SÉANCE 9

AUTOMATE À ÉTAT FINI

SUJETS

Algorithme pour créer des AFNs à partir d'expressions régulières

Algorithme pour convertir les AFNs au AFDs

Algorithme pour minimiser les AFDs

Plusieurs exemples….

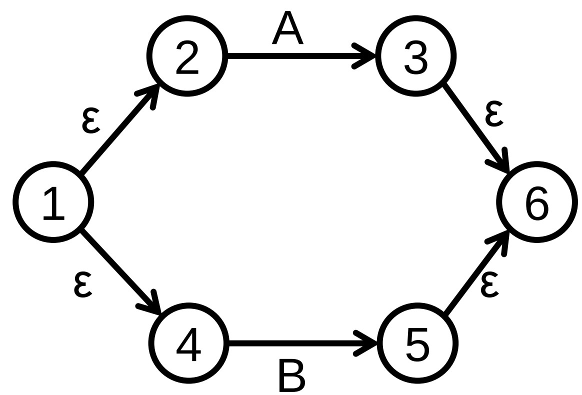
CRÉER DES AUTOMATES FINIS DÉTERMINISTES (AFD)

Afin de créer un AFD, on doit effectuer le suivant:

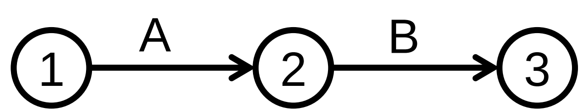
- Créer un automate fini non-déterministe (AFN) à partir d’une expression régulière
- Convertir le AFN en AFD

RÈGLES POUR LA CRÉATION DE AFN

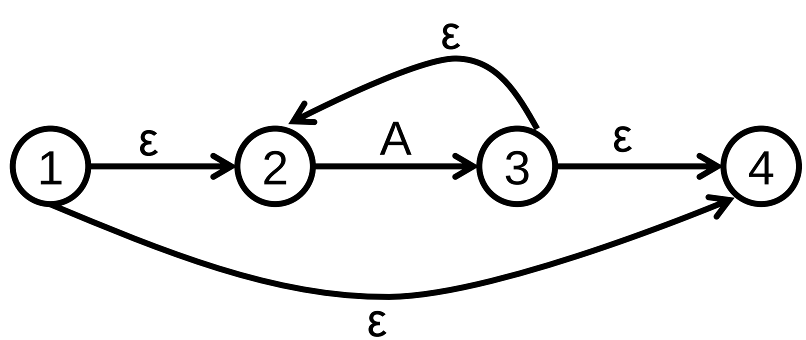
A | B



AB



A*



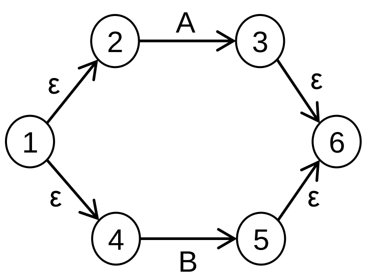
EXEMPLES DE CRÉATION DE AFN

x | yz

Selon les règles de précédence, ceci est équivalent à:

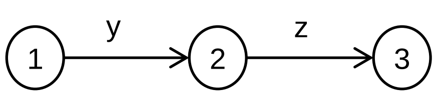
x | (yz)

Ceci possède la même forme que

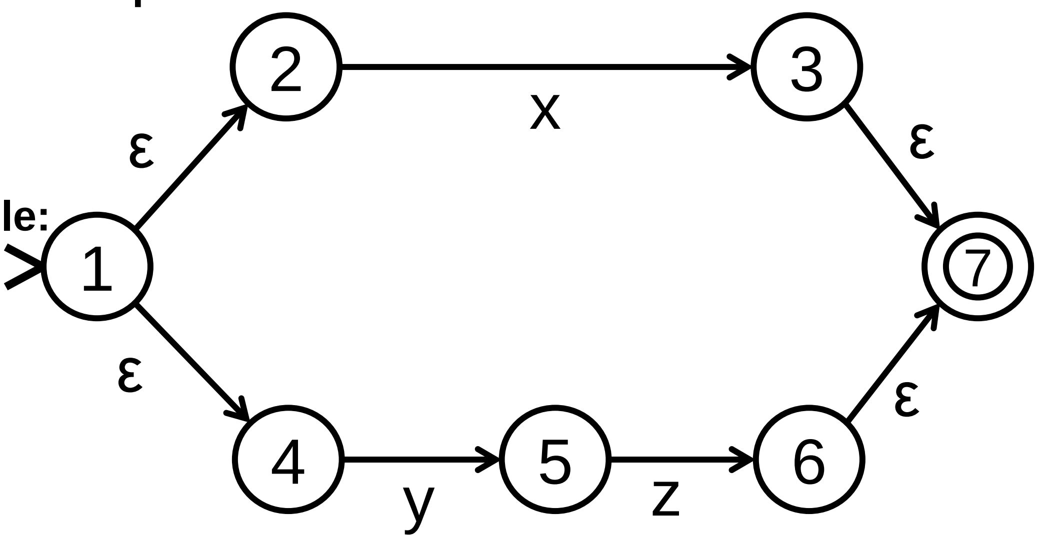


A | B

Et B peut être représenté tel que:



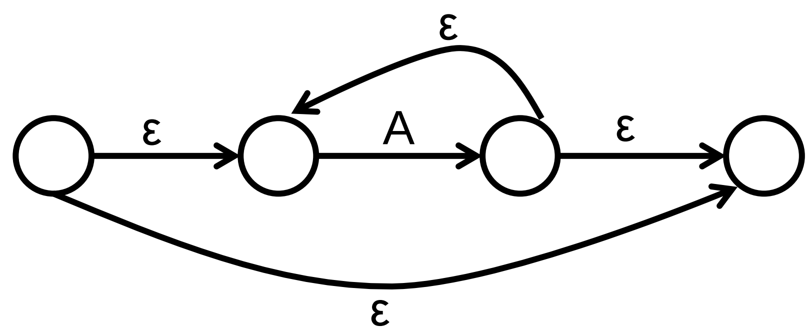
Mettant le tout ensemble:



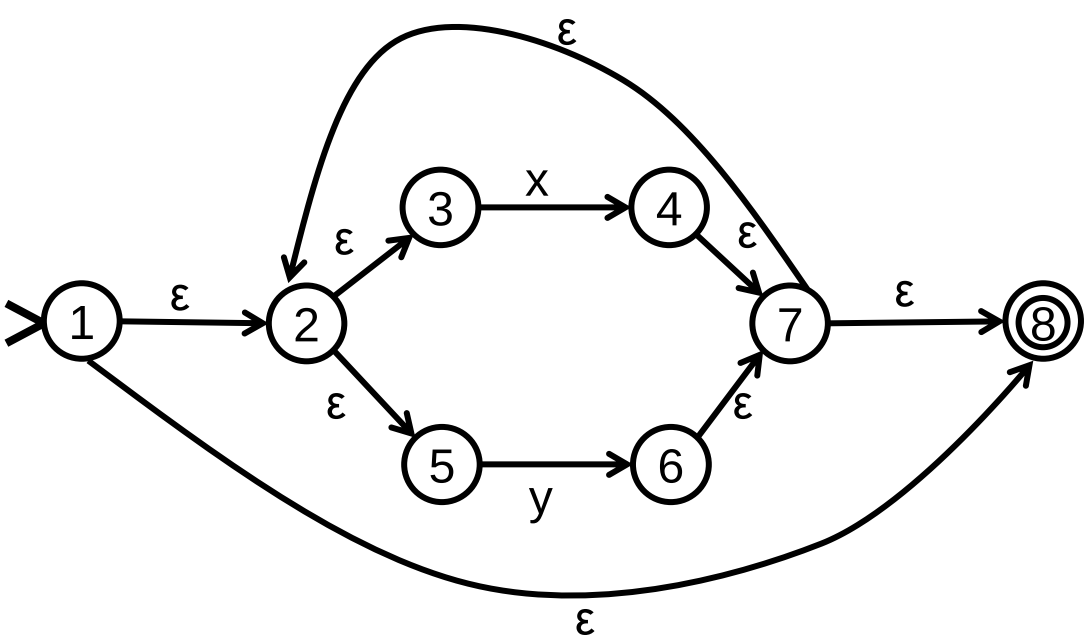
EXEMPLES DE CRÉATION DE AFN

(x | y)*

Nous avons vu A*:

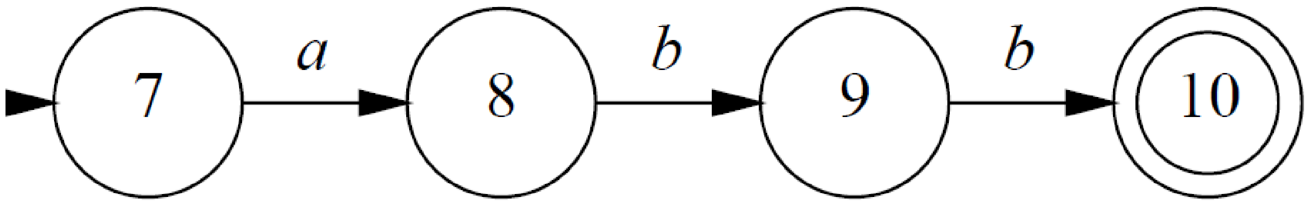


Donc, (x | y)*:



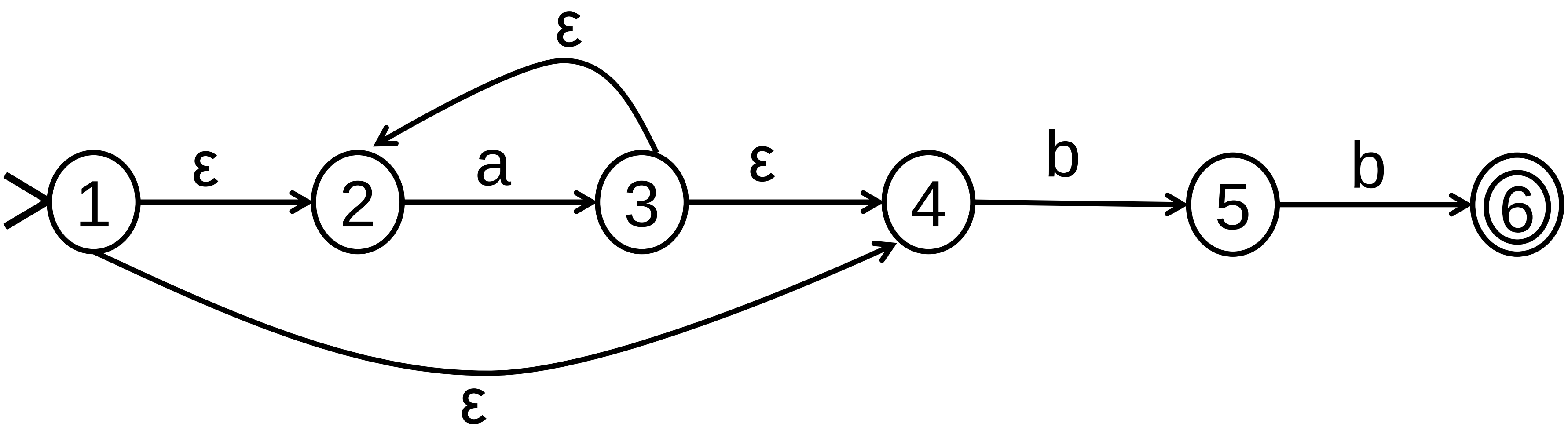
EXEMPLES DE CRÉATION DE AFN

abb



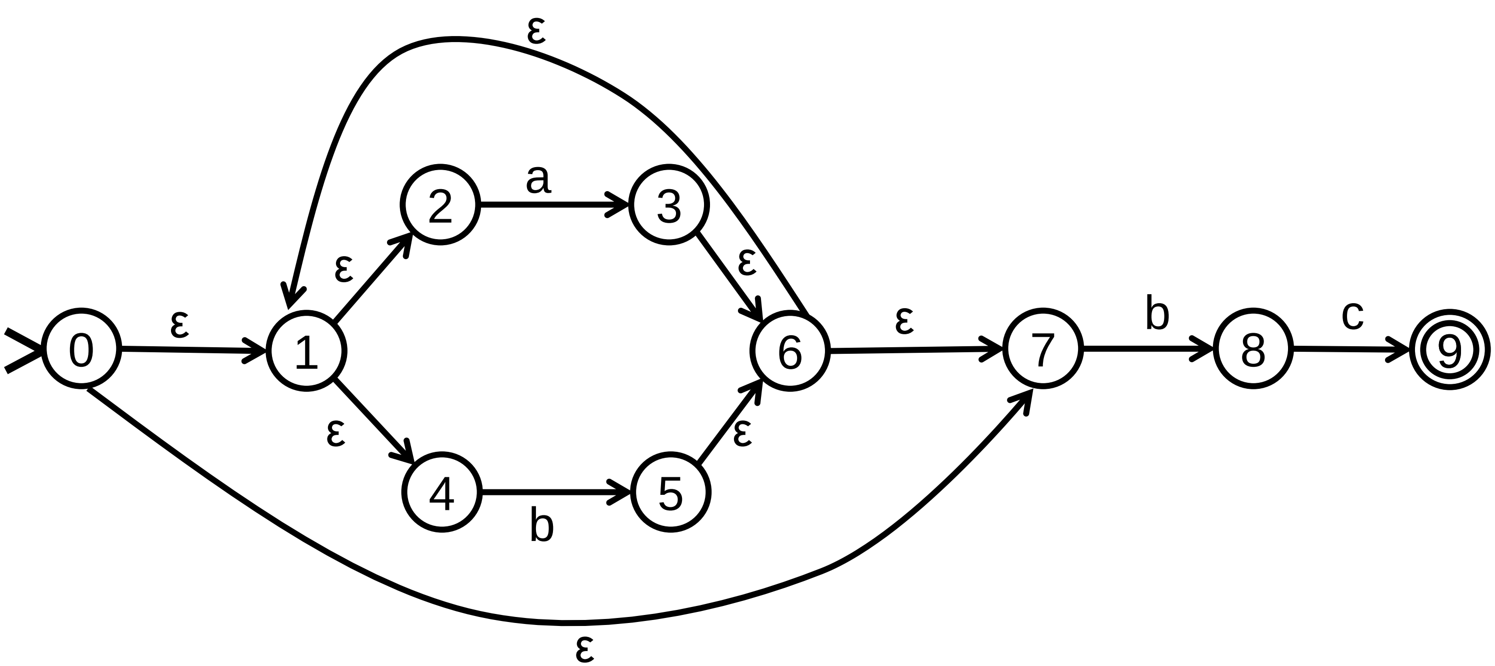
EXEMPLES DE CRÉATION DE AFN

a*bb



EXEMPLES DE CRÉATION DE AFN

(a|b)*bc



CONVERSION D'UN AFN EN UN AFD

AFN est facile à construire mais difficile à interpréter par un ordinateur

- On a besoin de converti un AFN à un AFD
- *L'algorithme de construction de sous-ensembles* réalise cette conversion

Dans le tableau de transition d'un AFN, chaque entrée est un ensemble d'états

Dans le tableau de transition d'un AFD, chaque entrée est un seul état.

Idée générale concernant la conversion d'AFN en AFD: chaque état AFD correspond à un ensemble d'états AFN

ALGORITHME DE CONSTRUCTION DE SOUS-ENSEMBLES

Algorithme: Construction de sous-ensembles – Utilisé pour construire un AFD à partir d'un AFN

Entrée: Un AFN "*N*"

Sortie: Un AFD "*D*" qui accepte le même langage

ALGORITHME DE CONSTRUCTION DE SOUS-ENSEMBLES

uOttawa
L'Université canadienne
Canada's university

Méthode:

- Soit ***s*** un état dans ***"N"*** et ***"T"*** un ensemble d'états, et utilisons les opérations suivantes:

Operation	Definition
ε-closure(s)	Ensembles des états dans ***N*** qui peuvent être atteints de l'état s avec des transitions ε
ε-closure(T)	Ensembles des états dans ***N*** qui peuvent être atteints de l'ensemble des états T avec des transitions ε
move(T,a)	Ensembles des états dans ***N*** qui peuvent être atteints de l'ensemble des états T avec une seule transition a

CONSTRUCTION DE SOUS-ENSEMBLES *(ALGORITHME PRINCIPAL)*

add state $T = \varepsilon$-*closure*(s_0) unmarked to *Dstates*
while $\exists$ unmarked state T in *Dstates*
    mark T
    for each input symbol a
        $U = \varepsilon$-*closure*(*move*(T, a))
        if $U \notin$ *Dstates* **then** add U to *Dstates* unmarked
        Dtrans$[T, a] = U$
    endfor
endwhile

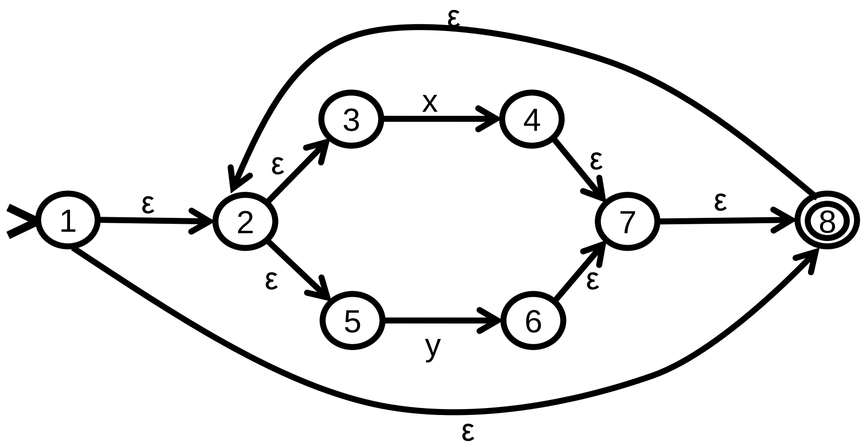
ε-*closure*(s_0) is the start state of D
A state of D is accepting if it contains at least one accepting state in N

CONSTRUCTION DE SOUS-ENSEMBLES
(COMPUTATION DE E-CLOSURE)

```
push all states in T onto stack
initialize ε-closure(T) to T
while stack is not empty
    pop t, the top element off the stack
    for each state u  with an edge from t to u labeled ε
        if u is not in ε-closure(T)
            add u to ε-closure(T)
            push u onto stack
        endif
    endfor
endwhile
```

EXEMPLE DE CONVERSION

Expression Régulière:
(x | y)*



États D={A,B,C}, où

- A = (1,2,3,5,8)
- B = (2,3,4,5,7,8)
- C = (2,3,5,6,7,8)

	x	y
A	B	C
B	B	C
C	B	C

* Dans cette exercice, nous allons faire la conversion d'un AFN

En un AFD.

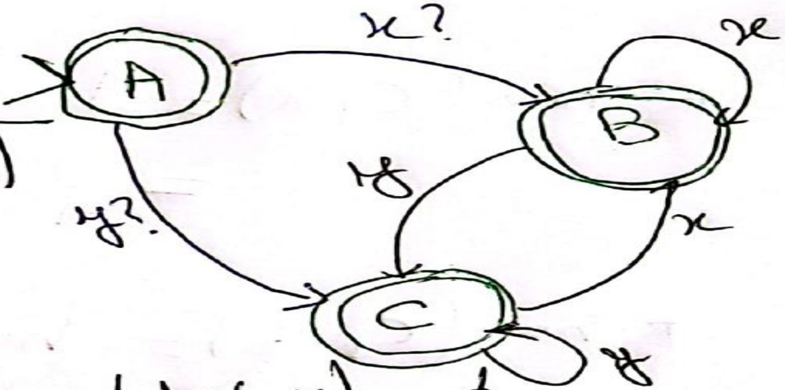
→ Si nous appliquions la méthode vue dans le slide 12.

L'état A de départ : c'est l'ensemble des états qui peuvent être atteints de l'état (AFN) avec des transitions ε.

$\Rightarrow$ * $A = \varepsilon\text{-}f(\{A\}) = \{1, 2, 3, 5, \underline{\underline{8}}\}$ état final accepté

tous les états atteints via ε

$\Rightarrow$ $\boxed{A = \{1, 2, 3, 5, \underline{\underline{8}}\}}$



* $trans(A, x) = trans(\{1,2,3,5,8\}, x)$
  * $trans(\{1\}, x) = \emptyset$
  * $trans(\{2\}, x) = \emptyset$
  * $trans(\{3, x\}) = 4$
  * $trans(\{5\}, x) = \emptyset$; $trans(\{8\}, x) = \emptyset$

$\Rightarrow$ $trans(A, x) = \{4\} = \varepsilon\text{-}f(\{4\})$

tous les états atteints via ε et x

$trans(A, x) = \{4\} = \{7, \underline{\underline{8}}, 2, 3, 4, 5\}$

$= \{2, 3, 4, 5, 7, \underline{\underline{8}}\} = B$

* $trans(A, y) = trans(\{1,2,3,5,8\}, y)$

$= \{6\} = \varepsilon\text{-}f(\{6\}) = \{2, 3, 5, 6, 7, \underline{\underline{8}}\}$

$= C$

Scanné avec CamScanner

* $trans(B, x) = trans(\{2,3,4,5,7,8\}, x)$
$= \{4\} = \varepsilon\text{-}f(\{4\})$
$= \{2,3,4,5,7,\underline{\underline{8}}\} = B$

* $trans(B, y) = trans(\{2,3,4,5,7,8\}, y)$
$= \{6\} = \varepsilon\text{-}f(\{6\})$
$= \{2,3,5,6,7,\underline{\underline{8}}\} = C$

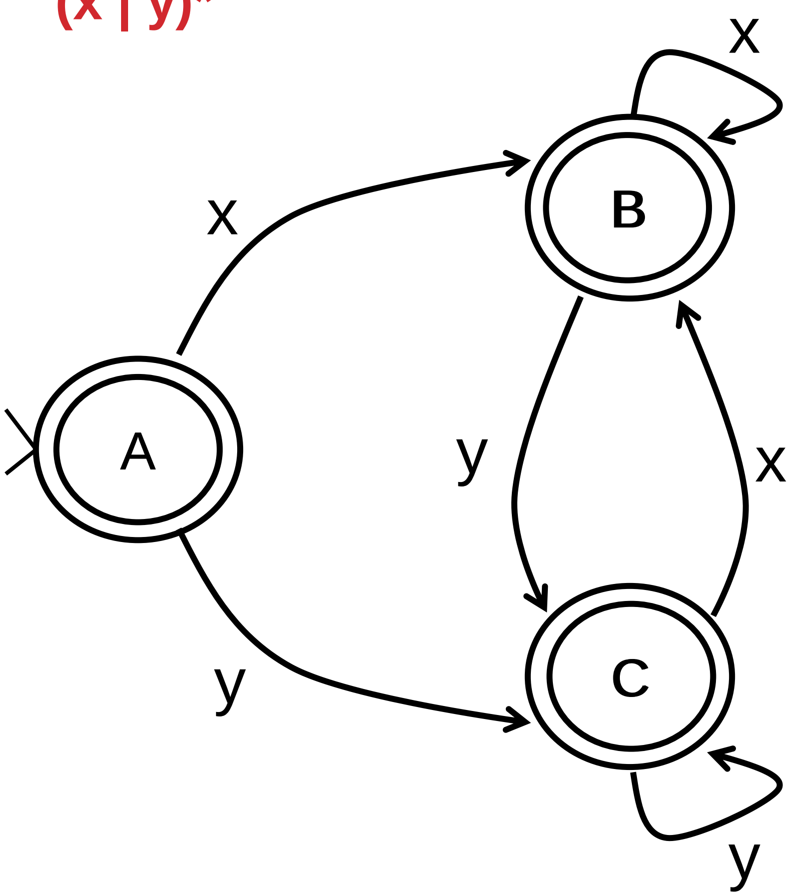
* $trans(C, x) = trans(\{2,3,5,6,7,8\}, x)$
$= \{4\} = \varepsilon\text{-}f(\{4\})$
$= \{2,3,4,5,7,\underline{\underline{8}}\} = B$

* $trans(C, y) = trans(\{2,3,5,6,7,8\}, y)$
$= \{6\} = \varepsilon\text{-}f(\{6\})$
$= \{2,3,5,6,7,\underline{\underline{8}}\} = C$

	x	y
$A = \{1,2,3,5,8\}$	$B = \{2,3,4,5,7,8\}$	$C = \{2,3,5,6,7,8\}$
$B = \{2,3,4,5,7,8\}$	$B = \{2,3,4,5,7,8\}$	$C = \{2,3,5,6,7,8\}$
$C = \{2,3,5,6,7,8\}$	$B = \{2,3,4,5,7,8\}$	$C = \{2,3,5,6,7,8\}$

EXEMPLE DE CONVERSION

Expression Régulière:
(x | y)*

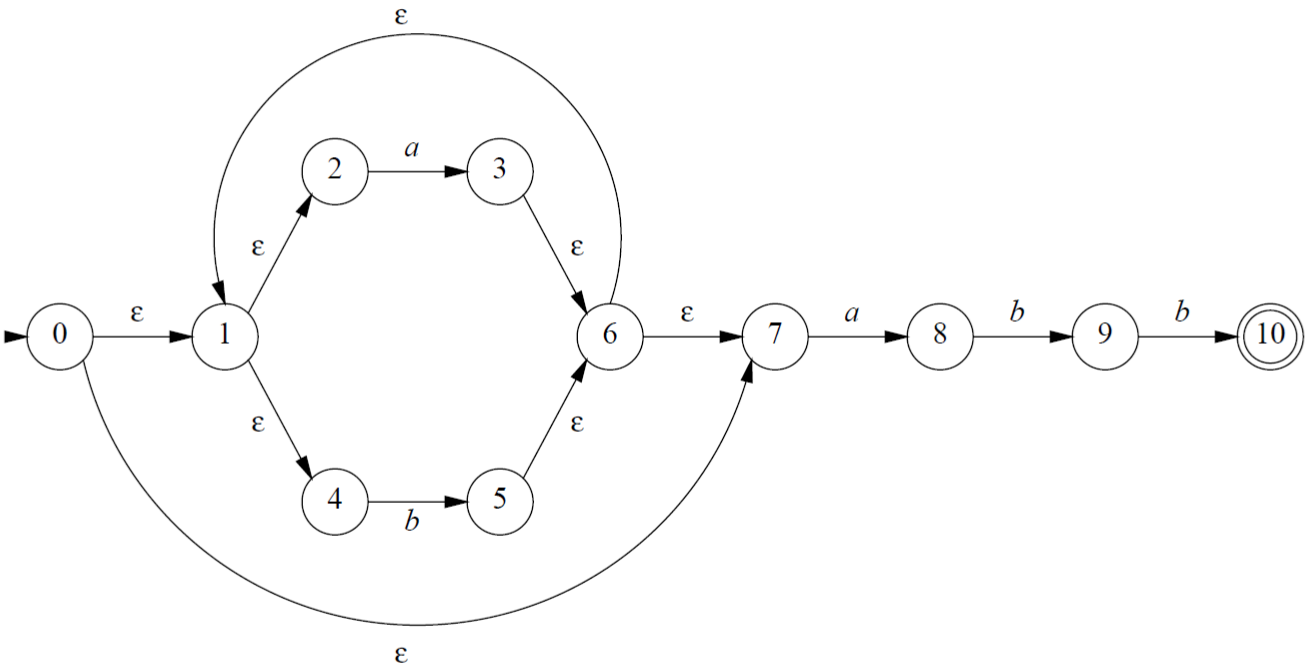


	x	y
A	B	C
B	B	C
C	B	C

AUTRE EXEMPLE DE CONVERSION

Expression Régulière:

(a | b)*abb



AUTRE EXEMPLE DE CONVERSION

Expression Régulière:

(a | b)*abb

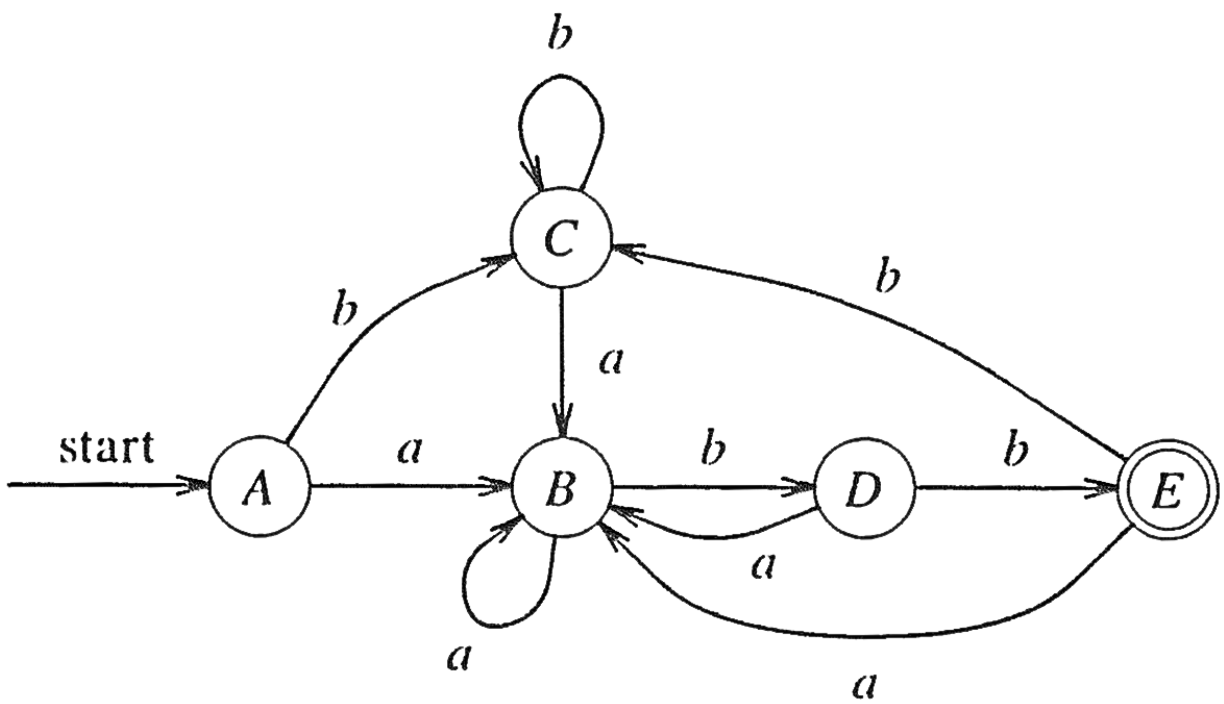
$$A = \{0,1,2,4,7\} \qquad D = \{1,2,4,5,6,7,9\}$$
$$B = \{1,2,3,4,6,7,8\} \qquad E = \{1,2,4,5,6,7,10\}$$
$$C = \{1,2,4,5,6,7\}$$

	a	b
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C

AUTRE EXEMPLE DE CONVERSION

Expression Régulière:

(a | b)*abb



MINIMISER LE NOMBRE D'ÉTATS D'UN AFD

Minimiser le nombre d'états d'un AFD en trouvant tous les groupes d'états qui peuvent être distingués par une chaîne de caractères d'entrée

Chaque groupe d'états qui ne peut pas être distingué est fusionné en un état simple

MINIMISER LE NOMBRE D'ÉTATS D'UN AFD

Algorithme: **Minimiser le nombre d'états d'un AFD**

Entrée: **Un AFD *"D"* avec un ensemble d'états S**

Sortie: **Un AFD *"M"* qui accepte le même langage que *"D"* mais ayant aussi peu d'états que possible**

MINIMISER LE NOMBRE D'ÉTATS D'UN AFD

Méthode:

1. Construire une partition initiale Π de l'ensemble d'états avec deux groupes:
   - Le groupe d'états acceptants
   - Tous les autres groupes d'états
2. Partitionner Π en Π_{new} (en utilisant la procédure montrée dans la diapo suivante)
3. Si Π_{new} != Π, alors **$\Pi = \Pi_{new}$** et répéter l'étape (2). Sinon, aller à l'étape (4)
4. Choisir un état dans chaque groupe de la partition Π comme étant représentant du groupe
5. Enlever les états morts

PROCÉDURE POUR CONSTRUIRE UNE NOUVELLE PARTITION

uOttawa
L'Université canadienne
Canada's university

for chaque groupe G de Π do begin

for chaque symbole "a" ∈ ∑ do begin

Partitionnez G dans des sous-groupes tels que deux états s et t de G sont dans le même sous-groupe si et seulement si

1. **s** et **t** n'ont pas des transitions sur "a", **ou**
2. **s** et **t** ont des transitions sur "a" aux états dans le même groupe dans Π

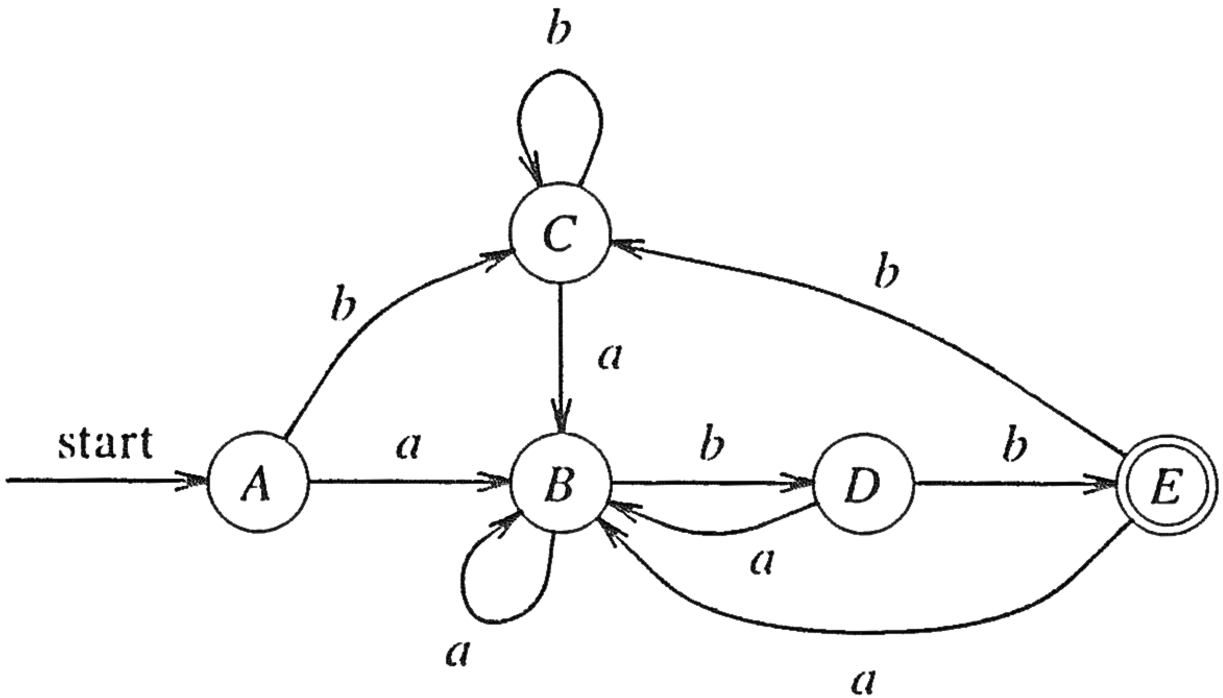
endfor

Remplacez G dans Π_{new} par l'ensemble de tous les sous-groupes formés

endfor

EXEMPLE DE MINIMISATION DE AFD

Minimisez l'automate suivant et dessinez le graphe de l'automate minimal obtenu.



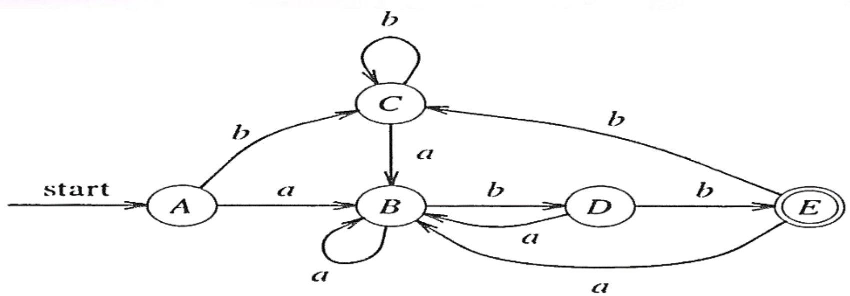
Étape 1 : Déterminer les deux classes (les états finaux et les autres)

$\Rightarrow$ É. Final : {E}

Les autres états : A, B, C, D.

$\rightarrow$ sont tous accessible à partir de l'état initial.

Étape 2 : Dresser un tableau de classement.



Classe obtenue	Les états	a	b
1	E	2	2
2	A	2	2
	B	2	2
	C	2	2
	D	2	1 *

L'état D n'est pas cohérent avec les autres états de la m̂ classe.

* Nous devons séparer l'état D des autres en créant une nouvelle classe 3

Étape 3 : Dresser un tableau avec 3 classes

ÉTAPE 3

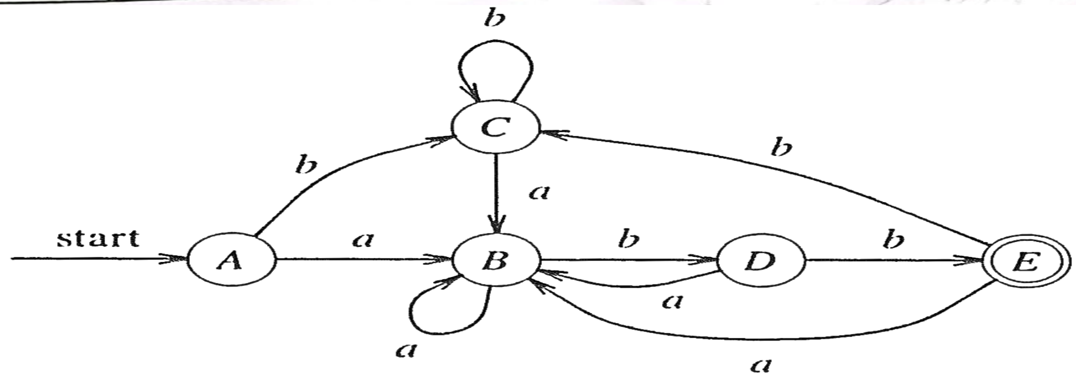
Classe obtenue	États	a	b
1	E	2	2
2	A	2	2
	B	2	3 *
	C	2	2
3	D	2	1

L'état B n'est pas cohérent avec les autres états de la m̂ classe.

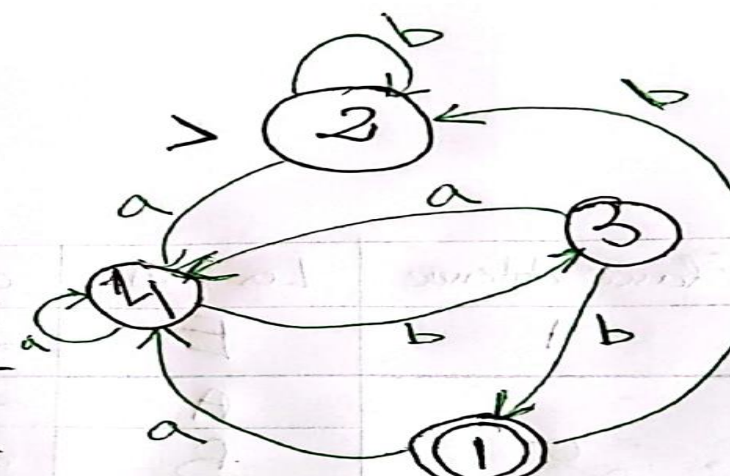
* Nous devons séparer à nouveau l'état B.

Scanné avec CamScanner

Étape 4 : Dresser un tableau avec 4 classes !



classes	États	a	b
1	E	4	2
2	start A	4	2
	C	4	2
3	D	4	1
4	B	4	3



puisque toute les classes sont cohérent ⟹ Nous pourons dessiner le graphe de l'automate minimal obtenu.

MERCI!

QUESTIONS?