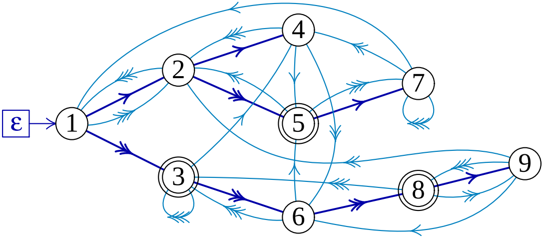
# Énumération asymptotique d'automates finis

**Frédérique Bassino**
Laboratoire d'Informatique de Paris Nord




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

Séminaire Philippe Flajolet, Mai 2012

# Pourquoi vouloir énumérer les automates finis ?

- Étudier de manière quantitative les propriétés des langages réguliers.
- Utiliser la représentation (canonique) des langages réguliers par des automates (minimaux).
- Enumérer, simuler ces structures.
- Analyser en moyenne des algorithmes manipulant ces structures
  (par exemple des algorithmes de minimisation)

# Pourquoi vouloir énumérer les automates finis ?

- Étudier de manière quantitative les propriétés des langages réguliers.
- Utiliser la représentation (canonique) des langages réguliers par des automates (minimaux).
- Enumérer, simuler ces structures.
- Analyser en moyenne des algorithmes manipulant ces structures
  (par exemple des algorithmes de minimisation)

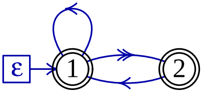
# Langages réguliers

- $\Sigma$ un alphabet fini: $\Sigma = \{a, b, c\}$
- $\Sigma^*$ l'ensemble des mots sur $\Sigma$: $\Sigma^* = \{\epsilon, a, b, c, aa, ab, \ldots\}$
- 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

$$X^* = \{x_1 x_2 \cdots x_n \mid n \geq 0, x_i \in X\}$$

Exemple : les *mots de Fibonacci*,

c'est à dire, les mots sur $\Sigma = \{a, b\}$ sans $bb$




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

Automate

Expression règulière

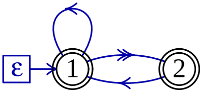
# Langages réguliers

- $\Sigma$ 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

$$X^* = \{x_1 x_2 \cdots x_n \mid n \geq 0, x_i \in X\}$$

Exemple : les *mots de Fibonacci*,

c'est à dire, les mots sur $\Sigma = \{a, b\}$ sans $bb$




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

Automate

Expression règulière

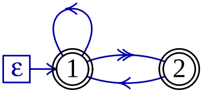
# Langages réguliers

- $\Sigma$ un alphabet fini: $\Sigma = \{a, b, c\}$
- $\Sigma^*$ l'ensemble des mots sur $\Sigma$: $\Sigma^* = \{\epsilon, a, b, c, aa, ab, \ldots\}$
- 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

$$X^* = \{x_1 x_2 \cdots x_n \mid n \geq 0, x_i \in X\}$$

Exemple : les *mots de Fibonacci*,

c'est à dire, les mots sur $\Sigma = \{a, b\}$ sans $bb$




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

Automate

Expression règulière

# Expressions rationnelles

Les langages finis 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

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

$$A = (\Sigma, Q, \delta, q_0, \mathcal{T})$$

Alphabet
Ensemble d'états
Fonction de transition
État initial
Ensemble d'états terminaux

- *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

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

$$A = (\Sigma, Q, \delta, q_0, \mathcal{T})$$

Alphabet

Ensemble d'états

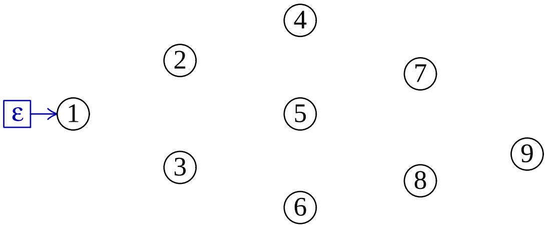
Fonction de transition

État initial

Ensemble d'états terminaux

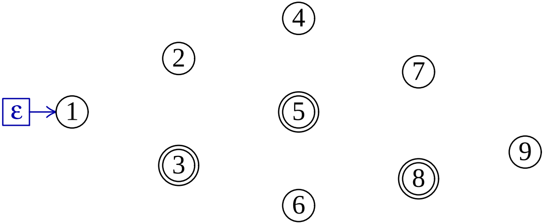
- *Notations.*
  Taille de l'alphabet : $|\Sigma| = k$. Nombre d'états : $|Q| = n$.
- On fait un choix canonique, $\Sigma \equiv [k]$, $Q \equiv [n]$ et $q_0 = 1$.
- Avec un ordre total sur $Q$, le couple $(\delta, \mathcal{T})$ permet de reconstruire l'automate $A$.

# Automates déterministes



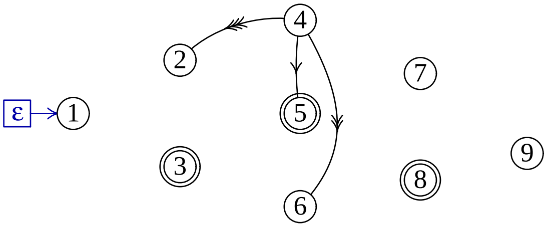

- Les états sont représentés par des cercles.
- Un état est final (dans $\mathcal{T}$) s'il est doublement cerclé.
- Pour chaque état $i \in Q$ et chaque lettre $\alpha \in \Sigma$, une transition sort de l'état $i$, vers l'état $\delta(i, \alpha)$.
- Pour chaque mot $w \in \Sigma^*$, $\delta(i, w)$ 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



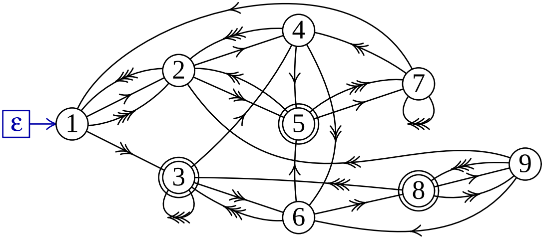

- Les états sont représentés par des cercles.
- Un état est final (dans $\mathcal{T}$) s'il est doublement cerclé.
- Pour chaque état $i \in Q$ et chaque lettre $\alpha \in \Sigma$, une transition sort de l'état $i$, vers l'état $\delta(i, \alpha)$.
- Pour chaque mot $w \in \Sigma^*$, $\delta(i, w)$ 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



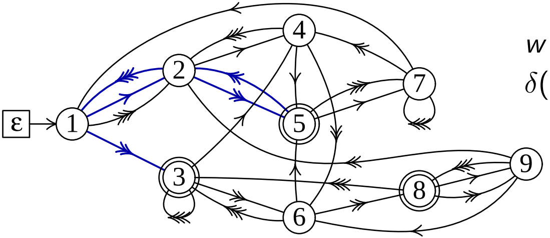

- Les états sont représentés par des cercles.
- Un état est final (dans $\mathcal{T}$) s'il est doublement cerclé.
- Pour chaque état $i \in Q$ et chaque lettre $\alpha \in \Sigma$, une transition sort de l'état $i$, vers l'état $\delta(i, \alpha)$.
- Pour chaque mot $w \in \Sigma^*$, $\delta(i, w)$ 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




- 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 $\delta(1, w) \in \mathcal{T}$.

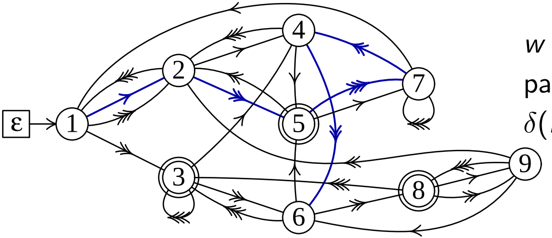
# Automates déterministes




$w = (abbcb)$ 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




$w = (abcbb)$ 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 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 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 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 complet de taille minimale qui le reconnaît : son automate minimal.

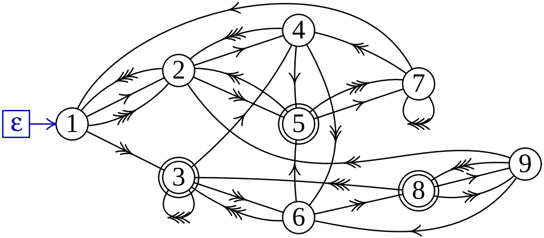
*Dans la suite, on va :*

- 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.

# Les automates : accessibilité

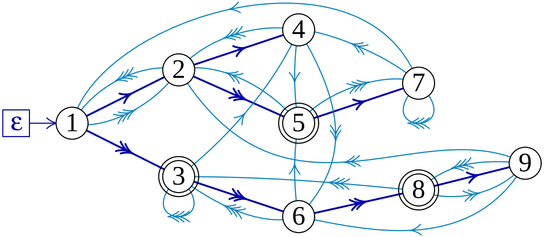



Un automate est **accessible** si, pour tout état $i$, il existe un mot $w$ tel que $\delta(1, w) = i$.

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



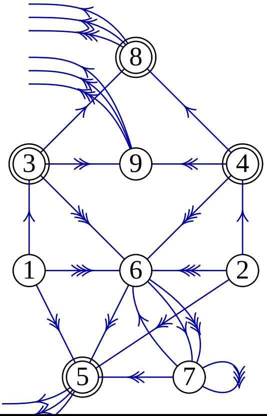

Un automate est **accessible** si, pour tout état $i$, il existe un mot $w$ tel que $\delta(1, w) = i$.

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

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 proprieté par un parcours en largeur simultané à partir des deux états.

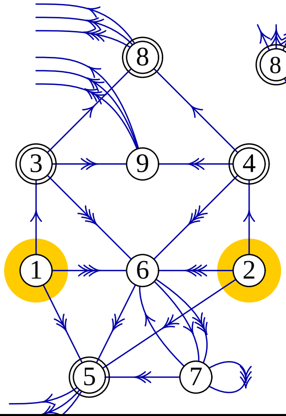


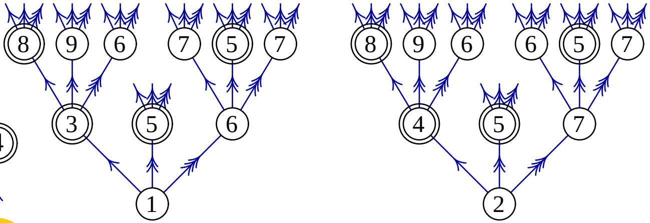

Un automate est **minimal** ssi il n'existe pas de couple d'états équivalents.

# 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 proprieté par un parcours en largeur simultané à partir des deux états.



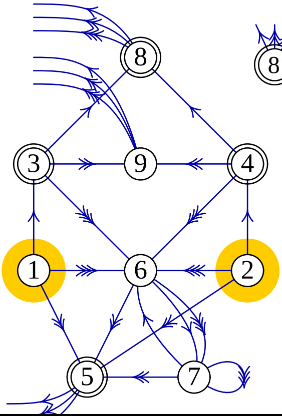



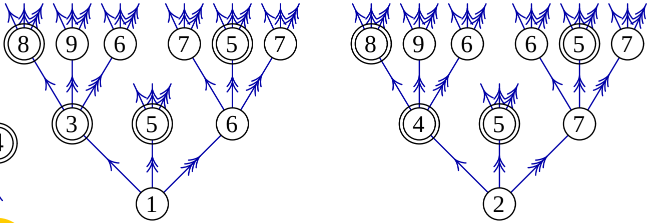

Un automate est **minimal** ssi il n'existe pas de couple d'états équivalents.

# 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 proprieté par un parcours en largeur simultané à partir des deux états.



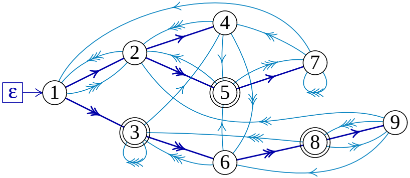


Un automate est **minimal** ssi il n'existe pas de couple d'états équivalents.

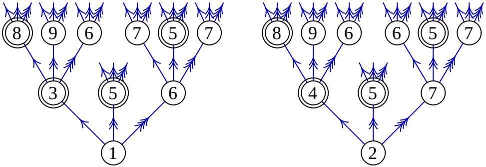
# Les automates : accessibilité et minimalité

Un automate est. . .

**accessible**, ssi les arbres de recherche issu de l'état initial couvre tous les états de l'automate




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

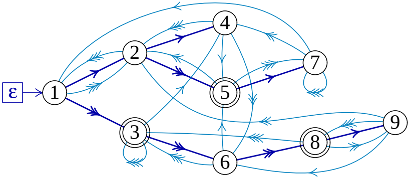



Il s'agit de contraintes non-locales et difficiles à exprimer d'un point de vue combinatoire.

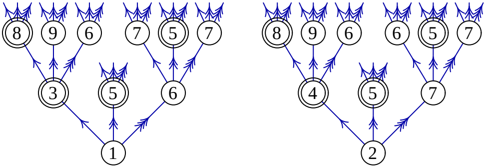
# Les automates : accessibilité et minimalité

Un automate est. . .

**accessible**, ssi les arbres de recherche issu de l'état initial couvre tous les états de l'automate




**minimal**, ssi il n'existe pas couple 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 et complets) 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 canoniquement étiquetés
- Asymptotiquement la moitié des automates unaires (déterministes, complets) accessibles sont minimaux.

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

Les automates unaires (déterministes et complets) vérifient

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

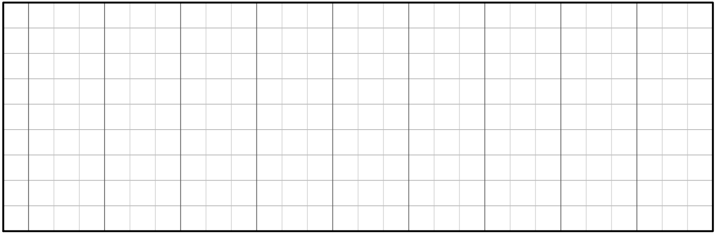
et sont donc caractérisés par le choix de l'état $c$ tel que $c = n \cdot a$.

*Nicaud 1999*

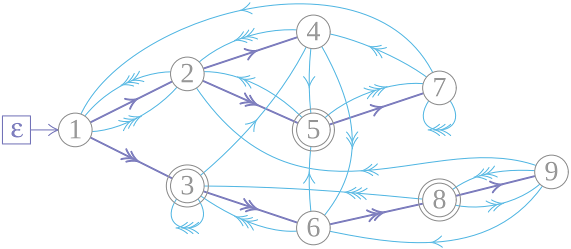
- Il y a $n\,2^n$ automates unaires canoniquement étiquetés
- Asymptotiquement la moitié des automates unaires (déterministes, complets) accessibles sont minimaux.

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

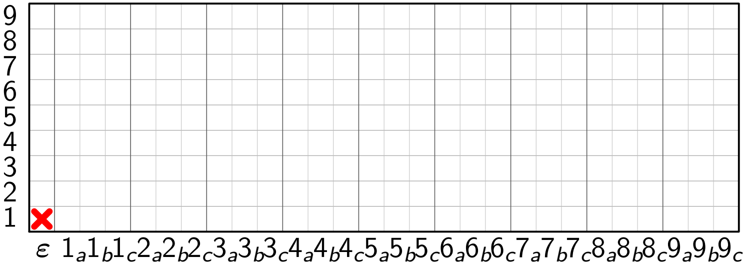
$n$

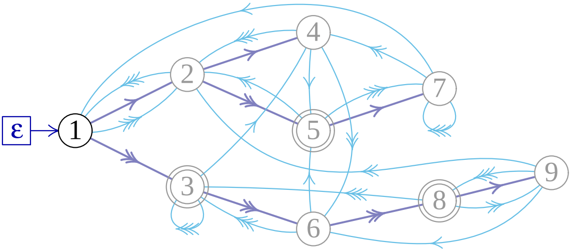


$kn + 1$

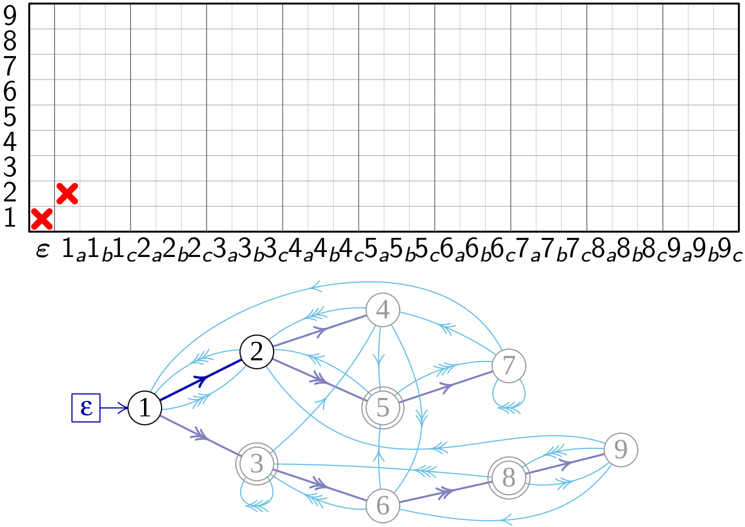


# Automates $k$-aires ($k \geq 2$) et tableaux

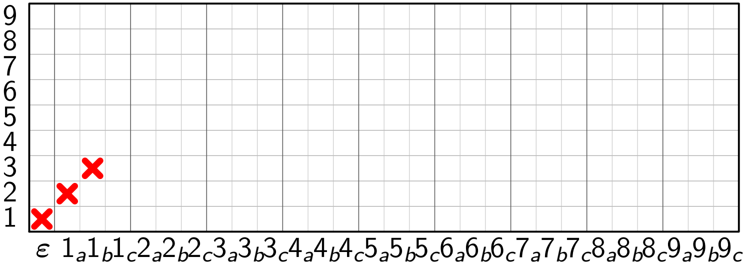


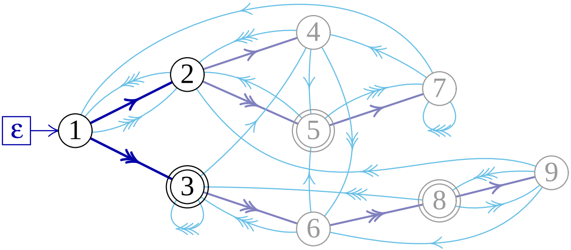



# Automates $k$-aires ($k \geq 2$) et tableaux

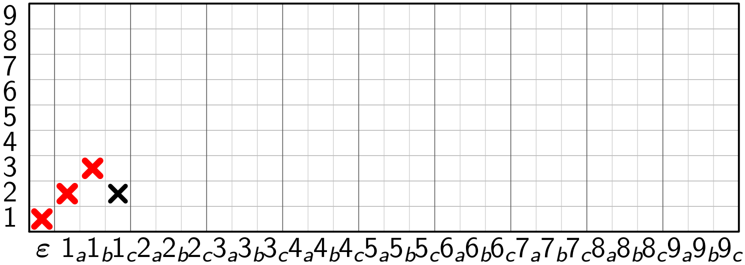


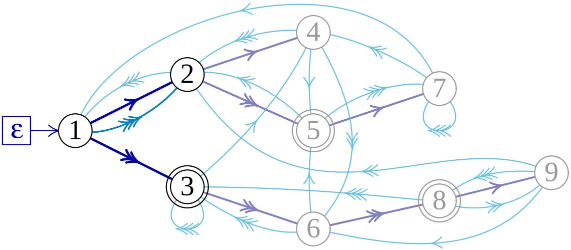
# Automates $k$-aires ($k \geq 2$) et tableaux



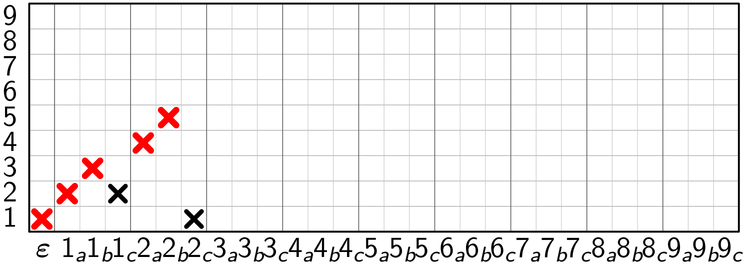



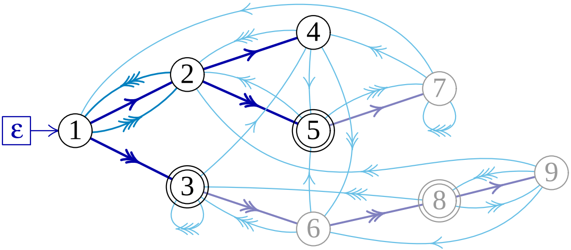
# Automates $k$-aires ($k \geq 2$) et tableaux



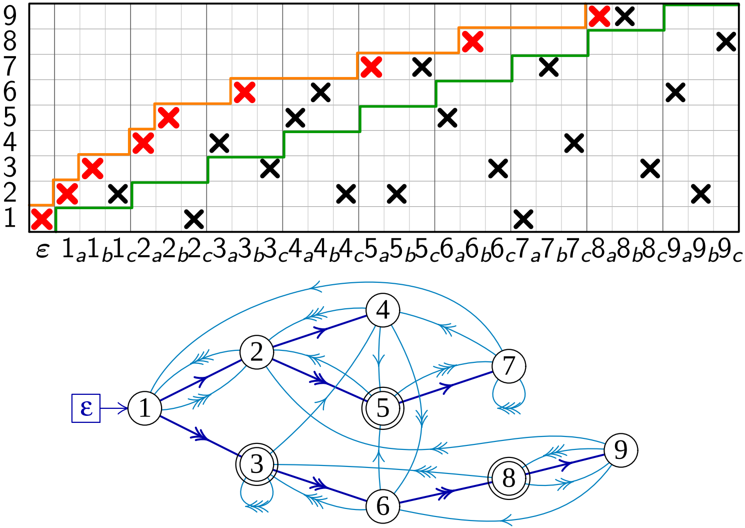



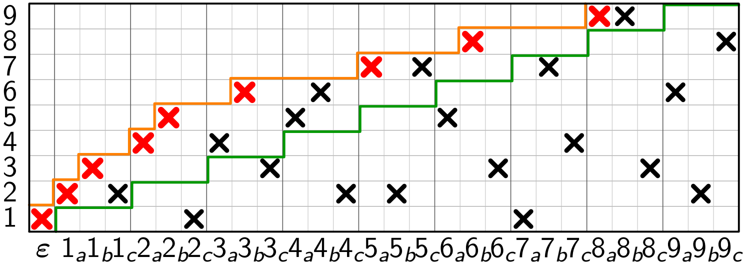
# Automates $k$-aires ($k \geq 2$) et tableaux



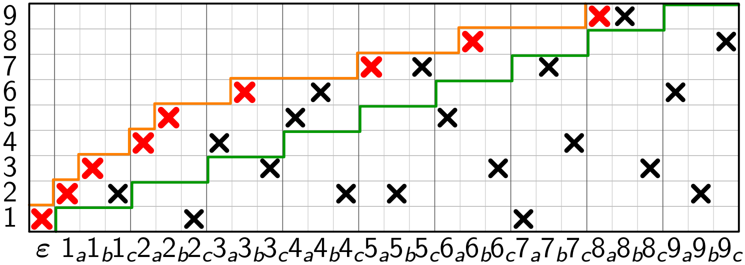
# Automates $k$-aires ($k \geq 2$) et tableaux





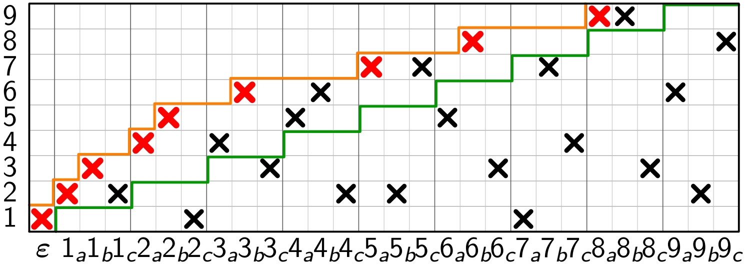

- Une croix par colonne

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



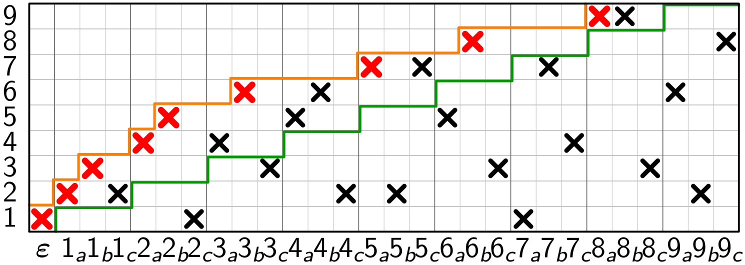

- Une croix par colonne
- Au moins une croix par ligne – la plus à gauche est rouge

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




- Une croix par colonne
- Au moins une croix par ligne – la plus à gauche est rouge
- Les croix rouges (*squelette*) forment une séquence monotone

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




- Une croix par colonne
- Au moins une croix par ligne – la plus à gauche est rouge
- Les croix rouges (*squelette*) forment une séquence monotone

- Enfin, il faut que les croix rouges restent à la gauche de la diagonale (*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.*

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

où $\left\{ {n \atop m} \right\}$ 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 $\left\{ {n \atop m} \right\}$ (Les *nombres de Stirling de seconde espèce* comptent le nombre de partitions n'un ensemble à $n$ éléments en $m$ sous-ensembles).
- 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.*

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

où $\left\{ {n \atop m} \right\}$ 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 $\left\{ {n \atop m} \right\}$
(Les *nombres de Stirling de seconde espèce* comptent le nombre de partitions n'un ensemble à $n$ éléments en $m$ sous-ensembles).
- 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.*

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

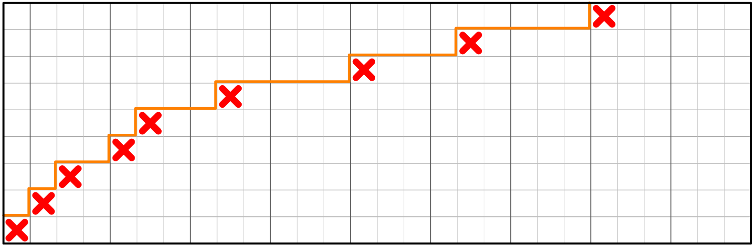
où $\left\{ {n \atop m} \right\}$ est le *nombre de Stirling de seconde espèce*, et $\omega_k$ satisfait $-k\omega_k = \ln(1 - \omega_k)$.

Qu'est ce qui se cache derrière ce théorème ?

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

# Relations entre squelette et croix noires et contrainte $k$-Dyck

*Q:* Combien de tableaux $T$ peuvent compléter un squelette donné ?



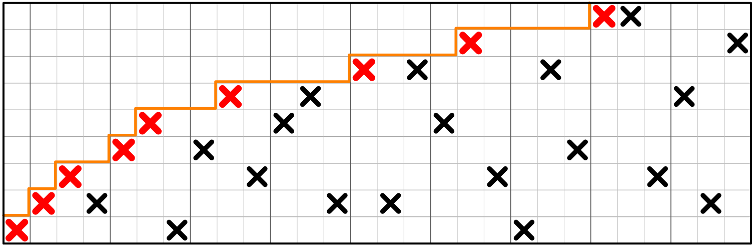
*R:* Code le squelette par la séquence $\mathbf{c} = \{c_y\}_{1 \le y \le n}$. C'est

$$\mu(\mathbf{c}) = \begin{cases} \prod_{y=1}^{n} y^{c_y} & \sum_y c_y = (k-1)n + 1 \\ 0 & \text{autrement} \end{cases}$$

*Corollaire :* $\sum_T \phi(\mathbf{c}(T)) = \sum_{\mathbf{c}} \mu(\mathbf{c})\phi(\mathbf{c})$.
La probabilité d'être $k$-Dyck est une fonction du type $\phi(\mathbf{c}(T))$.

# Relations entre squelette et croix noires et contrainte $k$-Dyck

*Q:* Combien de tableaux $T$ peuvent compléter un squelette donné ?



*R:* Code le squelette par la séquence $\mathbf{c} = \{c_y\}_{1 \le y \le n}$. C'est

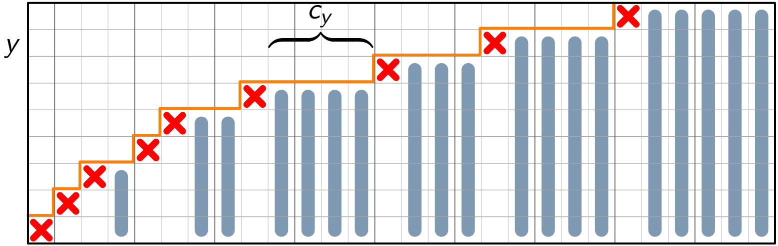
$$\mu(\mathbf{c}) = \begin{cases} \prod_{y=1}^{n} y^{c_y} & \sum_y c_y = (k-1)n + 1 \\ 0 & \text{autrement} \end{cases}$$

*Corollaire :* $\sum_T \phi(\mathbf{c}(T)) = \sum_{\mathbf{c}} \mu(\mathbf{c})\phi(\mathbf{c})$.

La probabilité d'être $k$-Dyck est une fonction du type $\phi(\mathbf{c}(T))$.

# Relations entre squelette et croix noires et contrainte $k$-Dyck

*Q:* Combien de tableaux $T$ peuvent compléter un squelette donné ?



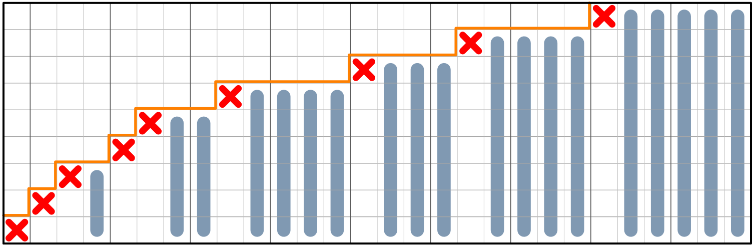

*R:* Code le squelette par la séquence $\mathbf{c} = \{c_y\}_{1 \le y \le n}$. C'est

$$\mu(\mathbf{c}) = \begin{cases} \prod_{y=1}^{n} y^{c_y} & \sum_y c_y = (k-1)n+1 \\ 0 & \text{autrement} \end{cases}$$

*Corollaire :* $\sum_T \phi(\mathbf{c}(T)) = \sum_{\mathbf{c}} \mu(\mathbf{c})\phi(\mathbf{c})$.
La probabilité d'être $k$-Dyck est une fonction du type $\phi(\mathbf{c}(T))$.

# Relations entre squelette et croix noires et contrainte $k$-Dyck

*Q:* Combien de tableaux $T$ peuvent compléter un squelette donné ?



*R:* Code le squelette par la séquence $\mathbf{c} = \{c_y\}_{1 \le y \le n}$. C'est

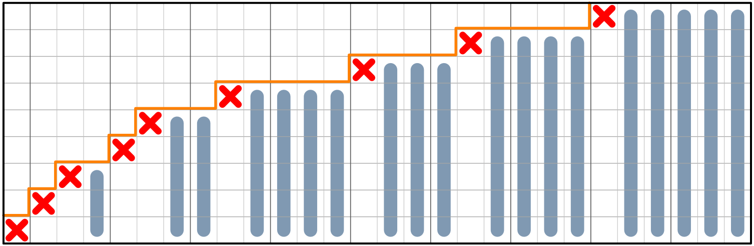
$$\mu(\mathbf{c}) = \begin{cases} \prod_{y=1}^{n} y^{c_y} & \sum_y c_y = (k-1)n + 1 \\ 0 & \text{autrement} \end{cases}$$

*Corollaire :* $\sum_T \phi(\mathbf{c}(T)) = \sum_{\mathbf{c}} \mu(\mathbf{c})\phi(\mathbf{c})$.

La probabilité d'être $k$-Dyck est une fonction du type $\phi(\mathbf{c}(T))$.

# Relations entre squelette et croix noires et contrainte $k$-Dyck

*Q:* Combien de tableaux $T$ peuvent compléter un squelette donné ?



*R:* Code le squelette par la séquence $\mathbf{c} = \{c_y\}_{1 \leq y \leq n}$. C'est

$$\mu(\mathbf{c}) = \begin{cases} \prod_{y=1}^{n} y^{c_y} & \sum_y c_y = (k-1)n+1 \\ 0 & \text{autrement} \end{cases}$$

*Corollaire :* $\sum_T \phi(\mathbf{c}(T)) = \sum_{\mathbf{c}} \mu(\mathbf{c})\phi(\mathbf{c})$.
La probabilité d'être $k$-Dyck est une fonction du type $\phi(\mathbf{c}(T))$.

# 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.*

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

où $\left\{ {n \atop m} \right\}$ 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 $\mu(\delta, \mathcal{T}) = \mu_{\text{uniforme}}(\delta) \times \mu^{(b)}_{\text{Bernoulli}}(\mathcal{T})$.

*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.*

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

où $\left\{ {n \atop m} \right\}$ 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 $\mu(\delta, \mathcal{T}) = \mu_{\text{uniforme}}(\delta) \times \mu^{(b)}_{\text{Bernoulli}}(\mathcal{T})$.

*Bassino, David et Sportiello, 2012*

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

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

# Calcul heuristique

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

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

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

D'une façon naïve, $\mathbb{P}[\delta(i,\alpha) = \delta(j,\alpha)] = 1/n$ (c'est vrai pour les application aléatoires). Parmi les fonctions de transition $\delta$ accessibles, on 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 $p_{ij} = (1-\beta)\left(\frac{\omega_k}{n}\right)^k\left(1 + \mathcal{O}\left(\frac{1}{n^{k-1}}\right)\right)$.

# Calcul heuristique

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

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

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

D'une façon naïve, $\mathbb{P}[\delta(i,\alpha) = \delta(j,\alpha)] = 1/n$ (c'est vrai pour les application aléatoires). Parmi les fonctions de transition $\delta$ accessibles, on 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 $p_{ij} = (1-\beta)\left(\frac{\omega_k}{n}\right)^k\left(1 + \mathcal{O}\left(\frac{1}{n^{k-1}}\right)\right)$.

# Calcul heuristique

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

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

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

D'une façon naïve, $\mathbb{P}[\delta(i,\alpha) = \delta(j,\alpha)] = 1/n$ (c'est vrai pour les application aléatoires). Parmi les fonctions de transition $\delta$ accessibles, on 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 $p_{ij} = (1-\beta)\left(\frac{\omega_k}{n}\right)^k\left(1 + \mathcal{O}\left(\frac{1}{n^{k-1}}\right)\right)$.

# Calcul heuristique

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

Ça veut dire que la partie dominante de la probabilité $p_{ij}$ est donnée par des couples $(i,j)$ tels que $\delta(i,\alpha) = \delta(j,\alpha)$ pour toutes $\alpha \in \Sigma$.
Cela correspond au motif

$i \qquad j$

$\ell_1 \qquad \ell_2 \ldots \ell_k$

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

En utilisant le Paradigme de Poisson - la somme de plusieurs variables indicatrices, rares et presque indépendantes suit presque une loi de Poisson -, $\mathbb{P}[A \text{ minimal}] \simeq \exp\left(-\binom{n}{2} p_{ij}\right)$,
ce qui correspond à notre théorème.

# Calcul heuristique

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

Ça veut dire que la partie dominante de la probabilité $p_{ij}$ est donnée par des couples $(i,j)$ tels que $\delta(i,\alpha) = \delta(j,\alpha)$ pour toutes $\alpha \in \Sigma$.

Cela correspond au motif

$i \quad j$

$\ell_1 \quad \ell_2 \ldots \ell_k$

*Conséquence importante !*

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

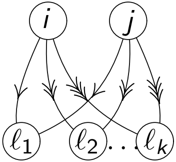
En utilisant le Paradigme de Poisson - la somme de plusieurs variables indicatrices, rares et presque indépendantes suit presque une loi de Poisson -, $\mathbb{P}[A \text{ minimal}] \simeq \exp\left(-\binom{n}{2} p_{ij}\right)$,
ce qui correspond à notre théorème.

# Calcul heuristique

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

Ça veut dire que la partie dominante de la probabilité $p_{ij}$ est donnée par des couples $(i,j)$ tels que $\delta(i,\alpha) = \delta(j,\alpha)$ pour toutes $\alpha \in \Sigma$.

Cela correspond au motif




*Conséquence importante !*

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

En utilisant le Paradigme de Poisson - la somme de plusieurs variables indicatrices, rares et presque indépendantes suit presque une loi de Poisson -, $\mathbb{P}[A \text{ minimal}] \simeq \exp\left(-\binom{n}{2} p_{ij}\right)$,

ce qui correspond à notre théorème. “□”

# Enumération des automates minimaux

*Bassino, David et Sportiello, 2011*

Pour $n$ grand, la probabilité que $A$ soit minimal converge vers

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

Certains motifs, dans les graphes orientés correspondant aux structures de transition, ont un rôle central dans la preuve.

$M$ $\quad i \quad j$ $\qquad\qquad$ $M^{(3)}$ $\quad i \quad j \quad h$

$\ell_1 \quad \ell_2 \ \ldots \ \ell_k$ $\qquad\qquad$ $\ell_1 \quad \ell_2 \ \ldots \ \ell_k$

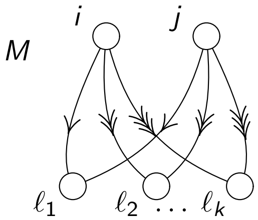
# Enumération des automates minimaux

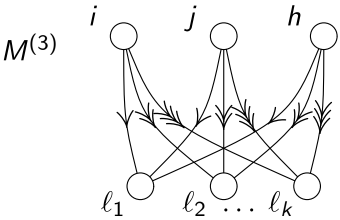
*Bassino, David et Sportiello, 2011*

Pour $n$ grand, la probabilité que $A$ soit minimal converge vers

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

Certains motifs, dans les graphes orientés correspondant aux structures de transition, ont un rôle central dans la preuve.



On va définir les quantités

- $P_{\mathrm{rare}}$ : probabilité que $A$ ne contienne pas de motifs $M$, et pourtant l'automate n'est pas minimal ;
- $P_{\mathrm{confl}}$ : probabilité que $A$ contienne des motifs $M^{(3)}$ ;
- $P(r)$ : probabilité que $A$ ne contienne pas des motifs $M^{(3)}$, et qu'il contienne exactement $r$ motifs $M$.

Pour une distribution de Bernoulli sur les états terminaux,

$$\mathbb{P}[A \text{ n'est pas minimal}] = \sum_{r\geq 1}\left(1-(2b(1-b))^r\right)P(r)+\mathcal{O}(P_{\mathrm{rare}}, P_{\mathrm{confl}})$$

Si on prouve que $\boxed{P_{\mathrm{rare}}, P_{\mathrm{confl}} = o(1-P(0))}$

et que $\boxed{P(r) = \mathrm{Poiss}_\rho(r)(1+o(1))}$ un certain $\rho$,

$$\mathbb{P}[A \text{ n'est pas minimal}] = \left(1-\exp(-\rho(1-2b(1-b)))\right)(1+o(1))\,.$$

# Quantités qui interviennent dans la preuve

On va définir les quantités

- $P_{\mathrm{rare}}$ : probabilité que $A$ ne contienne pas de motifs $M$, et pourtant l'automate n'est pas minimal ;
- $P_{\mathrm{confl}}$ : probabilité que $A$ contienne des motifs $M^{(3)}$ ;
- $P(r)$ : probabilité que $A$ ne contienne pas des motifs $M^{(3)}$, et qu'il contienne exactement $r$ motifs $M$.

Pour une distribution de Bernoulli sur les états terminaux,

$$\mathbb{P}[A \text{ n'est pas minimal}] = \sum_{r\geq 1} \left(1-(2b(1-b))^r\right) P(r) + \mathcal{O}(P_{\mathrm{rare}}, P_{\mathrm{confl}})$$

Si on prouve que $\boxed{P_{\mathrm{rare}}, P_{\mathrm{confl}} = o(1-P(0))}$

et que $\boxed{P(r) = \mathrm{Poiss}_\rho(r)(1+o(1))}$ un certain $\rho$,

$$\mathbb{P}[A \text{ n'est pas minimal}] = \left(1-\exp(-\rho(1-2b(1-b)))\right)(1+o(1)) .$$

# Le crible de Brun

N. Alon et J. Spencer, *The probabilistic method*

Soit $\{\mathcal{E}_j\}$ une collection d'évènements. On cherche $p(r) = \mathbb{P}[r \text{ des } \mathcal{E}_j \text{ sont vrais}]$. Soit

$$S^{(r)} = \sum_{(i_1,\ldots,i_r)} \mathbb{P}[\mathcal{E}_{i_1} \wedge \cdots \wedge \mathcal{E}_{i_r}]$$

**Crible de Brun**

S'il existe $\mu < +\infty$ tel que, pour $n$ grand, et pour chaque $r$,

$$S^{(r)} \to \frac{1}{r!}\mu^r$$

alors

$$p(r) \to \exp(-\mu)\frac{1}{r!}\mu^r .$$

# Estimations des motifs et de la boîte noire

En utilisant le crible de Brun, il suffit de montrer que

- **Estimation des $M$ motifs** :

$$S^{(r)} \sim \rho^r / r!, \text{ pour } \rho = \frac{1}{2}{\omega_k}^k n^{-k+2},$$

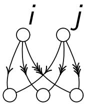
- **Estimation des $M^{(3)}$ motifs** :

$$P_{\text{confl}} \sim \frac{1}{6}{\omega_k}^{2k} n^{-2k+3}$$

- **Estimation de la “boîte noire”** :

$$P_{\text{rare}} = o(n^{-k+2}).$$

# Ingrédient de l'estimation de $P(r)$ et $P_{\text{confl}}$

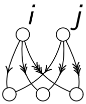
$Q$ : Quelle est la probabilité d'avoir un motif 

(avec $i < j$), conditionné à un squelette $\mathbf{c}$?

$R$ : Zéro, s'il y a des croix rouges dans les colonnes de $j$, $y^{-k}$ si ce n'est pas le cas.

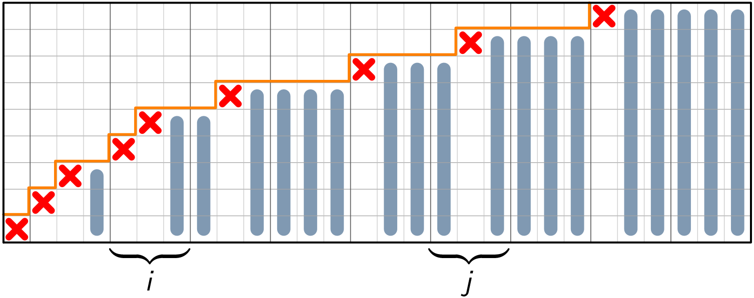
$$\mathbb{P}(M_{ij}|\mathbf{c})\,\mu(\mathbf{c}) \equiv \mu(\mathbf{c}') \qquad \mathbf{c}' = \mathbf{c} \setminus_{\text{colonnes de } j}$$

# Ingrédient de l'estimation de $P(r)$ et $P_{\text{confl}}$

$Q$ : Quelle est la probabilité d'avoir un motif  (avec $i < j$), conditionné à un squelette $\mathbf{c}$?



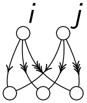
$R$ : Zéro, s'il y a des croix rouges dans les colonnes de $j$, $y^{-k}$ si ce n'est pas le cas.





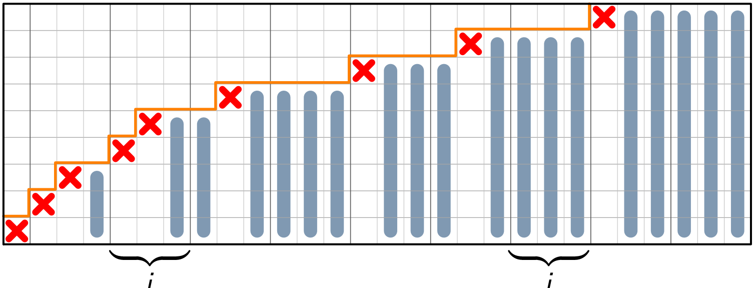
$\mathbb{P}(M_{ij}|\mathbf{c})\,\mu(\mathbf{c}) \equiv \mu(\mathbf{c}')$ $\qquad$ $\mathbf{c}' = \mathbf{c} \setminus_{\text{colonnes de } j}$

# Ingrédient de l'estimation de $P(r)$ et $P_{\mathrm{confl}}$

$Q$ : Quelle est la probabilité d'avoir un motif 

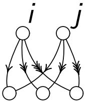
(avec $i < j$), conditionné à un squelette **c**?

$R$ : Zéro, s'il y a des croix rouges dans les colonnes de $j$, $y^{-k}$ si ce n'est pas le cas.



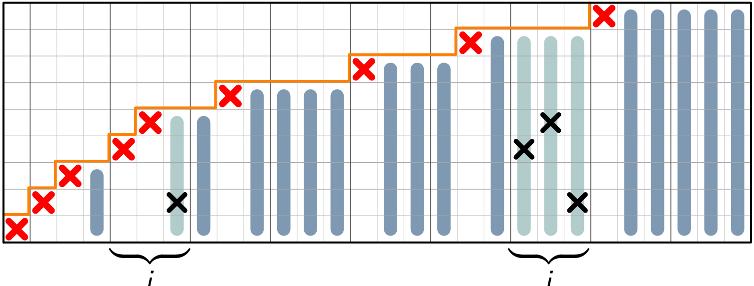

$$\mathbb{P}(M_{ij}|\mathbf{c})\,\mu(\mathbf{c}) \equiv \mu(\mathbf{c}') \qquad \mathbf{c}' = \mathbf{c}\setminus_{\text{colonnes de } j}$$

# Ingrédient de l'estimation de $P(r)$ et $P_{\text{confl}}$

$Q$ : Quelle est la probabilité d'avoir un motif  (avec $i < j$), conditionné à un squelette $\mathbf{c}$?



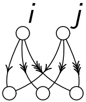
$R$ : Zéro, s'il y a des croix rouges dans les colonnes de $j$, $y^{-k}$ si ce n'est pas le cas.





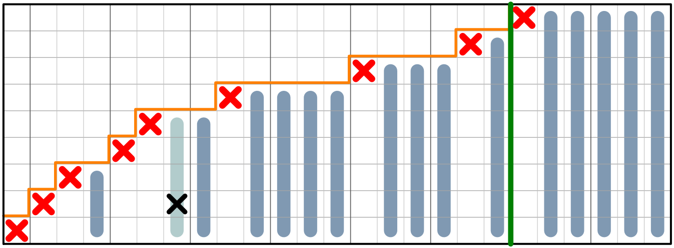
$$\mathbb{P}(M_{ij}|\mathbf{c})\,\mu(\mathbf{c}) \equiv \mu(\mathbf{c}') \qquad \mathbf{c}' = \mathbf{c} \setminus_{\text{colonnes de } j}$$

# Ingrédient de l'estimation de $P(r)$ et $P_{\text{confl}}$

$Q$ : Quelle est la probabilité d'avoir un motif 

(avec $i < j$), conditionné à un squelette $\mathbf{c}$?

$R$ : Zéro, s'il y a des croix rouges dans les colonnes de $j$, $y^{-k}$ si ce n'est pas le cas.



$$\mathbb{P}(M_{ij}|\mathbf{c})\,\mu(\mathbf{c}) \equiv \mu(\mathbf{c}') \qquad \mathbf{c}' = \mathbf{c} \setminus_{\text{colonnes de } j}$$

# Génération aléatoire

*Bassino-David-Sportiello 2012*

On peut engendrer aléatoirement des automates minimaux en temps moyen $\mathcal{O}(n^{3/2})$, avec un générateur de Boltzmann.

- Génération aléatoire d'automates accessibles en temps $\mathcal{O}(n^{3/2})$ (Bassino-Nicaud 2007),
  Basée sur un générateur de Boltzmann pour engendrer des partitions d'un ensemble à $kn$ éléments en $n$ sous-ensemble.
- Comme il y a une proportion constante d'automates minimaux, le rejet moyen est en $\mathcal{O}(1)$.

# Analyse en moyenne de la minimisation (Moore)

*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 à $n$ états est $\mathcal{O}(n \log \log n)$ *(David 2011)*.
- La borne supérieure vient de la distribution de la taille de la partie accessible d'une $k$-application aléatoire *(Carayol-Nicaud 2012)*
- La borne inférieure de l'algorithme de Moore appliquée aux automates minimaux à $n$ états est $\Omega(n \log \log n)$ *(Bassino-David- Nicaud 2009)*
- La conclusion vient de la probabilité qu'un automate accessible soit minimal.

# Analyse en moyenne de la minimisation (Hopcroft)

*Bassino-David-Sportiello 2012*

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

Il existe une famille d'implantations de l'algorithme de minimisation de Hopcroft qui sont toujours plus rapides que l'algorithme de Moore. *(David 2011)*.

# Que reste-t-il à faire ?

- Extension au cas des automates incomplets
- Extension au cas où $\mathbb{E}[|\mathcal{T}|] \to 0$ (par example, $\mathbb{E}[|\mathcal{T}|] \sim n^\gamma$, $0 < \gamma < 1$).
- Calcul de propriétés statistiques plus subtiles, en particulier la distribution de la profondeur de l'arbre de recherche (à partir de l'état initial).