LASF – Expressions rationnelles

Christophe Mouilleron

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

Plan

Plan

Définition

L'ensemble Expr(A) des expressions rationnelles sur l'alphabet A est défini de façon inductive par :

- $\emptyset \in \text{Expr}(A)$ et $\varepsilon \in \text{Expr}(A)$
- $a \in \text{Expr}(A)$ pour tout $a \in A$,
- si $e \in \text{Expr}(A)$, alors $e^* \in \text{Expr}(A)$,
- si $e_1, e_2 \in \text{Expr}(A)$, alors $e_1 + e_2 \in \text{Expr}(A)$,
- si $e_1, e_2 \in \text{Expr}(A)$, alors $e_1 \cdot e_2 \in \text{Expr}(A)$.

Exemples :

- $a + b^*$
- $(a + b)^*$ parenthèses nécessaires
- $a^*(ab + ba)b^*$ · omis

Langage associé à une expression rationnelle

À chaque expression rationnelle e correspond un langage dans A^*, qu'on note $[\![e]\!]$:

- $[\![\emptyset]\!] = \emptyset$ et $[\![\varepsilon]\!] = \{\varepsilon\}$,
- $[\![a]\!] = \{a\}$ pour tout $a \in A$,
- $[\![e^*]\!] = [\![e]\!]^*$,
- $[\![e_1 + e_2]\!] = [\![e_1]\!] \cup [\![e_2]\!]$,
- $[\![e_1 \cdot e_2]\!] = [\![e_1]\!] \cdot [\![e_2]\!]$.

Exemples :

- $[\![a^* b^*]\!] = ?$
- $[\![(a+b)^*]\!] = ?$
- $[\![(a+b)^* aaa]\!] = ?$

Langage associé à une expression rationnelle

À chaque expression rationnelle e correspond un langage dans A^*, qu'on note $[\![e]\!]$:

- $[\![\emptyset]\!] = \emptyset$ et $[\![\varepsilon]\!] = \{\varepsilon\}$,
- $[\![a]\!] = \{a\}$ pour tout $a \in A$,
- $[\![e^*]\!] = [\![e]\!]^*$,
- $[\![e_1 + e_2]\!] = [\![e_1]\!] \cup [\![e_2]\!]$,
- $[\![e_1 \cdot e_2]\!] = [\![e_1]\!] \cdot [\![e_2]\!]$.

Exemples :

- $[\![a^*b^*]\!] = \{a^m b^n \mid m, n \in \mathbb{N}\}$
- $[\![(a+b)^*]\!] = A^*$
- $[\![(a+b)^*aaa]\!] = \{uaaa, u \in A^*\}$

ATTENTION !

Ne pas confondre expression rationnelle et langage !

Les « langages » $(a+b)a^*$ et $(b+a)a^*$ sont égaux :
$\rightsquigarrow$ abus de notation pour $\{xa^n \mid x \in \{a,b\} \text{ et } n \in \mathbb{N}\}$.

Les expressions rationnelles $(a+b)a^*$ et $(b+a)a^*$ sont différentes :
$\rightsquigarrow$ différence syntaxique, même si le langage associé est le même.

Symboles supplémentaires

Notations pour alléger les expressions rationnelles :

- a^2 pour aa
- a^n pour $a \dots a$ (n fois la lettre a) pour $n > 1$
- a^+ pour au moins un a $a^+ = aa^*$
- A pour n'importe quelle lettre de A
- $e?$ pour 0 ou 1 fois e $e? = \varepsilon + e$

Exemples :

- $A^* a^3$?
- $(b + b^2)?a^+$?
- $(A^3)^*$?

Symboles supplémentaires

Notations pour alléger les expressions rationnelles :

- a^2 pour aa
- a^n pour $a \dots a$ (n fois la lettre a) — pour $n > 1$
- a^+ pour au moins un a — $a^+ = aa^*$
- A pour n'importe quelle lettre de A
- $e?$ pour 0 ou 1 fois e — $e? = \varepsilon + e$

Exemples :

- A^*a^3 — mots finissant par trois a
- $(b + b^2)?a^+$ — mots $b^m a^n$ avec $m \leq 2$ et $n \geq 1$
- $(A^3)^*$ — mots de longueur multiple de 3

Équivalence entre expressions rationnelles

Définition

Deux expressions rationnelles e_1 et e_2 sont dites équivalentes lorsque $[\![e_1]\!] = [\![e_2]\!]$. On note alors $e_1 \equiv e_2$.

Propriétés basiques :

- $\emptyset + e \equiv e + \emptyset \equiv e$
- $\emptyset \cdot e \equiv e \cdot \emptyset \equiv \emptyset$
- $\varepsilon \cdot e \equiv e \cdot \varepsilon \equiv e$
- $e_1 \cdot (e_2 + e_3) \equiv e_1 \cdot e_2 + e_1 \cdot e_3$
- $(e_1 + e_2) \cdot e_3 \equiv e_1 \cdot e_3 + e_2 \cdot e_3$

Équivalence entre expressions rationnelles (suite)

Autres propriétés :

- $e + e \equiv ?$

Équivalence entre expressions rationnelles (suite)

Autres propriétés :

- $e + e \equiv e$
- $(e^*)^* \equiv ?$

Équivalence entre expressions rationnelles (suite)

Autres propriétés :

- $e + e \equiv e$
- $(e^*)^* \equiv e^*$
- $(\varepsilon + e)^* \equiv ?$

Équivalence entre expressions rationnelles (suite)

Autres propriétés :

- $e + e \equiv e$
- $(e^*)^* \equiv e^*$
- $(\varepsilon + e)^* \equiv e^*$
- $e^* \equiv \varepsilon + e^+ \equiv \varepsilon + e^* e$
- $e_1(e_2 e_1)^* \equiv (e_1 e_2)^* e_1$
- $(e_1^* e_2)^* e_1^* \equiv e_2^*(e_1 e_2^*)^* \equiv (e_1 + e_2)^*$
- ...

Expressions régulières – Syntaxe de `grep`

Caractères spéciaux :

- `^` = début de ligne,
- `$` = fin de ligne,
- `.` = n'importe quelle lettre,
- `[]` = une lettre parmi l'ensemble/la plage,
- `()` = délimiteur,
- `|` = une des alternatives,
- `?` = 0 ou 1 fois ce qui précède,
- `+` = au moins une fois ce qui précède.

Exemples :

- `ch(ien|at)` — ligne contenant *chien* ou *chat*
- ? — ligne avec juste un nombre

Expressions régulières – Syntaxe de `grep`

Caractères spéciaux :

- `^` = début de ligne,
- `$` = fin de ligne,
- `.` = n'importe quelle lettre,
- `[]` = une lettre parmi l'ensemble/la plage,
- `()` = délimiteur,
- `|` = une des alternatives,
- `?` = 0 ou 1 fois ce qui précède,
- `+` = au moins une fois ce qui précède.

Exemples :

- `ch(ien|at)` ligne contenant *chien* ou *chat*
- `^(\+|-)?[0-9]+(\.[0-9]*)?$` ligne avec juste un nombre

Plan

Méthode naïve

On sait construire un automate qui reconnaît :

- $\emptyset = [\![\emptyset]\!]$,
- $\{\varepsilon\} = [\![\varepsilon]\!]$,
- $\{a\} = [\![a]\!]$.

vrai aussi pour $u \in A^*$

À partir de $\mathcal{A}_1$ reconnaissant $[\![e_1]\!]$ et $\mathcal{A}_2$ reconnaissant $[\![e_2]\!]$, on peut :

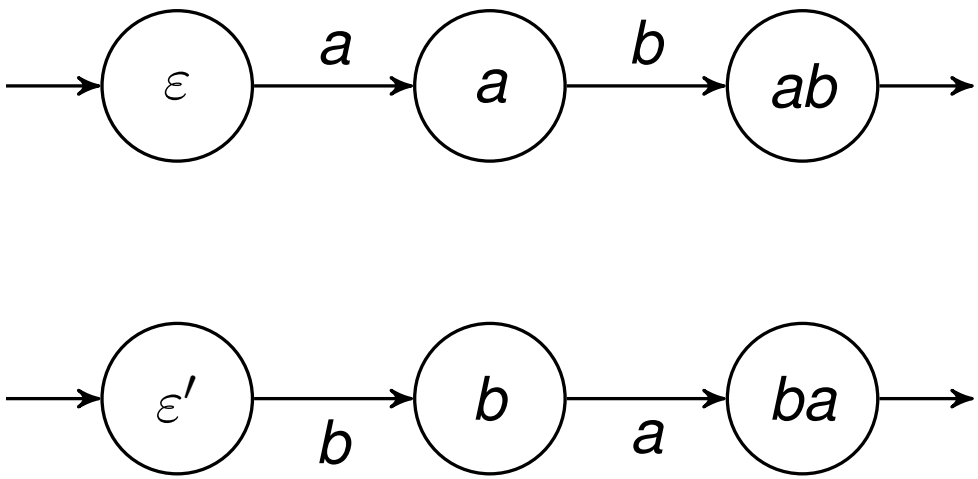
- construire $\mathcal{A}$ reconnaissant $[\![e_1]\!] \cup [\![e_2]\!] = [\![e_1 + e_2]\!]$,
- construire $\mathcal{A}$ reconnaissant $[\![e_1]\!] \cdot [\![e_2]\!] = [\![e_1 \cdot e_2]\!]$,
- construire $\mathcal{A}$ reconnaissant $[\![e_1]\!]^* = [\![e_1^*]\!]$.

$\rightsquigarrow$ Construction de façon inductive.

Méthode naïve – Exemple

Pour $e = (ab + ba)^*$:

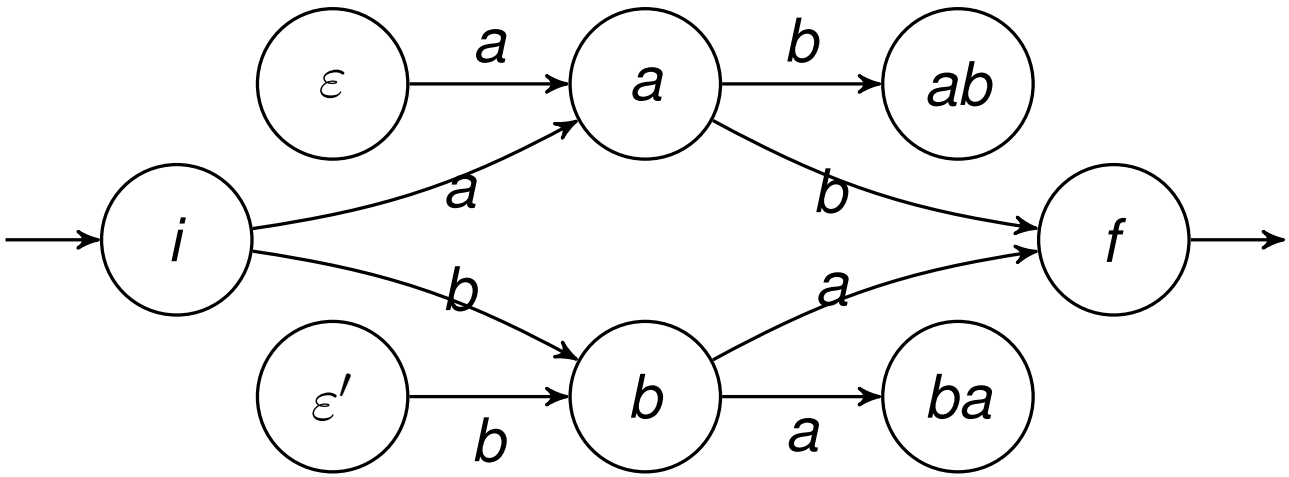
1. on construit deux automates reconnaissant $\{ab\}$ et $\{ba\}$,
2. on en déduit un automate correspondant à $ab + ba$,



Méthode naïve – Exemple

Pour $e = (ab + ba)^*$:

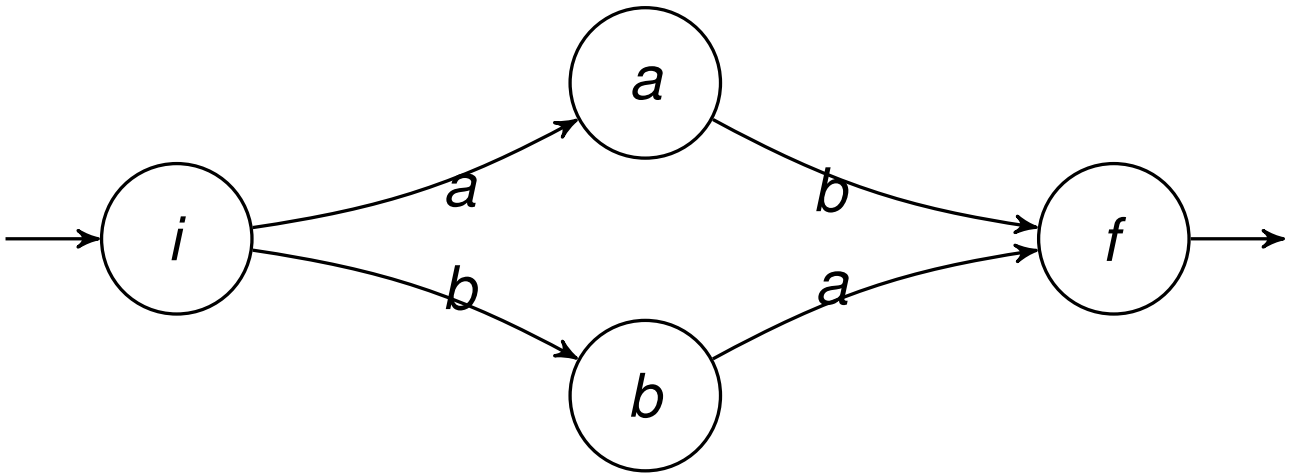
1. on construit deux automates reconnaissant $\{ab\}$ et $\{ba\}$,
2. on en déduit un automate correspondant à $ab + ba$,
3. on le normalise,



Méthode naïve – Exemple

Pour $e = (ab + ba)^*$:

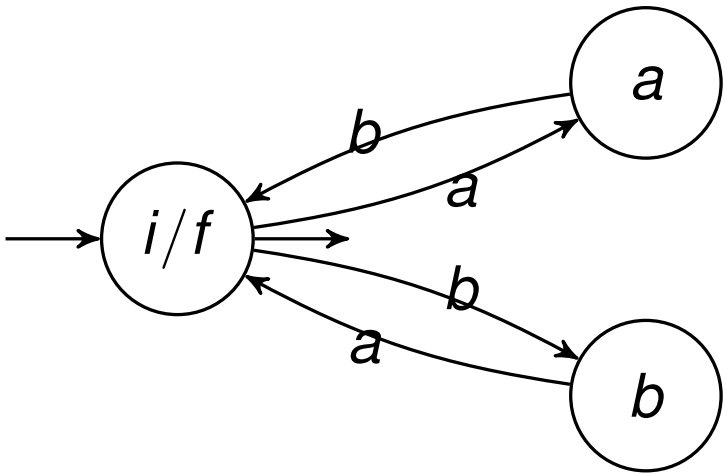
1. on construit deux automates reconnaissant $\{ab\}$ et $\{ba\}$,
2. on en déduit un automate correspondant à $ab + ba$,
3. on le normalise,
4. on émonde le résultat,



Méthode naïve – Exemple

Pour $e = (ab + ba)^*$:

1. on construit deux automates reconnaissant $\{ab\}$ et $\{ba\}$,
2. on en déduit un automate correspondant à $ab + ba$,
3. on le normalise,
4. on émonde le résultat,
5. on en déduit un automate correspondant à $(ab + ba)^*$.



Algorithme de Glushkov – Idée

Algorithme de Glushkov = procédé pour passer d'une expression rationnelle e à un automate fini.

Dans ce cours : présentation d'une version simplifiée.

Idées :

1. ajouter un numéro unique à chaque lettre de e,
2. associer un état i pour chaque lettre x_i dans e,
3. on est dans l'état i lorsqu'on vient de lire la lettre x_i,
4. on peut aller de l'état i à l'état j si et seulement si au moins un mot $u \in [\![e]\!]$ contient le facteur $x_i y_j$.

Algorithme de Glushkov

Algorithme 1 : `Glushkov`

Entrée : une expression rationnelle e
Sortie : un automate $\mathcal{A} = (A, Q, I, F, E)$ tel que $L(\mathcal{A}) = [\![e]\!]$

1 Numéroter de gauche à droite et de 1 à k les lettres dans e
`// pour` a^n, `numéroter` n `occurrences de` a
2 $A \leftarrow$ ensemble des lettres apparaissant dans e
3 $Q \leftarrow \{0, 1, \ldots, k\}$ `// on ajoute un état` 0
4 $I \leftarrow \{0\}$ `// 0 état initial`
5 $F \leftarrow \{$ indices des lettres les plus à droite dans e $\}$
6 **si** $\varepsilon \in [\![e]\!]$ **alors** $F \leftarrow F \cup \{0\}$
7 $E \leftarrow \emptyset$
8 **pour chaque** $x_i y_j$ *facteur d'un mot dans e numérotée* **faire**
9 $\quad E \leftarrow E \cup \{(i, y, j)\}$
10 **retourner** (A, Q, I, F, E)

Algorithme de Glushkov – Exemple

$e = (a + b)^*(a^2 + b^2(a + b)^*)$

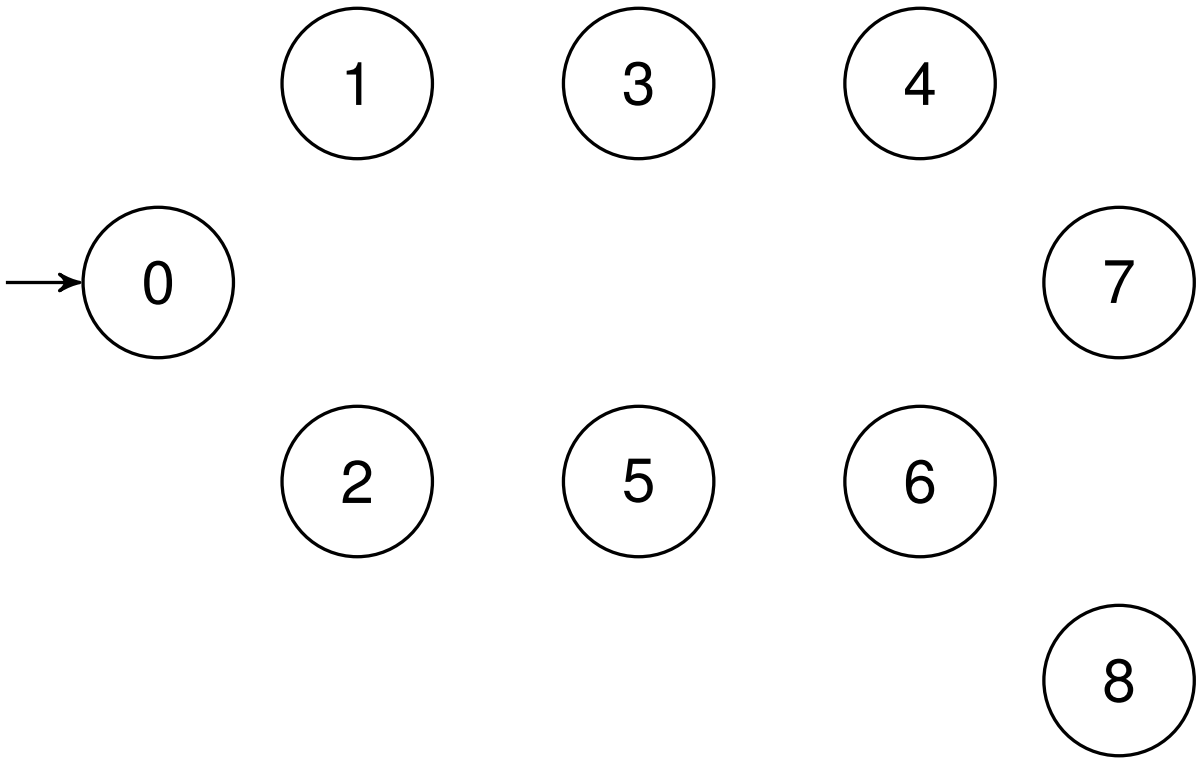
- $A := \{a, b\}$

Algorithme de Glushkov – Exemple

$e = (a+b)^*(a^2 + b^2(a+b)^*)$

$\rightsquigarrow e' = (a_1 + b_2)^*(a_3 a_4 + b_5 b_6(a_7 + b_8)^*)$

- $A := \{a, b\}$
- $Q := \{0, \ldots, 8\}$
- $I := \{0\}$

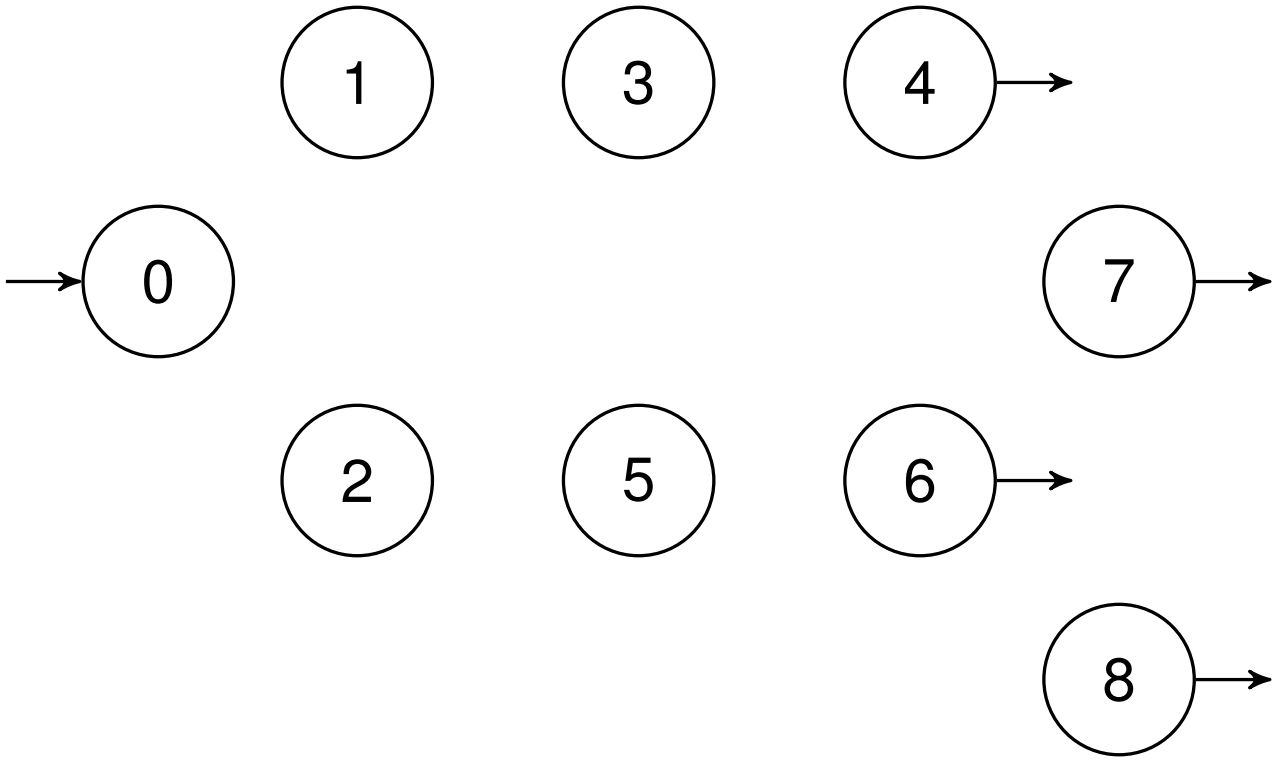


Algorithme de Glushkov – Exemple

$e = (a + b)^*(a^2 + b^2(a + b)^*)$

$\rightsquigarrow e' = (a_1 + b_2)^*(a_3 a_4 + b_5 b_6(a_7 + b_8)^*)$

- $A := \{a, b\}$
- $Q := \{0, \ldots, 8\}$
- $I := \{0\}$
- $F := \{4, 6, 7, 8\}$

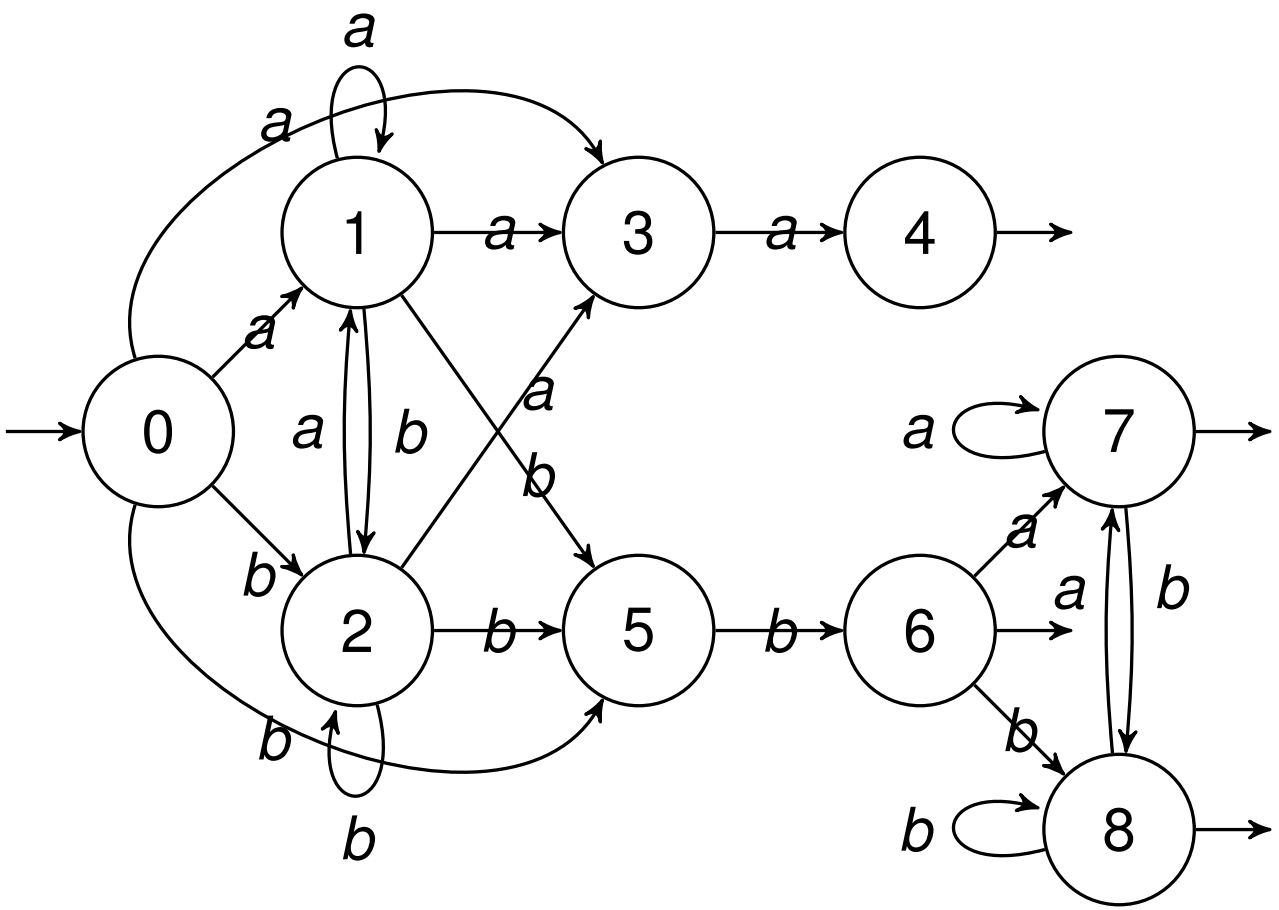


Algorithme de Glushkov – Exemple

$e = (a + b)^*(a^2 + b^2(a + b)^*)$

$\rightsquigarrow e' = (a_1 + b_2)^*(a_3 a_4 + b_5 b_6(a_7 + b_8)^*)$

- $A := \{a, b\}$
- $Q := \{0, \ldots, 8\}$
- $I := \{0\}$
- $F := \{4, 6, 7, 8\}$



Algorithme de Glushkov – Remarques

Penser à simplifier avant d'appliquer l'algorithme :

- utiliser A au lieu d'une alternative quand c'est possible,
- factoriser les lettres autant que possible !

$$e = (a+b)^*(a^2 + b^2(a+b)^*) \equiv A^*(a^2 + b^2A^*)$$

- $A := \{a, b\}$

Algorithme de Glushkov – Remarques

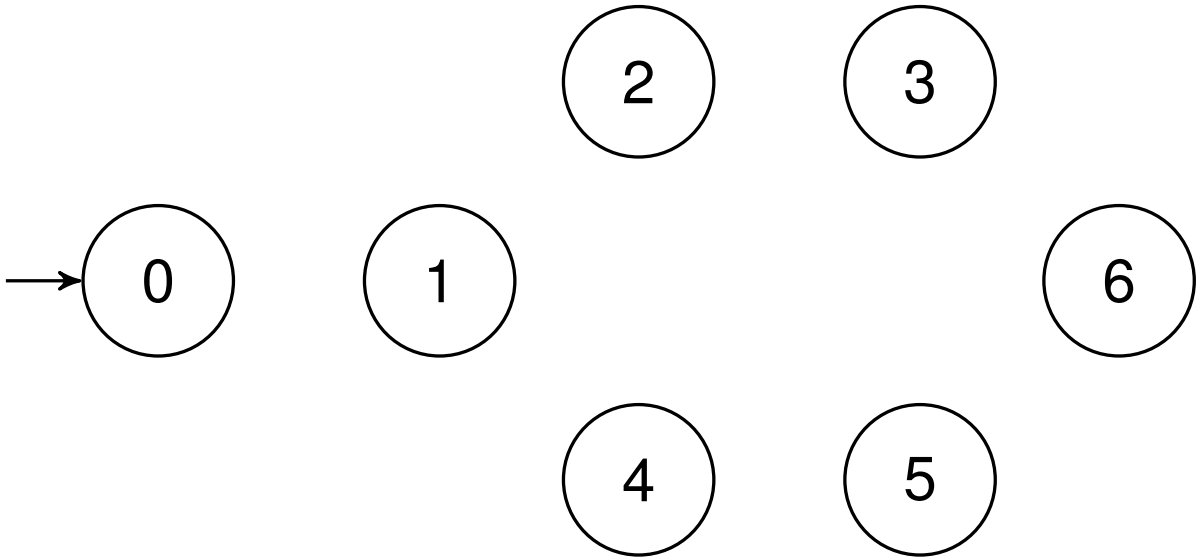
Penser à simplifier avant d'appliquer l'algorithme :

- utiliser A au lieu d'une alternative quand c'est possible,
- factoriser les lettres autant que possible !

$$e = (a+b)^*(a^2 + b^2(a+b)^*) \equiv A^*(a^2 + b^2A^*)$$

$$\rightsquigarrow e' = A_1^*(a_2a_3 + b_4b_5A_6^*)$$

- $A := \{a, b\}$
- $Q := \{0, \ldots, 6\}$
- $I := \{0\}$



Algorithme de Glushkov – Remarques

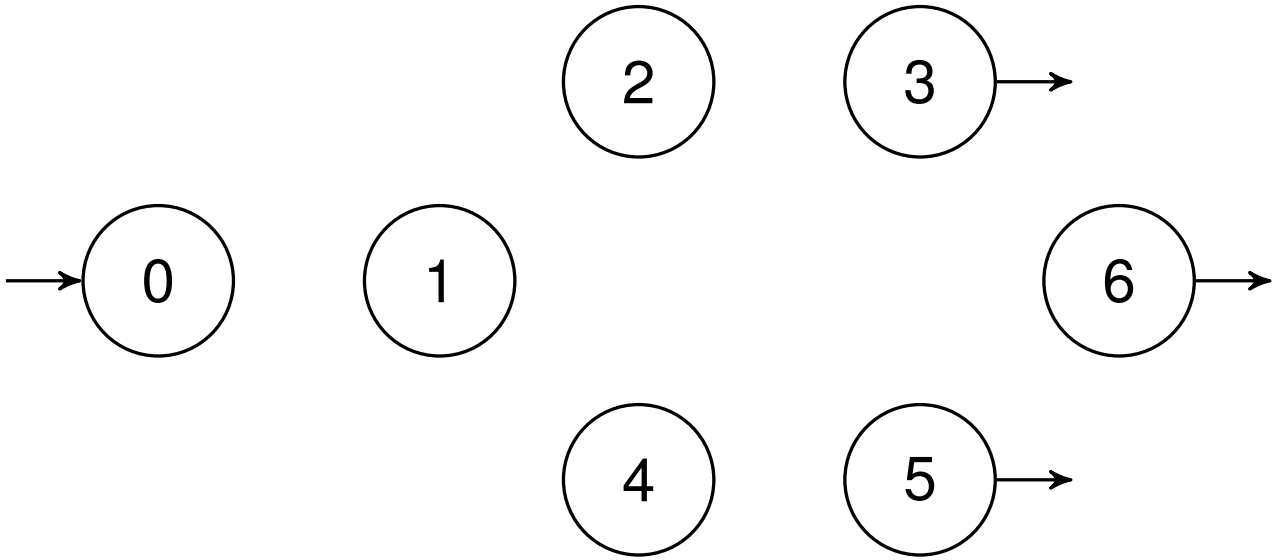
Penser à simplifier avant d'appliquer l'algorithme :

- utiliser A au lieu d'une alternative quand c'est possible,
- factoriser les lettres autant que possible !

$$e = (a+b)^*(a^2 + b^2(a+b)^*) \equiv A^*(a^2 + b^2A^*)$$

$$\rightsquigarrow e' = A_1^*(a_2a_3 + b_4b_5A_6^*)$$

- $A := \{a, b\}$
- $Q := \{0, \ldots, 6\}$
- $I := \{0\}$
- $F := \{3, 5, 6\}$



Algorithme de Glushkov – Remarques

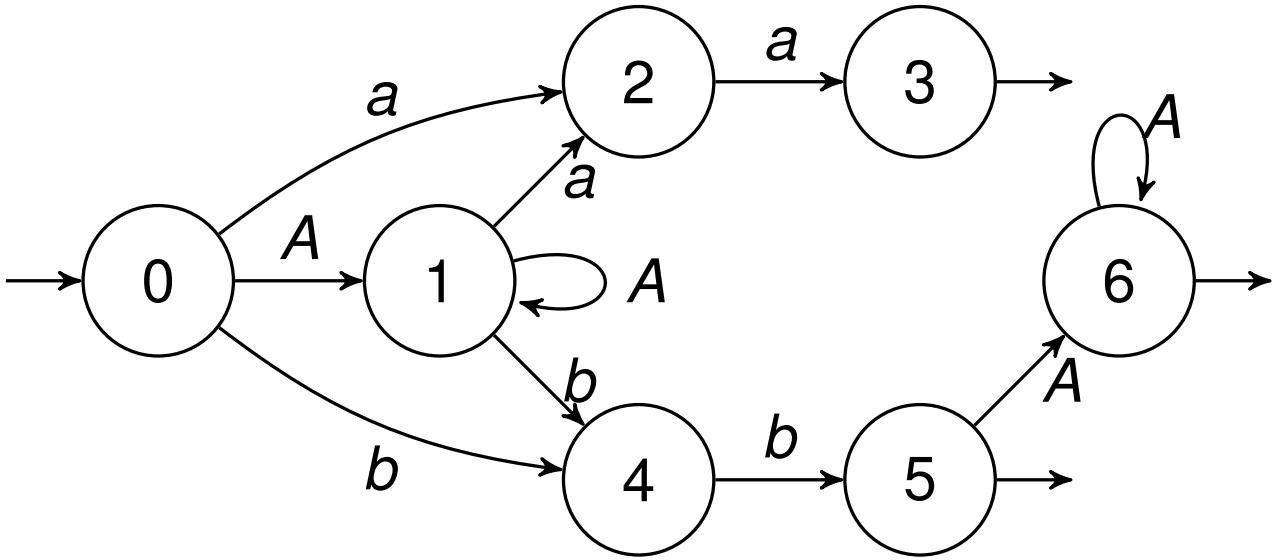
Penser à simplifier avant d'appliquer l'algorithme :

- utiliser A au lieu d'une alternative quand c'est possible,
- factoriser les lettres autant que possible !

$e = (a+b)^*(a^2 + b^2(a+b)^*) \equiv A^*(a^2 + b^2A^*)$

$\rightsquigarrow e' = A_1^*(a_2a_3 + b_4b_5A_6^*)$

- $A := \{a, b\}$
- $Q := \{0, \ldots, 6\}$
- $I := \{0\}$
- $F := \{3, 5, 6\}$



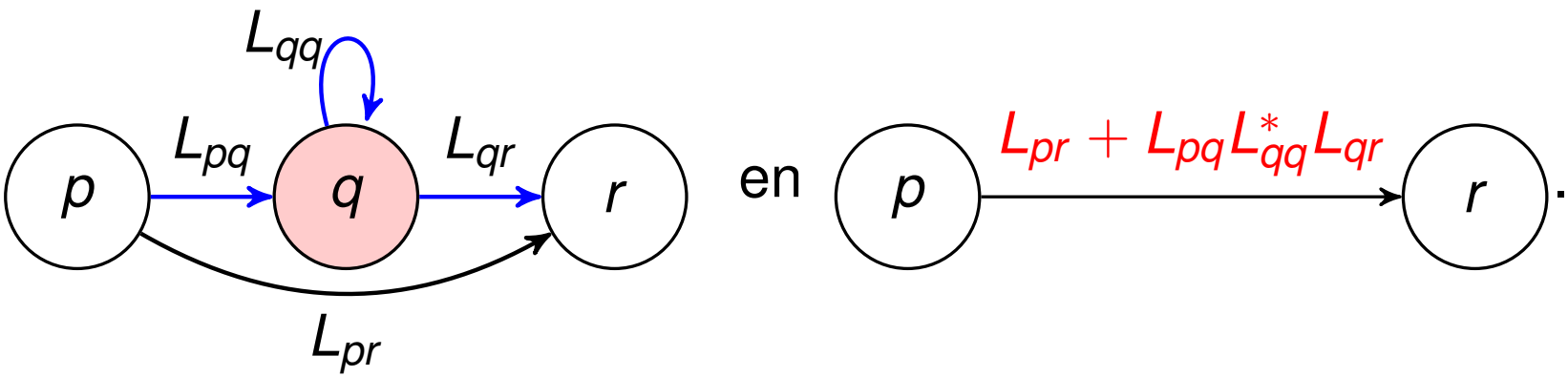
Plan

Méthode par élimination des états

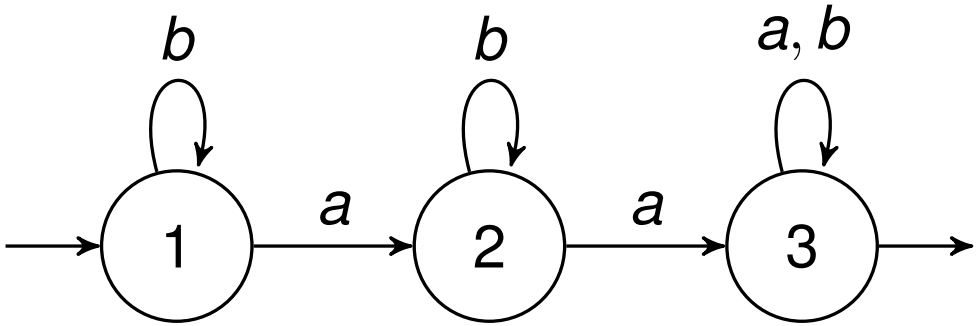
Idée :

1. étiqueter les transitions de l'automate avec des expressions rationnelles plutôt que des lettres,
2. ajouter un état i et une transition $i \xrightarrow{\varepsilon} p$ pour tout $p \in I$,
3. ajouter un état f et une transition $q \xrightarrow{\varepsilon} f$ pour tout $q \in F$,
4. supprimer les états de Q un à un en mettant à jour les transitions :

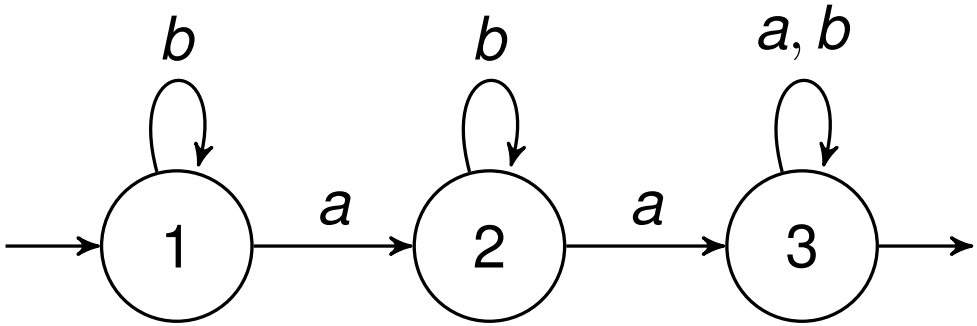
   Pour tout couple $(p, r) \in Q^2$ tel que $p \xrightarrow{L_{pq}} q \xrightarrow{L_{qr}} r$, changer

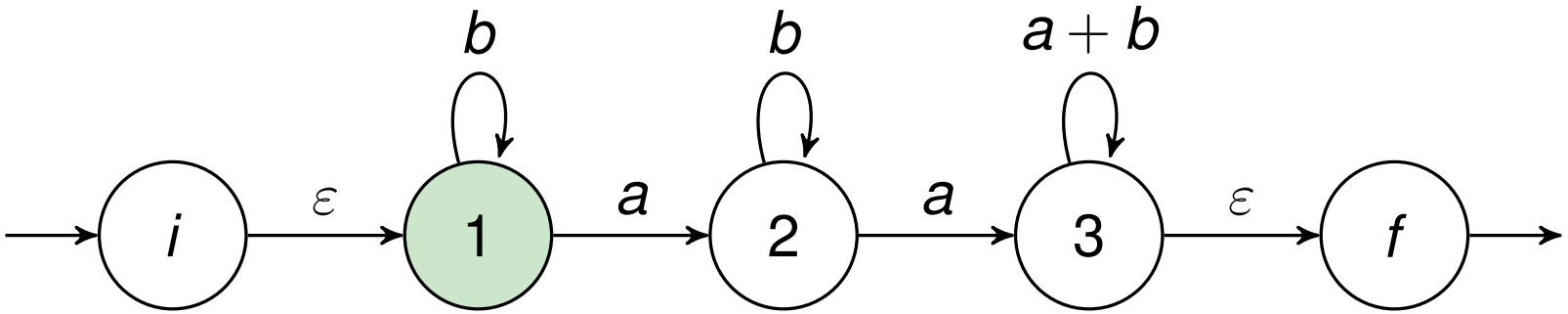


Méthode par élimination des états – Exemple

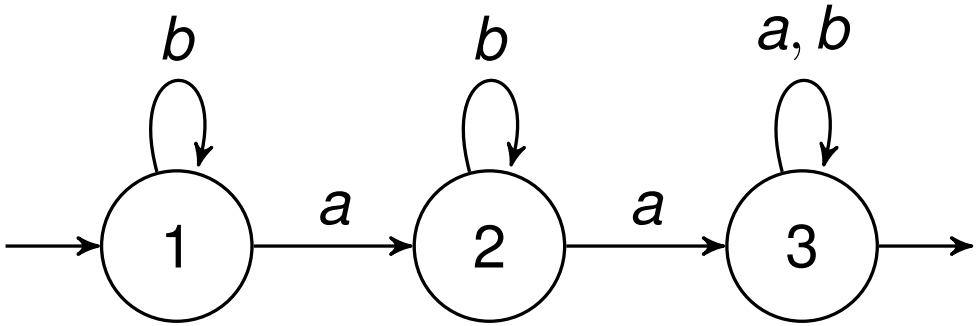


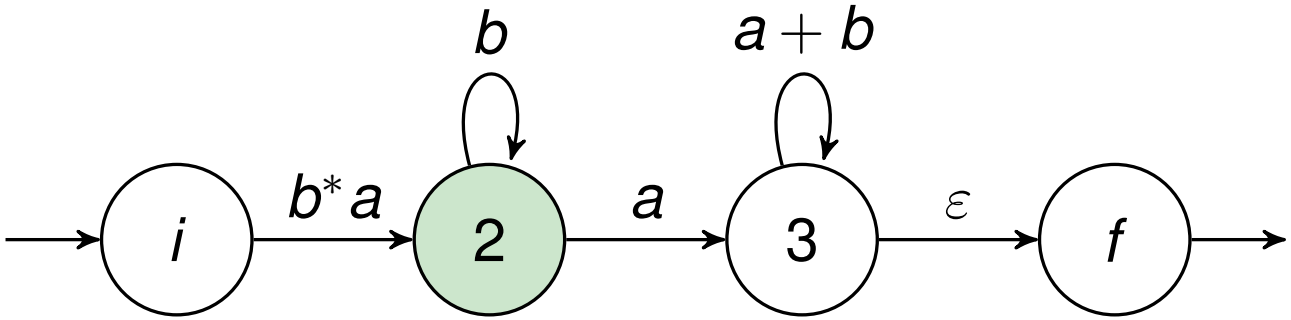
Méthode par élimination des états – Exemple



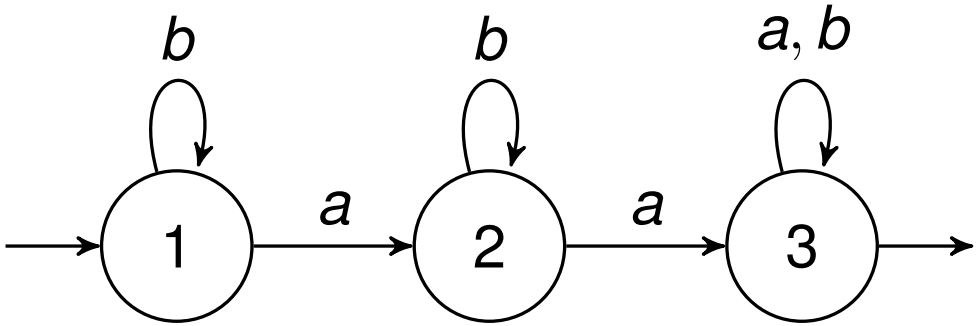


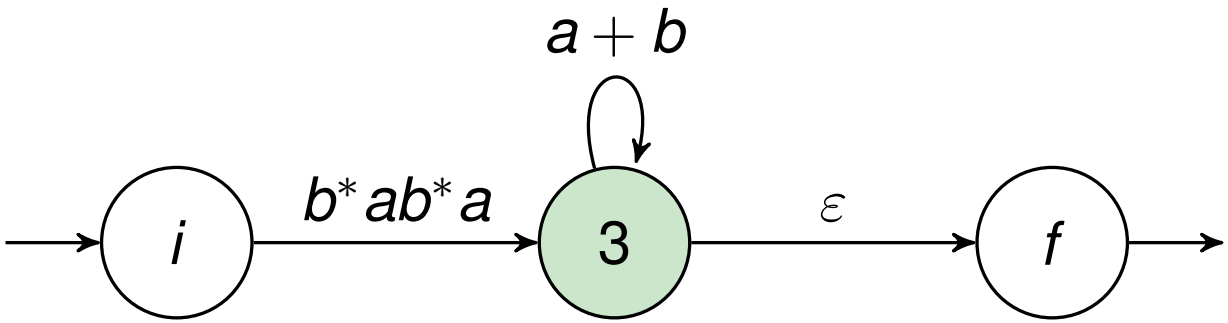
Méthode par élimination des états – Exemple



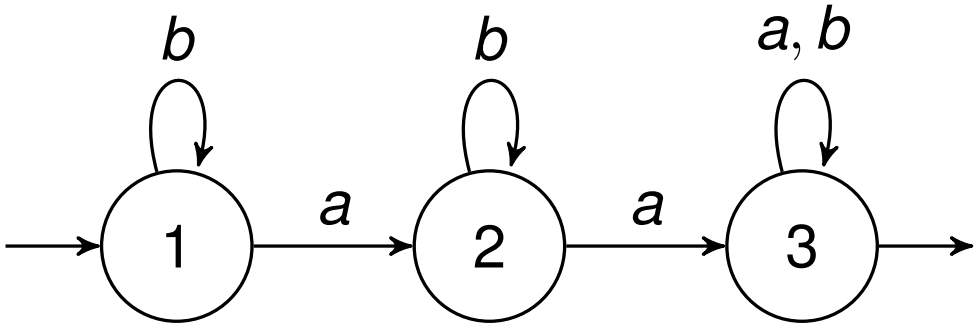


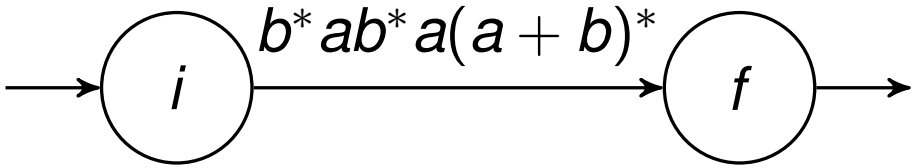
Méthode par élimination des états – Exemple





Méthode par élimination des états – Exemple





$$L(\mathcal{A}) = [\![b^*ab^*a(a+b)^*]\!]$$