Expressions Régulières

- Comment décrire un langage?
- Étant donnée une phrase, appartient-elle au langage?

Théorie des langages :

- Expressions régulières (rationnelles)
- Langages réguliers

Définitions

- On appelle alphabet un ensemble fini A non vide de symboles (lettres de 1 ou plusieurs caractères)
- On appelle mot toute séquence finie d'éléments de A
- On note ε le mot vide
- On note A^* l'ensemble infini contenant tous les mots possibles sur A
- On note A^+ l'ensemble infini des mots non vides possibles sur A alors $A^* = A^+ \cup \{\varepsilon\}$
- On note $|m|$ la longueur du mot m, i.e. le nombre de symboles de A composant le mot m. Rq : $|\varepsilon| = 0$
- On note A^n l'ensemble des mots de A^* de longueur n

Remarque : $A^* = \bigcup_{n=0}^{\infty} A^n$

Exemples

- Soit l’alphabet A={a,b,c} aba, abaa, bbacabb, c, ε sont des mots de A* de longueur respectives 3, 4, 7, 1 et 0
- Soit l’alphabet A={aa,b,c} aba n’est pas un mot de A*, aabaa,caa, bc, aaaa sont des mots de A* de longueur respectives 3, 2, 2 et 2

Définition

- On note **.** l'opérateur de concaténation de 2 mots .
- Si $u=u_1..u_n$ (avec $u_i \in A$) et $v=v_1..v_p$ (avec $v_i \in A$) alors la concaténation de u et v est le mot
  $u.v = u_1..u_nv_1..v_p$
- Le plus souvent, on se passe de l'opérateur de concaténation et l'on note *uv* à la place de *u.v*

Propriétés

- $|u.v| = |u|+|v|$
- Associativité : $(u.v).w = u.(v.w)$
- Élément neutre : $u.\varepsilon = \varepsilon.u = u$

Définition

- On appelle langage L sur un alphabet A tout sous-ensemble de A* : $L \subset A^*$

Exemples :

Soit l'alphabet A = {*a*,*b*,*c*}

- Soit L_1 l'ensemble des mots de A* ayant autant de *a* que de *b* . L_1 est le langage infini {ε , c,cc,ccc,…,ab, ba,…acbccbac… bcbcbcbccbccbacabaaaacacab, …}
- Soit L_2 l'ensemble des mots de A* ayant exactement 3 a . L_2 est ∞ {aaa, acaba,…}

Opérations sur les langages

- Union : $L_1 \cup L_2 = \{ w \text{ tq } w \in L_1 \text{ ou } w \in L_2\}$
- Intersection : $L_1 \cap L_2 = \{ w \text{ tq } w \in L_1 \text{ et } w \in L_2\}$
- Concaténation : $L_1 L_2 = \{ w = w_1 w_2 \text{ tq } w_1 \in L_1 \text{ et } w_2 \in L_2\}$
- Puissance $L^n = \{ w = w_1 \ldots w_n \text{ tq } w_i \in L \text{ pour tout } i \in \{1,\ldots,n\}\}$
- Étoile $L^* = \bigcup_{n\geq 0} L^n$

Langage régulier : Définition

Un langage régulier R sur un alphabet A est défini récursivement de la manière suivante :

- {ε} est un langage régulier sur A
- Si a est une lettre de A, {a} est un langage régulier sur A
- Si R est un langage régulier sur A, alors R^n et R* sont des langages réguliers sur A
- Si R et S sont des langages réguliers sur A, alors R∪S et RS sont des langages réguliers sur A

Expressions Régulières : Définition

Les expressions régulières (E.R.) sur un alphabet A et les langages qu'elles décrivent sont définis récursivement de la manière suivante :

- ε est une E.R. qui décrit le langage {ε}
- Si $a \in A$, alors a est une E.R. qui décrit {a}
- Si r est une E.R. qui décrit le langage R, alors $(r)^*$ et $(r)^+$ sont des E.R. décrivant respectivement R^* et R^+
- Si r et s sont des E.R. qui décrivent respectivement les langages R et S, alors (r)|(s) et (r)(s) sont des E.R. décrivant respectivement $R \cup S$ et RS
- Il n'y a pas d'autres E.R.

Expressions Régulières : Définition

- Si r est une E.R. qui décrit le langage R, alors $(r)^*$ et $(r)^+$ sont des E.R. décrivant respectivement R^* et R^+

Cette propriété s'exprime de la façon suivante :
Cet ensemble est clos opération de Kleene

- Si r et s sont des E.R. qui décrivent respectivement les langages R et S, alors (r)|(s) et (r)(s) sont des E.R. décrivant respectivement $R \cup S$ et RS

Cette propriété s'exprime de la façon suivante :
Cet ensemble est clos par union et concaténation

Remarques

- On convient des priorités décroissantes suivantes :

***. |**

C'est-à-dire par exemple :

ab*|c =((a)((b)*))|(c)

- La concaténation est distributive par rapport à | :

r(s|t) = rs | rt

(r|s)t = rt | st

Exemples

- $(a^*|b^*)^* = ((\varepsilon\ |a)b^*)^* = (a|b)^* = (b|a)^*$ {ε et tous les mots formés de a et de b} = A^* si $A=\{a,b\}$
- $(a)|((b)^*(c)) = a|b^*c$: soit le mot a soit 0 ou plusieurs b suivi d'1 c
- $(a|b)^*abb(a|b)^*$ l'ensemble des mots sur {a,b} ayant comme facteur abb
- $b^*ab^*ab^*ab^*$ l'ensemble des mots sur {a,b} ayant exactement 3 a
- $(abbc|baba)^+aa(cc|bb)^*$ commence par une série non vide de abbc ou/et baba puis un facteur aa et se termine par une série éventuellement vide de bb ou/et de c

Ambiguïté

On dit, par exemple, que (a|b)*a(a|b)* qui décrit tous les mots sur {a,b} ayant au moins un a est ambiguë.

Car on peut appliquer l'E.R. de plusieurs façons :

Sur, par exemple, abaab soit par :

- ε.a.baab avec ε ∈ (a|b)* et baab ∈ (a|b)*
- ab.a.ab avec ab ∈ (a|b)* et ab ∈ (a|b)*
- aba.a.b avec aba ∈ (a|b)* et b ∈ (a|b)*

Par contre l'E.R. b*a(a|b)* décrit le même langage et n'est pas ambiguë :

- abaab = ε.a.baab avec ε ∈ (a|b)* et baab ∈ (a|b)*

Automates à états finis : Définition

Un automate à états finis (A.E.F.) est défini par :

- Un ensemble fini E d'états
- Un état $e_0 \in E$ distingué : l'état initial
- Un ensemble T ($T \subset E$) d'états distingués comme états finaux (ou terminaux)
- Un alphabet Σ des symboles d'entrée
- Une fonction de transition Δ qui, à tout couple formé d'un état de E et d'un symbole de Σ, fait correspondre un ensemble (éventuellement vide) d'états :

$$\Delta(e_i, a) = \{e_{i1}, \ldots, e_{in}\}$$

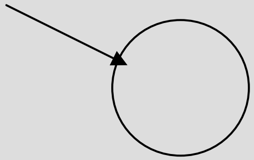
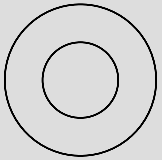
Définition

- Le langage reconnu par un automate est l'ensemble des chaînes qui permettent de passer de l'état initial à un des état terminaux

Exemple

- $\Sigma = \{a,b\}$, $E= \{0,1,2,3\}$, $e_0 = 0$, $T= \{3\}$
- $\Delta(0 , a) = \{0,1\}$, $\Delta(0 , b) = \{0\}$, $\Delta(1 , b) = \{2\}$, $\Delta(2 , b) = \{3\}$; $\Delta(e , l) = \emptyset$ sinon

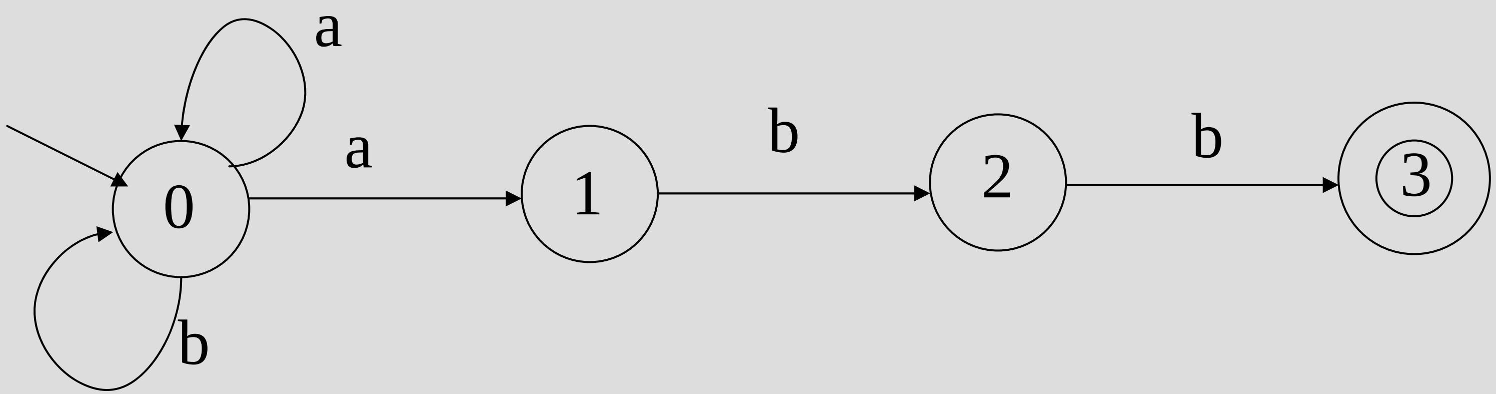
Graphiquement, on représente

- l'état initial par 
- Les états terminaux par 
- Les transitions par 

Exemple

- $\Sigma = \{a,b\}$, $E= \{0,1,2,3\}$, $e_0 = 0$, $T= \{3\}$
- $\Delta(0 , a) = \{0,1\}$, $\Delta(0 , b) = \{0\}$, $\Delta(1 , b) = \{2\}$, $\Delta(2 , b) = \{3\}$; $\Delta(e , l) = \varnothing$ sinon

Représentation graphique



Exemple

- $\Sigma = \{a,b\}$, $E = \{0,1,2,3\}$, $e_0 = 0$, $T = \{3\}$
- $\Delta(0, a) = \{0,1\}$, $\Delta(0, b) = \{0\}$, $\Delta(1, b) = \{2\}$, $\Delta(2, b) = \{3\}$; $\Delta(e, l) = \emptyset$ sinon

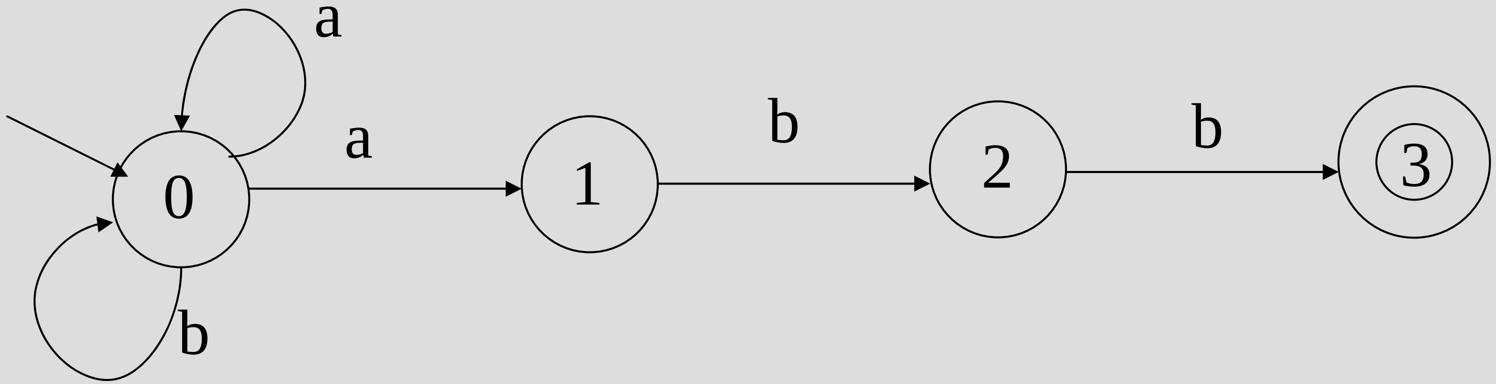
Représentation par une table de transitions:

État	a	b
0	0,1	0
1	-	2
2	-	3
3	-	-

Exemple

- $\Sigma = \{a,b\}$, $E= \{0,1,2,3\}$, $e_0 = 0$, $T= \{3\}$
- $\Delta(0 , a) = \{0,1\}$, $\Delta(0 , b) = \{0\}$, $\Delta(1 , b) = \{2\}$, $\Delta(2 , b) = \{3\}$; $\Delta(e , l) = \varnothing$ sinon

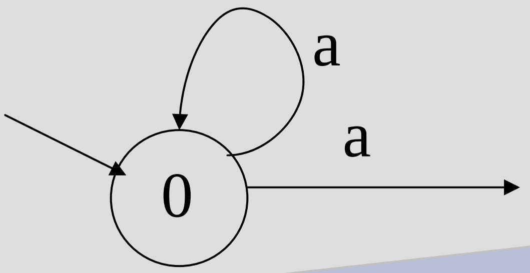
Expression Régulière correspondante à :



- Cet automate accepte l'expression régulière (langage régulier) (a|b)*abb

Définitions

- On dit qu' un A.E.F est un A.F.D. (Automate Fini Déterministe) lorsqu'il n'y a pas à choisir entre 2 transitions
- Il est non déterministe (A.F.N.) dans le cas contraire
- Un automate peut être facilement simulé par un algorithme, encore plus facilement un A.F.D.
- L'automate de l'exemple précédent est non déterministe à cause de



Expressions Régulières : (Rappel)

Les expressions régulières (E.R.) sur un alphabet A et les langages qu'elles décrivent sont définis récursivement de la manière suivante :

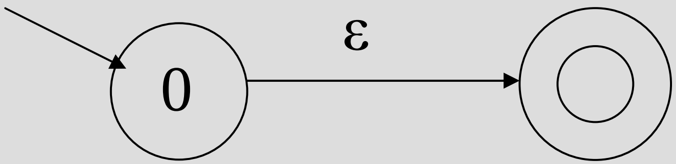
- ε est une E.R. qui décrit le langage {ε}
- Si $a \in A$, alors a est une E.R. qui décrit {a}
- Si r est une E.R. qui décrit le langage R, alors $(r)^*$ et $(r)^+$ sont des E.R. décrivant respectivement R^* et R^+
- Si r et s sont des E.R. qui décrivent respectivement les langages R et S, alors (r)|(s) et (r)(s) sont des E.R. décrivant respectivement $R \cup S$ et RS
- Il n'y a pas d'autres E.R.

Construction d'un A.F.N. à partir d'une E.R.

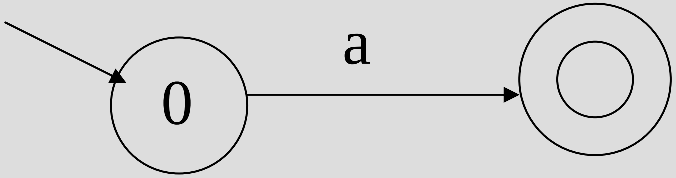
Construction de Thomson basée sur la récursivité des E.R.

Automate acceptant :

- La chaîne vide : (ε est une E.R. qui décrit le langage {ε})



- La lettre a (Si a∈A, alors a est une E.R. qui décrit {a})

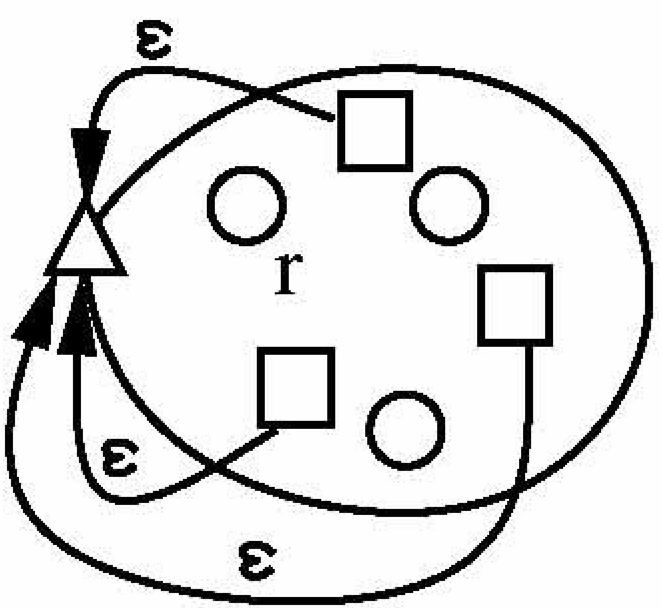


Construction d'un A.F.N. à partir d'une E.R.

Automate acceptant :

- r⁺ (Si r est une E.R. qui décrit le langage R, alors (r)* et (r)⁺ sont des E.R. décrivant respectivement R* et R⁺) (avec r+ = r r*)

L'état initial est figuré ci-contre par un triangle △
et les états finaux par des carrés □



Mettre des ε-transitions de chaque état terminal de A(r) vers son état initial. Pour l'*, rajouter une ε-transition de l'état initial vers un état de sortie

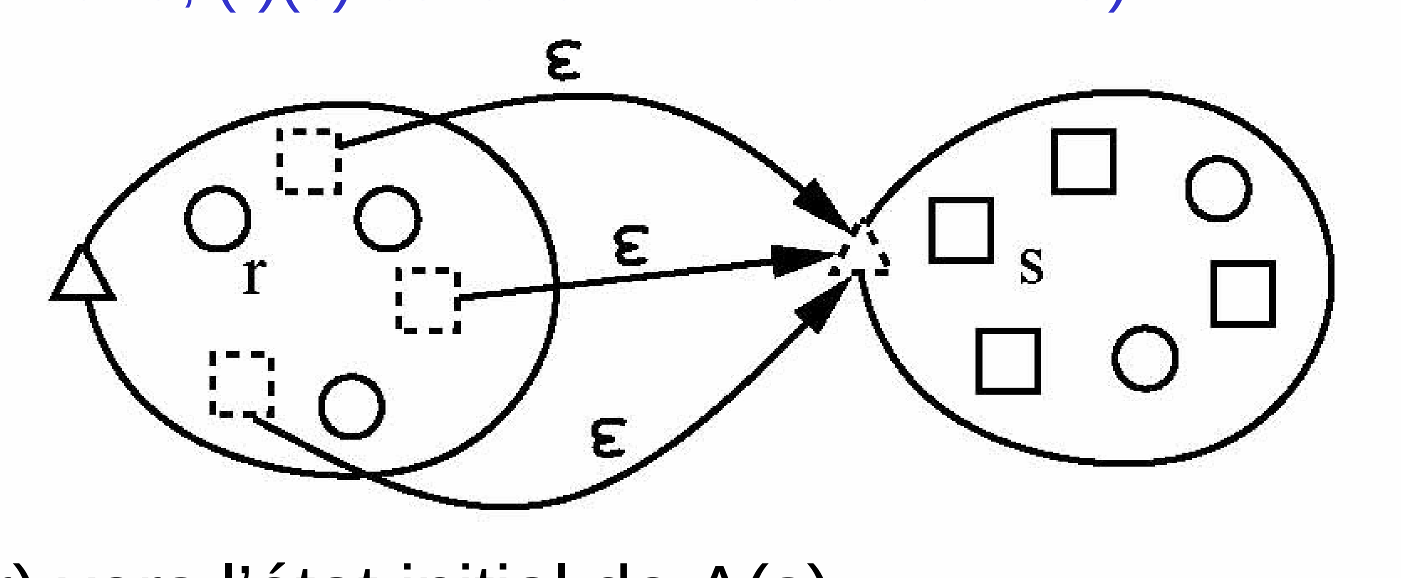
Construction d’un A.F.N. à partir d’une E.R.

Automate acceptant :

- (r)(s) (Si r et s sont des E.R. qui décrivent respectivement les langages R et S, (r)(s) est une E.R. décrivant RS)

L’état initial △

états finaux □



- Mettre une ε-transition de chaque état terminal de A(r) vers l’état initial de A(s)

- Les états terminaux de A(r) ne le sont plus, ceux de A(s) le restent

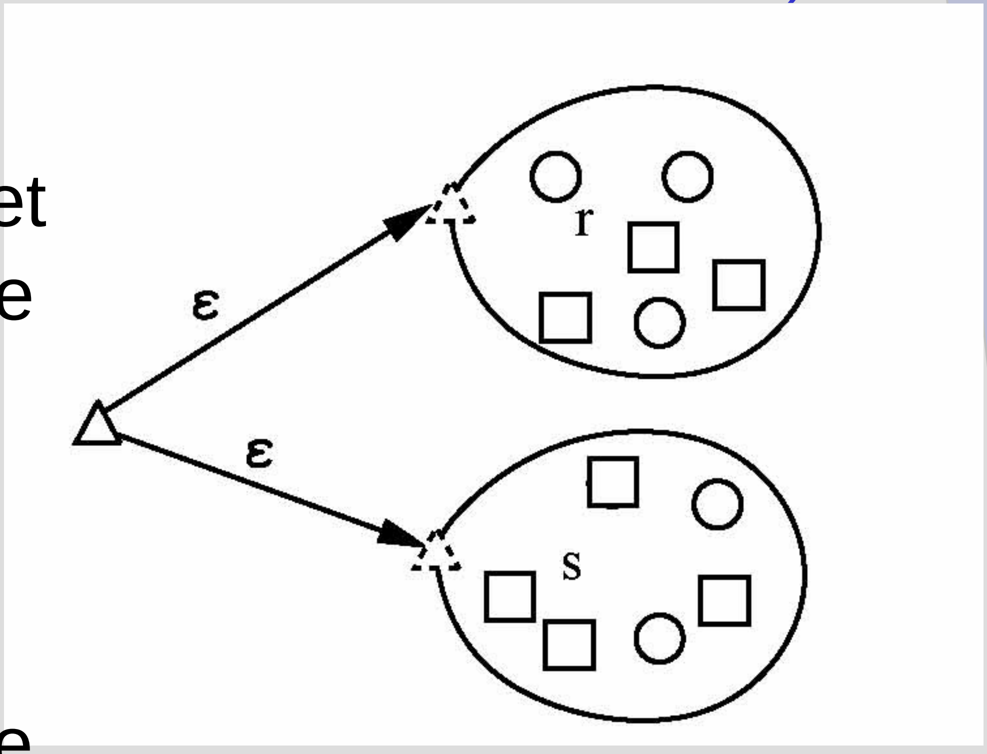
- L’état initial de s ne l’est plus, celui de r le reste

Construction d'un A.F.N. à partir d'une E.R.

Automate acceptant :

- r|s (Si r et s sont des E.R. qui décrivent respectivement les langages R et S, alors (r)(s) est une E.R. décrivant R ∪ S)

- Créer un nouvel état initial
- Mettre une ε-transition de cet état initial vers l'état initial de A(r) et A(s)
- Les états terminaux de A(r) et A(s) le restent
- Les états initiaux de r et s ne le sont plus



Exemple

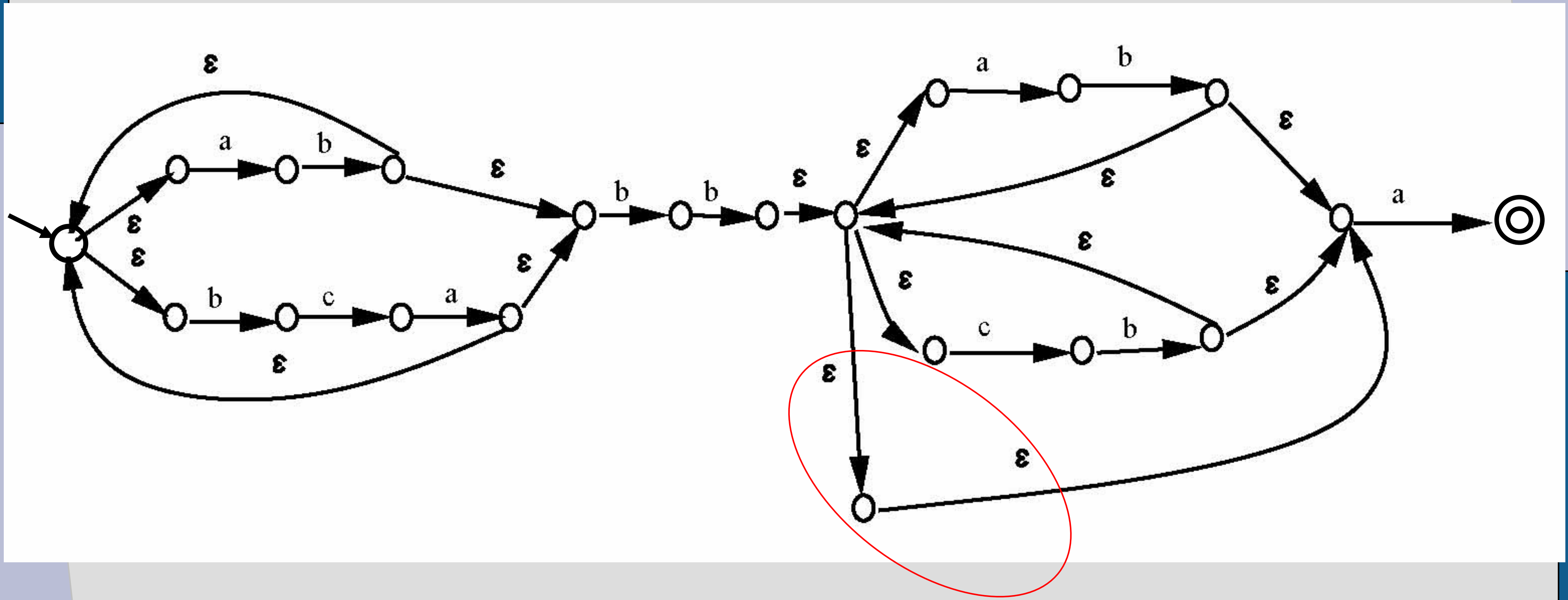
Automate reconnaissant :

$(ab|bca)^{+}\ bb\ (ab|cb)^{*}\ a$

Exemple

Automate reconnaissant :

(ab|bca)$^+$ bb (ab|cb)* a



AFD Automate Fini Déterministe

Un automate est dit déterministe lorsqu'à partir d'un état, il n'y a pas de choix possible pour la transition :

- Il ne doit pas posséder d'ε-transition
- Pour chaque état *e* et pour chaque transition *a*, il y a au plus un arc étiqueté par *a* qui quitte *e*.

Les programmes simulant les AFD sont facile à écrire.

Il existe des Algorithmes transformant des AFN en AFD