# Chapitre 4
# Les Expressions Régulières

# Plan

- 1. Définitions
- 2. Théorème de Kleene

  Calculer l’ER associée à un automate

  De l’expression régulière a l’automate
- 3. Lemme de l’étoile

# Langages réguliers

Un langage est dit régulier (rationnel) s'il existe une grammaire régulière qui le génère.

Soit la grammaire G=(VT, VN, S, R),

**Définition d'une grammaire régulière**:

G est dite régulière à si et seulement si toutes ses règles de production ont l'une des formes suivantes : $A \longrightarrow aB$ ou $A \longrightarrow a$ avec $A, B \in VN$ et $a \in VT$.

# langages réguliers

- **Définition**: Un langage est régulier si et seulement s'il existe une grammaire régulière qui l'engendre.

- **Définition**: Un langage est régulier si et seulement s'il existe un automate d'états finis qui le reconnait.

## Propriétés de fermeture de la classe des langages réguliers

- En plus des opérations régulières (. , *, l'union et le miroir), la classe des langages réguliers est fermée par rapport au complément et à l'intersection.

.

# langages rationnels

- **Définition**:

on appelle langage rationnel tout langage que l'on peut exprimer en un nombre fini d'opération (Expressions Régulières).

**Définition**:

Soit X un alphabet, les ER définies sur X et les ensembles qu'elles dénotent sont définis récursivement de la manière suivante:

1. $\varnothing$ est une expression régulière (ensemble vide).
2. a est une ER {a}.
3. Si $w_i \in X$ alors $w_i$ est une ER {wi}.
4. Si E1 et E2 sont deux ER alors E1.E2, E1$\cup$E2 et E1* sont des ER,

# Exemple

- $L = \{\omega \in \{0,1\} * \mathit{tel\ que}\ \omega \equiv 0\ [2]\}$

– E1 = (0∪1)*.0 = X*.0 (des zéros non significatifs).

– E2 = 1.(0∪1)*.0 ∪ 0 (pas de zéro non significatif).

- Priorité sur les opérateurs:

L'étoile de Kleen et le plus en exposant sont plus prioritaires que la concaténation, qui est elle même plus prioritaire que le plus sur la ligne.

* (Itération)

. (Concaténation)

∪ (union)

- Exemple: E = 0.1*∪0 = ((0.(1)*)∪0)

- **Définition**:

deux ER E1 et E2 sont équivalente si et seulement si elle définissent le même langage i.e. L(E1) = L(E2).

- Théorème de Kleen: La classe des langages rationnels est exactement égale à la classe des langages réguliers.

# Propriétés sur les expressions régulières

Si P, Q et R sont trois expressions régulières alors :

- $P + Q = Q + P$ ;
- $(P + Q) + R = P + (Q + R)$;
- $(P.Q).R = P.(Q.R)$;
- $P.\emptyset = \emptyset.P = \emptyset$;
- $P.\varepsilon = \varepsilon.P = P$ ;
- $P + \emptyset = \emptyset + P = P$ ;
- $P*.P = P.P*$;
- $(P + \varepsilon)*.R = P*.R$;
- $P* = \varepsilon + P.P*$;
- $\emptyset* = \varepsilon$;
- $(P*)* = P*$;
- $(P* + Q*)* = (P + Q)* = (P*.Q*)*$.

# De l’AEF $\leftrightarrow$ l’ER

- **Proposition** : Pour toute expression régulière E, il existe un AEF qui reconnaît le langage dénoté par E.

- **Proposition** : Pour tout AEF A, il existe une expression régulière E qui dénote le langage reconnu par A.

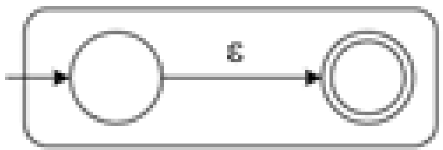
# De l’expression régulière à l’automate

# I. Associer un automate à une ER

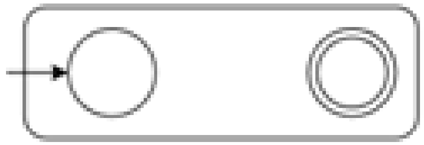
- Il est possible d'associer mécaniquement (et récursivement) un ε-transition à une expression régulière, nous utiliserons pour cela trois automates de base et trois automates génériques.

# Automate de base

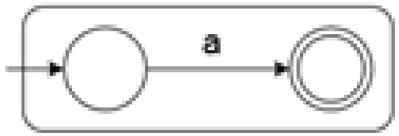
- Le premier automate permet de reconnaitre le langage associé à l'expression régulière ε.



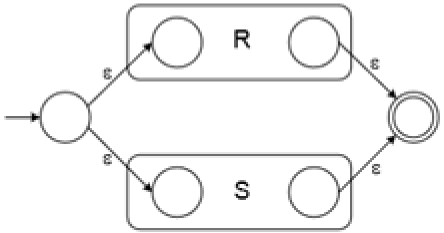

- Le deuxième permet de reconnaitre le langage associé à ∅.



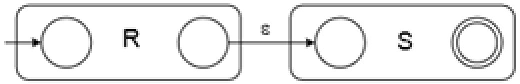
- Le troisième permet de reconnaitre le langage associé à l'expression régulière a.



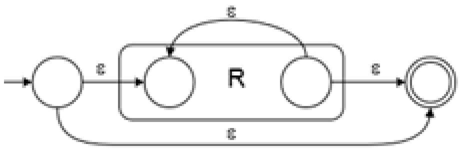
- Les ER sont produites par des opérations d'union, de produit et de clôture. On obtient les trois situations suivantes :
- L'expression R+S :




- L'expression RS :




- L'expression R* :




Note : Concernant l'expression (R) , il suffit d'utiliser l'automate associé à R.

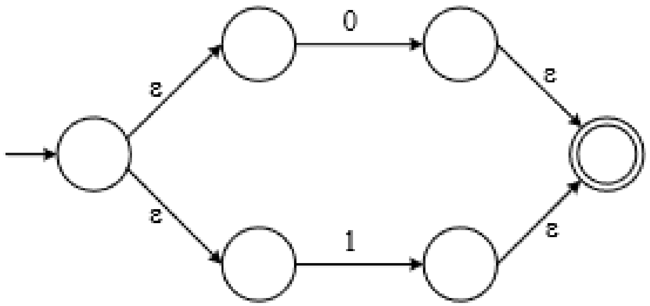
# Exemple

- Construire l'automate associé à l'expression régulière

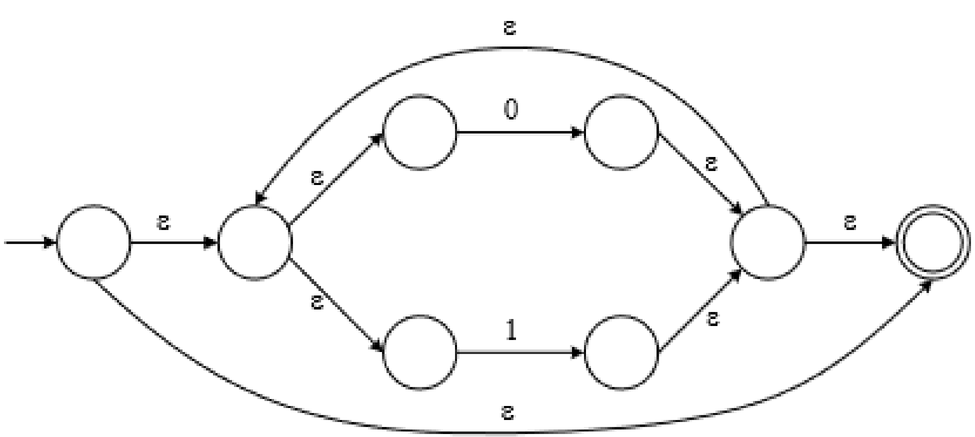
$$\mathbf{(0+1)^*1(0+1)}$$

# Solution

- **Etape 01 :** L’automate **(0+1)**

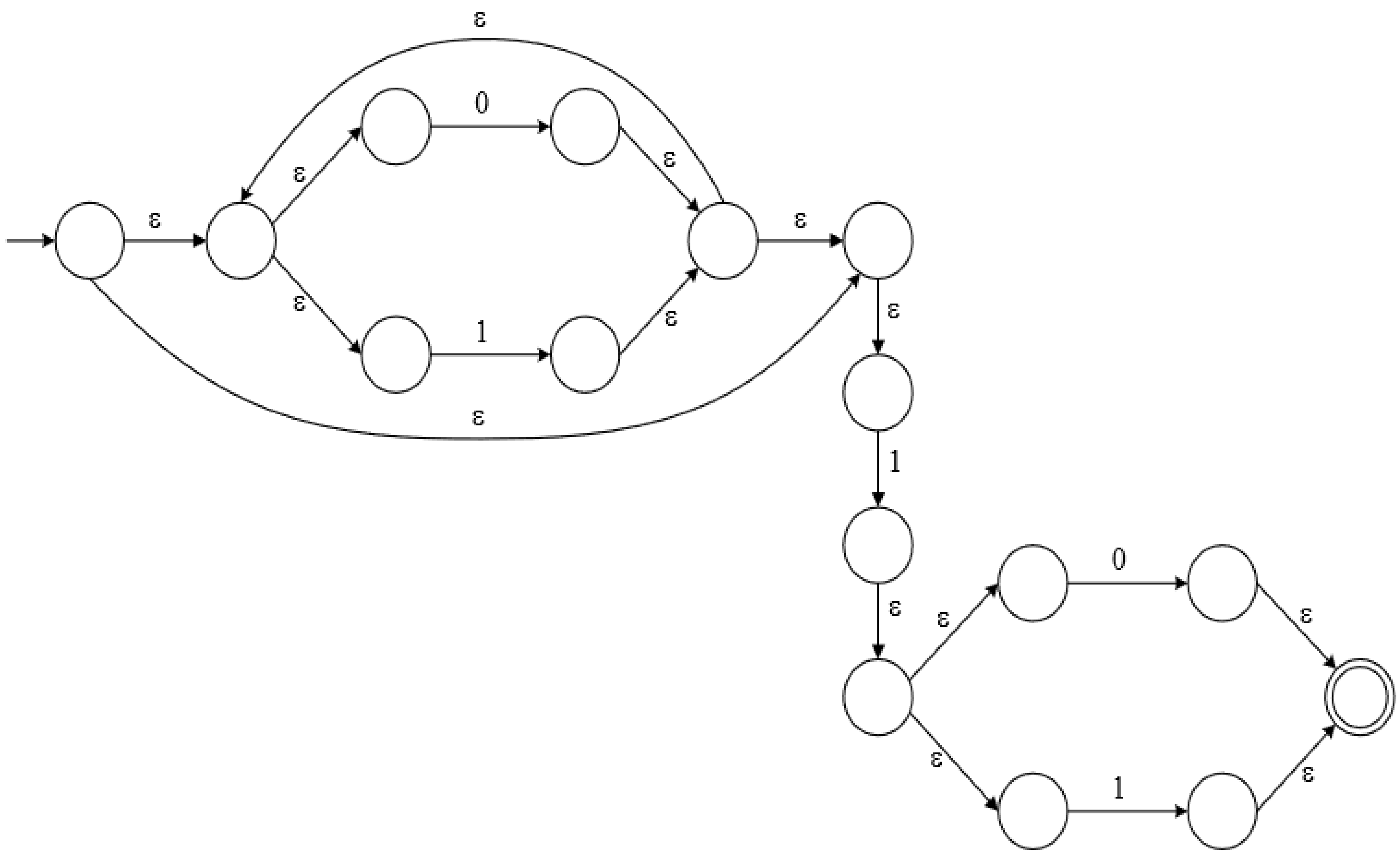



- **Etape 02 :** L’automate **(0+1)***



# Solution

- **Etape 03 :** L’automate **(0+1)*1(0+1)**



# II.Les dérive de nérode

# Dérivées

- **Définition**:

Soit L un langage sur un alphabet X et $\omega \in X^*$, la dérivée de L par rapport à $\omega$, notée $L_{||}\,\omega$, est définie par :

$$L \mid\mid \omega = \{z \in X^* / \omega.z \in L\}.$$

**Exemples** :

Soient les langages $L1 = \{\varepsilon, a, ab, aa, ba\}$ et $L2 = \{a^n / n > 0\}$.

$L1_{||}\, a = \{\varepsilon, b, a\}$;

$L1_{||}\, aa = \{\varepsilon\}$;

$L2_{||}\, a = L2$.

# Dérivées

- **Théorème de Nérode**: Un langage L est régulier sur X* si et seulement si le nombre de dérivées de L est fini.

Exemple: $L = \{a^i b^j \text{ tel que } i,j \geq 0\}$

– $L || a = \{a^i b^j \text{ tel que } i,j \geq 0\} = L$

– $L || b = \{b^j \text{ tel que } j \geq 0\} = L1$

# Propriétés des dérivées

- $a_i \parallel a_j = \begin{cases} \varepsilon & si \quad a_i = a_j \\ \varnothing & sinon \end{cases};$
- $(L_1 \bigcup L_2) \parallel a = L_1 \parallel a \bigcup L_2 \parallel a$
- $(L_1.L_2) \parallel a = \begin{cases} (L_1 \parallel a).L_2 & si \quad \varepsilon \notin L_1 \\ (L_1 \parallel a).L_2 \cup L_2 \parallel a & sinon \end{cases};$
- $L^* \parallel a = (L \parallel a).L^*;$
- $L \parallel \omega_1.\omega_2 = (L \parallel \omega_1) \parallel \omega_2;$
- $L \parallel \omega = \varnothing$ si aucun mot de $L$ ne commence par $\omega$;
- $\varepsilon \parallel a = \varnothing;$
- $\omega.L \parallel \omega = L.$

# Exemple1

**$L = \{a^i b^j$ tel que $i,j \geq 0\}$**

$L//\varepsilon = L$

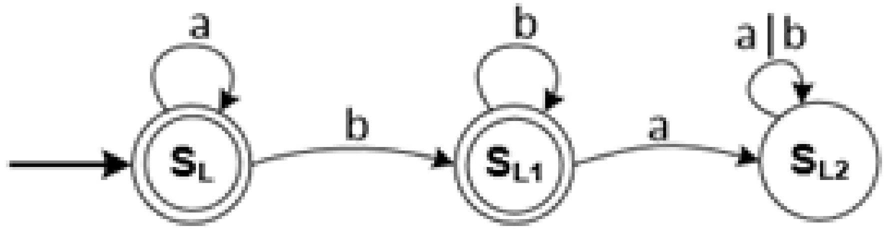
$L//a = \{a^i b^j$ tel que $i,j \geq 0\} = L$

$L//b = \{b^j$ tel que $j \geq 0\} = L_1$

$L_1//a = b//aL_1 = \varnothing L_1 = \varnothing = L_2$

$L_1//b = L_1$

$L_2//a = L_2//b = L_2$



# Exemple 2

**$L = \{a^i b^j$ tel que $i, j > 0\}$**

$L//a = \{a^i b^j$ tel que $i \geq 0$ , $j > 0\} = L_1$

$L//b = \varnothing = L_2$

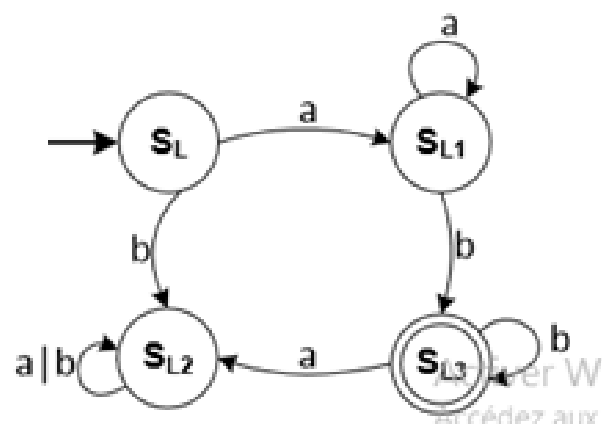
$L_1//a = L_1$

$L_1//b = \{b^j$ tel que $j \geq 0\} = L_3$

$L_2//a = L_2//b = L_2$

$L_3//a = \varnothing = L_2$

$L_3//b = L_3$



# Exemple 3

- Soit le langage L =(a+b)*ab(bb+a)*. Calculer L | | a.

$$L \parallel a = (a+b)^* \parallel a.ab(bb+a)^* + ab(bb+a)^* \parallel a$$
$$=(a+b) \parallel a.(a+b)^*ab(bb+a)^* + b(bb+a)^*$$
$$=(a \parallel a + b \parallel a).a.(a+b)^*ab(bb+a)^* + b(bb+a)^*$$
$$=(a+b)^*ab(bb+a)^* + b(bb+a)^*.$$

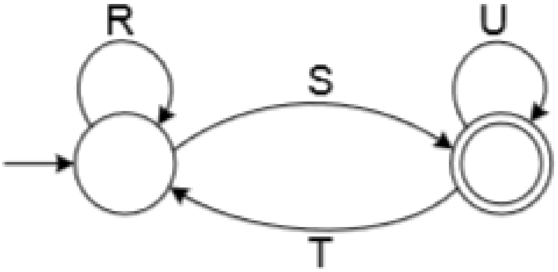
# Calculer l'ER associée à un automate

# Réduction d'automate

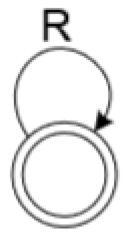
- La démarche pour construire une expression régulière à partir d'un automate est la suivante :

- Pour chaque état d'acceptation q, éliminer tous les états intermédiaire entre $e_0$ (état de départ) et q ;
- Si q ≠ e0, on obtient un automate à deux états.

L'expression régulière associée au langage est alors : (R+SU*T)*SU*



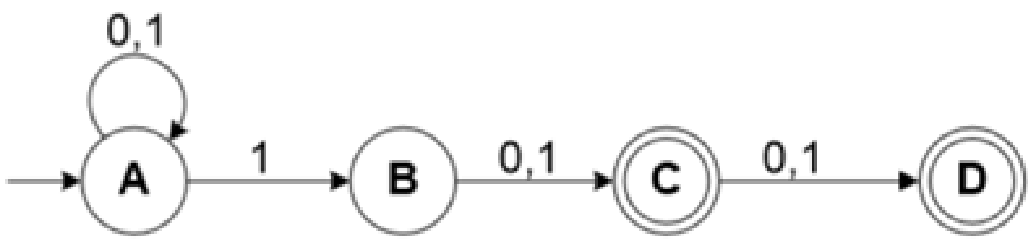
- Si $e_0$ est un état d'acceptation, alors on obtient un automate à un unique état. L'expression régulière associée au langage et alors : R*




- L'expression régulière représentant l'automate est alors l'union de toutes les expressions calculées à partir des automates réduites en appliquant les règles 2) et 3) pour chacun des états d'acceptation de l'automate initial.

# Exemple

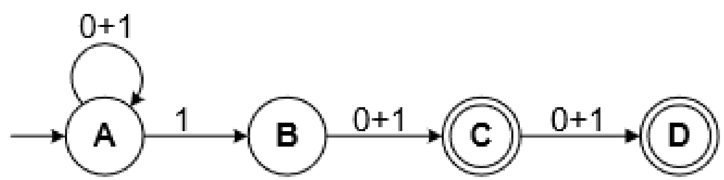
- Trouver l’ER de l’automate fini non déterministe suivant:



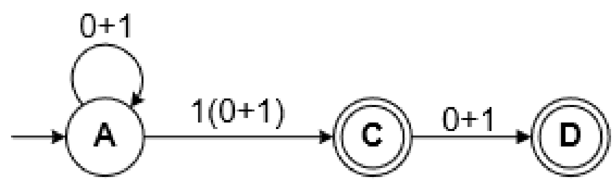

**Un AFN acceptant les mots ayant un 1 en avant dernier ou antépénultième position**

# Solution

- **<u>Etape 01 :</u>** Convertir l'automate en un automate étiqueté par des expressions régulières

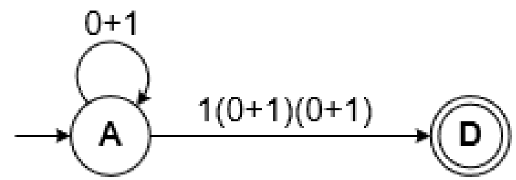



- **<u>Etape 02 :</u>** Eliminer l'état **B**



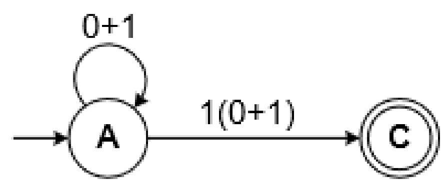
# Solution

- **Etape 03 :** Eliminer l'état **C**




L'expression régulière associée à cet automate est : **(0+1)*1(0+1)(0+1)**

- **Etape 04 :** Eliminer l'état **D**



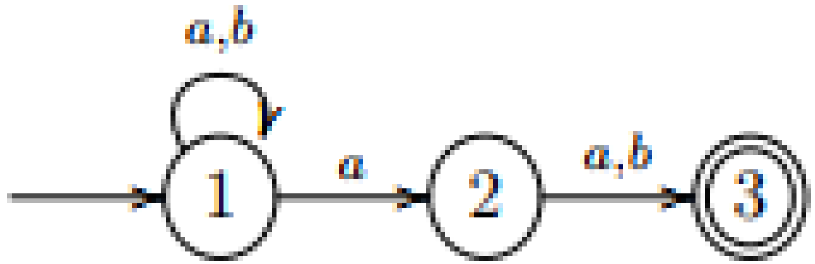



L'expression régulière associée à cet automate est : **(0+1)*1(0+1)**

# Equation d’Arden

# Des automates d'états fini aux équations d'Arden

Considérons l'automate




On note Lq le langage reconnu par l'état q de l'automate. On peut alors décrire l'automate comme un système d'équations sur les langages :

$$\begin{cases} L_1 & = & (a+b).L_1 + a.L_2 \\ L_2 & = & (a+b).L_3 \\ L_3 & = & \epsilon \end{cases}$$

Le langage reconnu par l'automate est le langage de son état initial

# Des équations d'Arden aux expressions régulières

- Pour obtenir l'expression régulière qui correspond à chaque langage Lq on résout le système d'équations à l'aide du lemme d'Arden.

**Lemme d'Arden**

*Soient* $A$ *et* $B$ *des langages ou des automates ou des expressions régulières (les trois représentations sont équivalentes), l'équation de langage* $L = A \cdot L \cup B$ *avec* $\epsilon \notin A$ *admet pour unique solution le langage* $L = A^* \cdot B$

# Application

- En appliquant le lemme d'Arden au système d'équations précédent on obtient

$$\begin{cases} L_1 &= \underbrace{(a+b)}_{A}.L_1 + \underbrace{a.L_2}_{B} = \underbrace{(a+b)^*}_{A}.\underbrace{a.L_2}_{B} = (a+b)^*.a.(a+b) \\ L_2 &= (a+b).L_3 = (a+b).\epsilon = (a+b) \\ L_3 &= \epsilon \end{cases}$$

# Exercice

- On considère le système d'équations suivant :

$$\begin{cases} L_1 = aL_1 + bL_2 + cL_3 \\ L_2 = cL_2 + bL_1 + aL_3 \\ L_3 = c + aL_2 + bL_3 \end{cases}$$

Trouver l'expression rationnelle de L1.

# Lemme d'itération (Lemme de l'étoile)

- Pour tout langage régulier L, il existe un entier $n \in \mathbb{N}$ tel que $\forall \omega \in L, |\omega| \geq n$ On peut décomposer en $uvy, u, y \in X^*$ et $v \in X^+$ tel que $uv^*y \in L$ $(uv^iy \in L, i \geq 0)$.
- Exemple: Montrer que le langage **L={a^i b^i , i ≥0}** n'est pas un langage régulier ($L \notin Reg(X^*)$).

**Démonstration par l'absurde:**

- Supposons que L est régulier :
- $\forall \omega \in L$ on a

$$\omega = uvy, u, y \in X^* \text{ et } v \in X^+ \text{ tel que } uv^*y \in L$$

- On prend : $\omega = a^n b^n$

# Exemple

1) $v \in a^+ \Rightarrow |v| = k$ , $k > 0$.

$\Rightarrow \omega = a^n b^n = a^{(n-k)} a^k b^n$

$\Rightarrow a^{(n-k)} (a^k)^i b^n \in L$ , $i \geq 0$

Si $i=0$ alors $a^{(n-k)} b^n \in L$ contradiction pour $k>0$.

2) $v \in b^+ \Rightarrow |v| = k$ , $k > 0$.

$\Rightarrow \omega = a^n b^n = a^n b^k b^{n-k}$

$\Rightarrow a^n (b^k)^i b^{n-k} \in L$ , $i \geq 0$

Si $i=0$ alors $a^n b^{n-k} \in L$ contradiction pour $k>0$.

# Exemple (suite)

3) $v \in a^+b^+ \Rightarrow v = a^{k1}b^{k2}$ , k1, k2 > 0.

$\Rightarrow \omega = a^n b^n = a^{n-k1}a^{k1}b^{k2}b^{n-k2}$

$\Rightarrow a^{n-k1}(a^{k1}b^{k2})^i b^{n-k2} \in L$ , $i \geq 0$

Si i=2 alors $a^{n-k1}a^{k1}b^{k2}a^{k1}b^{k2}b^{n-k2} \in L$ contradiction.

Donc L n'est pas un langage régulier ($L \notin Reg(X^*)$).

# 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 utiliser le théorème de Nerode pour montrer qu'un langage est régulier;
- On peut exploiter les propriétés de fermeture pour montrer qu'un langage est régulier.

# Méthodes pour montrer qu'un langage n'est pas régulier

- Pour montrer l'irrégularité d'un langage L, il ne suffit pas de ne pas pouvoir trouver un AEF le reconnaissant, on peut utiliser les deux méthodes suivantes pour le faire :

- Raisonnement par l'absurde pour le théorème de l'étoile;
- Exploitation des propriétés de fermeture des langages non réguliers.