Théorie des Langages

Intersection d’automates

Yves Kasparian

2025-2026

Enoncé du problème

- On suppose que l'on dispose de deux langages : L_1 et L_2.
- On suppose que l'on a un automate A_1 qui reconnait L_1 et un automate A_2 qui reconnait L_2.

Enoncé du problème

- On suppose que l'on dispose de deux langages : L_1 et L_2.
- On suppose que l'on a un automate A_1 qui reconnait L_1 et un automate A_2 qui reconnait L_2.

Comment à partir de A_1 et A_2 construire un automate A qui va reconnaitre le langage $L_1 \cap L_2$?

Construction de l’automate A

Construction de l'automate A

- Soient $q_1, \dots, q_n$ tous les états de A_1
- Soient $p_1, \dots, p_m$ tous les états de A_2

Construction de l'automate A

- Soient $q_1, \ldots, q_n$ tous les états de A_1
- Soient $p_1, \ldots, p_m$ tous les états de A_2

Si q_1 et p_1 sont les états initiaux de A_1 et A_2 alors :

Construction de l'automate A

- Soient $q_1, \dots, q_n$ tous les états de A_1
- Soient $p_1, \dots, p_m$ tous les états de A_2

Si q_1 et p_1 sont les états initiaux de A_1 et A_2 alors :

- On trace la table de transition de A en partant de l'état initial (q_1 , p_1).

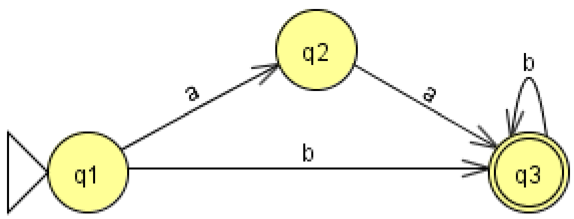
Construction de l'automate A

- Soient $q_1, \dots, q_n$ tous les états de A_1
- Soient $p_1, \dots, p_m$ tous les états de A_2

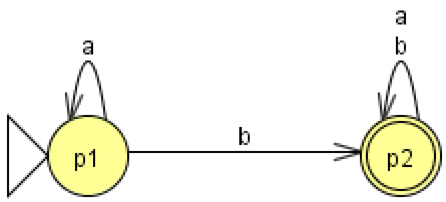
Si q_1 et p_1 sont les états initiaux de A_1 et A_2 alors :

- On trace la table de transition de A en partant de l'état initial (q_1 , p_1).
- On détermine toutes les transitions $(q_i \ , \ p_j)$ accessibles à partir de (q_1 , p_1).

Exemple



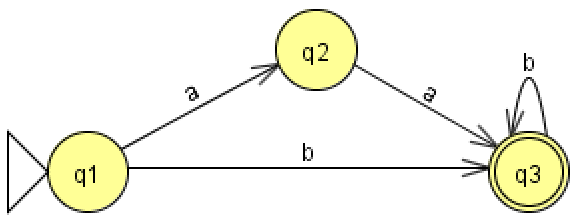
A_1



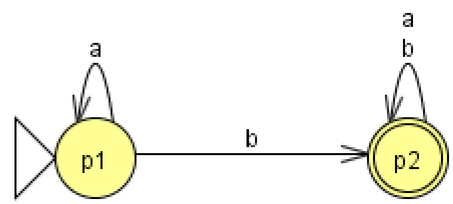
A_2

Question : Pouvez vous décrire les langages L_1 et L_2 reconnus par ces automates ?

Exemple



A_1

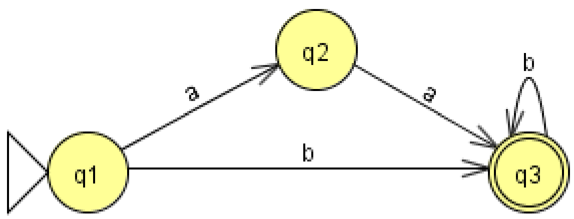


A_2

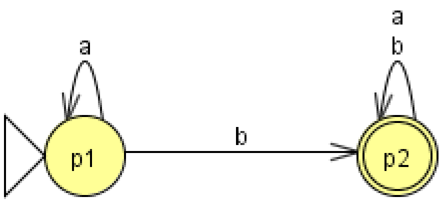
Question : Pouvez vous décrire les langages L_1 et L_2 reconnus par ces automates ?

Plus difficile : Pouvez vous déterminer les expressions régulières liées à ces automates ?

Exemple



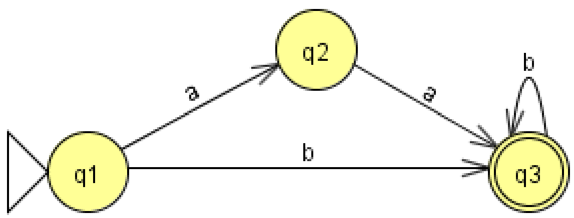
A_1



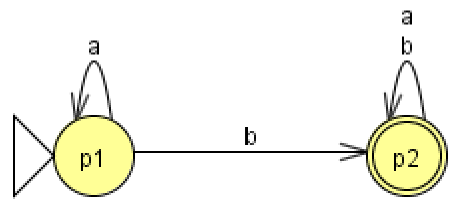
A_2

L_1 est le langage des mots qui :

Exemple



A_1

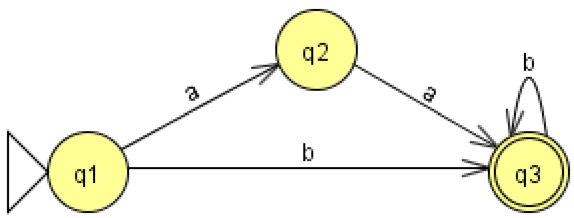


A_2

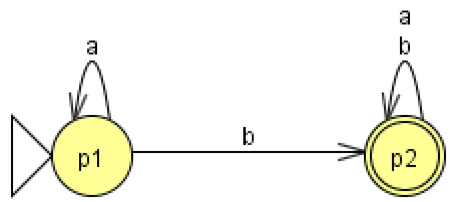
L_1 est le langage des mots qui :

- Contiennent 2 a puis autant de b que l'on veut (*possiblement aucun*).

Exemple



A_1



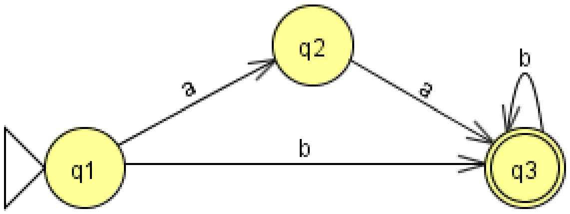
A_2

L_1 est le langage des mots qui :

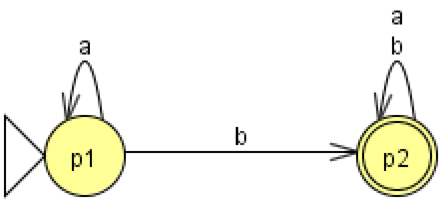
- Contiennent 2 a puis autant de b que l’on veut (*possiblement aucun*).

OU

Exemple



A_1



A_2

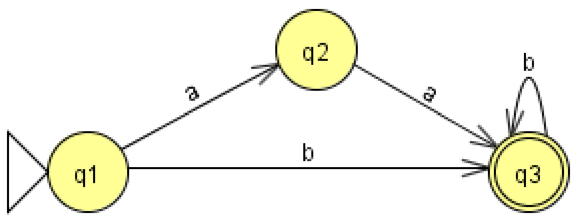
L_1 est le langage des mots qui :

- Contiennent 2 a puis autant de b que l'on veut (*possiblement aucun*).

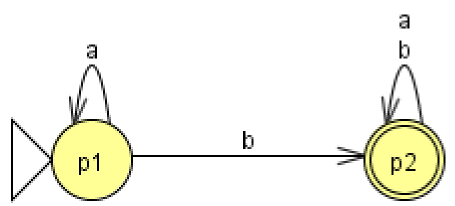
OU

- Ne contiennent aucun a et au moins un b.

Exemple



A_1



A_2

L_1 est le langage des mots qui :

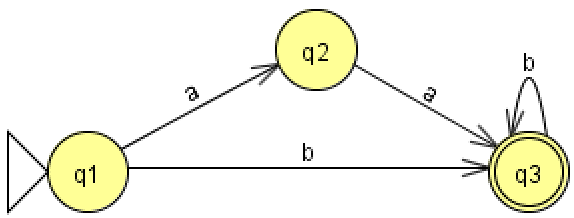
- Contiennent 2 a puis autant de b que l'on veut (*possiblement aucun*).

OU

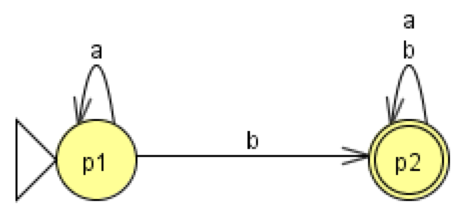
- Ne contiennent aucun a et au moins un b.

L_2 est le langage des mots qui contiennent au moins un b.

Exemple



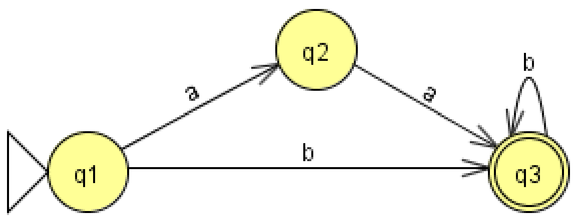
A_1



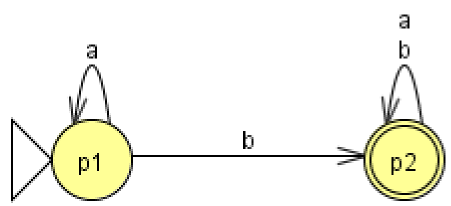
A_2

L'expression régulière pour L_1 est :

Exemple



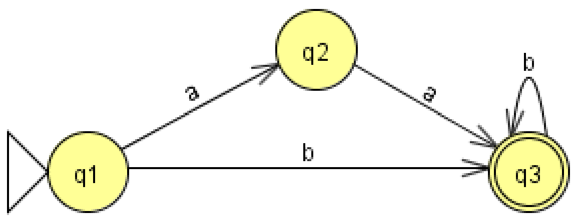
A_1



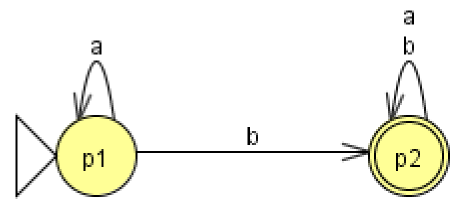
A_2

L'expression régulière pour L_1 est : aab* + bb*

Exemple



A_1

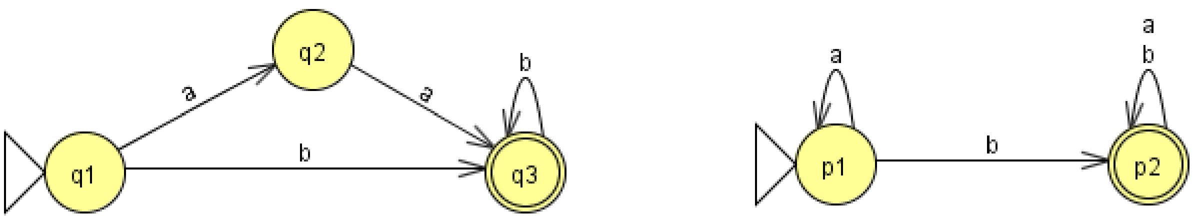


A_2

L'expression régulière pour L_1 est : aab* + bb*

Factorisée cela donne :

Exemple

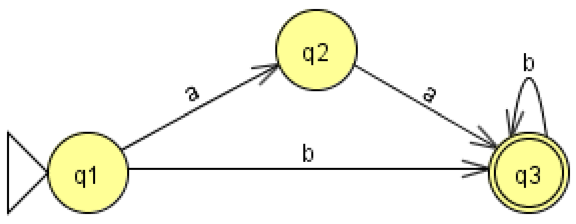


A_1 A_2

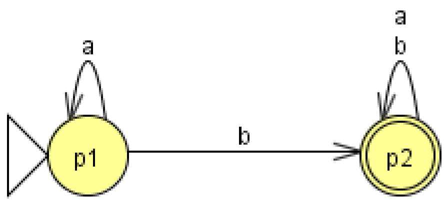
L’expression régulière pour L_1 est : aab* + bb*

Factorisée cela donne : (aa + b)b*

Exemple



A_1



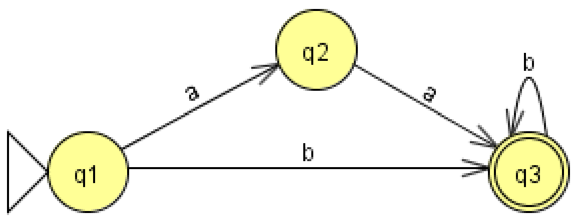
A_2

L'expression régulière pour L_1 est : aab* + bb*

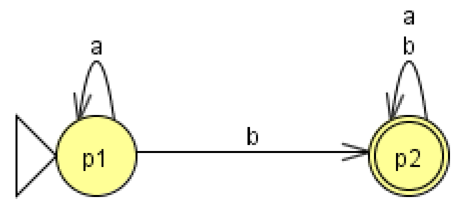
Factorisée cela donne : (aa + b)b*

L'expression régulière pour L_2 est :

Exemple



A_1



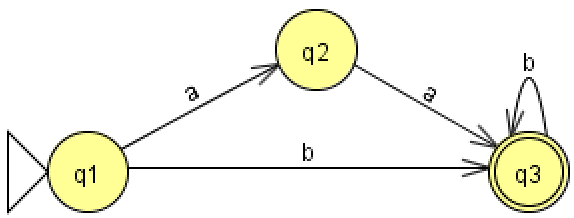
A_2

L'expression régulière pour L_1 est : aab* + bb*

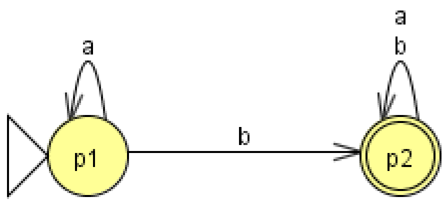
Factorisée cela donne : (aa + b)b*

L'expression régulière pour L_2 est : a* b (a + b)*

Exemple



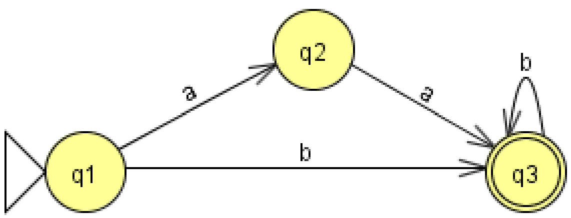
A_1



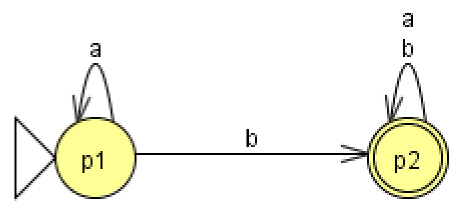
A_2

Est-ce que $L_1 \cap L_2$ est non vide ?

Exemple



A_1

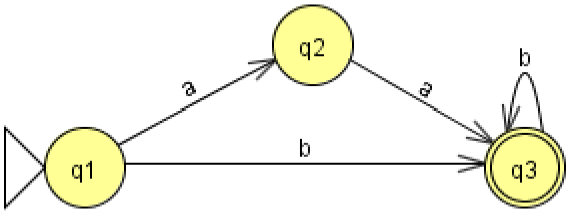


A_2

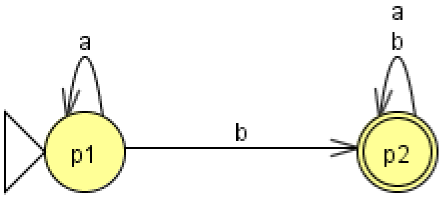
Est-ce que $L_1 \cap L_2$ est non vide ?

On constate que le mot b va être validé par les deux automates.

Exemple



A_1



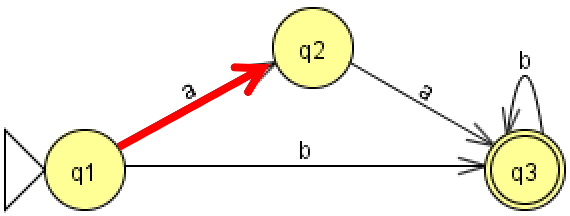
A_2

Est-ce que $L_1 \cap L_2$ est non vide ?

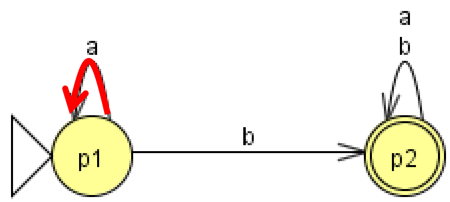
On constate que le mot b va être validé par les deux automates.

Donc $L_1 \cap L_2$ n'est pas vide.

Construction de l'automate



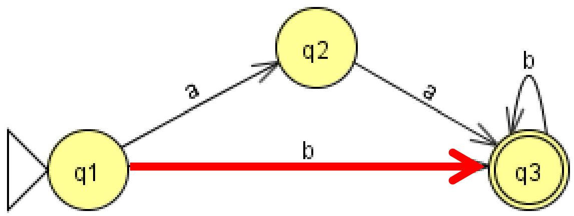
A_1



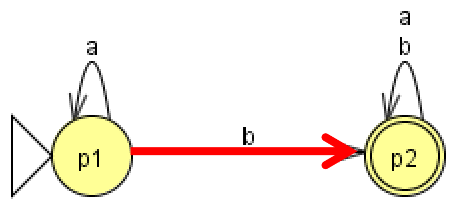
A_2

	a	b
→ (q1 , p1)	(q2 , p1)	

Construction de l’automate



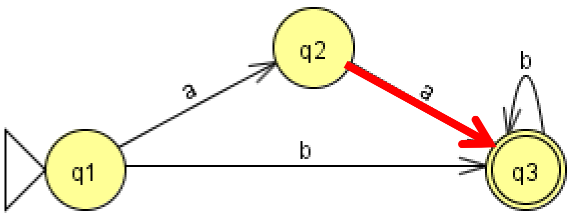
A_1



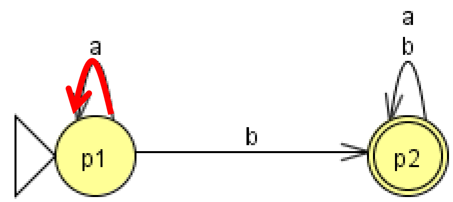
A_2

	a	b
→ (q1 , p1)	(q2 , p1)	(q3 , p2)

Construction de l'automate



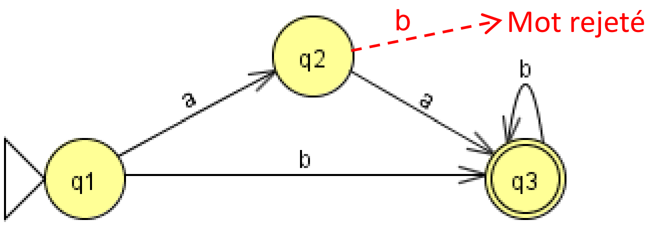
A_1



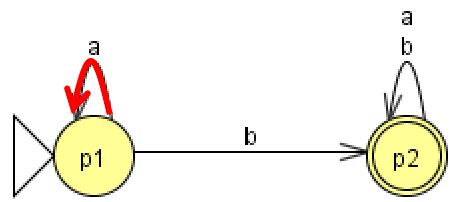
A_2

	a	b
→ (q1 , p1)	(q2 , p1)	(q3 , p2)
(q2 , p1)	(q3 , p1)	

Construction de l'automate



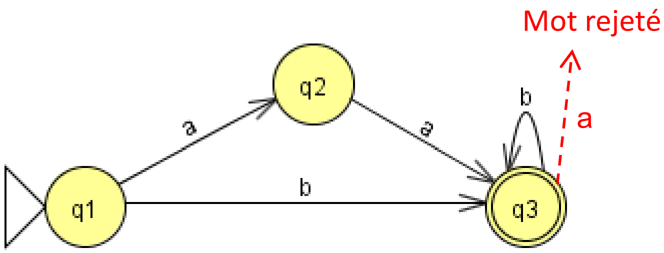
A_1



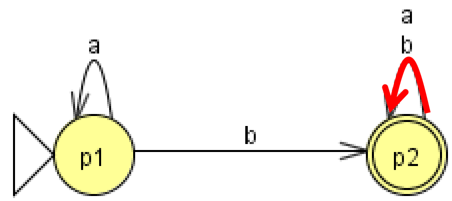
A_2

	a	b
→ (q1 , p1)	(q2 , p1)	(q3 , p2)
(q2 , p1)	(q3 , p1)	(Ø , p2)

Construction de l'automate



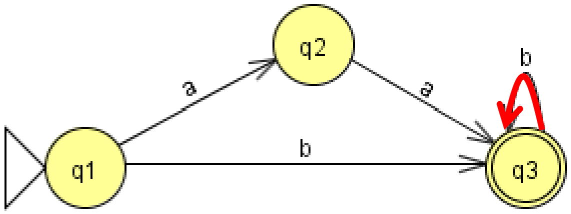
A_1



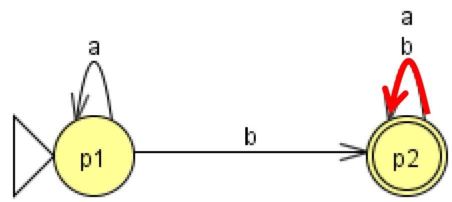
A_2

	a	b
→ (q1 , p1)	(q2 , p1)	(q3 , p2)
(q2 , p1)	(q3 , p1)	(Ø , p2)
(q3 , p2)	(Ø , p2)	

Construction de l'automate



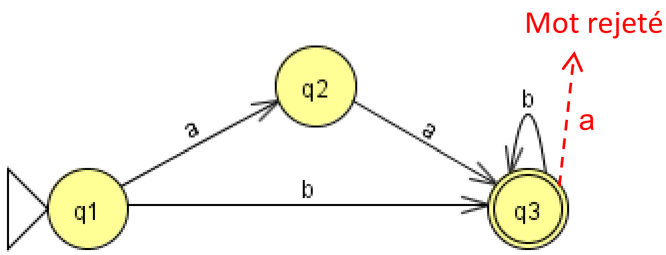
A_1



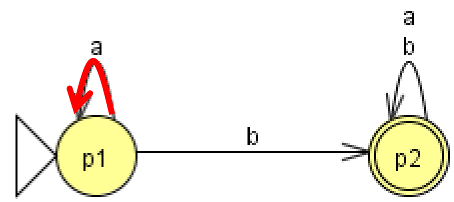
A_2

	a	b
→ (q1 , p1)	(q2 , p1)	(q3 , p2)
(q2 , p1)	(q3 , p1)	(Ø , p2)
(q3 , p2)	(Ø , p2)	(q3 , p2)

Construction de l’automate



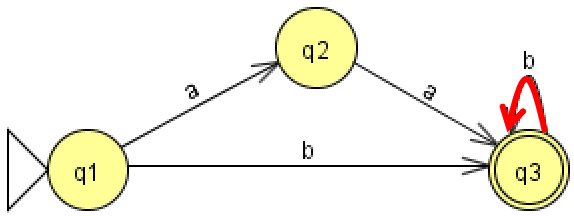
A_1



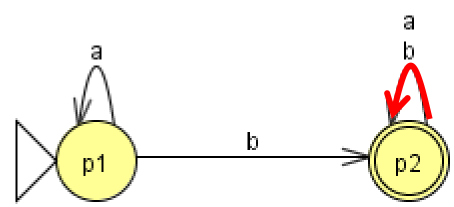
A_2

	a	b
→ (q1 , p1)	(q2 , p1)	($\underline{q3}$, $\underline{p2}$)
(q2 , p1)	($\underline{q3}$, p1)	(Ø , $\underline{p2}$)
($\underline{q3}$, $\underline{p2}$)	(Ø , $\underline{p2}$)	($\underline{q3}$, $\underline{p2}$)
($\underline{q3}$, p1)	(Ø , p1)	

Construction de l'automate



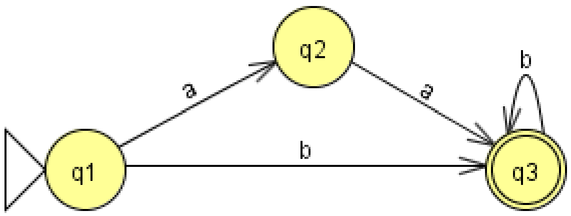
A_1



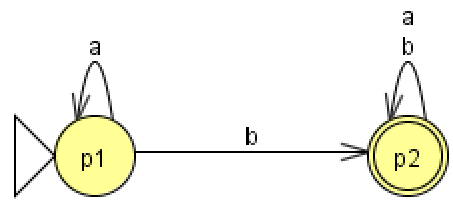
A_2

	a	b
→ (q1 , p1)	(q2 , p1)	(q3 , p2)
(q2 , p1)	(q3 , p1)	(Ø , p2)
(q3 , p2)	(Ø , p2)	(q3 , p2)
(q3 , p1)	(Ø , p1)	(q3 , p2)

Construction de l’automate



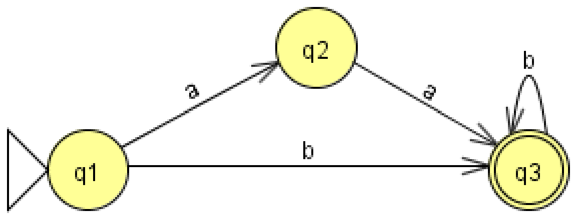
A_1



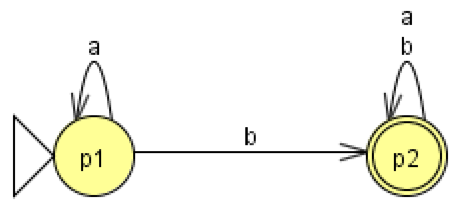
A_2

	a	b
→ (q1 , p1)	(q2 , p1)	(q3 , p2)
(q2 , p1)	(q3 , p1)	(Ø , p2)
(q3 , p2)	(Ø , p2)	(q3 , p2)
(q3 , p1)	(Ø , p1)	(q3 , p2)
En théorie il faut 2 x 3 = 6 lignes.		
Mais ici on a fait tous les états accessibles.		

Construction de l'automate



A_1

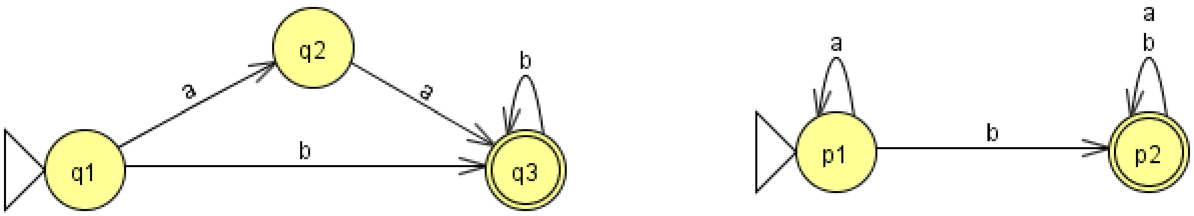


A_2

Etat initial

	a	b
→ (q1 , p1)	(q2 , p1)	(q3 , p2)
(q2 , p1)	(q3 , p1)	(Ø , p2)
(q3 , p2)	(Ø , p2)	(q3 , p2)
(q3 , p1)	(Ø , p1)	(q3 , p2)

Construction de l'automate



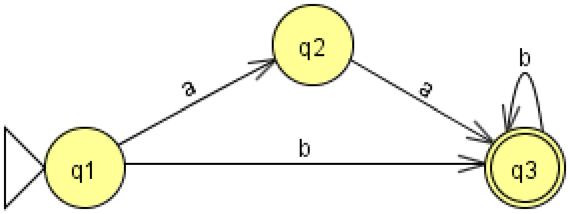
A_1 A_2

Etat initial

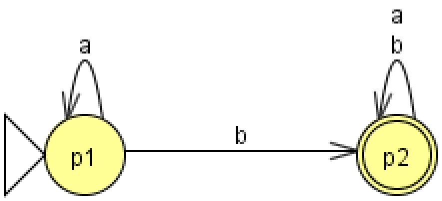
	a	b
→ (q1 , p1)	(q2 , p1)	(<u>q3</u> , <u>p2</u>)
(q2 , p1)	(<u>q3</u> , p1)	(Ø , <u>p2</u>)
(<u>q3</u> , <u>p2</u>)	(Ø , <u>p2</u>)	(<u>q3</u> , <u>p2</u>)
(<u>q3</u> , p1)	(Ø , p1)	(<u>q3</u> , <u>p2</u>)

Etat acceptant

Construction de l'automate



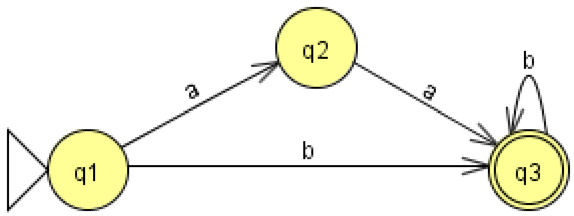
A_1



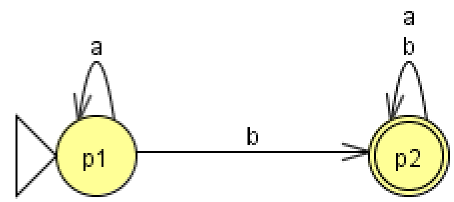
A_2

<u>Remarques :</u>

Construction de l’automate



A_1

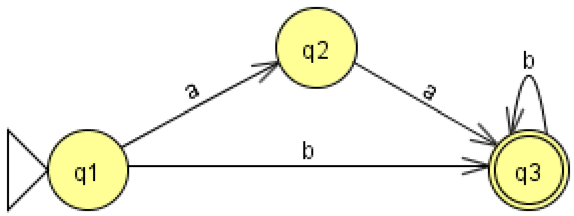


A_2

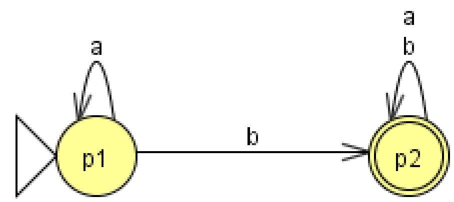
<u>Remarques :</u>

- Il n’y a que le couple (q1 , p1) qui est initial. C’est le couple qui est composé des états initiaux de A_1 et de A_2.

Construction de l’automate



A_1

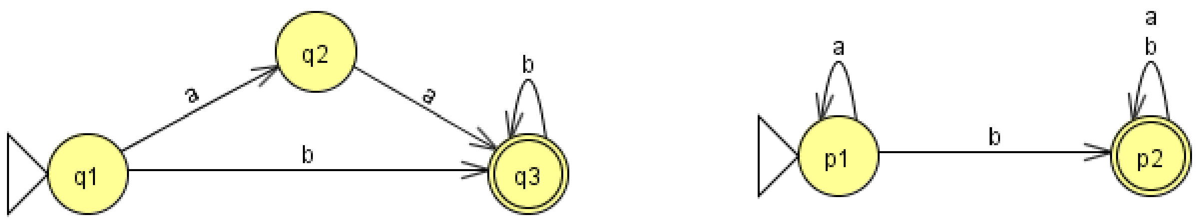


A_2

<u>Remarques :</u>

- Il n’y a que le couple (q1 , p1) qui est initial. C’est le couple qui est composé des états initiaux de A_1 et de A_2.
- Tous les couples (q_i , p_j) où q_i et p_j sont acceptants seront acceptants.

Construction de l'automate



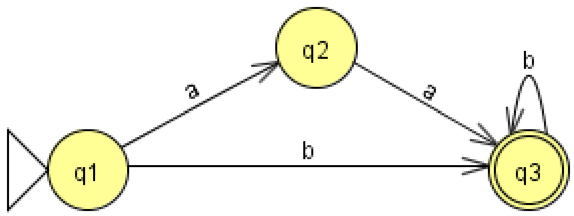
A_1 A_2

Remarques :

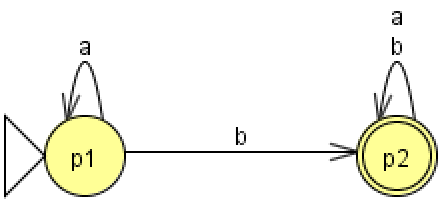
- Il n'y a que le couple (q1 , p1) qui est initial. C'est le couple qui est composé des états initiaux de A_1 et de A_2.
- Tous les couples (q_i , p_j) où q_i et p_j sont acceptants seront acceptants.

Ici il n'y a que (q3, p2)

Construction de l’automate



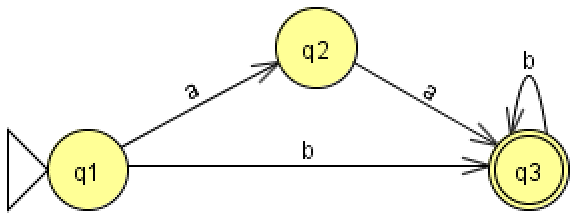
A_1



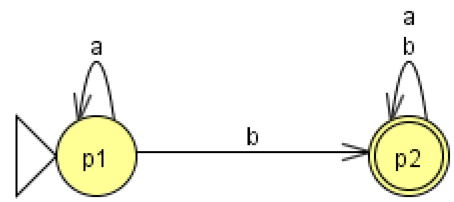
A_2

<u>Idée :</u>

Construction de l’automate



A_1

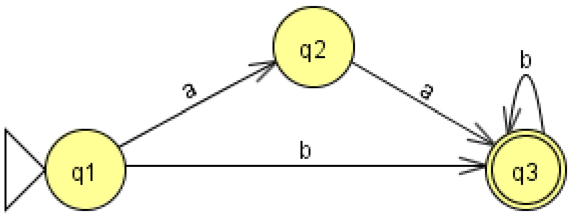


A_2

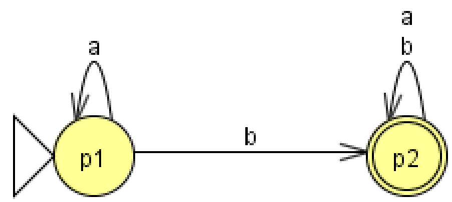
<u>Idée :</u>

On doit être accepté par les deux automates, donc il faut finir sur les états acceptants des deux automates (*pas juste un seul…*).

Construction de l’automate

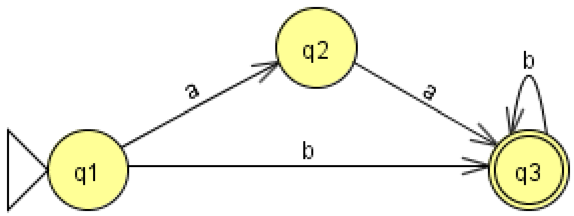


A_1

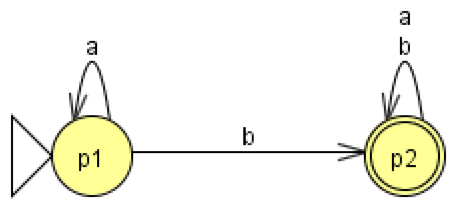


A_2

Construction de l'automate



A_1

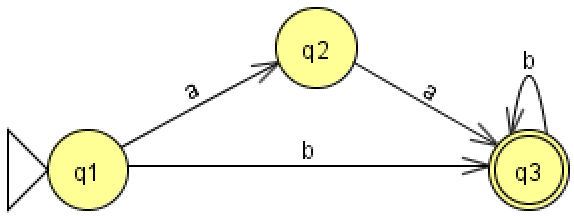


A_2

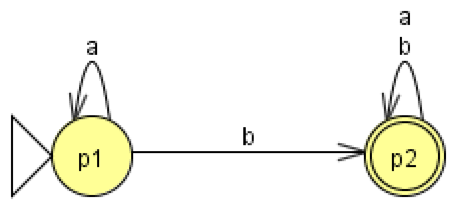
Table des transitions de l'automate :

	a	b
→ (q1 , p1)	(q2 , p1)	(q3 , p2)
(q2 , p1)	(q3 , p1)	(q3 , p2)
(q3 , p2)	(∅ , p2)	(q3 , p2)
(q3 , p1)	(∅ , p1)	(q3 , p2)

Construction de l’automate



A_1



A_2

Table des transitions de l’automate :

	a	b
→ (q1 , p1)	(q2 , p1)	(q3 , p2)
(q2 , p1)	(q3 , p1)	(q3 , p2)
(q3 , p2)	(Ø , p2)	(q3 , p2)
(q3 , p1)	(Ø , p1)	(q3 , p2)

On obtient l’automate :