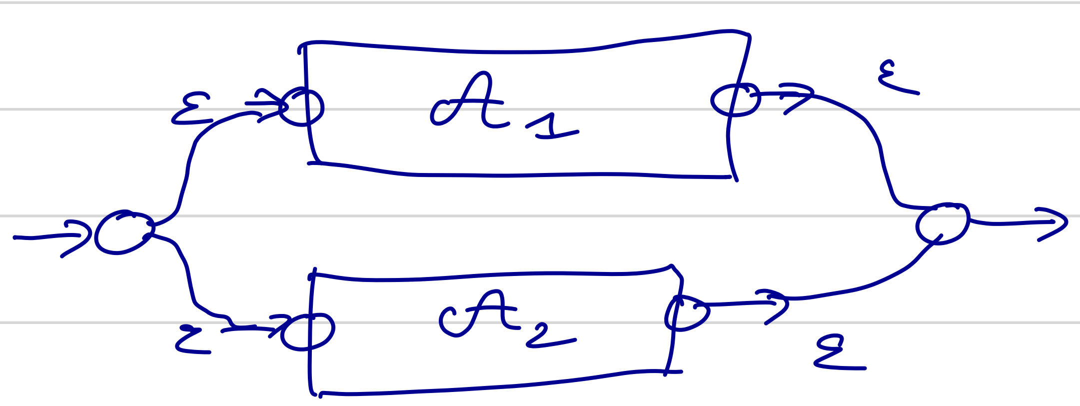
ε



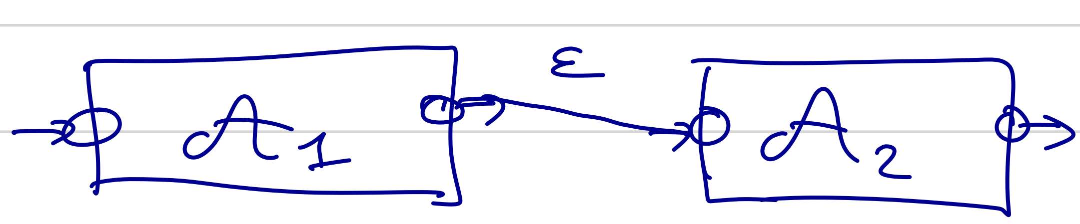
$a \in \Sigma$



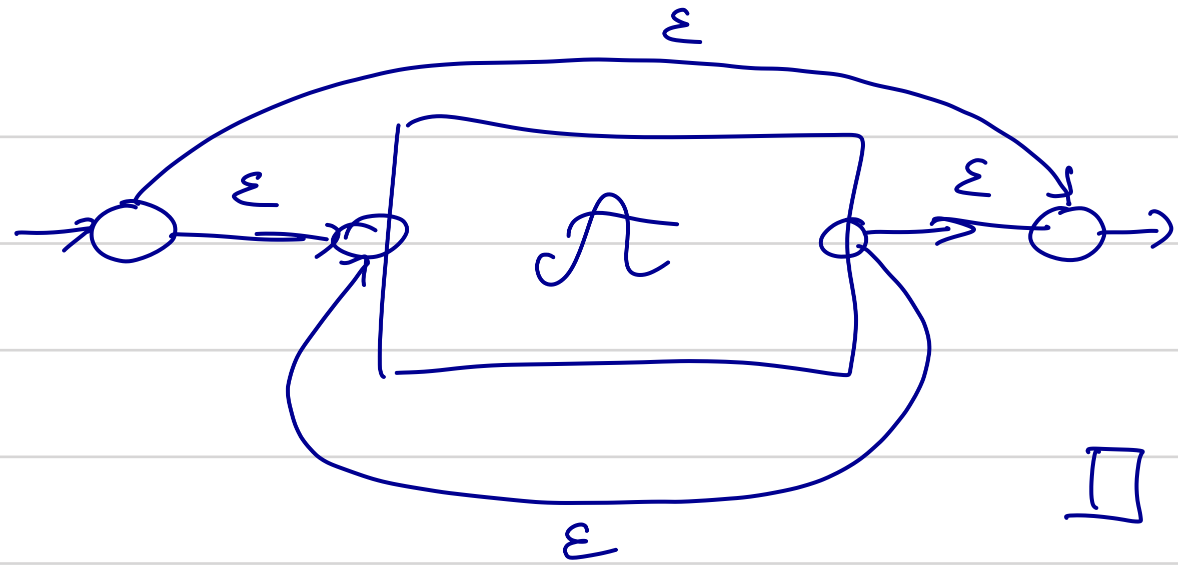
$(e_1 | e_2)$



$(e_1 e_2)$



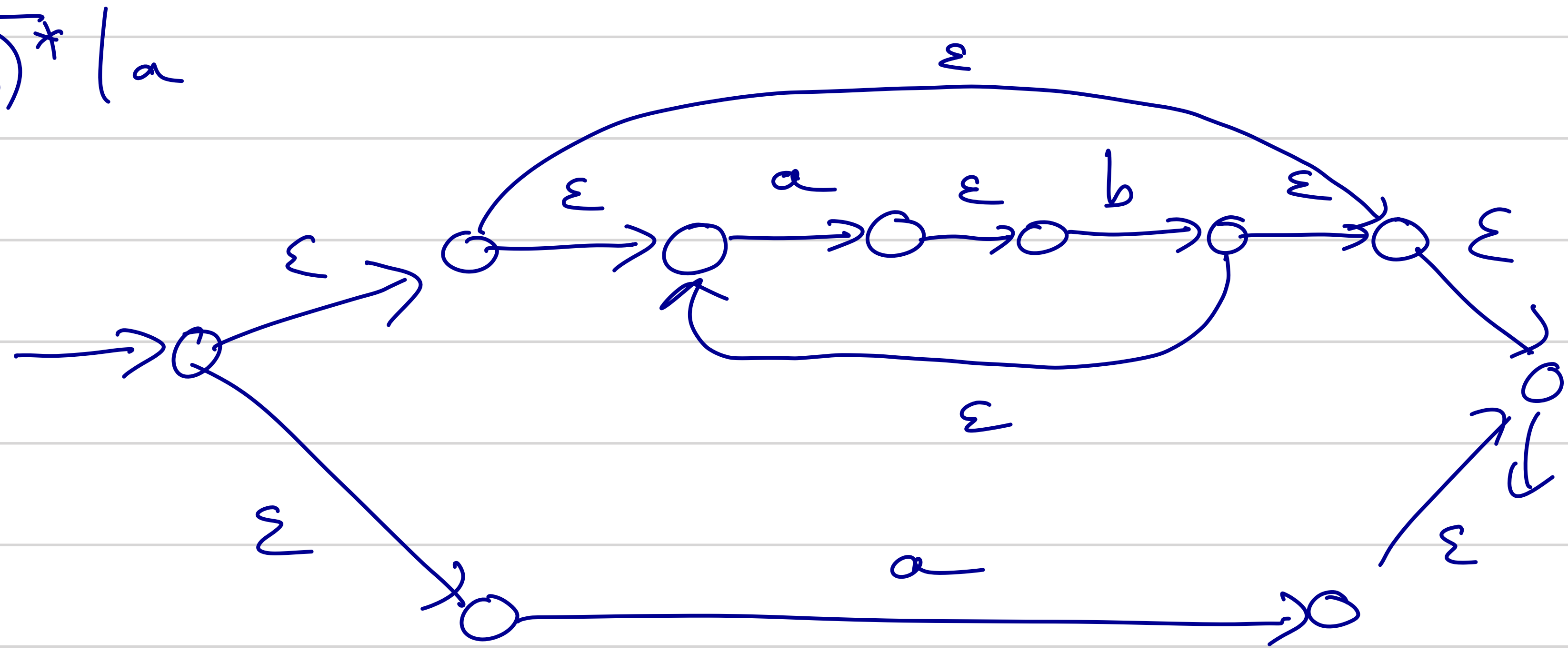
e^*



□

Exemple

$(ab)^* | a$



---

e.r.e

$\downarrow$ $O(|e|)$

ε-AFND $\mathcal{A}$

$\downarrow$ $O(2^{|\mathcal{A}|} \times |\mathcal{A}|)$

AFD $\mathcal{A}'$ intersection, complément

$\downarrow$

AFD minimal

$\xrightarrow{O(|u|)}$ u est accepté ou non

$u \in \Sigma^*$

élimination des états