Théorie des langages

Opérations sur les automates finis

Jérôme Delobelle
jerome.delobelle@u-paris.fr

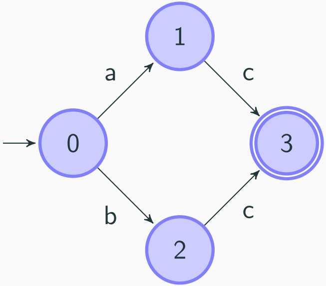
LIPADE - Université de Paris

Opérations sur les automates finis

1. Minimisation
2. Complémentation
3. Transitions instantanées
4. Union
5. Intersection
6. Produit

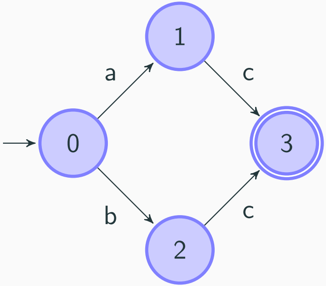
Minimisation

Minimisation : un exemple



- Automate déterministe
- Le langage reconnu est $L = \{ac, bc\}$
- Il existe d'autres automates qui reconnaissent L
- On souhaite trouver le plus petit automate, c'est à dire l'automate complet ayant le plus petit nombre d'états qui reconnaît L

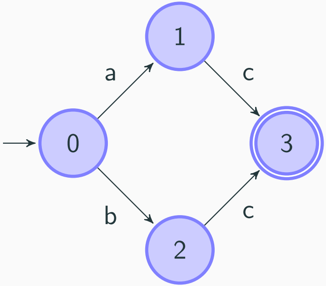
Minimisation : un exemple



- Ici, tous les mots reconnus à partir de l'état 2 le sont aussi à partir de l'état 1

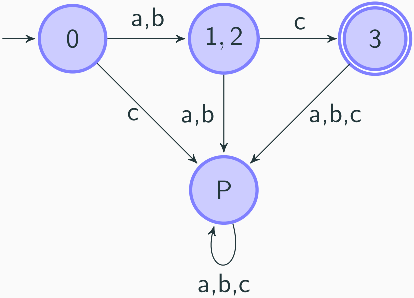
$\Rightarrow$ on fusionne ces deux états, et on complète

Minimisation : un exemple



- Ici, tous les mots reconnus à partir de l'état 2 le sont aussi à partir de l'état 1

$\Rightarrow$ on fusionne ces deux états, et on complète



Minimisation

Théorème

Tout langage rationnel est reconnu par un unique automate déterministe minimal.

Minimisation

Théorème

Tout langage rationnel est reconnu par un unique automate déterministe minimal.

$\rightarrow$ La minimalité porte sur le nombre d'états d'un automate complet !

Minimisation

Donnée : Un automate fini déterministe M

- On veut construire l'automate minimal qui reconnait le langage $\mathcal{L}(M)$

$\Rightarrow$ Fusionner les états équivalents

Etats séparés

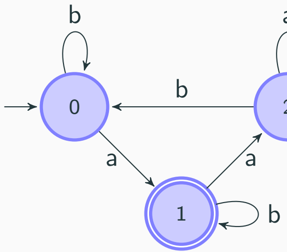
Intuition

Deux états sont séparés par un mot si le chemin étiqueté par ce mot et partant de l'un des deux états aboutit dans un état final, tandis que le chemin étiqueté par ce mot et partant de l'autre état aboutit dans un état qui n'est pas final.

Etats séparés

Intuition

Deux états sont séparés par un mot si le chemin étiqueté par ce mot et partant de l'un des deux états aboutit dans un état final, tandis que le chemin étiqueté par ce mot et partant de l'autre état aboutit dans un état qui n'est pas final.



- ϵ sépare les états 1 et 2
- *aab* ne sépare pas 1 et 2
- ϵ sépare les états 0 et 1
- *a* sépare les états 0 et 2

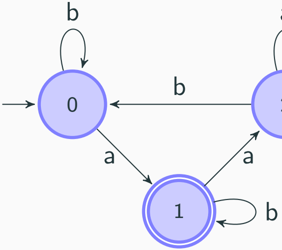
Test de minimalité

Minimalité

Un automate déterministe complet est minimal si et seulement si pour tout couple d'états p et q, il existe un mot qui sépare p et q.

- Ce résultat donne un moyen de montrer qu'un automate est minimal
  - → Il suffit d'exhiber pour chaque paire d'états p et q un mot qui les sépare
  - Par exemple, l'automate précédent est minimal
- Deux états qui ne sont séparés par aucun mot seront dit équivalents

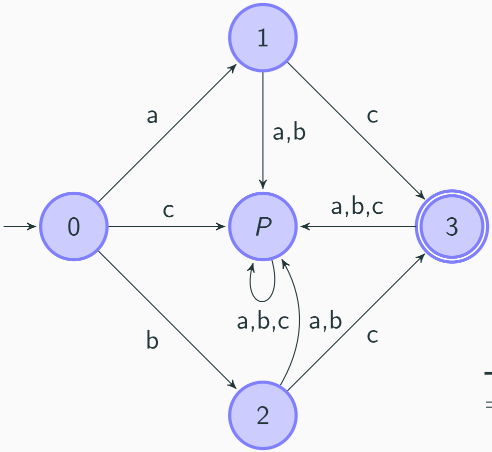
Exemple 1



- ϵ sépare les états 1 et 2
- ϵ sépare les états 0 et 1
- a sépare les états 0 et 2

$\Rightarrow$ automate minimal

Exemple 2



- c sépare les états 0 et 1
- c sépare les états 0 et 2
- ϵ sépare les états 0 et 3
- ac sépare les états 0 et P
- c sépare les états 2 et 3
- ...
- aucun mot ne sépare les états 1 et 2

$\Rightarrow$ automate non minimal

Congruence de Nérode

Equivalence d'états

Etant donné un **automate fini déterministe** M, deux états p et q sont équivalents si leurs langages associés respectifs sont identiques :

$$p \approx q \text{ si et seulement si } L_p(M) = L_q(M)$$

Autrement dit,
Soient p', q' définis $\forall w \in \Sigma^*$, par $(q, w) \overset{*}{\mapsto} (q', \epsilon)$ et $(p, w) \overset{*}{\mapsto} (p', \epsilon)$.
Alors

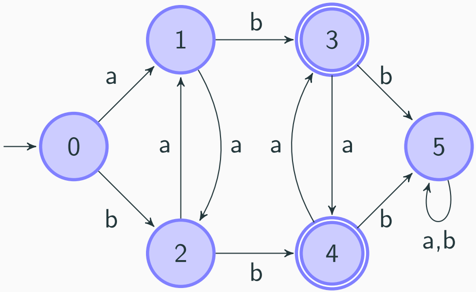
$$(p \approx q) \Leftrightarrow \forall w \in \Sigma^*, \begin{cases} q' \in F \text{ et } p' \in F, \text{ ou} \\ q' \notin F \text{ et } p' \notin F \end{cases}$$

- $p \approx q$ signifie que p et q ne sont séparés par aucun mot
- On note $p \approx_n q$ le fait que p et q ne sont séparés par aucun mot de longueur inférieure ou égale à n

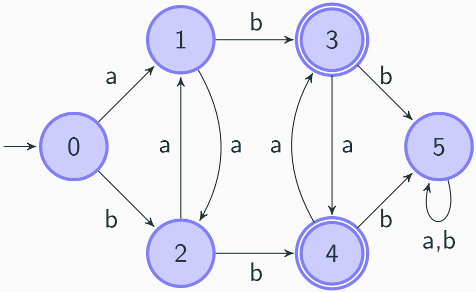
Congruence de Nérode

- La relation $\approx$ est une relation d'équivalence appelée congruence de Nérode.
  - Rappel : une relation d'équivalence est réflexive, symétrique et transitive
- $\forall n \geq 0$, la relation $\approx_n$ est également une relation d'équivalence
- Si q est un état, on note $[q]$ l'ensemble des états qui lui sont équivalents

Exemple

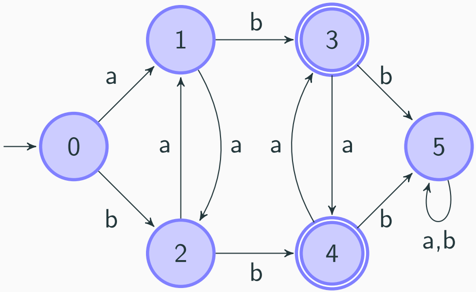


Exemple



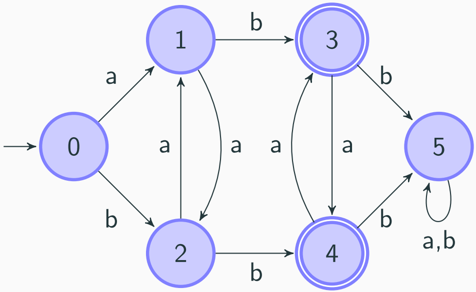
- $0 \approx 3$?

Exemple



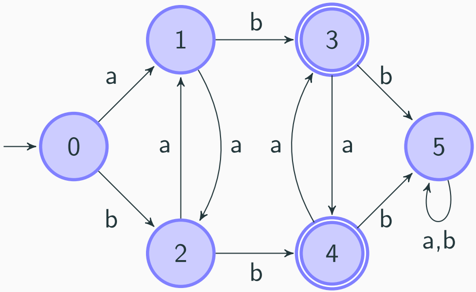
- $0 \approx 3$? Non, ϵ sépare 0 et 3

Exemple



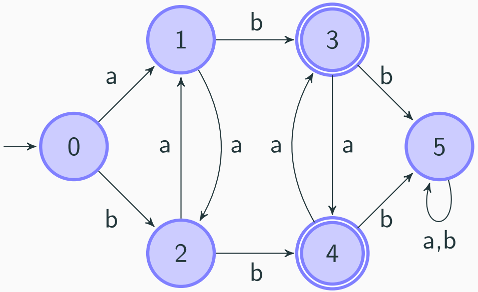
- $0 \approx 3$? Non, ϵ sépare 0 et 3
- $0 \approx 1$?

Exemple



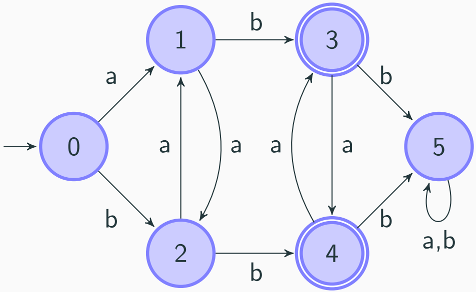
- $0 \approx 3$? Non, ϵ sépare 0 et 3
- $0 \approx 1$? Non, b sépare 0 et 1

Exemple



- $0 \approx 3$? Non, ϵ sépare 0 et 3
- $0 \approx 1$? Non, b sépare 0 et 1
- $3 \approx 4$?

Exemple



- $0 \approx 3$? Non, ϵ sépare 0 et 3
- $0 \approx 1$? Non, b sépare 0 et 1
- $3 \approx 4$? Oui
  - $3 \in F$, $4 \in F$, et
  - $\forall w \in \Sigma^*$, $(3, aw) \overset{*}{\mapsto} (p, \epsilon)$, $(4, aw) \overset{*}{\mapsto} (q, \epsilon)$, avec $p, q \in F$;
  - $\forall w \in \Sigma^*$, $(3, bw) \overset{*}{\mapsto} (p, \epsilon)$, $(4, bw) \overset{*}{\mapsto} (q, \epsilon)$, avec $p, q \notin F$;

Automate minimal

Soit $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un automate complet déterministe dont tous les états sont accessibles.

L'automate minimal associé à M est $M_{min} = \langle Q', \Sigma, \delta', [q_0], F' \rangle$, avec

- $Q' = \{[q], q \in Q\}$
- $\delta' = \{([p], x, [q] \mid \exists p' \in [p], \exists q' \in [q]$ tels que $(p', x, q') \in \delta\}$
- $F' = \{[f], f \in F\}$

Algorithme de Moore

Donnée : un automate **complet déterministe accessible**

Résultat : la congruence de Nérode et l'automate minimal reconnaissant le langage reconnu par l'automate donné en entrée

Principe : l'algorithme calcule lettre par lettre les mots séparant les états, et attribue un numéro (en chiffre romain) à chaque classe d'équivalence d'états.

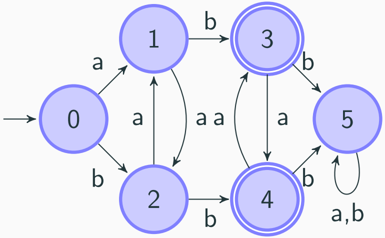
Algorithme de Moore

1. Construire un tableau dont les colonnes sont les différents états de l'automate
2. Séparer les états finaux et non finaux en deux classes :
   - Numéroter par I l'état de la première colonne ;
   - Numéroter par I ou II chaque état, de façon à associer le même numéro à tous les états finaux, et l'autre numéro à tous les états non finaux
3. Créer une ligne pour chaque $x \in \Sigma$. Associer à chaque état p le numéro correspondant à l'état q tel que $\delta(p, x) = q$
4. **Bilan :** on compare les numéro obtenus sur chaque ligne pour chaque état : deux colonnes différentes donnent deux classes d'équivalences différentes
   - Numéroter par I l'état de la première colonne
   - Si la seconde colonne est identique à la première, lui associer le numéro I . Sinon, lui associer le numéro II.
   - Poursuivre jusqu'à ce que toutes les colonnes soient numérotées
5. Recommencer les étapes 3. et 4. jusqu'à obtenir deux bilans successifs identiques

Algorithme de Moore

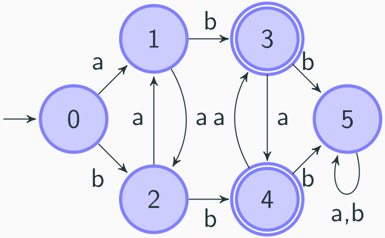
- Les états de l'automate minimal complet sont les classes obtenues dans le dernier bilan
- Les transitions sont celles trouvées lors de la dernière application de l'étape 3.
- L'état initial de l'automate minimal est la classe d'équivalence contenant l'état initial de l'automate donné en entrée
- Les états finaux sont les classes d'équivalences qui contiennent des états finaux de l'automate donné en entrée

Exemple



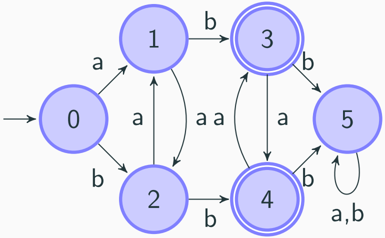
	0	1	2	3	4	5
ϵ						

Exemple



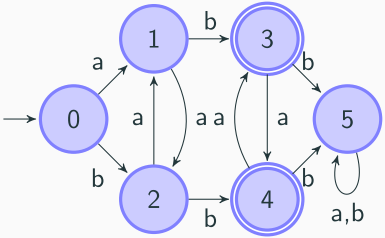
	0	1	2	3	4	5
ϵ	I	I	I	II	II	I

Exemple



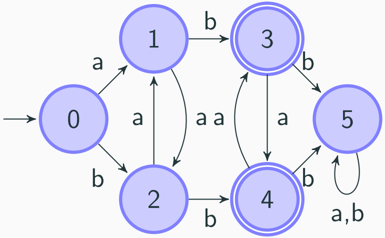
	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a						

Exemple



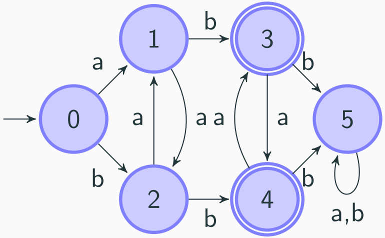
	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a	I	I	I	II	II	I

Exemple



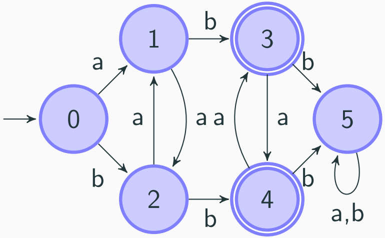
	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a	I	I	I	II	II	I
b	I	II	II	I	I	I

Exemple



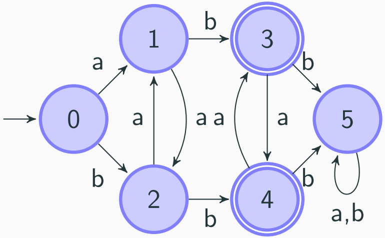
	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a	I	I	I	II	II	I
b	I	II	II	I	I	I
Bilan	I					

Exemple



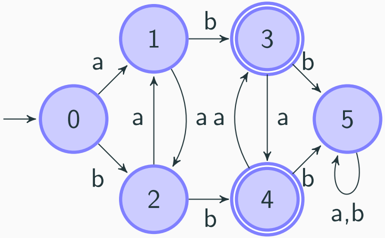
	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a	I	I	I	II	II	I
b	I	II	II	I	I	I
Bilan	I	II				

Exemple



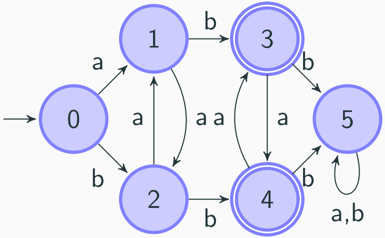
	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a	I	I	I	II	II	I
b	I	II	II	I	I	I
Bilan	I	II	II			

Exemple



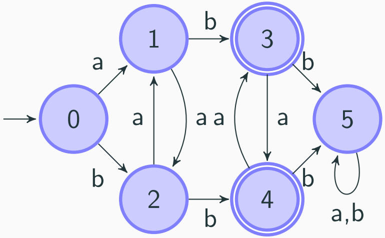
	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a	I	I	I	II	II	I
b	I	II	II	I	I	I
Bilan	I	II	II	III	III	I

Exemple



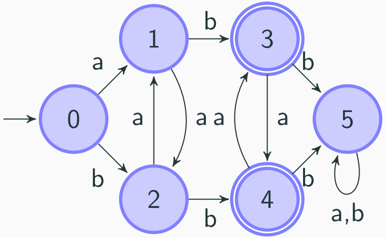
	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a	I	I	I	II	II	I
b	I	II	II	I	I	I
Bilan	I	II	II	III	III	I
a	II	II	II	III	III	I

Exemple



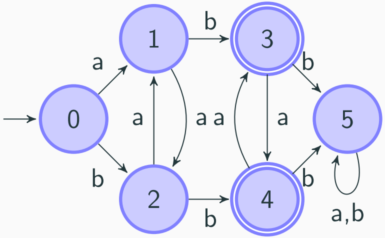
	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a	I	I	I	II	II	I
b	I	II	II	I	I	I
Bilan	I	II	II	III	III	I
a	II	II	II	III	III	I
b	II	III	III	I	I	I

Exemple



	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a	I	I	I	II	II	I
b	I	II	II	I	I	I
Bilan	I	II	II	III	III	I
a	II	II	II	III	III	I
b	II	III	III	I	I	I
Bilan	I	II	II	III	III	IV

Exemple

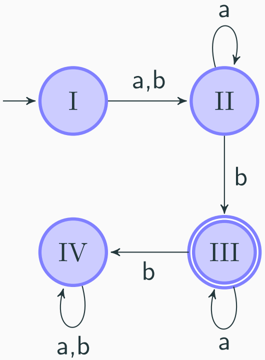


	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a	I	I	I	II	II	I
b	I	II	II	I	I	I
Bilan	I	II	II	III	III	I
a	II	II	II	III	III	I
b	II	III	III	I	I	I
Bilan	I	II	II	III	III	IV
a	II	II	II	III	III	IV
b	II	III	III	IV	IV	IV
Bilan	I	II	II	III	III	IV

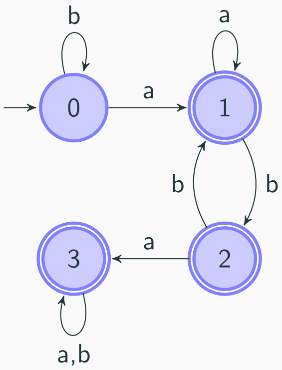
Exemple

	0	1	2	3	4	5
ϵ	I	I	I	II	II	I
a	I	I	I	II	II	I
b	I	II	II	I	I	I
Bilan	I	II	II	III	III	I
a	II	II	II	III	III	I
b	II	III	III	I	I	I
Bilan	I	II	II	III	III	IV
a	II	II	II	III	III	IV
b	II	III	III	IV	IV	IV
Bilan	I	II	II	III	III	IV

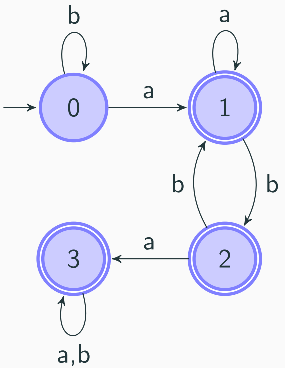
On obtient :



Un autre exemple

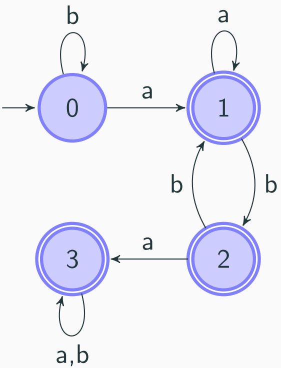


Un autre exemple



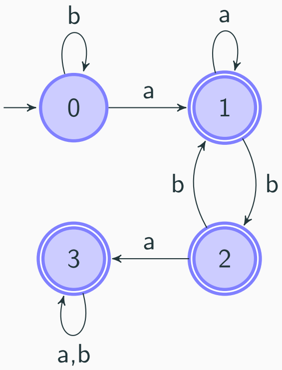
	0	1	2	3
ϵ	I	II	II	II

Un autre exemple



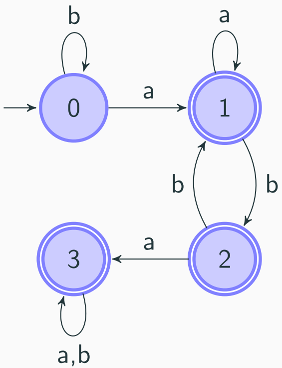
	0	1	2	3
ϵ	I	II	II	II
a	II	II	II	II

Un autre exemple



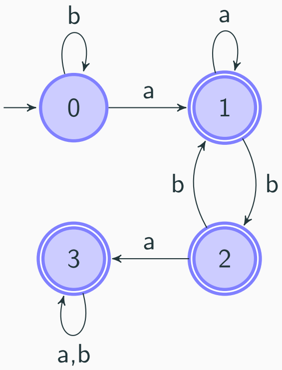
	0	1	2	3
ϵ	I	II	II	II
a	II	II	II	II
b	I	II	II	II

Un autre exemple



	0	1	2	3
ϵ	I	II	II	II
a	II	II	II	II
b	I	II	II	II
Bilan	I	II	II	II

Un autre exemple

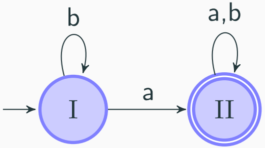


	0	1	2	3
ϵ	I	II	II	II
a	II	II	II	II
b	I	II	II	II
Bilan	I	II	II	II
a	II	II	II	II
b	I	II	II	II
Bilan	I	II	II	II

Un autre exemple

	0	1	2	3
ϵ	I	II	II	II
a	II	II	II	II
b	I	II	II	II
Bilan	I	II	II	II
a	II	II	II	II
b	I	II	II	II
Bilan	I	II	II	II

On obtient :



Complémentation

Complémentation

Complémentation

Pour tout automate fini M, il existe un automate fini M' tel que $\mathcal{L}(M') = \overline{\mathcal{L}(M)}$

Complémentation

Complémentation

Pour tout automate fini M, il existe un automate fini M' tel que $\mathcal{L}(M') = \overline{\mathcal{L}(M)}$

Intérêt : pour tout automate complet et déterministe M, et pour tout mot $w \in X^*$, il existe une dérivation $(s, w) \overset{*}{\mapsto} (q, \epsilon)$ telle que

- soit $q \in F$ et $w \in \mathcal{L}(M)$
- soit $q \notin F$ et $w \notin \mathcal{L}(M)$

$\Rightarrow$ Permet de pouvoir accepter tous les mots de X^* qui ne sont pas acceptés par M.

Complémentation

Complémentation

Pour tout automate fini M, il existe un automate fini M' tel que $\mathcal{L}(M') = \overline{\mathcal{L}(M)}$

Intérêt : pour tout automate complet et déterministe M, et pour tout mot $w \in X^*$, il existe une dérivation $(s, w) \overset{*}{\mapsto} (q, \epsilon)$ telle que

- soit $q \in F$ et $w \in \mathcal{L}(M)$
- soit $q \notin F$ et $w \notin \mathcal{L}(M)$

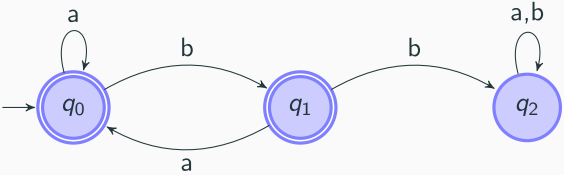
$\Rightarrow$ Permet de pouvoir accepter tous les mots de X^* qui ne sont pas acceptés par M.

Méthode pour trouver l'automate complémentaire

1. Déterminiser et compléter l'automate
2. Transformer tous les états finaux en états non finaux, et vice-versa

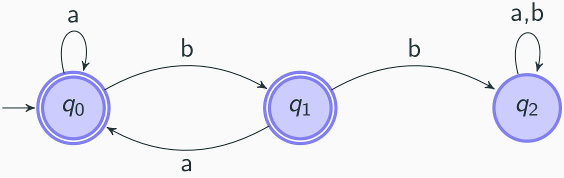
Exemple : Complémentation

- Automate M_1 complété

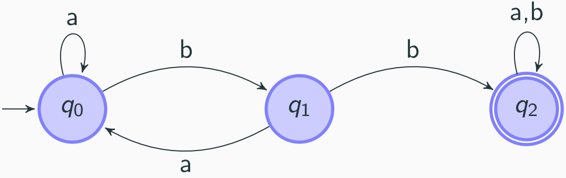


Exemple : Complémentation

- Automate M_1 complété



- Automate complémentaire à M_1 complété



Transitions instantanées

Automate avec transitions instantanées

Automate avec transitions instantanées

Une transition instantanée ou ϵ-transition permet de passer d'un état à un autre sans lire de symbole sur le ruban d'entrée.

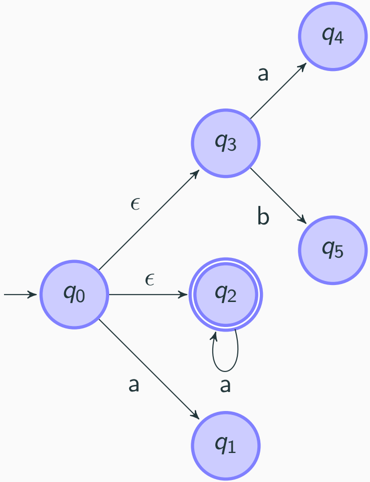
Automate avec transitions instantanées

Automate avec transitions instantanées

Une transition instantanée ou ϵ-transition permet de passer d'un état à un autre sans lire de symbole sur le ruban d'entrée.

Une ϵ-transition permet "d'ajouter" à un état les comportements de l'état cible (transition et éventuellement état final).

Exemple : transition instantanée



Union

Union de deux automates

Union

Soit deux automates finis A_1 et A_2. Il existe un automate fini qui reconnait $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$.

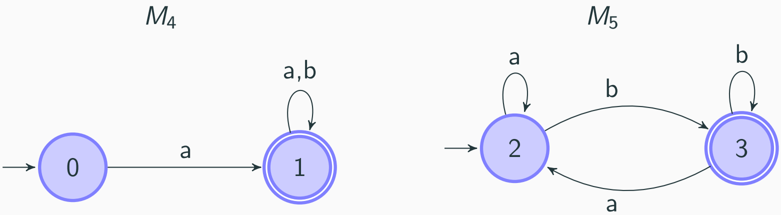
Union de deux automates

Union

Soit deux automates finis A_1 et A_2. Il existe un automate fini qui reconnait $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$.

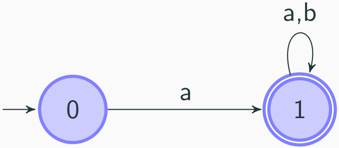
Pour calculer l'union de deux automates, il faut calculer l'équation qui correspond à chacun des langages.

Exemple : Union

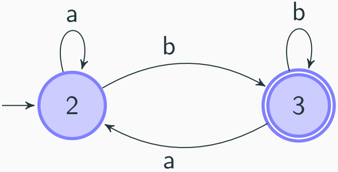


Exemple : Union

M_4



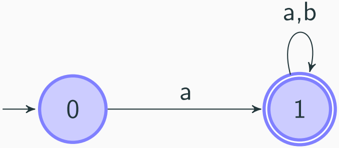
M_5

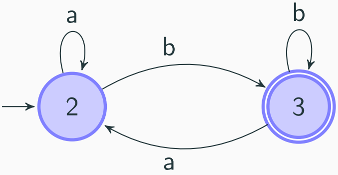


$$\mathcal{L}(M_4)\begin{cases} L_0 = aL_1 \\ L_1 = aL_1 + bL_1 + \epsilon \end{cases}$$

$$\mathcal{L}(M_5)\begin{cases} L_2 = aL_2 + bL_3 \\ L_3 = aL_2 + bL_3 + \epsilon \end{cases}$$

Exemple : Union





$$\mathcal{L}(M_4)\left\{\begin{array}{lcl} L_0 & = & aL_1 \\ L_1 & = & aL_1 + bL_1 + \epsilon \end{array}\right. \qquad \mathcal{L}(M_5)\left\{\begin{array}{lcl} L_2 & = & aL_2 + bL_3 \\ L_3 & = & aL_2 + bL_3 + \epsilon \end{array}\right.$$

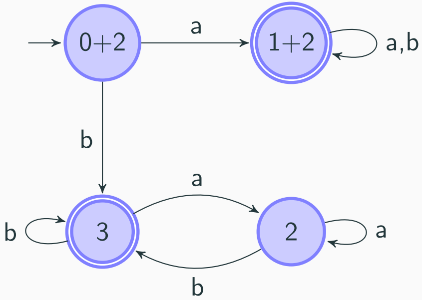
$$\mathcal{L}(M_4) \cup \mathcal{L}(M_5) = L_0 + L_2 \left\{\begin{array}{lcl} L_0 + L_2 & = & a(L_1 + L_2) + bL_3 \\ L_1 + L_2 & = & a(L_1 + L_2) + b(L_1 + L_3) + \epsilon \\ L_1 + L_3 & = & a(L_1 + L_2) + b(L_1 + L_3) + \epsilon \\ & = & L_1 + L_2 \\ L_3 & = & aL_2 + bL_3 + \epsilon \\ L_2 & = & aL_2 + bL_3 \end{array}\right.$$

Exemple : Union

$$\mathcal{L}(M_4) \cup \mathcal{L}(M_5) \begin{cases} L_0 + L_2 & = & a(L_1 + L_2) + bL_3 \\ L_1 + L_2 & = & a(L_1 + L_2) + b(L_1 + L_3) + \epsilon \\ L_3 & = & aL_2 + bL_3 + \epsilon \\ L_2 & = & aL_2 + bL_3 \end{cases}$$

Exemple : Union

$$\mathcal{L}(M_4) \cup \mathcal{L}(M_5) \begin{cases} L_0 + L_2 & = & a(L_1 + L_2) + bL_3 \\ L_1 + L_2 & = & a(L_1 + L_2) + b(L_1 + L_3) + \epsilon \\ L_3 & = & aL_2 + bL_3 + \epsilon \\ L_2 & = & aL_2 + bL_3 \end{cases}$$



Intersection

Intersection de deux automates

Intersection

Soit deux automates finis A_1 et A_2 qui reconnaissent $\mathcal{L}(A_1)$ et $\mathcal{L}(A_2)$.
Alors, il existe un automate fini qui reconnait $\mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

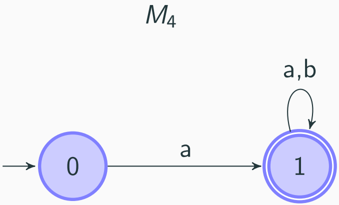
Intersection de deux automates

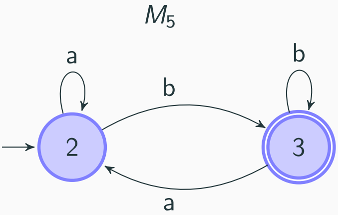
Intersection

Soit deux automates finis A_1 et A_2 qui reconnaissent $\mathcal{L}(A_1)$ et $\mathcal{L}(A_2)$. Alors, il existe un automate fini qui reconnait $\mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

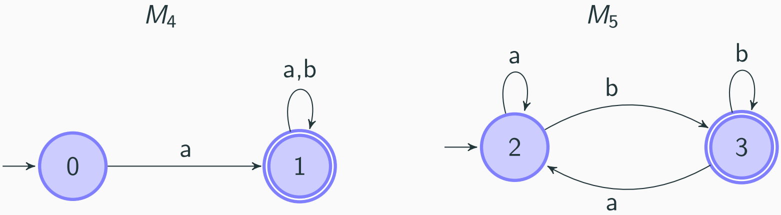
$$\mathcal{L}(A_1) \cap \mathcal{L}(A_2) = \overline{\overline{\mathcal{L}(A_1)} \cup \overline{\mathcal{L}(A_2)}}$$

Exemple : Intersection



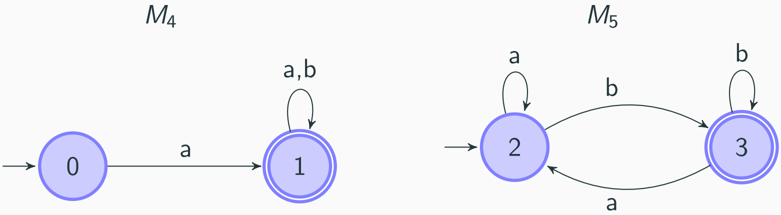


Exemple : Intersection



On déterminise M_4 et M_5

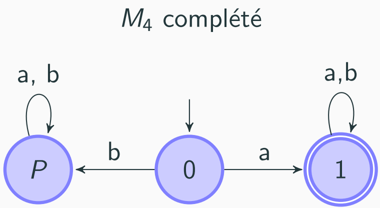
Exemple : Intersection

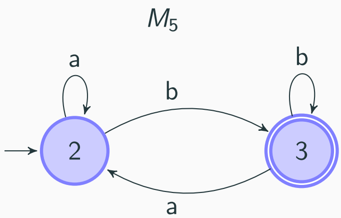


On déterminise M_4 et M_5

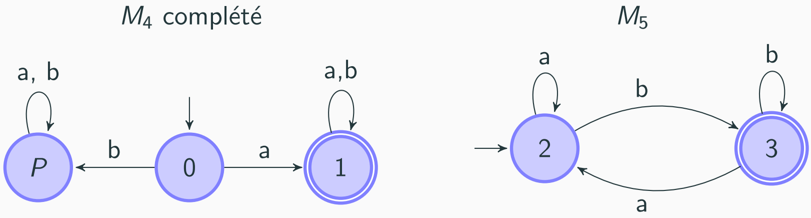
On complète M_4 et M_5

Exemple : Intersection



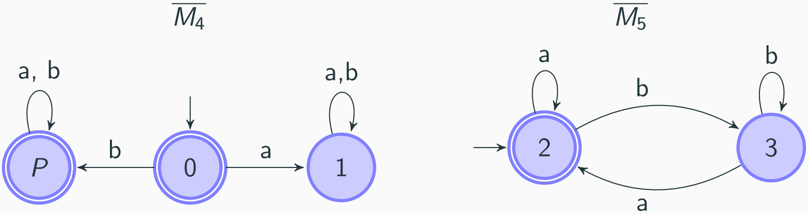


Exemple : Intersection

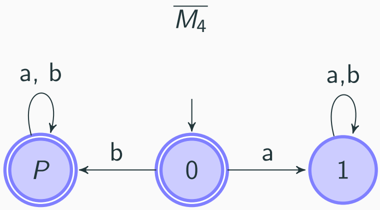


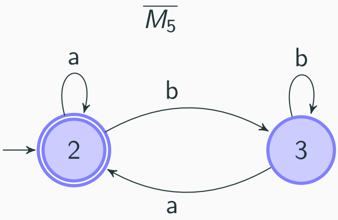
On complémente M_4 et M_5

Exemple : Intersection



Exemple : Intersection

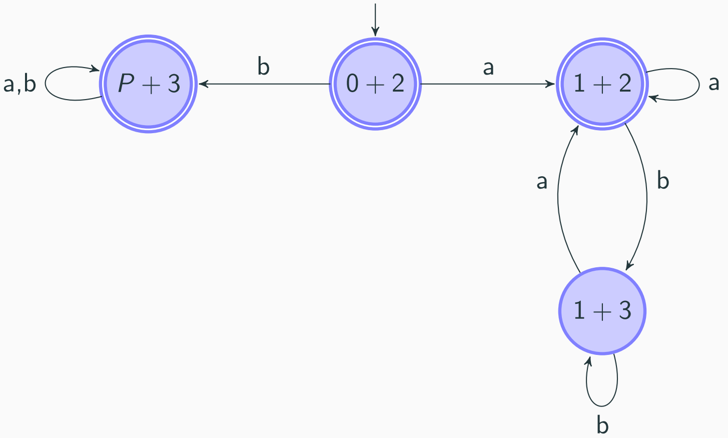




On fait l'union de $\overline{M_4}$ et $\overline{M_5}$

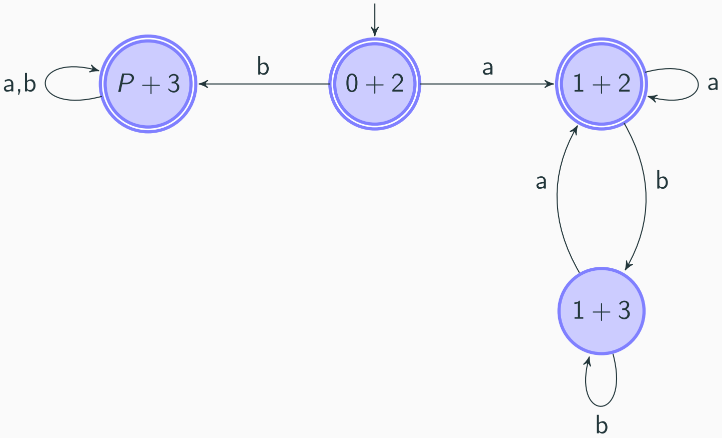
Exemple : Intersection

$\mathcal{L}(M_{45}) = \overline{\mathcal{L}(M_4)} \cup \overline{\mathcal{L}(M_5)}$. On obtient M_{45} :



Exemple : Intersection

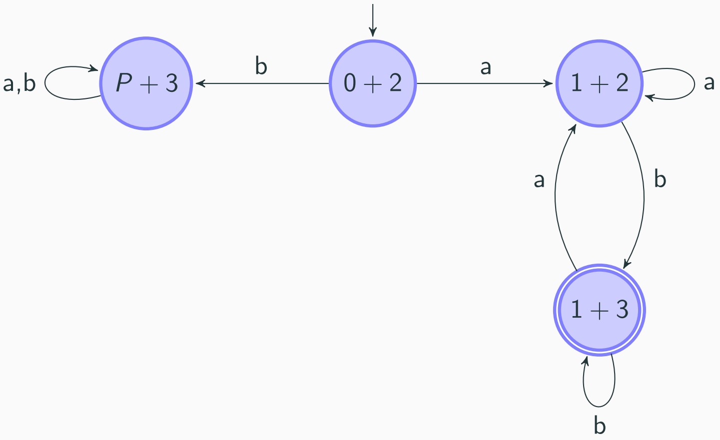
$\mathcal{L}(M_{45}) = \overline{\mathcal{L}(M_4)} \cup \overline{\mathcal{L}(M_5)}$. On obtient M_{45} :



On complémente M_{45}

Exemple : Intersection

On obtient : $\overline{\mathcal{L}(M_{45})} = \overline{\overline{\mathcal{L}(M_4)} \cup \overline{\mathcal{L}(M_5)}}$, soit l'automate :

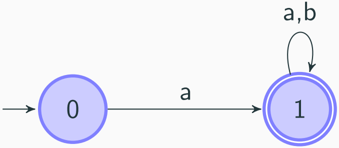


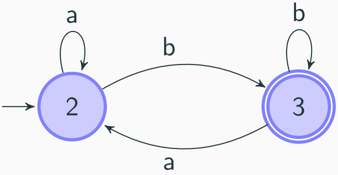
Produit

Produit de deux automates

Produit

Soit deux automates finis A_1 et A_2 qui reconnaissent $\mathcal{L}(A_1)$ et $\mathcal{L}(A_2)$.
Alors, il existe un automate fini qui reconnait $\mathcal{L}(A_1).\mathcal{L}(A_2)$.

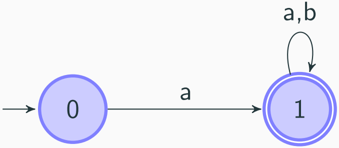


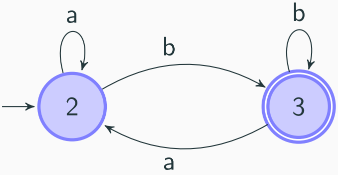


$$\mathcal{L}(M_4)\begin{cases} L_0 &= aL_1 \\ L_1 &= aL_1 + bL_1 + \epsilon \end{cases}$$

$$\mathcal{L}(M_5)\begin{cases} L_2 &= aL_2 + bL_3 \\ L_3 &= aL_2 + bL_3 + \epsilon \end{cases}$$

Exemple : Produit





$$\mathcal{L}(M_4)\left\{\begin{array}{lcl} L_0 & = & aL_1 \\ L_1 & = & aL_1 + bL_1 + \epsilon \end{array}\right.$$

$$\mathcal{L}(M_5)\left\{\begin{array}{lcl} L_2 & = & aL_2 + bL_3 \\ L_3 & = & aL_2 + bL_3 + \epsilon \end{array}\right.$$

$$\mathcal{L}(M_4).\mathcal{L}(M_5) = L_0L_2 \left\{\begin{array}{ll} L_0L_2 & = aL_1L_2 \\ L_1L_2 & = aL_1L_2 + bL_1L_2 + L_2 \\ & = aL_1L_2 + bL_1L_2 + aL_2 + bL_3 \\ & = a(L_1L_2 + L_2) + b(L_1.L_2 + L_3) \\ L_1L_2 + L_2 & = a(L_1L_2 + L_2 + L_2) + b(L_1L_2 + L_3 + L_3) \\ L_1L_2 + L_3 & = a(L_1L_2 + L_2) + b(L_1L_2 + L_3) + \epsilon \end{array}\right.$$

Exemple : Produit

$$\mathcal{L}(M_4).\mathcal{L}(M_5)\begin{cases} L_0.L_2 & = aL_1.L_2 \\ L_1.L_2 & = a(L_1.L_2) + b(L_1.L_2 + L_3) \\ L_1 L_2 + L_3 & = a(L_1.L_2) + b(L_1.L_2 + L_3) + \epsilon \end{cases}$$

Exemple : Produit

$$\mathcal{L}(M_4).\mathcal{L}(M_5) \begin{cases} L_0.L_2 & = aL_1.L_2 \\ L_1.L_2 & = a(L_1.L_2) + b(L_1.L_2 + L_3) \\ L_1L_2 + L_3 & = a(L_1.L_2) + b(L_1.L_2 + L_3) + \epsilon \end{cases}$$