Génération aléatoire d'automates non-déterministes intéressants

Julien David, Cyril Nicaud

Laboratoire d'Informatique de Paris Nord

Mercredi 14 décembre 2016

Automates finis

Automates finis

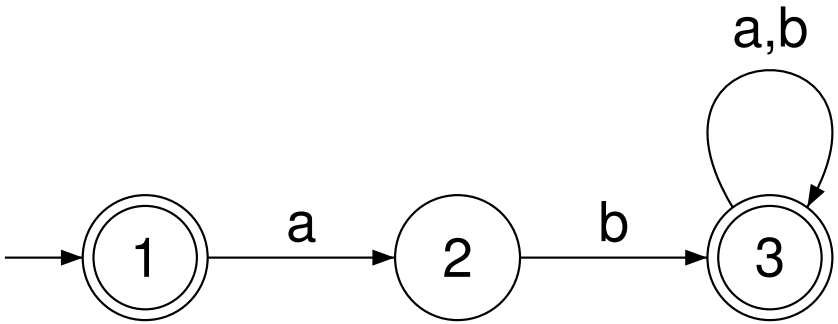
Définition

Un automate $\mathcal{A}$ est un quintuplet $< Q, \Sigma, I, F, T >$ où :

- Q est un ensemble d'états,
- Σ est un alphabet fini,
- $I \subseteq Q$ est l'ensemble des états initiaux,
- $F \subseteq Q$ est l'ensemble des états finals,
- $T \subseteq Q \times \Sigma \times Q$ est l'ensemble des transitions.

Automates finis

Un automate finis à n états sur un alphabet à k lettres.



- Ensemble d'états : $Q = \{1, 2, 3, 4\}$
- Alphabet : $\Sigma = \{a, b\}$
- Ensemble d'états initiaux : $\{1\}$
- Ensemble d'états terminaux : $F = \{1, 3\}$
- Langage reconnu par l'automate : $L = \varepsilon + a.b.\Sigma^*$

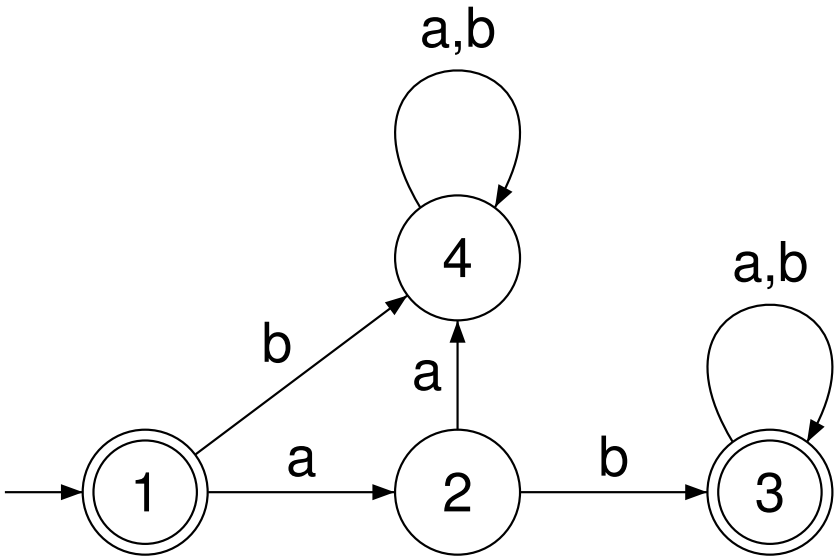
Propriétés

- Les automates sont des machines à états finis reconnaissant les langages rationnels.
- Les langages rationnels sont les langages obtenus à partir
  - d'un alphabet fini,
  - de l'ensemble vide et du mot vide ε,
  - l'union,
  - le produit de concaténation
  - l'étoile de Kleene : $X^* = \bigcup_{n \geq 0} X^n$
- Les langages rationnels sont également clos par intersection, par complémentaire

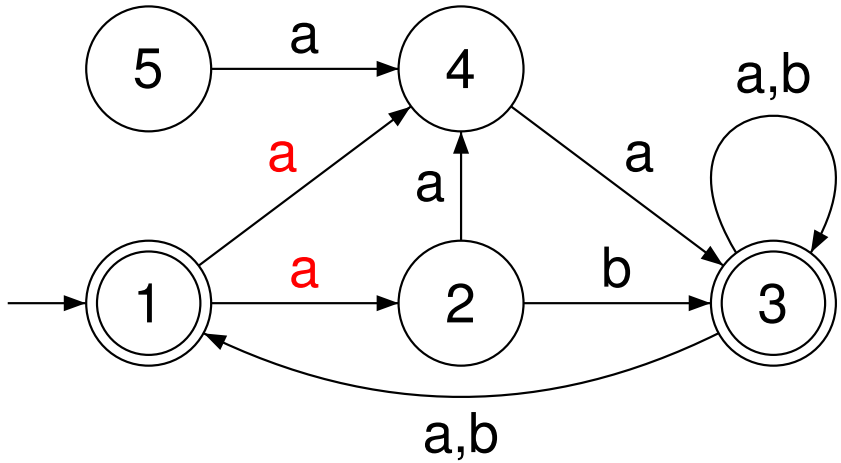
Automates déterministes

Déterministe

Un automate est déterministe s'il contient au plus un état initial et si pour tout état $q \in Q$ et toute lettre $a \in \Sigma$, il existe au plus un état $p \in Q$ telle que $(q, a, p) \in T$.



Automate déterministe

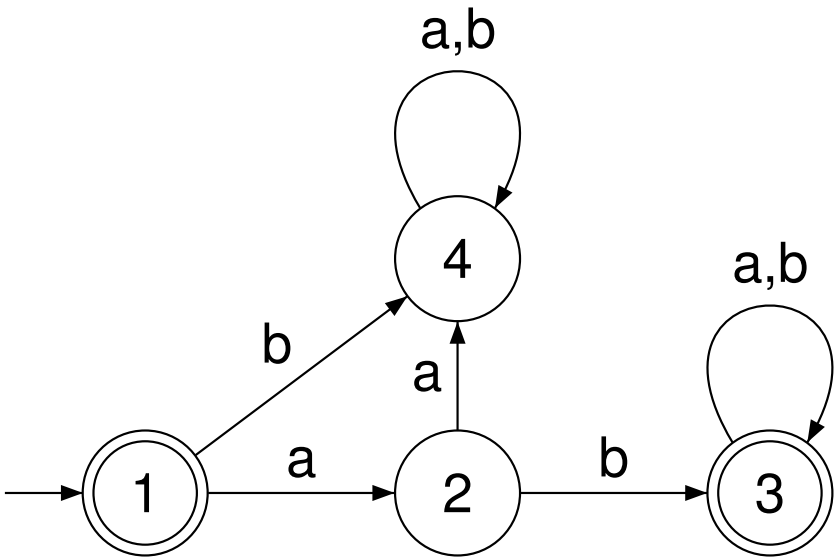


Automate non-déterministe

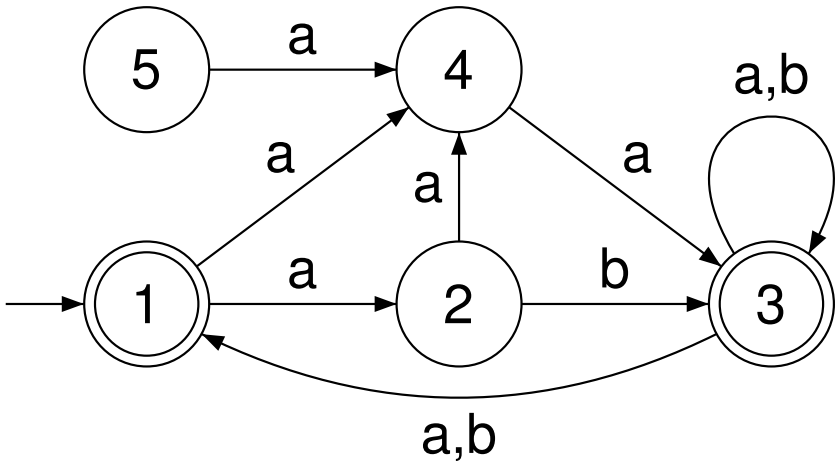
Automates complets

Complet

Un automate est complet si pour tout état $q \in Q$ et toute lettre $a \in \Sigma$, il existe au mois un état $p \in Q$ telle que $(q, a, p) \in T$.



Automate déterministe complet



Automate non-déterministe incomplet

Automates déterministes complets

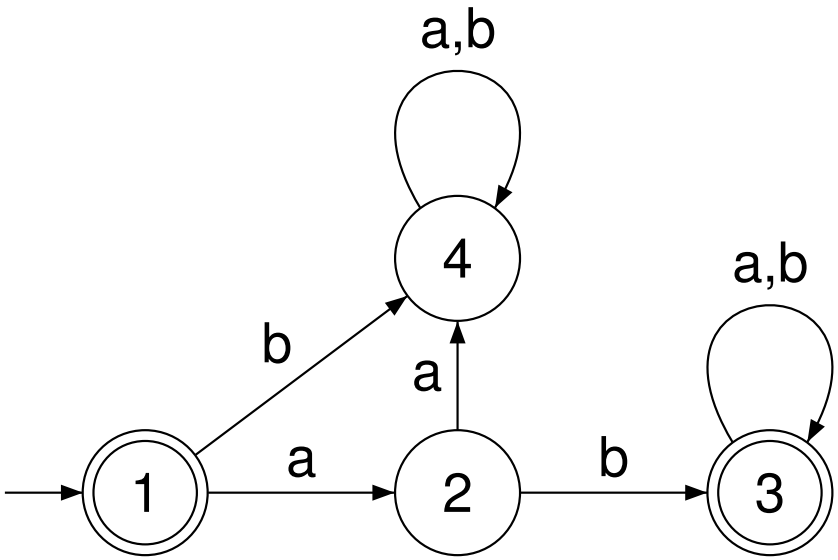
Énumération

- Un automate déterministe complet à n états sur un alphabet à k lettres à exactement kn transitions.
- Le nombre d'automates déterministes complets à n états est $n^{kn}2^n$.
- En comparaisons, le nombre d'automates à n états est 2^{kn^2+2n}.

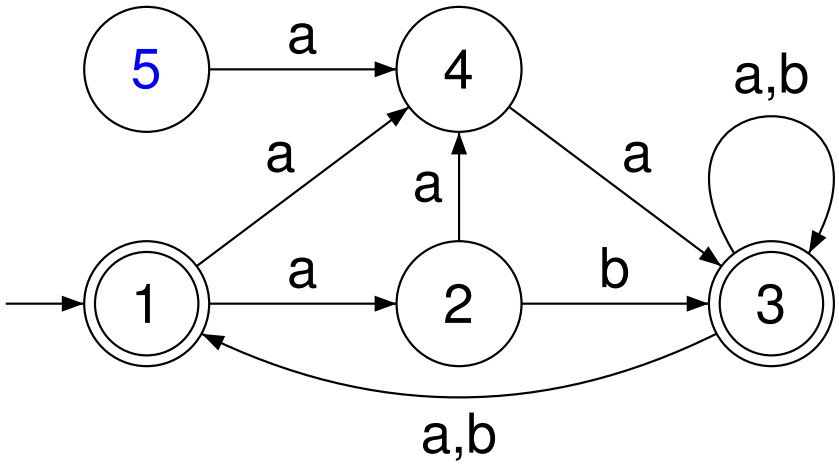
Automates accessibles

Accessible

Un automate est accessible si pour tout état $q \in Q$ il existe un chemin qui part de l'état initial et qui passe par q.



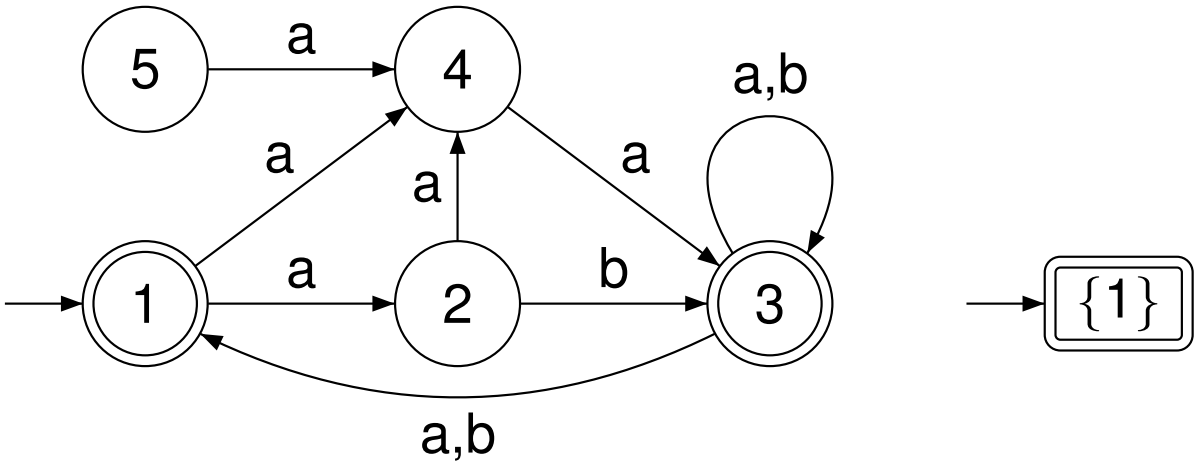
Automate déterministe accessible complet



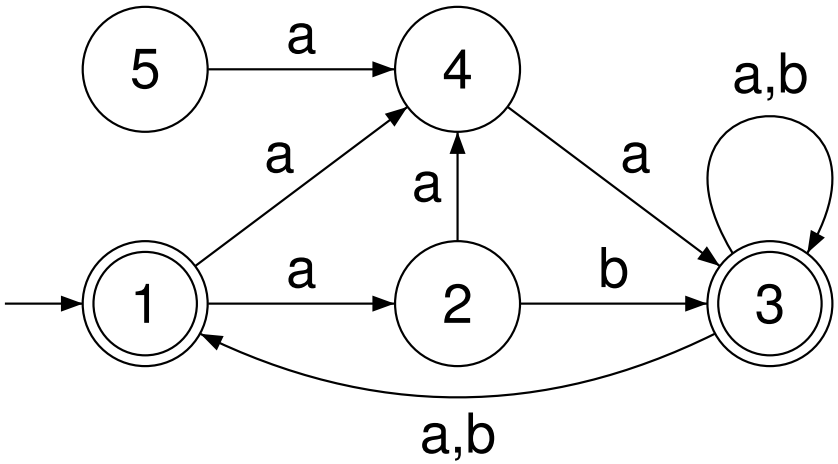
Automate non-déterministe non-accessible incomplet

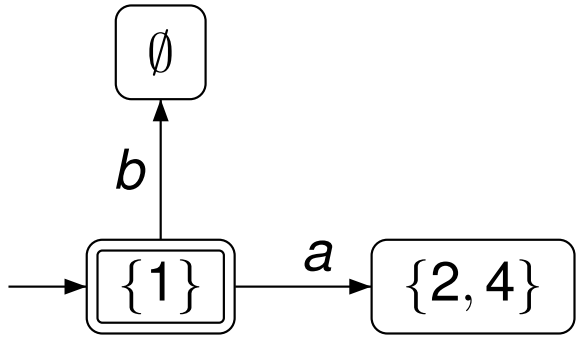
Déterminisation

Déterminisation

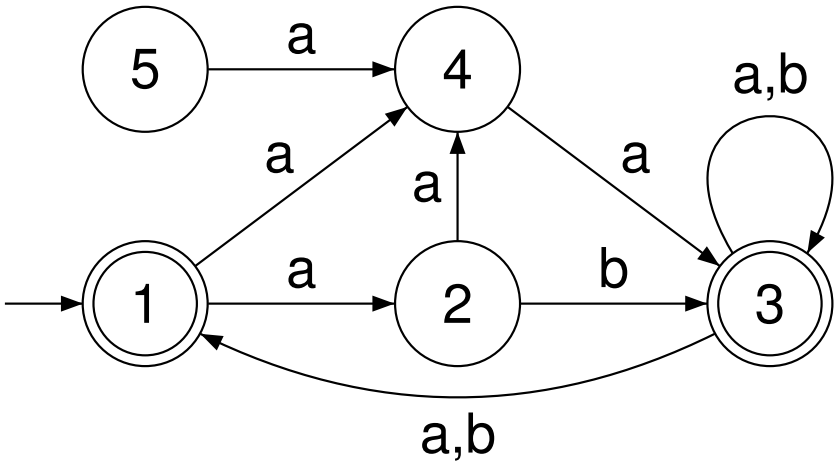


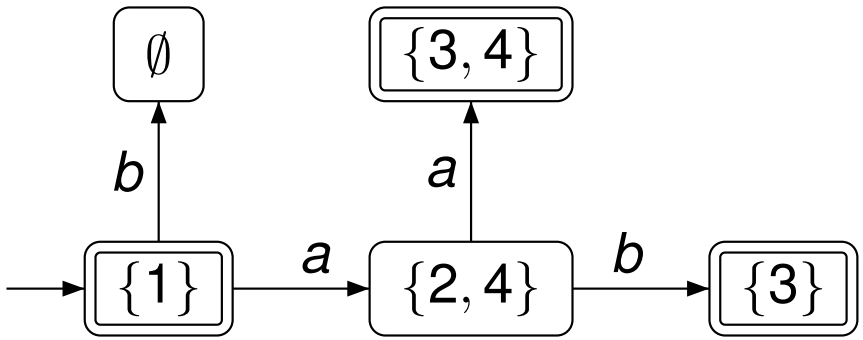
Déterminisation



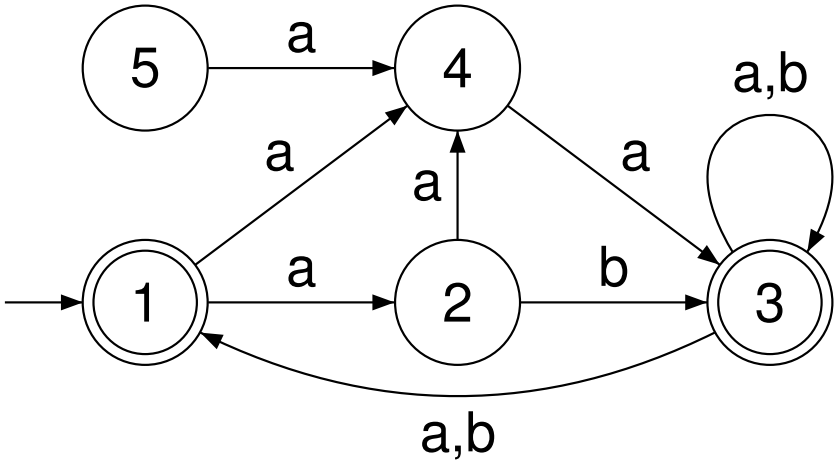


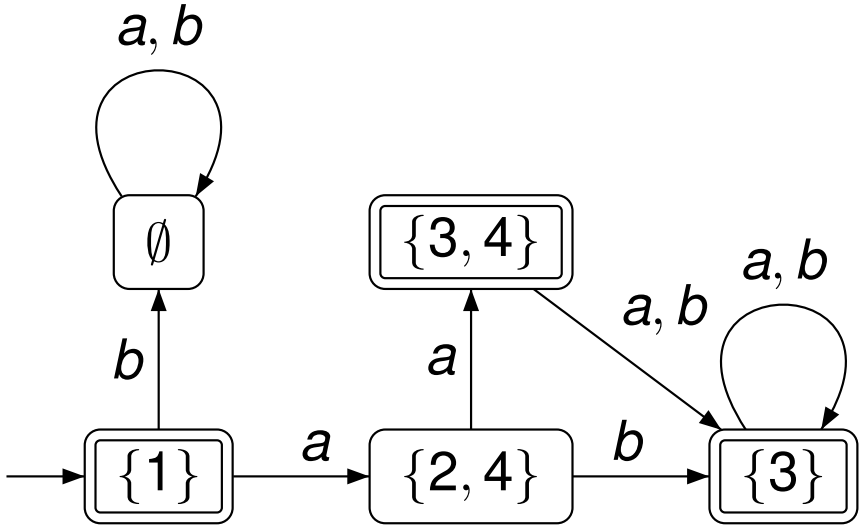
Déterminisation





Déterminisation

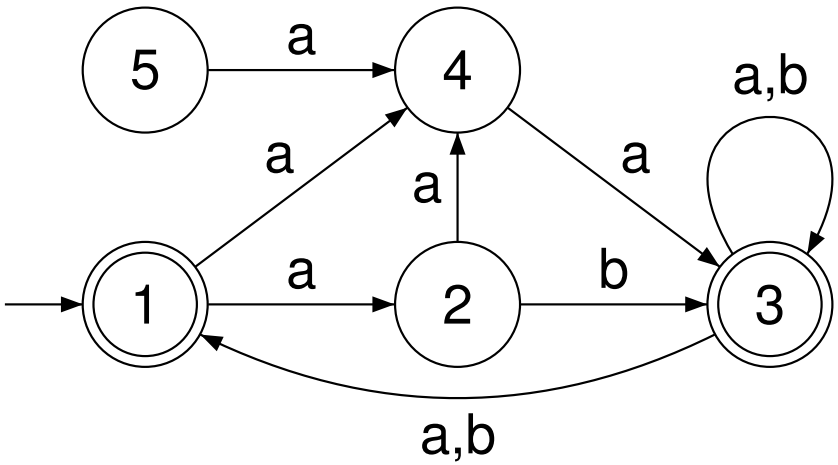


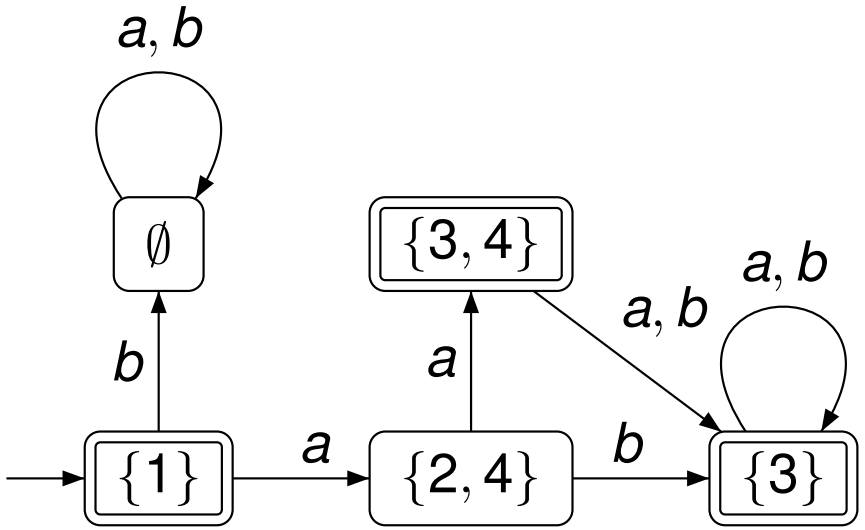


Résultat

- L'automate obtenu est déterministe accessible complet.
- Son nombre d'états peut être exponentiel dans celui de l'automate d'entrée.

Déterminisation





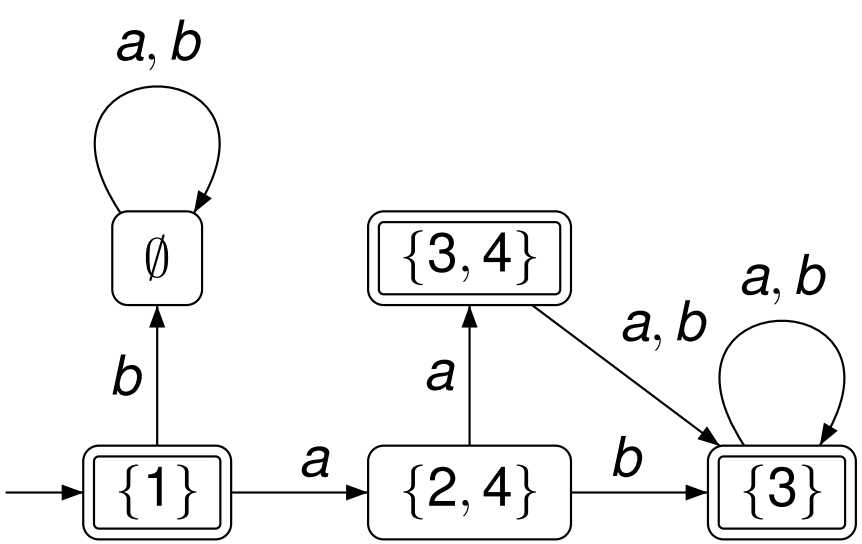
Résultat

- L'automate obtenu est déterministe accessible complet.
- Son nombre d'états peut être exponentiel dans celui de l'automate d'entrée.

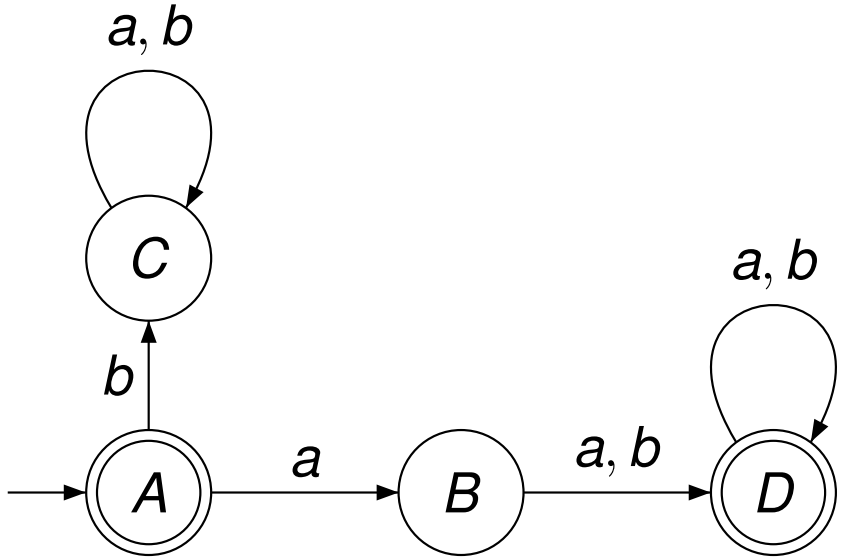
Automates minimaux

- Pour un langage rationnel donné, il existe une infinité d'automates finis reconnaissant ce langage.
- Pour tout langage rationnel, il existe un unique automate déterministe accessible complet qui reconnaît ce langage, tel que le nombre d'état est minimal.
  On parle de l'**automate minimal** du langage.
- La taille de l'automate minimal est une bonne notion de taille pour un langage rationnel. On parle de **complexité en états**.

Automates minimaux



Automate déterminisé



Automate minimal

Le langage reconnu est l'ensemble des mots de longueur au moins 2 commençant par *a* (plus le mot vide).

Génération aléatoire d'objets combinatoires

Objet combinatoire

Un objet combinatoire est un ensemble $\mathcal{O}$ muni d'une fonction de taille telle que le nombre d'objets d'une taille donnée est fini.

Génération aléatoire d'objets combinatoires

- Un générateur aléatoire est un algorithme permettant d'engendrer des objets combinatoires en suivant une distribution de probabilités préalablement fixée.
- Le plus souvent, l'objectif sera de garantir la distribution uniforme.

Génération aléatoire d'objets combinatoires

Objet combinatoire

Un objet combinatoire est un ensemble $\mathcal{O}$ muni d'une fonction de taille telle que le nombre d'objets d'une taille donnée est fini.

Génération aléatoire d'objets combinatoires

- Un générateur aléatoire est un algorithme permettant d'engendrer des objets combinatoires en suivant une distribution de probabilités préalablement fixée.
- Le plus souvent, l'objectif sera de garantir la distribution uniforme.

Génération aléatoire d'objets combinatoires

Pourquoi faire ?

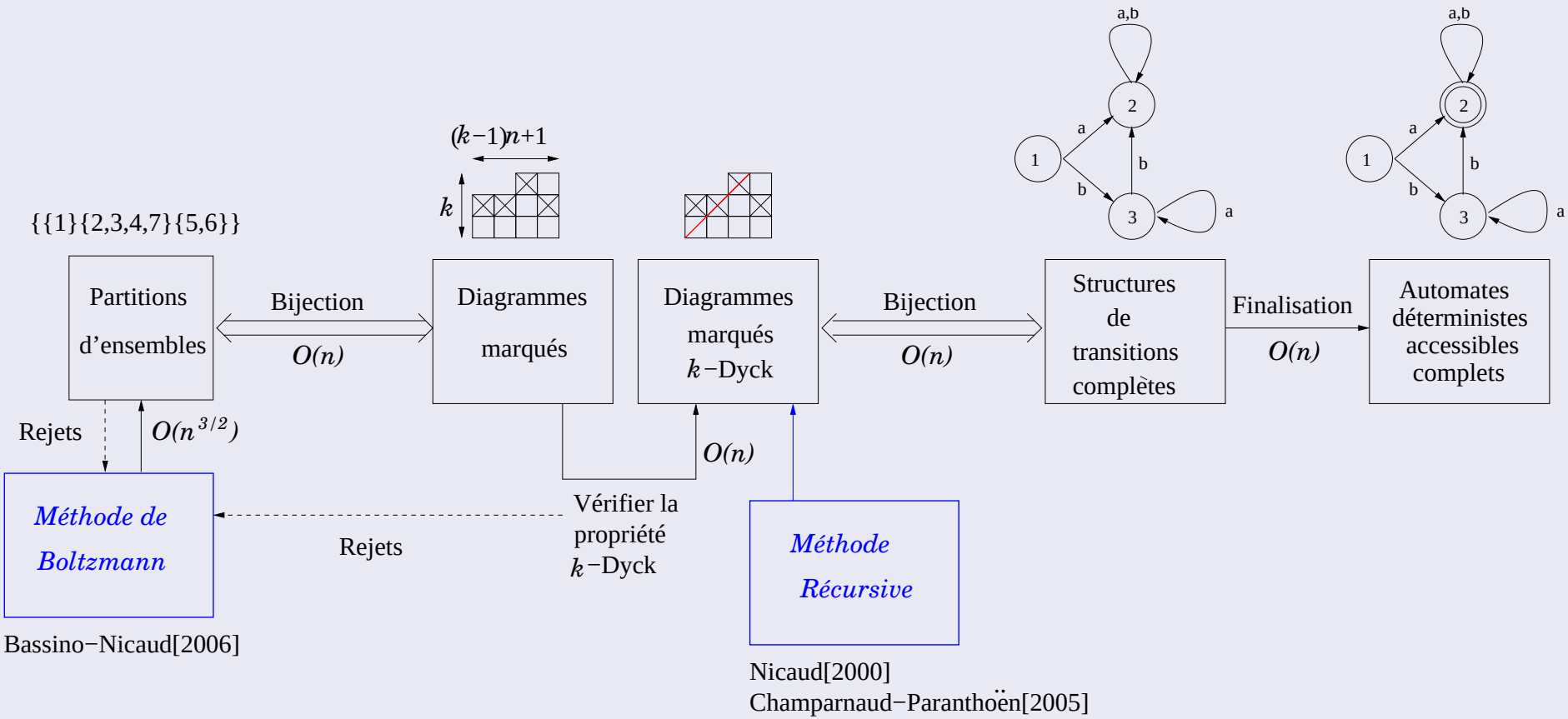
- Étudier les propriétés moyennes ou génériques des objets engendrés.
- Étudier le comportement moyen des algorithmes qui s'y appliquent.

État de l'art

Génération aléatoire d'automates déterministes accessibles complets.

Génération aléatoire d'automates déterministes : les algorithmes

De 2000 à 2006



Algorithme de Carayol-Nicaud

Théorème (Carayol-Nicaud 2012)

Pour la distribution uniforme sur l'ensemble des automates déterministes complets à n états sur un alphabet de taille k, la distribution de la taille de la partie accessible tends vers une loi Gausienne centrée en $\alpha_k n$ *(*$\alpha_k = 1 + \frac{1}{k} W_0(-ke^{-k})$ *) et d'écart-type* $\sigma_k \sqrt{n}$.

Corollaire

- Les auteurs engendrent des automates déterministes complets de taille $\frac{n}{\alpha_k}$ et gardent la partie accessible si celle ci est de taille n. Sinon l'automate est rejeté et on recommence.
- La complexité moyenne de cette méthode est $\Theta(n\sqrt{n})$.

Algorithme de Carayol-Nicaud

Théorème (Carayol-Nicaud 2012)

Pour la distribution uniforme sur l'ensemble des automates déterministes complets à n états sur un alphabet de taille k, la distribution de la taille de la partie accessible tends vers une loi Gausienne centrée en $\alpha_k n$ *(*$\alpha_k = 1 + \frac{1}{k} W_0(-ke^{-k})$ *) et d'écart-type* $\sigma_k \sqrt{n}$.

Corollaire

- Les auteurs engendrent des automates déterministes complets de taille $\frac{n}{\alpha_k}$ et gardent la partie accessible si celle ci est de taille n. Sinon l'automate est rejeté et on recommence.
- La complexité moyenne de cette méthode est $\Theta(n\sqrt{n})$.

Algorithme de Carayol-Nicaud

Théorème (Carayol-Nicaud 2012)

Pour la distribution uniforme sur l'ensemble des automates déterministes complets à n états sur un alphabet de taille k, la distribution de la taille de la partie accessible tends vers une loi Gausienne centrée en $\alpha_k n$ *(*$\alpha_k = 1 + \frac{1}{k} W_0(-ke^{-k})$ *) et d'écart-type* $\sigma_k \sqrt{n}$.

Berend-Kontorovich 2016

Les auteurs obtiennent un résultat moins précis, mais en utilisant une méthode probabiliste.

- Les auteurs montrent que la probabilité que la taille de la partie accessible ne soient pas dans l'intervalle $[\alpha_k n - \sqrt{n} \log n, \alpha_k n + \sqrt{n} \log n]$ est en $\Theta\left(\frac{1}{n^k}\right)$

Génération aléatoire d'automates déterministes : quelques résultats

Complexité pire cas des algorithmes de minimisation

Sur les automates déterministes accessibles complets :

- Algorithme de Moore $\mathcal{O}(n^2)$
- Algorithme de Hopcroft $\mathcal{O}(n \log n)$

Sur n'importe quel automate :

- Algorithme de Brzozowski $\mathcal{O}(n2^n)$

Complexité moyenne/générique des algorithmes de minimisation

- Bassino-D-Nicaud (2009) : algorithme de Moore $\mathcal{O}(n \log n)$
- D. (2011) : algorithme de Moore et de Hopcroft en $\mathcal{O}(n \log \log n)$
- Nicaud-De Felice (2016) : algorithme de Brzozowski $\Omega(n^{\log n})$

Génération aléatoire d'automates déterministes : quelques résultats

Complexité pire cas des algorithmes de minimisation

Sur les automates déterministes accessibles complets :

- Algorithme de Moore $\mathcal{O}(n^2)$
- Algorithme de Hopcroft $\mathcal{O}(n \log n)$

Sur n'importe quel automate :

- Algorithme de Brzozowski $\mathcal{O}(n2^n)$

Complexité moyenne/générique des algorithmes de minimisation

- Bassino-D-Nicaud (2009) : algorithme de Moore $\mathcal{O}(n \log n)$
- D. (2011) : algorithme de Moore et de Hopcroft en $\mathcal{O}(n \log \log n)$
- Nicaud-De Felice (2016) : algorithme de Brzozowski $\Omega(n^{\log n})$

État de l'art

Génération aléatoire d'automates non-déterministes.

Modèles aléatoires d'automates non-déterministes

Intérêts

- Les automates non-déterministes permettent de représenter des langages dont la complexité en états est exponentiellement plus grande.
- l'étude de l'algorithme de déterminisation, central en théorie des automates.

Précision : dans tout ce qui va suivre, les automates auront un unique état initial, l'état 1.

Génération aléatoire d'automates non-déterministes : la problématique

Un résutat négatif

Champarnaud, Hansel, Paranthoën, Ziadi (2004) : le déterminisé d'un automate aléatoire à n états sur un alphabet à k lettres a presque sûrement moins de $k + 2$ états.

Génération aléatoire d'automates non-déterministes : la problématique

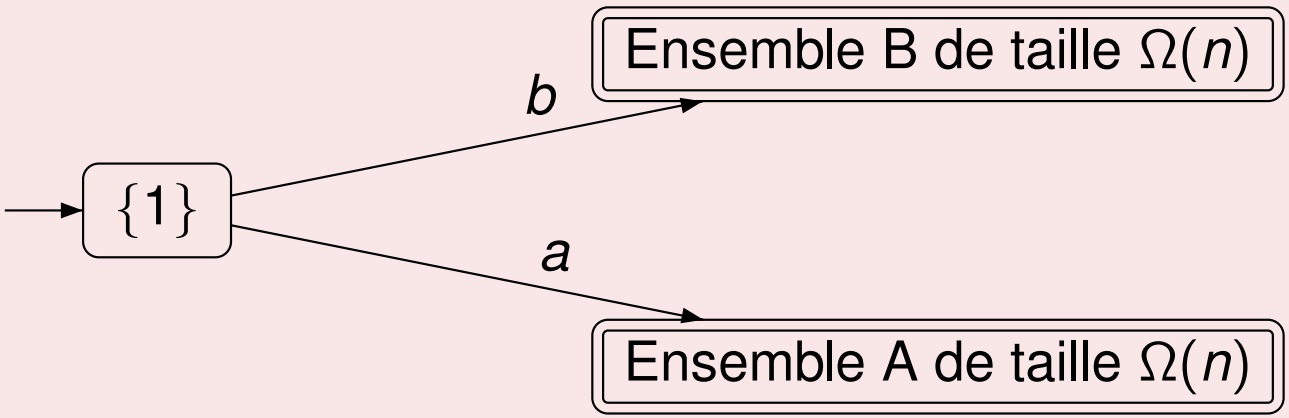
Un résultat négatif : l'idée

- la distribution uniforme sur les automates à n états sur un alphabet à k lettres signifie que chaque automate est tiré avec probabilité $\frac{1}{2^{kn^2}2^n}$
- cela revient à tirer chaque transition/triplet $(p, a, q) \in T$ avec probabilité $\frac{1}{2}$
- et de décider si un état est final avec probabilité $\frac{1}{2}$.
- le nombre de transition sortant d'un état p étiquetté par une lettre a est presque surement supérieure à $\frac{n}{2} - \sqrt{n}$.

Autrement dit, on tire $n \times k \times n$ variables aléatoires i.i.d selon une loi de Bernouilli de paramètre $\frac{1}{2}$ et n variables i.i.d selon une loi de Bernouilli de paramètre $\frac{1}{2}$

Génération aléatoire d'automates non-déterministes : la problématique

Un résutat négatif : l'idée

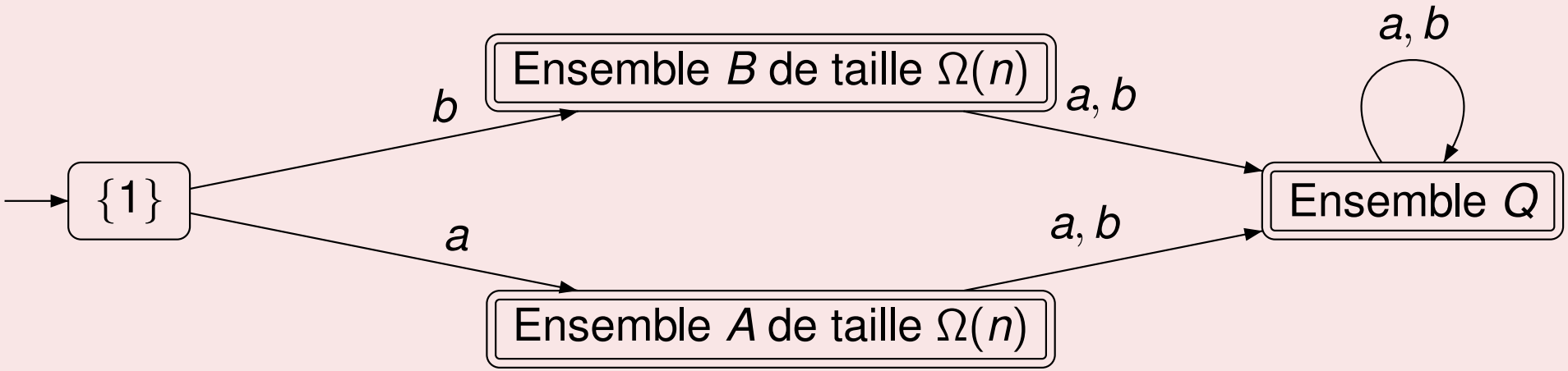


Presque sûrement, on a :

- les états A et B sont finals.
- $|A \setminus B| = \Omega(n)$ et $|B \setminus A| = \Omega(n)$

Génération aléatoire d'automates non-déterministes : la problématique

Un résutat négatif : l'idée



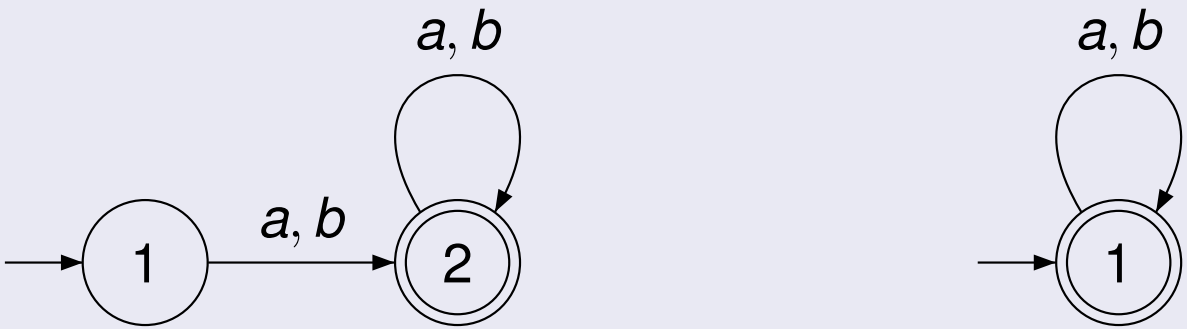
Presque sûrement, on a :

- les transitions sortantes des états A et B arrivent dans Q

Génération aléatoire d'automates non-déterministes : la problématique

Autrement dit

La distribution uniforme sur les automates de taille n produit presque sûrement soit automate reconnaissant le langage Σ^+, soit celui reconnaissant Σ^*, représentables par :



Génération aléatoire d'automates non-déterministes : la problématique

Étude expérimentale

Vardi, Tabakov (2005) : les auteurs utilisent le modèle suivant.

- un unique état initial,
- chaque transition (p, a, q) est ajoutée dans l'automate avec probabilité $\frac{\rho}{n}$
- chaque état est final avec probabilité $\frac{1}{2}$ **OU** on tire un unique état aléatoirement.
- on reste uniforme parmi les automates à n états et m transitions.

Résultat

La taille moyenne de l'automate déterminisé est supérieure à n lorsque

$$\rho \in [1, \ldots, 2]$$

Génération aléatoire d'automates non-déterministes : la problématique

Étude expérimentale

Vardi, Tabakov (2005) : les auteurs utilisent le modèle suivant.

- un unique état initial,
- chaque transition (p, a, q) est ajoutée dans l'automate avec probabilité $\frac{\rho}{n}$
- chaque état est final avec probabilité $\frac{1}{2}$ **OU** on tire un unique état aléatoirement.
- on reste uniforme parmi les automates à n états et m transitions.

Résultat

La taille moyenne de l'automate déterminisé est supérieure à n lorsque

$$\rho \in [1, \ldots, 2]$$

Génération aléatoire d'automates non-déterministes : la problématique

Un premier générateur

Héam, Joly (2016) : les auteurs définissent une chaîne de Markov sur l'ensemble des automates non-déterministes, non-isomorphes à n états et m transitions.

Problèmes

- malgré un temps de mélange rapide, le coût d'une marche aléatoire est très élévé à cause du calcul d'un étiquetage canonique.
- les auteurs ne dépassent pas les automates de taille 10 expérimentalement.

Génération aléatoire d'automates non-déterministes : la problématique

- Partir du modèle des kn^2 variables i.i.d de paramètre $\rho \in [1..2]$ semble plus efficace d'un point de vue algorithmique.
- Question : garantir que l'on tire des automates non-isomorphes est il intéressant du point de vue de la déterminisation ?
- Cela revient à engendrer des automates canoniques.

Génération aléatoire d'automates non-déterministes : la problématique

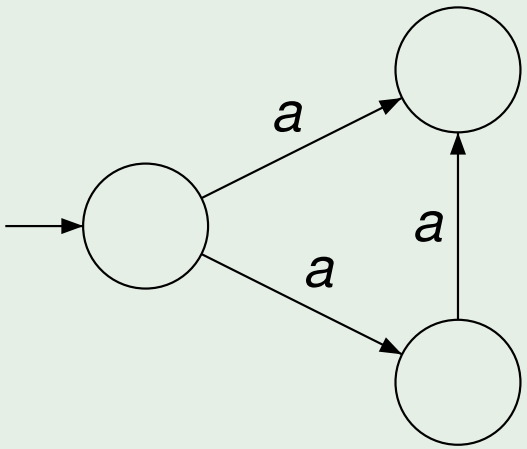
Génération uniforme à isomorphisme près : une bonne idée ?

Considérons deux matrices binaires encodant les transitions sortantes étiquettées par a.

0	1	1
0	0	0
0	0	1

0	1	1
0	0	1
0	0	0

Ces deux matrices encodent le même automate canonique (sans automorphisme)



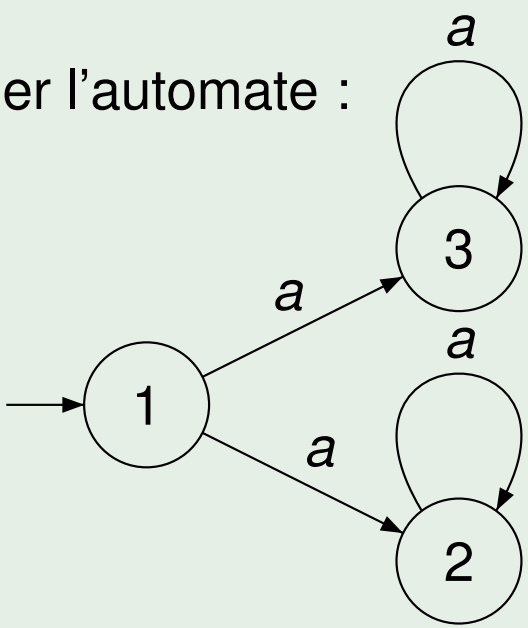
Génération aléatoire d'automates non-déterministes : la problématique

Génération uniforme à isomorphisme près : une bonne idée ?

A l'inverse, la matrice

0	1	1
0	1	0
0	0	1

est la seule à encoder l'automate :



pour lequel on a l'automorphisme suivant : $(1, 3, 2)$

Génération aléatoire d'automates non-déterministes : la problématique

Génération à isomorphisme près

Engendrer des automates à isomorphisme près signifie donner plus de poids aux automates possédant des automorphismes.

Définition

Deux états p et q d'un automate sont dits permutables s'il existe un automorphisme σ tel que $\sigma(p) = q$ et $\sigma(q) = p$.

Plus précisément, soit $\mathcal{P}$ la partition d'ensemble induite par la relation "être permutable". Le nombre de matrices associées à un automate canonique est :

$$\frac{(n-1)!}{\prod_{P \in \mathcal{P}} |P|!}$$

Génération aléatoire d'automates non-déterministes : la problématique

Définition

Deux états p et q d'un automate sont dits permutables s'il existe un automorphisme σ tel que $\sigma(p) = q$ et $\sigma(q) = p$.

Lemma

Soit $N =< Q, \Sigma, I, F, T >$ un automate non-déterministe.
Pour tout $p, q \in Q$ tel que p et q sont permutables, pour tout état P du déterminisé $det(\mathcal{N})$, on a

$$p \in P \Longleftrightarrow q \in P$$

Génération aléatoire d'automates non-déterministes : des modèles intéressants

Un modèle intéressant pour la génération aléatoire doit

- permettre d'obtenir des automates drastiquement plus petits que leur déterminisé.

et garantit que :

- les automates sont accessibles,
- le nombre d'états permutables est borné.

Premier modèle

- Partons de la distribution utilisée par Vardi pour $1 \leq \rho \leq 2$.
- On souhaite estimer la distribution de la taille de la partie accesssible $Acc_{n,k,\rho}$.
  - On montre que celle ci est inférieure à $\frac{n}{2}$ avec une probabilité constante
  - On montre que si $|Acc_{n,k,\rho}| > \frac{n}{2}$, on a
    $\mathbb{P}\left(|Acc_{n,k,\rho}| - \beta_{\rho,k} n > k\sqrt{n}\log n\right) = \Theta\left(\frac{1}{n^k}\right)$
    $(\beta_k = 1 + \frac{1}{k\rho} W_0(-k\rho e^{-k\rho}))$

$\mathbb{P}\left(|Acc_{n,k,\rho}| < \frac{n}{2}\right)$

On majore la probabilité que la partie accessible soit de taille $x+1$ comme suit :

$$\binom{n-1}{x}\left(1-\left(1-\frac{\rho}{n}\right)^{kx}\right)^{x}\left(1-\frac{\rho}{n}\right)^{k(x+1)(n-x-1)}$$

Quelques calculs plus tard...

Pour $k=2$ et $\rho=1$, on a :

$$\sum_{x=0}^{\frac{n}{2}} \mathbb{P}\left(|Acc_{n,k,\rho}| = x\right) < 0.5$$

La somme décroit lorsque ρ ou k augmentent.

On majore la probabilité que la partie accessible soit de taille $x+1$ comme suit :

$$\binom{n-1}{x}\left(1-\left(1-\frac{\rho}{n}\right)^{kx}\right)^{x}\left(1-\frac{\rho}{n}\right)^{k(x+1)(n-x-1)}$$

Quelques calculs plus tard...

Pour $k=2$ et $\rho=1$, on a :

$$\sum_{x=0}^{\frac{n}{2}}\mathbb{P}\left(|Acc_{n,k,\rho}|=x\right) < 0.5$$

La somme décroit lorsque ρ ou k augmentent.

Cas où $|Acc_{n,k,\rho}| \geq \frac{n}{2}$

Berend et Kontorovitch utilisent un processus aléatoire. On les imites.
On effectue un parcours en largeur des états p de la partie accessible et leur transition sortante dans l'ordre alphabétique. Soit ν_t le nombre d'états accessibles après avoir observé tous les t premiers couples (p, a). Dans notre cas :

$$\nu_t = \nu_{t-1} + x, \text{ avec probabilité } \binom{n - \nu_{t-1}}{x} \frac{\rho}{n}^x \left(1 - \frac{\rho}{n}\right)^{n-\nu_{t-1}-x}$$

où k est la taille de l'alphabet.
Si la taille de la partie accessible est x, alors $\nu_{kx+1} = x$
On cherche à estimer la plus petite valeur de t pour laquelle $\nu_t \leq \frac{t-1}{k}$.

Cas où $|Acc_{n,k,\rho}| \geq \frac{n}{2}$

- On a $\mathbb{E}\nu_t = n(1 - (1 - \frac{\rho}{n})^t)$
- on pose $F(t) = \mathbb{E}\nu_t - \frac{t-1}{k}$
- $\mathbb{P}(|Acc_{n,k,\rho}| \in [x, y]) \leq \sum_{t=x}^{y} \mathbb{P}(\nu_t - \mathbb{E}\nu_t \leq -F(t))$
- On utilise l'inégalité de Chernoff :

$$\mathbb{P}(\nu_t - \mathbb{E}\nu_t \leq -F(t)) \leq e^{-2\frac{F(t)^2}{t}}$$

- pour $k = 2$ et $\rho = 1$, on montre que $\frac{F(t)^2}{t} \geq \log^2 n$.
- on obtient que $\mathbb{P}(|Acc_{n,k,\rho}| \in [\frac{n/2}{,}\beta_{\rho,k}n - k\sqrt{n}\log n]) \leq \Theta\left(\frac{1}{n^2}\right)$.

Algorithm 1: Random accessible NFA (smart) algorithm

Input: Nombre d'états n, alphabet Σ à k lettres, une probabilité $\frac{\rho}{n}$, $1 \leq \rho \leq 2$
Output: Un automate accesssible $\mathcal{N} =< Q, I, F, T, \Sigma >$ à n états

$cn \leftarrow \frac{n}{\beta_{\rho,k}}$;
répéter
  $\mathcal{N} \leftarrow$ créer un automate à cn états;
  Choisir les états terminaux selon le modèle aléatoire;
  pour $i \in \{1, \ldots, n\}$ **faire**
    pour $a \in \Sigma$ **faire**
      $outdegree \leftarrow Poisson(\rho)$;
      pour $j \in \{1, \ldots, outdegre\}$ **faire**
        répéter
          $arrival \leftarrow$ nombre aléatoire entre 1 et n;
        jusqu'à $(i, a, arrival) \in T$;
        Ajouter la transition $(i, a, arrival)$ dans T;
jusqu'à *Acc ne contient pas n states* ;
$Acc \leftarrow$ partie accessible de $\mathcal{N}$;
return Acc;

Généralisation

Généralisation

Il est possible d'adapter la méthode de preuve précédente en utilisant d'autres loi de probabilité à la place de la loi de Poisson. Il suffit que cette loi vérifie que :

- le nombre moyen de transitions sortantes d'un état est égal à $1 + \varepsilon$.
- $\mathbb{E}\nu_t$ est telle que l'on puisse montrer facilement que $\frac{F(t)^2}{t} \geq \log^2 n$.

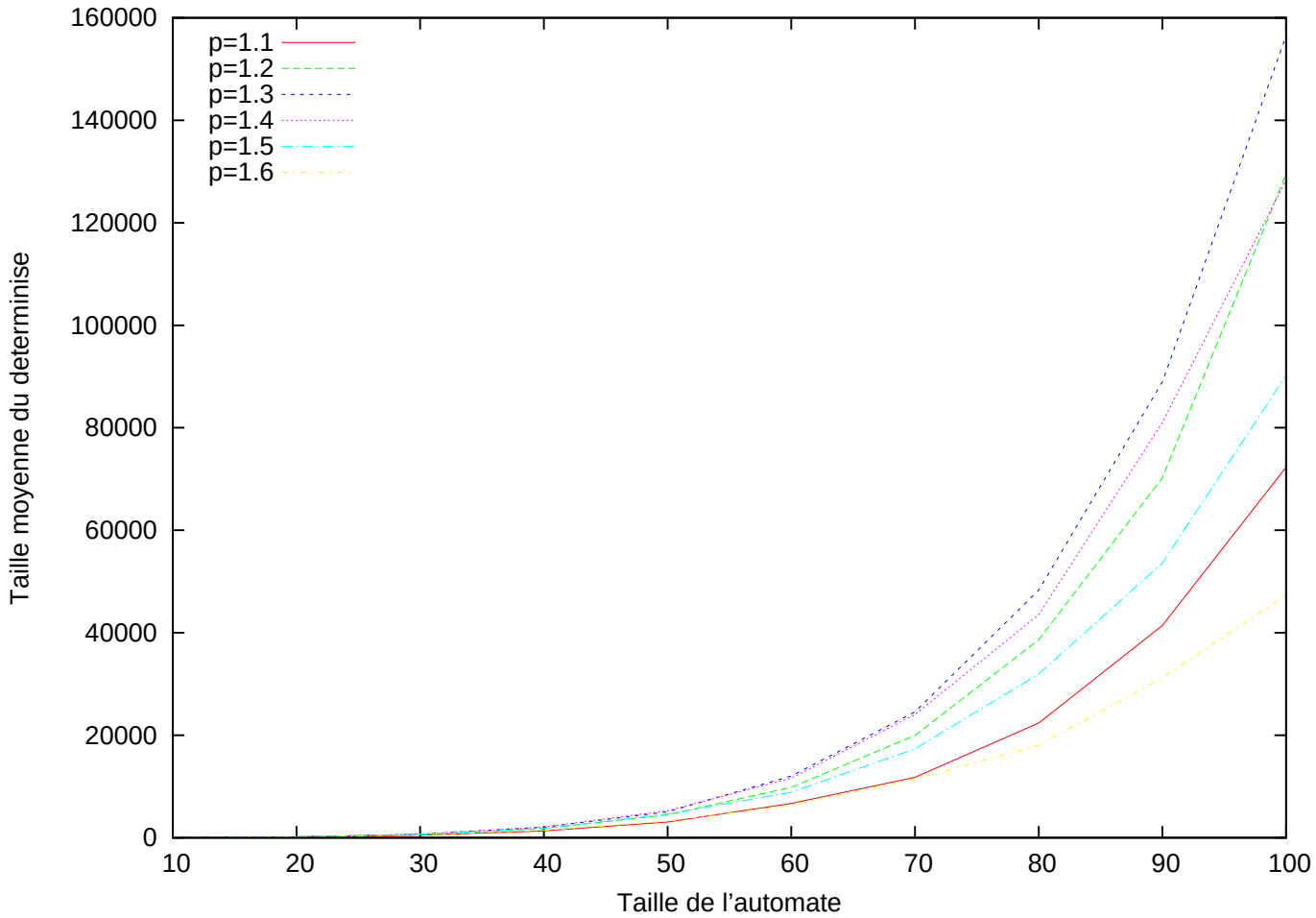
Arnaque !

Et le nombre d'états permutables ? ? ?

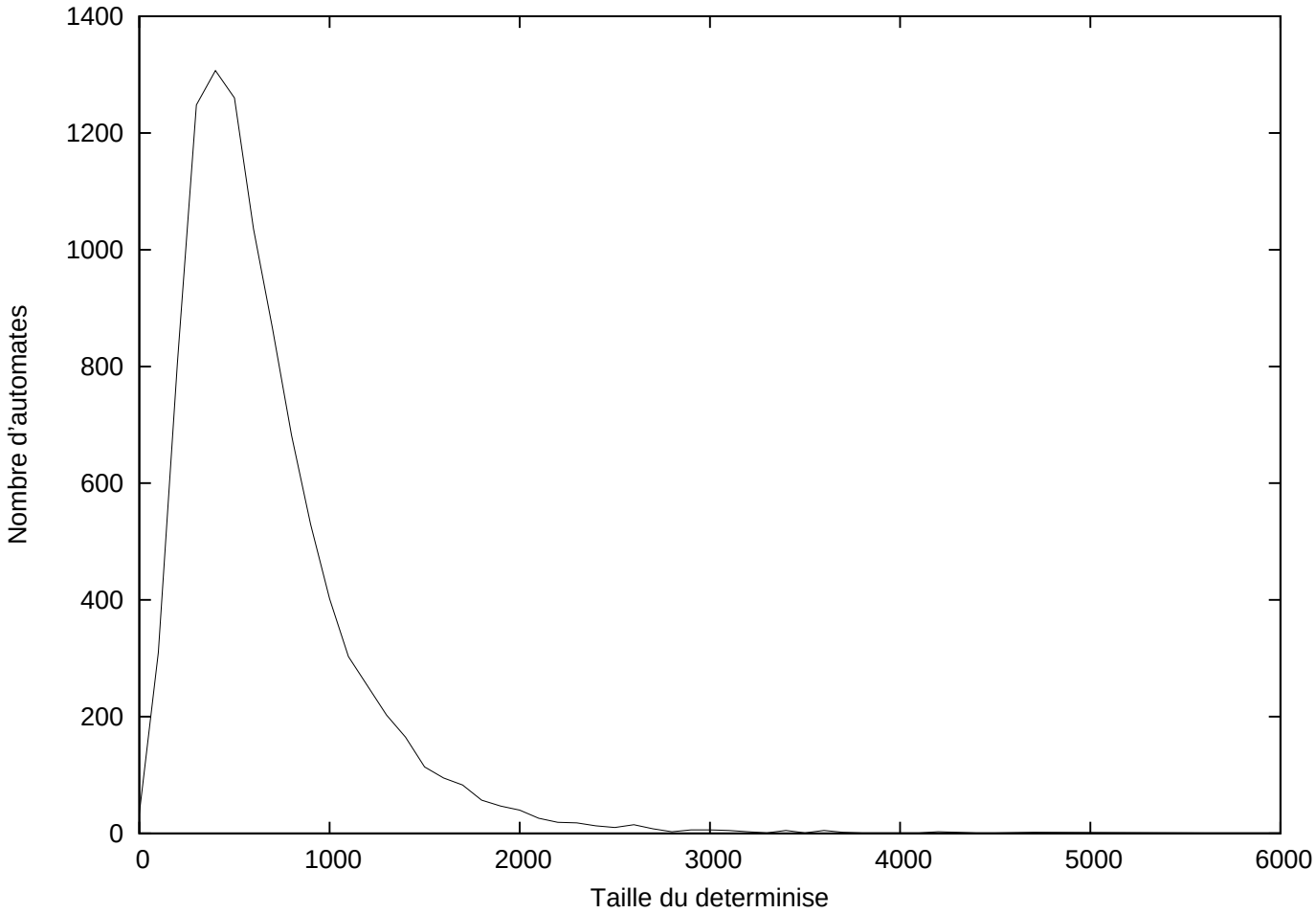
- On peut montrer qu'il existe une proportion constante d'états permutables
- la constante est petite donc l'influence n'apparaît pas sur les simulations, où un automate de taille 1000 contient expérimentalement moins de 2 états permutables.

Résultats expérimentaux : Loi de Poisson

Évolution de la taille moyenne du déterminisé pour $k = 2$.



Résultats expérimentaux : Loi de Poisson



Déterminisation de 10.000 automates de taille 30.

Résultats expérimentaux : Loi géométrique

Évolution de la taille moyenne du déterminisé pour $k = 2$.