Les expressions régulières

Samia Mazouz
USTHB Département Informatique
2015-2016

Expressions Régulières

Les expressions régulières sur un alphabet X sont définies comme suit :

Cas de base

- $\varnothing$ est une expression régulière et **décrit le langage vide**
- **ε** est une expression régulière et **décrit le langage {ε}**
- $\forall$**a**$\in$**X, a** est une expression régulière et **décrit le langage {a}**

Induction

- Si r et s sont deux expressions régulières sur décrivant respectivement les langages R et S alors :
- **r+s** est une expression régulière et **décrit le langage R**$\cup$**S**
- **r.s** est une expression régulière et **décrit le langage R.S**
- **r*** est une expression régulière et **décrit le langage R***
- **(r)** est une expression régulière et **décrit le langage R.**

Expressions Régulières

Remarque

Pour simplifier les expressions, nous supposons que l'étoile '*' est plus prioritaire que la concaténation ' .'qui est plus prioritaire que l'addition '+'.

Exemples

- (a+b)* dénote le lange ({a}$\cup${b})* qui correspond à tous les mots sur {a, b}
- (a+b)*aab représente tous les mots de {a, b}* se terminant par aab
- (a+b)*aba(a+b)* représente tous les mots de {a, b}* ayant aba comme sous mot.

On note Rat(X*) l'ensemble des expressions régulières sur l'alphabet X.

Expressions Régulières

Définition

Deux expressions régulières sont équivalentes si elles dénotent le même langage.

Propriétés sur les expressions régulières

- Commutativité : $p + q = q + r$
- Associativité : $p+ (q + r) = (p + q) + r$ $\quad p(qr) = (pq)r$
- Distribution : $(p + q)r = pr + qr$ $\quad p(q+r) = pq+pr$
- Elément absorbant : $p. \emptyset = \emptyset.p= \emptyset$
- Elément neutre : $p.\varepsilon = \varepsilon.p = p$ $\quad p+ \emptyset = \emptyset +p = p$
- $p. p^* = p^*.p$ $\quad p^* = (p+ \varepsilon)^*$ $\quad p^*.p^* = p^*p^* = \varepsilon + p.p^*$

  $\emptyset^* = \varepsilon$ $\quad (p^*)^* = p^*$
- $(p^*+q^*)^* = (p+q)^* = (p^*.q^*)^*$

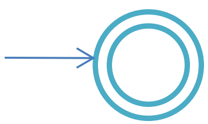
Expression Régulière et Automate d’Etats Fini

Proposition

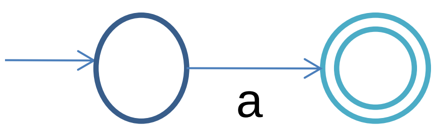
A toute expression régulière E, il existe un automate d’états fini A(E) tel que reconnaissant le langage dénoté par E.

Démonstration

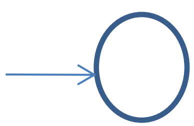
- Pour ε, on lui associe l’automate :



- Pour a, on lui associe l’automate :

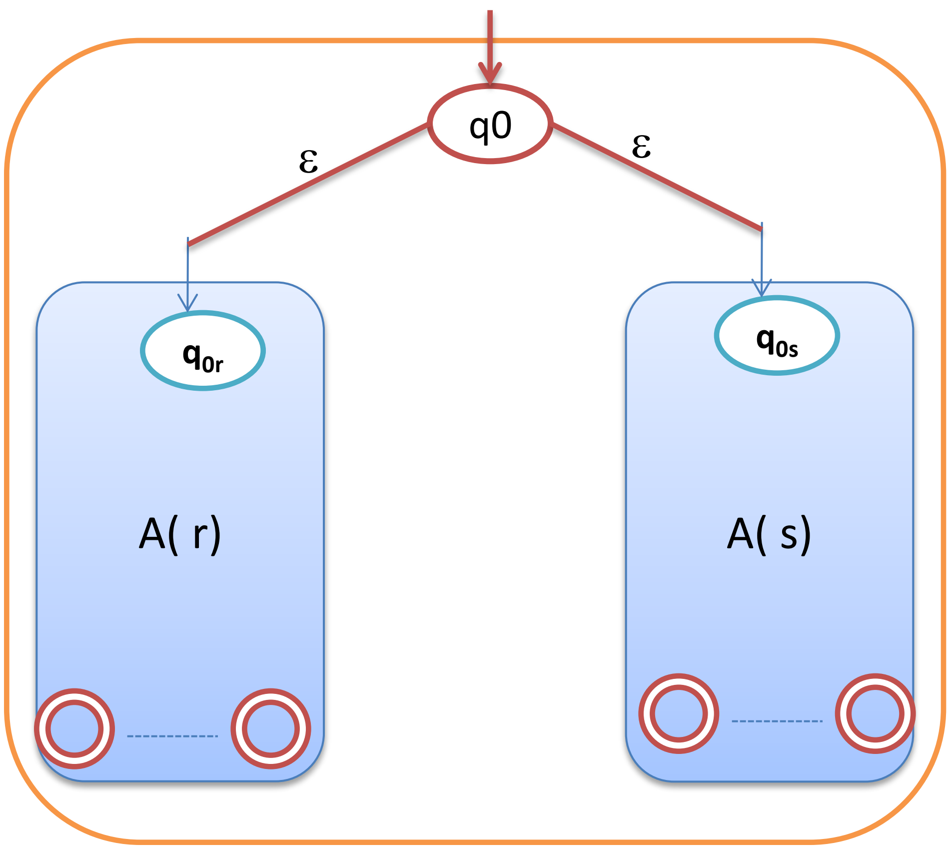


- $\varnothing$, on associe l’automate :



Expression Régulière et Automate d'Etats Fini

- Pour r+s, on associe l'automate A(r+s)



A(r)=(Xr, Qr, q0r, δr, Fr)

A(s)= (Xs,Qs,q0s, δs, Fs)

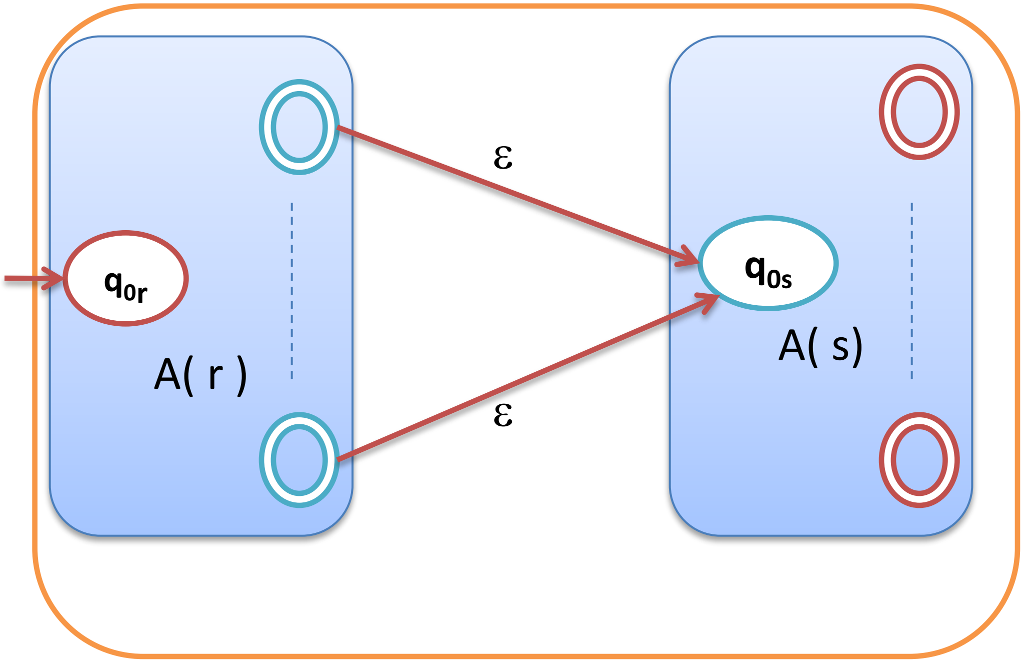
A(r+s)=(X, Q, q0, δ, F) tq

- X = Xr ∪ Xs
- Q = Qr ∪ Qs ∪ {q0}
- q0 / q0 ∉ Qr ∪ Qs
- δ = δr+ δs +

  δ(q0, ε)={q0r, q0s}
- F =Fr ∪ Fs

Expression Régulière et Automate d'Etats Fini

- Pour r.s, on associe l'automate A(r.s)



A(r)=(Xr, Qr, q0r, δr, Fr)

A(s)= (Xs,Qs,q0s, δs, Fs)

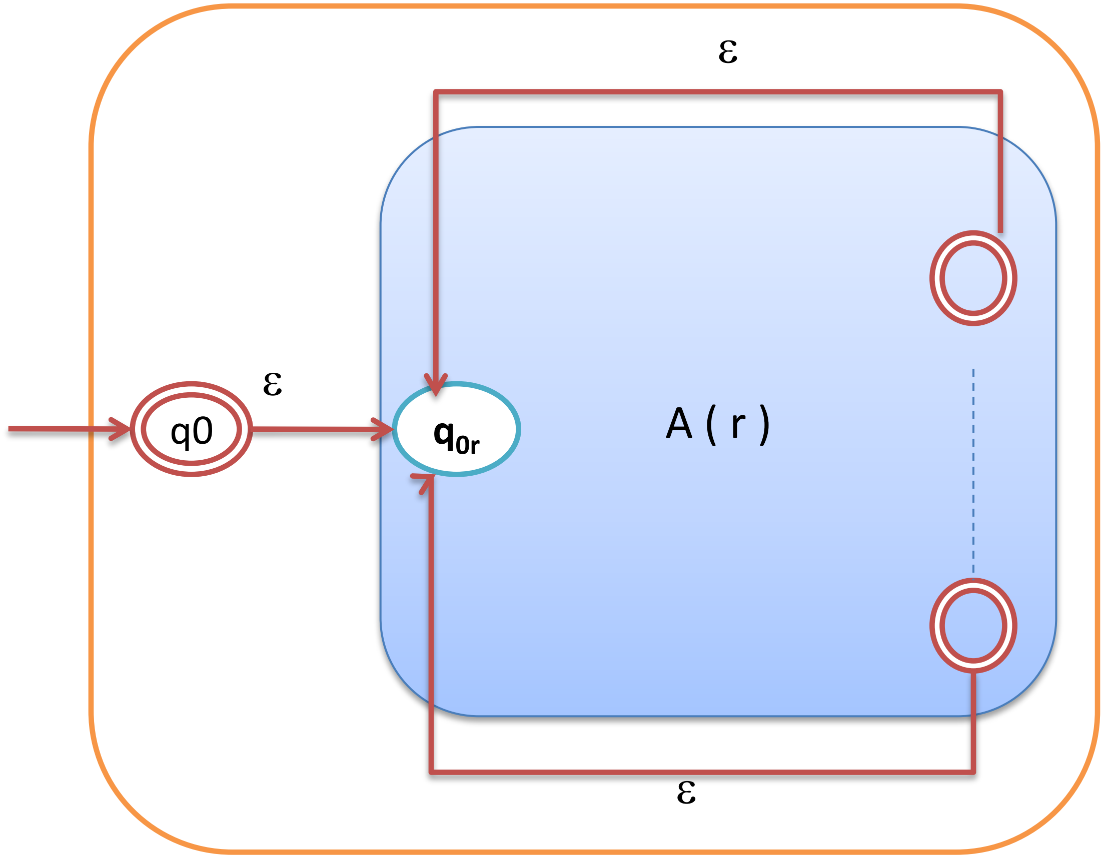
A(r.s)=(X, Q, q0, δ, F) tq

- X = Xr ∪ Xs
- Q = Qr ∪ Qs
- q0=q0r
- δ = δr+ δs +

  ∀q∈Fr, δ(q, ε)=q0s
- F =Fs

Expression Régulière et Automate d'Etats Fini

- Pour r^*, on associe l'automate $A(r^*)$:



$A(r)=(Xr, Qr, q0r, \delta r, Fr)$

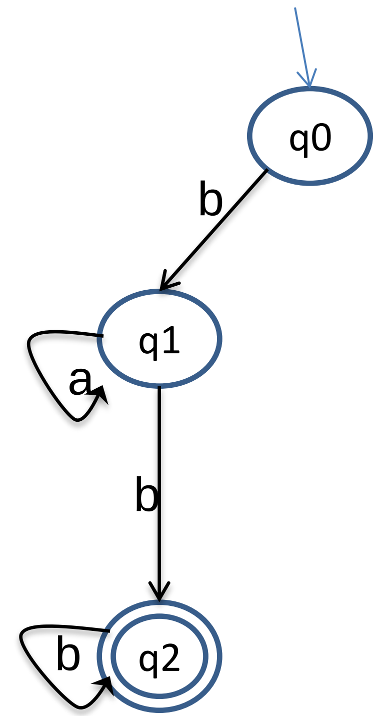
$A(r^*)=(X, Q, q0, \delta, F)$ tq

- $X = Xr$
- $Q = Qr \cup \{q0\}$
- $q0$ / $q0 \notin Qr$
- $\delta = \delta r$
  - $+ \delta(q0, \varepsilon)=q0r$
  - $+ \forall q \in Fr, q0r \in \delta(q, \varepsilon)$
- $F = Fr \cup \{q0\}$

Expression Régulière et Automate d'Etats Fini

Proposition

A tout automate d'états fini A lui correspond une expression régulière qui le dénote.



Le langage reconnu par cet automate est dénoté par l'expression régulière   ba*bb*

Remarque

Plusieurs méthodes permettant de trouver l'expression dénotant le langage reconnu par un automate d'états fini ont été proposées dans la littérature.

Transformation d'un automate en ER par élimination d'états

On cherche à trouver une expression régulière dénotant l'automate A. La méthode est basée sur la suppression successive des états et transitions de l'automate A.

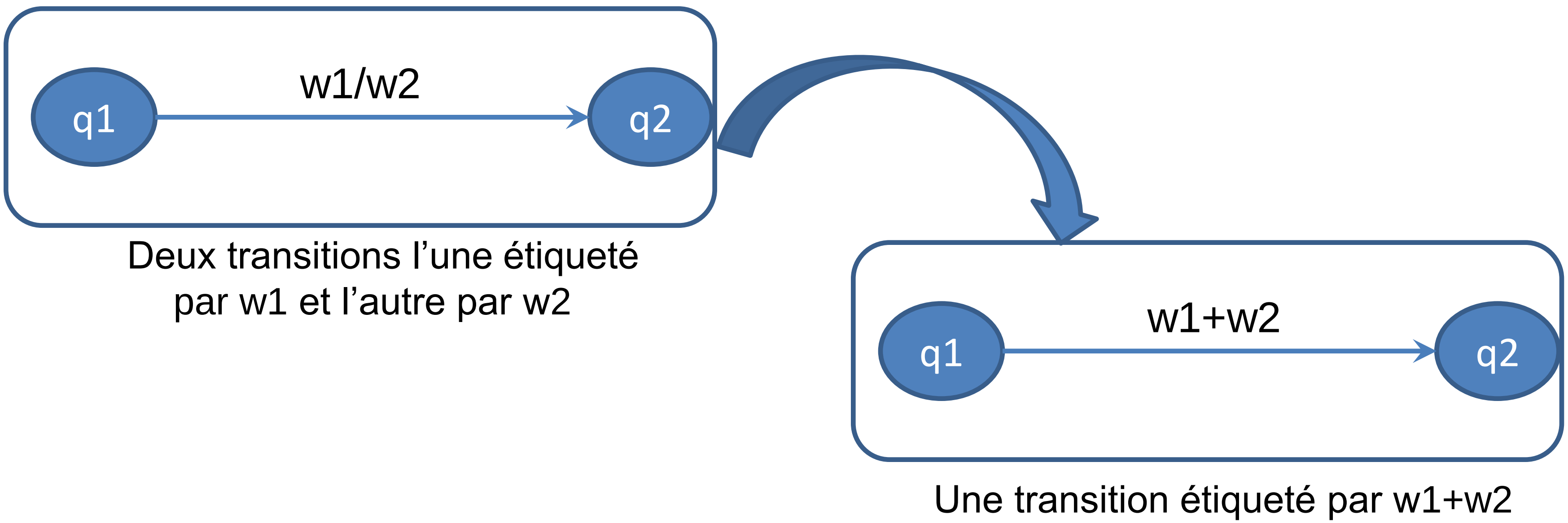
Etape préliminaire

1. Ajouter à l'automate A deux nouveaux états, p_0 état initial et p_F état final
2. Relier d'une part p_0 à l'état initial de A par ε
3. Relier tous les états finaux de A à p_F par ε

Transformation d'un automate en ER par élimination d'états

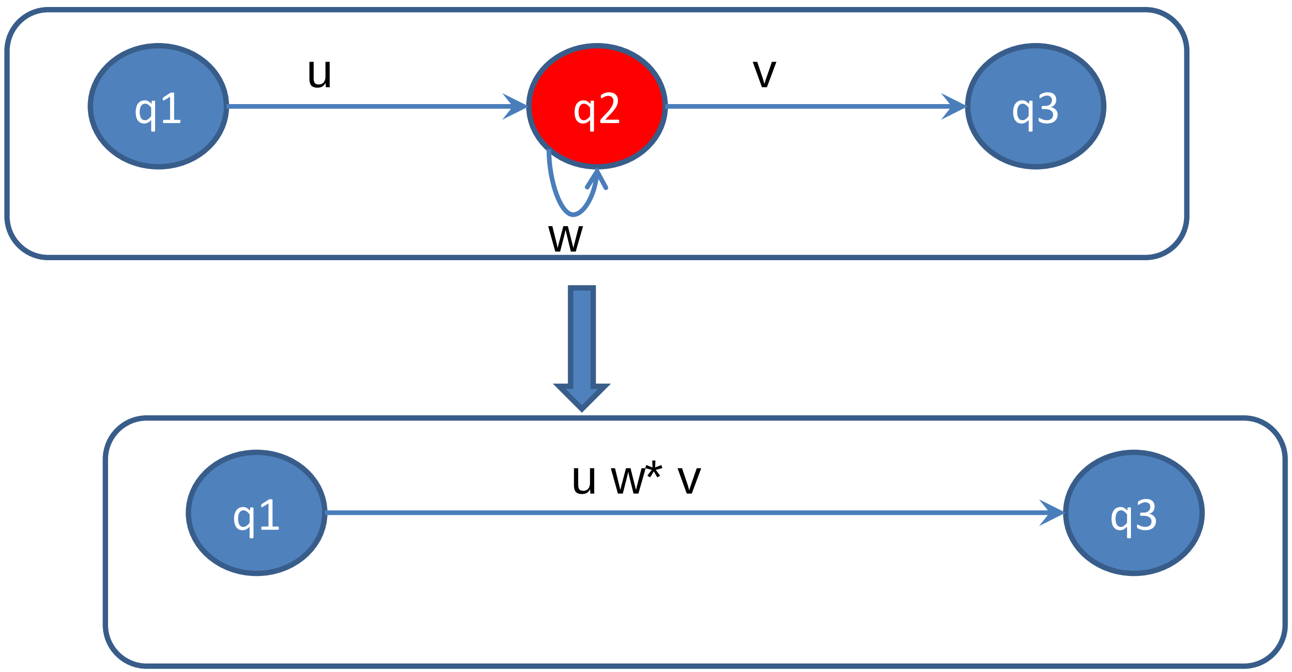
Répéter les réductions suivantes tant que possible :

1. Réduction 1 : s'il existe plusieurs transitions entre deux états, les remplacer par une transition unique



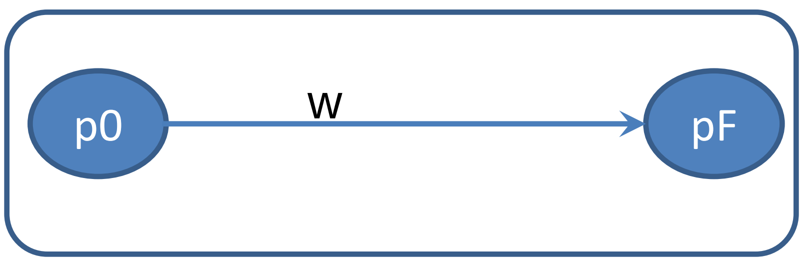
Transformation d’un automate en ER par élimination d’états

2. Réduction 2 : Suppression de l’état q2 et remplacer les transitions de la forme q1—u--> q2, q2 –w-->q2 et q2—v-->q3 par une seule transition reliant directement q1 à q3 et étiquetée par uw*v

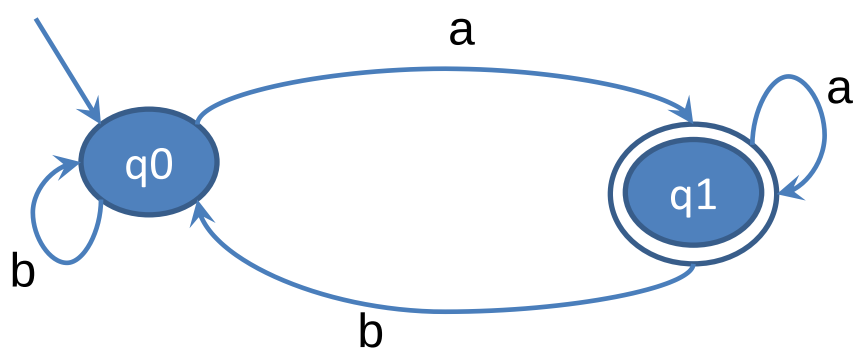


Transformation d'un automate en ER par élimination d'états

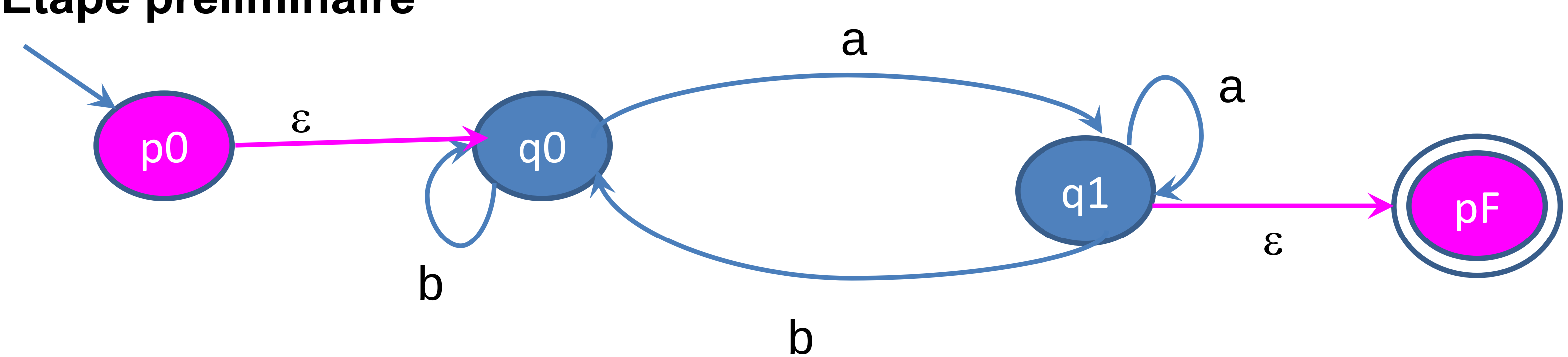
A la fin on obtient un automate avec seulement les deux états p0 et pF ainsi qu'une seule transition reliant ces deux états. L'étiquette de cette transition représente l'expression régulière dénotant le langage reconnu par l'automate A.



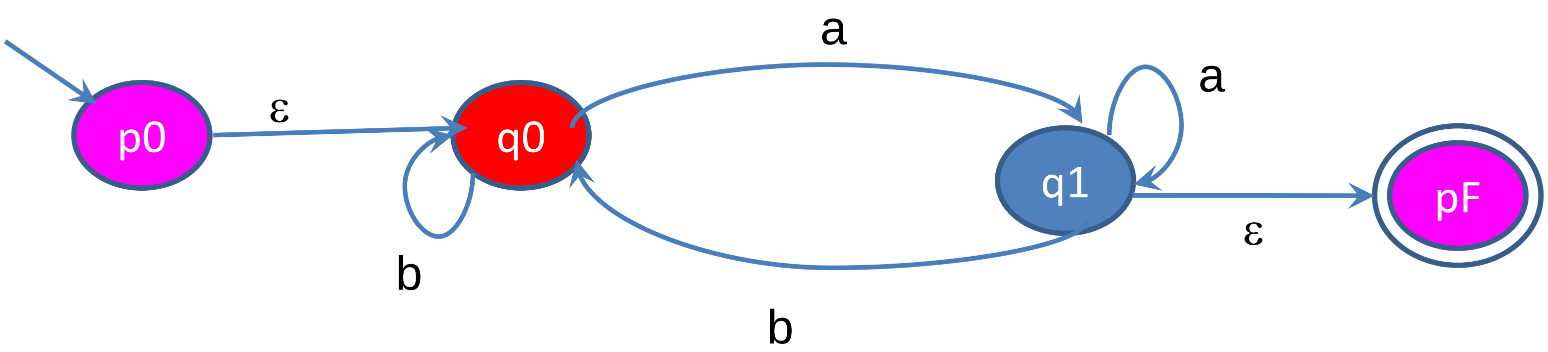
Transformation d'un automate en ER par élimination d'états



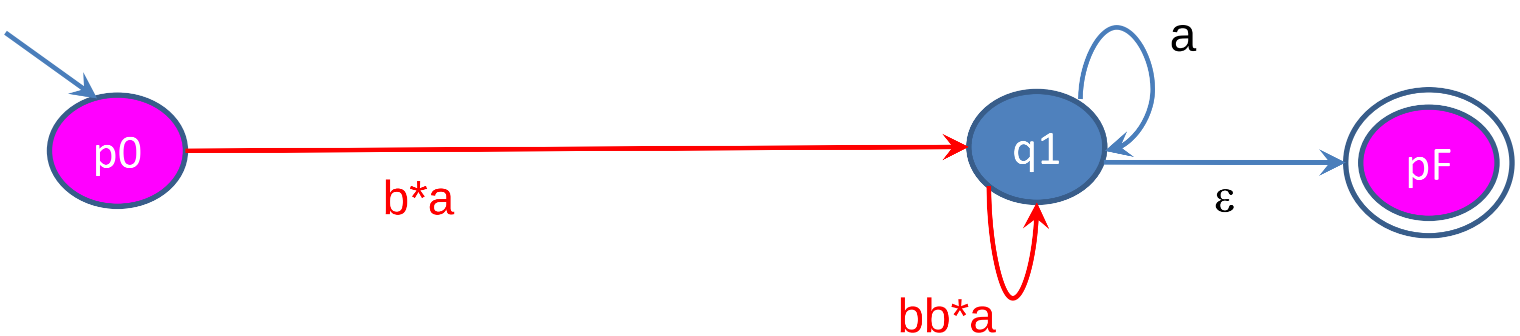
Etape préliminaire



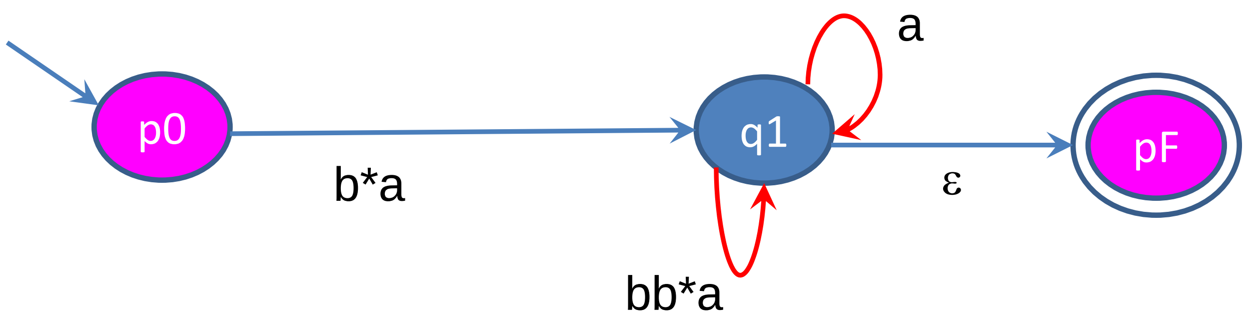
Transformation d'un automate en ER par élimination d'états



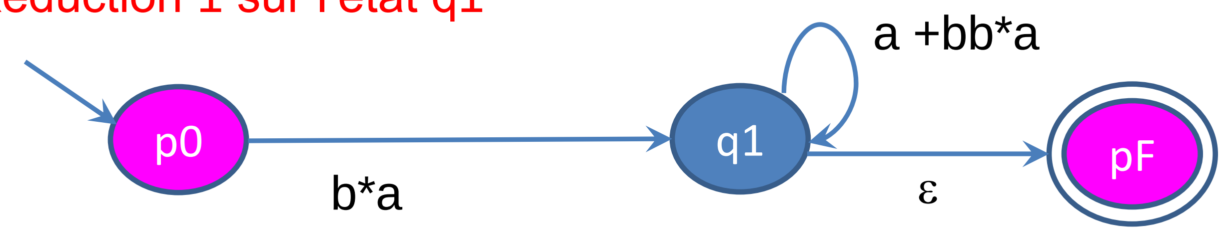
Réduction 2 : **Suppression de l'état q_0: 2 chemins p0-q0-q1 et q1-q0-q1**



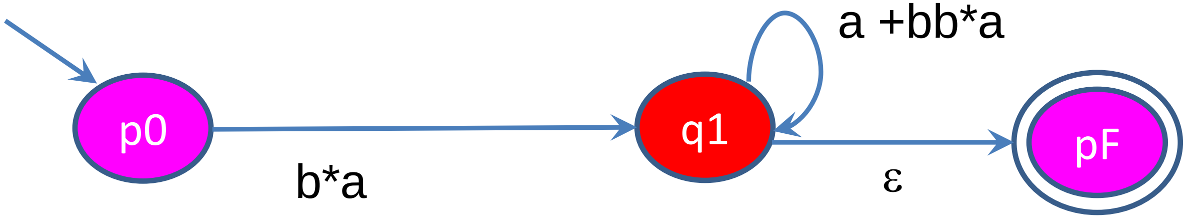
Transformation d’un automate en ER par élimination d’états



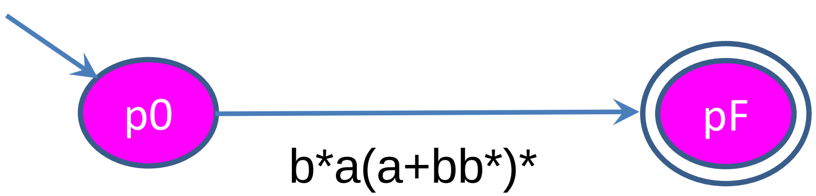
Réduction 1 sur l’état q1



Transformation d’un automate en ER par élimination d’états



Réduction 2 : Suppression de l’état q1 et on a un chemin p0-q1-pF



L’expression régulière associée à l’AEF est b*a(a+bb)*