

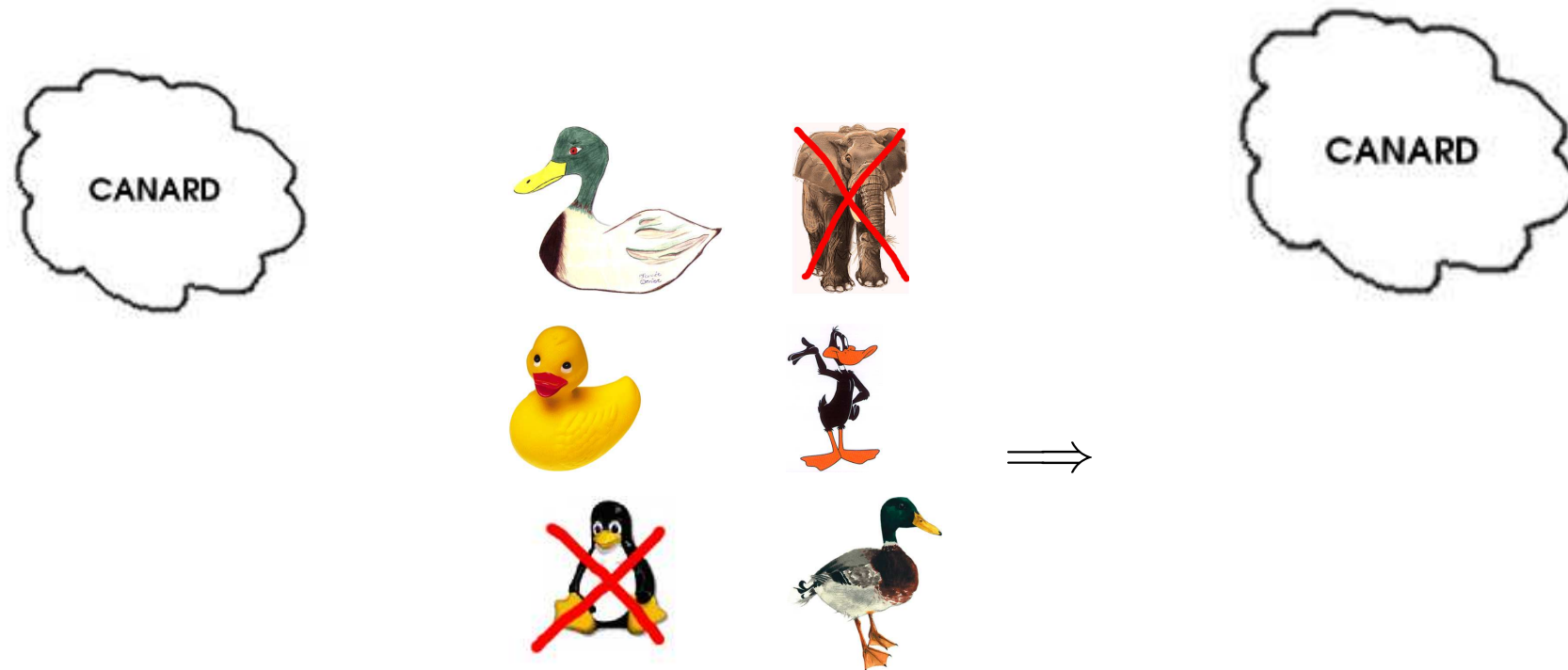
---

# Inférence grammaticale

A. Lemay

GRAPPA - Lille 3

# Apprentissage Automatique



## Inférence grammaticale :



The sky is blue.

The cat eats the mouse.

The grass is green.

~~Le chat regarde la couris.~~

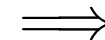
Your flowers are beautiful.

John loves Mary.

My tailor is rich.

~~Ich bin fröh.~~

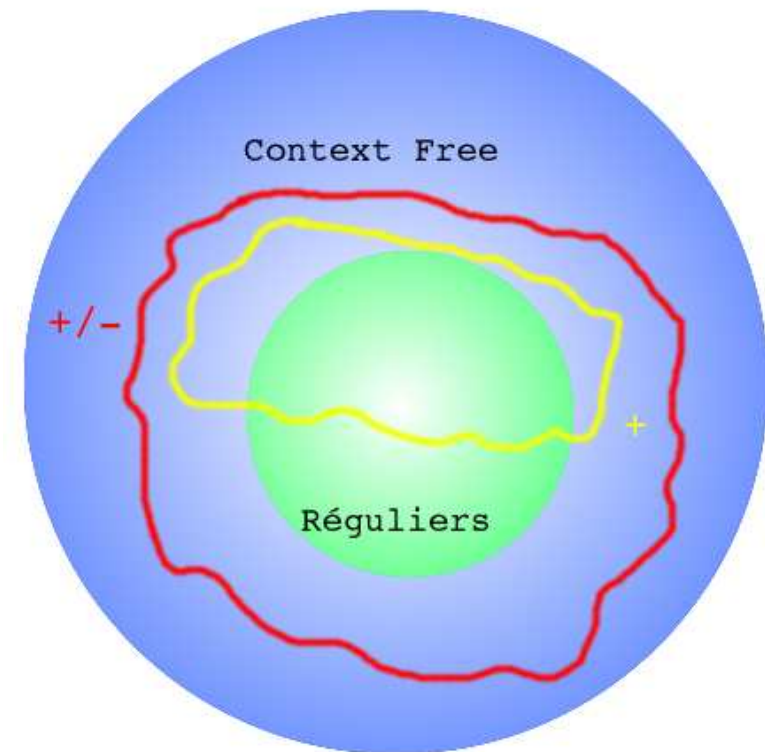
John runs.



Classes de langages identifiables efficacement :

## Hiérarchie de Chomsky

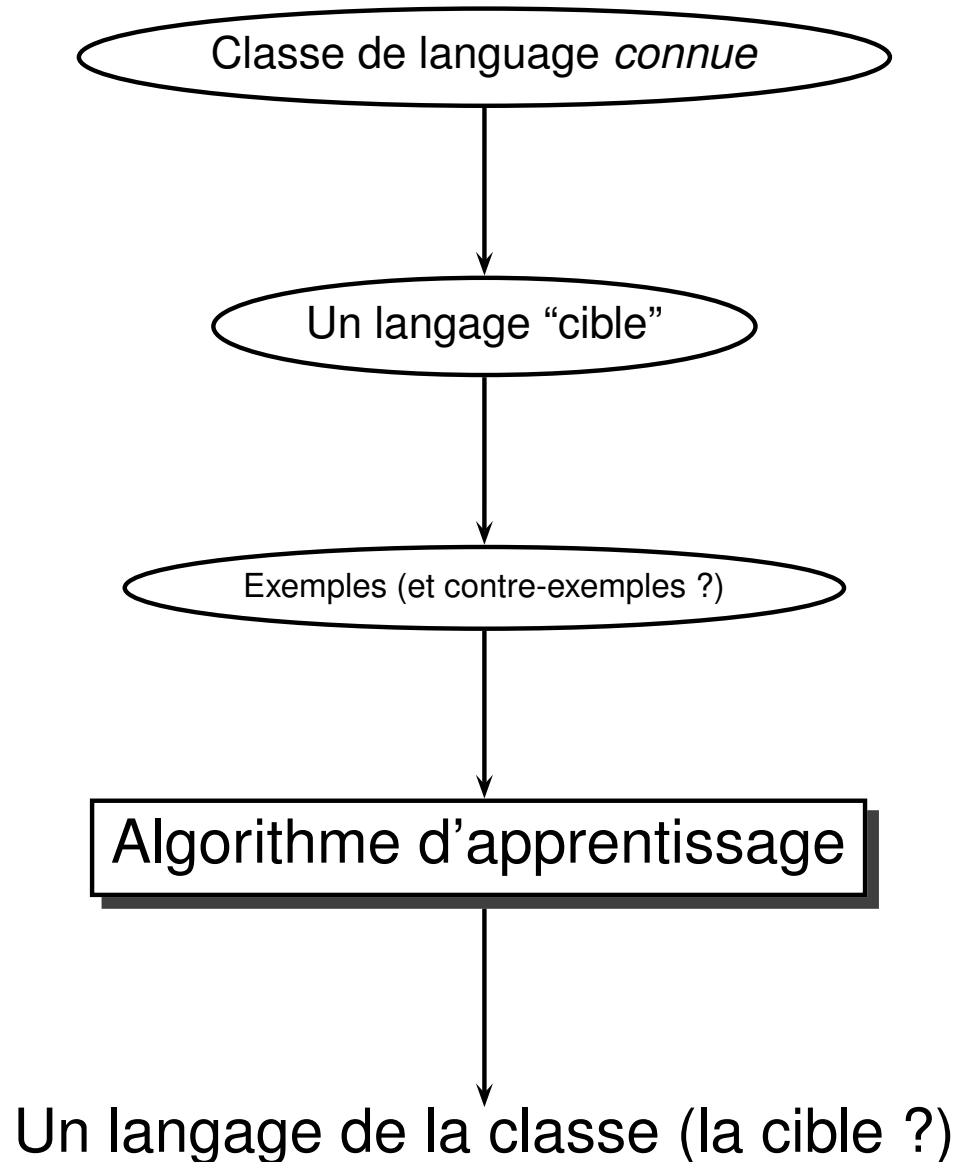
	+/-	+
Langages Context-free	NON	NON
Langages Réguliers	OUI	NON
Langages réversibles	OUI	OUI



1. Apprentissage de langages réguliers
  - (a) Un algorithme générique
  - (b) Apprentissage par exemples positifs/négatifs (RPNI)
  - (c) Apprentissage par exemples positifs (ZR)
2. Autres cadres d'apprentissage
  - (a) Langages stochastiques
  - (b) transductions
  - (c) langages d'arbres
  - (d) autres

---

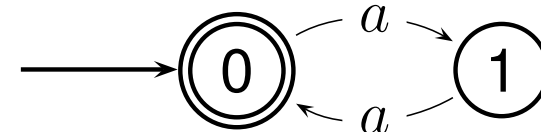
## 1.1 Algorithme générique d'apprentissage de réguliers



les langages réguliers

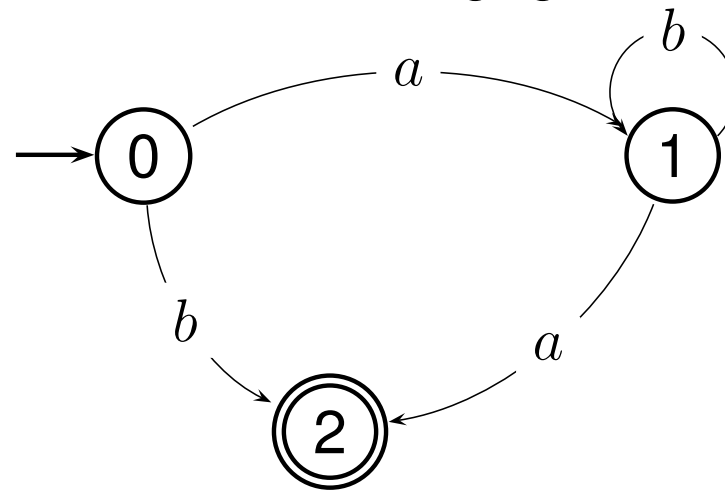
Ex :  $a^{2n}$

$a-, aa+, \varepsilon+, aaaa-, aaaaa+, aaaaaa-$



# Automate déterministe minimal et langages résiduels

AFD minimal du langage  $ab^*a + b$



$$L_{q_0} = ab^*a + b = \varepsilon^{-1}L \quad L_{q_1} = b^*a = a^{-1}L$$

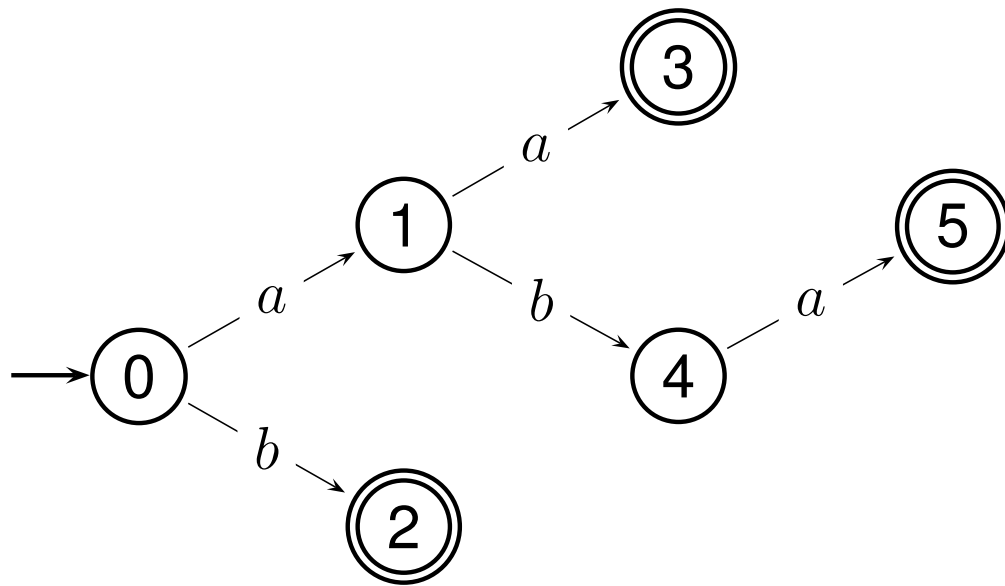
$$L_{q_2} = \varepsilon = b^{-1}L$$

- états de l'AFD minimal  $\Leftrightarrow$  langages résiduels (Myhill-Nérode)
- But (ici) de l'apprentissage : retrouver l'AFD minimal de  $L_{cible}$   
 $\Rightarrow$  Idem : *identifier* les langages résiduels de  $L_{cible}$

**Entrée** : Un ensemble d'exemples positifs (et négatifs ?)

1. Construction de l'arbre préfixe sur l'échantillon positif

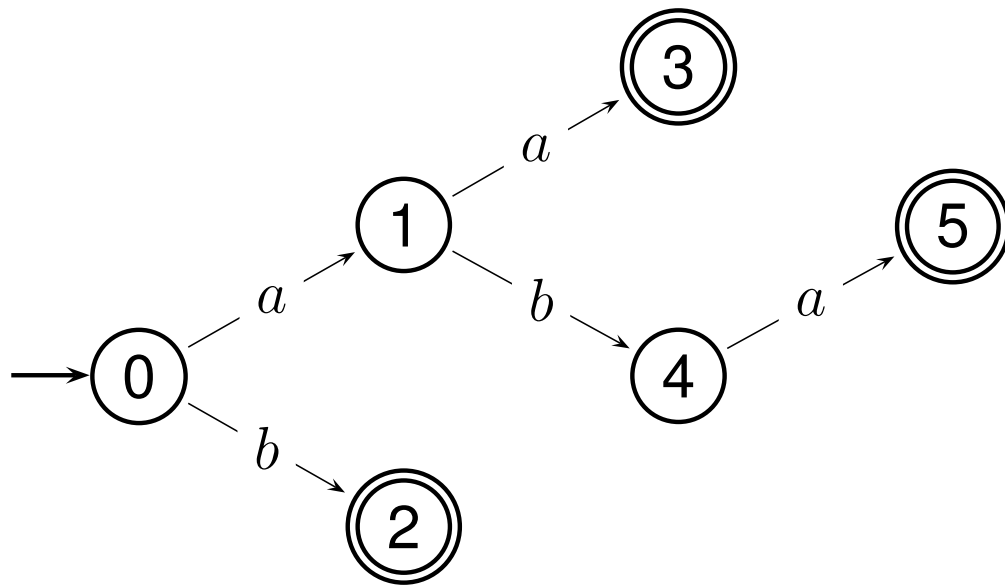
Ex :  $S^+ = \{b, aa, aba\}$



**Entrée** : Un ensemble d'exemples positifs (et négatifs ?)

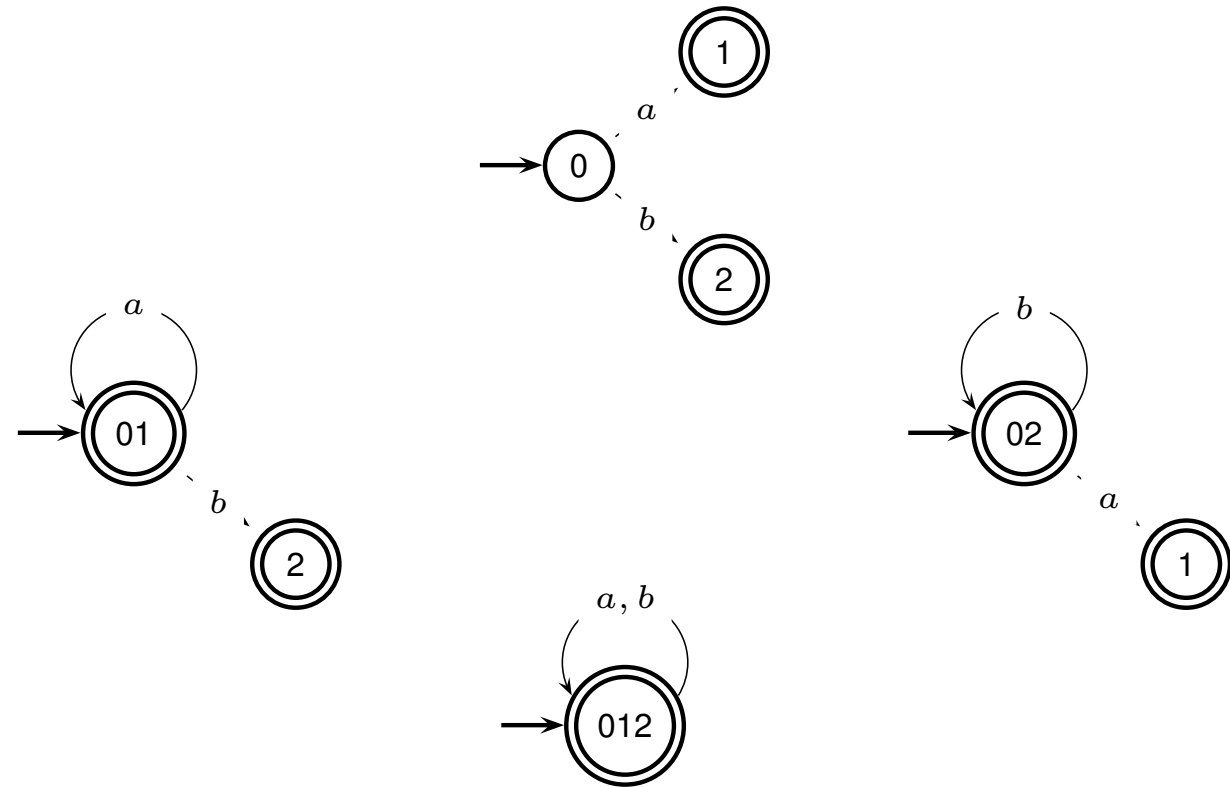
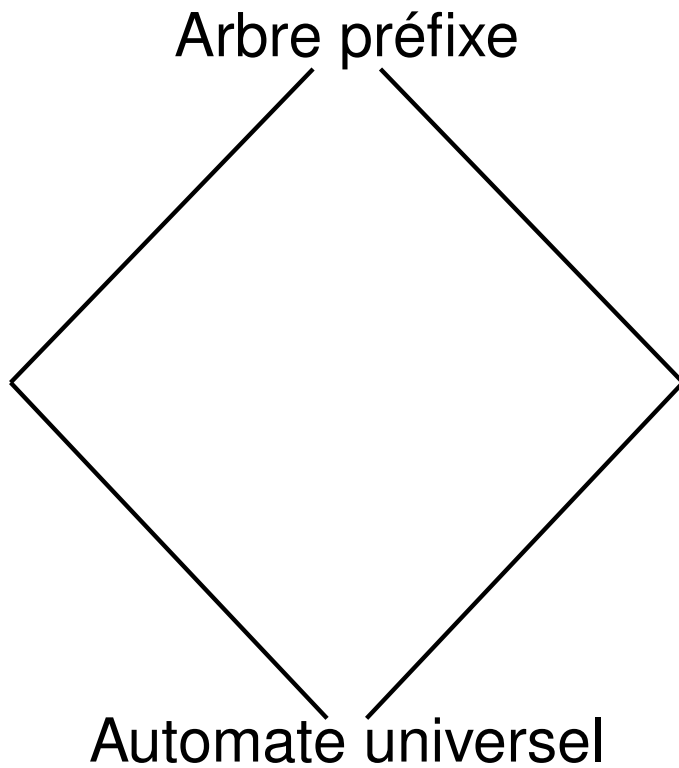
1. Construction de l'arbre préfixe sur l'échantillon positif

Ex :  $S^+ = \{b, aa, aba\}$



Chaque état : une estimation d'un langage résiduel

Ex :  $L_{q_1} \rightarrow a^{-1}L$  (En particulier  $L_{q_1} \subseteq a^{-1}L$ )



Si l'échantillon est *Structurellement complet*, l'automate cible est dans le treillis

**Entrée** : Un ensemble d'exemples positifs (et négatifs ?)

1. Construction de l'arbre préfixe
2. Recherche de la cible dans le treillis des fusions

Fusion des états correspondant à des résiduels **équivalents** :

$$L_q \simeq u^{-1}L_{cible}$$

$$L_{q'} \simeq v^{-1}L_{cible} \quad \Rightarrow \quad \text{Fusion de } q \text{ et } q'$$

$$u^{-1}L_{cible} = u^{-1}L_{cible}$$

**Sortie** : Un automate (consistant avec les échantillons)

Si l'échantillon est *caractéristique* : l'AFD minimal de la cible

---

## 1.2 Apprentissage de langages réguliers par +/-

## Idée

- $u^{-1}L \neq v^{-1}L \Rightarrow$  Il existe  $w$  tel que  $uw \in L$  et  $vw \notin L$  (ou l'inverse).
- Les positifs et les négatifs vont nous guider pour identifier les résiduels.
- Ceci nous donne un critère pour la fusion d'états !

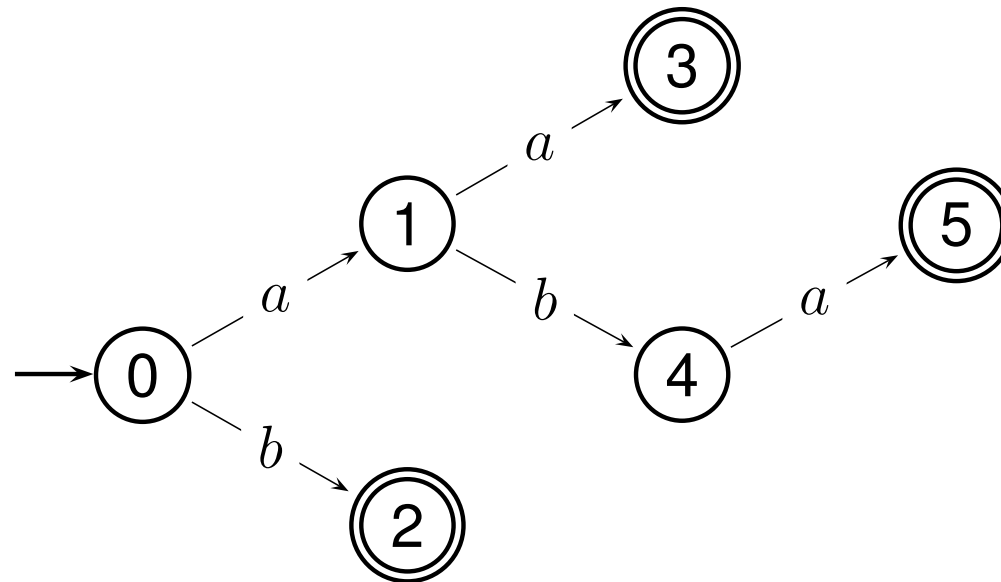
Algorithme RPNI :

**Entrée** : Un échantillon

1. Construction de l'arbre préfixe
2. Parcours en largeur de l'automate :
  - Fusion de  $(q_1, q_2)$  s'il n'y a rien qui l'empêche (i.e.  $L_{q_1}$  et  $L_{q_2}$  correspondent au même résiduel)

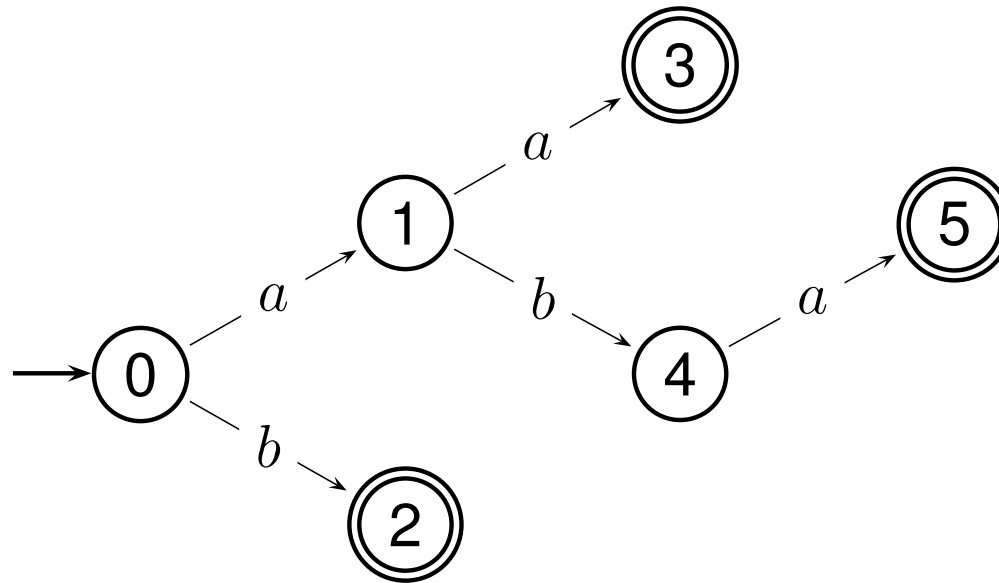
**Sortie** : un automate déterministe (la cible ?)

$$S^+ = \{b, aa, aba\}, S^- = \{\varepsilon, a, ab\}$$



## 1. Construction de l'arbre préfixe

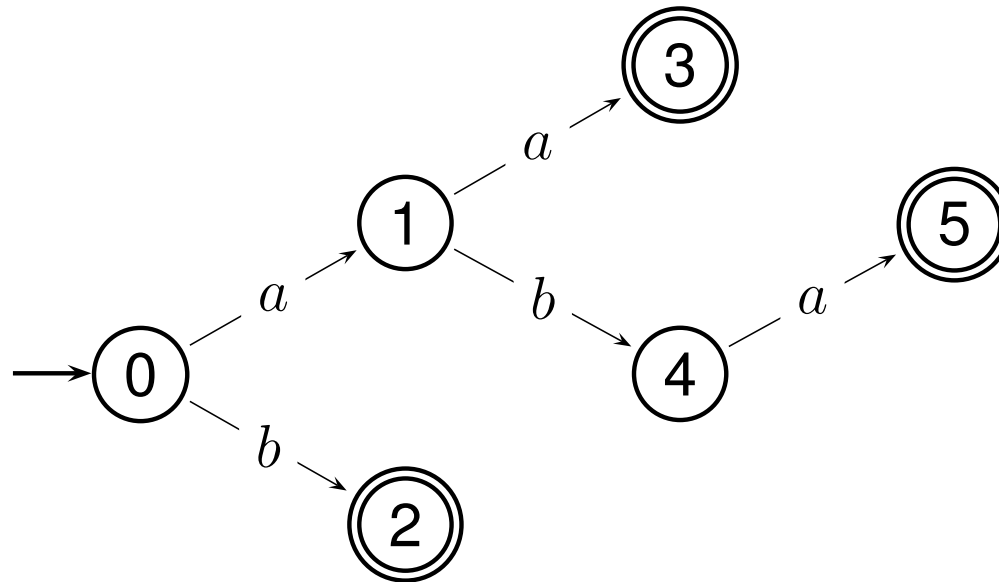
$$S^+ = \{b, aa, aba\}, S^- = \{\varepsilon, a, ab\}$$



2. Fusion de 0 et 1 ?

**NON** :  $b \in \varepsilon^{-1}L_{cible}, b \notin a^{-1}L_{cible}$

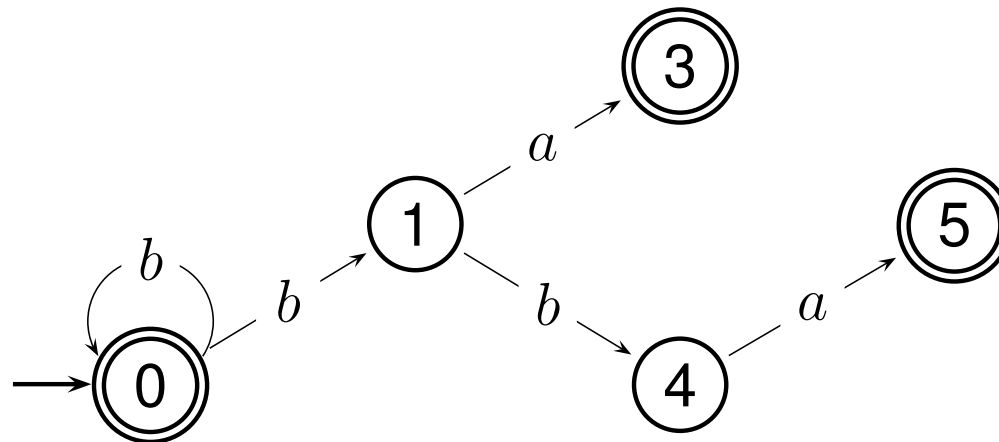
$$S^+ = \{b, aa, aba\}, S^- = \{\varepsilon, a, ab\}$$



2. Fusion de 0 et 2 ?

**NON** :  $\varepsilon \notin \varepsilon^{-1}L_{cible}, \varepsilon \notin b^{-1}L_{cible}$

$$S^+ = \{b, aa, aba\}, S^- = \{\varepsilon, a, ab\}$$

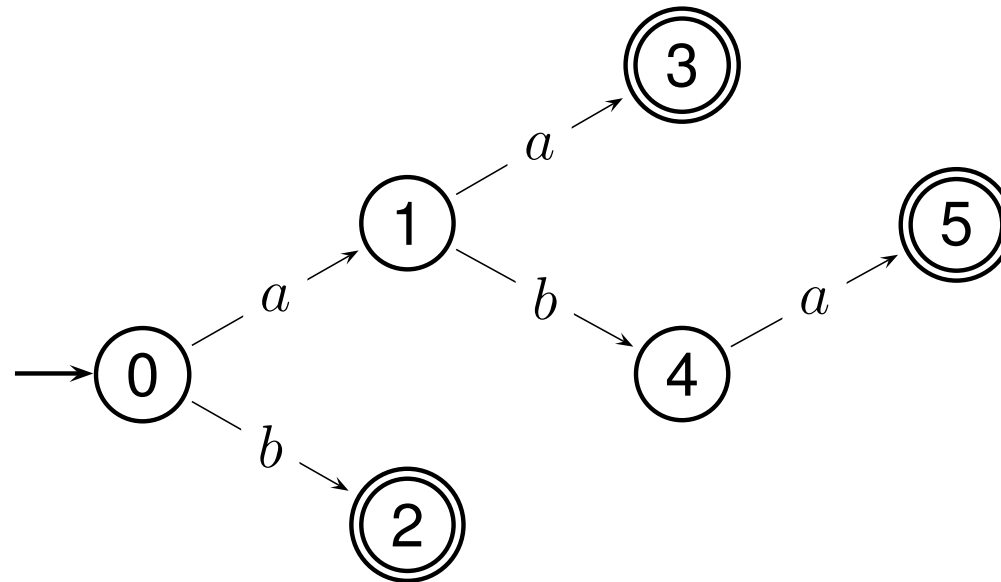


2. Fusion de 0 et 2 ?

**NON** :  $\varepsilon \notin \varepsilon^{-1}L_{cible}, \varepsilon \notin b^{-1}L_{cible}$

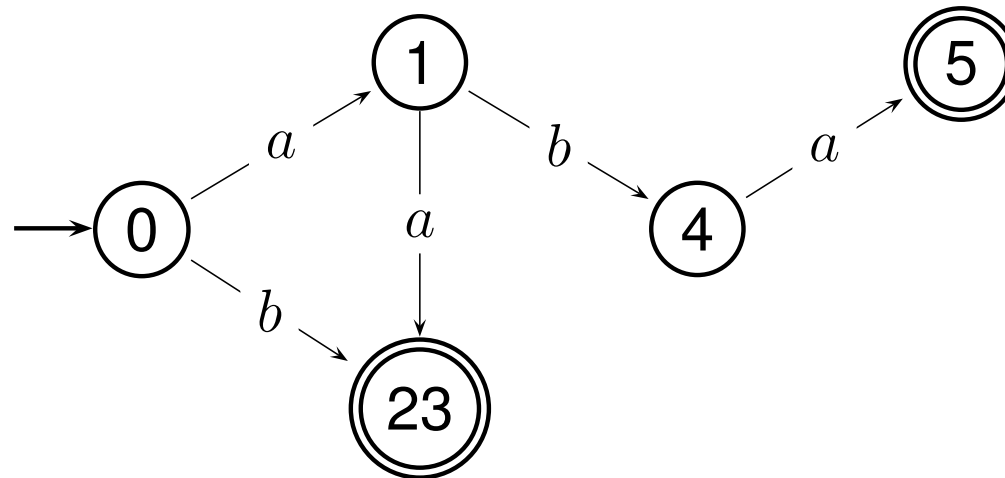
$\varepsilon$  est reconnu !

$$S^+ = \{b, aa, aba\}, S^- = \{\varepsilon, a, ab\}$$



2. Fusion de 2 et 3 ?

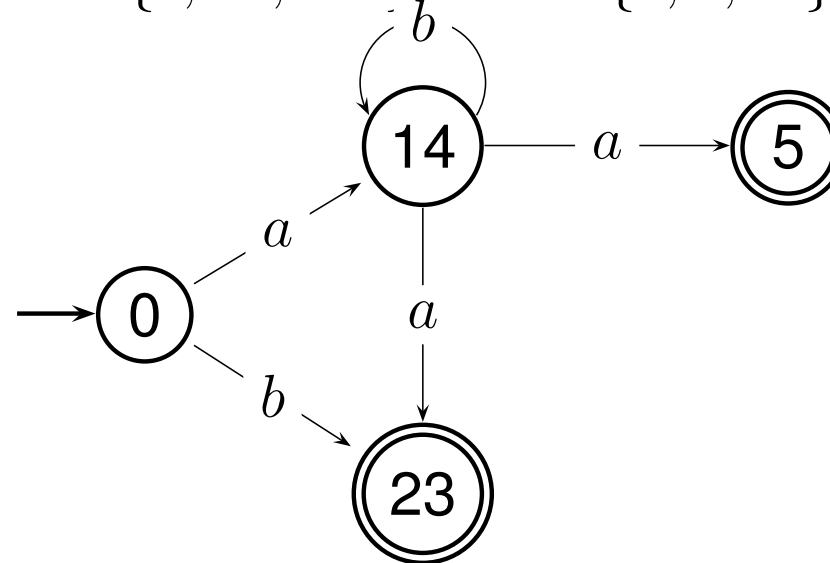
$$S^+ = \{b, aa, aba\}, S^- = \{\varepsilon, a, ab\}$$



2. Fusion de 2 et 3 ?

OUI : pas de problème (Fusion 2 - 3)

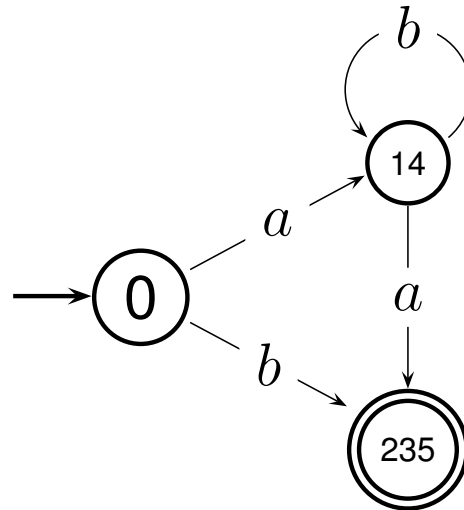
$$S^+ = \{b, aa, aba\} \quad S^- = \{\varepsilon, a, ab\}$$



2. Fusion de 1 et 4 ?

OUI : pas de problème (Fusion 1 - 4)

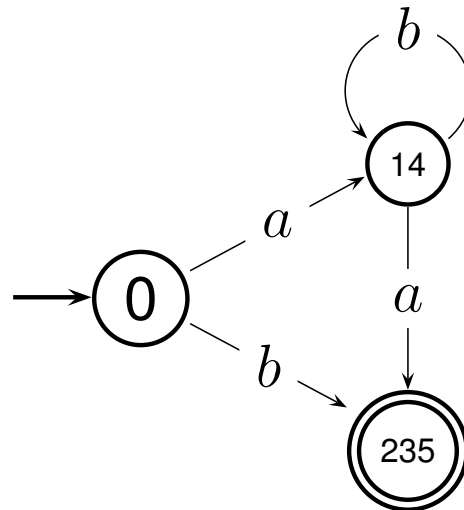
$$S^+ = \{b, aa, aba\}, S^- = \{\varepsilon, a, ab\}$$



2. Fusion de 1 et 4 ?

OUI : pas de problème (Fusion **déterministe** 1 - 4)

$$S^+ = \{b, aa, aba\}, S^- = \{\varepsilon, a, ab\}$$



**Sortie :** Un automate déterministe (la cible ?)

Algorithme RPNI (tel que présenté par Oncina et Garcia) :

**Entrée** : Un échantillon

1. Construction de l'arbre préfixe  $A_0$ ,  $i \leftarrow 0$
2. Parcours en largeur de l'automate :
  - $A \leftarrow$  Fusion **déterministe** de  $(q_1, q_2)$
  - **Si**  $A$  reconnaît un exemple négatif **Alors**  $A_{i-1} \leftarrow A_i$
  - **Sinon**  $A_i \leftarrow A$
  - $i \leftarrow i + 1$

**Sortie** : un automate déterministe (la cible ?)

## Variantes de RPNI :

- Trakhtenbrot - Barzdin  
(antérieur à RPNI)
- Red-Blue (et autres 'Evidence Driven State Merging')  
(cf. concours ABBADINGO, gagné par Rod Price et Hugues Juille)  
améliorations heuristiques
- DeLeTe (F. Denis, A. Lemay, A. Terlutte)  
Idée : apprendre des automates non-déterministes (AFER) en s'intéressant aux inclusions plutôt qu'aux égalités de langages.

---

## 1.2 Apprentissage de langages réguliers par positifs

# Apprentissage par exemples positifs

---

Résultats d' "apprenabilité":

Langages réguliers : **NON**

# Apprentissage par exemples positifs

Résultats d' "apprenabilité":

Langages réguliers : **NON**

Exemple de classes de langages:

- $C_1 = \{\varepsilon, a^{\leq 1}, a^{\leq 2}, a^{\leq 3}, \dots\}$

- $S = \{a, a^3, a^4\}$

⇒ Langage observé :  $a^{\leq 4}$  ? (**apprenable**)

# Apprentissage par exemples positifs

Résultats d' "apprenabilité":

Langages réguliers : **NON**

Exemple de classes de langages:

- $C_1 = \{\varepsilon, a^{\leq 1}, a^{\leq 2}, a^{\leq 3}, \dots\}$

- $S = \{a, a^3, a^4\}$

⇒ Langage observé :  $a^{\leq 4}$  ? (**apprenable**)

- $C_2 = \{\varepsilon, a^{\leq 1}, a^{\leq 2}, a^{\leq 3}, \dots\} \cup \{a^*\}$

- $S = \{a, a^3, a^4\}$

⇒ Langage observé :  $a^{\leq 4}$  ? ou  $a^*$  ? (**non apprenable**)

# Apprentissage par exemples positifs

Résultats d' "apprenabilité":

Langages réguliers : **NON**

Exemple de classes de langages:

- $C_1 = \{\varepsilon, a^{\leq 1}, a^{\leq 2}, a^{\leq 3}, \dots\}$

- $S = \{a, a^3, a^4\}$

⇒ Langage observé :  $a^{\leq 4}$  ? (**apprenable**)

- $C_2 = \{\varepsilon, a^{\leq 1}, a^{\leq 2}, a^{\leq 3}, \dots\} \cup \{a^*\}$

- $S = \{a, a^3, a^4\}$

⇒ Langage observé :  $a^{\leq 4}$  ? ou  $a^*$  ? (**non apprenable**)

la classe des langages réguliers contient  $C_2$  : **non apprenable** (par exemples positifs)

# Apprentissage par exemples positifs

Résultats d' "apprenabilité":

Langages réguliers :	NON
Langages $k$ -réversibles <sup>1</sup> :	OUI
Langages $k$ -testables <sup>2</sup> :	OUI
Langages $f$ -distinguables <sup>3</sup> :	OUI
Langages à résiduels premiers disjoints <sup>4</sup> :	OUI

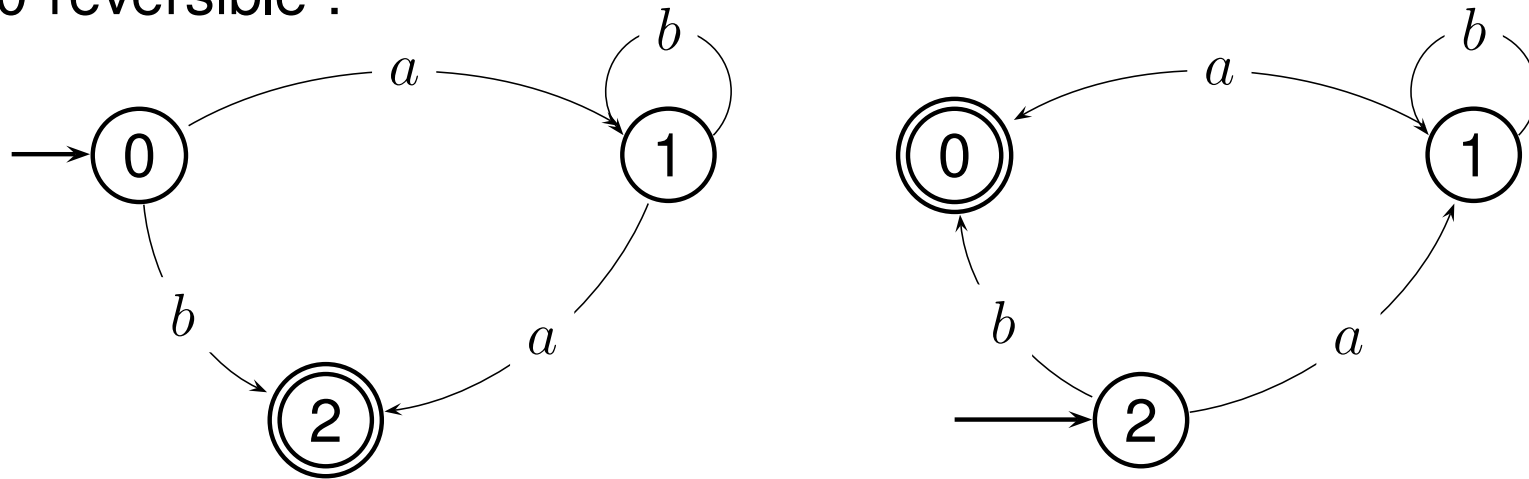
<sup>1</sup> Angluin (82)

<sup>2</sup> Garcia, Vidal (90)

<sup>3</sup> Fernau (00)

<sup>4</sup> Denis, Lemay, Terlutte (02)

Automate 0-réversible :



- Automate 0-réversible : automate **déterministe** dont le **miroir** est déterministe.
- Langage 0-réversible = reconnu par un automate 0-réversible
- $q_1 \neq q_2 \Rightarrow L_{q_1} \cap L_{q_2} = \emptyset$  ( $u^{-1}L \neq v^{-1}L \Rightarrow u^{-1}L \cap v^{-1}L = \emptyset$ )
- Apprenable par exemples positifs (ZR [Angluin 82])

**Entrée :** un échantillon (positif)

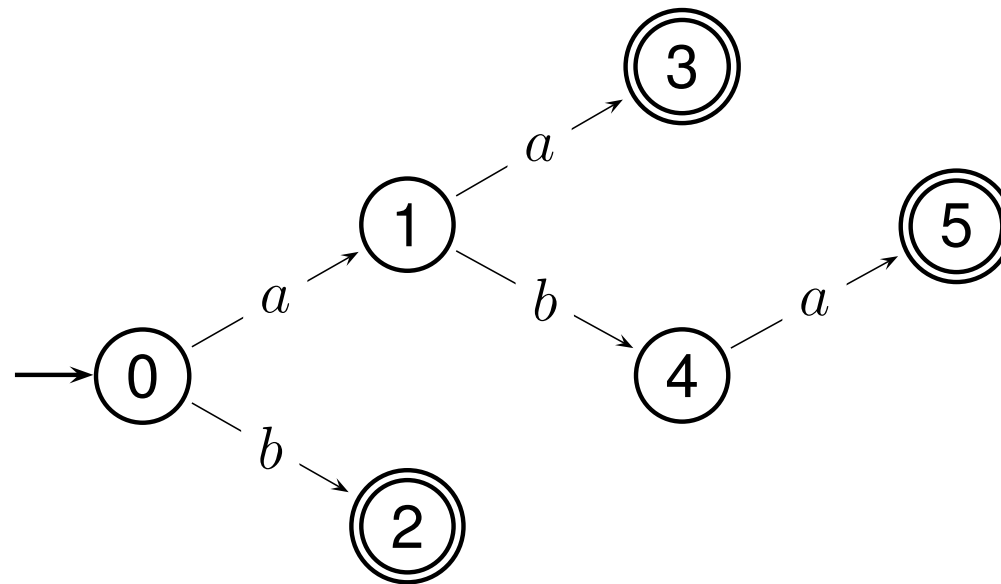
1. construction de l'arbre préfixe

2. Parcours de l'automate :

$\Rightarrow$  si  $L_{q_1} \cap L_{q_2} \neq \emptyset \Rightarrow$  fusion déterministe de  $q_1$  et  $q_2$

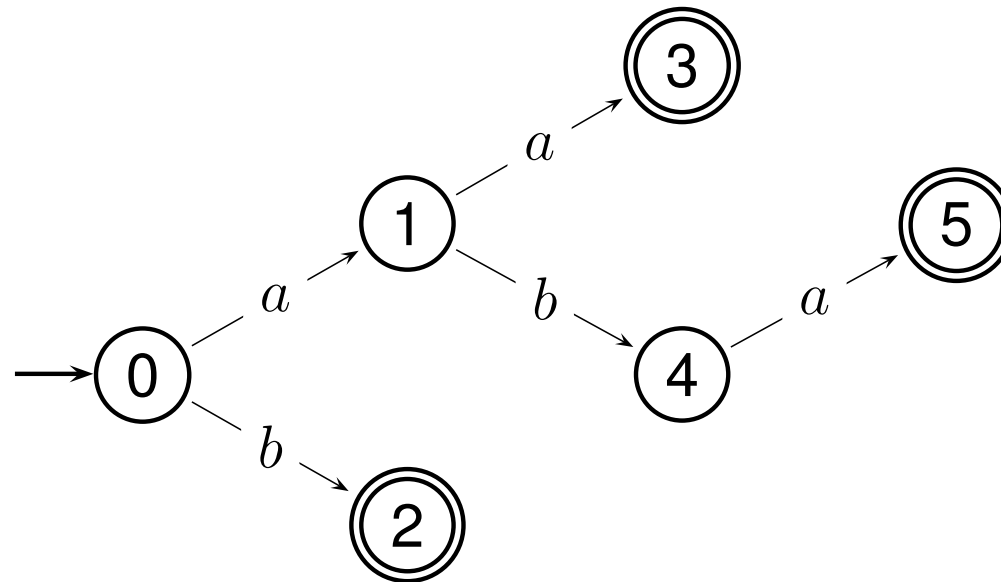
**Sortie :** un automate 0-réversible (la cible ?)

$$S = \{b, aa, aba\}$$



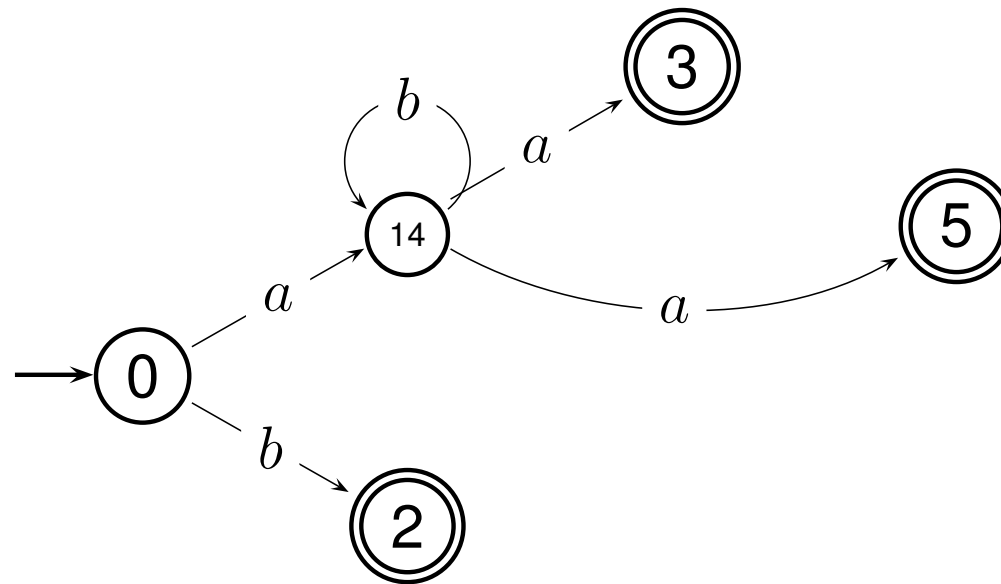
## 1. Construction de l'arbre préfixe

$$S = \{b, aa, aba\}$$



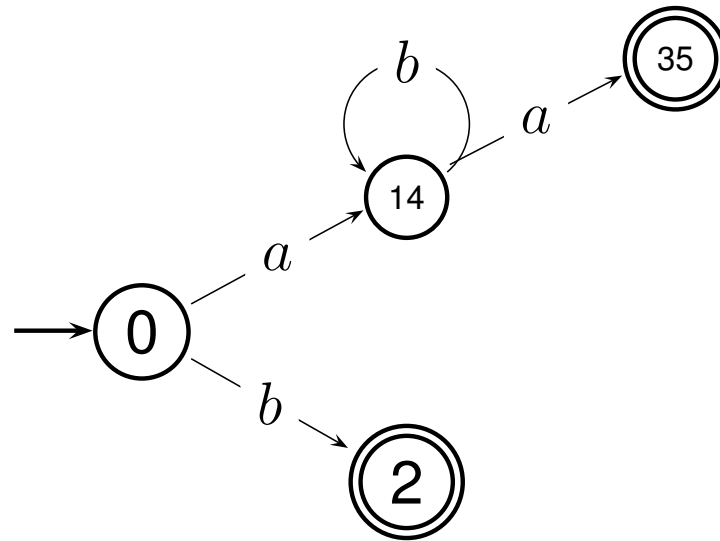
2. Fusion de 1 et 4 ( $a \in L_{q_1} \cap L_{q_4}$ )

$$S = \{b, aa, aba\}$$



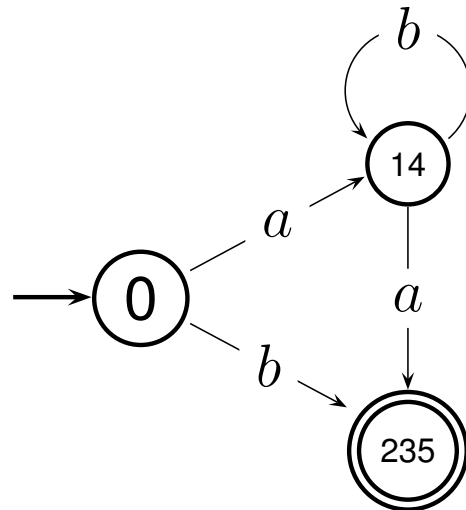
## 2. Fusion déterministe de 1 et 4

$$S = \{b, aa, aba\}$$



2. Fusion de 2 et 35 ( $\varepsilon \in L_{q_1} \cap L_{q_{35}}$ )

$$S = \{b, aa, aba\}$$



3. Sortie : un automate 0-réversible ! (la cible ?)

Langages décrits uniquement par des propriétés locales :

- Ensemble des départs possibles : (Ex :  $\{a\}$ )
- Ensemble des suites de lettres possibles (Ex :  $\{ab, ba\}$ )
- Ensembles des fins possibles (Ex :  $\{b\}$ )

Langage décrit dans l'exemple : mots de la forme  $ababab \dots ab$

**Algorithme d'apprentissage** : apprentissage *par coeur* des trois ensembles.

Extension de ce résultat aux  $k$ -testables (Garcia-Vidal 90)

---

## 2) Apprentissage d'autres classes de concepts

## Dans la hiérarchie de Chomsky

Par exemples positifs et négatifs

- Langages “Even-Linear” (Takada 88) (règles de la forme  $X \rightarrow aYb$ )
- Langages linéaires déterministes (De la Higuera, Oncina 02)

Par exemples positifs :

- langages algébriques “très simples” (Yokomori 91)
- langages  $f$ -distinguables (Fernau 00)
- langages à résiduels premiers disjoints (Denis, Lemay, Terlutte 02)
- ...

---

## 2.1 Langages d'arbres

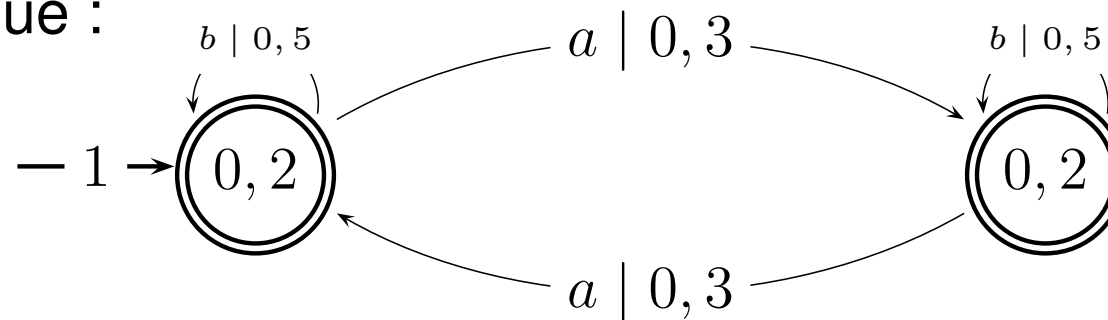
Langages **réguliers** d'arbres : représentés par des **automates** d'arbres

- Par exemples positifs et négatifs :
  - langages réguliers d'arbres : GIFT (variante de RPNI) de la Higuera/Bernard 99)
- Par exemples positifs : langages réversibles d'arbres
  - langages réversibles d'arbres : Sakakibara 90 - Kanazawa 94 - Marion/Besombes 2003 (variantes de ZR)
  - langages k-testables d'arbres : Kosala 02
  - langages d'arbres à résiduels premiers disjoints (GRAPPA 2003)

---

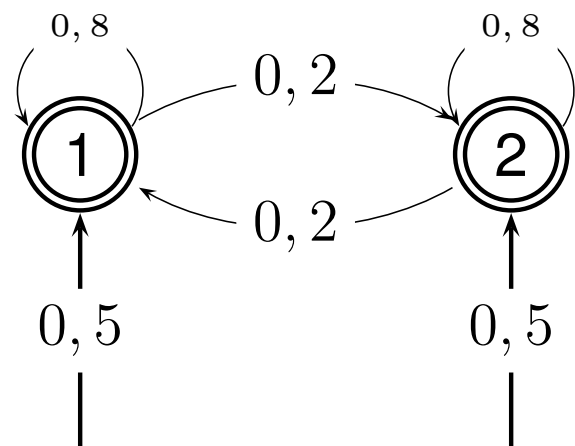
## 2.2 Langages stochastiques

Automate stochastique :

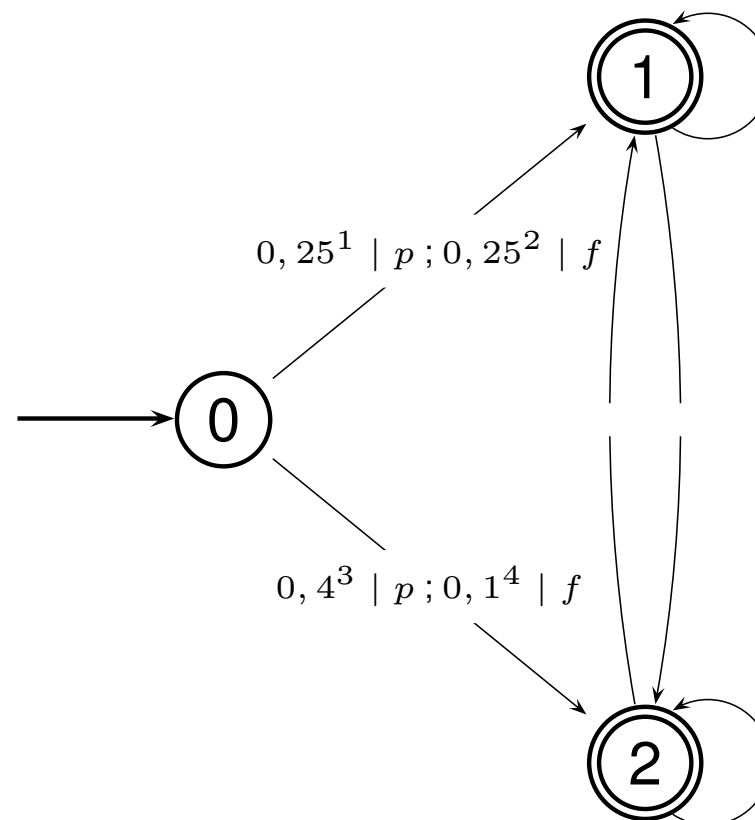


- $P(aa) = 1 \times 0,3 \times 0,3 \times 0,3 \times 0,2$
- Langages régulier stochastiques : distribution de probabilité sur  $\Sigma^*$  décrit par un automate stochastique.

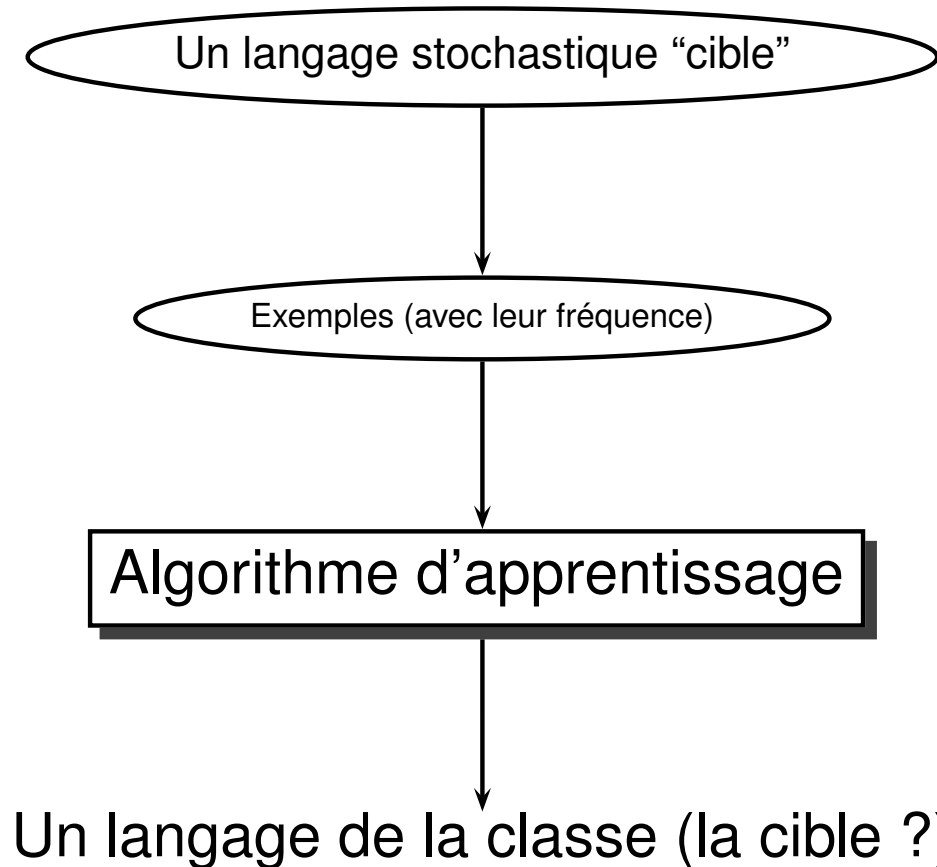
## HMM $\Leftrightarrow$ automate stochastique



$$\begin{aligned}
 p_1(\text{pile}) &= 0,5 & p_2(\text{pile}) &= 0,8 \\
 p_1(\text{face}) &= 0,5 & p_2(\text{face}) &= 0,2
 \end{aligned}$$

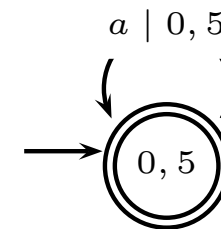


$$\begin{aligned}
 {}^1 0,25 &= 0,5 \times 0,5 & {}^3 0,4 &= 0,5 \times 0,8 \\
 {}^2 0,25 &= 0,5 \times 0,5 & {}^4 0,1 &= 0,5 \times 0,2
 \end{aligned}$$



$$\text{(Ex : } p(a^n) = 0,5^{n+1}\text{)}$$

$$(\varepsilon, aa, \varepsilon, \varepsilon, a, aa, \varepsilon, aaaa, \dots)$$



**Pas de négatifs !** Mais un exemple qui devrait apparaître et qui n'est pas présent est supposé négatif.

# Apprentissage d'automates stochastiques déterministes

---

**Entrée** : des exemples  $\varepsilon, \varepsilon, a, \varepsilon, aaa, aa, a, \varepsilon, \dots$

Algorithmes : **ALERGIA**<sup>1</sup>, **rlips**<sup>2</sup>, **MDI**<sup>3</sup> (variantes de RPNI)

**Sortie** : un automate stochastique **déterministe**

**Critère de fusion** : fusion si la distribution de probabilité après fusion correspond toujours aux observations

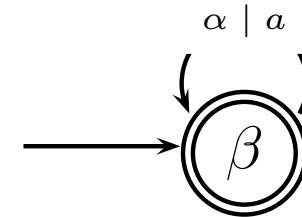
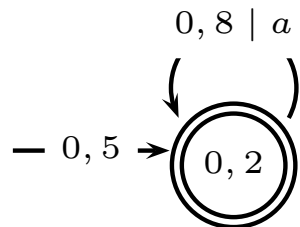
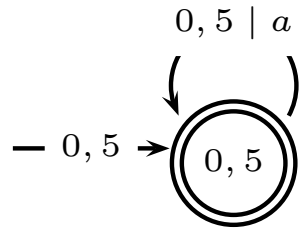
<sup>1</sup> Carrasco, Oncina (1994)

<sup>2</sup> Carrasco, Oncina (1999)

<sup>3</sup> de la Higuera, Dupont, Thollard (2000)

# Apprentissage d'automates non déterministes

Expressivité différentes pour automates stochastiques déterministes et non déterministes !



$$p(\varepsilon) = 0,5 \times 0,5 + 0,5 \times 0,2 = 0,35$$

$$p(a) = 0,5 \times 0,5 \times 0,5 + 0,5 \times 0,8 \times 0,2 = 0.285$$

$$p(a^n) = 0,5 \times 0,5^n \times 0,5 + 0,5 \times 0,8^n \times 0,2$$

$$p(\varepsilon) = \beta$$

$$p(a) = \alpha \times \beta$$

$$p(a^n) = \alpha^n \times \beta$$

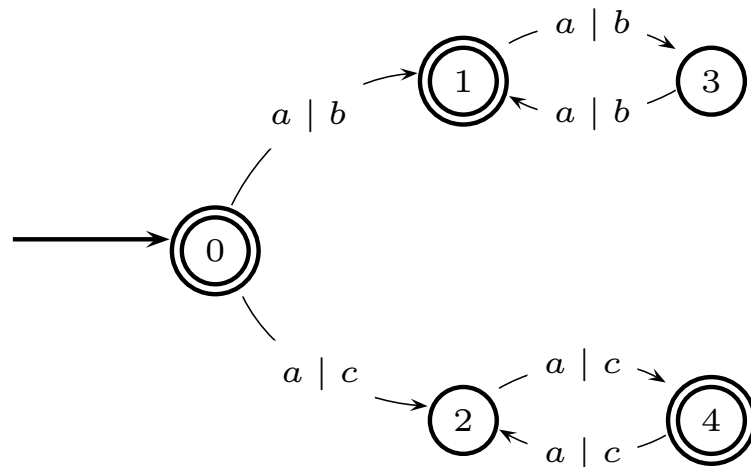
Apprentissage de non déterministe ?

- NON en règle général
- OUI pour certaines sous-classes (ex : Denis, Dupont, Esposito, Lemay - ICGI'02)

---

## 2.3 Transducteurs

Un transducteur rationnel :



$$\tau(a^{2n} = c^{2n})$$

$$\tau(a^{2n+1} = b^{2n+1})$$

**Exemple :**  $\tau(aaa) = bbb$

Classes de transduction rationnelles apprenables :

- transducteurs sous-séquentiels gauche (Oncina, Garcia, Vidal, 91)
- automates déterministes à 2 bandes (Yokomori, 96)
- fonctions rationnelles préservant les longueurs (en cours, GRAPPA)

---

## 2.4 Autres classes

- Formules logiques (idem langages d'arbres)  
GIFT (de la Higuera, Bernard 99)
- langages d'arbres stochastiques  
(Carrasco, Oncina, Calera-rubio 99)
- transducteurs d'arbres / Requêtes  
(En cours, GRAPPA)
- ...

---

# Conclusion

Champs d'application de l'inférence grammaticale :

- traitement du langage naturel (transducteurs, automates d'arbres)
- traitement de la parole (transducteurs, automates stochastiques)
- bio-informatique (automates stochastiques)
- Extraction d'information (automates d'arbres, transducteurs)