NGUYEN Thi Minh Tuyen

Théorie des automates et langages formels

Langages réguliers

Théorème

Théorème: Soit L_1 et L_2 deux langages réguliers. Alors les langages L_1+L_2, L_1L_2, et L_1^* sont aussi des langages réguliers.

Preuve 1: Si $\mathbf{L_1}$ et $\mathbf{L_2}$ sont deux langages réguliers alors il existe des expressions régulières $\mathbf{r_1}$ et $\mathbf{r_2}$ qui correspondent à $\mathbf{L_1}$ et $\mathbf{L_2}$.

- Le langage associé à l'expression régulière $\mathbf{r_1+r_2}$ est $\mathbf{L_1+L_2}$.
- Le langage associé à l'expression régulière $\mathbf{r_1r_2}$ est $\mathbf{L_1L_2}$.
- Le langage associé à l'expression régulière $\mathbf{r_1^*}$ est $\mathbf{L_1^*}$.

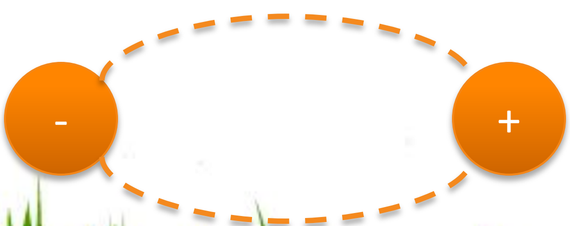
Donc $\mathbf{L_1+L_2}$, $\mathbf{L_1L_2}$, et $\mathbf{L_1^*}$ sont représentés par des expressions régulières.

Par définition ils sont donc des langages réguliers.

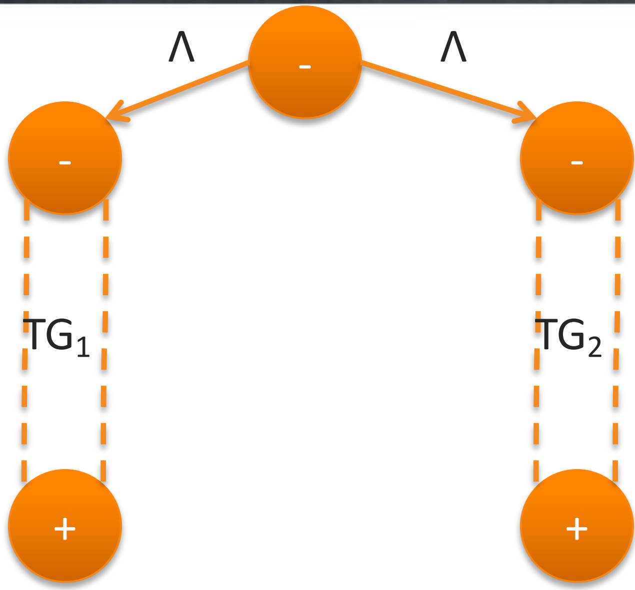
Preuve 2: en utilisant le théorème de Kleene [1]

Si L_1 et L_2 sont deux langages réguliers, alors il existe des expressions régulières $\mathbf{r_1}$ et $\mathbf{r_2}$ qui les représentes .

Par le théorème de Kleene, ils existes des graphes de transitions qui correspondent à L_1 et L_2 . On peut transformer ces graphes en graphes de transitions avec un seul état de départ sans entrée et un seul état final sans sortie. Soit TG_1 et TG_2 les deux graphes de transitions sous cette forme et qui reconnaissent L_1 et L_2.

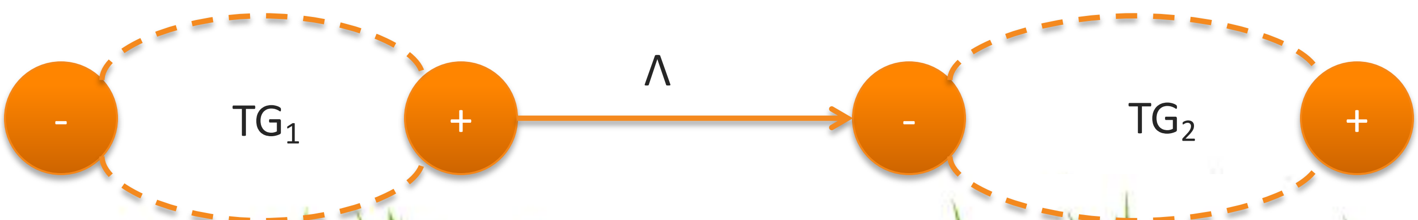


Preuve 2: en utilisant le théorème de Kleene [2]



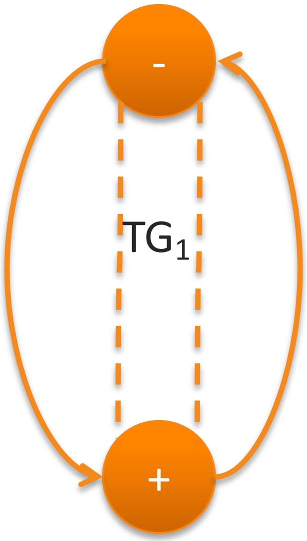
Un graphe de transition qui reconnaît L_1+L_2.

Un graphe de transition qui reconnaît L_1L_2.



Preuve 2: en utilisant le théorème de Kleene [3]

Un graphe de transition qui reconnaît L_1*.



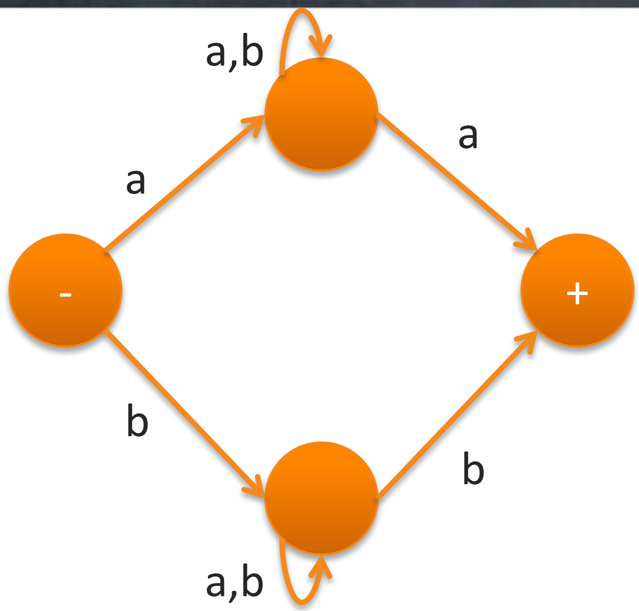
- Il existe des graphes de transitions qui reconnaissent L_1+L_2, L_1L_2, et L_1*. Alors, il existe des expressions régulières qui les représentes (Par le théorème de Kleene). Donc et par définition

$$L_1+L_2,\ L_1L_2,\ \text{et } L_1{}^*$$

sont des langages réguliers.

Exemple

TG_1



Les mots qui commencent et se terminent par la même lettre.

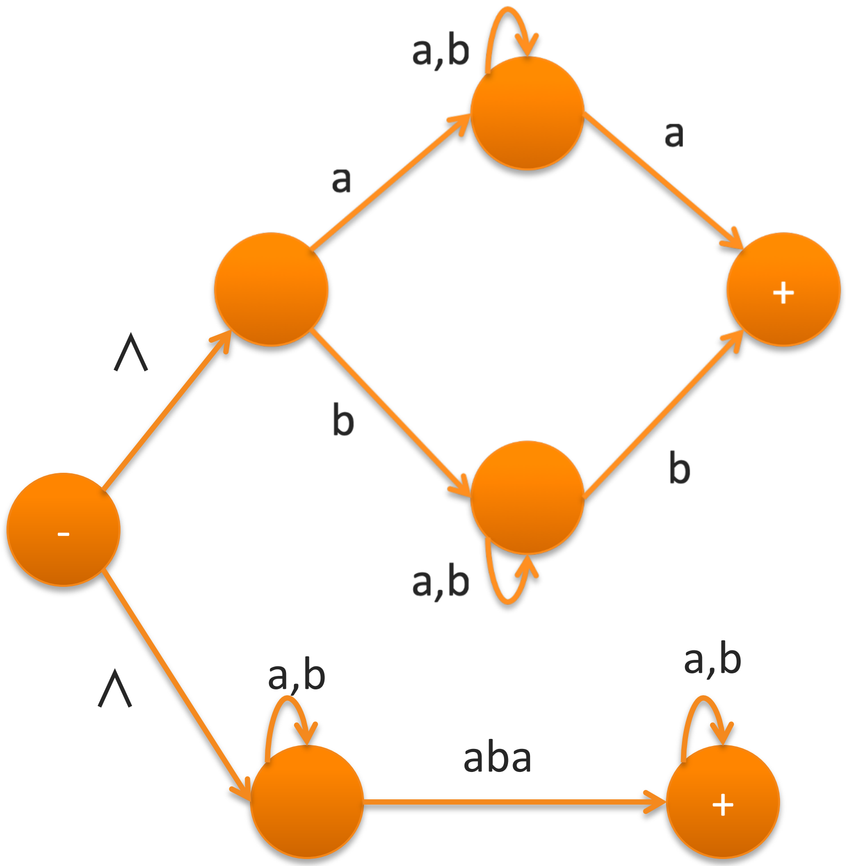
$L_1 = a(a+b)^*a + b(a+b)^*b$

TG_2

Les mots qui contiennent la suite **aba**.

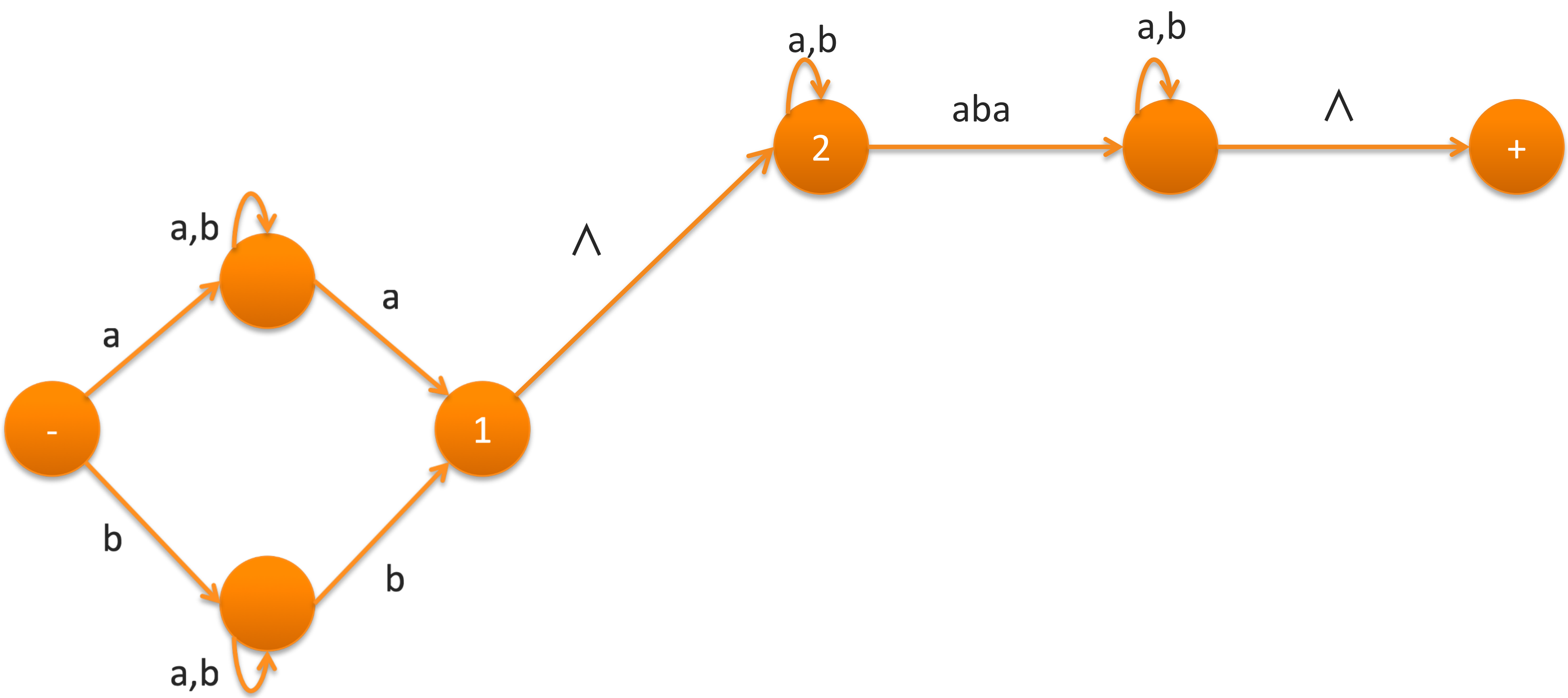
$\mathbf{L_2 = (a+b)^*aba(a+b)^*}$

$L_1 + L_2$



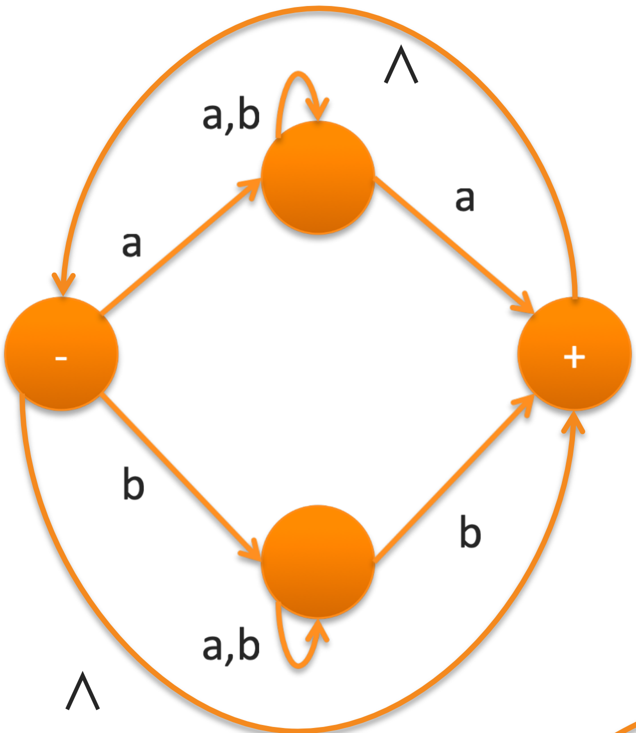
(a+b)*aba(a+b)* + a(a+b)*a + b(a+b)*b

L_1L_2



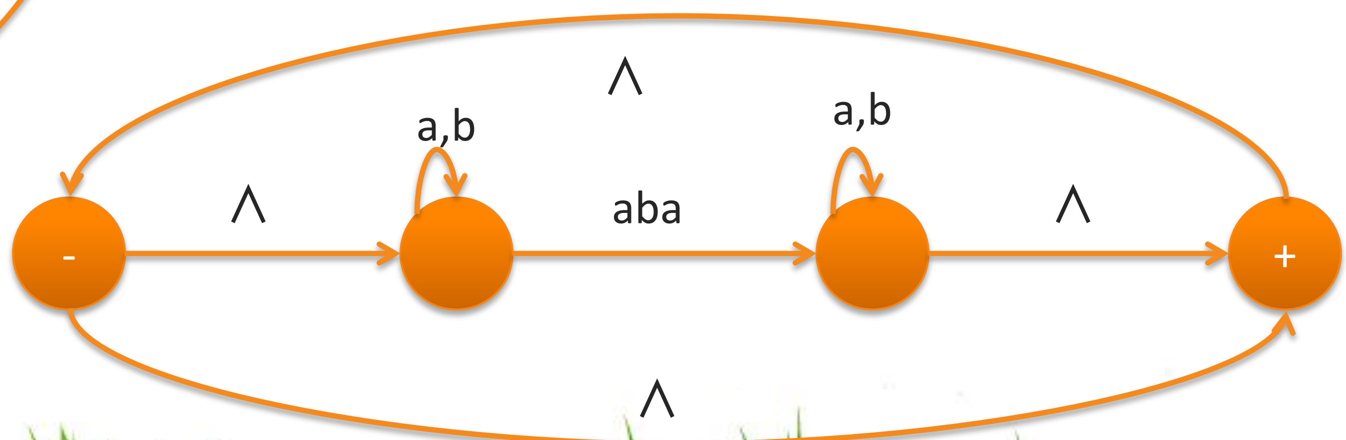
(a(a+b)*a + b(a+b)*b)((a+b)*aba(a+b)*)

L*



$$L_1^* = (a(a+b)^*a + b(a+b)^*b)^*$$

$$L_2^* = ((a+b)^*aba(a+b)^*)^*$$



Définition: le complément d'un langage

- Soit L un langage sur l'alphabet Σ, le complément de L, dénoté L', est le langage de tous les mots de Σ^* qui ne sont pas dans L ($L' = \Sigma^* - L$).
- Exemple:

  S = {a,b}

  L= tous les mots qui contiennent aa.

  L'=?

  $b^*(abb^*)^*(a+\Lambda)$

  Remarque: $(L')' = L$.

Théorème

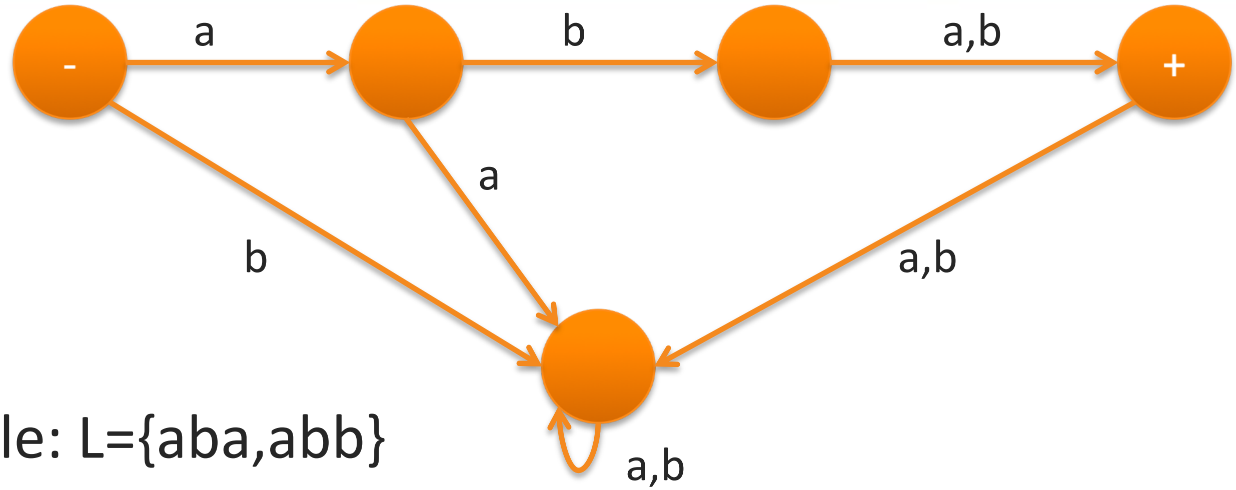
Théorème: Si L est un langage régulier, alors L' est aussi un langage régulier.

Démonstration: Il existe un automate fini qui reconnaît L (d'après le théorème de Kleene). Donc, tout mot qui est accepté par cet automate se termine dans un état final, et tout mot qui n'est pas accepté se termine dans un état non-final.

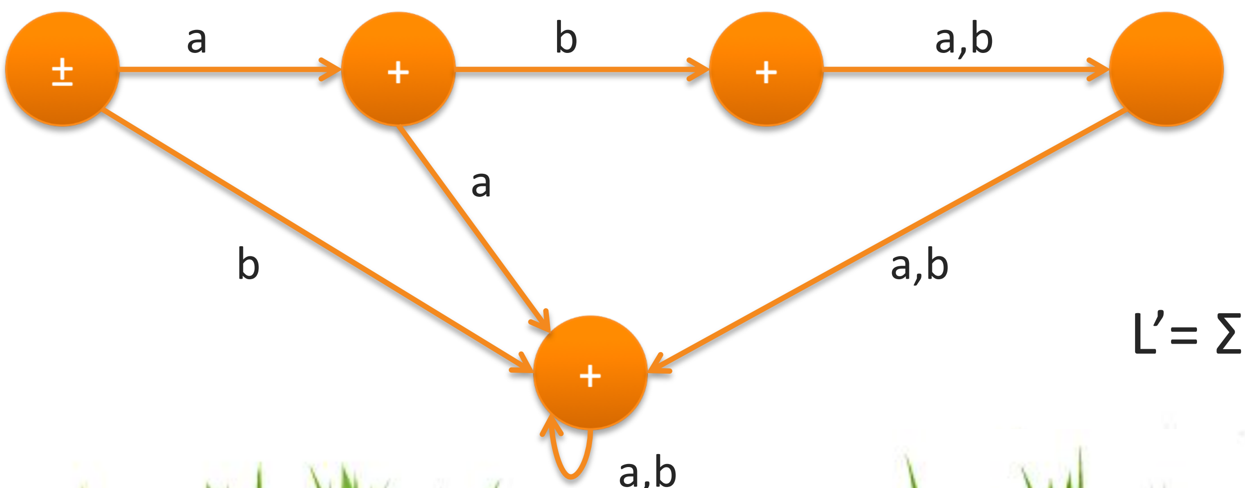
Renversons les états : tout état final devient état non-final, tout état non-final devient état final.

Le nouvel automate accepte exactement tous les mots qui ne sont pas dans L. Donc L' est régulier.

Exemple



Exemple: L={aba,abb}



L’= Σ* - {aba,abb}

Théorème: Soit $\mathbf{L_1}$ et $\mathbf{L_2}$ deux langages réguliers, alors $\mathbf{L_1 \cap L_2}$ est aussi un langage régulier.

- **Démonstration**:

$$L_1 \cap L_2 = (L_1' + L_2')'.$$

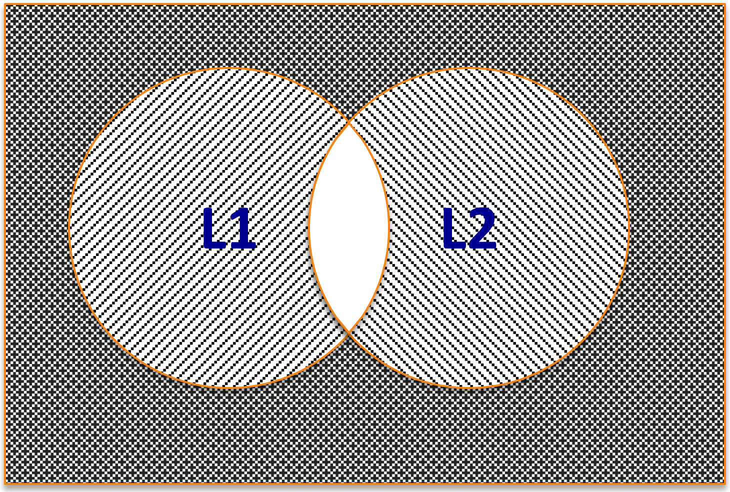
Si L_1 et L_2 sont réguliers, alors L_1' et L_2' sont aussi réguliers.

Si L_1' et L_2' sont réguliers, alors $L_1'+L_2'$ est régulier.

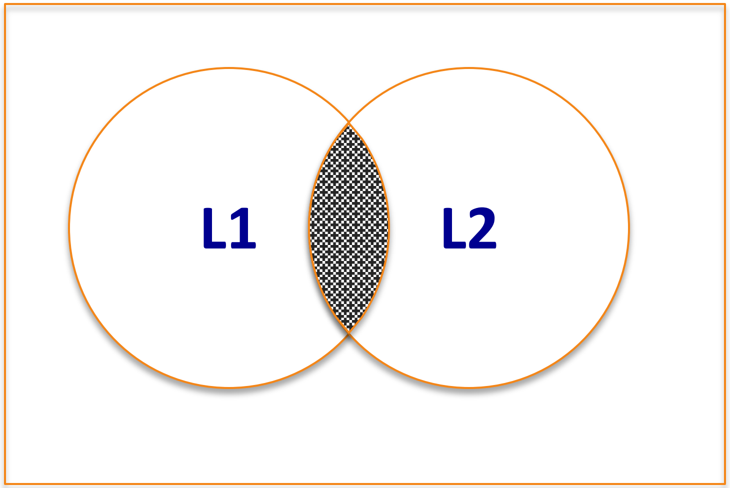
Si $L_1'+ L_2'$ est régulier, alors $(L_1'+L_2')'$ est régulier.

Donc, $L_1 \cap L_2$ est un langage régulier.

$L_1 \cap L_2 = (L_1' + L_2')'$

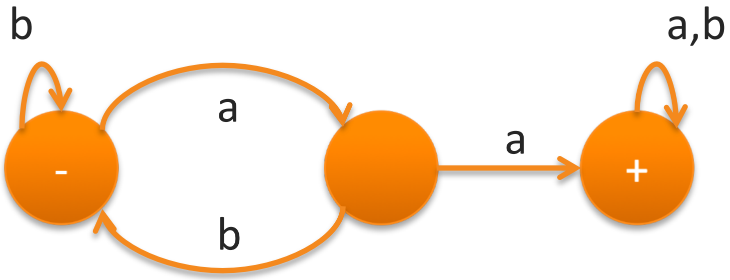


$L_1' + L_2'$

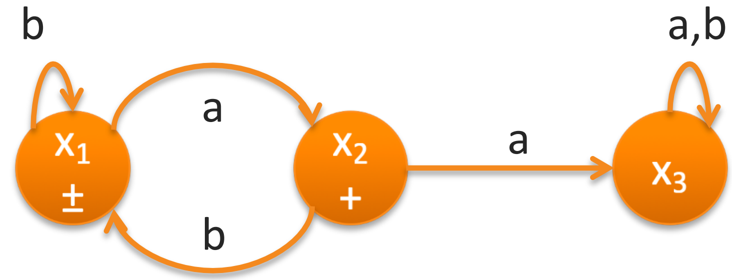


$(L_1' + L_2')'$

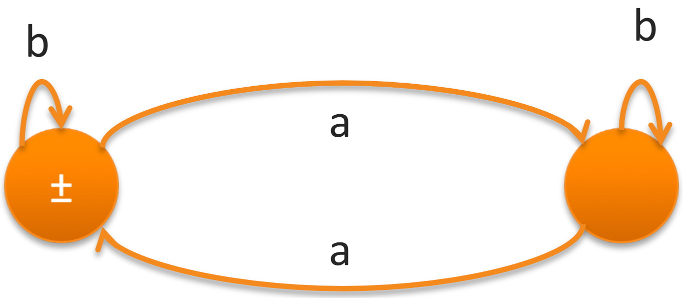
Exemple



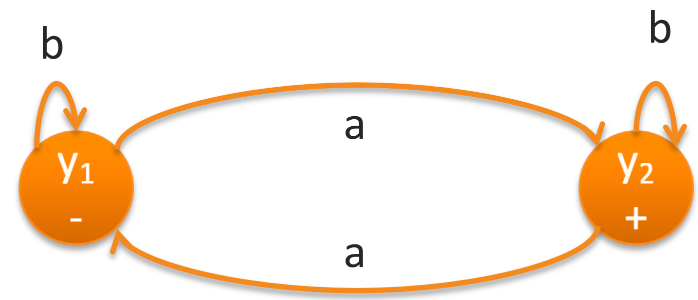
L_1: double a



L'$_1$

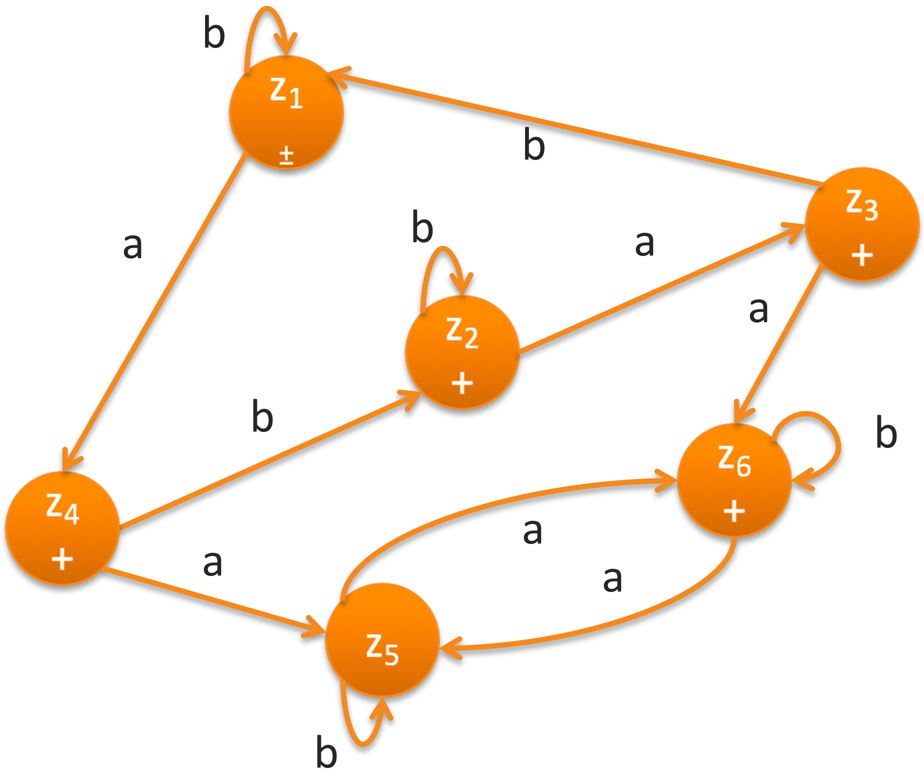


L_2: nombre paire de a

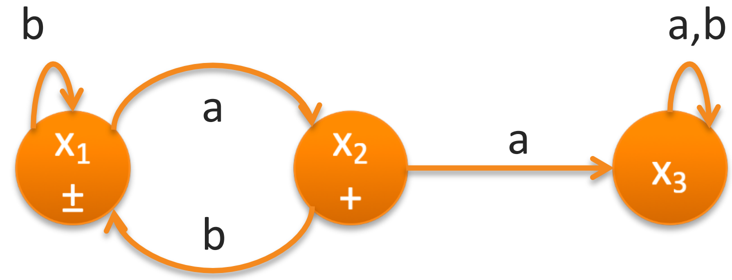


L'$_2$

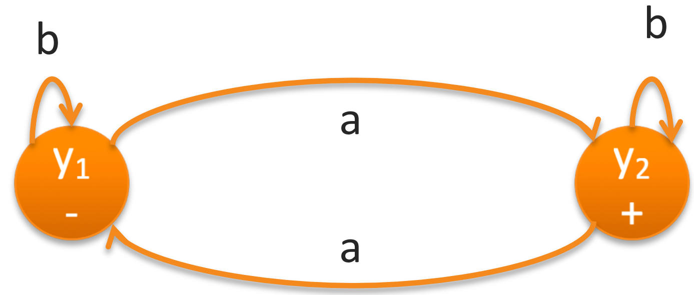
Exemple



$L_1' + L_2'$

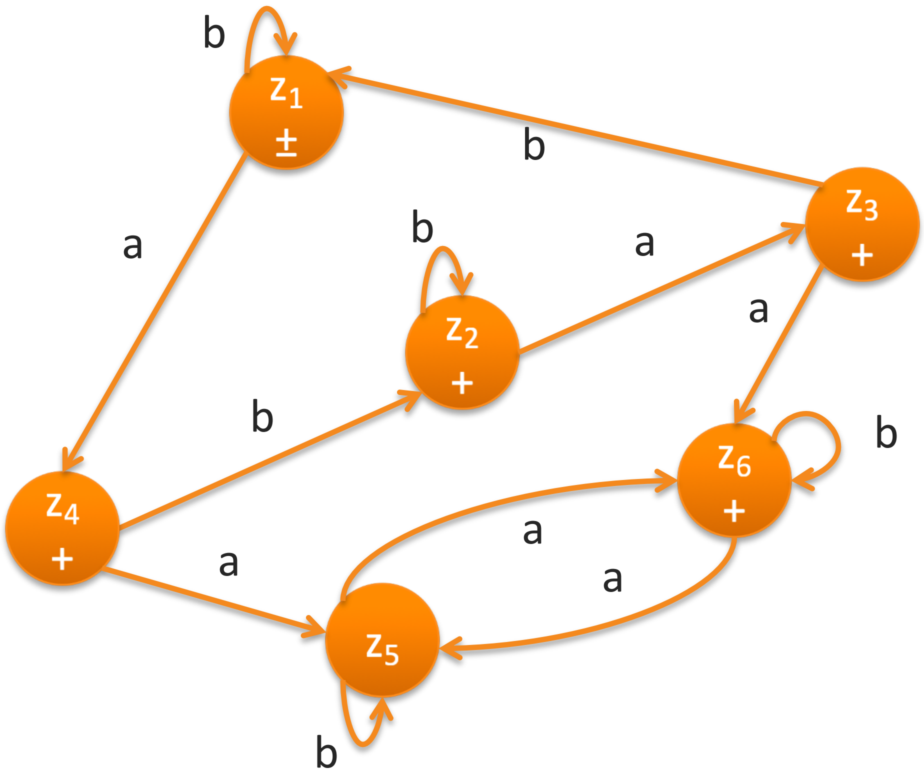


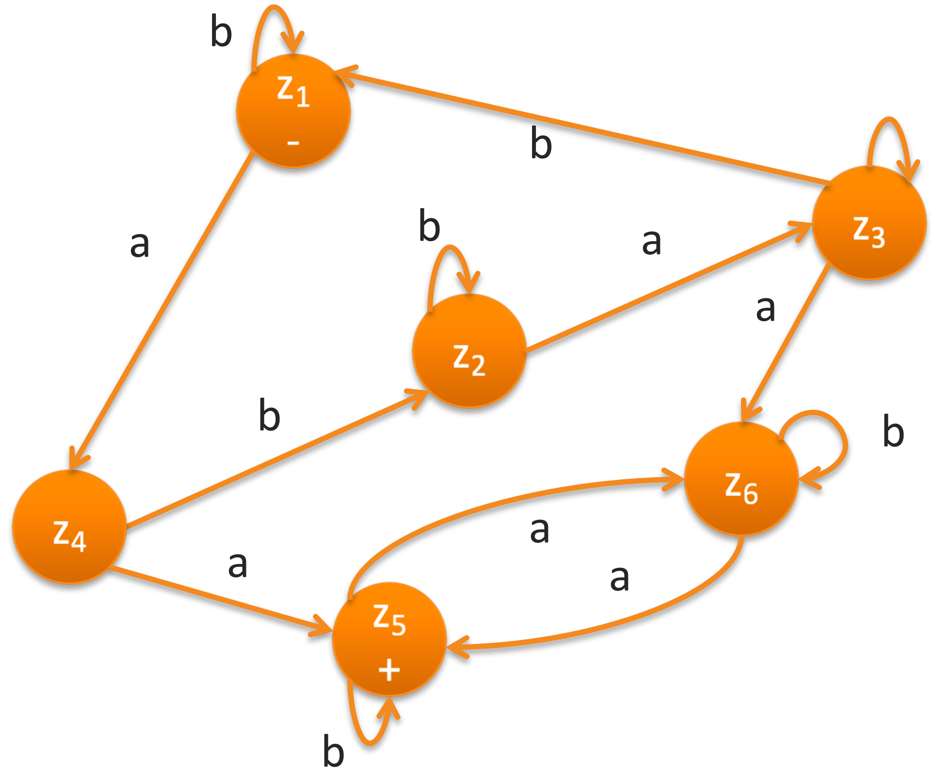
L'_1



L'_2

Exemple





$L_1' + L_2'$

$(L_1'+L_2')'=L_1 \cap L_2$

Construire un automate fini pour $L_1 \cap L_2$ à partir des automates finis de L_1 et de L_2.

1. On utilise le même algorithme que celui de la construction de l'automate fini pour L_1+L_2 (vu dans la preuve du théorème de Kleene).
2. L'unique différence est dans les états finaux: un état $\{x_i, y_j\}$ est final si est seulement si les deux états x_i, y_j sont finaux dans les deux automates finis L_1 et L_2.

Question?