LASF – Minimisation des automates finis

Christophe Mouilleron

ensiie
école nationale supérieure d'informatique
pour l'industrie et l'entreprise

Plan

1. Minimalité d'un automate fini déterministe
2. Procédés de minimalisation
   - À partir d'un automate existant
   - À partir d'un langage
3. Applications de la minimalité

Plan

1. Minimalité d'un automate fini déterministe
2. Procédés de minimalisation
   - À partir d'un automate existant
   - À partir d'un langage
3. Applications de la minimalité

Notion de minimalité

Définition (automate minimal)

On dit qu'un automate $\mathcal{A}$ reconnaissant un langage L est un automate minimal lorsque :

1. $\mathcal{A}$ déterministe,
2. pour tout automate $\mathcal{B}$ avec moins d'états que $\mathcal{A}$, $L(\mathcal{B}) \neq L$

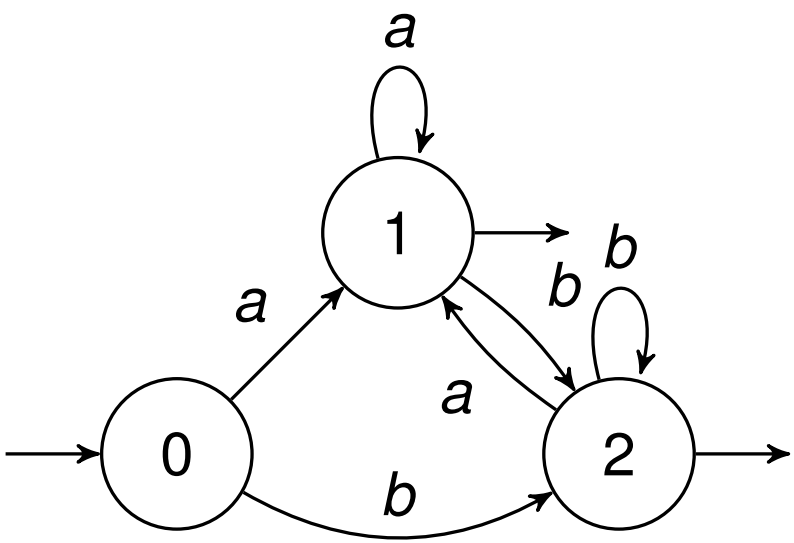
Notion de minimalité

Définition (automate minimal)

On dit qu'un automate $\mathcal{A}$ reconnaissant un langage L est un automate minimal lorsque :

1. $\mathcal{A}$ déterministe,
2. pour tout automate $\mathcal{B}$ avec moins d'états que $\mathcal{A}$, $L(\mathcal{B}) \neq L$

Exemple :



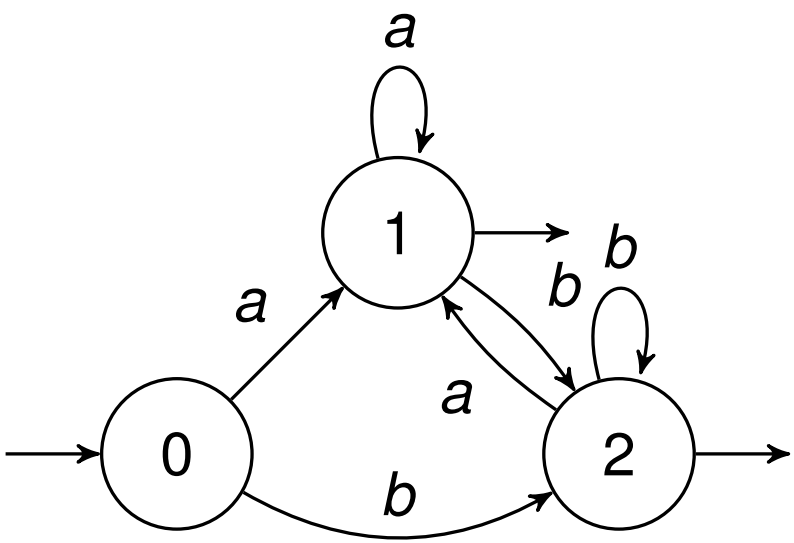
Notion de minimalité

Définition (automate minimal)

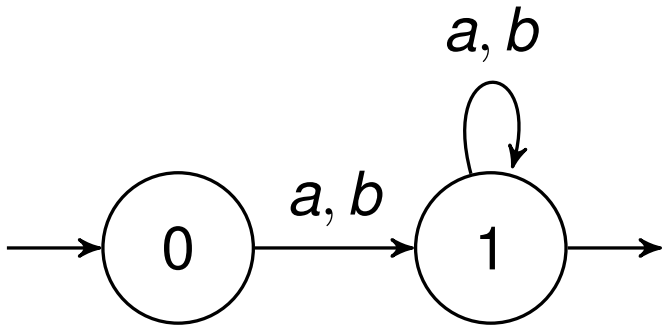
On dit qu'un automate $\mathcal{A}$ reconnaissant un langage L est un automate minimal lorsque :

1. $\mathcal{A}$ déterministe,
2. pour tout automate $\mathcal{B}$ avec moins d'états que $\mathcal{A}$, $L(\mathcal{B}) \neq L$

Exemples : $L = A^+$



Non minimal



Minimal

Séparation

Définition (séparation)

Deux états s et t sont séparables s'il existe un mot $u \in A^*$ tel que :

- soit $\delta(s, u) \in F$ et $\delta(t, u) \notin F$, — u reconnu depuis s, pas depuis t
- soit $\delta(s, u) \notin F$ et $\delta(t, u) \in F$. — u reconnu depuis t, pas depuis s

On dit alors que u sépare s et t.

Idée :

- u sépare s et t $\Rightarrow$ s et t différents en terme de reconnaissance
- s et t non séparables $\Rightarrow$ même rôle en terme de reconnaissance

Propriétés remarquables

Propriété

Le mot ε sépare les états finaux (F) de ceux qui ne le sont pas ($Q \setminus F$).

Propriété

Dans un automate fini déterministe complet accessible, les états puits sont toujours séparés des autres.

$\rightsquigarrow$ Du coup, un seul état puits suffit !

Équivalence de Nérode

Notation (équivalence de Nérode)

On note $s \sim t$ lorsque s et t ne sont pas séparables.

Propriété :

- La relation $\sim$ est bien une relation d'équivalence.

Équivalence de Nérode

Notation (équivalence de Nérode)

On note $s \sim t$ lorsque s et t ne sont pas séparables.

Propriété :

- La relation $\sim$ est bien une relation d'équivalence.

Idée du procédé de minimisation :

1. calcul des classes d'équivalence pour $\sim$
2. fusion des états dans la même classe d'équivalence

Plan

1 Minimalité d'un automate fini déterministe

2 Procédés de minimalisation
- À partir d'un automate existant
- À partir d'un langage

3 Applications de la minimalité

Plan

1. Minimalité d'un automate fini déterministe
2. Procédés de minimalisation
   - À partir d'un automate existant
   - À partir d'un langage
3. Applications de la minimalité

Algorithme de Moore

Algorithme de minimisation de Moore = processus de raffinement

Idée :

- calcul des séparations par des mots de longueur au plus n
- pour $n = 0$, on a 2 classes ε sépare F et $Q \setminus F$

Algorithme de Moore

Algorithme de minimisation de Moore = processus de raffinement

Idée :

- calcul des séparations par des mots de longueur au plus n
- pour $n = 0$, on a 2 classes — ε sépare F et $Q \setminus F$
- pour passer de n à $n + 1$:
  1. pour chaque état, écrire son paquet courant et les paquets d'arrivée pour chaque lettre
  2. créer de nouveaux paquets en regroupant les états dont les résultats sont identiques

Algorithme de Moore

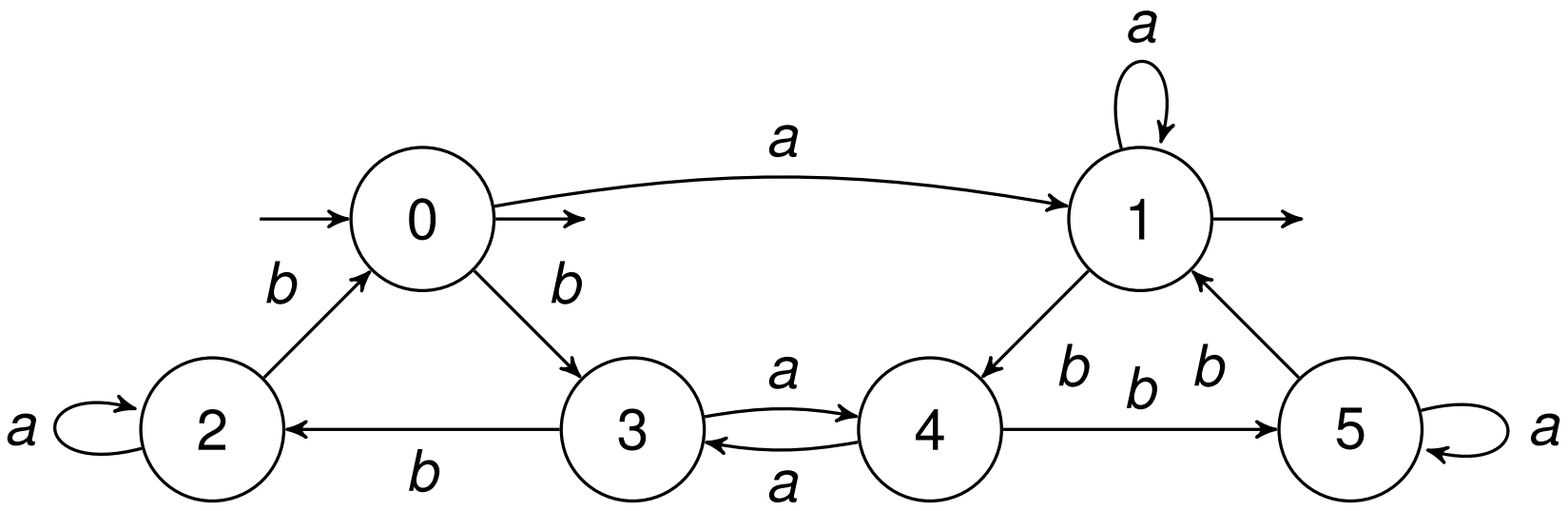
Algorithme de minimisation de Moore = processus de raffinement

Idée :

- calcul des séparations par des mots de longueur au plus n
- pour $n = 0$, on a 2 classes — ε sépare F et $Q \setminus F$
- pour passer de n à $n + 1$:
  1. pour chaque état, écrire son paquet courant et les paquets d'arrivée pour chaque lettre
  2. créer de nouveaux paquets en regroupant les états dont les résultats sont identiques
- incrémenter n jusqu'à atteindre un point fixe

Algorithme de Moore – Exemple

On cherche à minimiser l'automate :



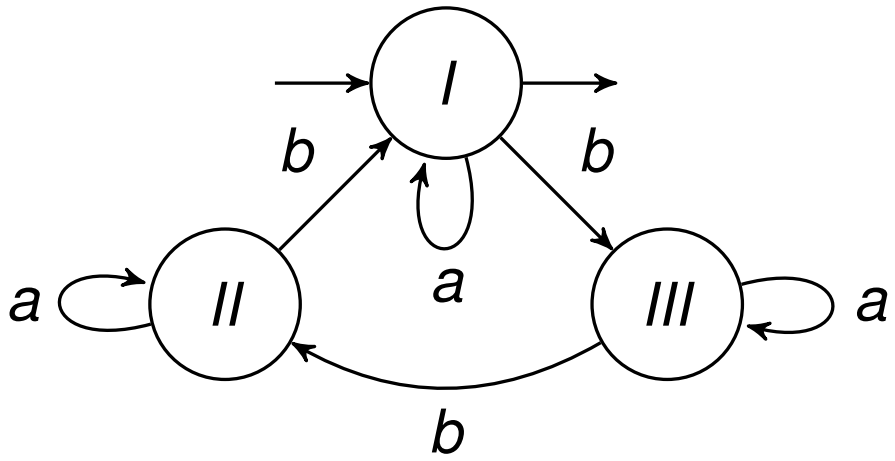
$\rightsquigarrow$ Exemple traité au tableau.

Algorithme de Moore – Exemple (suite)

On arrive à :

états	0	1	2	3	4	5
étape 0	I	I	II	II	II	II
$\rightarrow$	(I,II)	(I,II)	(II,I)	(II,II)	(II,II)	(II,I)
étape 1	I	I	II	III	III	II
$\rightarrow$	(I,III)	(I,III)	(II,I)	(III,II)	(III,II)	(II,I)
étape 2	I	I	II	III	III	II

Automate minimal :



Plan

1. Minimalité d'un automate fini déterministe
2. Procédés de minimalisation
   - À partir d'un automate existant
   - À partir d'un langage
3. Applications de la minimalité

Liens entre états et résiduels

On se donne un langage L reconnu par un AFD $\mathcal{A} = (A, Q, q_0, F, \delta)$.

Propriété

Si $q_0 \xrightarrow{u} q$, alors $\{w,\ \delta(q, w) \in F\} = u^{-1}L$.

On peut donc associer à chaque état q un résiduel à gauche de L.

Liens entre états et résiduels

On se donne un langage L reconnu par un AFD $\mathcal{A} = (A, Q, q_0, F, \delta)$.

Propriété

Si $q_0 \xrightarrow{u} q$, alors $\{w,\ \delta(q, w) \in F\} = u^{-1}L$.

On peut donc associer à chaque état q un résiduel à gauche de L.

Conséquence 1 : Le nombre de résiduels à gauche de L est fini.

Conséquence 2 : On peut définir l'automate des résiduels de L par

- $Q = \{u^{-1}L,\ u \in A^*\}$, Q est bien fini
- $I = \{\varepsilon^{-1}L\} = \{L\}$ et $F = \{q \in Q,\ \varepsilon \in q\}$,
- les transitions sont de la forme $u^{-1}L \xrightarrow{a} (ua)^{-1}L$.

Construction de l'automate des résiduels

Algorithme 1 : Construction de l'automate des résiduels

Entrée : un langage L sur un langage A, supposé reconnaissable

Sortie : $\mathcal{A} =$ automate des résiduels de L

1 créer un état 0 auquel on associe $L_0 = L$
2 $Q \leftarrow \{0\}, \quad W \leftarrow \{0\}, \quad E \leftarrow \emptyset$
3 **tant que** $W \neq \emptyset$ **faire**
4 choisir i dans W et le retirer de W
5 **pour chaque** $a \in A$ **faire**
6 **si** *il existe* j *tel que* $L_j = a^{-1}L_i$ **alors** ajouter (i, a, j) à E
7 **sinon**
8 ajouter un état k à Q et W, et lui associer $L_k = a^{-1}L_i$
9 ajouter la transition (i, a, k) à E
10 $F \leftarrow \{i,\ \varepsilon \in L_i\}$
11 **retourner** $(A, Q, \{0\}, E, F)$

Automates des résiduels – Exemple.

On considère $L = \{a^n b^m,\ m, n \in \mathbb{N}\}$.

1. on part de $L_0 = L$

Automates des résiduels – Exemple.

On considère $L = \{a^n b^m,\ m, n \in \mathbb{N}\}$.

1. on part de $L_0 = L$
2. pour $i = 0$: $\quad a^{-1}L_0 = L_0 \quad$ et $\quad b^{-1}L_0 = \{b^m,\ m \in \mathbb{N}\} = L_1$

Automates des résiduels – Exemple.

On considère $L = \{a^n b^m,\ m, n \in \mathbb{N}\}$.

1. on part de $L_0 = L$
2. pour $i = 0$: $a^{-1}L_0 = L_0$ et $b^{-1}L_0 = \{b^m,\ m \in \mathbb{N}\} = L_1$
3. pour $i = 1$: $a^{-1}L_1 = \emptyset = L_2$ et $b^{-1}L_1 = L_1$

Automates des résiduels – Exemple.

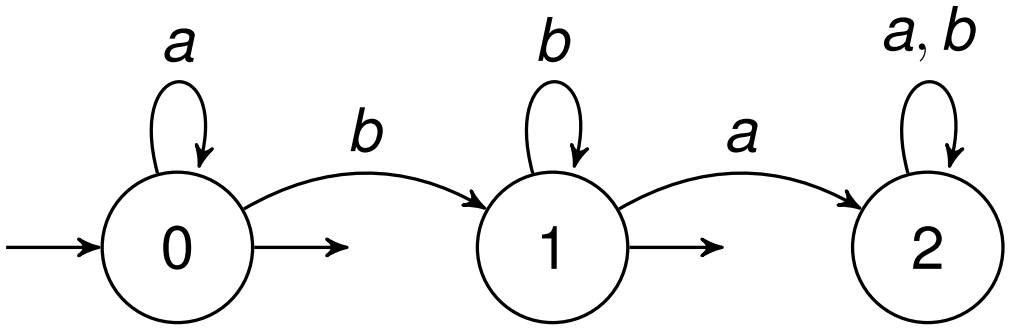
On considère $L = \{a^n b^m,\ m, n \in \mathbb{N}\}$.

1. on part de $L_0 = L$
2. pour $i = 0$: $a^{-1}L_0 = L_0$ et $b^{-1}L_0 = \{b^m,\ m \in \mathbb{N}\} = L_1$
3. pour $i = 1$: $a^{-1}L_1 = \emptyset = L_2$ et $b^{-1}L_1 = L_1$
4. pour $i = 2$: $a^{-1}L_2 = b^{-1}L_2 = L_2$

Automates des résiduels – Exemple.

On considère $L = \{a^n b^m,\ m, n \in \mathbb{N}\}$.

1. on part de $L_0 = L$
2. pour $i = 0$: $a^{-1}L_0 = L_0$ et $b^{-1}L_0 = \{b^m,\ m \in \mathbb{N}\} = L_1$
3. pour $i = 1$: $a^{-1}L_1 = \emptyset = L_2$ et $b^{-1}L_1 = L_1$
4. pour $i = 2$: $a^{-1}L_2 = b^{-1}L_2 = L_2$



Propriétés de l'automate des résiduels

Propriété

L'automate des résiduels de L est fini, déterministe, complet et minimal (modulo l'éventuel état puits).

Propriétés de l'automate des résiduels

Propriété

L'automate des résiduels de L est fini, déterministe, complet et minimal (modulo l'éventuel état puits).

Théorème

Tout automate fini déterministe complet minimal reconnaissant L est structurellement équivalent à l'automate des résiduels de L.

$\rightsquigarrow$ unicité de l'automate minimal modulo renommage des états

Preuve :
Si p correspond à $u^{-1}L$ et q à $v^{-1}L$, alors $p \sim q \Leftrightarrow u^{-1}L = v^{-1}L$.

Plan

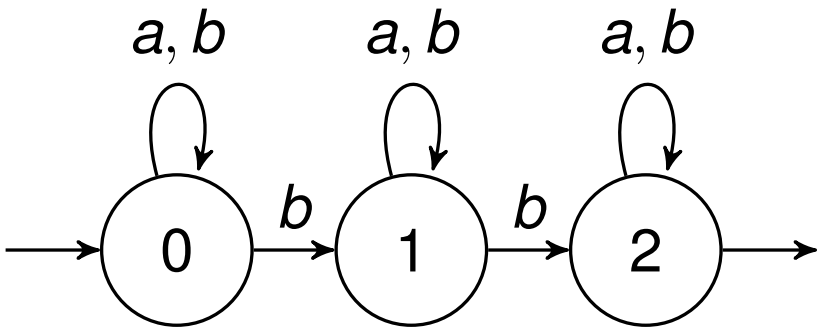
1. Minimalité d'un automate fini déterministe
2. Procédés de minimalisation
   - À partir d'un automate existant
   - À partir d'un langage
3. Applications de la minimalité

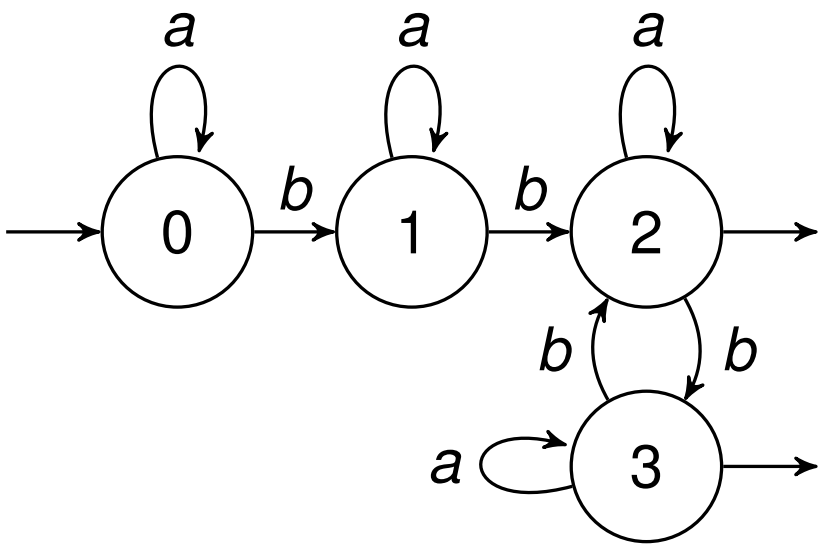
Égalité entre deux langages reconnaissables

Test d'égalité entre deux langages L_1 et L_2 reconnus par $\mathcal{A}_1$ et $\mathcal{A}_2$:

- on calcule l'AFD complet minimal $\mathcal{A}_1^M$ équivalent à $\mathcal{A}_1$,
- on calcule l'AFD complet minimal $\mathcal{A}_2^M$ équivalent à $\mathcal{A}_2$,
- on teste l'égalité entre $\mathcal{A}_1^M$ et $\mathcal{A}_2^M$.

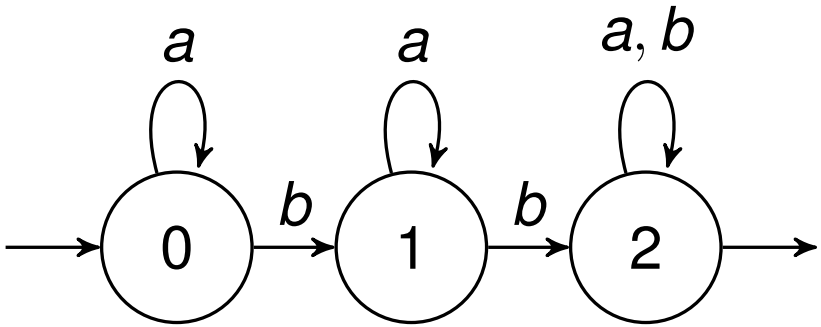
Exemple :

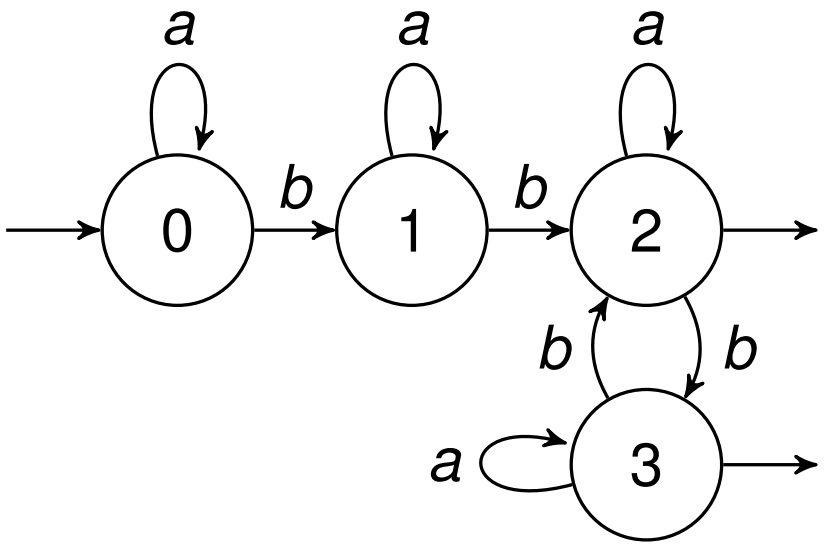




Détail de l'exemple

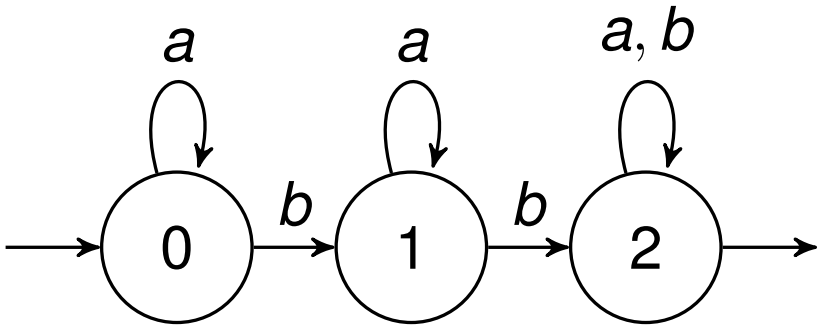
Après déterminisation/complétion :

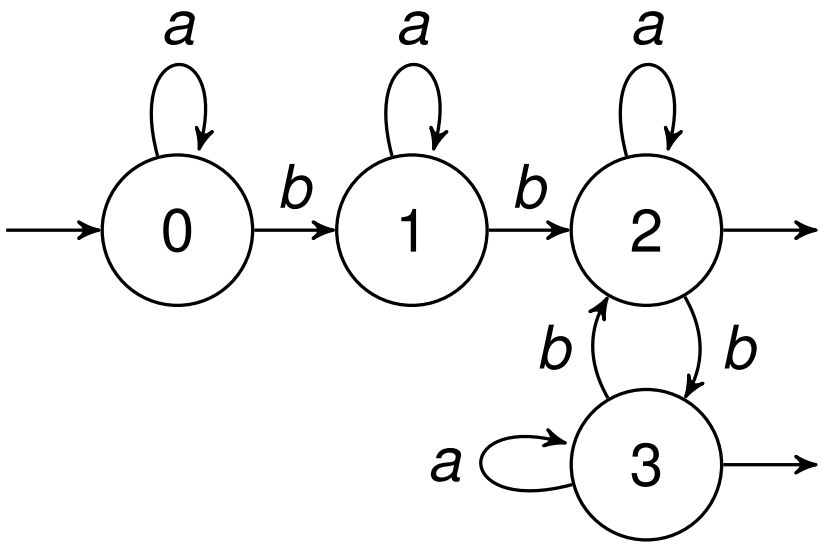




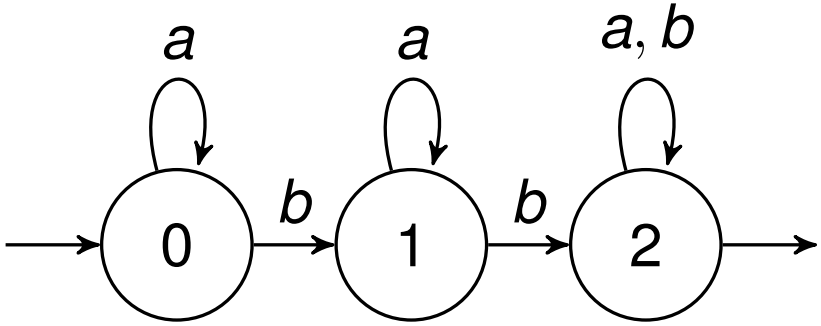
Détail de l'exemple

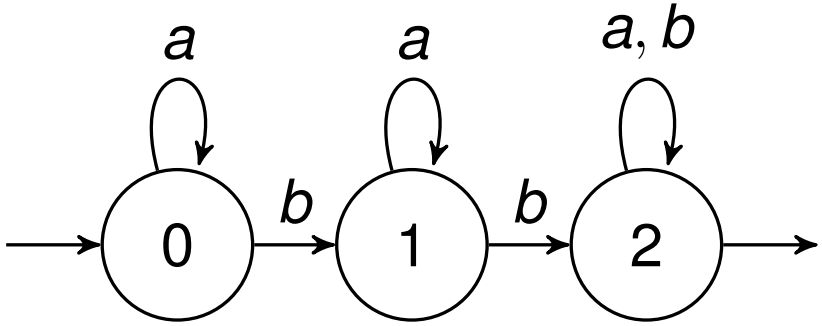
Après déterminisation/complétion :





Après minimisation : même automate, donc même langage reconnu





Équivalence entre deux expressions rationnelles

Test d'équivalence entre deux expressions rationnelles e_1 et e_2 :

- construire $\mathcal{A}_1$ et $\mathcal{A}_2$ à partir de e_1 et e_2 via Glushkov par ex.
- déterminiser et minimiser $\mathcal{A}_1 \quad \rightsquigarrow \quad \mathcal{A}_1^M$
- déterminiser et minimiser $\mathcal{A}_2 \quad \rightsquigarrow \quad \mathcal{A}_2^M$
- tester l'égalité entre $\mathcal{A}_1^M$ et $\mathcal{A}_2^M$

Exemple : $(a^*b)^*a^* \equiv (a+b)^*$.

$\rightsquigarrow$ À essayer chez vous !