

LASF – Expressions rationnelles

Christophe Moulleron



- 1 Expressions rationnelles
- 2 De l'expression rationnelle à l'automate fini
 - Méthode naïve
 - Algorithme de Glushkov
- 3 De l'automate fini à l'expression rationnelle/au langage
 - Méthode par élimination des états

- 1 Expressions rationnelles
- 2 De l'expression rationnelle à l'automate fini
 - Méthode naïve
 - Algorithme de Glushkov
- 3 De l'automate fini à l'expression rationnelle/au langage
 - Méthode par élimination des états

Définition

L'ensemble $\text{Expr}(A)$ 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 :

- $a + b^*$
- $(a + b)^*$
- $a^*(ab + ba)b^*$

parenthèses nécessaires

• omis

Langage associé à une expression rationnelle

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

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

Exemples :

- $\llbracket a^* b^* \rrbracket = ?$
- $\llbracket (a + b)^* \rrbracket = ?$
- $\llbracket (a + b)^* aaa \rrbracket = ?$

Langage associé à une expression rationnelle

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

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

Exemples :

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

ATTENTION!

Ne pas confondre **expression rationnelle** et **langage** !

Les « **langages** » $(a + b)a^*$ et $(b + a)a^*$ sont égaux :
↪ 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 :
↪ 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)
- a^+ pour au moins un a
- A pour n'importe quelle lettre de A
- $e?$ pour 0 ou 1 fois e

pour $n > 1$

$$a^+ = aa^*$$

$$e? = \varepsilon + e$$

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)
- a^+ pour au moins un a
- A pour n'importe quelle lettre de A
- $e?$ pour 0 ou 1 fois e

pour $n > 1$

$$a^+ = aa^*$$

$$e? = \varepsilon + e$$

Exemples :

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

mots finissant par trois a

mots $b^m a^n$ avec $m \leq 2$ et $n \geq 1$

mots de longueur multiple de 3

Équivalence entre expressions rationnelles

Définition

Deux expressions rationnelles e_1 et e_2 sont dites **équivalentes** lorsque $\llbracket e_1 \rrbracket = \llbracket e_2 \rrbracket$. On note alors $e_1 \equiv e_2$.

Propriétés basiques :

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

Autres propriétés :

- $e + e \equiv ?$

Autres propriétés :

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

Équivalence entre expressions rationnelles (suite)

Autres propriétés :

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

Autres propriétés :

- $e + e \equiv e$
- $(e^*)^* \equiv e^*$
- $(\varepsilon + e)^* \equiv e^*$
- $e^* \equiv \varepsilon + e^+ \equiv \varepsilon + e^*e$
- $e_1(e_2e_1)^* \equiv (e_1e_2)^*e_1$
- $(e_1^*e_2)^*e_1^* \equiv e_2^*(e_1e_2^*)^* \equiv (e_1 + e_2)^*$
- ...

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

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

- 1 Expressions rationnelles
- 2 De l'expression rationnelle à l'automate fini
 - Méthode naïve
 - Algorithme de Glushkov
- 3 De l'automate fini à l'expression rationnelle/au langage
 - Méthode par élimination des états

Méthode naïve

On sait construire un automate qui reconnaît :

- $\emptyset = \llbracket \emptyset \rrbracket$,
- $\{\varepsilon\} = \llbracket \varepsilon \rrbracket$,
- $\{a\} = \llbracket a \rrbracket$.

vrai aussi pour $u \in A^*$

À partir de \mathcal{A}_1 reconnaissant $\llbracket e_1 \rrbracket$ et \mathcal{A}_2 reconnaissant $\llbracket e_2 \rrbracket$, on peut :

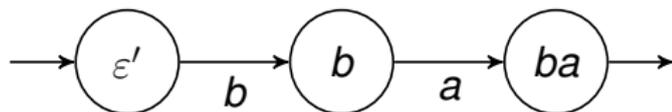
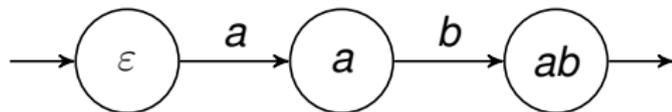
- construire \mathcal{A} reconnaissant $\llbracket e_1 \rrbracket \cup \llbracket e_2 \rrbracket = \llbracket e_1 + e_2 \rrbracket$,
- construire \mathcal{A} reconnaissant $\llbracket e_1 \rrbracket \cdot \llbracket e_2 \rrbracket = \llbracket e_1 \cdot e_2 \rrbracket$,
- construire \mathcal{A} reconnaissant $\llbracket e_1 \rrbracket^* = \llbracket e_1^* \rrbracket$.

↪ Construction de façon inductive.

Méthode naïve – Exemple

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

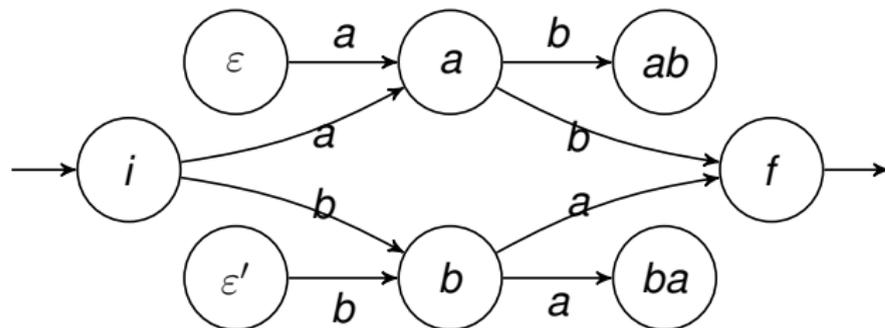
- 1 on construit deux automates reconnaissant $\{ab\}$ et $\{ba\}$,
- 2 on en déduit un automate correspondant à $ab + ba$,



Méthode naïve – Exemple

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

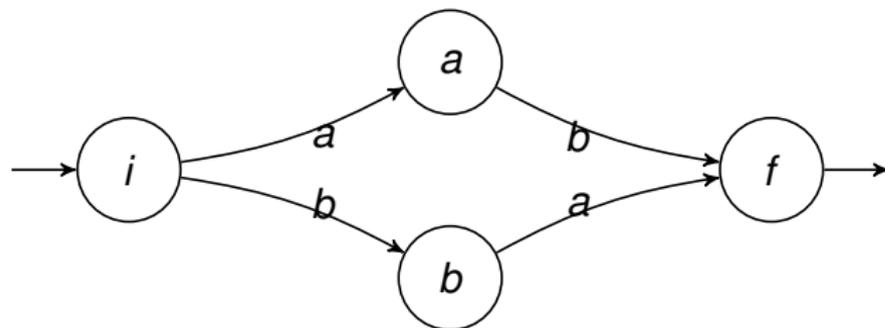
- 1 on construit deux automates reconnaissant $\{ab\}$ et $\{ba\}$,
- 2 on en déduit un automate correspondant à $ab + ba$,
- 3 on le normalise,



Méthode naïve – Exemple

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

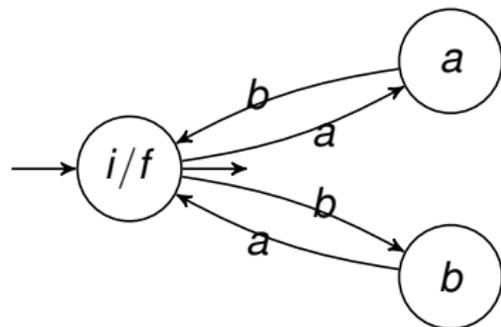
- 1 on construit deux automates reconnaissant $\{ab\}$ et $\{ba\}$,
- 2 on en déduit un automate correspondant à $ab + ba$,
- 3 on le normalise,
- 4 on émonde le résultat,



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 $u \in \llbracket e \rrbracket$ contient le facteur $x_i y_j$.

Algorithme 1 : Glushkov

Entrée : une expression rationnelle e

Sortie : un automate $\mathcal{A} = (A, Q, I, F, E)$ tel que $L(\mathcal{A}) = \llbracket e \rrbracket$

- 1 Numéroté de gauche à droite et de 1 à k les lettres dans e
// pour a^n , numéroté n occurrences de a
 - 2 $A \leftarrow$ ensemble des lettres apparaissant dans e
 - 3 $Q \leftarrow \{0, 1, \dots, k\}$ // 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** $\varepsilon \in \llbracket e \rrbracket$ **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 $E \leftarrow E \cup \{(i, y, j)\}$
 - 10 **retourner** (A, Q, I, F, E)
-

Algorithme de Glushkov – Exemple

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

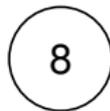
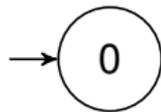
- $A := \{a, b\}$

Algorithme de Glushkov – Exemple

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

$$\rightsquigarrow e' = (a_1 + b_2)^*(a_3a_4 + b_5b_6(a_7 + b_8)^*)$$

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

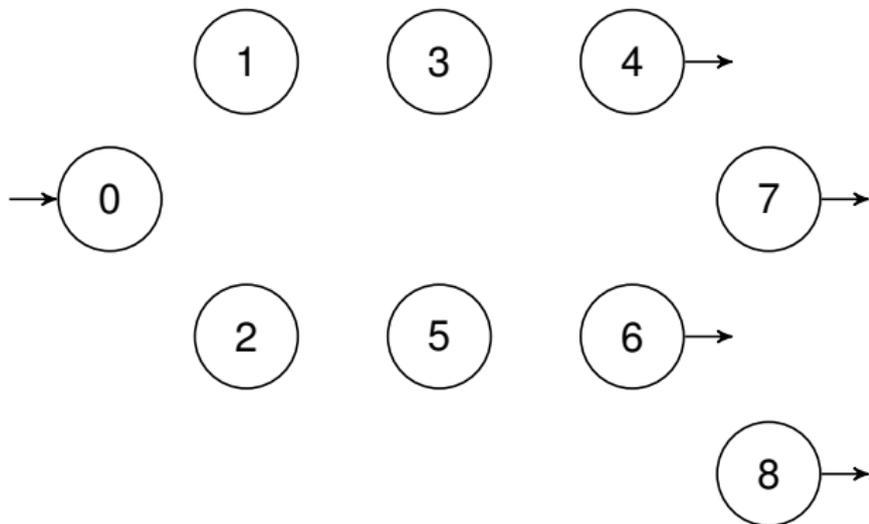


Algorithme de Glushkov – Exemple

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

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

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

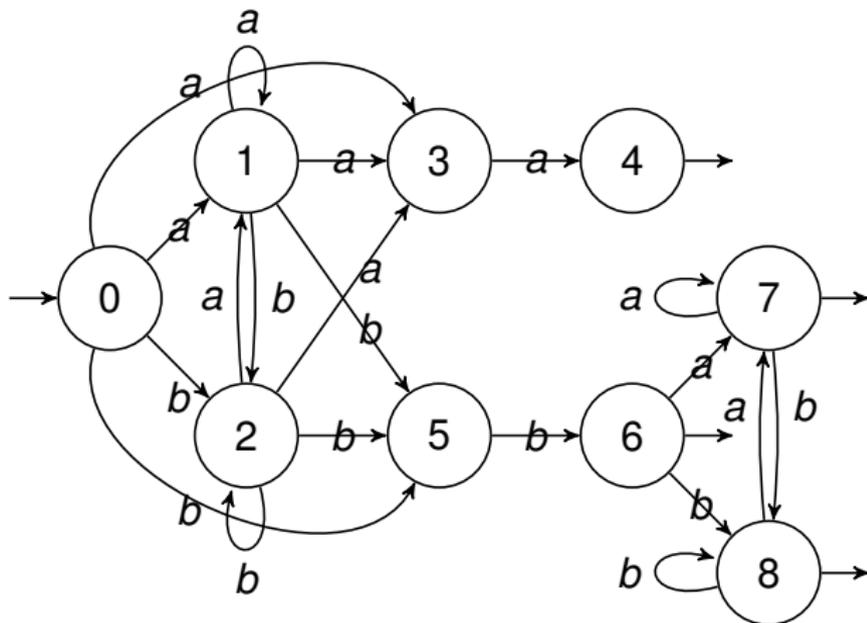


Algorithme de Glushkov – Exemple

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

$$\rightsquigarrow e' = (a_1 + b_2)^*(a_3a_4 + b_5b_6(a_7 + b_8)^*)$$

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



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 !

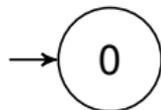
$$e = (a + b)^*(a^2 + b^2(a + b)^*) \equiv A^*(a^2 + b^2A^*)$$

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

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

- $Q := \{0, \dots, 6\}$

- $I := \{0\}$



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 !

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

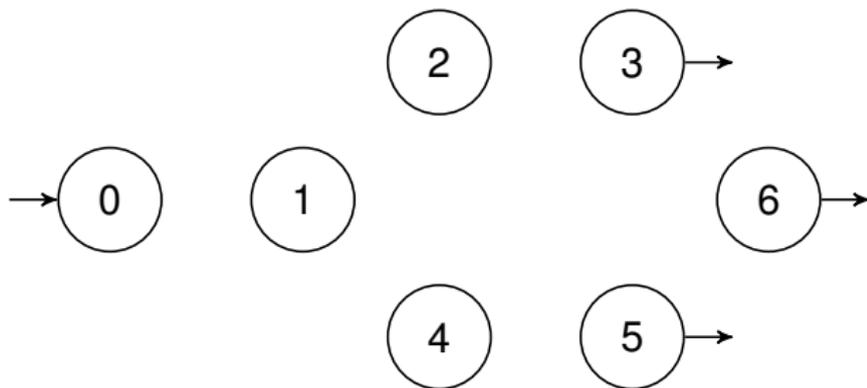
$$\rightsquigarrow e' = A_1^*(a_2a_3 + b_4b_5A_6^*)$$

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

- $Q := \{0, \dots, 6\}$

- $I := \{0\}$

- $F := \{3, 5, 6\}$



Algorithme de Glushkov – Remarques

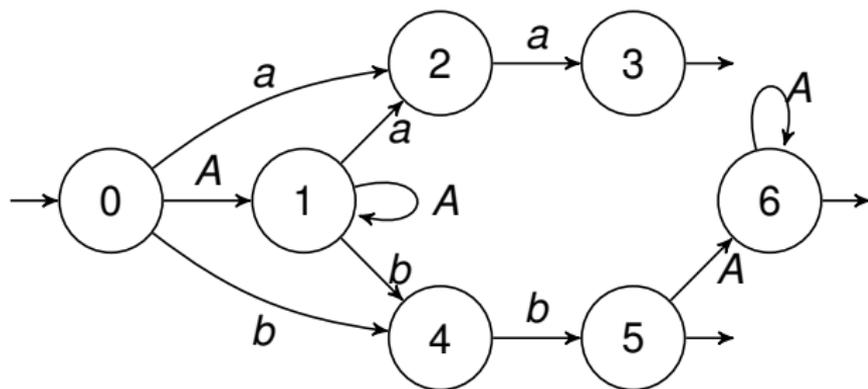
Penser à **simplifier** avant d'appliquer l'algorithme :

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

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

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

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



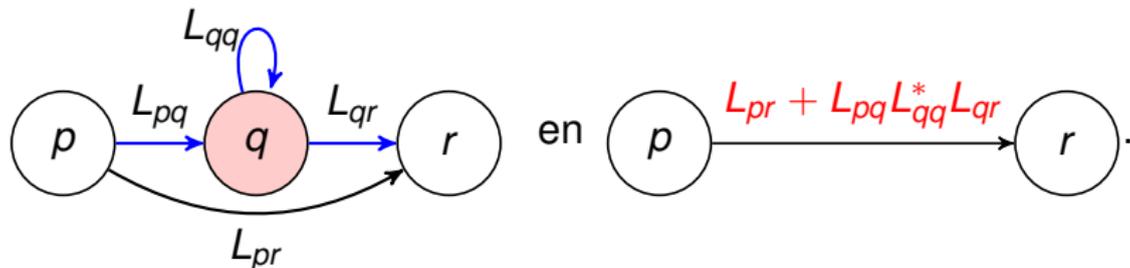
- 1 Expressions rationnelles
- 2 De l'expression rationnelle à l'automate fini
 - Méthode naïve
 - Algorithme de Glushkov
- 3 De l'automate fini à l'expression rationnelle/au langage
 - Méthode par élimination des états

Méthode par élimination des états

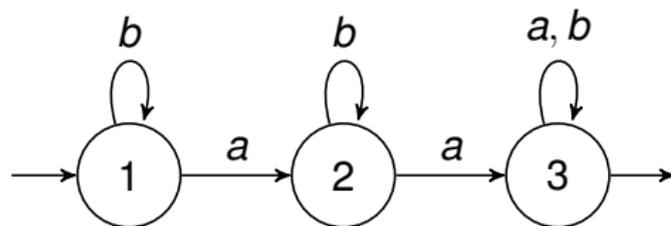
Idée :

- 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 :

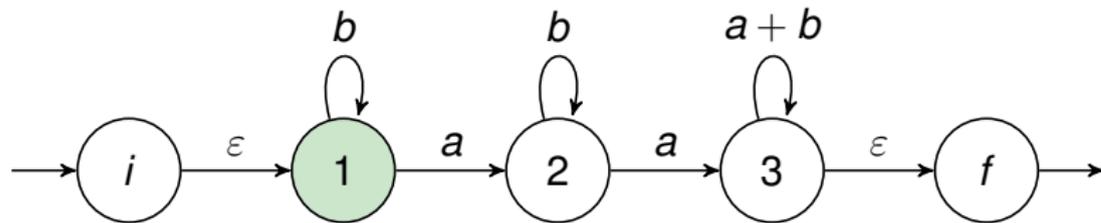
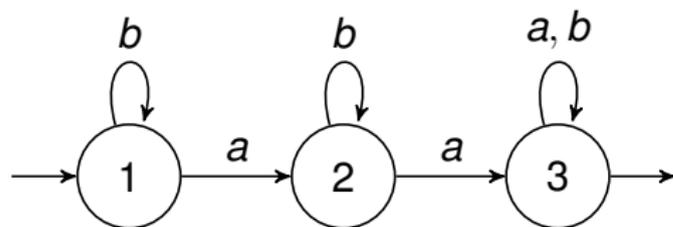
Pour tout couple $(p, r) \in Q^2$ tel que $p \xrightarrow{L_{pq}} q \xrightarrow{L_{qr}} r$, changer



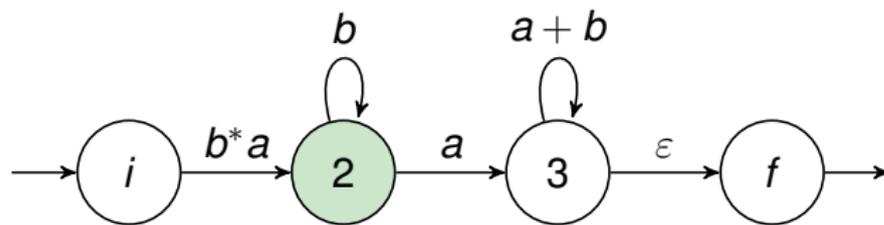
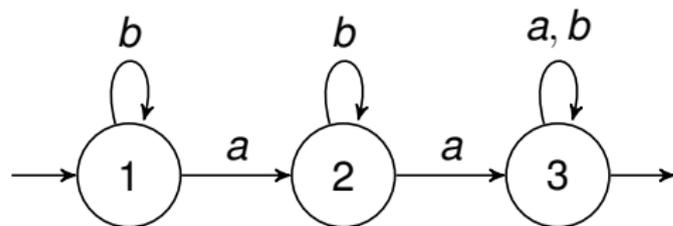
Méthode par élimination des états – Exemple



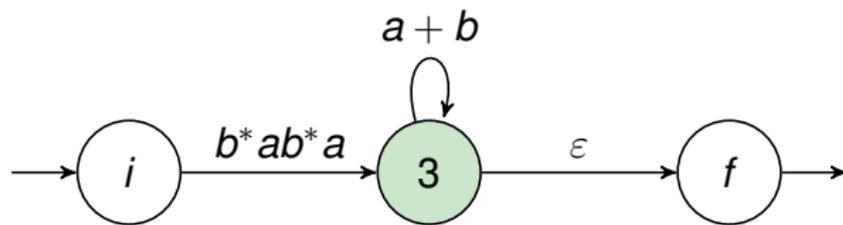
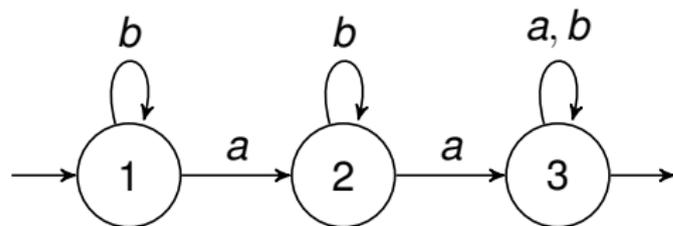
Méthode par élimination des états – Exemple



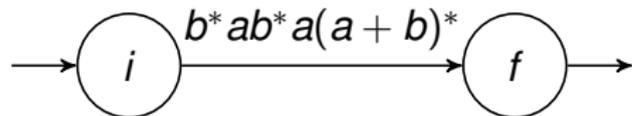
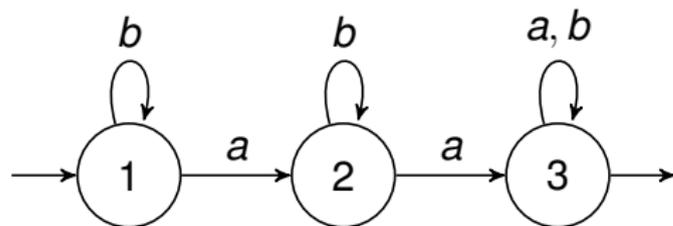
Méthode par élimination des états – Exemple



Méthode par élimination des états – Exemple



Méthode par élimination des états – Exemple



$$L(\mathcal{A}) = \llbracket b^*ab^* a(a+b)^* \rrbracket$$