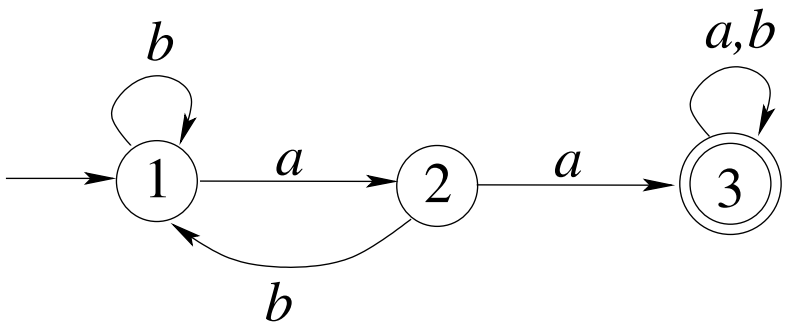
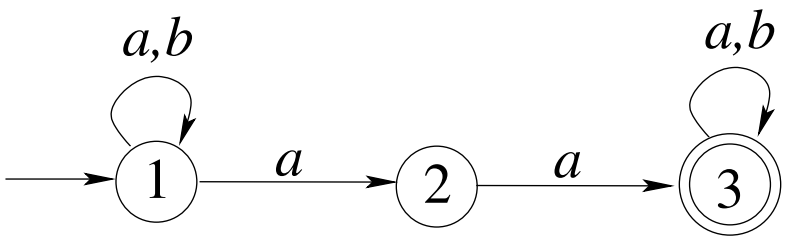
Non-déterminisme

Définition

Un automate est déterministe si pour tout état et toute lettre de l'alphabet, il y a au plus une transition étiquetée par cette lettre qui part de cet état.
Sinon, c'est un automate non-déterministe.

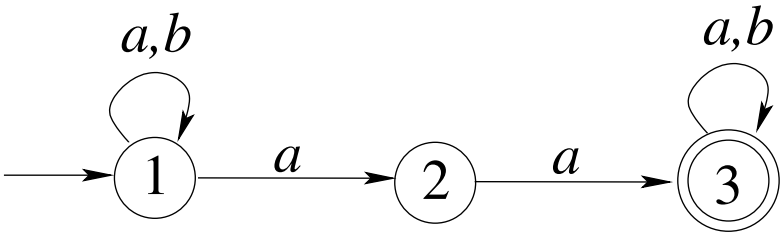


déterministe



non-déterministe

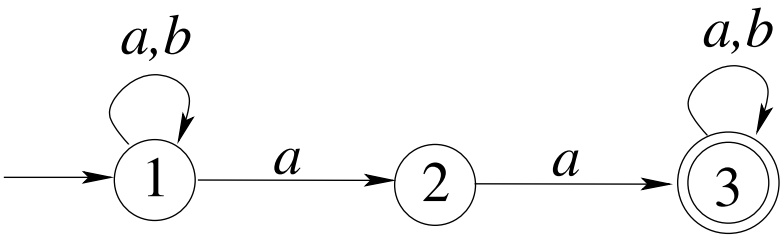
Table de transition d'un automate non-déterministe



	a	*b*
1	1, 2	1
2	3	
3	3	3

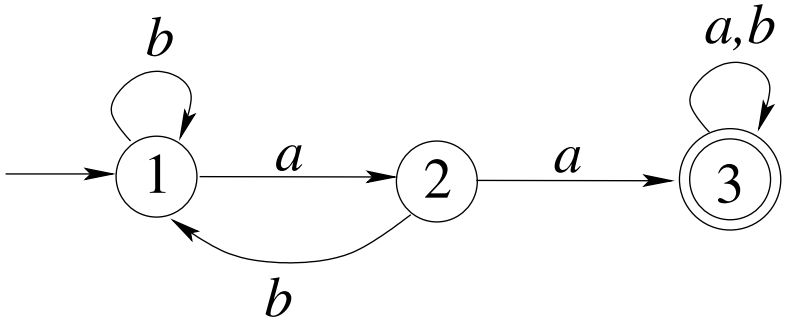
- À partir de l'état 1, en lisant la lettre *a*, on peut atteindre l'état 1 OU l'état 2

Exemple



Cet automate non-déterministe reconnaît le langage L = {mots contenant deux a consécutifs}

Cet automate déterministe reconnaît le même langage L



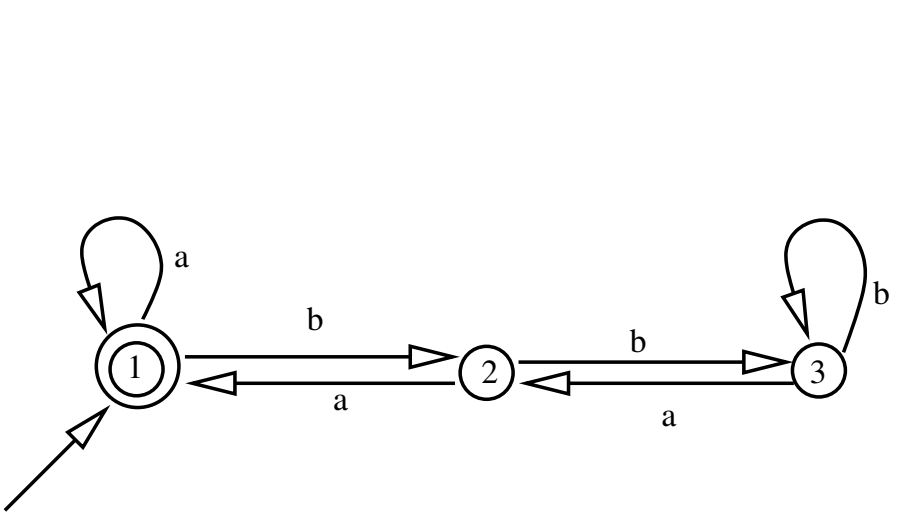
Une question

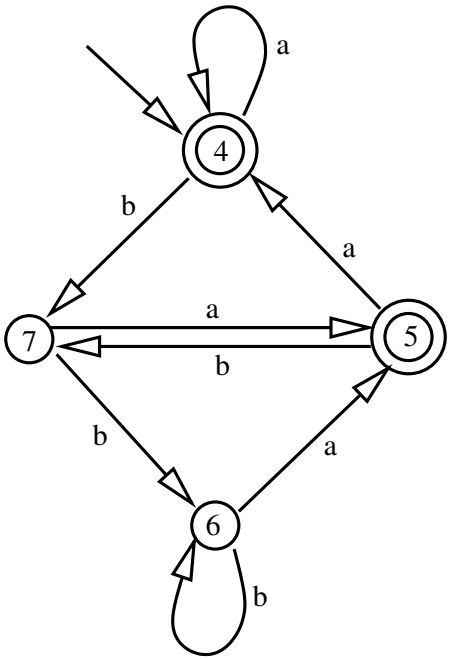
Est-ce que le non déterminisme est plus puissant ? Peut-on rendre un automate déterministe ?

Exemple

Question

Les automates $\mathcal{A}$ et $\mathcal{A}'$ acceptent-ils le même langage ?

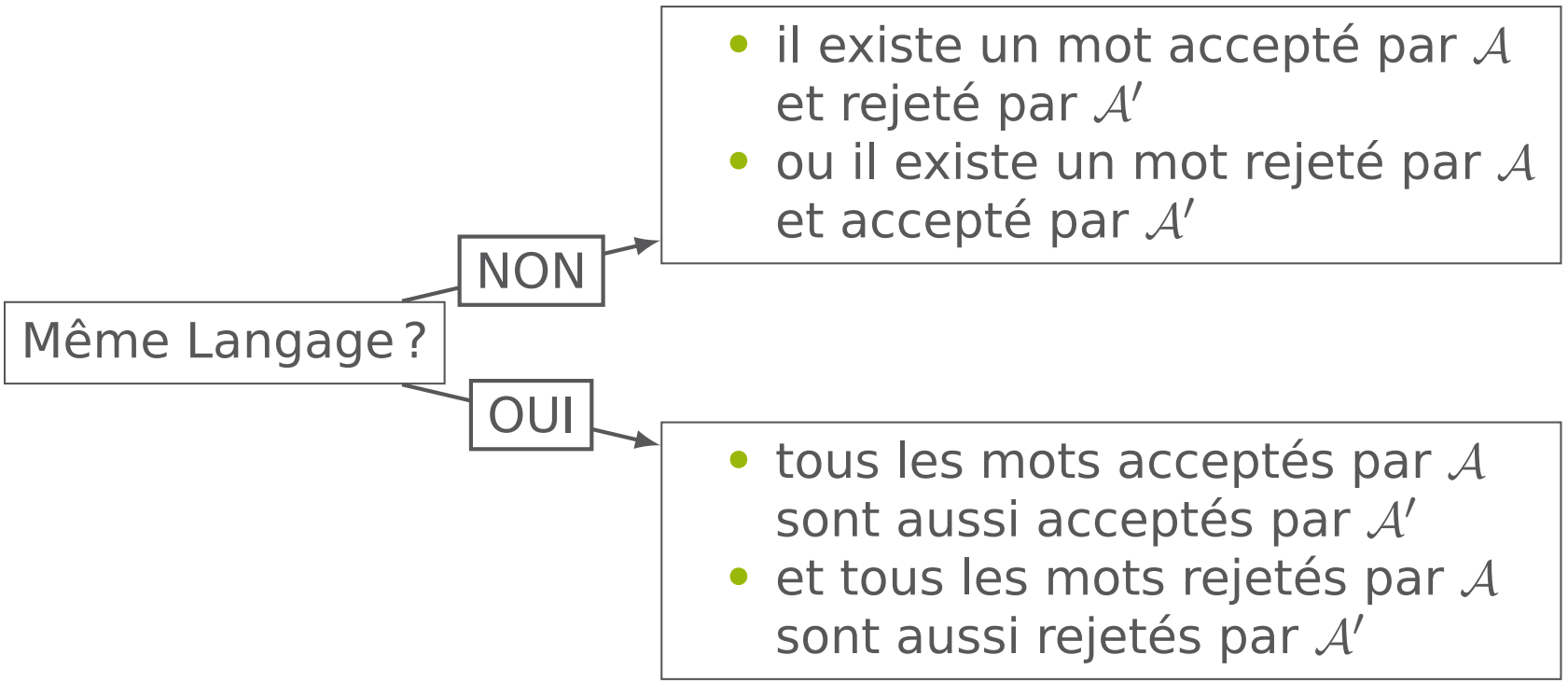




Exemple

Question

Les automates $\mathcal{A}$ et $\mathcal{A}'$ acceptent-ils le même langage ?



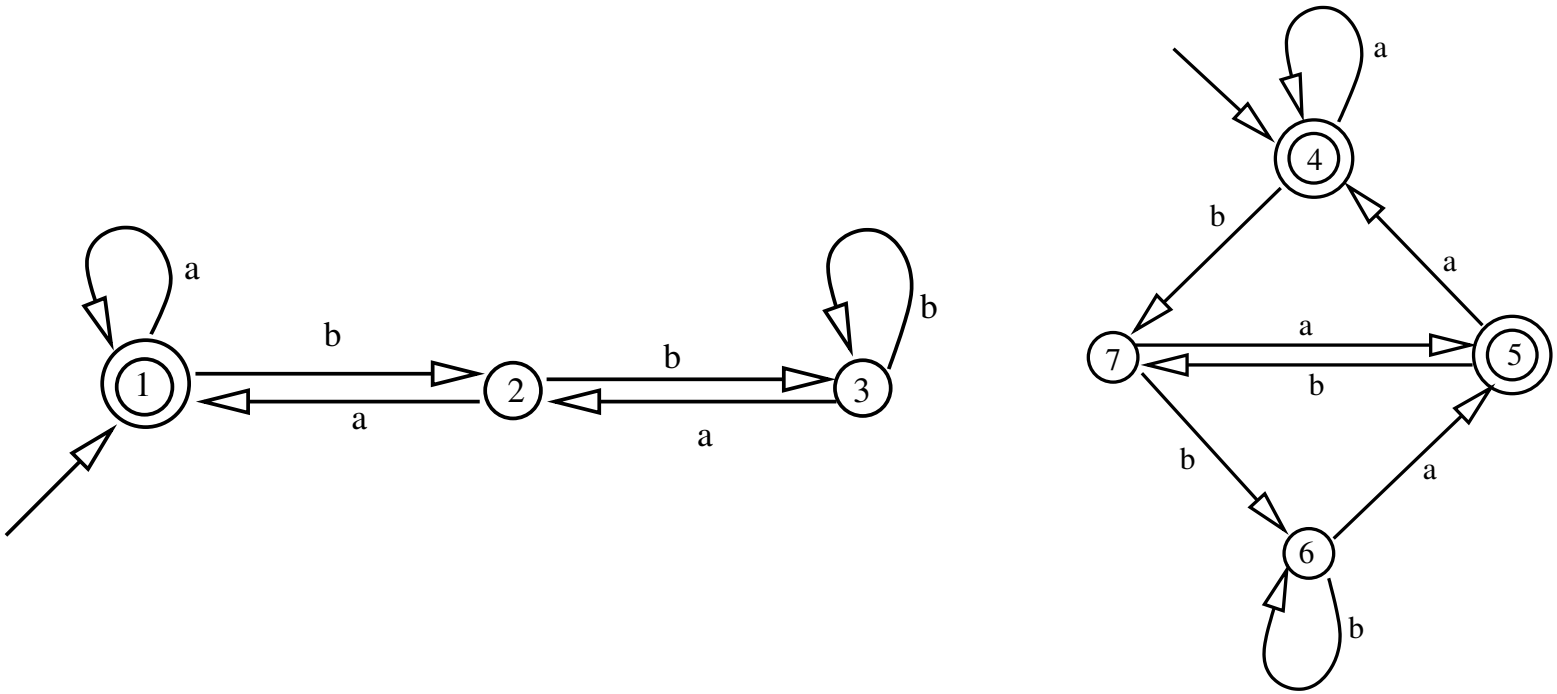
Idée générale

Définition

Deux automates qui acceptent le même langage s'appellent des automates équivalents

- Pour prouver que $\mathcal{A}$ et $\mathcal{A}'$ ne sont pas équivalents, il suffit de trouver un mot accepté par l'un et rejeté par l'autre
- Pour prouver que $\mathcal{A}$ et $\mathcal{A}'$ sont équivalents, on doit montrer qu'ils acceptent exactement les mêmes mots (ce qui paraît plus difficile)

Exemple (2)



Le mot *bba* est rejeté par $\mathcal{A}$ et accepté par $\mathcal{A}'$

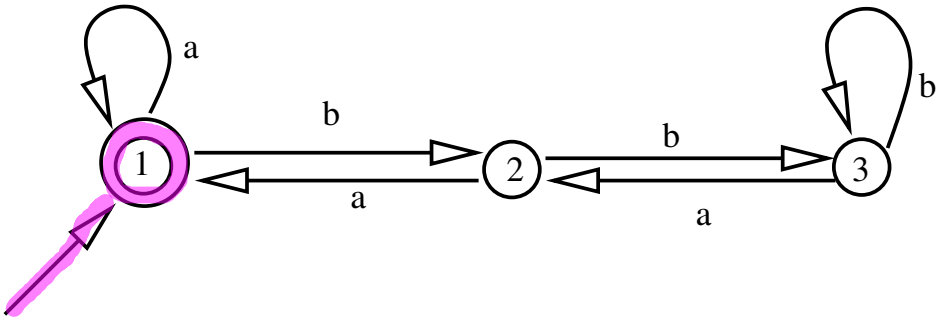
Conclusion : $\mathcal{A}$ et $\mathcal{A}'$ ne sont pas équivalents

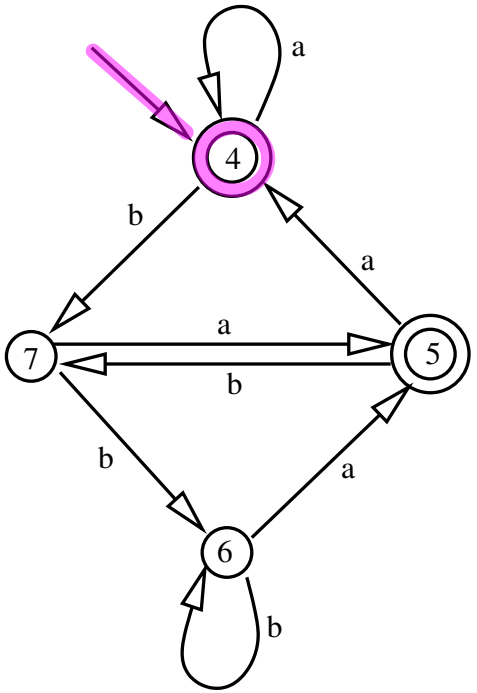
Méthode pour les automates déterministes

On suit des chemins « parallèles »dans les deux automates déterministes $\mathcal{A}$ et $\mathcal{A}'$ à partir de leur état initial respectif.

- Si on rencontre un couple d'états (q, q') avec q acceptant et q' non acceptant (ou le contraire), alors on a trouvé un mot w accepté par $\mathcal{A}$ et rejeté par $\mathcal{A}'$ (ou le contraire). Donc $\mathcal{A}$ et $\mathcal{A}'$ ne sont pas équivalents.
- Sinon, on répond qu'ils sont équivalents.

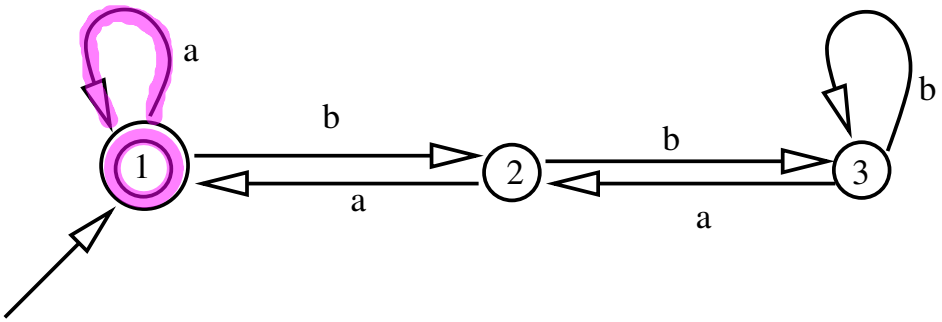
Exemple (3)

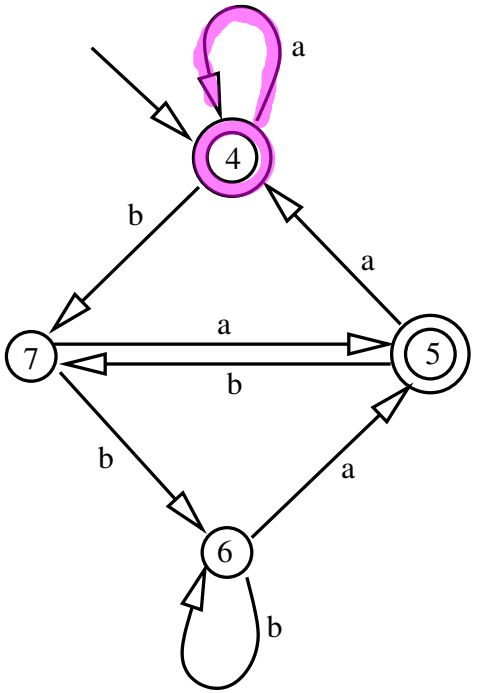




	a	b
$(\underline{1}, \underline{4})$		

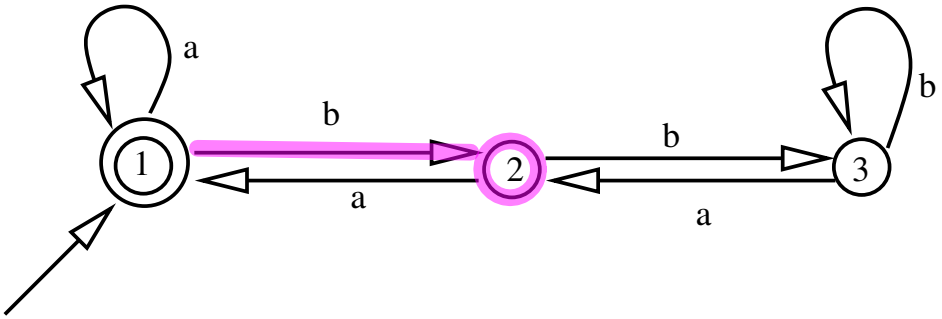
Exemple (3)

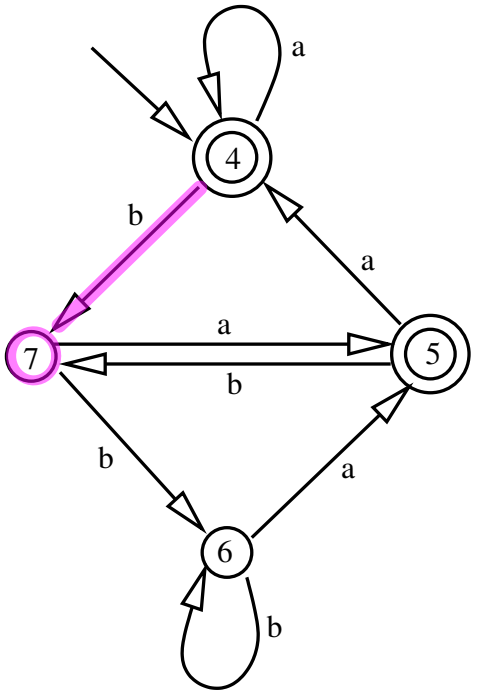




	a	b
$(\underline{1}, \underline{4})$	$(\underline{1}, \underline{4})$	

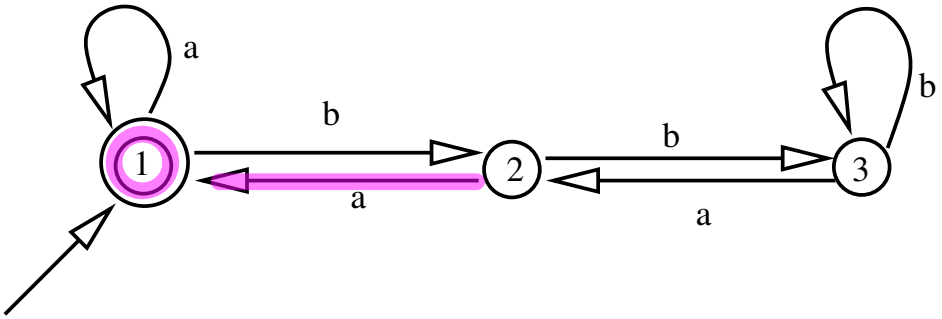
Exemple (3)

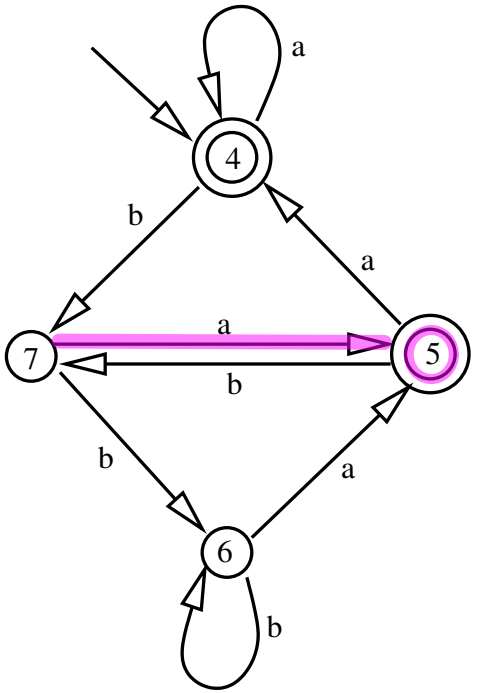




	a	b
$(\underline{1}, \underline{4})$	$(\underline{1}, \underline{4})$	$(2, 7)$

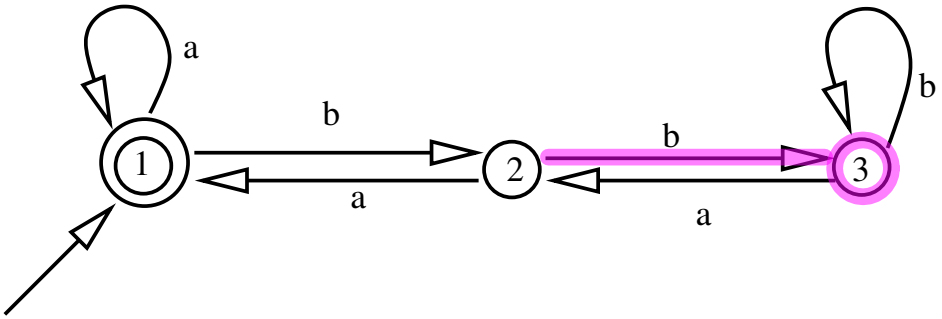
Exemple (3)

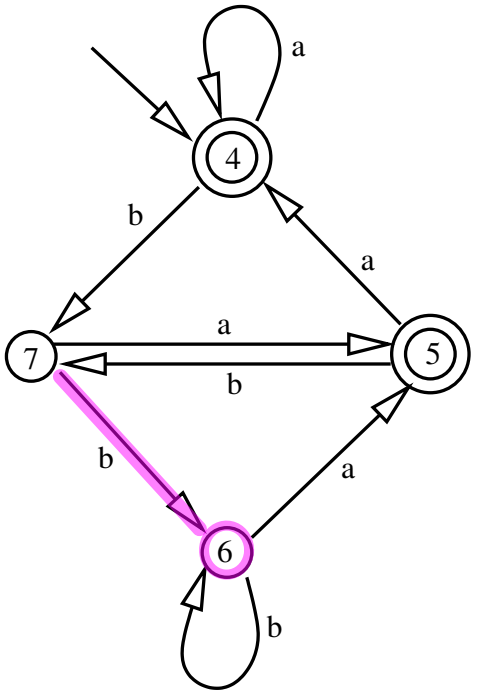




	a	b
$(\underline{1}, \underline{4})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(2, 7)$	$(\underline{1}, \underline{5})$	

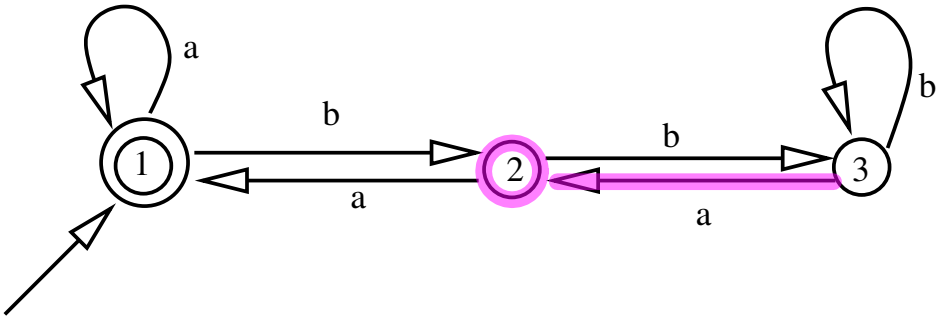
Exemple (3)

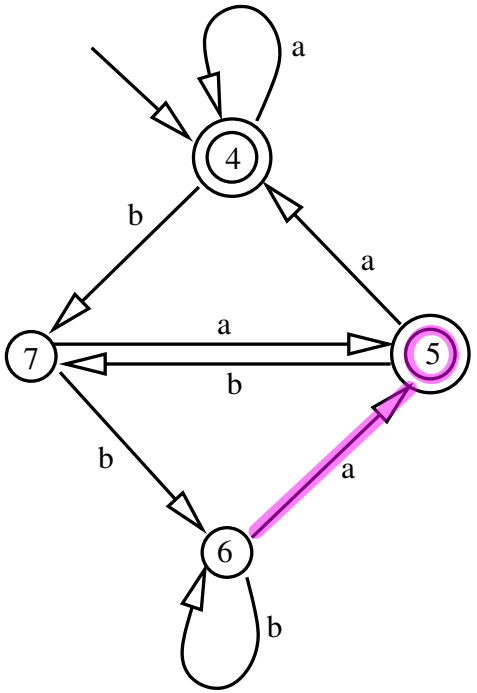




	a	b
$(\underline{1}, \underline{4})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(2, 7)$	$(\underline{1}, \underline{5})$	$(3, 6)$

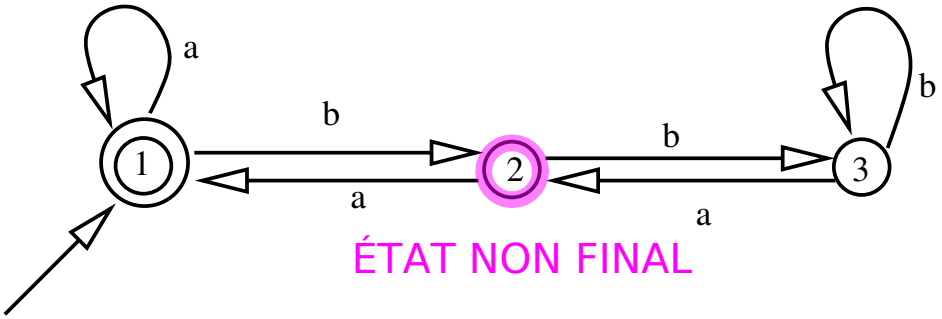
Exemple (3)

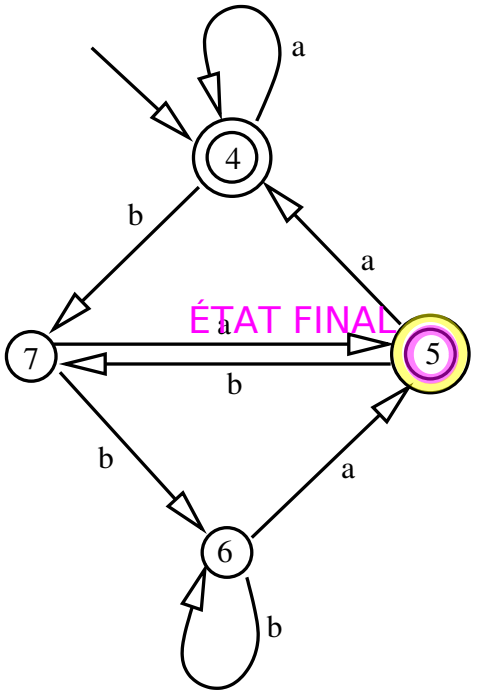




	a	b
$(\underline{1}, \underline{4})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(2, 7)$	$(\underline{1}, \underline{5})$	$(3, 6)$
$(\underline{1}, \underline{5})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(3, 6)$	$(2, \underline{5})$	...

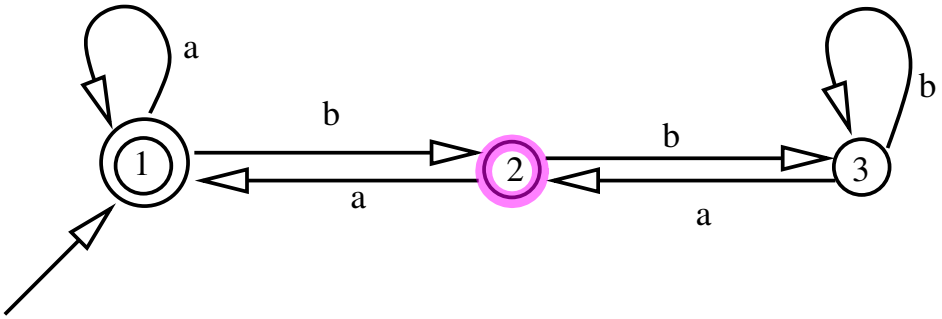
Exemple (3)

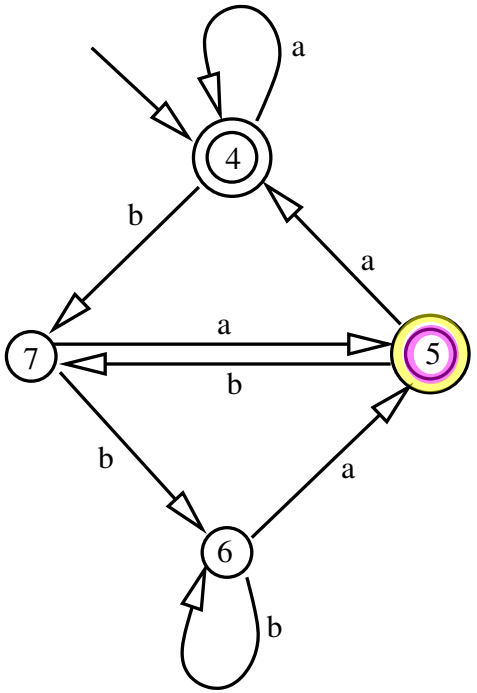




	a	*b*
$(\underline{1}, \underline{4})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(2, 7)$	$(\underline{1}, \underline{5})$	$(3, 6)$
$(\underline{1}, \underline{5})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(3, 6)$	$(2, \underline{5})$	…

Exemple (3)

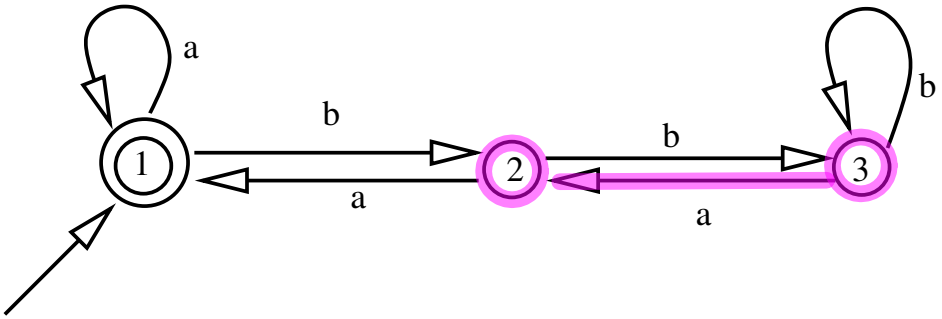


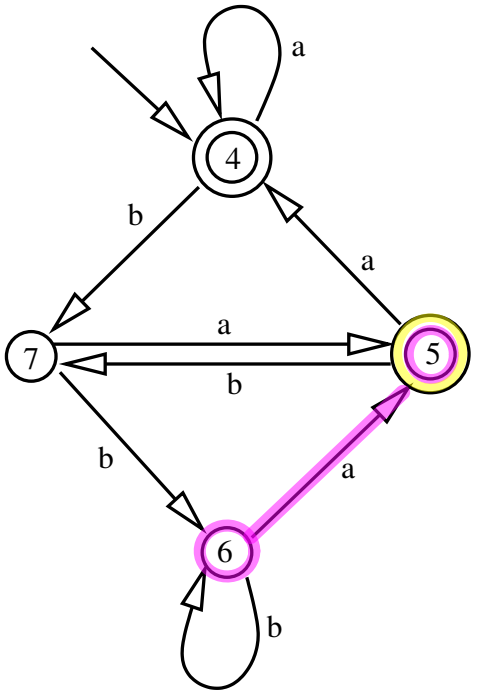


	a	b
$(\underline{1}, \underline{4})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(2, 7)$	$(\underline{1}, \underline{5})$	$(3, 6)$
$(\underline{1}, \underline{5})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(3, 6)$	$(2, \underline{5})$	…

En remontant le calcul, on s'aperçoit que le mot bba est rejeté par $\mathcal{A}$ et accepté par $\mathcal{A}'$

Exemple (3)

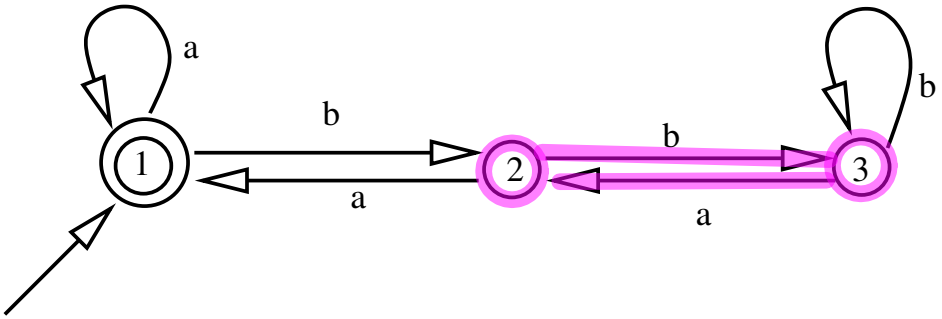


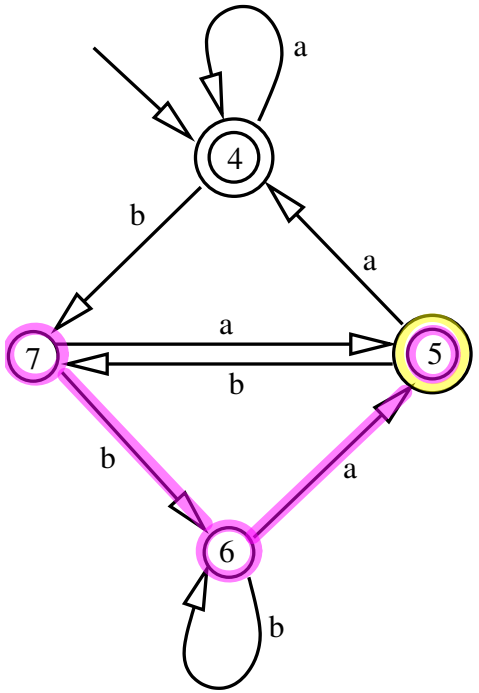


	a	b
$(\underline{1}, \underline{4})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(2, 7)$	$(\underline{1}, \underline{5})$	$(3, 6)$
$(\underline{1}, \underline{5})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(3, 6)$	$(2, \underline{5})$	…

En remontant le calcul, on s'aperçoit que le mot bb**a** est rejeté par $\mathcal{A}$ et accepté par $\mathcal{A}'$

Exemple (3)

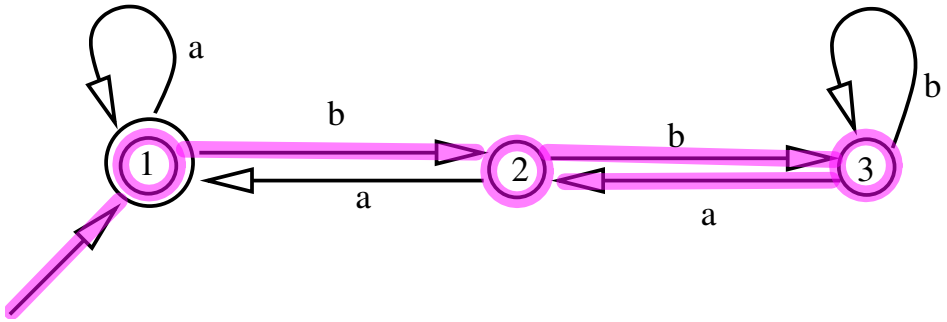


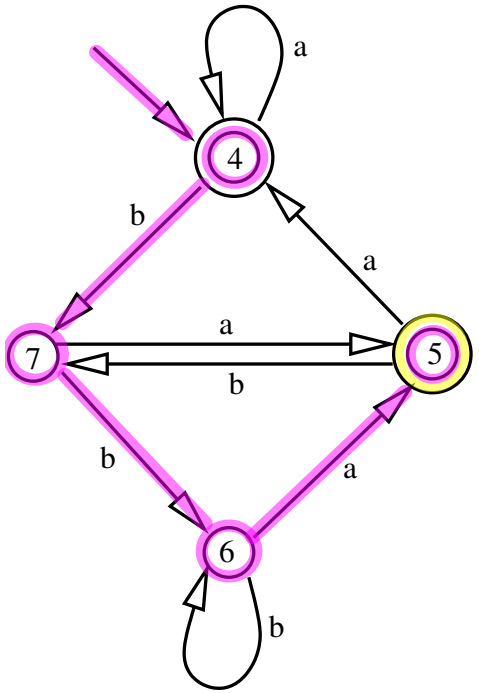


	a	b
$(\underline{1}, \underline{4})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(2, 7)$	$(\underline{1}, \underline{5})$	$(3, 6)$
$(\underline{1}, \underline{5})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(3, 6)$	$(2, \underline{5})$	…

En remontant le calcul, on s'aperçoit que le mot b**b**a est rejeté par $\mathcal{A}$ et accepté par $\mathcal{A}'$

Exemple (3)

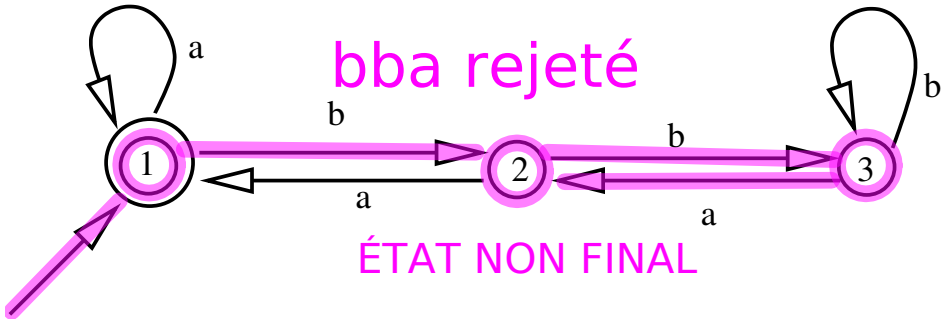


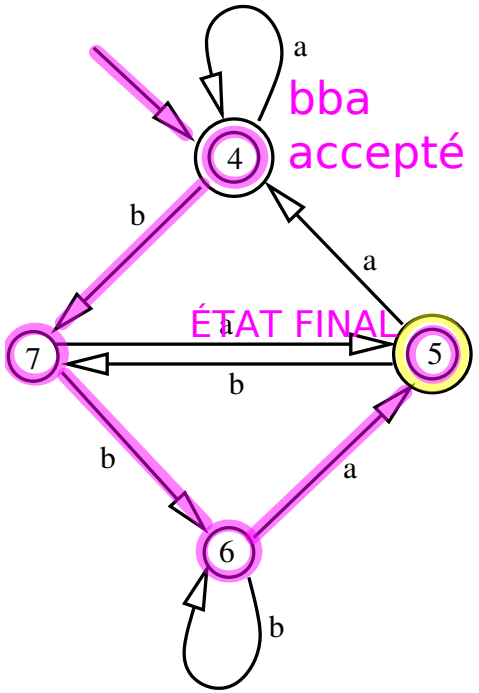


	a	b
$(\underline{1}, \underline{4})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(2, 7)$	$(\underline{1}, \underline{5})$	$(3, 6)$
$(\underline{1}, \underline{5})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(3, 6)$	$(2, \underline{5})$	…

En remontant le calcul, on s'aperçoit que le mot **b**ba est rejeté par $\mathcal{A}$ et accepté par $\mathcal{A}'$

Exemple (3)





	a	b
$(\underline{1}, \underline{4})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(2, 7)$	$(\underline{1}, \underline{5})$	$(3, 6)$
$(\underline{1}, \underline{5})$	$(\underline{1}, \underline{4})$	$(2, 7)$
$(3, 6)$	$(2, \underline{5})$	…

En remontant le calcul, on s'aperçoit que le mot bba est rejeté par $\mathcal{A}$ et accepté par $\mathcal{A}'$

Pourquoi cette méthode fonctionne ?

1. La méthode termine
2. La méthode donne bien la bonne réponse
   - quand la réponse est NON.
   - quand la réponse est OUI.

car il n'y a qu'un nombre fini de couples d'états (q, q'), donc le calcul se termine au bout d'un temps fini.

Pourquoi cette méthode fonctionne ?

1. La méthode termine
2. La méthode donne bien la bonne réponse
   - quand la réponse est NON.
   - quand la réponse est OUI.

Pourquoi cette méthode fonctionne ?

1. La méthode termine
2. La méthode donne bien la bonne réponse
   - quand la réponse est NON.
   - quand la réponse est OUI.

Si la méthode trouve un mot w accepté par $\mathcal{A}$ et rejeté par $\mathcal{A}'$ (ou le contraire), alors les deux automates $\mathcal{A}$ et $\mathcal{A}'$ sont bien non-équivalents.

Pourquoi cette méthode fonctionne ?

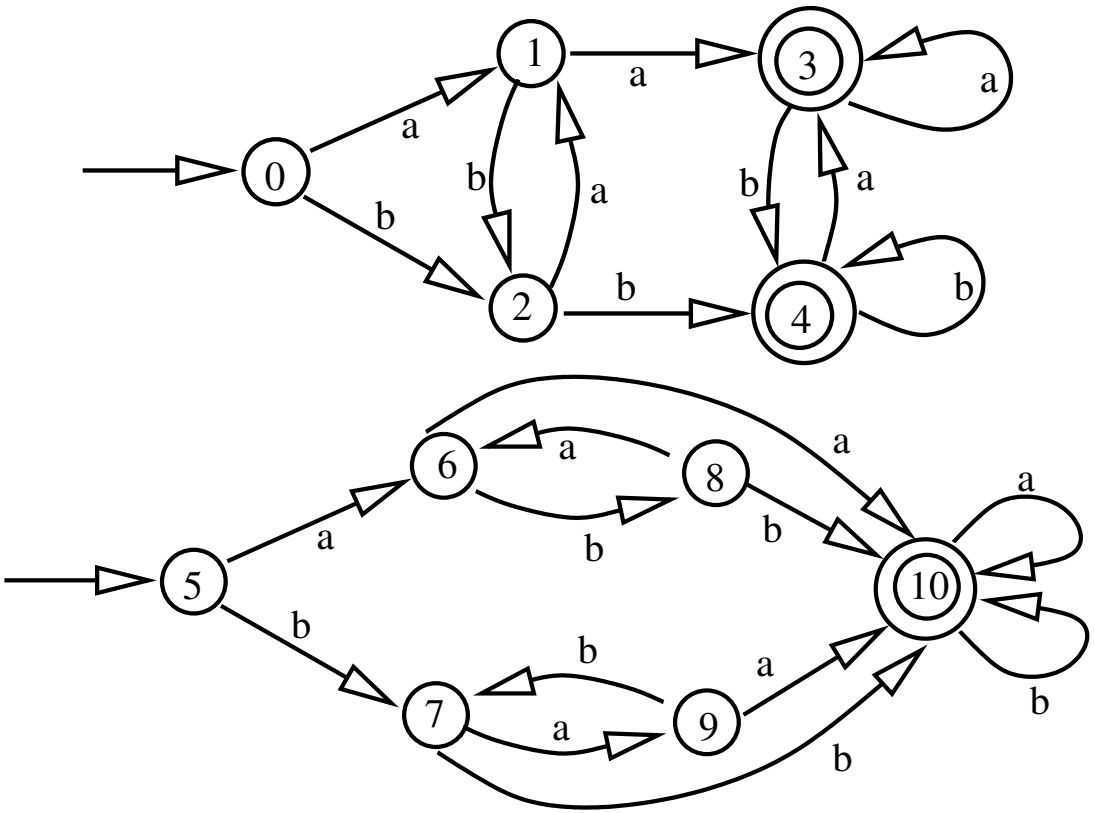
1. La méthode termine
2. La méthode donne bien la bonne réponse
   - quand la réponse est NON.
   - quand la réponse est OUI.

Sinon, tout chemin d'exécution dans l'automate $\mathcal{A}$ montrant qu'un mot w est accepté/rejeté par $\mathcal{A}$ correspond à un chemin d'exécution pour l'automate $\mathcal{A}'$ donnant le même résultat (on trouve ce chemin dans la table de transition de l'automate « bi-processeur »que nous avons construit)

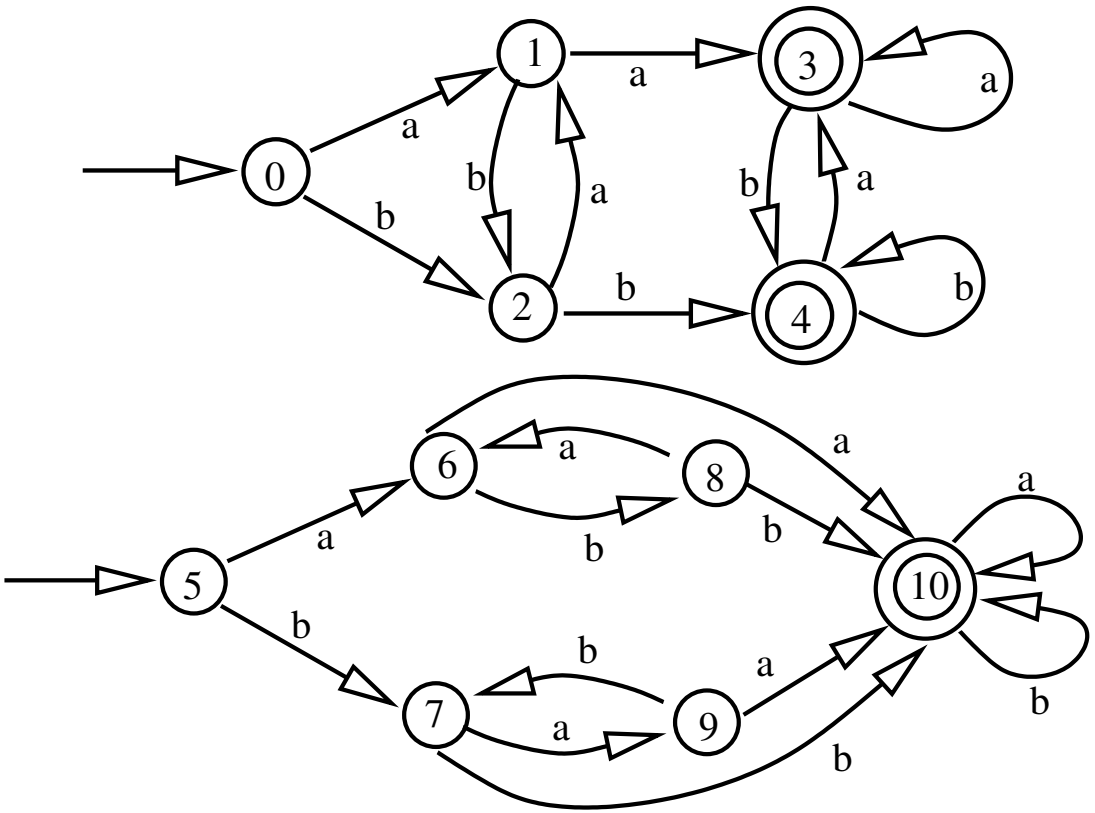
Pourquoi cette méthode fonctionne ?

1. La méthode termine
2. La méthode donne bien la bonne réponse
   - quand la réponse est NON.
   - quand la réponse est OUI.

Autre exemple

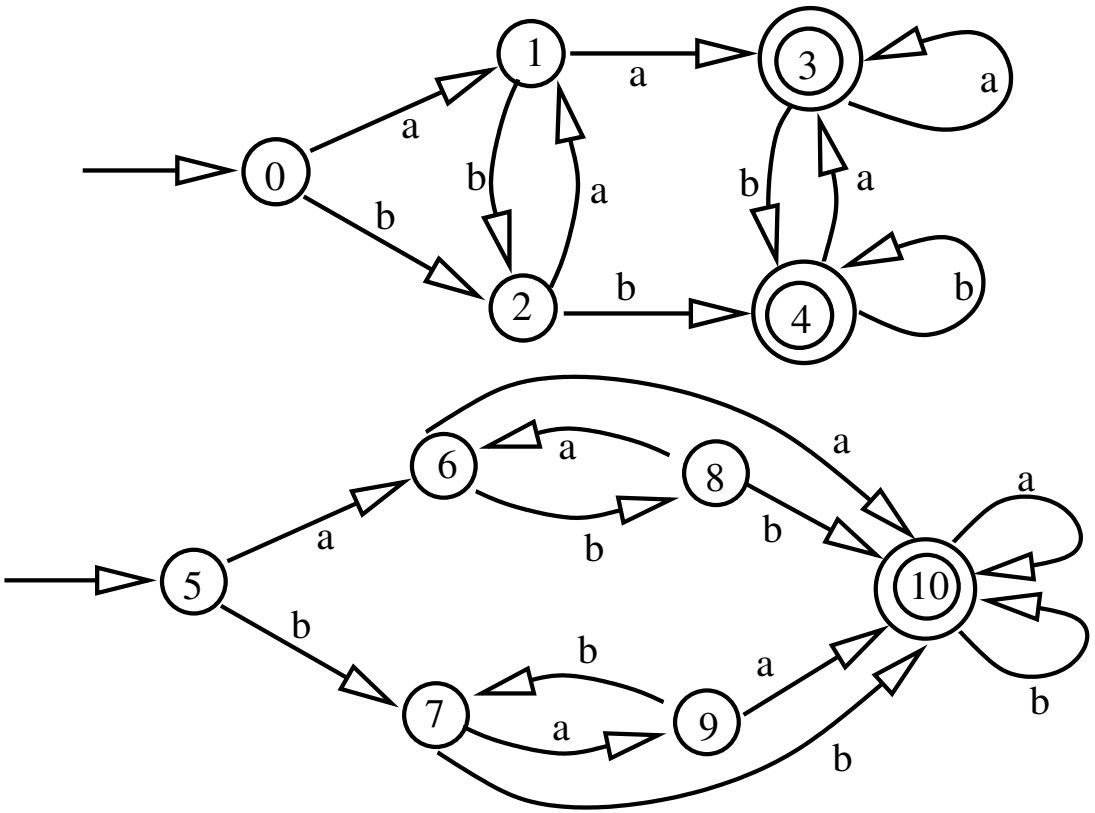


Autre exemple



	a	b
$(0, 5)$	$(1, 6)$	$(2, 7)$
$(1, 6)$	$(\underline{3}, \underline{10})$	$(2, 8)$
$(2, 7)$	$(1, 9)$	$(\underline{4}, \underline{10})$
$(\underline{3}, \underline{10})$	$(\underline{3}, \underline{10})$	$(\underline{4}, \underline{10})$
$(2, 8)$	$(1, 6)$	$(\underline{4}, \underline{10})$
$(1, 9)$	$(\underline{3}, \underline{10})$	$(2, 7)$
$(\underline{4}, \underline{10})$	$(\underline{3}, \underline{10})$	$(\underline{4}, \underline{10})$

Autre exemple



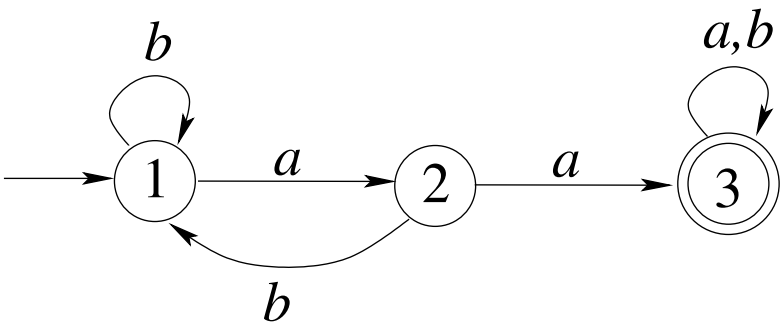
	a	b
$(0, 5)$	$(1, 6)$	$(2, 7)$
$(1, 6)$	$(\underline{3}, \underline{10})$	$(2, 8)$
$(2, 7)$	$(1, 9)$	$(\underline{4}, \underline{10})$
$(\underline{3}, \underline{10})$	$(\underline{3}, \underline{10})$	$(\underline{4}, \underline{10})$
$(2, 8)$	$(1, 6)$	$(\underline{4}, \underline{10})$
$(1, 9)$	$(\underline{3}, \underline{10})$	$(2, 7)$
$(\underline{4}, \underline{10})$	$(\underline{3}, \underline{10})$	$(\underline{4}, \underline{10})$

Le calcul se termine sans trouver de mot qui sépare les deux automates, donc ils sont équivalents.

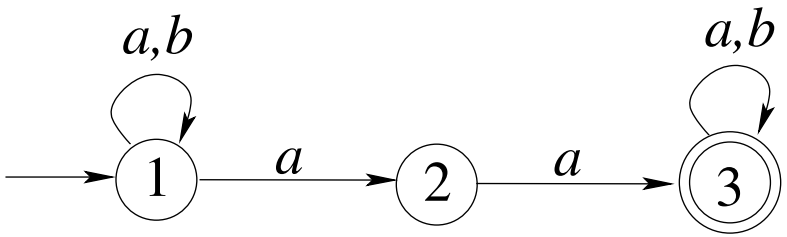
Non-déterminisme

Définition

Un automate est déterministe si pour tout état et toute lettre de l'alphabet, il y a au plus une transition étiquetée par cette lettre qui part de cet état.
Sinon, c'est un automate non-déterministe.

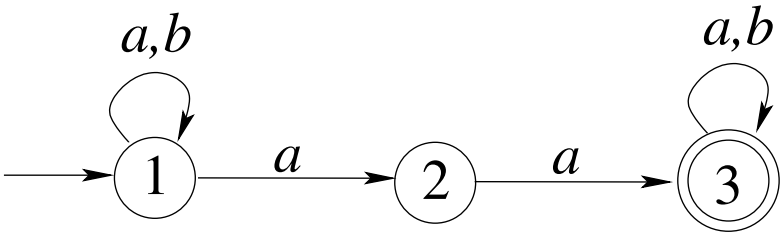


déterministe



non-déterministe

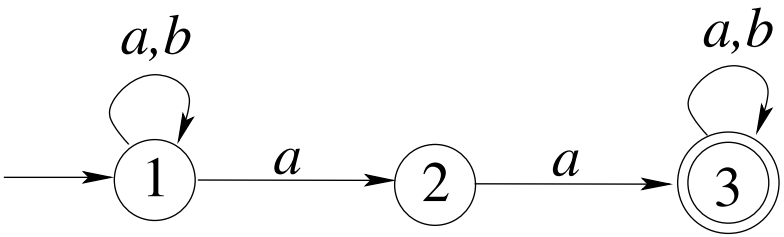
Table de transition d'un automate non-déterministe



	a	*b*
1	1, 2	1
2	3	
3	3	3

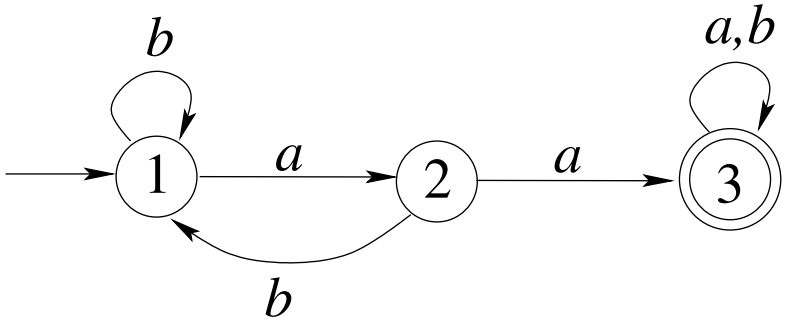
- À partir de l'état 1, en lisant la lettre *a*, on peut atteindre l'état 1 OU l'état 2

Exemple



Cet automate non-déterministe reconnaît le langage L = {mots contenant deux a consécutifs}

Cet automate déterministe reconnaît le même langage L



Déterminisation

Question

Existe-t-il toujours un automate déterministe qui accepte le même langage qu'un automate non-déterministe donné ?

Réponse

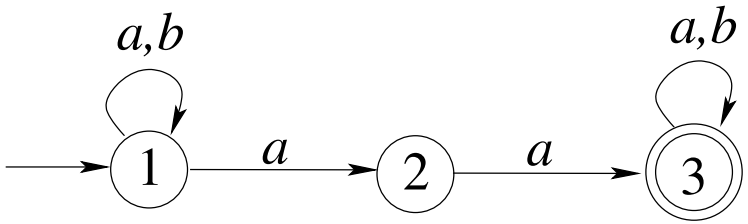
À partir d'un automate non-déterministe qui reconnaît un langage L, on peut toujours calculer un automate déterministe qui reconnaît le même langage L.

Idée générale

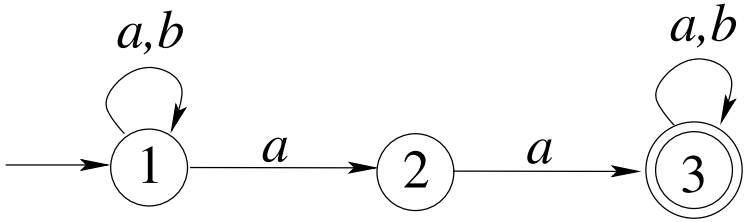
L'automate déterministe que l'on va construire sera composé de *super-états* qui vont *simuler* des ensembles d'états de l'automate initial. La méthode utilisée est la suivante :

- le *super-état* initial est formé de l'état initial ;
- le *super-état* S' issu d'un *super-état* S en appliquant une transition t est formé de l'ensemble des états obtenus en appliquant la transition t à partir de chacun des états formant S ;
- le ou les *super-états* acceptants sont les *super-états* contenant au moins un état final.

Mise en œuvre

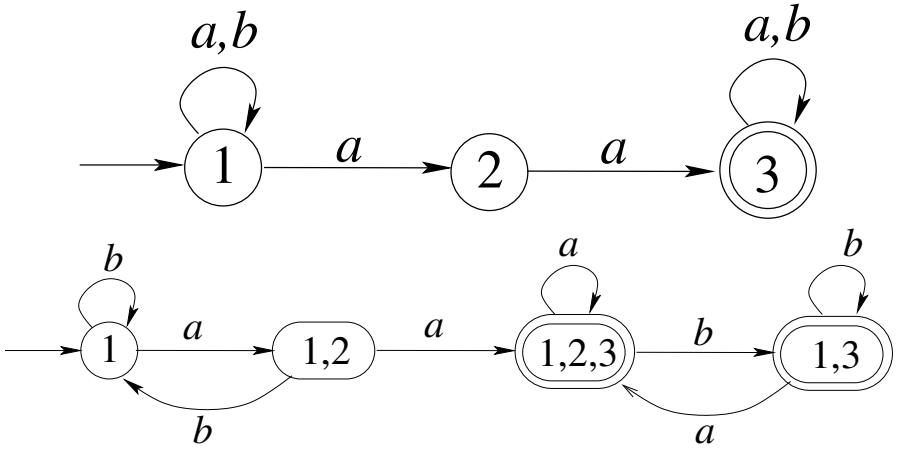


Mise en œuvre



super-états	*a*	*b*
→{1}	{1,2}	{1}
{1,2}	{1,2,3}	{1}
{1,2,3}	{1,2,3}	{1,3}
{1,3}	{1,2,3}	{1,3}

Mise en œuvre



super-états	*a*	*b*
→{1}	{1,2}	{1}
{1,2}	{1,2,3}	{1}
{1,2,3}	{1,2,3}	{1,3}
{1,3}	{1,2,3}	{1,3}

Remarques

- Les automates non-déterministes sont aussi naturels et acceptables que les automates déterministes
- Mais il arrive qu'on soit obligé de déterminiser un automate (par exemple avant d'utiliser la méthode « automates équivalents »), c'est pourquoi vous devez savoir le faire
- Les états de l'automate déterministe obtenu correspondent à des ensembles d'états de l'automate de départ, il peut donc arriver qu'ils soient très nombreux (jusqu'à 2^n, si n est le nombre d'états de l'automate non-déterministe)