Université Mohammed Khider-BISKRA

Faculté des Sciences Exactes et la nature et de la vie

Département d'informatique

Cours du Module

Théorie des langages

Pr. CHERIF Foudil

Chapitre 4: Les langages réguliers

4.1 Définition d'une grammaire régulière

Les grammaires de type 3 sont appelées aussi les grammaires régulières à droite (respectivement à gauche)

Les règles de la grammaire sont de la forme:

$A \rightarrow a\,B$ (respectivement $A \rightarrow B\,a$)

ou $A \rightarrow a$

Avec $A, B \in V_n$ et $a \in V_t$

Autrement dit, le membre de gauche de chaque règle est constitué d'un seul symbole non terminal, et le membre de droite est constitué d'un symbole terminal éventuellement suivi (respectivement précédé) d'un seul symbole non terminal.

Chapitre 4: Les langages réguliers

Les langages réguliers sont les langages engendrés par des grammaires régulières.

Les grammaires régulières sont utilisées beaucoup dans la phase d'analyse lexicale.

Exemple: G=({a,b}, {S,A}, S, R)

R=($S \rightarrow a S$

$S \rightarrow bA$

$A \rightarrow a$)

Chapitre 4: Les langages réguliers

4.2 Définition d'un automate

Les automates sont des machines fictives(programmes), qui prennent en entrée une chaîne de symboles et qui effectuent un algorithme de reconnaissance de la chaîne. Si l'algorithme se termine dans certaines conditions, l'automate accepte cette chaîne.

Le langage reconnu par un automate, est l'ensemble des chaînes qu'il accepte.

Nous avons deux types de systèmes:

- Les systèmes de génération qui sont les grammaires
- Les systèmes de reconnaissance qui sont les automates

4.3 Automate d'états finis

4.3.1 Définition

Un AEF est composé d'un ensemble fini d'états (représentés graphiquement par des cercles), d'une fonction de transition décrivant l'action qui permet de passer d'un état à l'autre (ce sont les flèches), d'un état initial (l'état désigné par carré) et d'un ou plusieurs états finaux (désignés par des triangles).

Un AEF est donc un graphe orienté et étiqueté (valué)où les nœuds correspondent aux états et les valeur des arcs aux symboles terminaux

Les AEF n'utilisent aucune mémoire pour reconnaitre une chaine.

4.3.2 Définition formelle

Un automate d'états finis est un quintuplet:

$A = (V_t, Q, q_0, f, F)$

- V_t : est le vocabulaire terminal, ensemble fini non vide de symboles
- Q : est l'ensemble d'états de l'automate, ensemble fini non vide
- q_0 : L'ensemble Q contient un état particulier q_0 appelé état initial. $q_0 \in Q$
- F : L'ensemble Q contient un sous ensemble d'états particuliers F appelés états finaux. $F \subset Q$
- f : est une application de $Q \times V_t \cup \{ \epsilon \} \rightarrow Q$

L'automate s'arrête sur un état final ou la lecture complète de la chaîne d'entrée.

Représentations d'un AEF

Il existe trois principales représentations pour les AEF:

-la représentation matricielle,

-sous forme de graphe orienté étiqueté

-ou sous forme de fonctions (relations)

a) Fonction de transition

f est la fonction de transition de A $f(q, a) = q'$

Indique que si l'automate est dans l'état q et qu'il rencontre le symbole a, il passe à l'état q'. De plus pour tout q de Q $f(q, \epsilon) = q$

Exemple:

Soit A l'AEF défini par le quintuplé (Vt, Q, $q0$, f, F)) tel que :

Vt =$\{a, b\}$,

Q= $\{q_0, q_1\}$,

q_0 : état initial

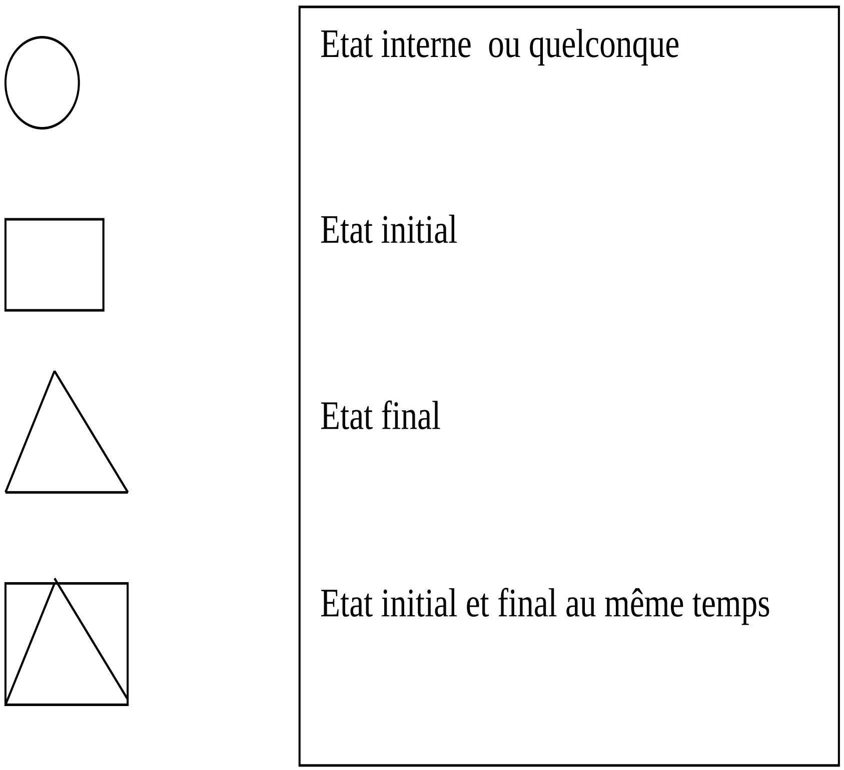
F= $\{q_1\}$

et on définit les transitions suivantes : $f(q_0, a) = q_0$, $f(q_0, b) = q_1$, $f(q_1, b) = q_1$

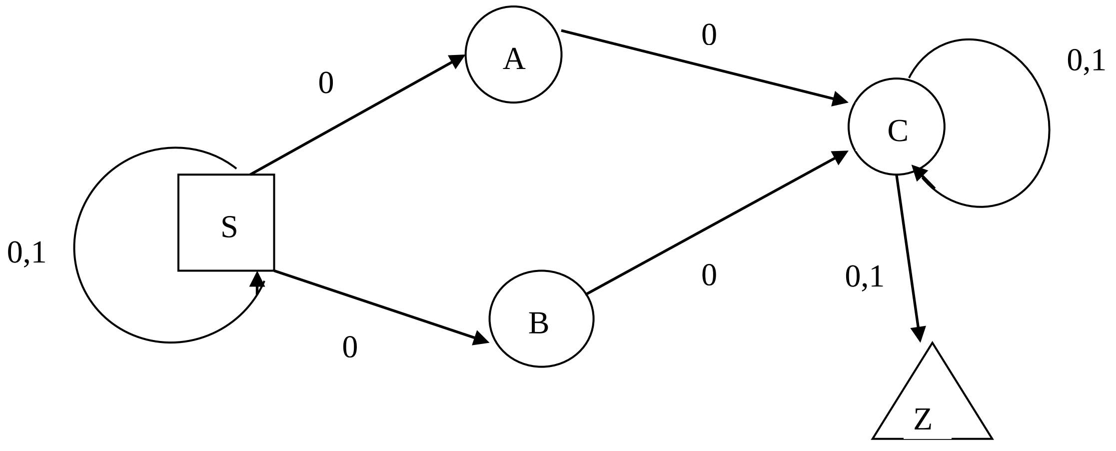
Chapitre 4: Les langages réguliers

b) Graphe orienté:

On représente un automate d'états finis par un graphe orienté et valué, dont les arcs portent des symboles de V_t et dont les nœuds portent les états.



Exemple:



c) Tableau de transitions (matricielle)

Soit A l'AEF défini par le quintuplé (Vt, *Q, q0, f, F)*) tel que :

Vt =$\{a, b\}$, Q= $\{q_0, q_1\}$, q_0 : état initial F= $\{q_1\}$

et on définit les transitions suivantes : f(q_0, a) = q_0, f(q_0, b) = q_1, f(q_1, b) = q_1

La fonction de transition peut être représentée par cette matrice:

Q \ V_t	a	b
q_0	q_0	q_1
q_1		q_1

Langage reconnu par un automate d'états finis:

Le langage reconnu par un automate d'états finis est l'ensemble des chaînes dont les symboles font passer de l'état initial jusqu'à un de ses états finaux par une succession de transitions utilisant tous ses symboles dans l'ordre.

Définition d'une configuration

La configuration de l'AEF A, à un certain instant, est donnée par l'état courant de l'AEF et du mot qui reste à lire (état courant, mot qui reste à lire).

La configuration initiale est (q_0, ω) où q_0 est l'état initial de l'AEF et ω le mot soumis à A (à reconnaître).

La configuration finale est donnée par (q_f, ε) où q_f est un état final et le mot vide indique qu'il ne reste rien à lire (le mot appartient au langage).

Chapitre 4: Les langages réguliers

Définition d'une dérivation directe

On dit qu'une configuration (q_1, $a\omega$) dérive directement la configuration (q_2, ω) si et seulement si f(q_1, a) = q_2 où f est la fonction de transition, a$\in Vt$ et $\omega \in Vt^*$.

On note (q_1, $a\omega$) |= (q_2, ω).

Définition d'une dérivation indirecte

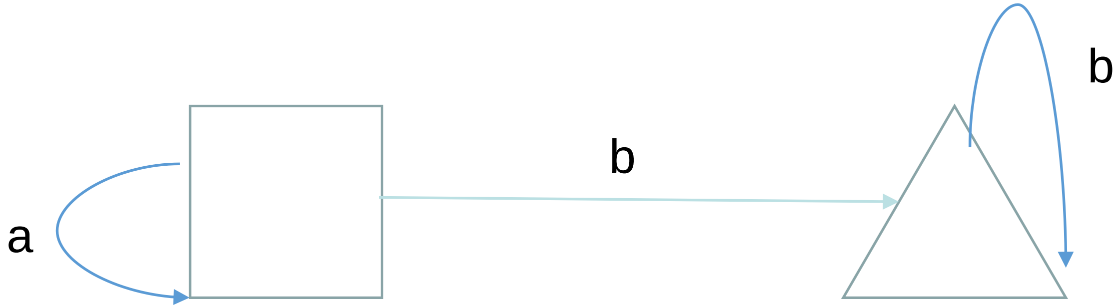
On dit qu'une configuration (q, ω_1) dérive indirectement une autre configuration (p,ω_2), si et seulement s'il existe 0, 1 ou plusieurs dérivations directes qui, à partir de (q,ω_1), mènent à la configuration (p, ω_2). On note (p, ω_1) $|=_*$(q, ω_2).

Définition du langage reconnu par un AEF

Le langage reconnu par l'AEF A noté $L(A)$ est défini par :

$L(A) = \{\omega \in Vt^* /(q_0, \omega) \mid= (q_f, \varepsilon)\}$ avec $q_f \in F$.

Chapitre 4: Les langages réguliers

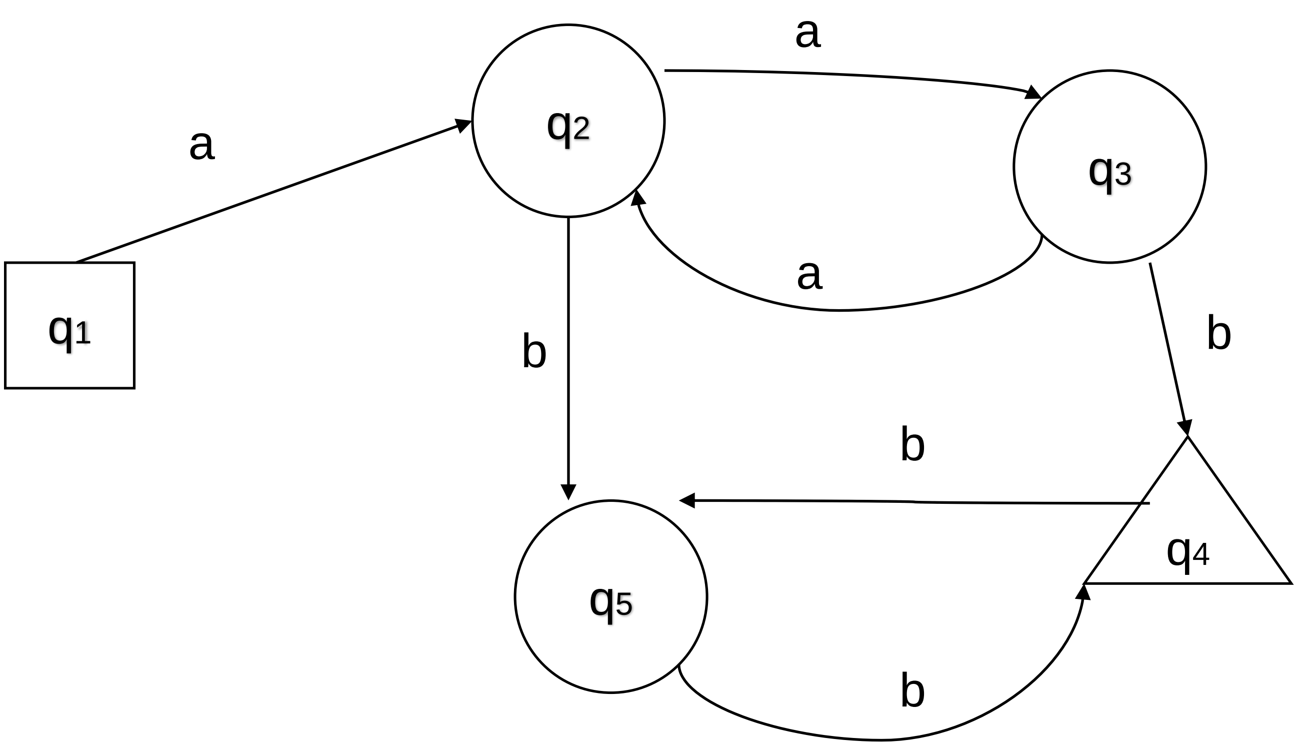


Exemple 1:

La chaine minimale est b
Le langage reconnu par cet automate est:

$L(A)= \{ a^n \ b \ b^m \ / \ \ n \geq 0 \ ; \ \ m \geq 0 \ \}$

Exemple 2: Trouver le langage reconnu par cet AEF



Remarque: l'état 4 est un état final et au même temps un état interne

Les chemins possibles:

- $q_1\ q_2\ q_5\ q_4$
- $q_1\ q_2\ q_3\ q_4$
- $q_1\ q_2\ q_3\ q_2\ q_5\ q_4$
- $q_1\ q_2\ q_3\ q_4\ q_5\ q_4$

Les chaines minimales:

- aab et abb

Les chaînes reconnues:

- $a\ (aa)^*\ b\ \ b\ (bb)^*$
- $a\ a\ (aa)^*\ b\ (bb)^*$

$$L(A) = \{ a\ (aa)^*\ b\ \ b\ (bb)^* \} \cup \{ a\ a\ (aa)^*\ b\ (bb)^* \}$$

Chapitre 4: Les langages réguliers

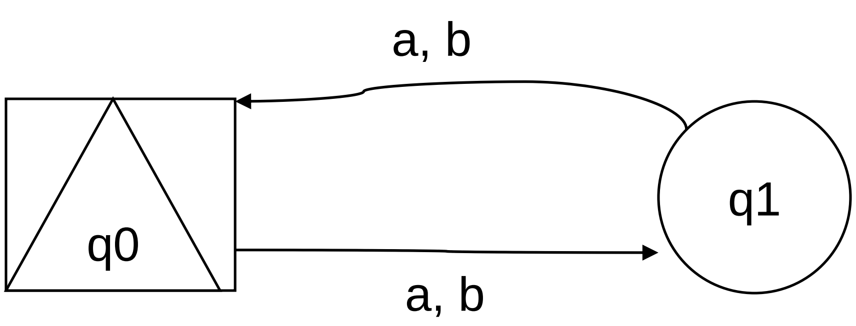
Remarques:

1- Un automate d'états finis reconnait un seul langage, mais le même langage peut être reconnu par plusieurs automates,

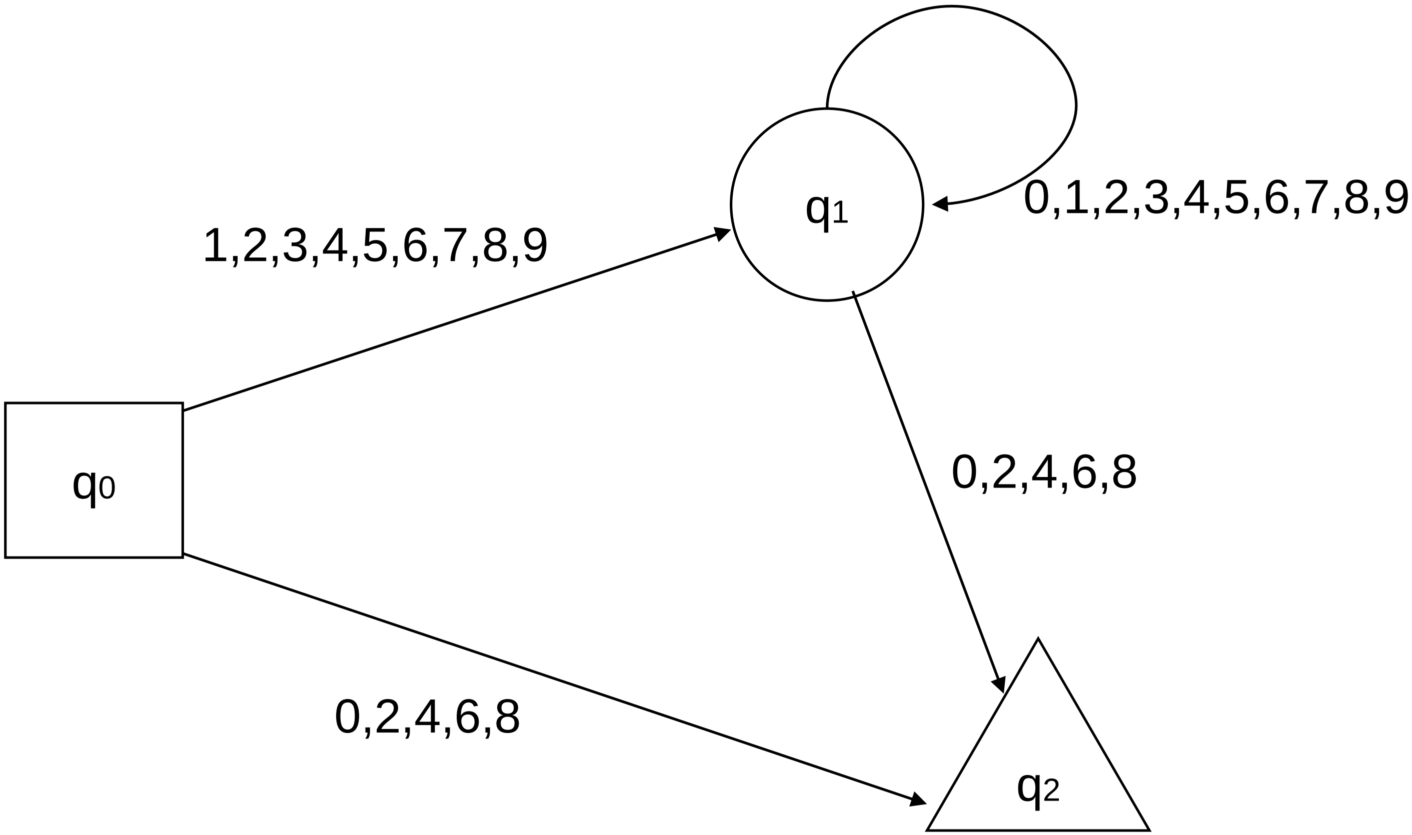
2- On dit que deux automates d'états finis A1 et A2 sont équivalents si et seulement si ces deux automates reconnaissent le même langages. L(A1)= L(A2)

Exemple de construction d'AEF:

L1= { w/ w ∈ {a,b}* et |w| ≡ 0 [2] }

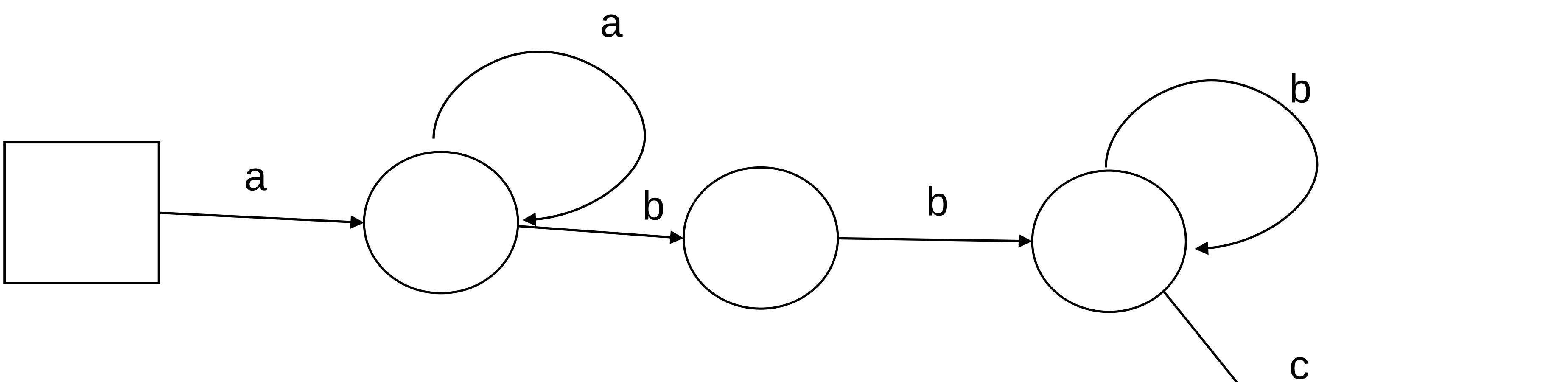


L2= { w/ w $\in$ *N* *et* $w \equiv 0 \; [2]$ }

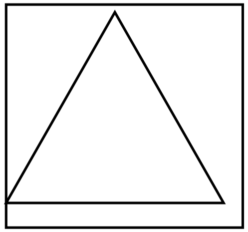


Chapitre 4: Les langages réguliers

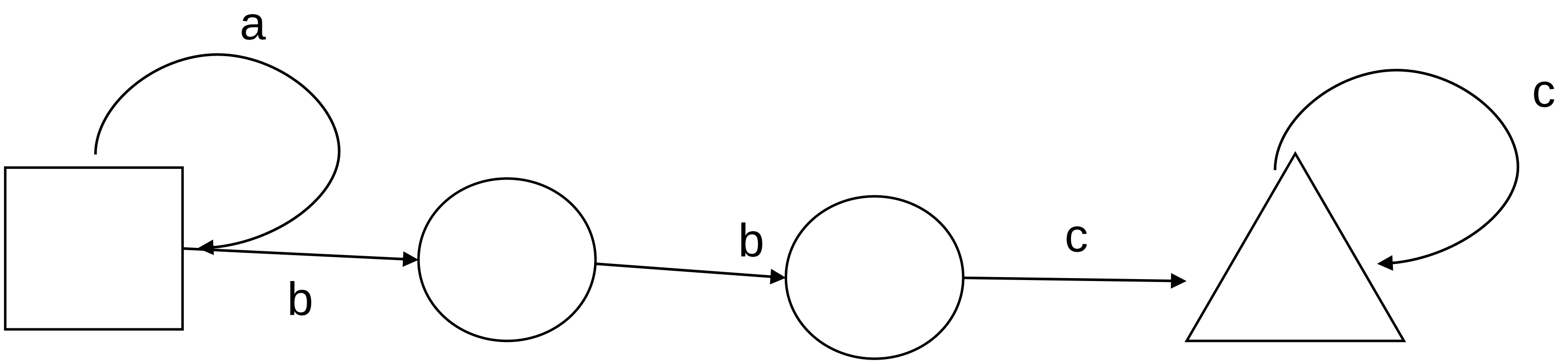
L3= { $a^i b^j c^k$ / $i \geq 1$, $k \geq 1$ et $j > 1$ }



L4= { ε }



$L5 = \{ a^i \, b^2 \, c^k \;/\; i \geq 0 \text{ et } k > 0 \}$



Chapitre 4: Les langages réguliers

4.4 Variantes d'automates d'états finis

a) Automates d'états finis déterministes (AEFD)

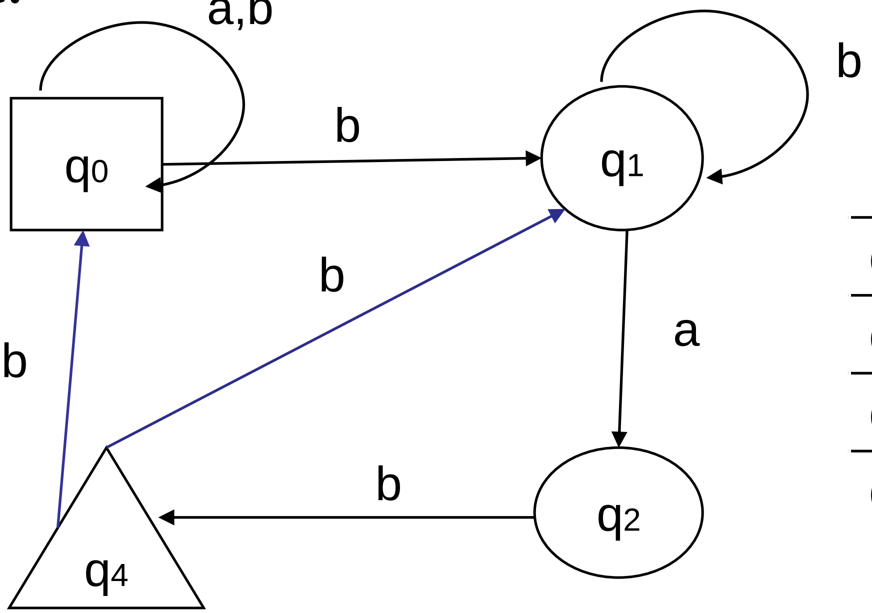
Un automate d'états fini **déterministe** est un automate tel que sa fonction de transition est une fonction. C'est-à-dire tel que, pour tout état et pour tout symbole, il existe **au plus** un état dans lequel l'automate peut passer.

b) Automates d'états finis non déterministes (AEFND)

Un automate d'états fini **non déterministe** est un automate tel qu'il existe au moins un couple formé d'un état et d'un symbole, qui admet plus d'une image par la fonction de transition. L'automate doit faire des choix pour progresser.

Chapitre 4: Les langages réguliers

Exemple:



	a	**b**
q_0	q_0	q_0, q_1
q_1	q_2	q_1
q_2		q_4
q_4	q_0, q_1	

Cet automate est non déterministe : pour le même état et le même symbole nous avons deux états, où on trouve plus d'une valeur dans une case du tableau de transitions

$f(q_0,b) = q0$ et $f(q_0,b) = q_1$

$f(q_4,b) = q_0$ et $f(q_4,b) = q_1$

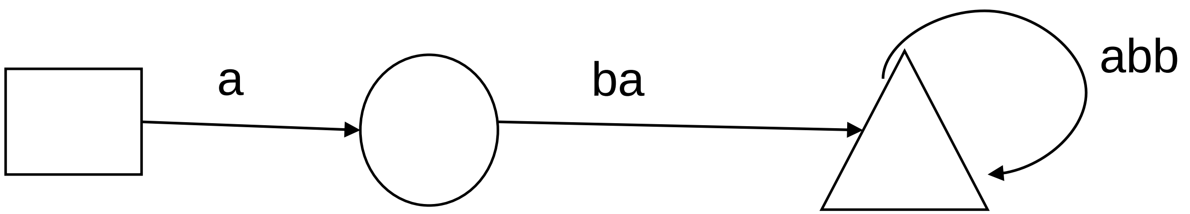
Le problème des AEFND:

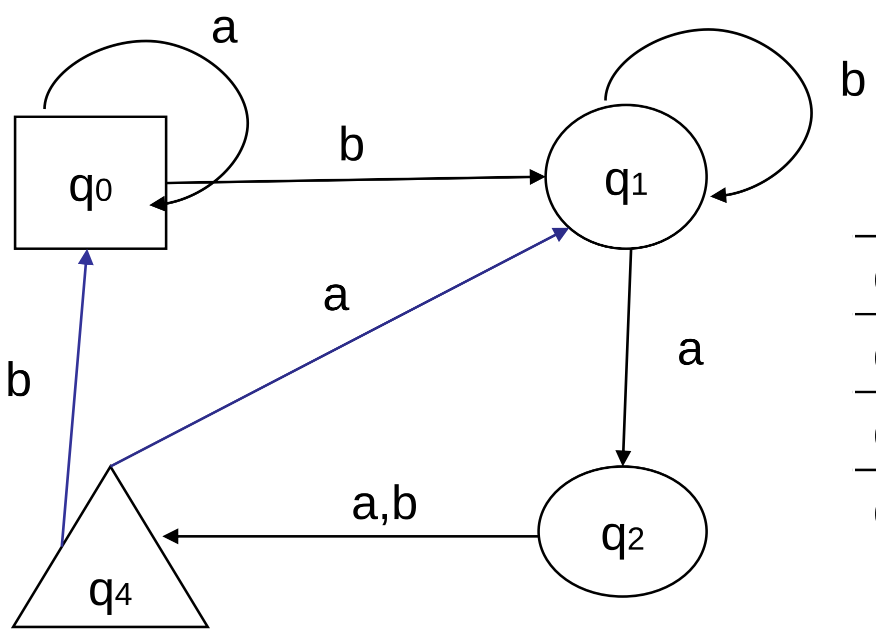
Ces automates analyse plus lentement les chaines que les automates déterministes, on doit faire des choix pour progresser et faire des retours par la suite en cas d'échec,

c) Automate d'états finis généralisé (AEFG)

- Les transitions peuvent engendrées par des mots au lieu de symboles.

- Les transitions causées par le mot vide (ε) sont appelées transitions spontanées ou vide (ε-transition). C'est un changement d'état sans lecture .





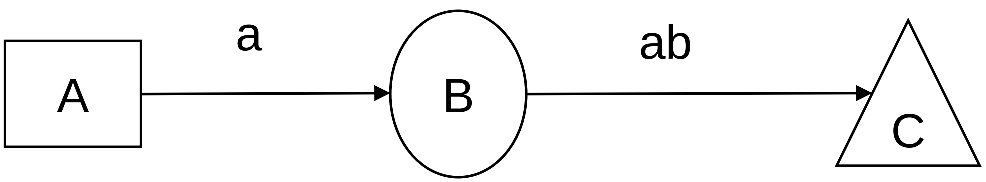
	a	**b**
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_4	q_4
q_4	q_1	q_0

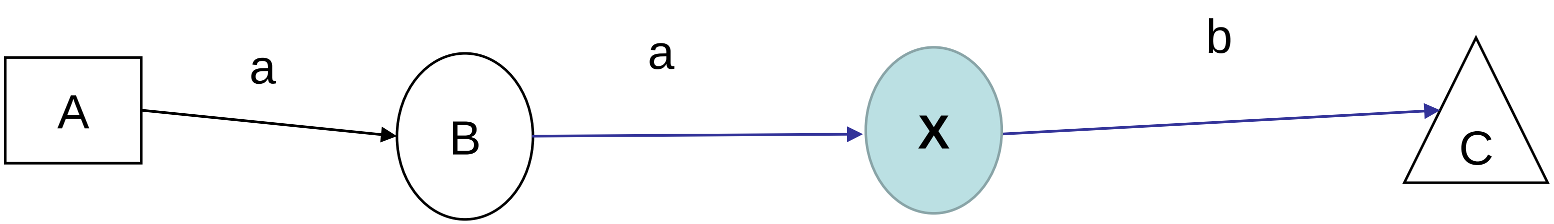
d) Automate d'états finis complet (AEFC)

Un automate est dit complet s'il est déterministe et que pour chaque état et pour chaque symbole, il existe exactement une transition.

4.5 Transformation d’un AEF généralisé en un automate simple

Tout automate d’états finis généralisés admet un automate simple équivalent en ajoutant des transitions supplémentaires.





4.6 Transformation d'un AEF non déterministe à un automate déterministe

A tout automate d'états fini non déterministe correspond un automate d'états fini déterministe équivalent et réciproquement.

La Passage d'un automate non déterministe à un automate déterministe se fait suivant l'algorithme suivant :

Le but est de trouver les éléments de l'automate déterministe.

$A=(V_t, Q, q_0, f, F)$ automate non déterministe donné

$A'=(V'_t, Q', q'_0, f', F')$ automate déterministe reconnaissant le même langage

1)$V'_t = V_t$

2)$q'_0 = \{q_0\}$

3)Q' est inclus dans l'ensemble des combinaisons de Q, $P(Q)$

4)$F' = B \in Q'$ si $B \cap F \neq \theta$ ==> $B \in F'$

5) f' : f'(B, x) = M

soit $B = \{ q_1, q_2, \ldots, q_i, \ldots, q_k \}$

$f(q_1, x) = M_1$ ….. $f(q_i, x) = M_i$, …., $f(q_k, x) = M_k$

$M_i \in Q'$ M_i inclus dans $P(Q)$

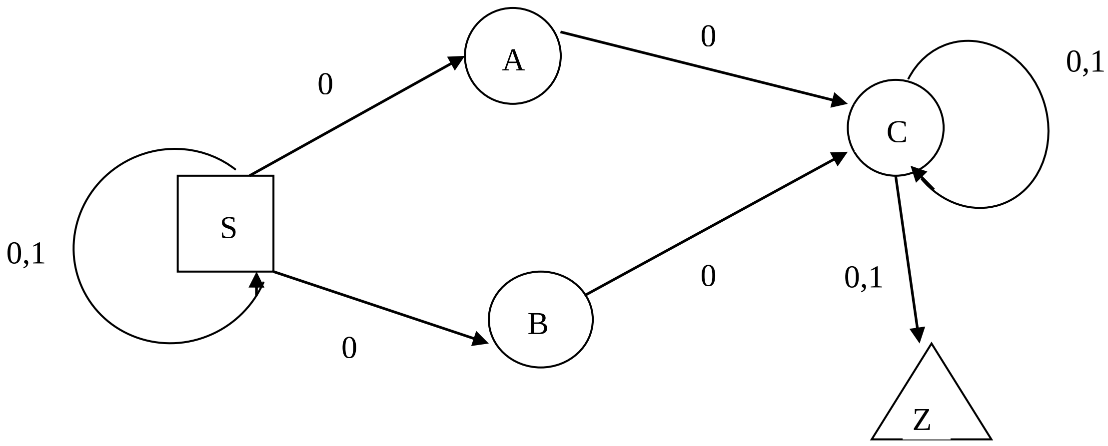
$M = \cup M_i \quad 1 \leq i \leq k$

6) Suppression des transitions vides.

4.6 Transformation d'un AEFND à un AEFD (autre algorithme)

- Partir de l'état initial $E(0) = \{q0\}$ (c'est l'état initial du nouvel automate);
- Construire $E(1)$ l'ensemble des états obtenus à partir de $E(0)$ par toutes les transitions avec les lettres de Vt . $E(1) = E(0) \cup \{qi\} / \delta(q0, x) = qi, x \in Vt$ et $q0 \in E(0)$;
- Recommencer l'étape précédente pour toutes les transitions possibles et pour chaque nouvel ensemble $E(i)$. $E(j) = E(j\text{-}1) \cup \{qi\} / \delta(qi\text{-}1, x) = qi, x \in Vt$ et $qi\text{-}1 \in E(j\text{-}1)$;
- Tous les ensembles contenant au moins un état final du premier automate deviennent des états finaux.

Exemple d'application: rendre l'automate déterministe



La construction du nouveau tableau de transitions se fait de la manière suivante :

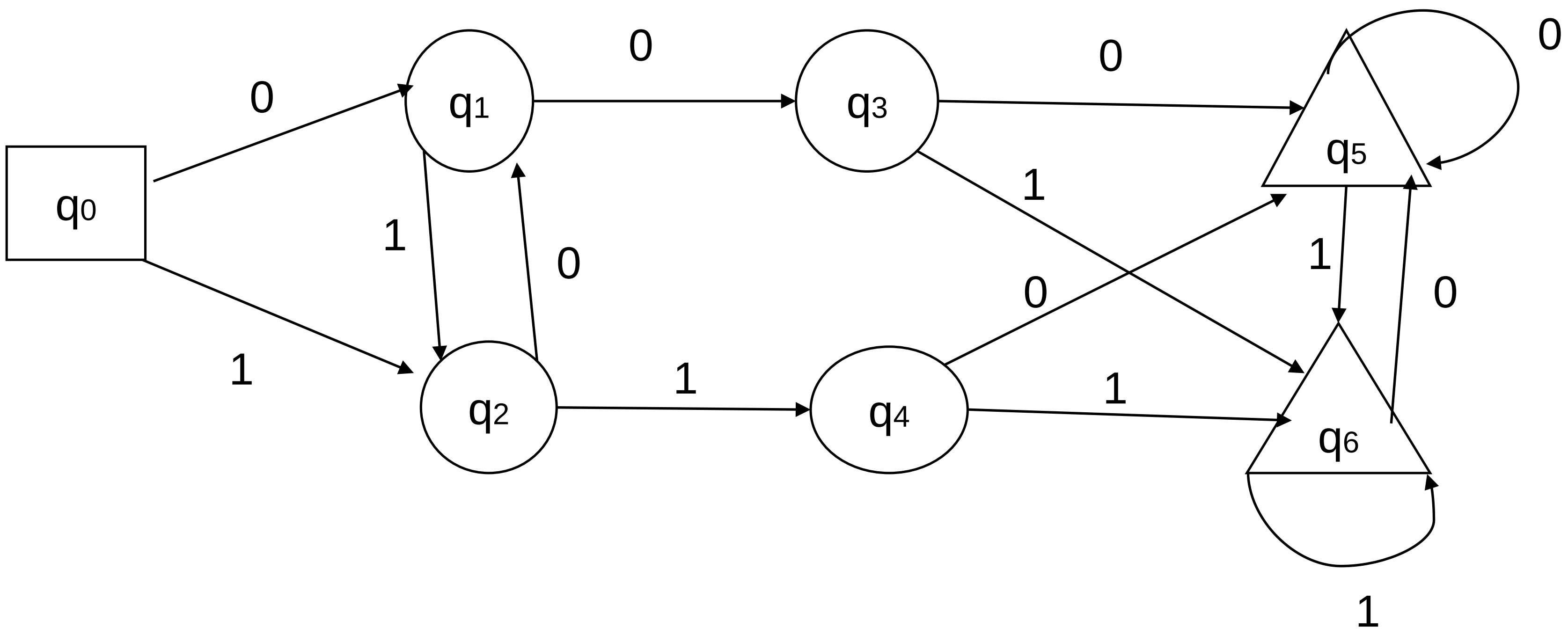
	0	1
{S } = q0	{S,B} q1	{S,A} q2
{S,B} = q1	{S,B,C} q3	{S,A} q2
{S,A} = q2	{S,B} q1	{S,A,C} q4
{S,B,C} = q3	{S,B,C,Z} q5	{S,A,C,Z} q6
{S,A,C}= q4	{S,B,C,Z} q5	{S,A,C,Z} q6
{S,B,C,Z} = q5	{S,B,C,Z} q5	{S,A,C,Z} q6
{S,A,C,Z} = q6	{S,B,C,Z} q5	{S,A,C,Z} q6

Nouveau tableau de transition de l'automate déterministe

L'état initial : q_0

F= { q_5, q_6 }

L'automate d'états finis déterministe équivalent



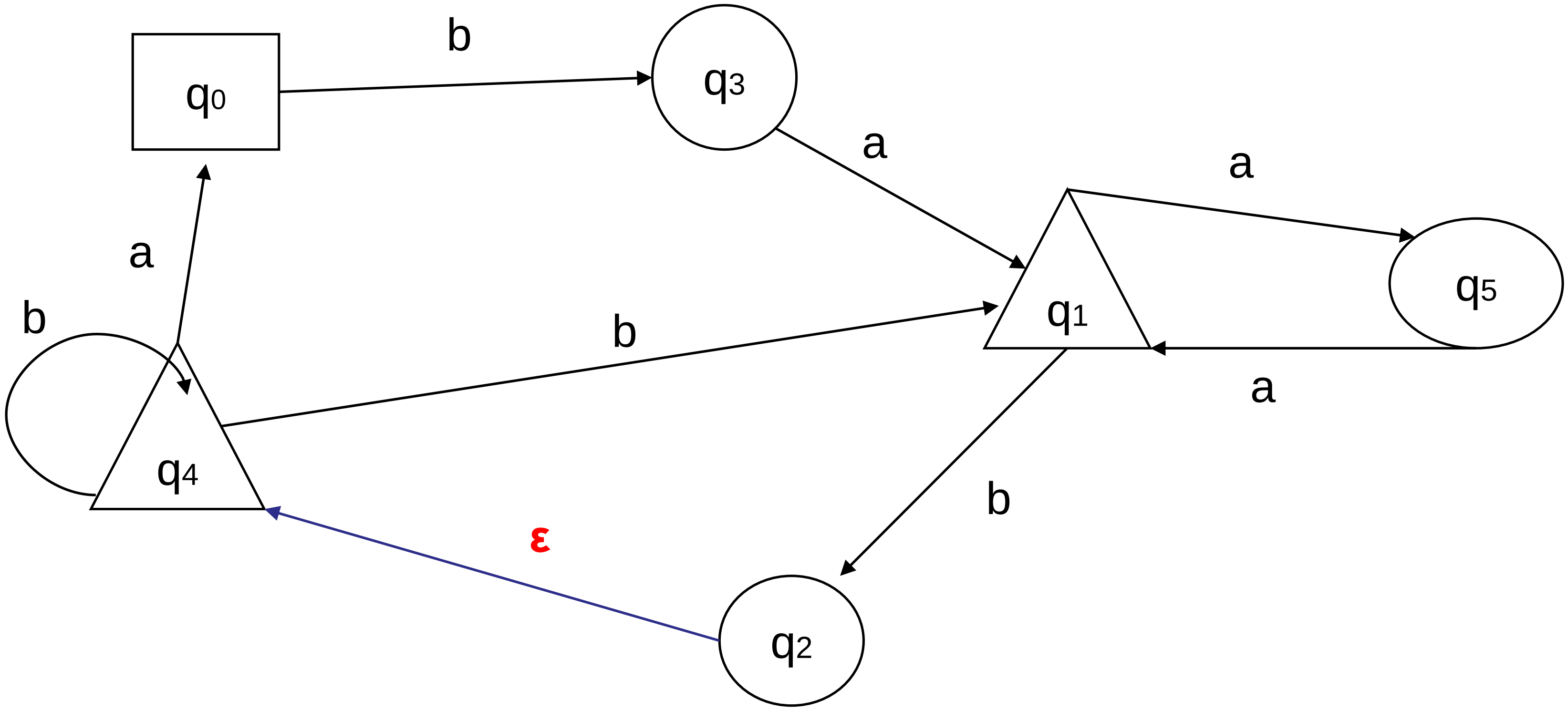
4.6 Elimination des transitions vides

La suppression des ces transitions nous donne un AEF simple, pour ce faire il faut d'abord enlever les transitions par ε:

$$\delta(q_i, \varepsilon) = q_j \quad \begin{cases} Si\ q_j \in F \quad alors \quad q_i \in F \\ et \\ \forall a \in Vt\ :\ si\ \delta(q_j, a) = q_k \quad alors \quad \delta(q_i, a) = q_k \end{cases}$$

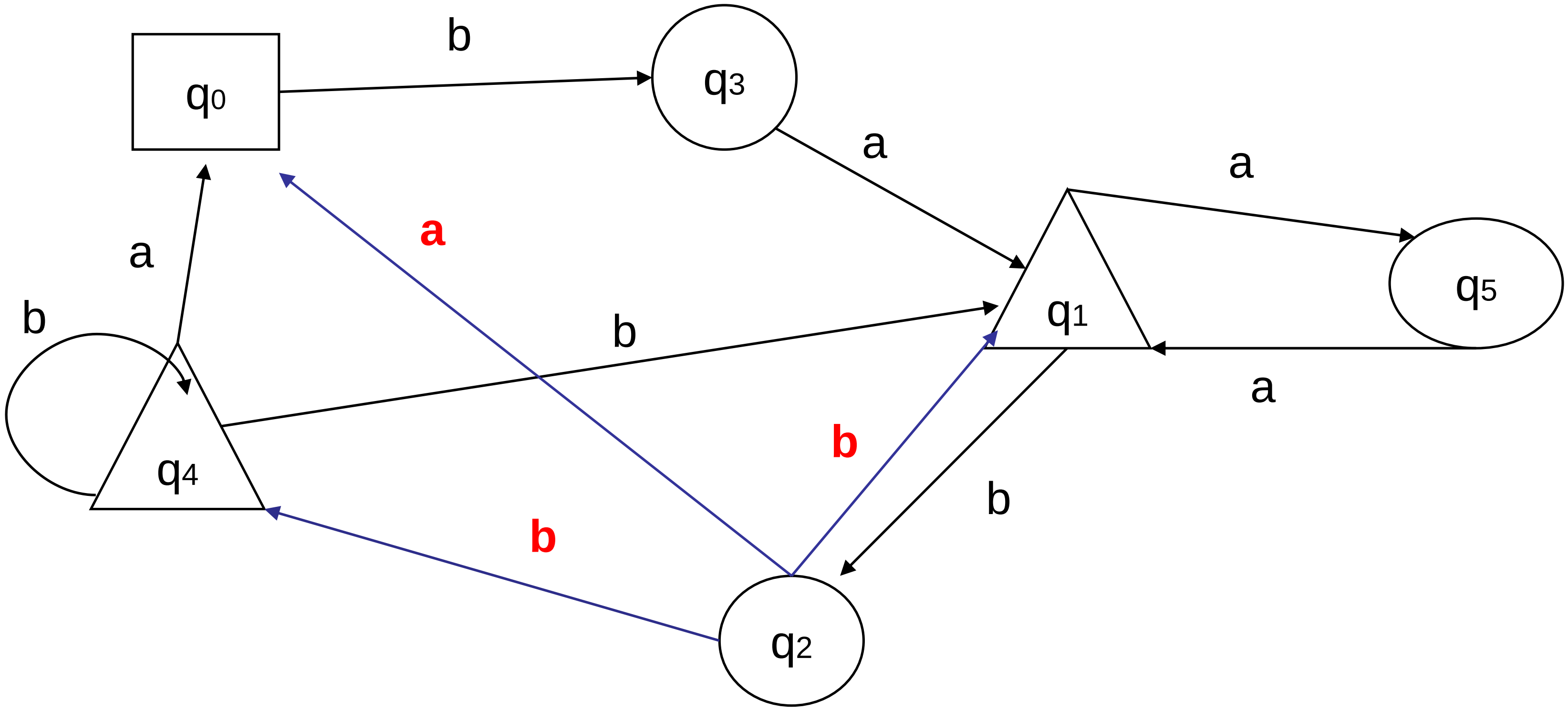
Un exemple de cette transformation est illustré sur AEF suivant:

Nous avons une seule **ε** transition f(q2, **ε**) = q4 , q4 état final donc q2 devient état final
Nous avons 3 transitions f(q4,a) = q0 , f(q4,b)= q1 et f(q4,b) = q4, donc on ajoute 3 transitions en remplaçant q4 par q2



Chapitre 4: Les langages réguliers

On supprime la transition vide et on la remplace par ces transitions: f(q2,a) = q0 , f(q2,b)= q1 et f(q2,b) = q4,



4.7 Passage d'une grammaire régulière à un AEF

Pour toute grammaire régulière $G = (Vt, Vn, S, R)$, il existe un AEF $A = (Vt, Q, q_0, f, F)$ tel que L(G)=L(A). Le passage se fait en associant une transition à chaque règle de la grammaire. L'automate construit n'st pas automatiquement déterministe.

Soit $G = (Vt, Vn, S, R)$, une grammaire régulière , il s'agit de trouver un AEF

$A = (Vt', Q, q_0, f, F)$ tel que L(G)=L(A)

1) $Vt' = Vt$ 2) $Q = Vn \cup q_f$ tel que $q_f \in F$ 3) $q_0 = S$ 4) $F = \{q_f\}$.

5) Pour chaque règle : $A \rightarrow a\ B$, on associe la transition $f(A, a) = B$.

Pour chaque règle de la forme $A \rightarrow a$, on associe la transition $f(A, a) = q_f$.

6) Si la grammaire possède la règle $S \rightarrow \varepsilon$ alors S devient un état final $F = \{ q_f, S\}$

4.7 Exemple de passage

Trouvons l'AEF équivalent à la grammaire suivante :

G=({a,b,c},{S,A,B}, S, R)

R = ($S \rightarrow aS \mid bA$

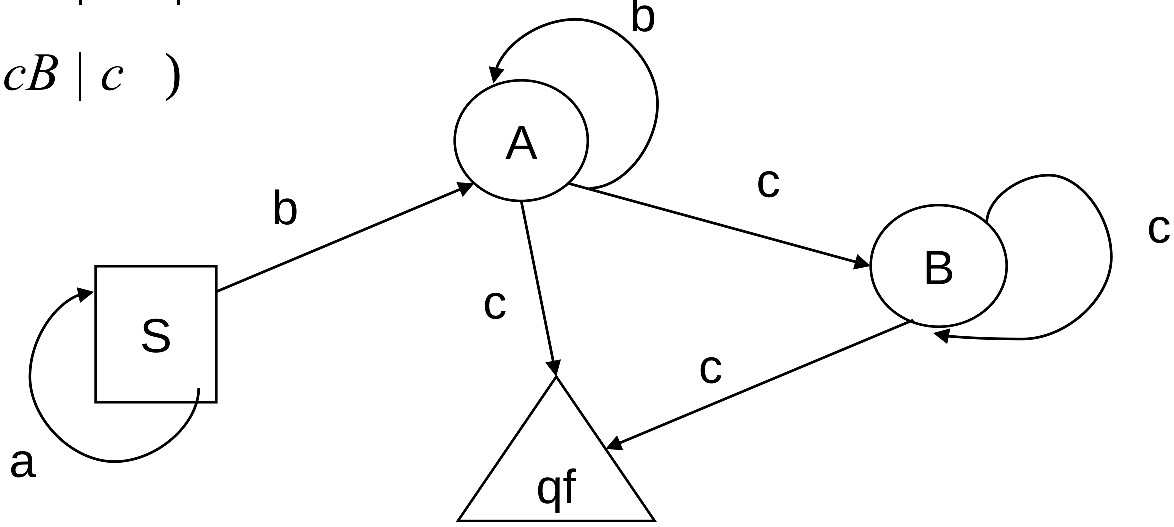
$A \rightarrow bA \mid cB \mid c$

$B \rightarrow cB \mid c$)

Q= { S,A,B, qf}

F={qf}

État initial : S



4.8 Passage AEF à une grammaire régulière

Pour tout AEF A = (Vt, Q, q0,f, F) il existe une grammaire régulière G = (Vt', Vn, S, R) équivalente tel que L(G)=L(A).

Le passage se fait comme suit:

1) Vt' = Vt 2) Vn = Q 3) $q_0 = S$

4)
- Si $f(q, a) = p \in A$ alors $q \rightarrow a\,p \in R$
- Si $f(q, a) = q_f \in A$ alors $q \rightarrow a \in R$
- Si $q_0 \in F$ alors $q_0 \rightarrow \varepsilon \in R$

4.8 Exemple de passage : Trouver la grammaire équivalente à cet automate :

G=({a,b},{S,A,B, C,D}, S, R)

R = f(S,a) = A devient **S → a A**

f(A,a) = B devient **A → a B**

f(A,b) = C devient **A → b C**

f(S,a) = A devient **S → a A**

f(B,a) = A devient **B → a A**

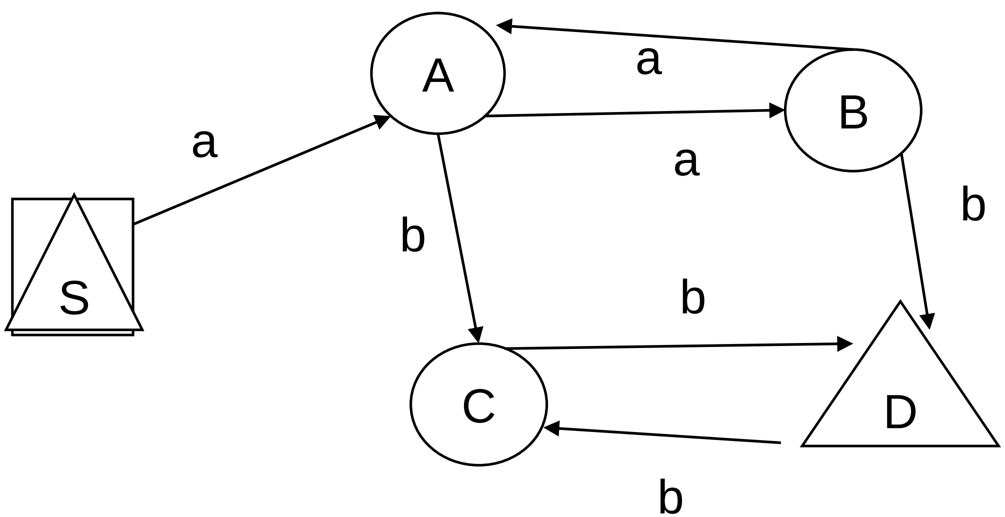
f(B,b) = D (D interne) devient **B → b D**

f(B,b) = D (D final) devient **B → b**

f(D,b) = C devient **D → b C**

f(C,b) = D (D interne) devient **C → b D**

f(C,b) = D (D final) devient **C → b**



L'état S est un état initial et final au même temps, donc on ajoute la règle :

S → ε

4.9 Les expressions régulières

a) Définition :

Les expressions régulières (ER) fournissent une autre méthode de définition des langages réguliers. Elles sont plus pratiques que les deux autres systèmes (grammaires régulières et automates).

Chaque expression régulière décrit un ensemble de chaines terminales. Le formalisme des expressions régulières utilise 03 opérations:

- La concaténatiion
- La fermeture notée * (puissance)
- L'alternative notée + ou / (choix entre deux expressions)

Chapitre 4: Les langages réguliers

4.9 Les expressions régulières

b) Définition formelle:

— ϕ est une expression régulière qui dénote (représente) le langage vide

— ε est une expression régulière qui dénote le langage $\{\varepsilon\}$

— a (où $a \in Vt$) est une expression régulière qui dénote le langage $\{a\}$.

Induction:

ε et a des expressions régulières;

Si e, e' sont des expressions régulières alors e+e' , e.e', e*, e+ sont des expressions régulières.

Remarques :

- L'exposant est plus prioritaire que la concaténation qui est prioritaire que la somme.

- Deux expressions régulières sont dites ε-équivalentes ssi elles dénotent le même langage.

4.9 Les expressions régulières

c) **Exemples de construction** des langages engendrés par des expressions régulières:

$(a + b)^* = \{ \varepsilon, a, b, aa, bb, ab, ba, aaa, bbb, aab, aba, \ldots\ldots\}$ langage infini

$a^*b + b^*a = \{ b, ab, aab, aaab, aa\ldots.ab, a, ba, bba, bbba, bb\ldots ba, \ldots. \}$

$ab^*(c+ a) = ab^*c + ab^*a = \{ ac, abc, abbc, abbbc, \ldots.., aa, aba, abba, abbba, \ldots.\}$

4.10 Passage expression régulière à une grammaire régulière

0 (0 + 1)* 0

G=({0 ,1},{S,A}, S, R)

R = ($S \rightarrow 0\,A$ $A \rightarrow 0\,A$ $A \rightarrow 1\,A$ $A \rightarrow 0$)

(0 + 1)* 0 (0 + 1) (0 + 1)

G=({a,b},{S,A, B }, S, R)

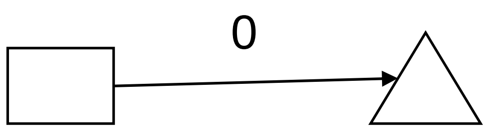
R = ($S \rightarrow 0S\ /\ 1S$ $S \rightarrow 0\,A$ $A \rightarrow 0\,B\ /\ 1B$ $B \rightarrow 0\ /\ 1$)

4.11 Passage expression régulière à un automate d'états finis

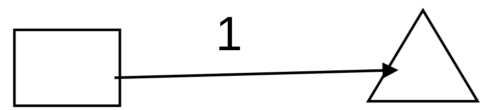
Il existe des automates pour chaque expression régulière

Exemple de construction: **(0 + 1)* 0 1 0**

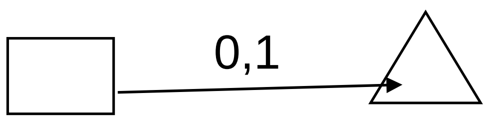
0 :



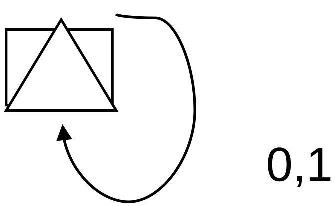
1 :



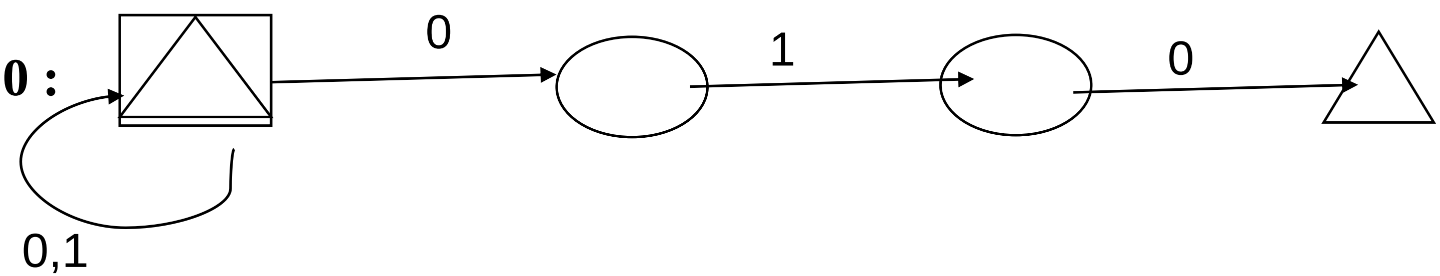
(0 + 1) :



(0 + 1)* :



(0 + 1)* 0 1 0 :



4.11 Méthodes pour montrer qu'un langage est régulier

On peut montrer la régularité d'un langage L, par l'une des méthodes suivantes :

- Tous les langages finis sont réguliers
- Si on trouve un AEF qui reconnaît un langage L, alors L est régulier
- Si on trouve une grammaire régulière générant L, alors ce langage est régulier
- On peut exploiter les propriétés de fermeture pour montrer qu'un langage est régulier.

4.12 Lemme de l'étoile (lemme de pompage ou d'itération)

Soit L un langage régulier infini. Il existe un entier k tel que pour tout mot x de L plus long que k, x se factorise en $x = uvw$, avec

(i) $|v| \geq 1$ (ii) $|uv| \leq k$ et (iii) pour tout $i \geq 0$, $uv^i w$ est également un mot de L.

Le lemme de l'étoile est utilisé pour démontrer qu'un langage donné n'est pas régulier, soit en définissant la contraposée du lemme ou en raisonnant par l'absurde. Dans notre cours nous utilisons l'absurde, ainsi nous supposons que le langage est régulier et nous trouvons un mot du langage qui ne vérifie pas la conclusion du lemme.

4.12 Lemme de l'étoile (lemme de pompage ou d'itération)

Exercice : Montrons que le langage $L1 = \{a^i b^i, i \geq 0\}$ n'est pas régulier

On suppose que L1 est régulier.

Soit k la constante du lemme, soit $x = a^k b^k$ un mot de L_1.

Le lemme nous permet de déduire qu'il existe une décomposition $x = uvw$ tel que : $|uv| \leq k$ donc :

$u = a^{k_1}, v = a^{k_2}, w = a^{k_3} b^k$ (c'est le seul type de découpage respectant la contrainte du lemme).

D'après le lemme pour tout i dans N, $uv^i w$ appartient à L_1, donc :

pour i=0, $uv^0 w = a^{k_1} (a^{k_2})^0 a^{k_3} b^k \in L_1$ alors $k_2 \neq 0$ ce qui est impossible. Donc L_1 n'est pas régulier.

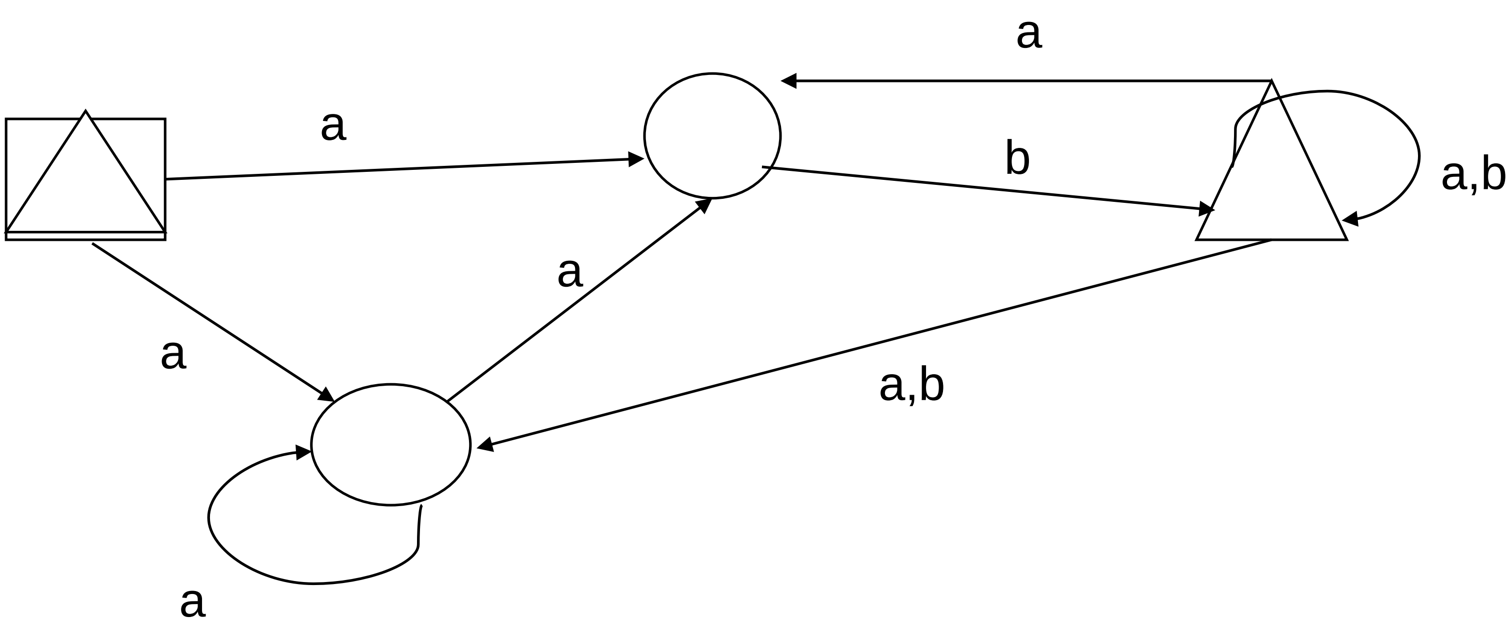
4.13 Propriétés des langages réguliers

- L’union de deux langages réguliers est un langages régulier
- La concaténation de deux langages réguliers est un langage régulier
- L’itération d’un langage régulier est un langage régulier

4.14 Exemple important

- Construire un automate d'états finis pour cette expression régulière :

- **(a* a b (a + b)*)***



Remarque: éviter de mettre la boucle à l'état initial