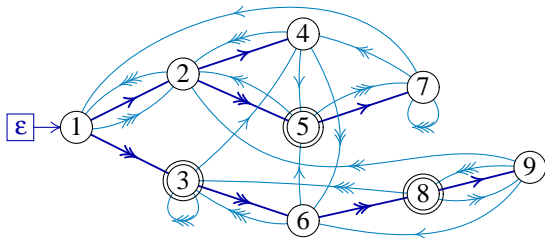


# Énumération asymptotique d'automates finis

Frédérique Bassino

Laboratoire d'Informatique de Paris Nord



Issu de travaux en collaboration avec  
*J. David, C. Nicaud et A. Sportiello*

Médaille d'Argent de Mireille Bouquet Mélou,  
Octobre 2014

# Pourquoi vouloir énumérer les automates finis ?

On cherche à

- ▶ Étudier de manière quantitative les propriétés des langages réguliers en utilisant leur représentation (**canonique**) par des automates (**minimaux**),
- ▶ Analyser en moyenne des algorithmes manipulant ces structures (par exemple des algorithmes de minimisation)

Pour cela, on a besoin

- ▶ D'énumérer, de simuler les automates (**minimaux**) vus comme des structures **combinatoires**.

# Pourquoi vouloir énumérer les automates finis ?

On cherche à

- ▶ Étudier de manière quantitative les propriétés des langages réguliers en utilisant leur représentation (**canonique**) par des automates (**minimaux**),
- ▶ Analyser en moyenne des algorithmes manipulant ces structures (par exemple des algorithmes de minimisation)

Pour cela, on a besoin

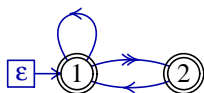
- ▶ D'énumérer, de simuler les automates (**minimaux**) vus comme des structures **combinatoires**.

# Langages réguliers

- ▶  $\Sigma$  un alphabet fini:  $\Sigma = \{a, b, c\}$
- ▶  $\Sigma^*$  l'ensemble des mots sur  $\Sigma$ :  $\Sigma^* = \{\epsilon, a, b, c, aa, ab, \dots\}$
- ▶ Un langage régulier : langage obtenu à partir de sous-ensembles finis de  $\Sigma^*$  et d'un nombre fini d'opérations rationnelles (union  $\cup$ , concaténation  $\cdot$  et étoile  $*$ ) où

$$X^* = \{x_1x_2 \cdots x_n \mid n \geq 0, x_i \in X\}$$

Exemple : les *mots de Fibonacci*,  
c'est-à-dire, les mots sur  $\Sigma = \{a, b\}$  sans  $bb$



Automate

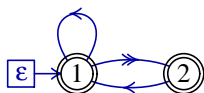
$$(a + ba)^*(\epsilon + b)$$

Expression régulière

- ▶  $\Sigma$  un alphabet fini:  $\Sigma = \{a, b, c\}$
- ▶  $\Sigma^*$  l'ensemble des mots sur  $\Sigma$ :  $\Sigma^* = \{\epsilon, a, b, c, aa, ab, \dots\}$
- ▶ Un langage régulier : langage obtenu à partir de sous-ensembles finis de  $\Sigma^*$  et d'un nombre fini d'opérations rationnelles (union  $\cup$ , concaténation  $\cdot$  et étoile  $*$ ) où

$$X^* = \{x_1x_2 \cdots x_n \mid n \geq 0, x_i \in X\}$$

Exemple : les *mots de Fibonacci*,  
c'est-à-dire, les mots sur  $\Sigma = \{a, b\}$  sans  $bb$



Automate

$$(a + ba)^*(\epsilon + b)$$

Expression régulière

Les langages réguliers peuvent être représentés par des expressions rationnelles écrites sur

- ▶ un alphabet fini  $\Sigma \cup \{\epsilon\}$
- ▶ avec les opérateurs binaires : union  $+$  et concaténation  $\cdot$
- ▶ et l'opérateur unaire  $*$ .

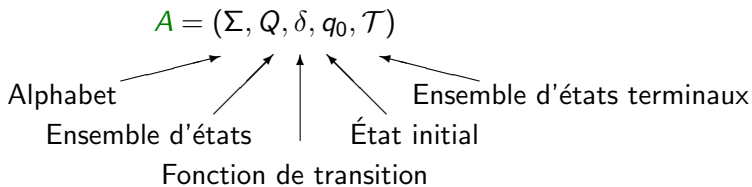
Expression rationnelle :  $(a + ba)^*(\epsilon + b)$

- ▶ Cette représentation des langages n'a pas de forme canonique
- ▶ Tester si une expression représente  $\Sigma^*$  est PSPACE-complet.

# Automates déterministes et complets

Un automate  $A$  est **déterministe et complet** s'il n'a qu'un **seul état initial** et que pour tout état  $q$  et pour toute lettre  $\ell$ , il existe **exactement une transition d'étiquette  $\ell$  partant de  $q$** .

$A$  est alors identifié à un quintuplet



► *Notations.*

Taille de l'alphabet :  $|\Sigma| = k$ . Nombre d'états :  $|Q| = n$ .

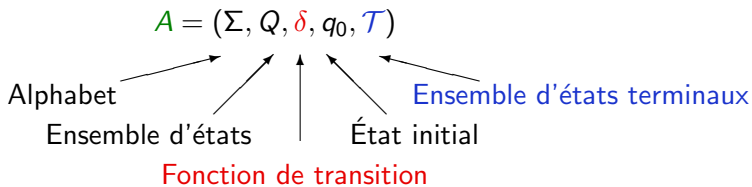
► On fait un choix canonique,  $\Sigma \equiv [k]$ ,  $Q \equiv [n]$  et  $q_0 = 1$ .

► Avec un ordre total sur  $Q$ , le couple  $(\delta, \mathcal{T})$  permet de reconstruire l'automate  $A$ .

# Automates déterministes et complets

Un automate  $A$  est **déterministe et complet** s'il n'a qu'un **seul état initial** et que pour tout état  $q$  et pour toute lettre  $\ell$ , il existe **exactement une transition d'étiquette  $\ell$  partant de  $q$** .

$A$  est alors identifié à un quintuplet



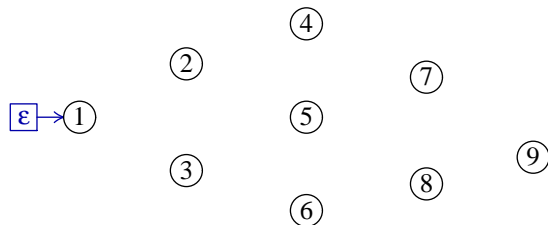
► *Notations.*

Taille de l'alphabet :  $|\Sigma| = k$ . Nombre d'états :  $|Q| = n$ .

- On fait un choix canonique,  $\Sigma \equiv [k]$ ,  $Q \equiv [n]$  et  $q_0 = 1$ .
- Avec un ordre total sur  $Q$ , le couple  $(\delta, \mathcal{T})$  permet de reconstruire l'automate  $A$ .

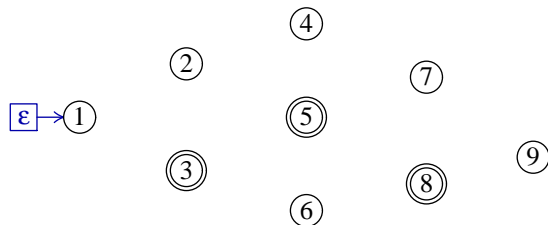


# Automates déterministes et complets



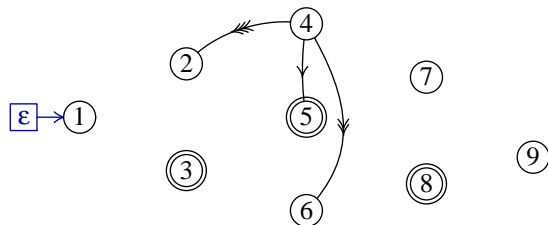
- ▶ Les états sont représentés par des cercles.
- ▶ Un état est final (dans  $\mathcal{T}$ ) s'il est doublement cerclé.
- ▶ Pour chaque état  $i \in Q$  et chaque lettre  $\alpha \in \Sigma$ , une transition sort de l'état  $i$ , vers l'état  $\delta(i, \alpha)$ .
- ▶ Pour chaque mot  $w \in \Sigma^*$ ,  $\delta(i, w)$  est l'état d'arrivée du chemin, partant de  $i$ , correspondant à la concaténation des transitions.
- ▶ Le mot  $w$  est reconnu par l'automate ssi  $\delta(1, w) \in \mathcal{T}$ .

# Automates déterministes et complets



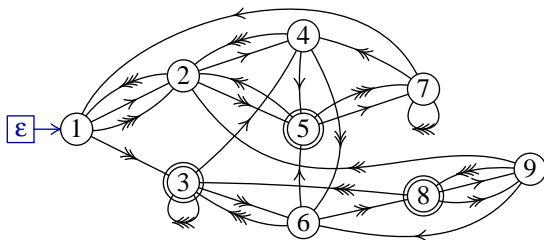
- ▶ Les états sont représentés par des cercles.
- ▶ Un état est final (dans  $\mathcal{T}$ ) s'il est doublement cerclé.
- ▶ Pour chaque état  $i \in Q$  et chaque lettre  $\alpha \in \Sigma$ , une transition sort de l'état  $i$ , vers l'état  $\delta(i, \alpha)$ .
- ▶ Pour chaque mot  $w \in \Sigma^*$ ,  $\delta(i, w)$  est l'état d'arrivée du chemin, partant de  $i$ , correspondant à la concaténation des transitions.
- ▶ Le mot  $w$  est reconnu par l'automate ssi  $\delta(1, w) \in \mathcal{T}$ .

# Automates déterministes et complets



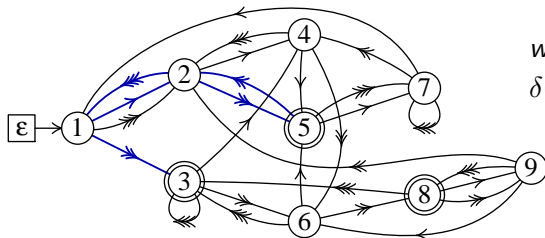
- ▶ Les états sont représentés par des cercles.
- ▶ Un état est final (dans  $\mathcal{T}$ ) s'il est doublement cerclé.
- ▶ Pour chaque état  $i \in Q$  et chaque lettre  $\alpha \in \Sigma$ , une transition sort de l'état  $i$ , vers l'état  $\delta(i, \alpha)$ .
- ▶ Pour chaque mot  $w \in \Sigma^*$ ,  $\delta(i, w)$  est l'état d'arrivée du chemin, partant de  $i$ , correspondant à la concaténation des transitions.
- ▶ Le mot  $w$  est reconnu par l'automate ssi  $\delta(1, w) \in \mathcal{T}$ .

# Automates déterministes et complets



- ▶ Les états sont représentés par des cercles.
- ▶ Un état est final (dans  $\mathcal{T}$ ) s'il est doublement cerclé.
- ▶ Pour chaque état  $i \in Q$  et chaque lettre  $\alpha \in \Sigma$ , une transition sort de l'état  $i$ , vers l'état  $\delta(i, \alpha)$ .
- ▶ Pour chaque mot  $w \in \Sigma^*$ ,  $\delta(i, w)$  est l'état d'arrivée du chemin, partant de  $i$ , correspondant à la concaténation des transitions.
- ▶ Le mot  $w$  est **reconnu** par l'automate ssi  $\delta(1, w) \in \mathcal{T}$ .

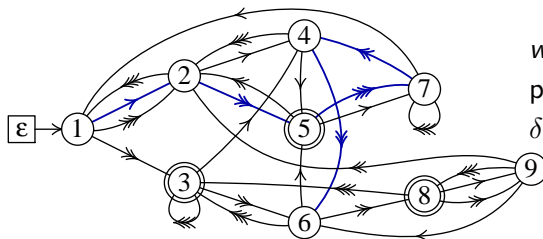
# Automates déterministes et complets



$w = (abbc b)$  est reconnu;  
 $\delta(i, w) = 3$ .

- ▶ Les états sont représentés par des cercles.
- ▶ Un état est final (dans  $\mathcal{T}$ ) s'il est doublement cerclé.
- ▶ Pour chaque état  $i \in Q$  et chaque lettre  $\alpha \in \Sigma$ , une transition sort de l'état  $i$ , vers l'état  $\delta(i, \alpha)$ .
- ▶ Pour chaque mot  $w \in \Sigma^*$ ,  $\delta(i, w)$  est l'état d'arrivée du chemin, partant de  $i$ , correspondant à la concaténation des transitions.
- ▶ Le mot  $w$  est **reconnu** par l'automate ssi  $\delta(1, w) \in \mathcal{T}$ .

# Automates déterministes et complets



$w = (abcbb)$  n'est pas reconnu;  
 $\delta(i, w) = 6$ .

- ▶ Les états sont représentés par des cercles.
- ▶ Un état est final (dans  $\mathcal{T}$ ) s'il est doublement cerclé.
- ▶ Pour chaque état  $i \in Q$  et chaque lettre  $\alpha \in \Sigma$ , une transition sort de l'état  $i$ , vers l'état  $\delta(i, \alpha)$ .
- ▶ Pour chaque mot  $w \in \Sigma^*$ ,  $\delta(i, w)$  est l'état d'arrivée du chemin, partant de  $i$ , correspondant à la concaténation des transitions.
- ▶ Le mot  $w$  est **reconnu** par l'automate ssi  $\delta(1, w) \in \mathcal{T}$ .

# Langages réguliers et automates minimaux

## Théorème de Kleene

Les langages reconnus par automates finis sont les langages réguliers.

- ▶ Pour chaque langage régulier  $\mathcal{L}$ , il existe un **unique** automate déterministe et complet de taille minimale qui le reconnaît : son **automate minimal**.

*Dans la suite, on va :*

- ▶ Compter (asymptotiquement pour  $n$  grand) les automates minimaux de taille  $n$ .
- ▶ Caractériser l'ensemble des automates minimaux, parmi tous les automates déterministes finis.  
*(propriétés statistiques, probabilités d'occurrence de motifs).*

*Conséquences :*

- ▶ Génération aléatoire des automates minimaux,
- ▶ Analyse en moyenne d'algorithmes sur les langages réguliers.

# Langages réguliers et automates minimaux

## Théorème de Kleene

Les langages reconnus par automates finis sont les langages réguliers.

- ▶ Pour chaque langage régulier  $\mathcal{L}$ , il existe un **unique** automate déterministe et complet de taille minimale qui le reconnaît : son **automate minimal**.

*Dans la suite, on va :*

- ▶ Compter (asymptotiquement pour  $n$  grand) les automates minimaux de taille  $n$ .
- ▶ Caractériser l'ensemble des automates minimaux, parmi tous les automates déterministes finis.  
*(propriétés statistiques, probabilités d'occurrence de motifs).*

*Conséquences :*

- ▶ Génération aléatoire des automates minimaux,
- ▶ Analyse en moyenne d'algorithmes sur les langages réguliers.



# Langages réguliers et automates minimaux

## Théorème de Kleene

Les langages reconnus par automates finis sont les langages réguliers.

- ▶ Pour chaque langage régulier  $\mathcal{L}$ , il existe un **unique** automate déterministe et complet de taille minimale qui le reconnaît : son **automate minimal**.

*Dans la suite, on va :*

- ▶ Compter (asymptotiquement pour  $n$  grand) les automates minimaux de taille  $n$ .
- ▶ Caractériser l'ensemble des automates minimaux, parmi tous les automates déterministes finis.  
*(propriétés statistiques, probabilités d'occurrence de motifs).*

*Conséquences :*

- ▶ Génération aléatoire des automates minimaux,
- ▶ Analyse en moyenne d'algorithmes sur les langages réguliers.

## Théorème de Kleene

Les langages reconnus par automates finis sont les langages réguliers.

- ▶ Pour chaque langage régulier  $\mathcal{L}$ , il existe un **unique** automate déterministe et complet de taille minimale qui le reconnaît : son **automate minimal**.

*Dans la suite, on va :*

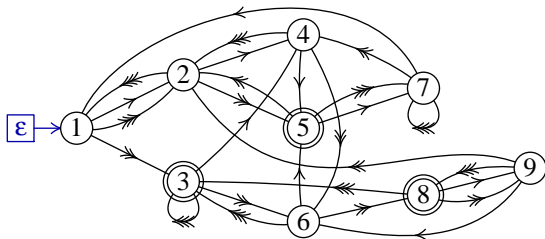
- ▶ Compter (asymptotiquement pour  $n$  grand) les automates minimaux de taille  $n$ .
- ▶ Caractériser l'ensemble des automates minimaux, parmi tous les automates déterministes finis.  
*(propriétés statistiques, probabilités d'occurrence de motifs).*

*Conséquences :*

- ▶ Génération aléatoire des automates minimaux,
- ▶ Analyse en moyenne d'algorithmes sur les langages réguliers.

# Énumération d'Automates

# Les automates : accessibilité

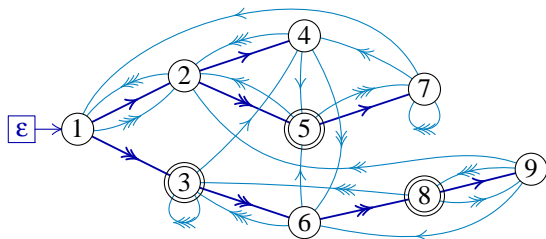


Un automate est **accessible** si, pour tout état  $i$ , il existe un mot  $w$  tel que  $\delta(1, w) = i$ .

Dans ce cas, un ordre canonique sur  $Q$  est donné par le parcours en largeur (selon l'ordre de  $\Sigma$ ) de l'automate à partir de l'état initial 1.

L'arbre de recherche associé couvre alors tous les états de l'automate

# Les automates : accessibilité



Un automate est **accessible** si, pour tout état  $i$ , il existe un mot  $w$  tel que  $\delta(1, w) = i$ .

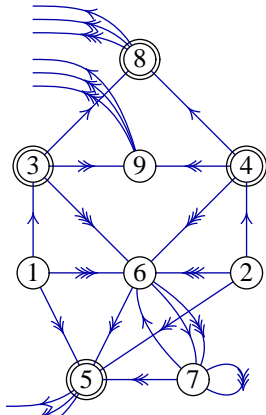
Dans ce cas, un ordre canonique sur  $Q$  est donné par le parcours en largeur (selon l'ordre de  $\Sigma$ ) de l'automate à partir de l'état initial 1.

L'arbre de recherche associé couvre alors tous les états de l'automate

# Les automates : Minimalité

Deux états  $i$  et  $j$  sont les **équivalents** - au sens de Myhill–Nerode - ssi, pour tout mot  $w$ , ou bien  $\delta(i, w), \delta(j, w) \in \mathcal{T}$  ou bien  $\delta(i, w), \delta(j, w) \notin \mathcal{T}$ .

On peut vérifier cette propriété par un parcours en largeur simultané de l'automate à partir des chacun des deux états.

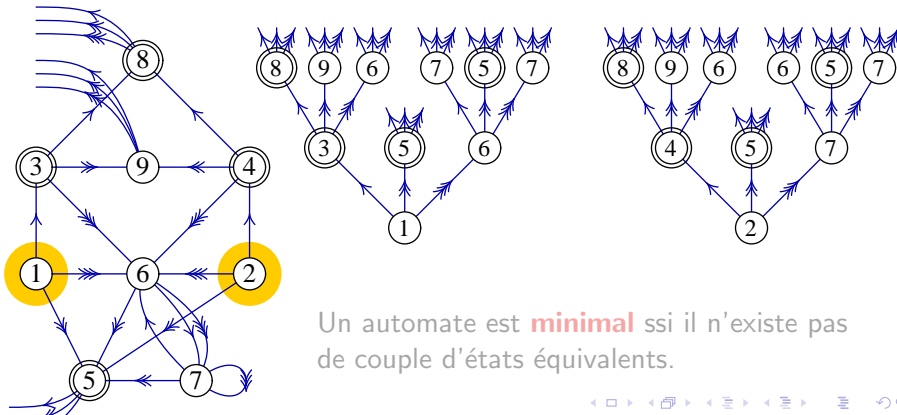


Un automate est **minimal** ssi il n'existe pas de couple d'états équivalents.

# Les automates : Minimalité

Deux états  $i$  et  $j$  sont les **équivalents** - au sens de Myhill–Nerode - ssi, pour tout mot  $w$ , ou bien  $\delta(i, w), \delta(j, w) \in \mathcal{T}$  ou bien  $\delta(i, w), \delta(j, w) \notin \mathcal{T}$ .

On peut vérifier cette propriété par un parcours en largeur simultané de l'automate à partir des chacun des deux états.

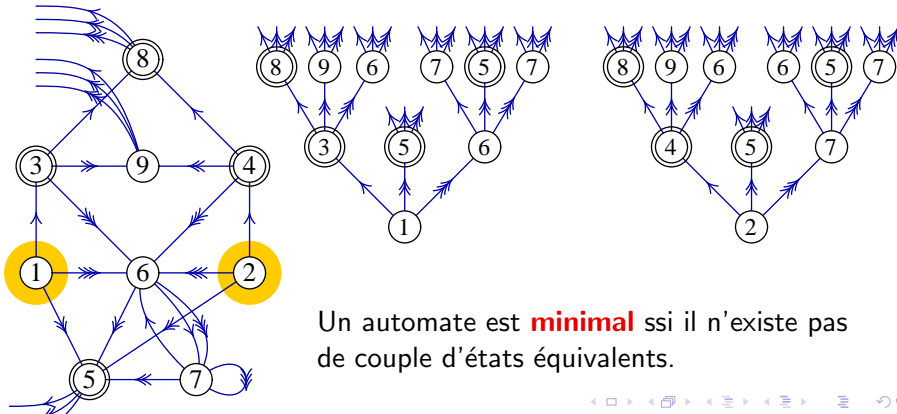


Un automate est **minimal** ssi il n'existe pas de couple d'états équivalents.

# Les automates : Minimalité

Deux états  $i$  et  $j$  sont les **équivalents** - au sens de Myhill–Nerode - ssi, pour tout mot  $w$ , ou bien  $\delta(i, w), \delta(j, w) \in \mathcal{T}$  ou bien  $\delta(i, w), \delta(j, w) \notin \mathcal{T}$ .

On peut vérifier cette propriété par un parcours en largeur simultané de l'automate à partir des chacun des deux états.



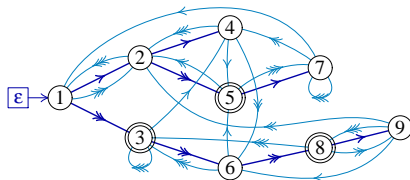
Un automate est **minimal** ssi il n'existe pas de couple d'états équivalents.



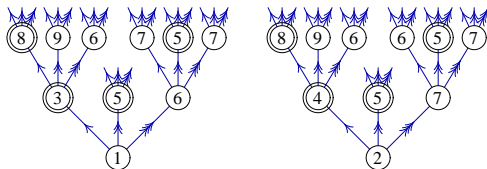
# Les automates : accessibilité et minimalité

Un automate est...

**accessible**, ssi les arbres de recherche issus de l'état initial couvrent tous les états de l'automate



**minimal**, ssi il n'existe pas de paire d'arbres de recherche certifiant que deux états sont équivalents

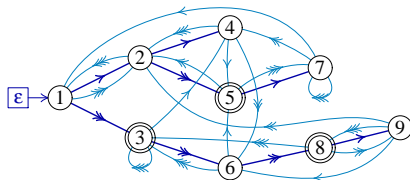


Il s'agit de contraintes **non-locales** et difficiles à exprimer d'un point de vue combinatoire.

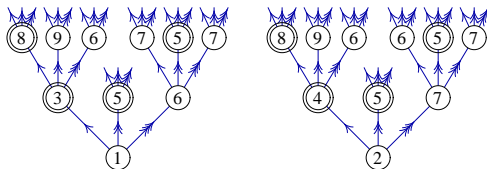
# Les automates : accessibilité et minimalité

Un automate est...

**accessible**, ssi les arbres de recherche issus de l'état initial couvrent tous les états de l'automate



**minimal**, ssi il n'existe pas de paire d'arbres de recherche certifiant que deux états sont équivalents



Il s'agit de contraintes **non-locales** et difficiles à exprimer d'un point de vue combinatoire.

# Enumérer les automates unaires ( $k = 1$ )

Les automates unaires (déterministes, complets) accessibles vérifient

$$i = 1 \cdot a^{i-1}$$

et sont donc caractérisés par le choix de l'état  $c$  tel que  $c = n \cdot a$ .

*Nicaud 1999*

- ▶ Il y a  $n 2^n$  automates unaires (déterministes, complets) accessibles canoniquement étiquetés.
- ▶ Si tout état est final avec probabilité  $b$ , alors une fraction  $1 - 2b(1 - b)$  des automates unaires (déterministes, complets) accessibles sont minimaux (la moitié d'entre eux dans le cas uniforme où  $b = 1/2$ ).

# Enumérer les automates unaires ( $k = 1$ )

Les automates unaires (déterministes, complets) accessibles vérifient

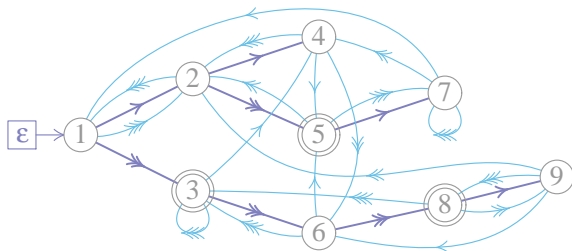
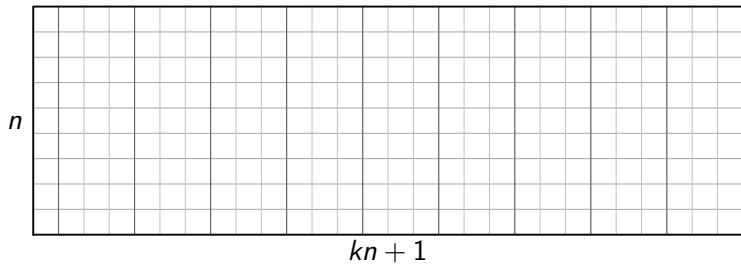
$$i = 1 \cdot a^{i-1}$$

et sont donc caractérisés par le choix de l'état  $c$  tel que  $c = n \cdot a$ .

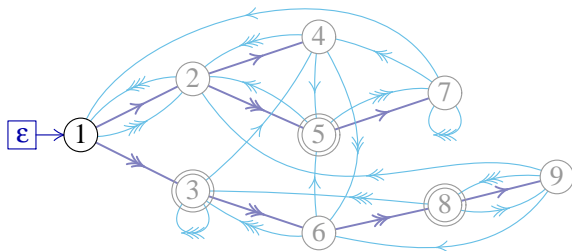
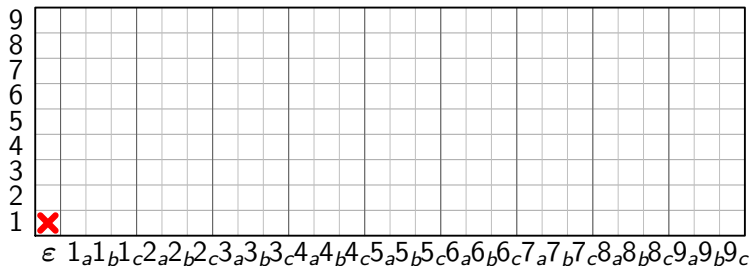
*Nicaud 1999*

- ▶ Il y a  $n 2^n$  automates unaires (déterministes, complets) accessibles canoniquement étiquetés.
- ▶ Si tout état est final avec probabilité  $b$ , alors une fraction  $1 - 2b(1 - b)$  des automates unaires (déterministes, complets) accessibles sont minimaux (la moitié d'entre eux dans le cas uniforme où  $b = 1/2$ ).

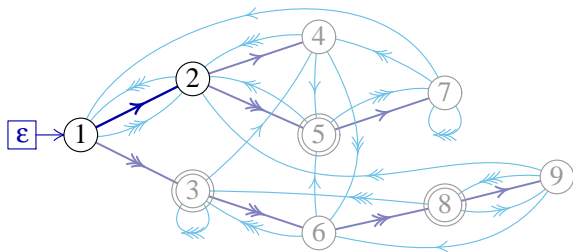
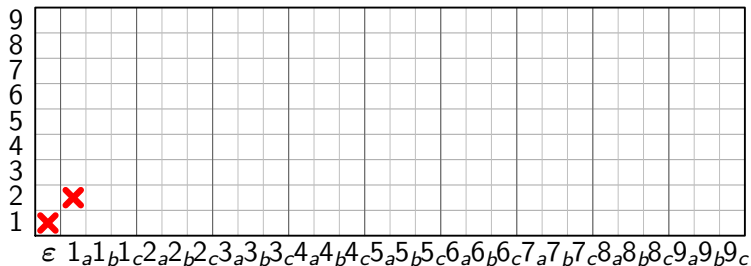
# Automates $k$ -aires ( $k \geq 2$ ) et tableaux



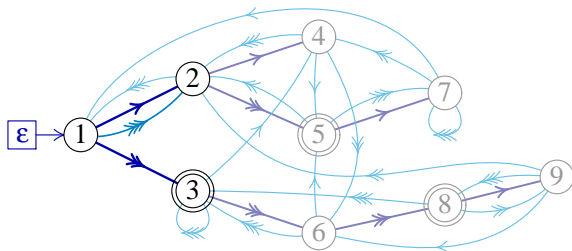
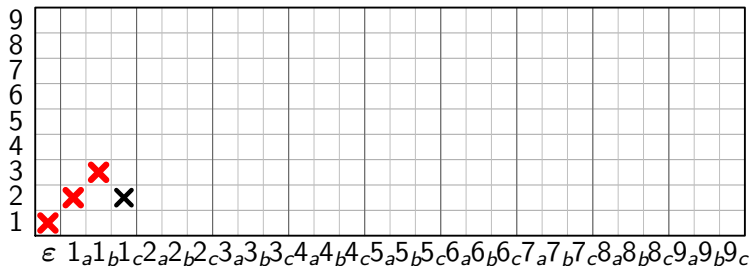
# Automates $k$ -aires ( $k \geq 2$ ) et tableaux



# Automates $k$ -aires ( $k \geq 2$ ) et tableaux

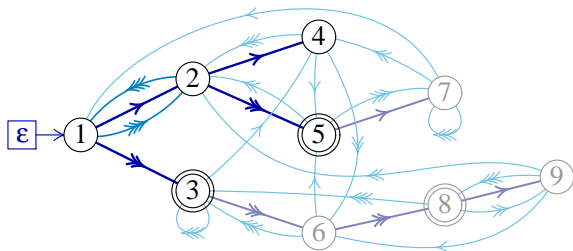
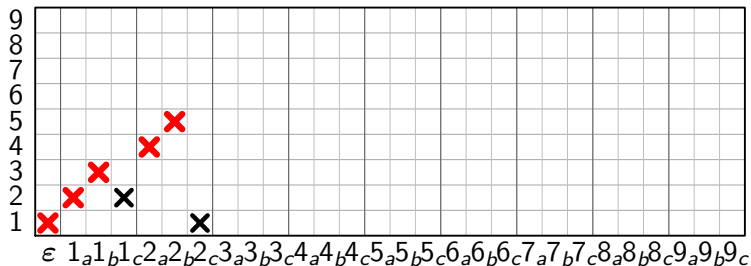


# Automates $k$ -aires ( $k \geq 2$ ) et tableaux

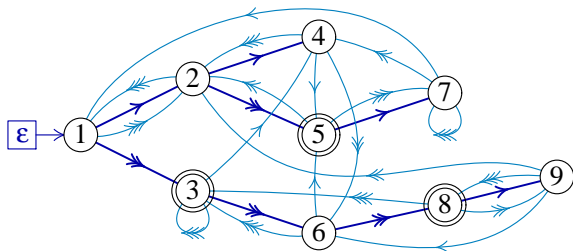
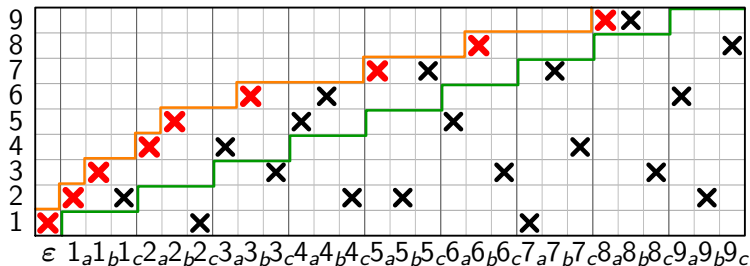




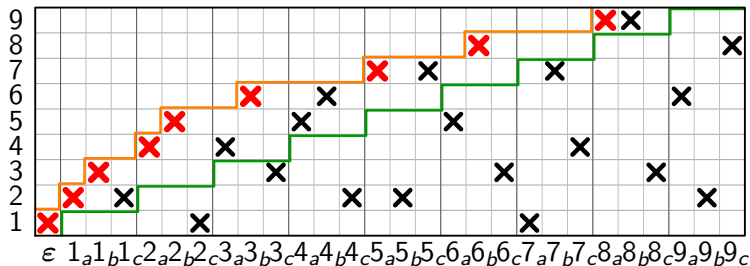
# Automates $k$ -aires ( $k \geq 2$ ) et tableaux



# Automates $k$ -aires ( $k \geq 2$ ) et tableaux

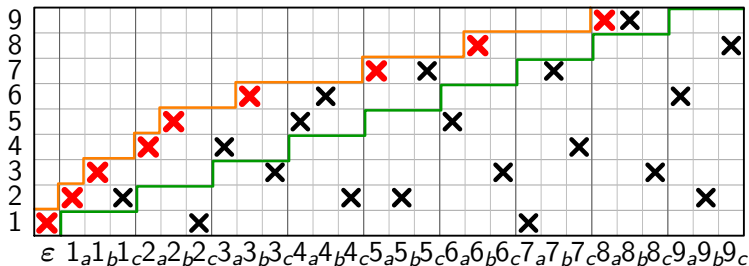


# Automates $k$ -aires ( $k \geq 2$ ) et tableaux



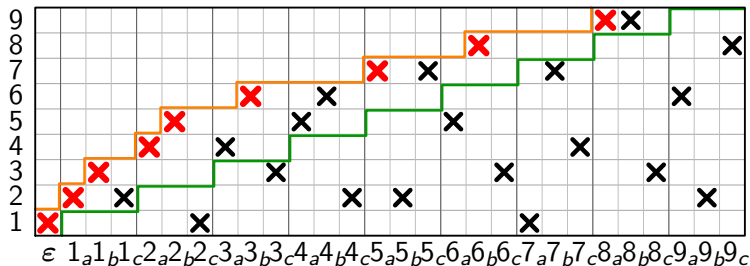
- Exactly one cross per column

# Automates $k$ -aires ( $k \geq 2$ ) et tableaux



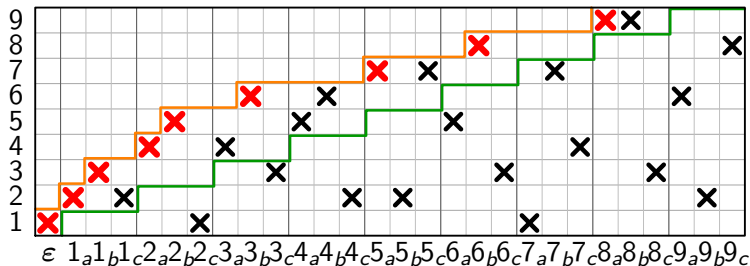
- Exactly one cross per column
- At least one cross per line – the one on the left is red

# Automates $k$ -aires ( $k \geq 2$ ) et tableaux



- Exactly one cross per column
- At least one cross per line – the leftmost one is red
- The red crosses (*squelette*) form a monotone sequence

# Automates $k$ -aires ( $k \geq 2$ ) et tableaux



- Exactly one cross per column
- At least one cross per line – the leftmost one is red
- The red crosses (*squelette*) form a monotone sequence
- Finally, the red crosses must stay to the left of the diagonal (*contrainte  $k$ -Dyck*)

# Enumérer les automates accessibles

Soit  $\mathcal{D}_{n,k}$  l'ensemble des structures de transition (déterministes complètes) accessibles, sur un alphabet de taille  $|\Sigma| = k \geq 2$ , avec  $|Q| = n$  états.

*Korshunov, 1978; Bassino et Nicaud, 2007; Lebensztayn, 2010.*

$$|\mathcal{D}_{n,k}| = \omega_k \left\{ \begin{matrix} kn+1 \\ n \end{matrix} \right\} (1 + \mathcal{O}(n^{-1})),$$

où  $\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$  est le *nombre de Stirling de seconde espèce*, et  $\omega_k$  satisfait  $-k\omega_k = \ln(1 - \omega_k)$ .

Qu'est ce qui se cache derrière ce théorème ?

- Le nombre de tableaux  $n \times m$  est exactement  $\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$  :  
Les *nombre de Stirling de 2nde espèce* comptent le nombre de partitions d'un ensemble à  $n$  éléments en  $m$  sous-ensembles non vides.
- Un tableau est  $k$ -Dyck, si le squelette ne passe pas sous la diagonale : cela se produit avec une probabilité d'ordre  $\omega_k$ .

# Enumérer les automates accessibles

Soit  $\mathcal{D}_{n,k}$  l'ensemble des structures de transition (déterministes complètes) accessibles, sur un alphabet de taille  $|\Sigma| = k \geq 2$ , avec  $|Q| = n$  états.

*Korshunov, 1978; Bassino et Nicaud, 2007; Lebensztayn, 2010.*

$$|\mathcal{D}_{n,k}| = \omega_k \left\{ \begin{matrix} kn+1 \\ n \end{matrix} \right\} (1 + \mathcal{O}(n^{-1})),$$

où  $\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$  est le *nombre de Stirling de seconde espèce*, et  $\omega_k$  satisfait  $-k\omega_k = \ln(1 - \omega_k)$ .

Qu'est ce qui se cache derrière ce théorème ?

- Le nombre de tableaux  $n \times m$  est exactement  $\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$  :  
Les *nombre de Stirling de 2nde espèce* comptent le nombre de partitions d'un ensemble à  $n$  éléments en  $m$  sous-ensembles non vides.
- Un tableau est  $k$ -Dyck, si le squelette ne passe pas sous la diagonale : cela se produit avec une probabilité d'ordre  $\omega_k$ .



# Enumérer les automates accessibles

Soit  $\mathcal{D}_{n,k}$  l'ensemble des structures de transition (déterministes complètes) accessibles, sur un alphabet de taille  $|\Sigma| = k \geq 2$ , avec  $|Q| = n$  états.

*Korshunov, 1978; Bassino et Nicaud, 2007; Lebensztayn, 2010.*

$$|\mathcal{D}_{n,k}| = \omega_k \left\{ \begin{matrix} kn+1 \\ n \end{matrix} \right\} (1 + \mathcal{O}(n^{-1})),$$

où  $\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$  est le *nombre de Stirling de seconde espèce*, et  $\omega_k$  satisfait  $-k\omega_k = \ln(1 - \omega_k)$ .

Qu'est ce qui se cache derrière ce théorème ?

- Le nombre de tableaux  $n \times m$  est exactement  $\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$  :  
Les *nombre de Stirling de 2nde espèce* comptent le nombre de partitions d'un ensemble à  $n$  éléments en  $m$  sous-ensembles non vides.
- Un tableau est  $k$ -Dyck, si le squelette ne passe pas sous la diagonale : cela se produit avec une probabilité d'ordre  $\omega_k$ .

# Enumérer les automates minimaux

Soit  $\mathcal{A}_{n,k}$  l'ensemble des automates accessibles, sur un alphabet de taille  $|\Sigma| = k \geq 2$ , avec  $|Q| = n$  états.

*Korshunov, 1978; Bassino et Nicaud, 2007; Lebensztayn, 2010.*

$$|\mathcal{A}_{n,k}| = 2^n \omega_k \left\{ \begin{matrix} kn+1 \\ n \end{matrix} \right\} (1 + \mathcal{O}(n^{-1})) ,$$

où  $\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$  est le *nombre de Stirling de seconde espèce*, et  $\omega_k$  satisfait  $-k\omega_k = \ln(1 - \omega_k)$ .

On munit l'ensemble des automates accessibles  $A = (\delta, \mathcal{T})$  de la mesure  $\mu(\delta, \mathcal{T}) = \mu_{\text{uniforme}}(\delta) \times \mu_{\text{Bernoulli}}^{(b)}(\mathcal{T})$ .

*Bassino, David et Sportiello, 2012*

Quand  $n$  tend vers  $+\infty$ , la probabilité qu'un automate accessible  $A$  soit minimal tend vers

$$\exp\left(-\frac{1-2b(1-b)}{2} \omega_k^k n^{-k+2}\right) .$$

# Enumérer les automates minimaux

Soit  $\mathcal{A}_{n,k}$  l'ensemble des automates accessibles, sur un alphabet de taille  $|\Sigma| = k \geq 2$ , avec  $|Q| = n$  états.

*Korshunov, 1978; Bassino et Nicaud, 2007; Lebensztayn, 2010.*

$$|\mathcal{A}_{n,k}| = 2^n \omega_k \left\{ \begin{matrix} kn+1 \\ n \end{matrix} \right\} (1 + \mathcal{O}(n^{-1})) ,$$

où  $\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$  est le *nombre de Stirling de seconde espèce*, et  $\omega_k$  satisfait  $-k\omega_k = \ln(1 - \omega_k)$ .

On munit l'ensemble des automates accessibles  $A = (\delta, \mathcal{T})$  de la mesure  $\mu(\delta, \mathcal{T}) = \mu_{\text{uniforme}}(\delta) \times \mu_{\text{Bernoulli}}^{(b)}(\mathcal{T})$ .

*Bassino, David et Sportiello, 2012*

Quand  $n$  tend vers  $+\infty$ , la probabilité qu'un automate accessible  $A$  soit minimal tend vers

$$\exp\left(-\frac{1-2b(1-b)}{2} \omega_k^k n^{-k+2}\right) .$$

$$\mathbb{P}[i \sim j] = (\mathbb{P}[i, j \in \mathcal{T}] + \mathbb{P}[i, j \notin \mathcal{T}]) \\ \times \prod_{\alpha \in \Sigma} (\mathbb{P}[\delta(i, \alpha) = \delta(j, \alpha)] + \mathbb{P}[\delta(i, \alpha) \sim \delta(j, \alpha)])$$

- Si  $\mathcal{T}$  suit une distribution de Bernoulli,  $\mu_{\text{Bernoulli}}^{(b)}(\mathcal{T})$ , ( $\beta = 2b(1 - b)$ )

$$\mathbb{P}[i, j \in \mathcal{T}] + \mathbb{P}[i, j \notin \mathcal{T}] = 1 - \beta < 1.$$

- D'une façon naïve,  $\mathbb{P}[\delta(i, \alpha) = \delta(j, \alpha)] = 1/n$ 
  - Pour les application aléatoires, c'est vrai
  - Pour les fonctions de transition  $\delta$  accessibles, il y a un **facteur de correction**  $\omega_k$ .
- En posant  $p_{ij} = \mathbb{P}[i \sim j]$ ,

$$p_{ij} = (1 - \beta) \left( \frac{\omega_k}{n} + p_{ij} \right)^k$$

La seule solution  $0 < p_{ij} < 1$  est  $p_{ij} = (1 - \beta) \left( \frac{\omega_k}{n} \right)^k \left( 1 + \mathcal{O}\left(\frac{1}{n^{k-1}}\right) \right)$ .

$$\mathbb{P}[i \sim j] = (\mathbb{P}[i, j \in \mathcal{T}] + \mathbb{P}[i, j \notin \mathcal{T}]) \\ \times \prod_{\alpha \in \Sigma} (\mathbb{P}[\delta(i, \alpha) = \delta(j, \alpha)] + \mathbb{P}[\delta(i, \alpha) \sim \delta(j, \alpha)])$$

- Si  $\mathcal{T}$  suit une distribution de Bernoulli,  $\mu_{\text{Bernoulli}}^{(b)}(\mathcal{T})$ , ( $\beta = 2b(1 - b)$ )

$$\mathbb{P}[i, j \in \mathcal{T}] + \mathbb{P}[i, j \notin \mathcal{T}] = 1 - \beta < 1.$$

- D'une façon naïve,  $\mathbb{P}[\delta(i, \alpha) = \delta(j, \alpha)] = 1/n$ 
  - Pour les application aléatoires, c'est vrai
  - Pour les fonctions de transition  $\delta$  accessibles, il y a un **facteur de correction**  $\omega_k$ .
- En posant  $p_{ij} = \mathbb{P}[i \sim j]$ ,

$$p_{ij} = (1 - \beta) \left( \frac{\omega_k}{n} + p_{ij} \right)^k$$

La seule solution  $0 < p_{ij} < 1$  est  $p_{ij} = (1 - \beta) \left( \frac{\omega_k}{n} \right)^k \left( 1 + \mathcal{O}\left(\frac{1}{n^{k-1}}\right) \right)$ .

$$\mathbb{P}[i \sim j] = (\mathbb{P}[i, j \in \mathcal{T}] + \mathbb{P}[i, j \notin \mathcal{T}]) \\ \times \prod_{\alpha \in \Sigma} (\mathbb{P}[\delta(i, \alpha) = \delta(j, \alpha)] + \mathbb{P}[\delta(i, \alpha) \sim \delta(j, \alpha)])$$

- Si  $\mathcal{T}$  suit une distribution de Bernoulli,  $\mu_{\text{Bernoulli}}^{(b)}(\mathcal{T})$ , ( $\beta = 2b(1 - b)$ )

$$\mathbb{P}[i, j \in \mathcal{T}] + \mathbb{P}[i, j \notin \mathcal{T}] = 1 - \beta < 1.$$

- D'une façon naïve,  $\mathbb{P}[\delta(i, \alpha) = \delta(j, \alpha)] = 1/n$ 
  - Pour les application aléatoires, c'est vrai
  - Pour les fonctions de transition  $\delta$  accessibles, il y a un **facteur de correction**  $\omega_k$ .
- En posant  $p_{ij} = \mathbb{P}[i \sim j]$ ,

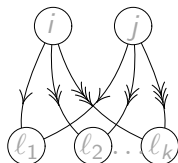
$$p_{ij} = (1 - \beta) \left( \frac{\omega_k}{n} + p_{ij} \right)^k$$

La seule solution  $0 < p_{ij} < 1$  est  $p_{ij} = (1 - \beta) \left( \frac{\omega_k}{n} \right)^k \left( 1 + \mathcal{O}\left(\frac{1}{n^{k-1}}\right) \right)$ .

$$p_{ij} = (1 - \beta) \left( \frac{\omega_k}{n} + p_{ij} \right)^k \quad \longrightarrow \quad p_{ij} \simeq (1 - \beta) \left( \frac{\omega_k}{n} \right)^k$$

signifie que la partie dominante de la probabilité  $p_{ij}$  est donnée par les paires  $(i, j)$  telles que  $\delta(i, \alpha) = \delta(j, \alpha)$  pour tout  $\alpha \in \Sigma$ .

Cela correspond au motif



*Conséquence importante !*

Un automate non-minimal a presque sûrement de **petits certificats** de non-minimalité.

D'après le **Paradigme de Poisson**, la somme  $X$  de plusieurs variables indicatrices, rares et presque indépendantes suit presque une loi de Poisson de moyenne  $\mathbb{E}[X]$ , et  $\mathbb{P}[X = 0] \simeq \exp(-\mathbb{E}[X])$ :

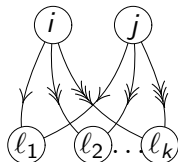
$$\mathbb{P}[A \text{ minimal}] \simeq \exp\left(-\binom{n}{2} p_{ij}\right),$$

ce qui correspond à notre théorème.

$$p_{ij} = (1 - \beta) \left( \frac{\omega_k}{n} + p_{ij} \right)^k \quad \longrightarrow \quad p_{ij} \simeq (1 - \beta) \left( \frac{\omega_k}{n} \right)^k$$

signifie que la partie dominante de la probabilité  $p_{ij}$  est donnée par les paires  $(i, j)$  telles que  $\delta(i, \alpha) = \delta(j, \alpha)$  pour tout  $\alpha \in \Sigma$ .

Cela correspond au motif



*Conséquence importante !*

Un automate non-minimal a presque sûrement de **petits certificats** de non-minimalité.

D'après le **Paradigme de Poisson**, la somme  $X$  de plusieurs variables indicatrices, rares et presque indépendantes suit presque une loi de Poisson de moyenne  $\mathbb{E}[X]$ , et  $\mathbb{P}[X = 0] \simeq \exp(-\mathbb{E}[X])$ :

$$\mathbb{P}[A \text{ minimal}] \simeq \exp\left(-\binom{n}{2} p_{ij}\right),$$

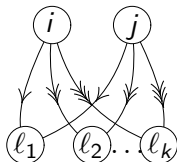
ce qui correspond à notre théorème.



$$p_{ij} = (1 - \beta) \left( \frac{\omega_k}{n} + p_{ij} \right)^k \quad \longrightarrow \quad p_{ij} \simeq (1 - \beta) \left( \frac{\omega_k}{n} \right)^k$$

signifie que la partie dominante de la probabilité  $p_{ij}$  est donnée par les paires  $(i, j)$  telles que  $\delta(i, \alpha) = \delta(j, \alpha)$  pour tout  $\alpha \in \Sigma$ .

Cela correspond au motif



*Conséquence importante !*

Un automate non-minimal a presque sûrement de **petits certificats** de non-minimalité.

D'après le **Paradigme de Poisson**, la somme  $X$  de plusieurs variables indicatrices, rares et presque indépendantes suit presque une loi de Poisson de moyenne  $\mathbb{E}[X]$ , et  $\mathbb{P}[X = 0] \simeq \exp(-\mathbb{E}[X])$ :

$$\mathbb{P}[A \text{ minimal}] \simeq \exp\left(-\binom{n}{2} p_{ij}\right),$$

ce qui correspond à notre théorème.

# Génération Aléatoire

# Générateurs de Boltzmann

(Duchon, Flajolet, Louchard et Schaeffer 2004)

- ▶ Un générateur de Boltzmann engendre des objets avec une distribution de probabilités proportionnelle à leur taille  
$$\mathbb{P}_x(\gamma) = C \frac{x^{|\gamma|}}{|\gamma|!} \text{ (ou } C x^{|\gamma|})$$
- ▶ Les objets engendrés n'ont pas tous la même taille, mais deux objets de même taille ont la même probabilité d'être engendrés.
- ▶ Le paramètre  $x$  est choisi en fonction de la taille moyenne visée.
- ▶ Un algorithme avec rejet peut être utilisé pour engendrer des objets de taille fixée. La complexité dépendra de la concentration de la taille autour de la moyenne.
- ▶ Très peu de précalcul et peu d'espace mémoire est utilisé.

# Générateur de Boltzmann pour les partitions d'ensemble

## But

Engendrer aléatoirement pour la distribution uniforme les partitions d'un ensemble à  $kn$  éléments en  $n$  sous-ensembles non vides.

- ▶ Une partition en  $n$  sous-ensembles non vides = ensemble de  $n$  ensembles non-vides
- ▶ La série exponentielle des ensembles non vides selon leur cardinal  $N(z) = \sum_{n \geq 1} \frac{z^n}{n!} = e^z - 1$ .
- ▶ Dans le modèle de Boltzmann, la taille de chacun des  $n$  sous-ensembles suit une loi de Poisson de paramètre  $x$  :  
 $\mathbb{P}_x(|\gamma| = s) = \frac{1}{(e^x - 1)} \frac{x^s}{s!}$ .
- ▶ La taille moyenne de la partition engendrée est

$$\mathbb{E}_x(\text{taille partition}) = nx \frac{e^x}{e^x - 1}.$$

- ▶  $\mathbb{E}_x(\text{taille partition}) = kn$  pour  $x = \zeta_k$  (point de col de  $\left\{ \begin{matrix} kn \\ n \end{matrix} \right\}$ )

# Générateurs de Boltzmann pour les partitions d'ensemble

- ▶ Engendrer la taille de chacun des  $n$  sous-ensembles suivant une loi de Poisson de paramètre  $x = \zeta_k$  (complexité linéaire).
- ▶ La probabilité que la partition engendrée soit de taille exactement  $kn$  est asymptotiquement  $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ .  
Le nombre moyen de rejets est donc  $\mathcal{O}(\sqrt{n})$ .
- ▶ Tirer une permutation aléatoire de  $\{1, \dots, kn\}$  pour étiqueter la structure (complexité linéaire)

La complexité moyenne de la génération d'une partition d'un ensemble à  $kn$  éléments en  $n$  sous-ensembles est en  $\mathcal{O}(n^{3/2})$ .

## Une amélioration récente (Bassino-Sportiello 2015)

- ▶ Peut-on faire mieux pour engendrer ces partitions ?
- ▶ **Oui !** Cela peut être réalisé en **temps linéaire** soit grâce une généralisation d'un algorithme de Devroye (2012) pour engendrer les arbres de Galton-Watson, soit par un algorithme mêlant méthode de Boltzmann et méthode récursive.

# Générateurs de Boltzmann pour les partitions d'ensemble

- ▶ Engendrer la taille de chacun des  $n$  sous-ensembles suivant une loi de Poisson de paramètre  $x = \zeta_k$  (complexité linéaire).
- ▶ La probabilité que la partition engendrée soit de taille exactement  $kn$  est asymptotiquement  $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ .  
Le nombre moyen de rejets est donc  $\mathcal{O}(\sqrt{n})$ .
- ▶ Tirer une permutation aléatoire de  $\{1, \dots, kn\}$  pour étiqueter la structure (complexité linéaire)

La complexité moyenne de la génération d'une partition d'un ensemble à  $kn$  éléments en  $n$  sous-ensembles est en  $\mathcal{O}(n^{3/2})$ .

## Une amélioration récente (Bassino-Sportiello 2015)

- ▶ Peut-on faire mieux pour engendrer ces partitions ?
- ▶ **Oui !** Cela peut être réalisé en **temps linéaire** soit grâce une généralisation d'un algorithme de Devroye (2012) pour engendrer les arbres de Galton-Watson, soit par un algorithme mêlant méthode de Boltzmann et méthode récursive.

## *Bassino-Sportiello 2015*

On peut engendrer aléatoirement des automates minimaux en temps linéaire en moyenne.

- ▶ Génération aléatoire d'automates déterministes et complets en temps linéaire,  
basée sur la génération en temps linéaire des partitions d'un ensemble à  $kn$  éléments en  $n$  sous-ensembles.
- ▶ Le rejet moyen est en  $\mathcal{O}(1)$   
car il y a une proportion constante d'automates minimaux.

# Analyse d'Algorithmes



# Analyse en moyenne de la minimisation (Moore)

- ▶ La complexité dans le pire cas de l'algorithme de minimisation dû à Moore est  $\mathcal{O}(n^2)$ .

*Bassino-David-Sportiello 2012*

La complexité moyenne de la minimisation avec l'algorithme de Moore est en  $\Theta(n \log \log n)$  pour la distribution uniforme sur les automates déterministes, complets et accessibles.

- ▶ La complexité de l'algorithme de minimisation de Moore pour la distribution uniforme sur les automates déterministes est en  $\mathcal{O}(n \log \log n)$  (*David 2011*).
- ▶ La borne supérieure vient de la distribution de la taille de la partie accessible d'un automate complet et déterministe aléatoire sur un alphabet à  $k$  lettres.
- ▶ La borne inférieure de la complexité de l'algorithme de Moore appliquée aux automates minimaux à  $n$  états est  $\Omega(n \log \log n)$
- ▶ Une proportion  $\mathcal{O}(1)$  d'automates accessibles sont minimaux.

# Analyse en moyenne de la minimisation (Moore)

- ▶ La complexité dans le pire cas de l'algorithme de minimisation dû à Moore est  $\mathcal{O}(n^2)$ .

## *Bassino-David-Sportiello 2012*

La complexité moyenne de la minimisation avec l'algorithme de Moore est en  $\Theta(n \log \log n)$  pour la distribution uniforme sur les automates déterministes, complets et accessibles.

- ▶ La complexité de l'algorithme de minimisation de Moore pour la distribution uniforme sur les automates déterministes est en  $\mathcal{O}(n \log \log n)$  (*David 2011*).
- ▶ La borne supérieure vient de la distribution de la taille de la partie accessible d'un automate complet et déterministe aléatoire sur un alphabet à  $k$  lettres.
- ▶ La borne inférieure de la complexité de l'algorithme de Moore appliquée aux automates minimaux à  $n$  états est  $\Omega(n \log \log n)$
- ▶ Une proportion  $\mathcal{O}(1)$  d'automates accessibles sont minimaux.

*Bassino-David-Sportiello 2012*

Pour la distribution uniforme sur les automates déterministes complets et accessibles, il existe des implantations de l'algorithme de Hopcroft dont la complexité moyenne est  $\mathcal{O}(n \log \log n)$ .

Il existe une implantation de l'algorithme de Hopcroft qui simule l'algorithme de Moore. (*David 2011*).

# Que reste-t-il à faire ?

- ▶ Enumérer les automates minimaux quand  $\mathbb{E}[|\mathcal{T}|] \rightarrow 0$  (par exemple,  $\mathbb{E}[|\mathcal{T}|] \sim n^\gamma$ ,  $0 < \gamma < 1$ ).
- ▶ Enumérer les automates minimaux incomplets.
- ▶ Étudier les automates minimaux résultant de l'intersection et de l'union de deux langages réguliers.
- ▶ Étudier les modèles de structures non-déterministes, la complexité en moyenne de la détermination.
- ▶ Étudier la complexité en moyenne des algorithmes issus de la recherche de motifs ou de la vérification et utilisant ce type de structures combinatoires.

Merci de votre attention !

Félicitations, Mireille !