Analyse d'algorithmes, langages et automates

Frédérique Bassino

LIPN, Université Paris 13 et CNRS

Journées ALEA 2011

Objectifs

- Etudier de manière quantitative les propriétés des langages réguliers

Objectifs

- Etudier de manière quantitative les propriétés des langages réguliers
- Utiliser des structures combinatoires pour les représenter
  - automates minimaux,
  - automates de Glushkov dérivés des expressions rationnelles

Objectifs

- Etudier de manière quantitative les propriétés des langages réguliers
- Utiliser des structures combinatoires pour les représenter
  - automates minimaux,
  - automates de Glushkov dérivés des expressions rationnelles
- Enumérer, simuler ces structures
- Analyser en moyenne des algorithmes manipulant ces structures (Algorithme de minimisation de Moore)

Objectifs

- Etudier de manière quantitative les propriétés des langages réguliers
- Utiliser des structures combinatoires pour les représenter
  - automates minimaux,
  - automates de Glushkov dérivés des expressions rationnelles
- Enumérer, simuler ces structures
- Analyser en moyenne des algorithmes manipulant ces structures (Algorithme de minimisation de Moore)
- Pas d'étude des propriétés d'un langage aléatoire dans les différents modèles considérés.

Langages réguliers

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

Langages réguliers

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

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

Expressions rationnelles

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

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

Expressions rationnelles : $a^*b + a(c + d^*)$

- Cette representation des langages n'a pas de forme canonique
- Tester si une expression représente A^* est PSPACE-complet.

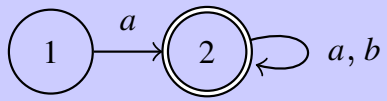
Automates finis

Un automate fini $\mathcal{A}$ est

- un graphe orienté
- dont les transitions sont étiquetées sur un alphabet fini
- avec un ensemble I d'états initiaux
- et un ensemble F d'états terminaux.

- Le langage reconnu par un automate fini est l'ensemble des étiquette des chemins d'un état initial à un état final.

Automate fini :



D'après le théorème de Kleene, les langages reconnus par un automate sont les langages réguliers.

Automates minimaux

A chaque langage régulier, on peut associer de manière unique son automate minimal.

Un automate est déterministe et complet

- s'il n'a qu'un seul état initial
- et que pour tout état q et pour toute lettre ℓ, il existe exactement une transition d'étiquette ℓ partant de q.

Un automate déterministe est accessible si tout état de l'automate peut être atteint par un chemin partant de l'état initial.

Automates minimaux

A chaque langage régulier, on peut associer de manière unique son automate minimal.

Un automate est déterministe et complet

- s'il n'a qu'un seul état initial
- et que pour tout état q et pour toute lettre ℓ, il existe exactement une transition d'étiquette ℓ partant de q.

Un automate déterministe est accessible si tout état de l'automate peut être atteint par un chemin partant de l'état initial.

L'automate minimal d'un langage régulier est l'automate accessible déterministe et complet ayant le plus petit nombre d'états qui reconnaît ce langage.

Génération aléatoire d'automates

Grandes étapes

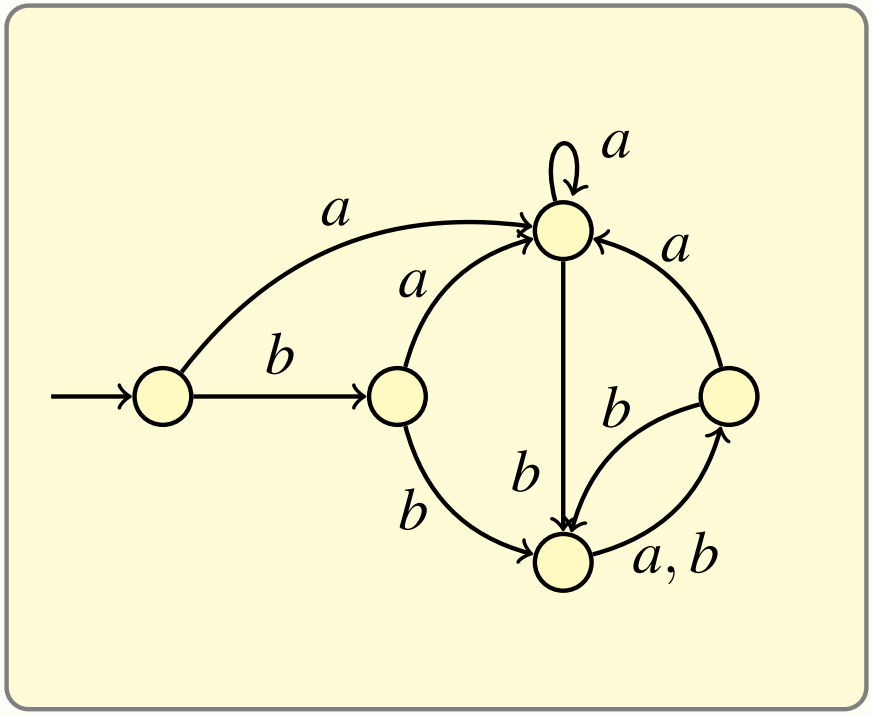
Objectif

Engendrer aléatoire et uniformément les langages réguliers selon le nombre d'états de leur automate minimal.

- Engendrer aléatoirement et uniformément des automates accessibles déterministes et complets à n états sur un alphabet à k lettres.
- Transformer ces automates en partitions d'ensembles particulières.
- Engendrer ces partitions d'ensembles grâce à un générateur de Boltzmann.

d'un automate à une partition d'ensemble

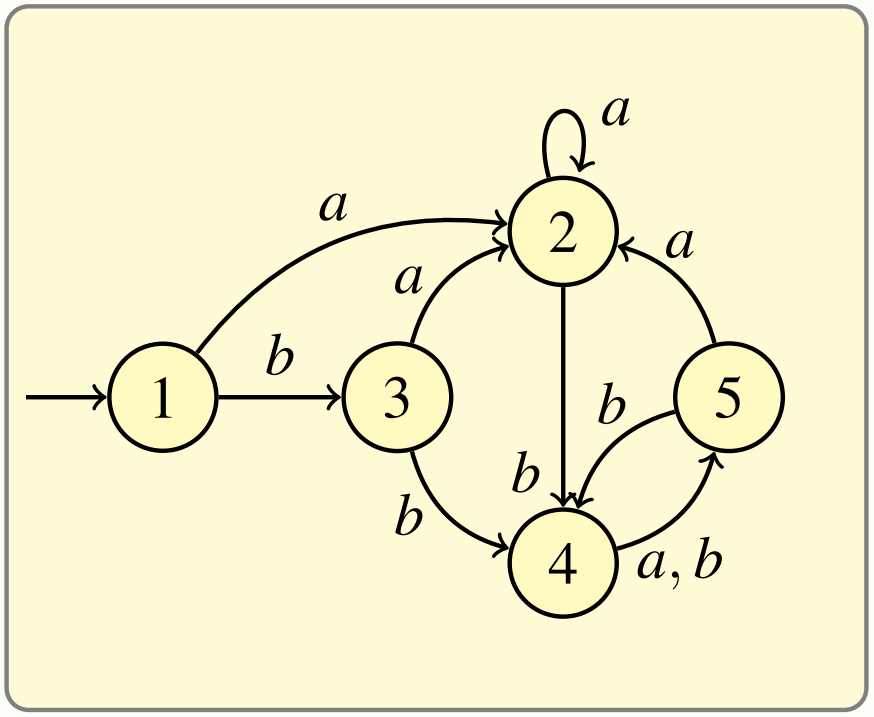
Automate déterministe accessible et complet.



- Les états teminaux sont omis. Lors de la génération aléatoire, ils seront tirés indépendemment.
- Pour éviter les structures isomorphes, on étiquette les états de manière canonique suivant un parcours en largeur.

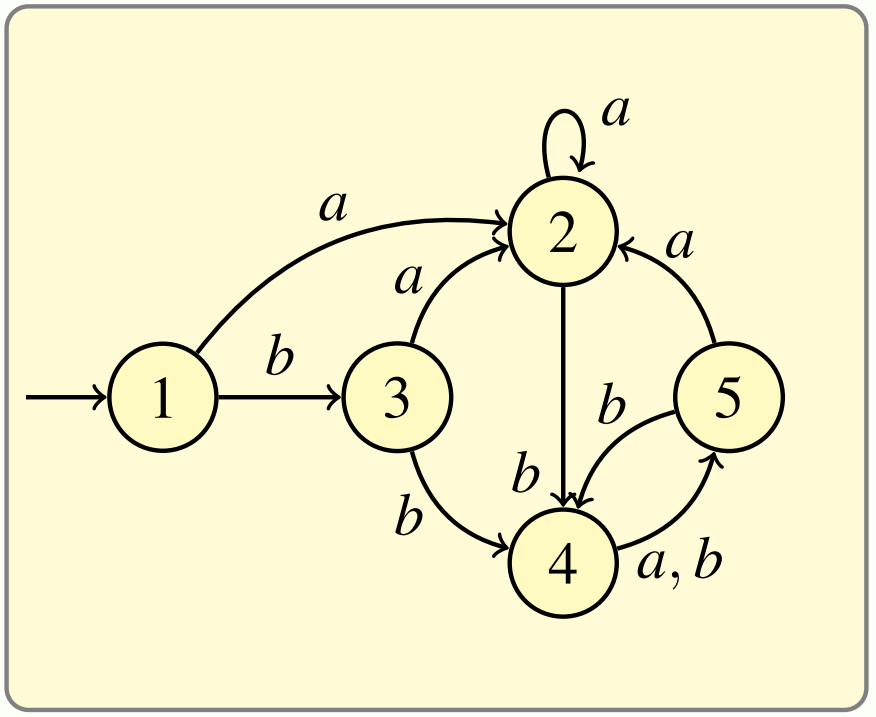
d'un automate à une partition d'ensemble

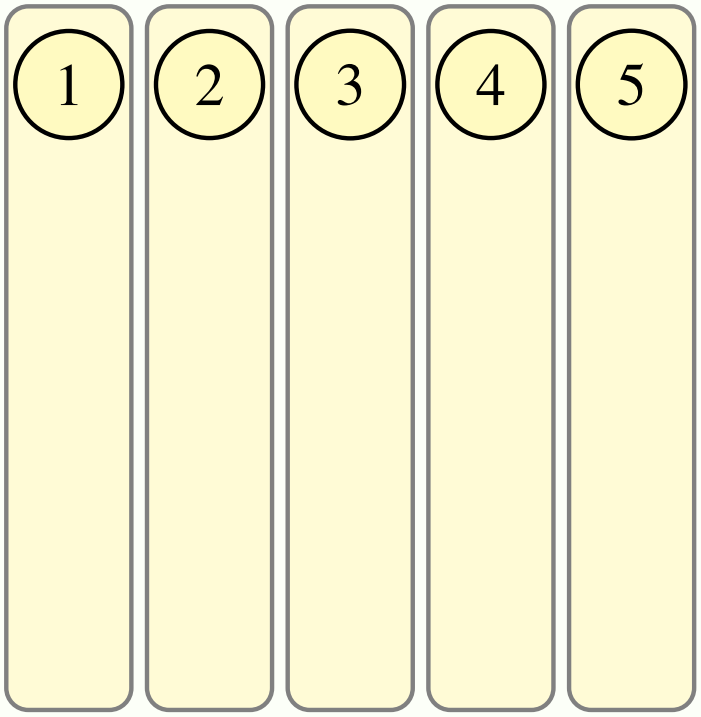
Automate déterministe accessible et complet.



d'un automate à une partition d'ensemble

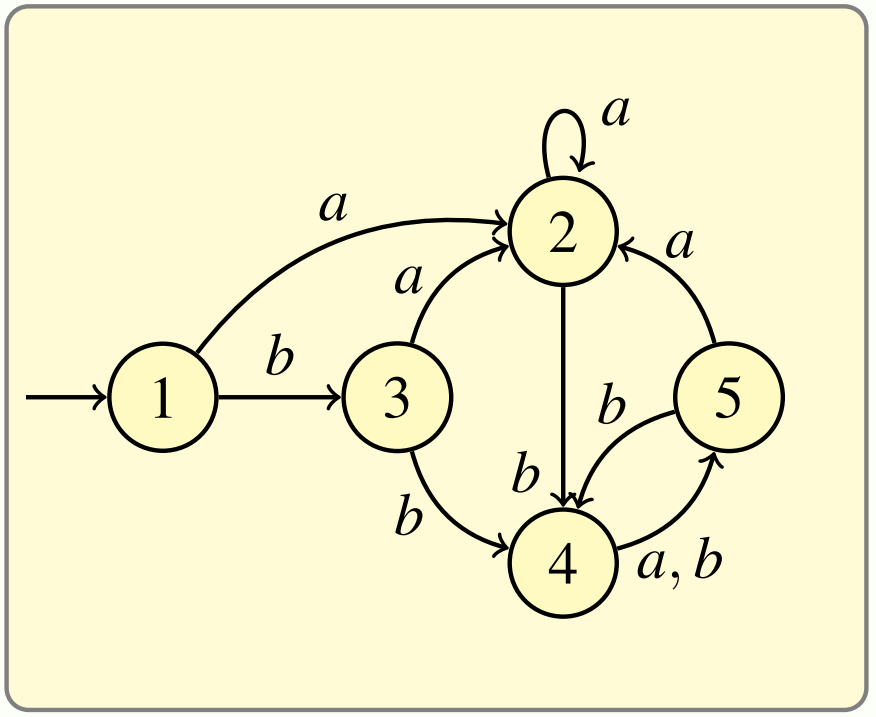
Automate déterministe accessible et complet.

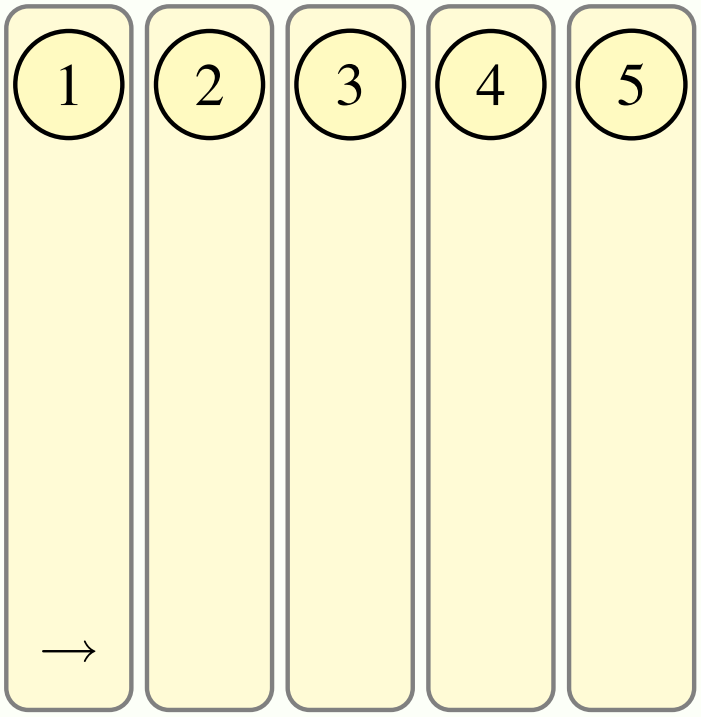




d'un automate à une partition d'ensemble

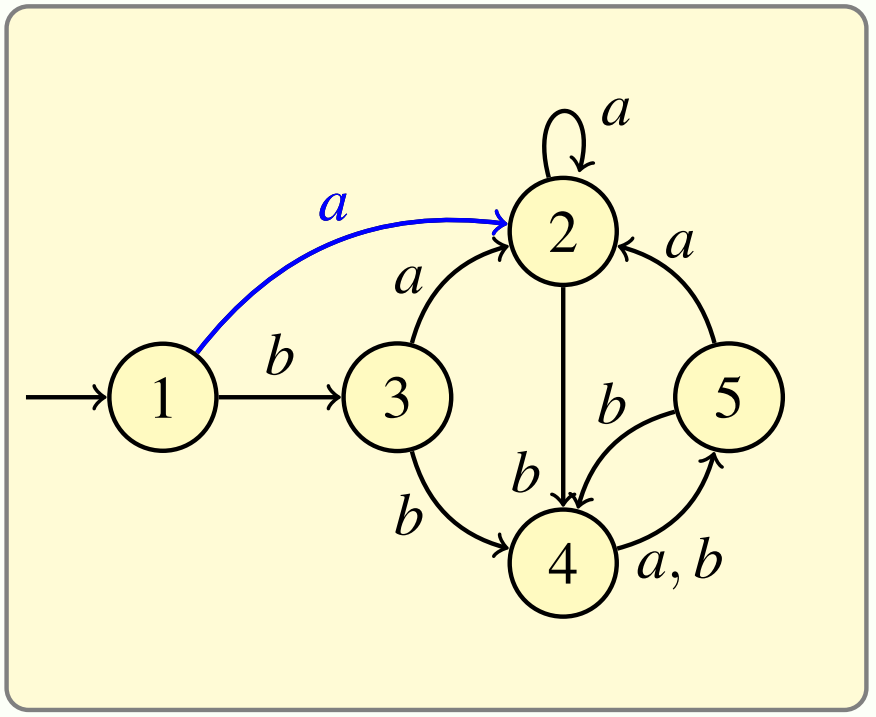
Automate déterministe accessible et complet.

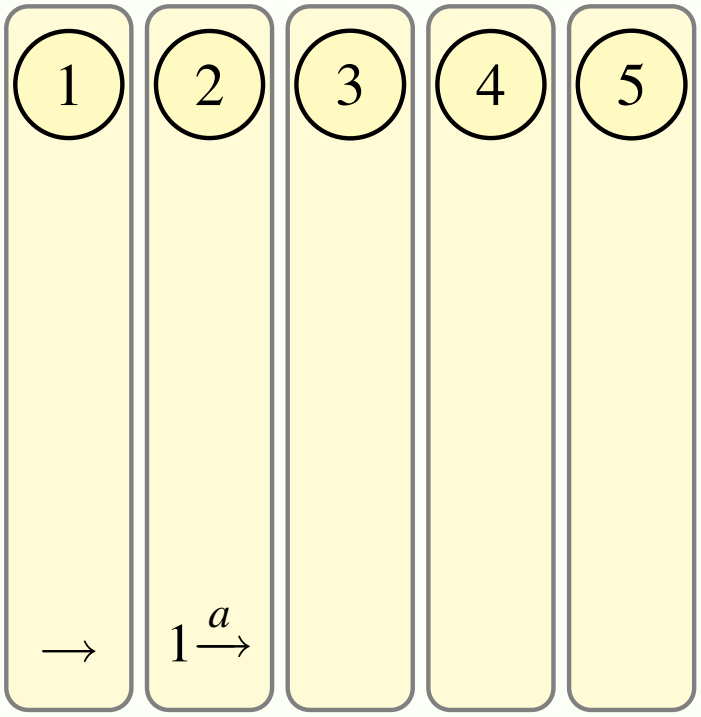




d'un automate à une partition d'ensemble

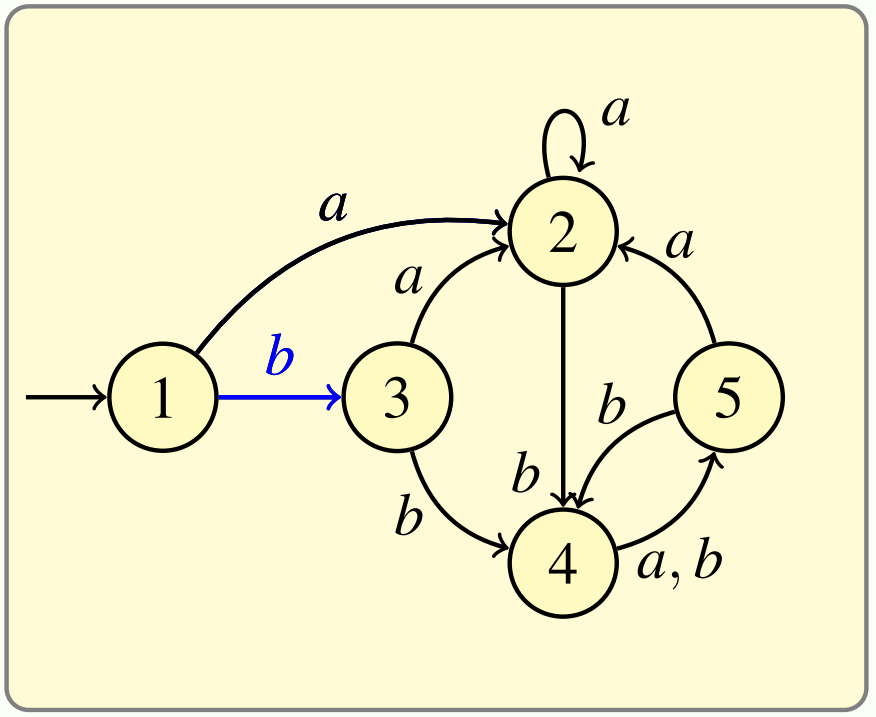
Automate déterministe accessible et complet.

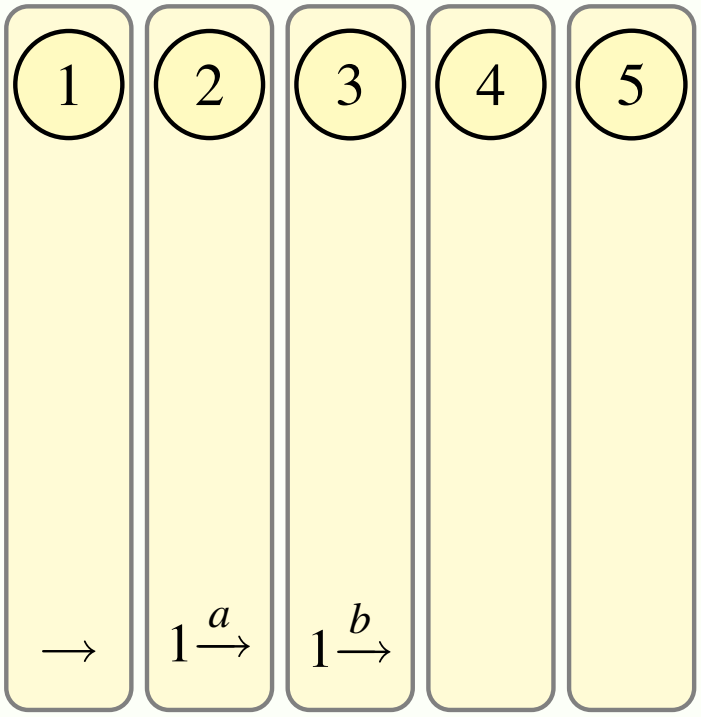




d'un automate à une partition d'ensemble

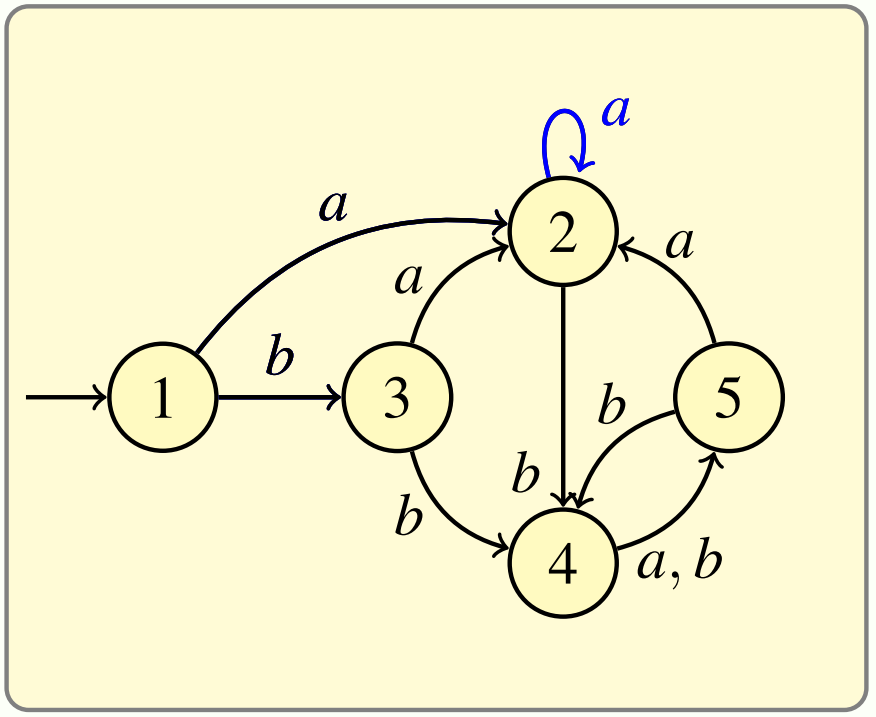
Automate déterministe accessible et complet.

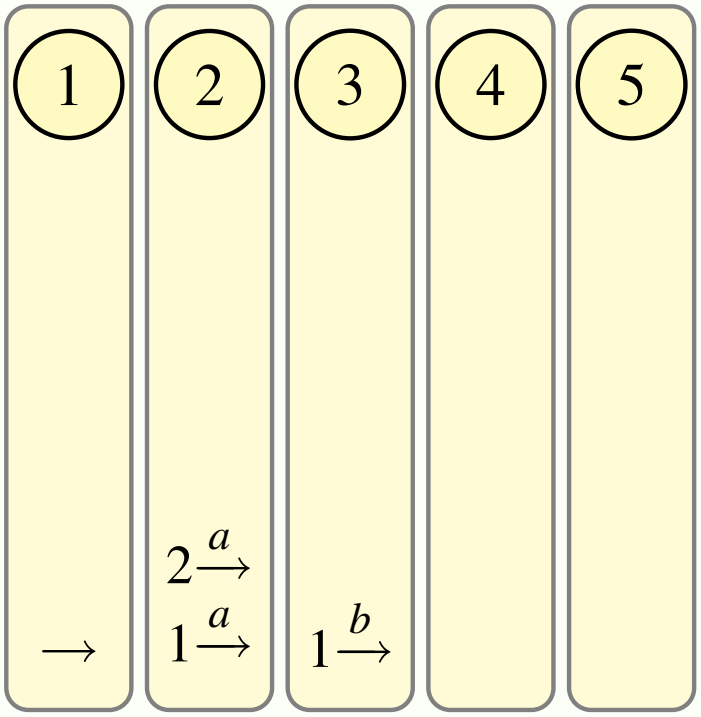




d'un automate à une partition d'ensemble

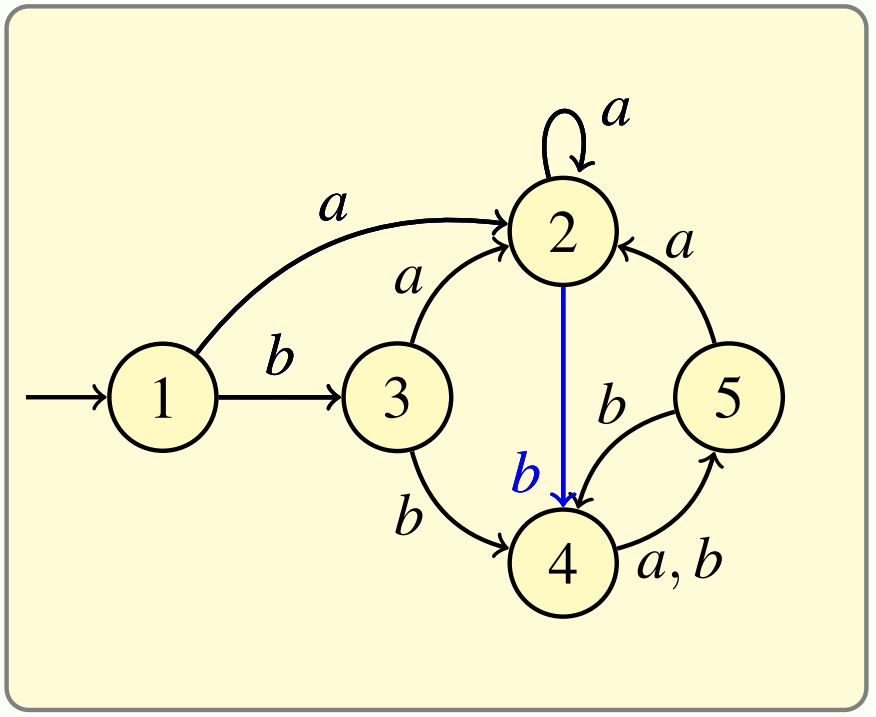
Automate déterministe accessible et complet.

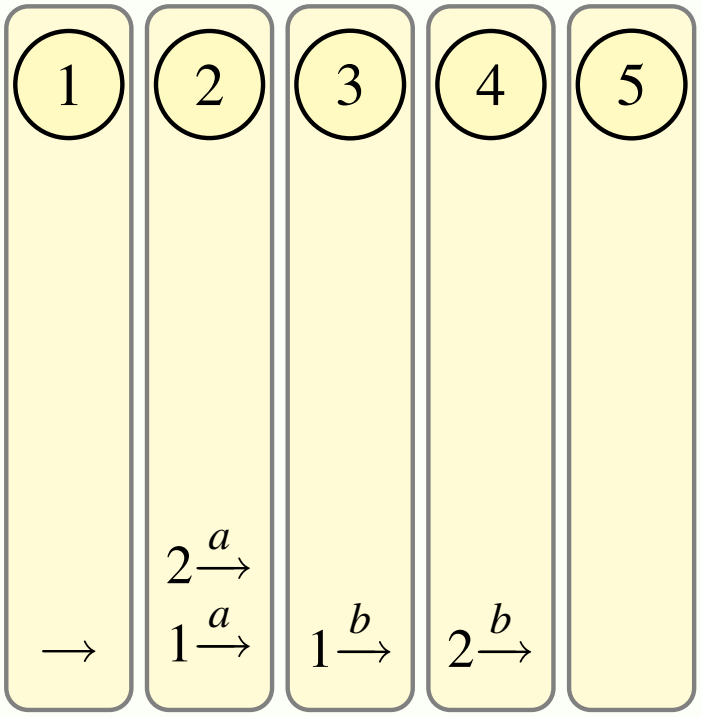




d'un automate à une partition d'ensemble

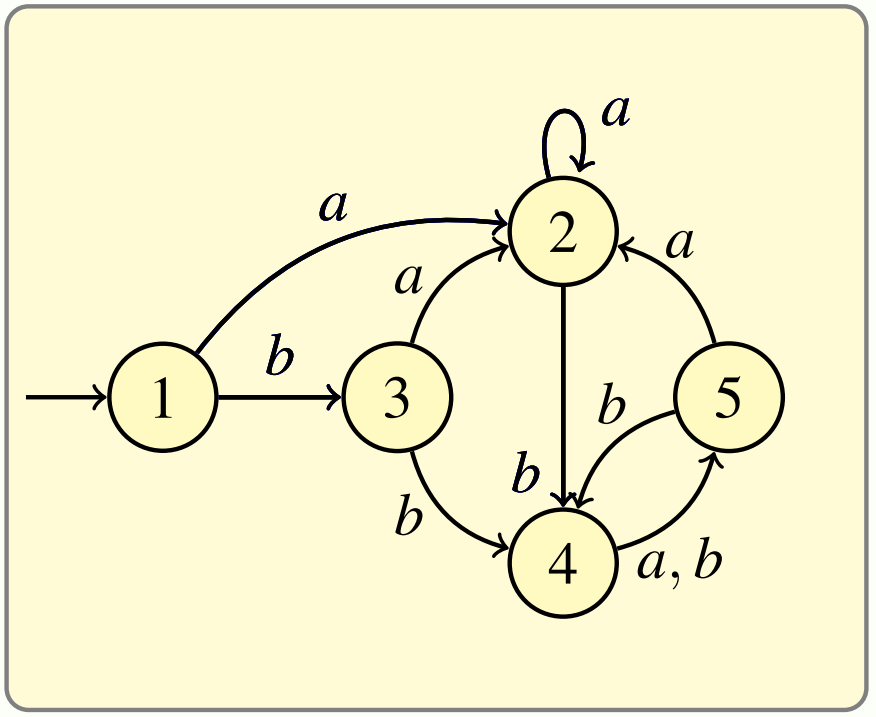
Automate déterministe accessible et complet.





d'un automate à une partition d'ensemble

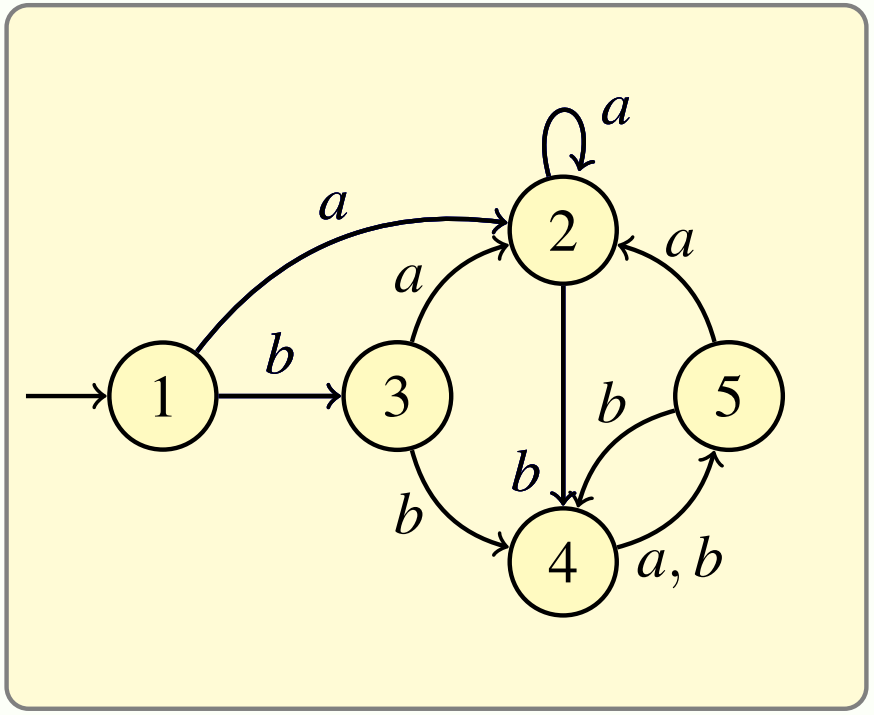
Automate déterministe accessible et complet.



1	2	3	4	5
	$5 \xrightarrow{a}$		$5 \xrightarrow{b}$	
	$3 \xrightarrow{a}$		$3 \xrightarrow{b}$	$4 \xrightarrow{b}$
	$2 \xrightarrow{a}$		$2 \xrightarrow{b}$	$4 \xrightarrow{a}$
$\rightarrow$	$1 \xrightarrow{a}$	$1 \xrightarrow{b}$		

d'un automate à une partition d'ensemble

Automate déterministe accessible et complet.



1	2	3	4	5
	$5 \xrightarrow{a}$			
	$3 \xrightarrow{a}$		$5 \xrightarrow{b}$	
	$2 \xrightarrow{a}$		$3 \xrightarrow{b}$	$4 \xrightarrow{b}$
$\rightarrow$	$1 \xrightarrow{a}$	$1 \xrightarrow{b}$	$2 \xrightarrow{b}$	$4 \xrightarrow{a}$

En numerotant les flêches selon leur ordre d'apparition (parcours en largeur), on obtient la partition des $kn + 1$ flèches en n parts suivante :

$$\{\{1\}, \{2, 4, 6, 10\}, \{3\}, \{5, 7, 11\}, \{8, 9\}\}$$

Définition

Une partition des $kn + 1$ flèches en n part est dite réalisable quand on peut l'obtenir à partir d'un automate accessible.

Définition

Une partition des $kn + 1$ flèches en n part est dite réalisable quand on peut l'obtenir à partir d'un automate accessible.

Théorème [Bassino, Nicaud 07 ; Korshunov 78]

La proportion de partitions réalisables est en $\Theta(1)$.

Définition

Une partition des $kn+1$ flèches en n part est dite réalisable quand on peut l'obtenir à partir d'un automate accessible.

Théorème [Bassino, Nicaud 07 ; Korshunov 78]

La proportion de partitions réalisables est en $\Theta(1)$.

Corollaire

Le nombre d'automates déterministes, accessibles et complets est de l'ordre de $\left\{ {kn+1 \atop n} \right\} 2^n$

Définition

Une partition des $kn+1$ flèches en n part est dite réalisable quand on peut l'obtenir à partir d'un automate accessible.

Théorème [Bassino, Nicaud 07 ; Korshunov 78]

La proportion de partitions réalisables est en $\Theta(1)$.

Corollaire

Le nombre d'automates déterministes, accessibles et complets est de l'ordre de $\left\{ {kn+1 \atop n} \right\} 2^n$ ou encore, de l'ordre de $n \left\{ {kn \atop n} \right\} 2^n$.

Générateur de Boltzmann

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 $N(z) = e^z - 1$.
- 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

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

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

Générateurs de Boltzmann

- 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 $\mathcal{O}(\sqrt{n})$.
- Tirer une partition aléatoire de $\{1, \ldots, kn\}$ pour étiqueter le structure (complexité linéaire)

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

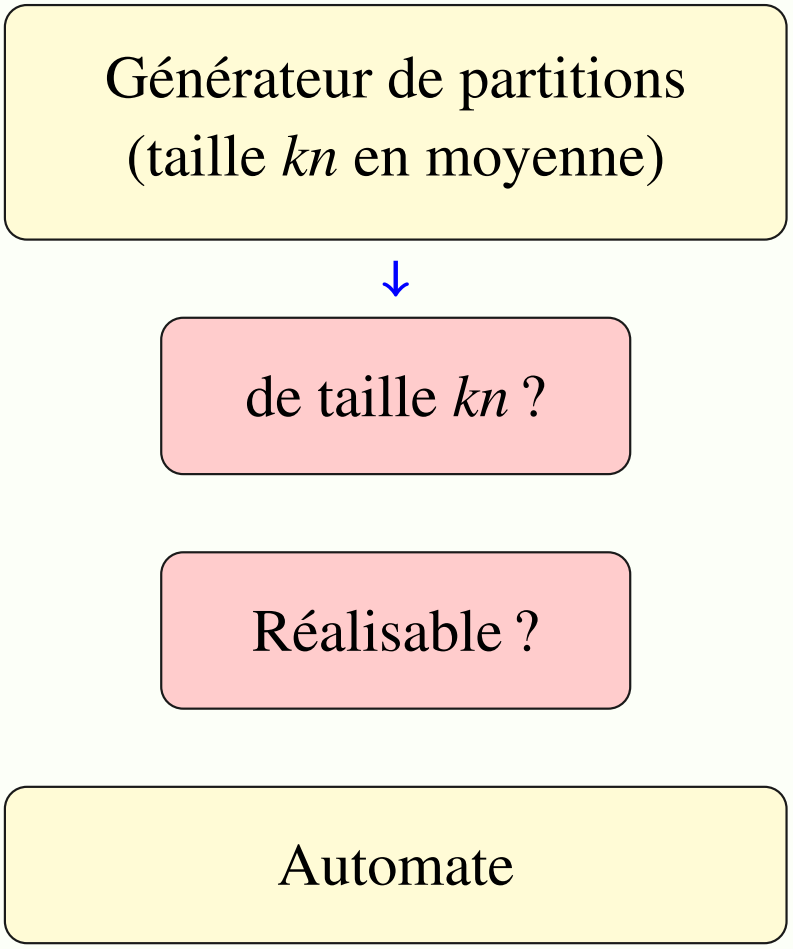
Générateurs de Boltzmann

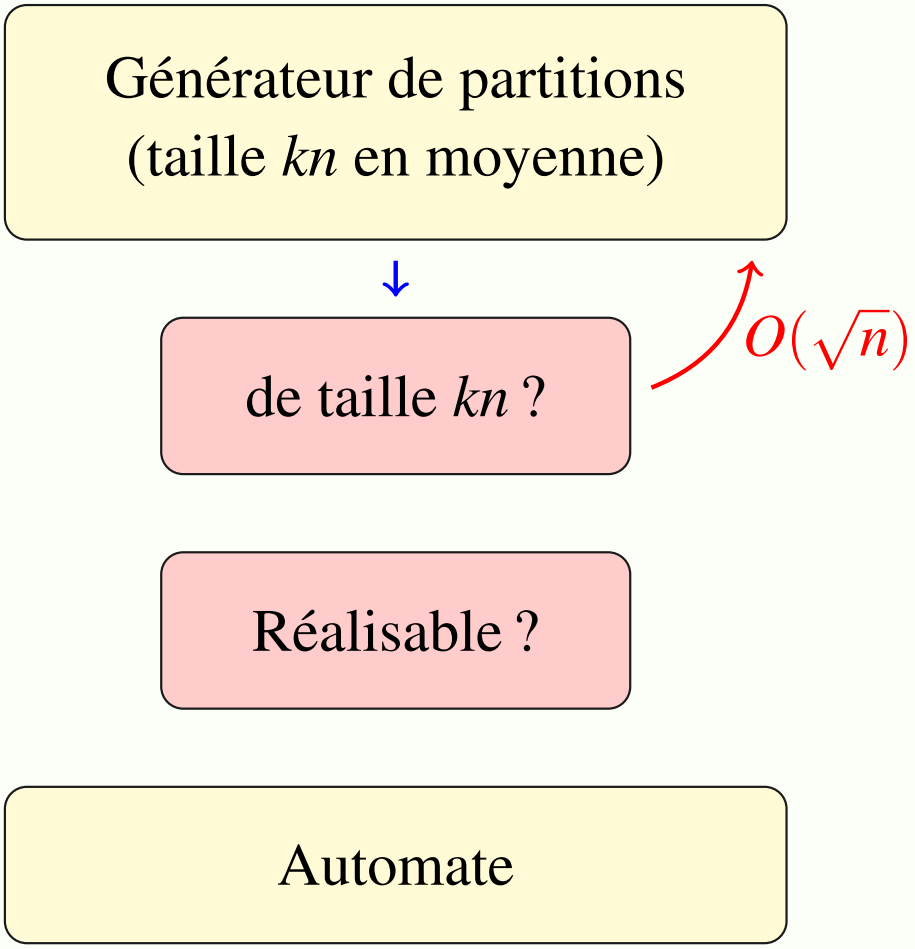
- Engendrer la taille de chacun des n sous-ensembles suivant une loi de Poisson de paramètre $x = \zeta_k$ (complexité linéaire).
- La probabilité que la partition engendrée soit de taille exactement kn est asymptotiquement $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$.
  Le nombre moyen de rejets est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

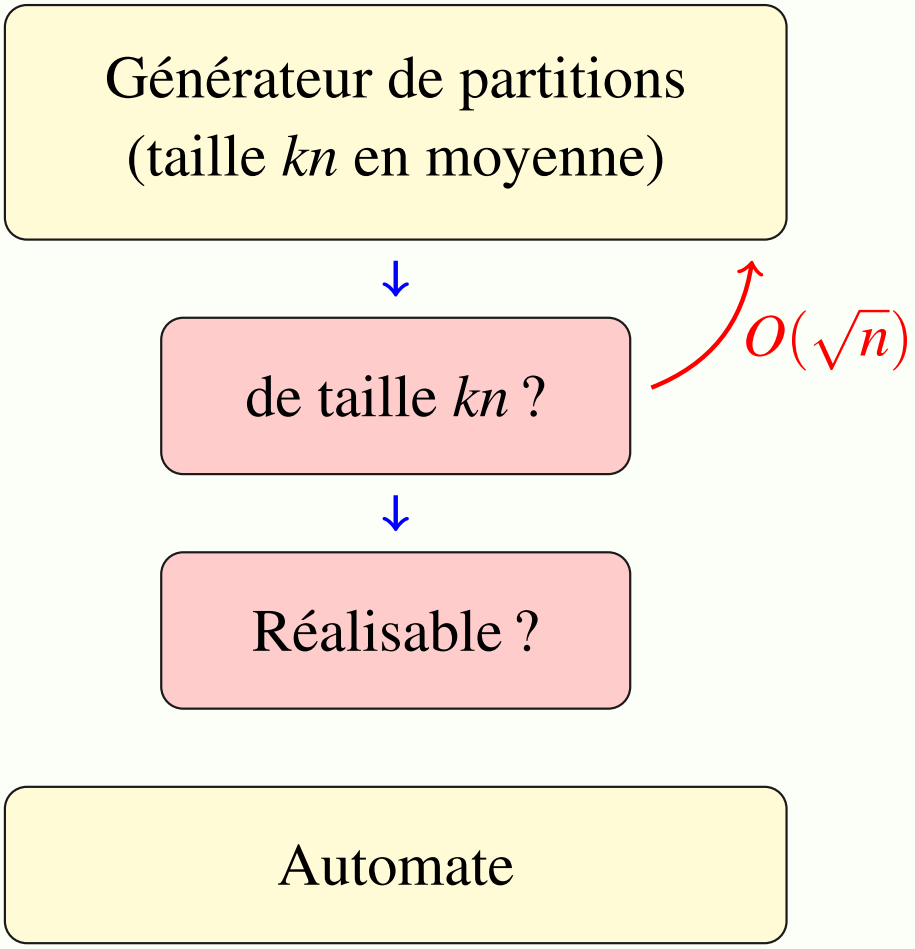
La complexité moyenne en temps de la génération d'une partition d'un ensemble à kn éléments en n sous-ensembles non vies est en $\mathcal{O}(n^{3/2})$

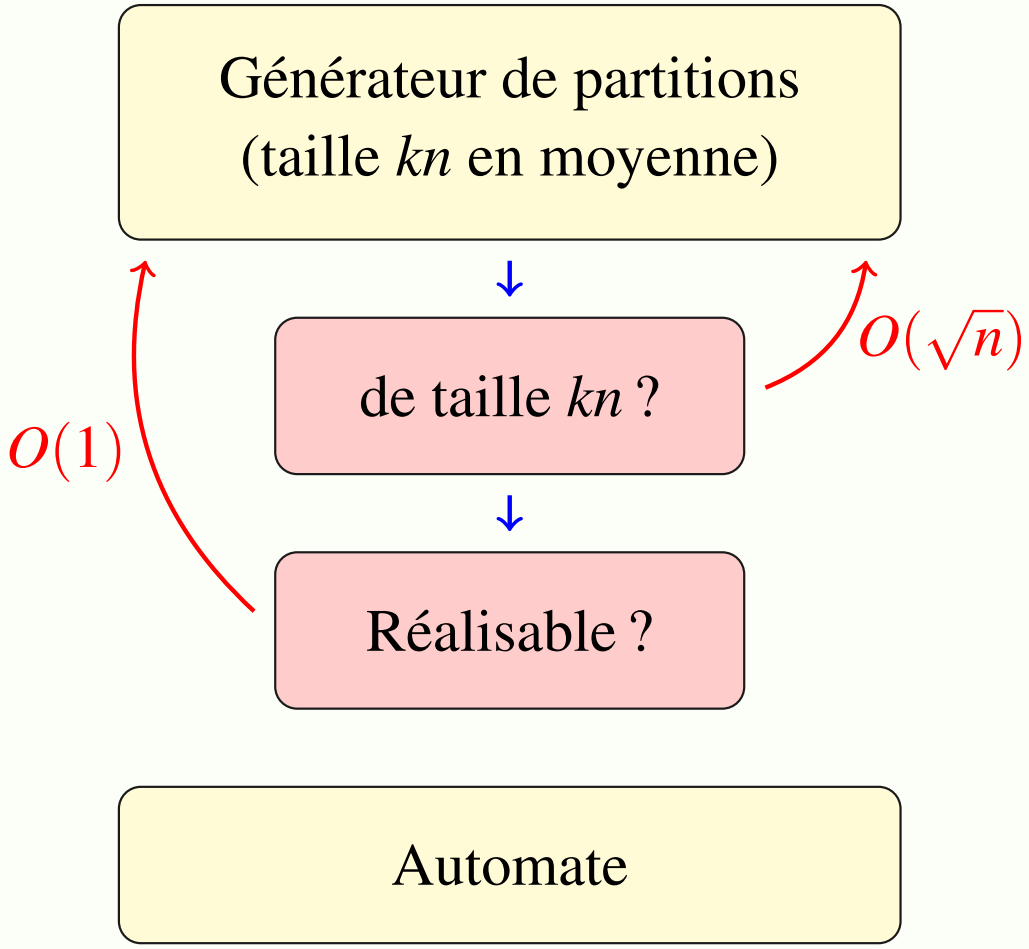
Question

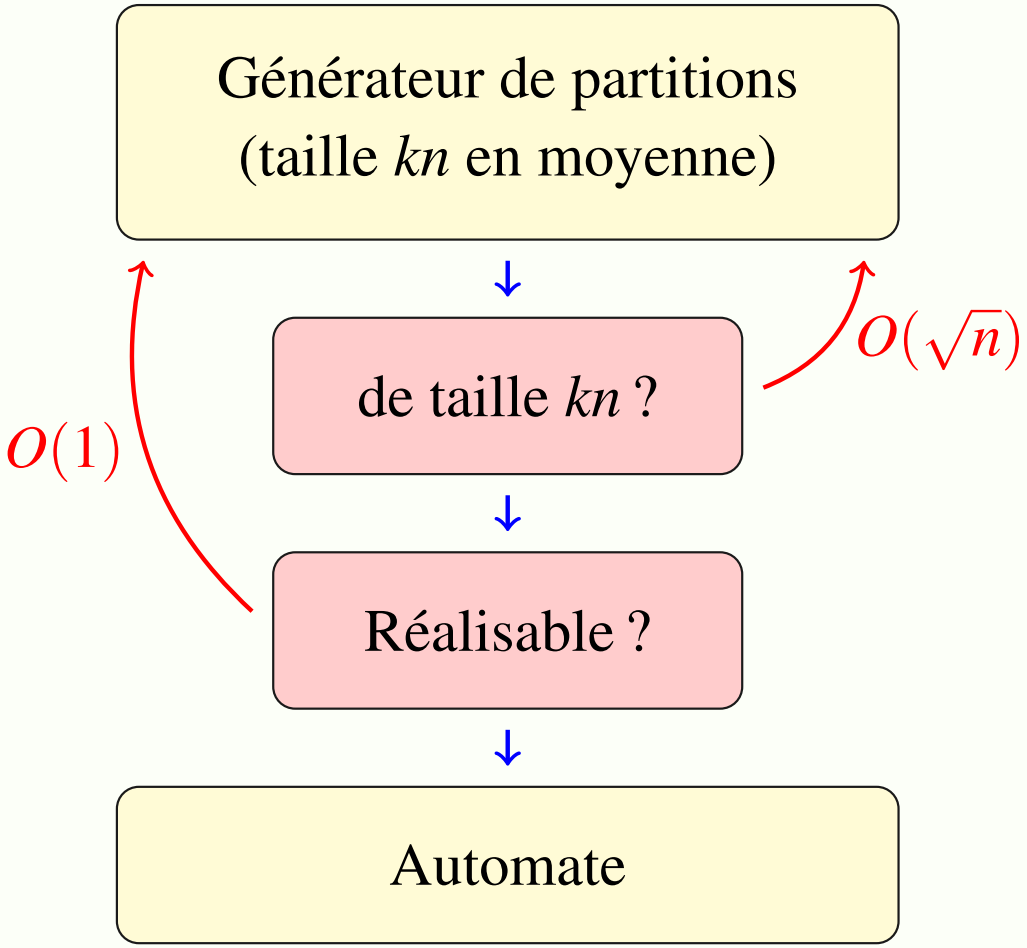
Peut-on faire mieux que du $\Theta(n\sqrt{n})$ pour engendrer les partitions ?


Générateur de partitions
(taille kn en moyenne)
de taille kn ?
Réalisable ?
Automate


Générateur de partitions
(taille kn en moyenne)
de taille kn ?
$O(\sqrt{n})$
Réalisable ?
Automate


Générateur de partitions
(taille kn en moyenne)
de taille kn ?
$O(\sqrt{n})$
Réalisable ?
Automate


Générateur de partitions
(taille kn en moyenne)
de taille kn ?
Réalisable ?
Automate
$O(\sqrt{n})$
$O(1)$


Générateur de partitions
(taille kn en moyenne)
de taille kn ?
Réalisable ?
Automate
$O(\sqrt{n})$
$O(1)$

Générateurs d'automates

Théorème [Bassino, Nicaud 07]

En utilisant un générateur de Boltzmann, on peut tirer au sort uniformément au hasard des automates déterministe, complets et accessibles à n états en temps moyen $\Theta(n^{3/2})$.

Générateurs d'automates

Théorème [Bassino, Nicaud 07]

En utilisant un générateur de Boltzmann, on peut tirer au sort uniformément au hasard des automates déterministe, complets et accessibles à n états en temps moyen $\Theta(n^{3/2})$.

Extensions :

- Automates incomplets [Bassino, David, Nicaud 2008]
- Nombre fixé de transitions manquantes [Héam, Nicaud, Schmitz 2010]
- Transducteurs déterministes en entrée [Héam, Nicaud, Schmitz 2010]

Résultats expérimentaux

Temps moyen (en secondes) nécessaire pour engendrer un automate déterministe accessible complet à n états sur un alphabet à k lettres

n	100	500	1 000	5 000	10 000
$k = 2$	0.0014	0.010	0.035	0.37	1.1
$k = 3$	0.0017	0.014	0.045	0.51	1.4
$k = 4$	0.0021	0.020	0.054	0.55	1.8

- Résultats obtenus avec la bibliothèque C++ REGAL développée par J. David
- 20 000 automates de chaque taille ont été engendrés.

Automates minimaux (expérimentalement)

Proportion d'automates minimaux

Taille	10	100	500	1 000	5 000	10 000
$k = 2$ (%)	81.33	85.06	85.32	85.09	85.32	85,24
$k = 3$ (%)	96.92	99.78	99.99	99.98	99.99	99.99
$k = 4$ (%)	99.53	99.99	99.99	99.99	99.99	99.99

- Résultats obtenus avec la bibliothèque C++ REGAL développée par J. David
- 20 000 automates de chaque taille ont été engendrés.
- Générateur par rejet dont l'efficacité n'est prouvée théoriquement

Automates minimaux (expérimentalement)

Proportion d'automates minimaux

Taille	10	100	500	1 000	5 000	10 000
$k = 2$ (%)	81.33	85.06	85.32	85.09	85.32	85,24
$k = 3$ (%)	96.92	99.78	99.99	99.98	99.99	99.99
$k = 4$ (%)	99.53	99.99	99.99	99.99	99.99	99.99

- Résultats obtenus avec la bibliothèque C++ REGAL développée par J. David
- 20 000 automates de chaque taille ont été engendrés.
- Générateur par rejet dont l'efficacité n'est prouvée théoriquement

En cours

L'énumération des automates minimaux (en collaboration avec Andrea Sportiello)

Analyse en moyenne :

algorithme de Moore

Algorithmes de minimisation

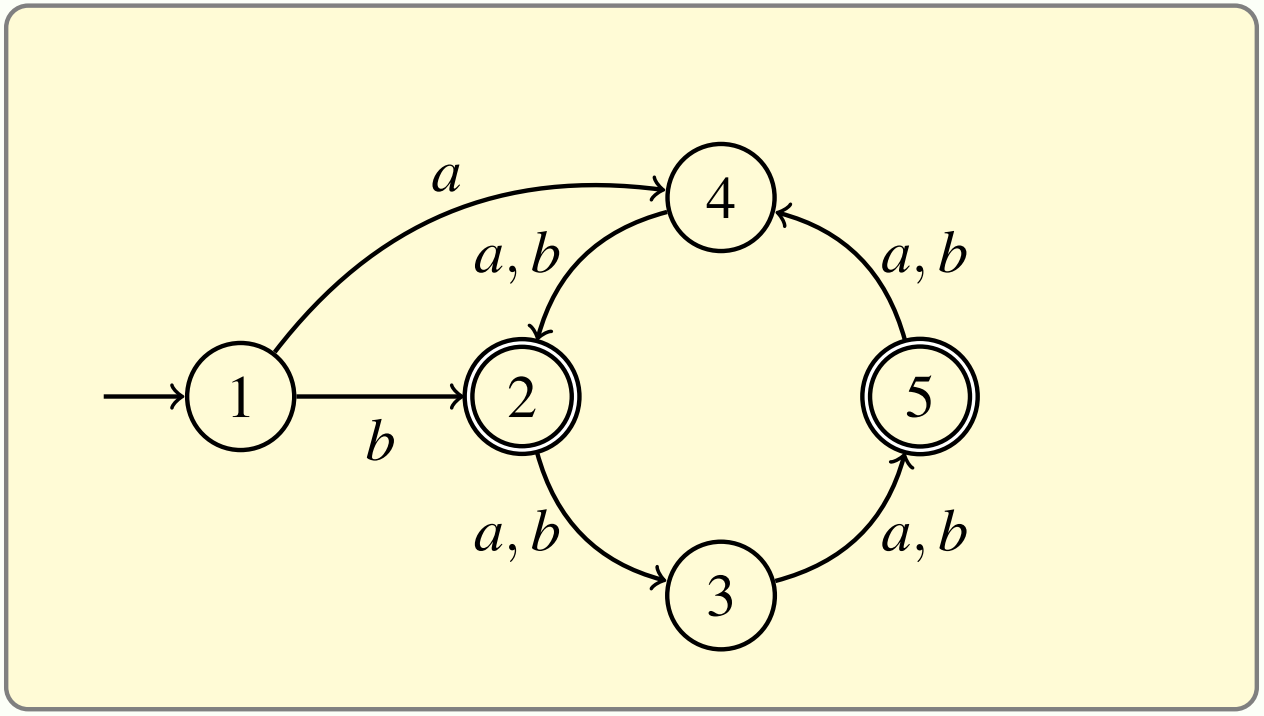
- A tout langage régulier est associé de manière canonique son automate minimal.
- C'est le plus petit automate accessible déterministe et complet qui reconnaît ce langage.

- Algorithme de Moore, en $O(n^2)$ dans le pire cas [Moore 56]
- Algorithme de Hopcroft, en $O(n \log n)$ dans le pire cas [Hopcroft 71]

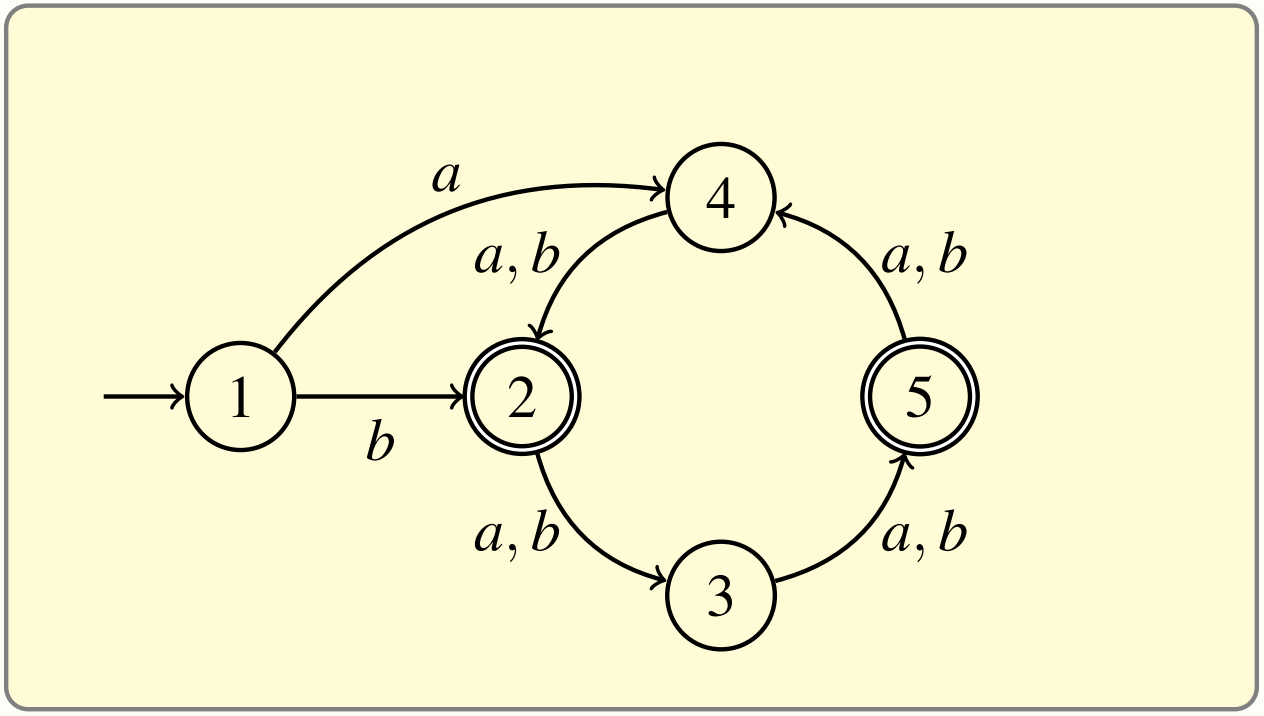
Question

Que peut-on dire de leurs complexités en moyenne ?

Equivalence de Nerode

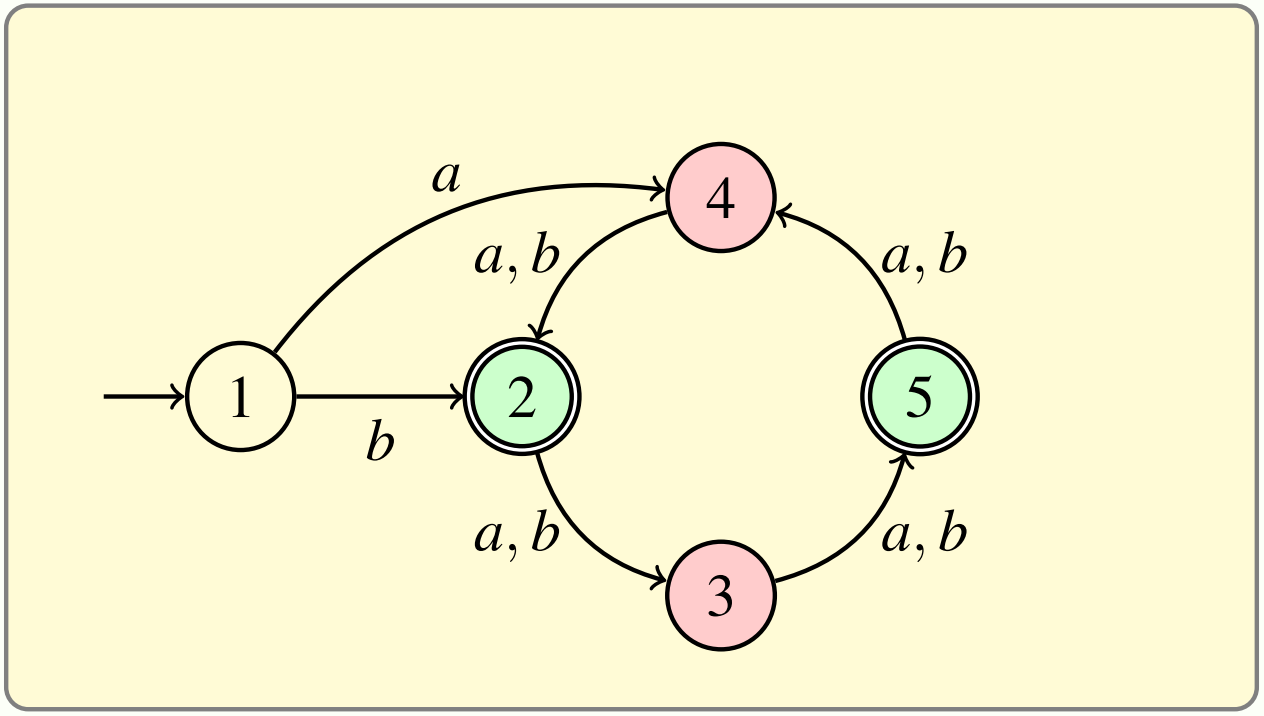


Equivalence de Nerode



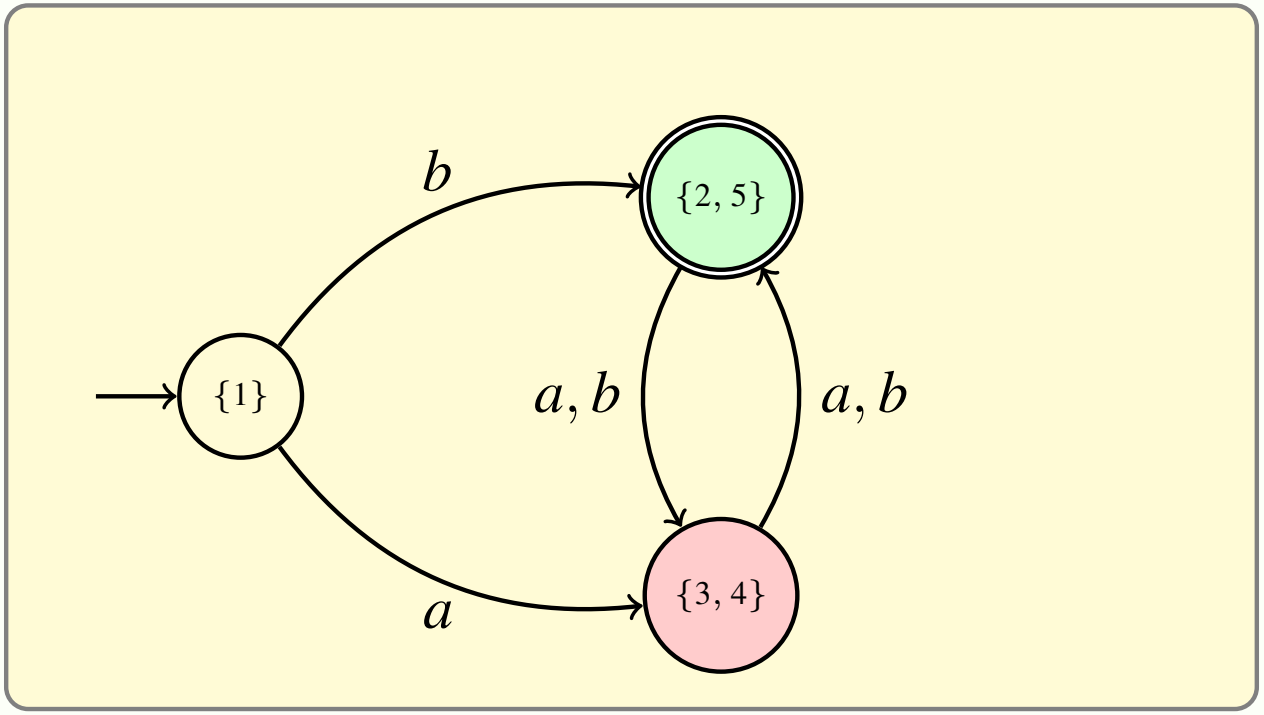
- On note $L_p = \{u \mid p \cdot u \in F\}$ le langage reconnu en prenant p comme état initial.
- Equivalence de Nérode : $p \sim q$ ssi $L_p = L_q$

Equivalence de Nerode



- On note $L_p = \{u \mid p \cdot u \in F\}$ le langage reconnu en prenant p comme état initial.
- Equivalence de Nérode : $p \sim q$ ssi $L_p = L_q$

Automate minimisé



- On fusionne les états équivalent.
- On obtient l'automate minimal du langage.

L'algorithme de Moore

- $p \sim q \Leftrightarrow L_p = L_q$
- $p \sim_i q \Leftrightarrow L_p \cap A^{\leq i} = L_q \cap A^{\leq i}$
- $\sim_{i+1}$ se calcule à partir de $\sim_i$
- Si $\sim_{i+1} = \sim_i$ alors $\sim = \sim_i$

L'algorithme de Moore

- $p \sim q \Leftrightarrow L_p = L_q$
- $p \sim_i q \Leftrightarrow L_p \cap A^{\leq i} = L_q \cap A^{\leq i}$
- $\sim_{i+1}$ se calcule à partir de $\sim_i$
- Si $\sim_{i+1} = \sim_i$ alors $\sim = \sim_i$

Moore($\mathcal{A}$)

1. Calculer $\sim_0$
2. Tant que $\sim_{i+1} \neq \sim_i$
   3. $i := i + 1$
   4. Calculer $\sim_{i+1}$
5. Retourner $\mathcal{A}/\sim_i$

L'algorithme de Moore

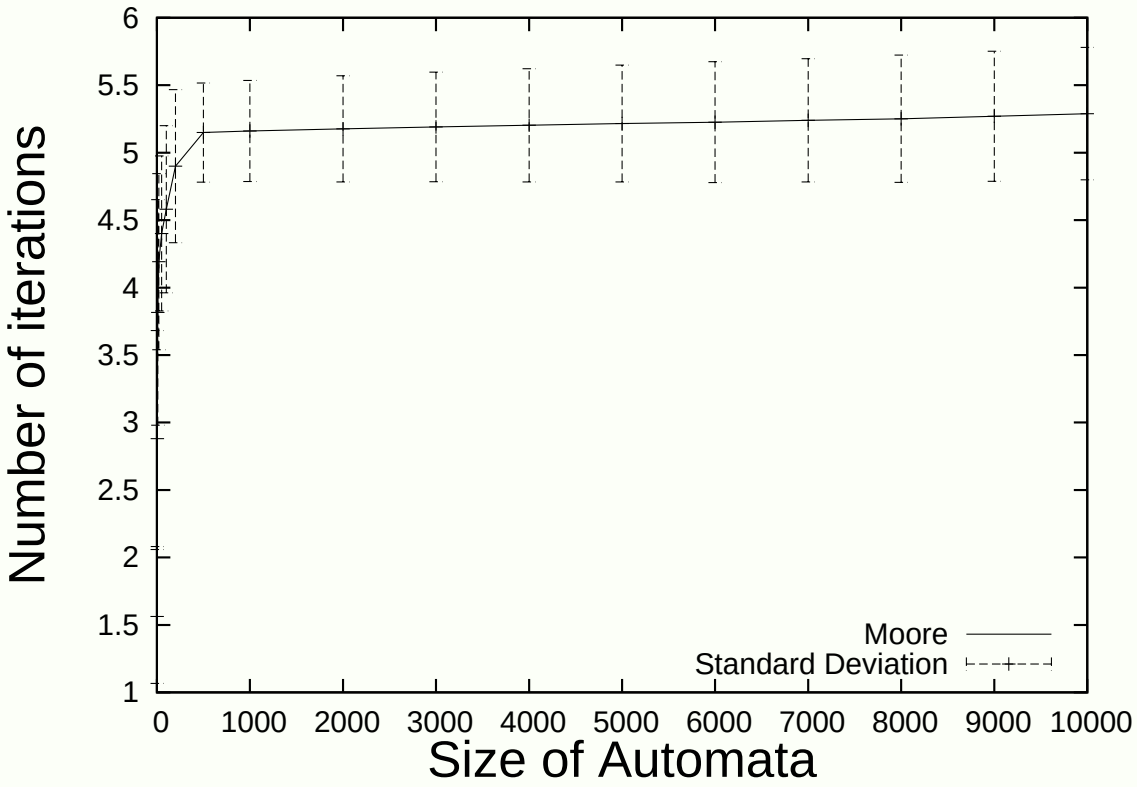
- $p \sim q \Leftrightarrow L_p = L_q$
- $p \sim_i q \Leftrightarrow L_p \cap A^{\leq i} = L_q \cap A^{\leq i}$
- $\sim_{i+1}$ se calcule à partir de $\sim_i$
- Si $\sim_{i+1} = \sim_i$ alors $\sim = \sim_i$

Moore($\mathcal{A}$)

1. Calculer $\sim_0$
2. Tant que $\sim_{i+1} \neq \sim_i$
   3. $i := i + 1$
   4. Calculer $\sim_{i+1}$
5. Retourner $\mathcal{A}/\sim_i$

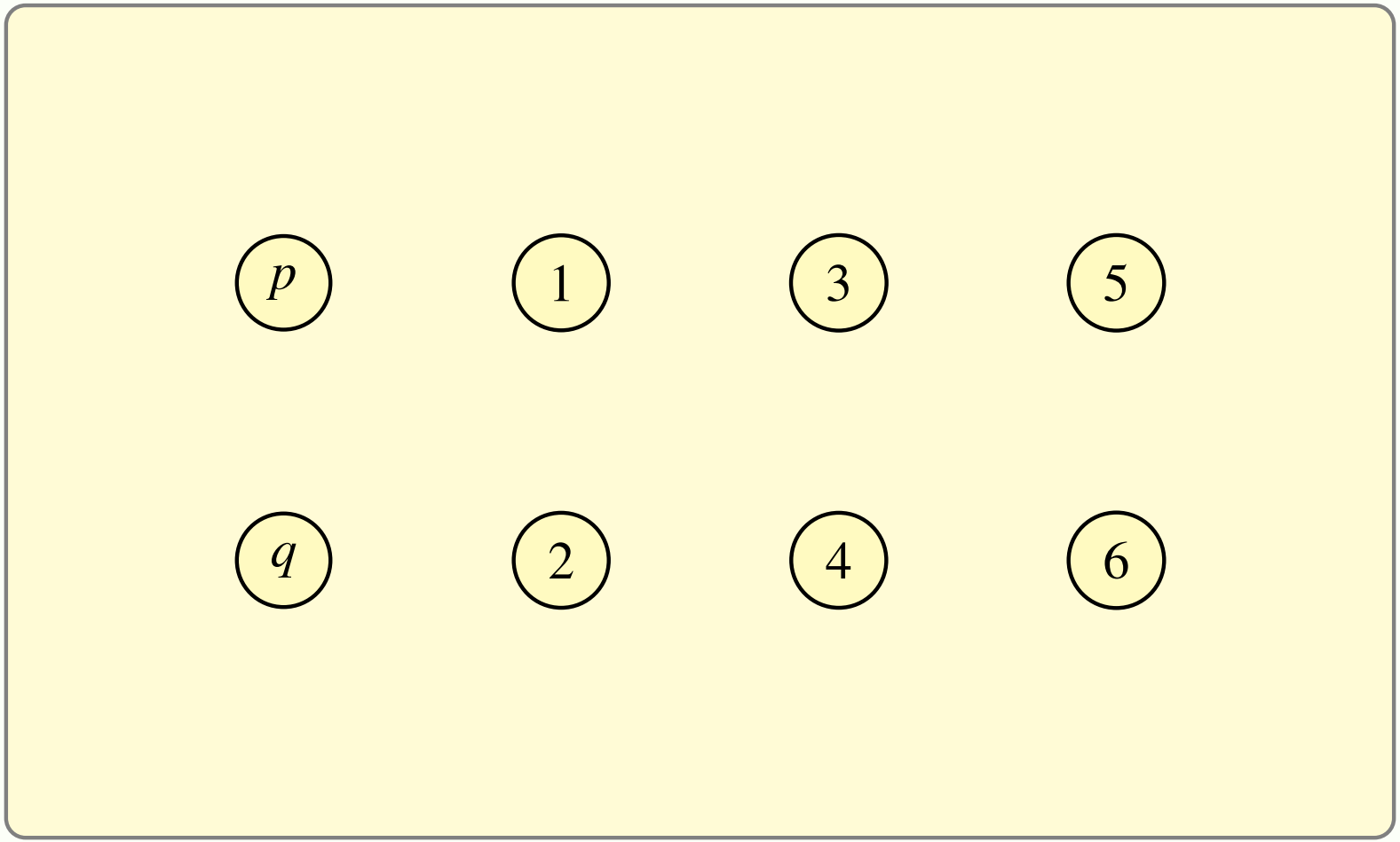
- Calculer $\sim_{i+1}$ se fait en temps $\Theta(kn)$
- Il y a au plus $n - 1$ itérations
- Complexité quadratique dans le pire cas.
- Combien y a-t-il d'itérations en moyenne ?

Nombre moyen d'itérations de l'algorithme de Moore



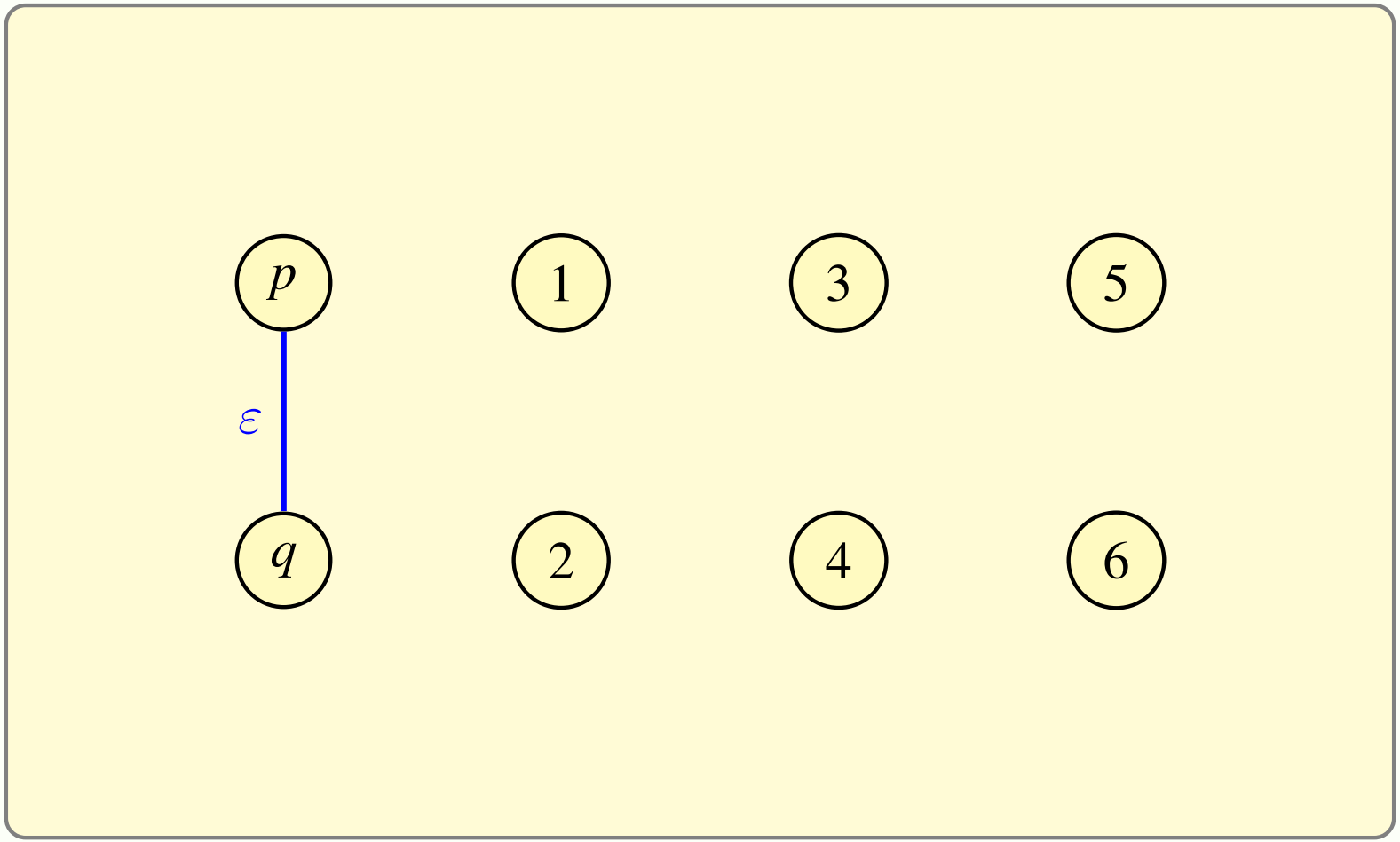
Equivalence et contraintes sur les états

On fixe le graphe de l'automate.
Si $u = abaab$ le plus court mot qui sépare L_p et L_q : $p \sim_4 q$ et $p \nsim_5 q$
Quels états terminaux peut-on choisir ?



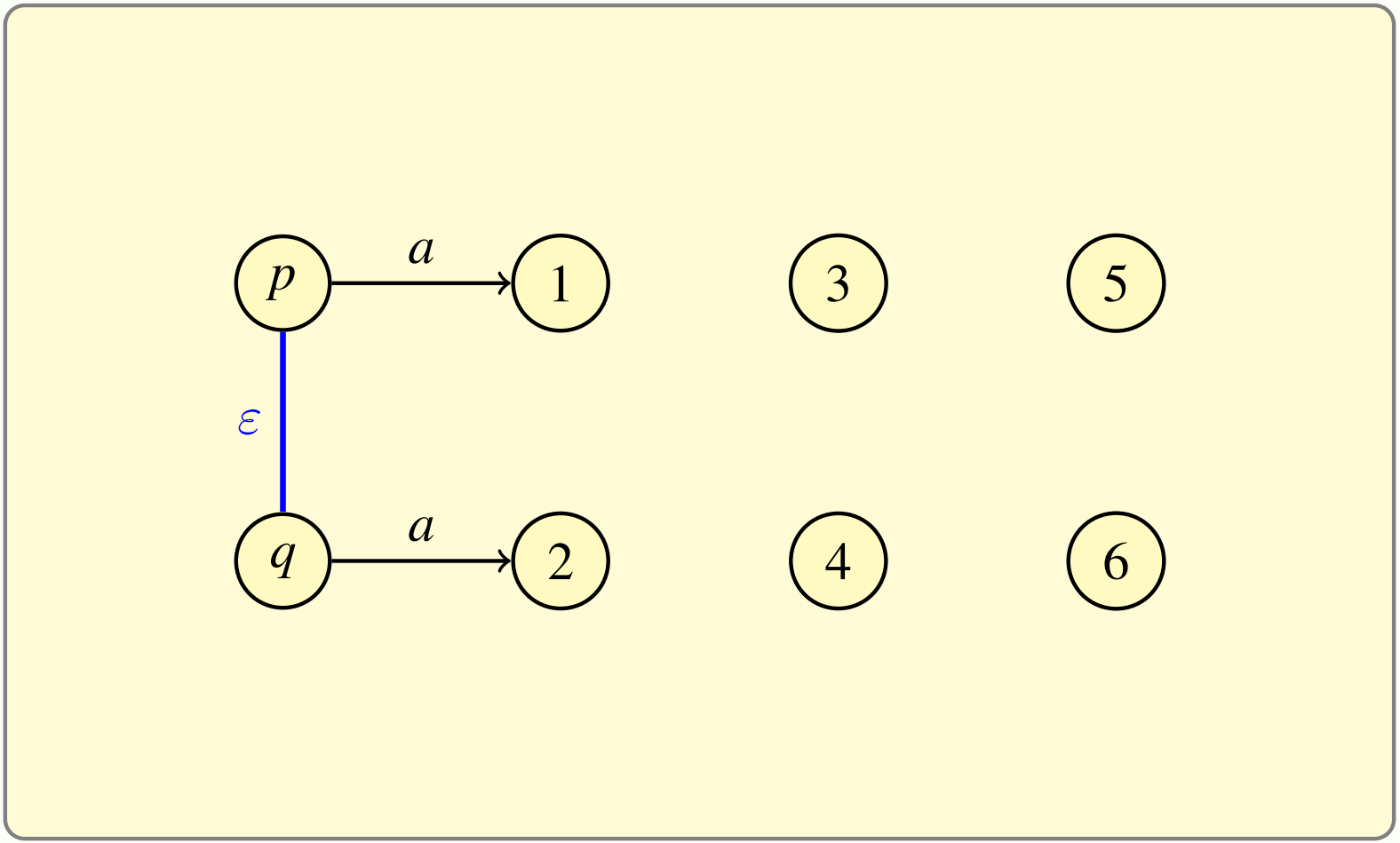
Equivalence et contraintes sur les états

On fixe le graphe de l'automate.
Si $u = abaab$ le plus court mot qui sépare L_p et L_q : $p \sim_4 q$ et $p \nsim_5 q$
Quels états terminaux peut-on choisir ?



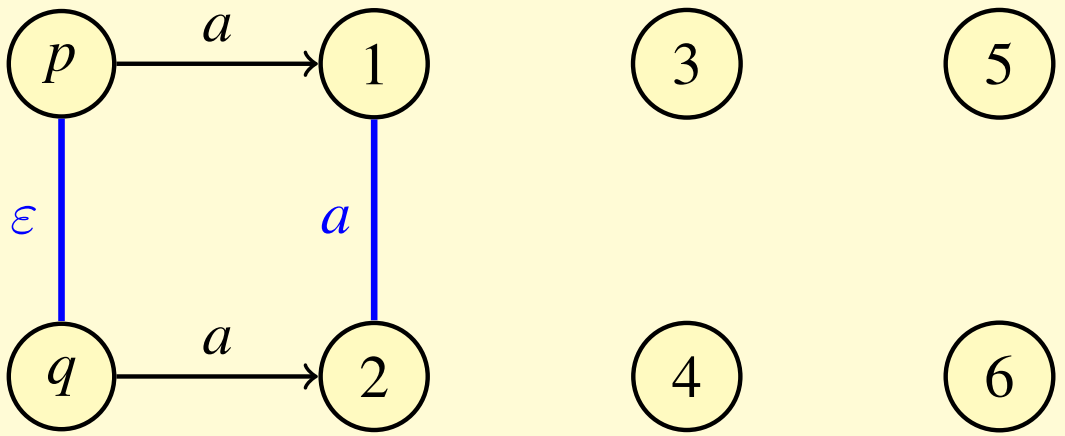
Equivalence et contraintes sur les états

On fixe le graphe de l'automate.
Si $u = abaab$ le plus court mot qui sépare L_p et L_q : $p \sim_4 q$ et $p \nsim_5 q$
Quels états terminaux peut-on choisir ?



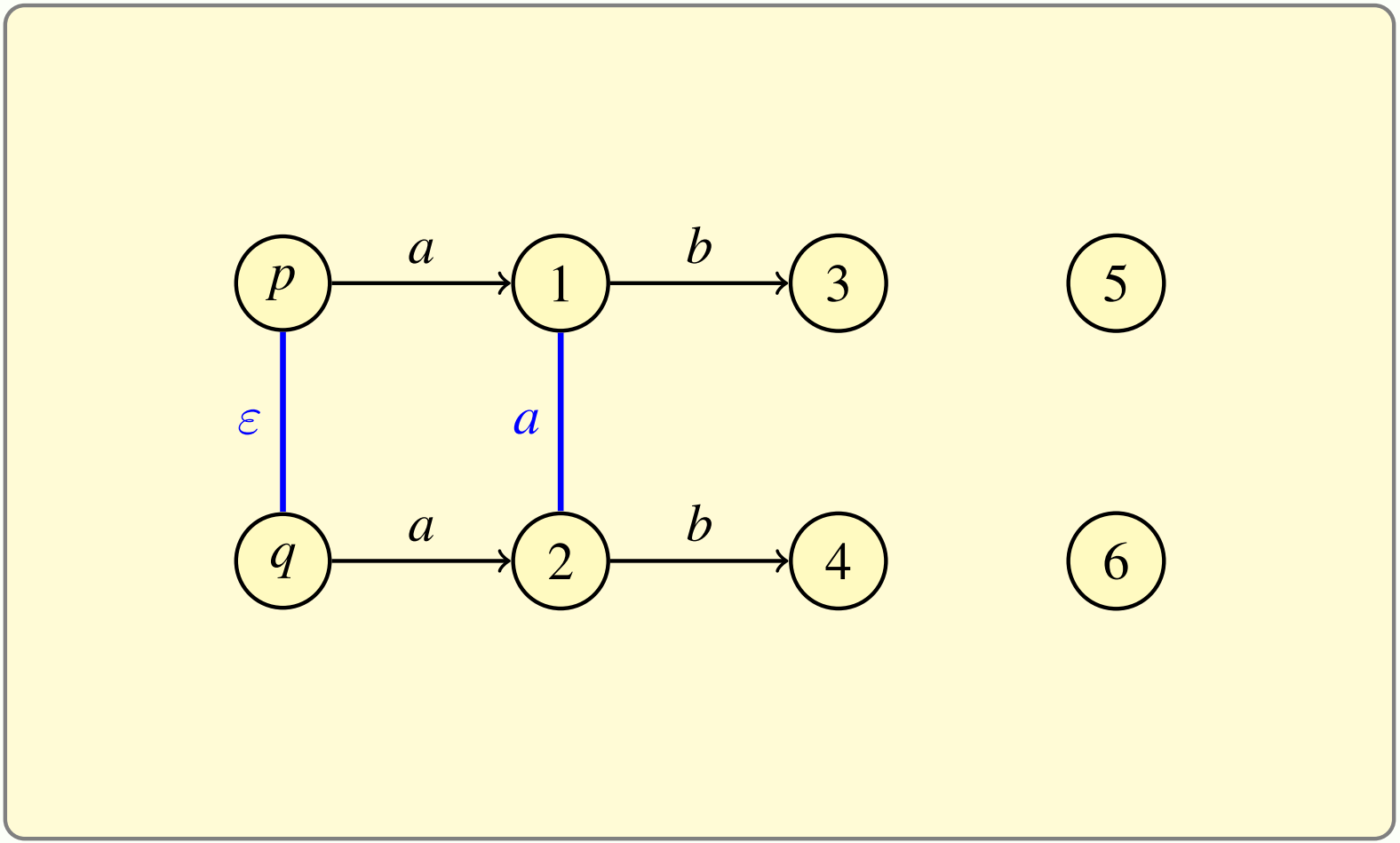
Equivalence et contraintes sur les états

On fixe le graphe de l'automate.
Si $u = abaab$ le plus court mot qui sépare L_p et L_q : $p \sim_4 q$ et $p \not\sim_5 q$
Quels états terminaux peut-on choisir ?



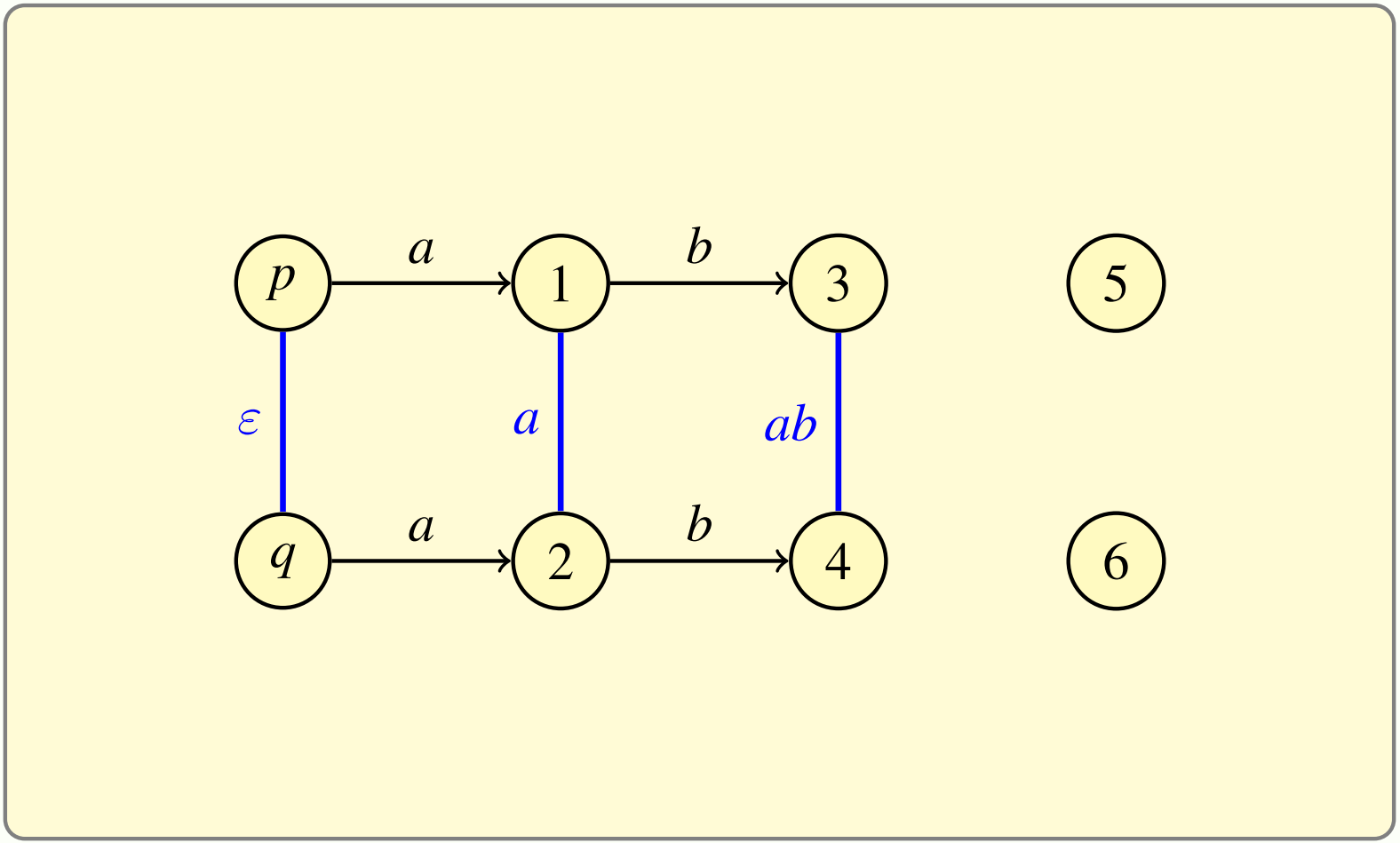
Equivalence et contraintes sur les états

On fixe le graphe de l'automate.
Si $u = abaab$ le plus court mot qui sépare L_p et L_q : $p \sim_4 q$ et $p \nsim_5 q$
Quels états terminaux peut-on choisir ?



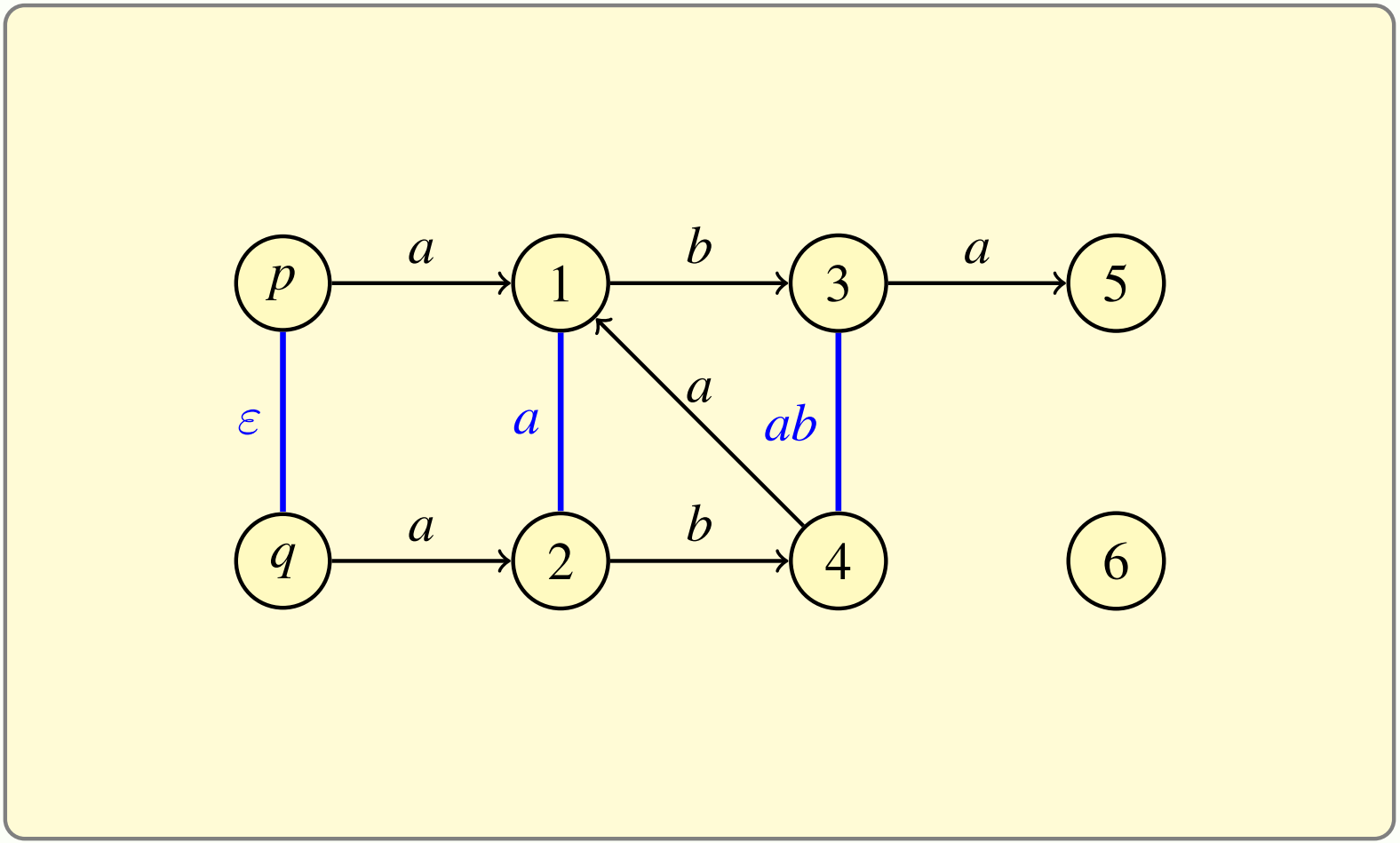
Equivalence et contraintes sur les états

On fixe le graphe de l'automate.
Si $u = abaab$ le plus court mot qui sépare L_p et L_q : $p \sim_4 q$ et $p \not\sim_5 q$
Quels états terminaux peut-on choisir ?



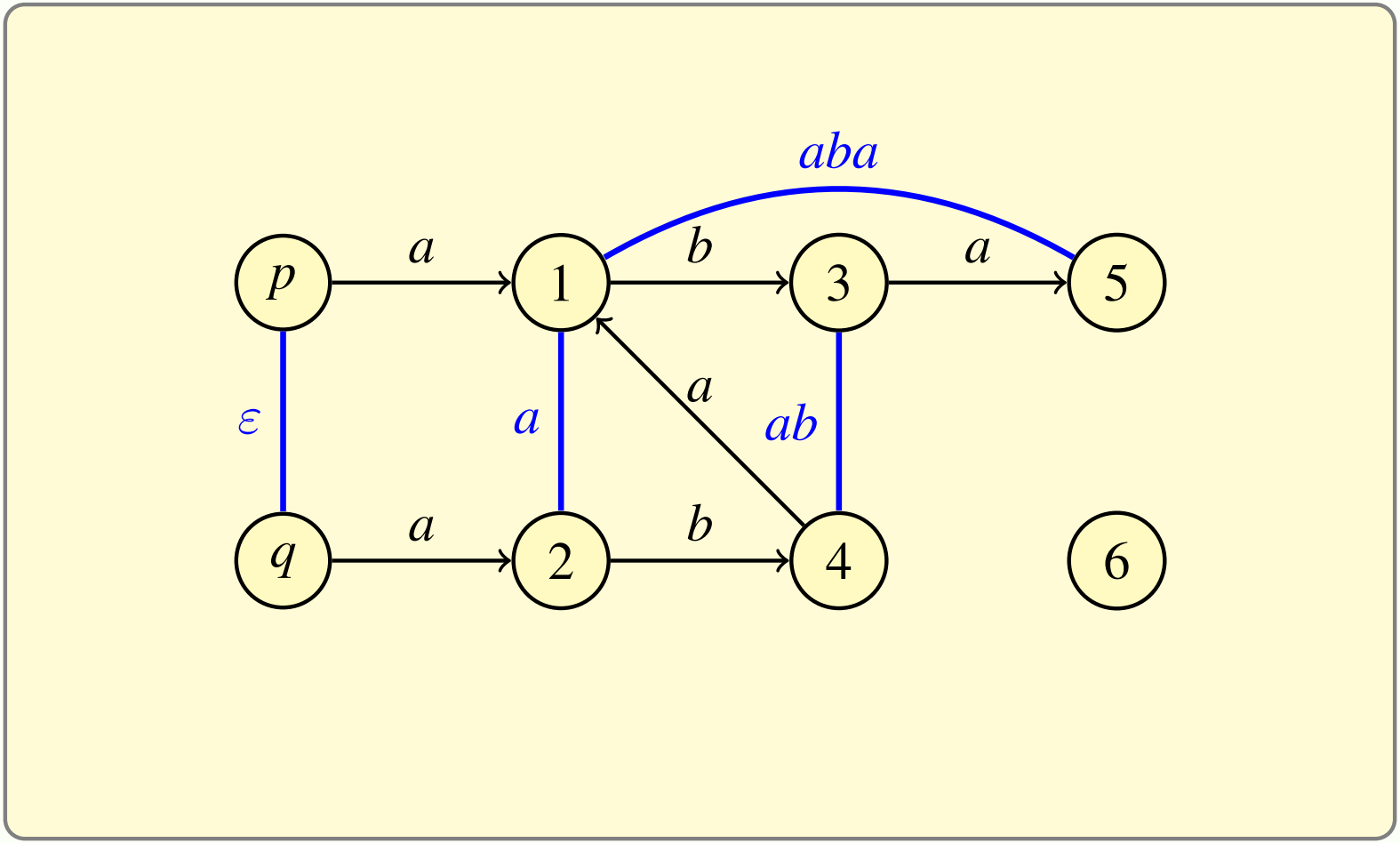
Equivalence et contraintes sur les états

On fixe le graphe de l'automate.
Si $u = abaab$ le plus court mot qui sépare L_p et L_q : $p \sim_4 q$ et $p \nsim_5 q$
Quels états terminaux peut-on choisir ?



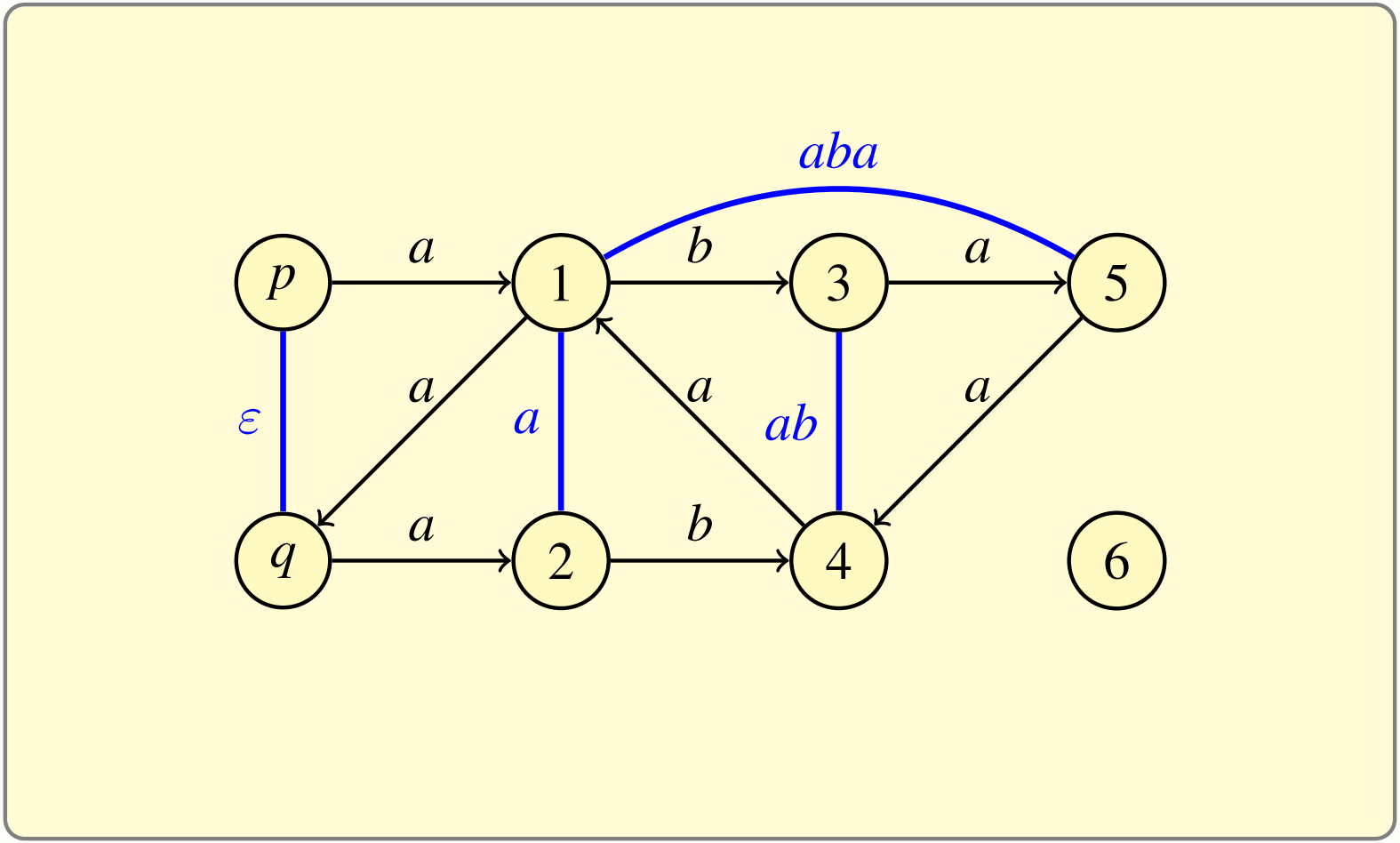
Equivalence et contraintes sur les états

On fixe le graphe de l'automate.
Si $u = abaab$ le plus court mot qui sépare L_p et L_q : $p \sim_4 q$ et $p \not\sim_5 q$
Quels états terminaux peut-on choisir ?



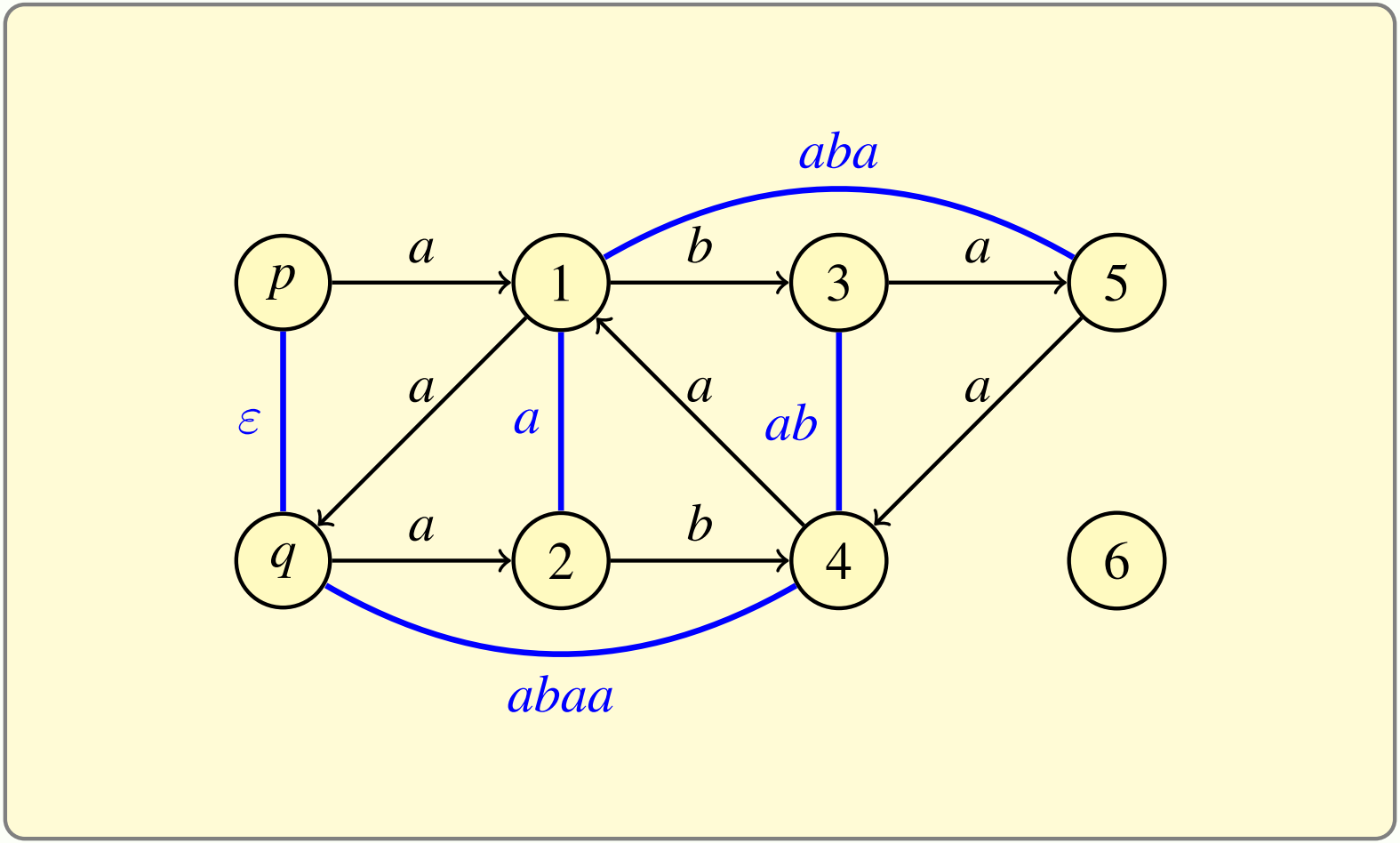
Equivalence et contraintes sur les états

On fixe le graphe de l'automate.
Si $u = abaab$ le plus court mot qui sépare L_p et L_q : $p \sim_4 q$ et $p \not\sim_5 q$
Quels états terminaux peut-on choisir ?



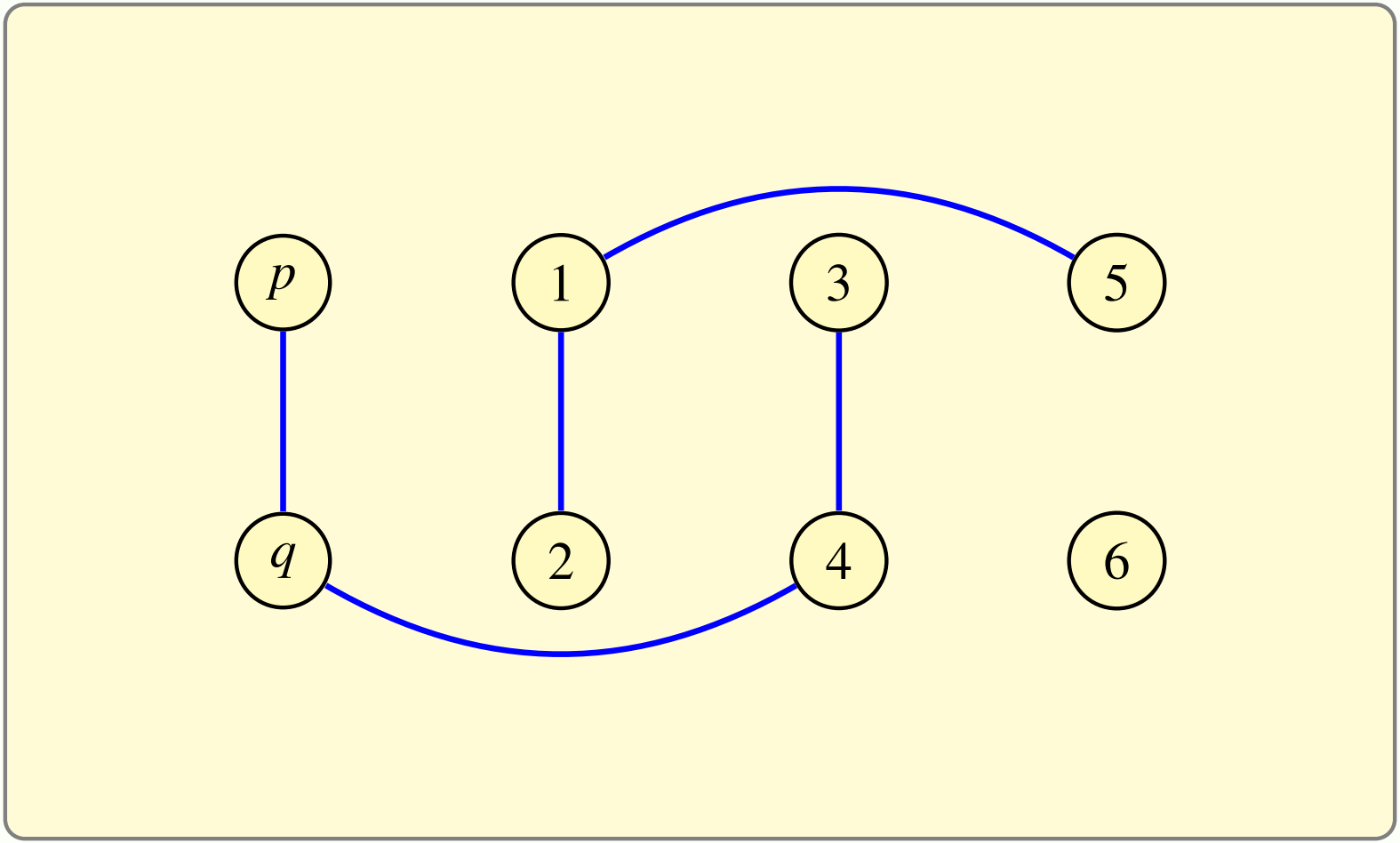
Equivalence et contraintes sur les états

On fixe le graphe de l'automate.
Si $u = abaab$ le plus court mot qui sépare L_p et L_q : $p \sim_4 q$ et $p \not\sim_5 q$
Quels états terminaux peut-on choisir ?



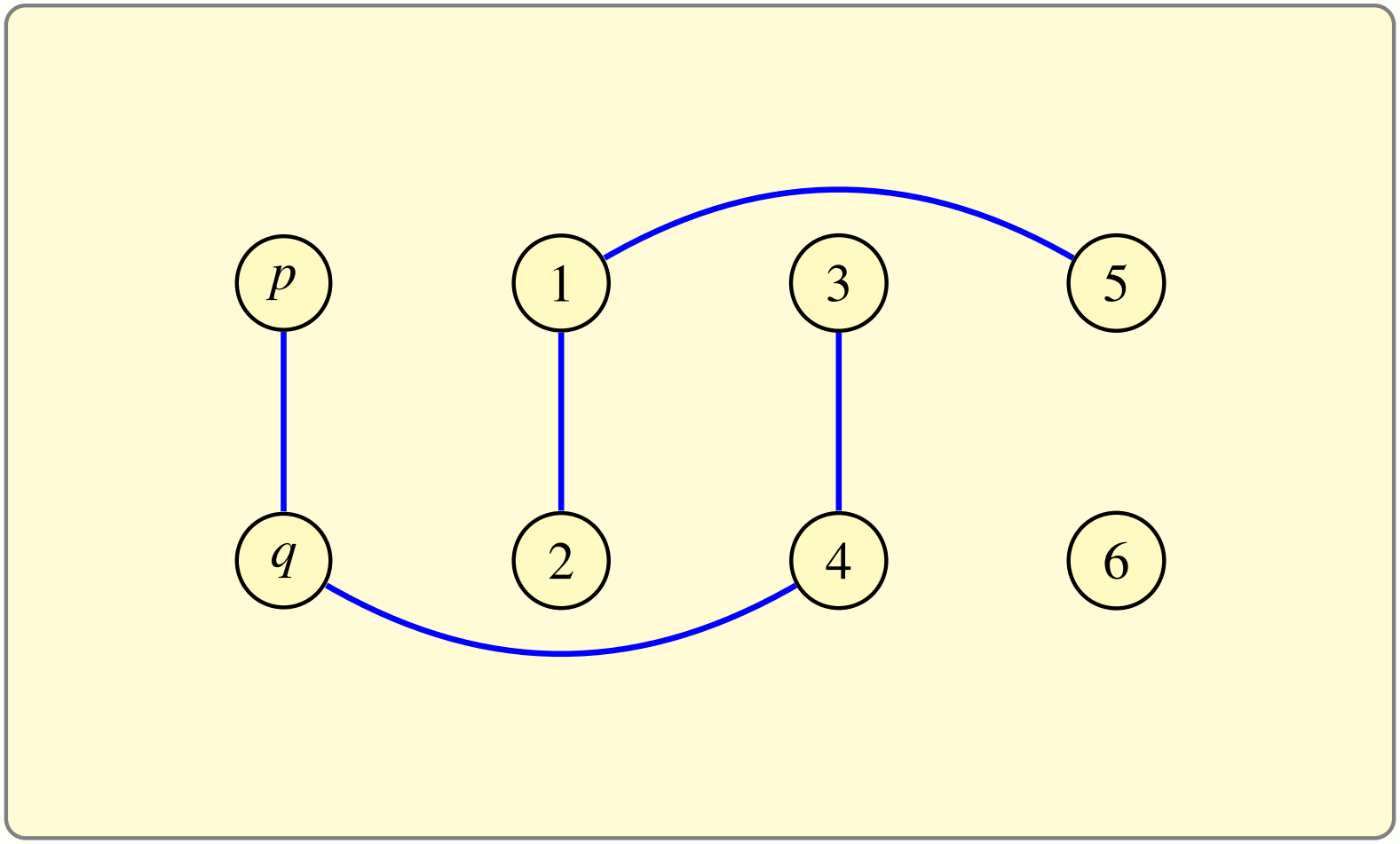
Equivalence et contraintes sur les états

On fixe le graphe de l'automate.
$u = abaab$ le plus court mot qui sépare L_p et L_q : $p \sim_4 q$ et $p \not\sim_5 q$.
Quels états terminaux peut-on choisir ?



Probabilité de satifaire les contraintes

Probabilité que le choix des états terminaux satisfasse les contraintes : $2^{3-8} = 2^{-5}$ pour un mot de longueur 5



Proposition

Si on fixe le graphe de l'automate et deux états p et q, $p \sim_{\ell-1} q$ et $p \nsim_{\ell} q$ avec probabilité au plus $2^{-\ell}$, en tirant uniformément l'ensemble d'états terminaux.

Proposition

Si on fixe le graphe de l'automate et deux états p et q, $p \sim_{\ell-1} q$ et $p \not\sim_{\ell} q$ avec probabilité au plus $2^{-\ell}$, en tirant uniformément l'ensemble d'états terminaux.

Théorème [Bassino, David, Nicaud 09]

Pour la distribution uniforme sur les automates déterministes complets et accessibles, le nombre moyen d'itérations de l'algorithme de Moore est en $O(\log n)$. Sa complexité moyenne est donc en $O(n \log n)$.

Proposition

Si on fixe le graphe de l'automate et deux états p et q, $p \sim_{\ell-1} q$ et $p \not\sim_{\ell} q$ avec probabilité au plus $2^{-\ell}$, en tirant uniformément l'ensemble d'états terminaux.

Théorème [Bassino, David, Nicaud 09]

Pour la distribution uniforme sur les automates déterministes complets et accessibles, le nombre moyen d'itérations de l'algorithme de Moore est en $O(\log n)$. Sa complexité moyenne est donc en $O(n \log n)$.

Ce résultat a été généralisé aux distributions telles que la probabilité de respecter ℓ contraintes décroît exponentiellement avec ℓ.

Théorème [David 10]

Pour la distribution uniforme sur les automates déterministes complets (pas nécessairement accessibles), le nombre moyen d'itérations de l'algorithme de Moore est en $O(\log \log n)$. Sa complexité moyenne est donc en $O(n \log \log n)$.

Théorème [David 10]

Pour la distribution uniforme sur les automates déterministes complets (pas nécessairement accessibles), le nombre moyen d'itérations de l'algorithme de Moore est en $O(\log \log n)$. Sa complexité moyenne est donc en $O(n \log \log n)$.

Conjecture

Le résultat est vrai pour la distribution uniforme sur les automates déterministes complets et accessibles.

Pour ce résultat, J. David utilise la structure typique d'un automate aléatoire pour la distribution uniforme.

Expressions rationnelles et Automates de Glushkov

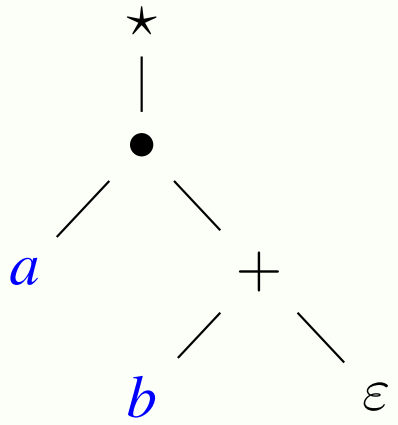
Expressions rationnelles

L'ensemble des expressions rationnelles non vides $\mathcal{R}$ sur un alphabet fini A est l'ensemble des mots sur l'alphabet $A \cup \{\epsilon, \cup, *, \cdot, (,)\}$ définis inductivement par

- $\epsilon \in \mathcal{R}, A \subset \mathcal{R}$
- Si $R \in \mathcal{R}$, $R^* \in \mathcal{R}$
- Si $R_1, R_2 \in \mathcal{R}$, $R_1 \cup R_2, R_1 \cdot R_2 \in \mathcal{R}$

Représentation par arbre

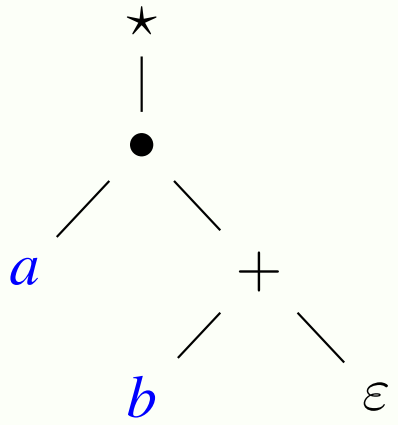
Une expression rationnelle peut-être vue comme un arbre :



- Arbre de $(a \cdot (b + \varepsilon))^*$
- La taille de l'expression est le nombre de nœuds, ici 6

Représentation par arbre

Une expression rationnelle peut-être vue comme un arbre :



- Arbre de $(a \cdot (b + \varepsilon))^*$
- La taille de l'expression est le nombre de nœuds, ici 6

Problème

Enumérer les langages réguliers selon la taille des plus petites expressions rationnelles qui les représentent est un problème difficile.

Etudier des automates non-déterministes particuliers : les automates de Glushkov

d'une expression rationnelle à un automate de Glushkov

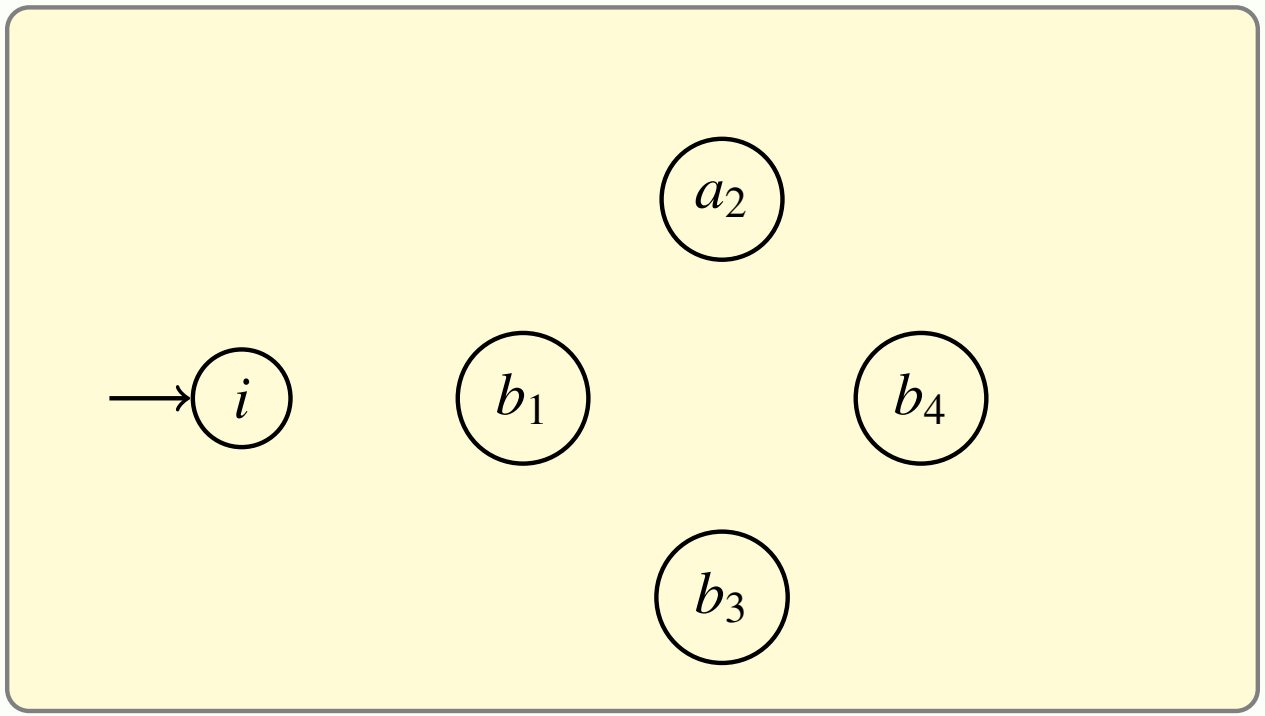
$$E = b^* \cdot (a + b \cdot b)^*,$$

d'une expression rationnelle à un automate de Glushkov

$E = b^* \cdot (a + b \cdot b)^*$, on distingue les lettres : $\tilde{E} = b_1^* \cdot (a_2 + b_3 \cdot b_4)^*$

d'une expression rationnelle à un automate de Glushkov

$E = b^* \cdot (a + b \cdot b)^*$, on distingue les lettres : $\tilde{E} = b_1^* \cdot (a_2 + b_3 \cdot b_4)^*$

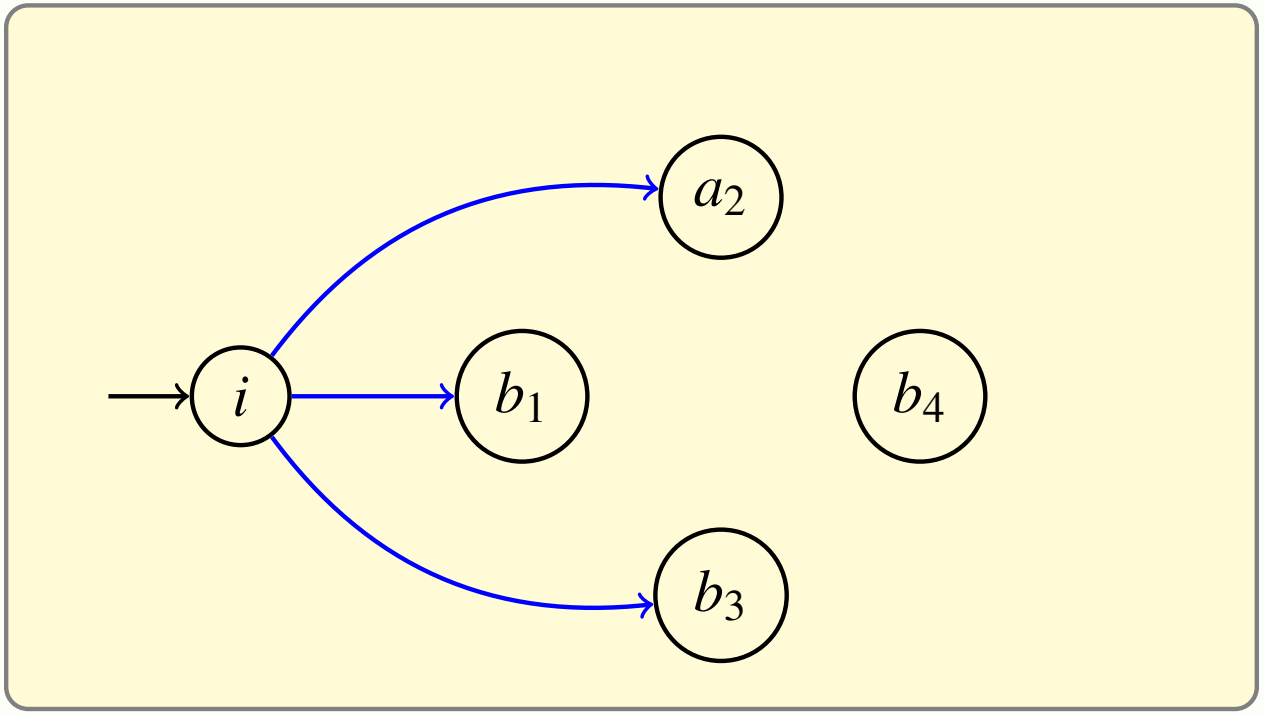


Ensembles

- $First(\tilde{E}) = \{\alpha \mid \alpha \text{ commence un mot de } \tilde{L}\}$
- $Last(\tilde{E}) = \{\alpha \mid \alpha \text{ termine un mot de } \tilde{L}\}$
- $Follow(\tilde{E}) = \{\alpha \to \beta \mid \alpha\beta \text{ est facteur d'un mot de } \tilde{L}\}$

Automates de Glushkov

$E = b^* \cdot (a + b \cdot b)^*$, on distingue les lettres : $\tilde{E} = b_1^* \cdot (a_2 + b_3 \cdot b_4)^*$

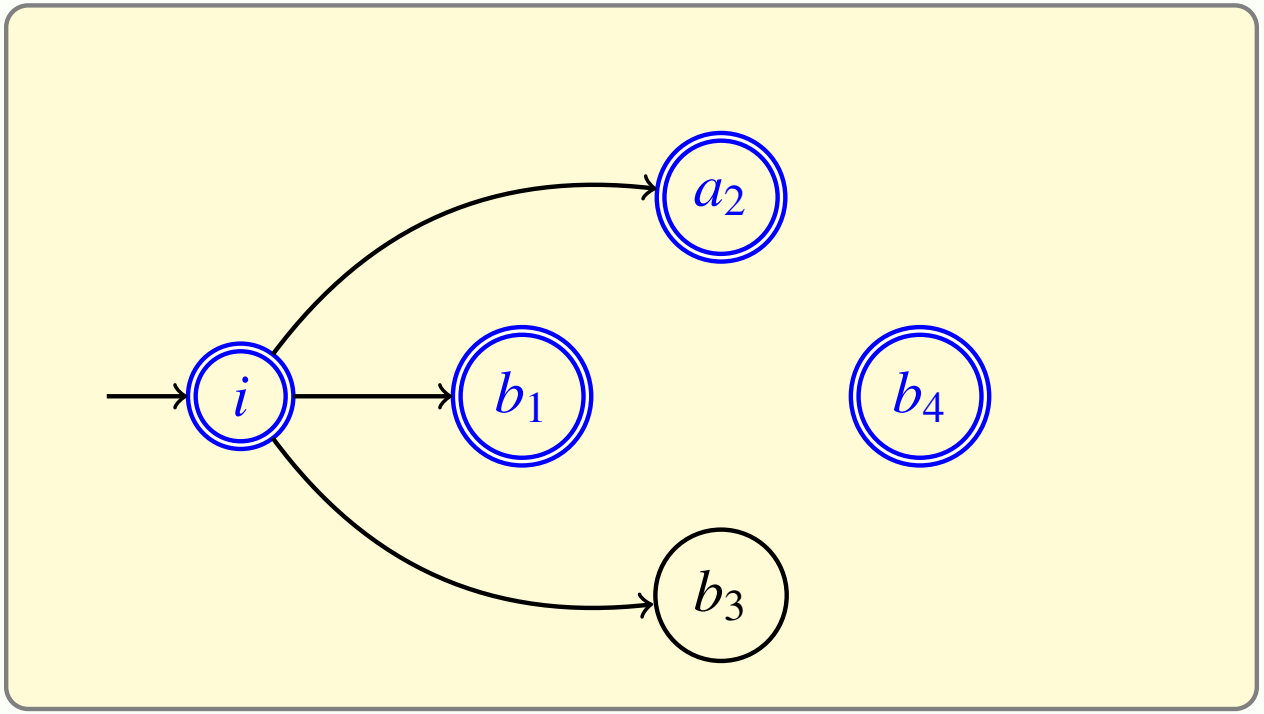


Ensembles

- $First(\tilde{E}) = \{\alpha \mid \alpha \text{ commence un mot de } \tilde{L}\}$
- $Last(\tilde{E}) = \{\alpha \mid \alpha \text{ termine un mot de } \tilde{L}\}$
- $Follow(\tilde{E}) = \{\alpha \to \beta \mid \alpha\beta \text{ est facteur d'un mot de } \tilde{L}\}$

Automates de Glushkov

$E = b^* \cdot (a + b \cdot b)^*$, on distingue les lettres : $\tilde{E} = b_1^* \cdot (a_2 + b_3 \cdot b_4)^*$

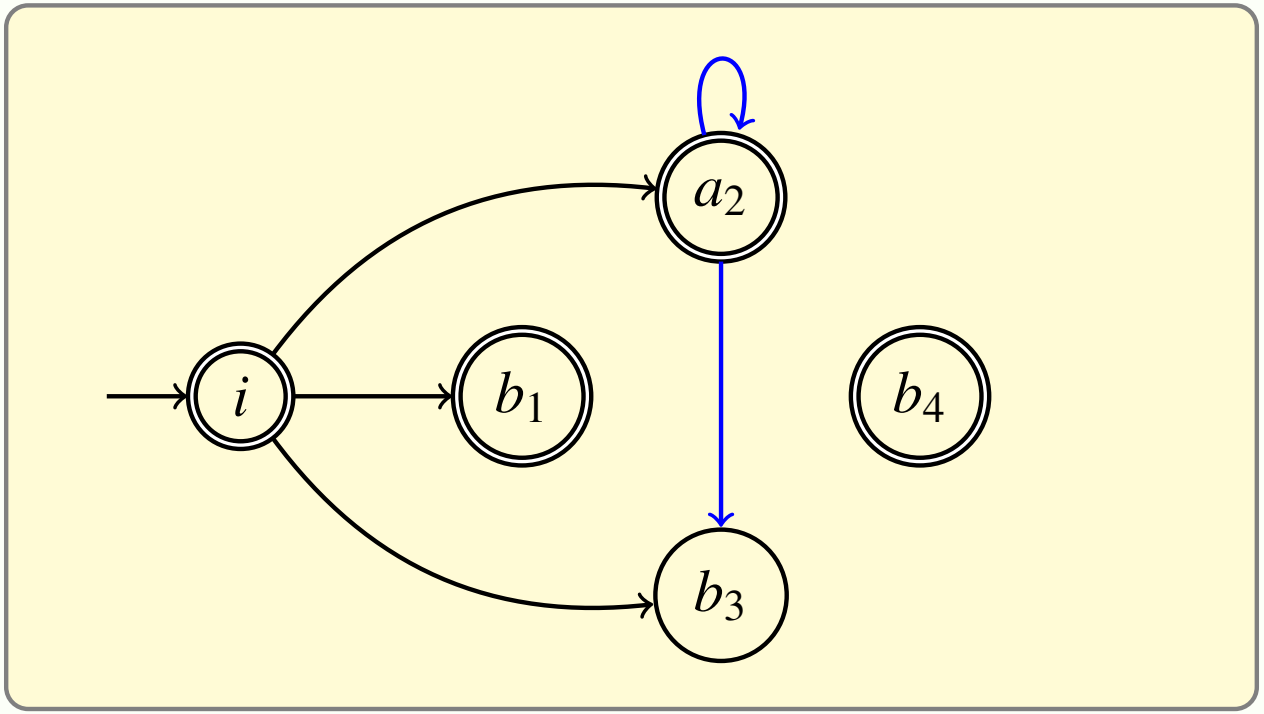


Ensembles

- $First(\tilde{E}) = \{\alpha \mid \alpha \text{ commence un mot de } \tilde{L}\}$
- $Last(\tilde{E}) = \{\alpha \mid \alpha \text{ termine un mot de } \tilde{L}\}$
- $Follow(\tilde{E}) = \{\alpha \to \beta \mid \alpha\beta \text{ est facteur d'un mot de } \tilde{L}\}$

Automates de Glushkov

$E = b^* \cdot (a + b \cdot b)^*$, on distingue les lettres : $\tilde{E} = b_1^* \cdot (a_2 + b_3 \cdot b_4)^*$

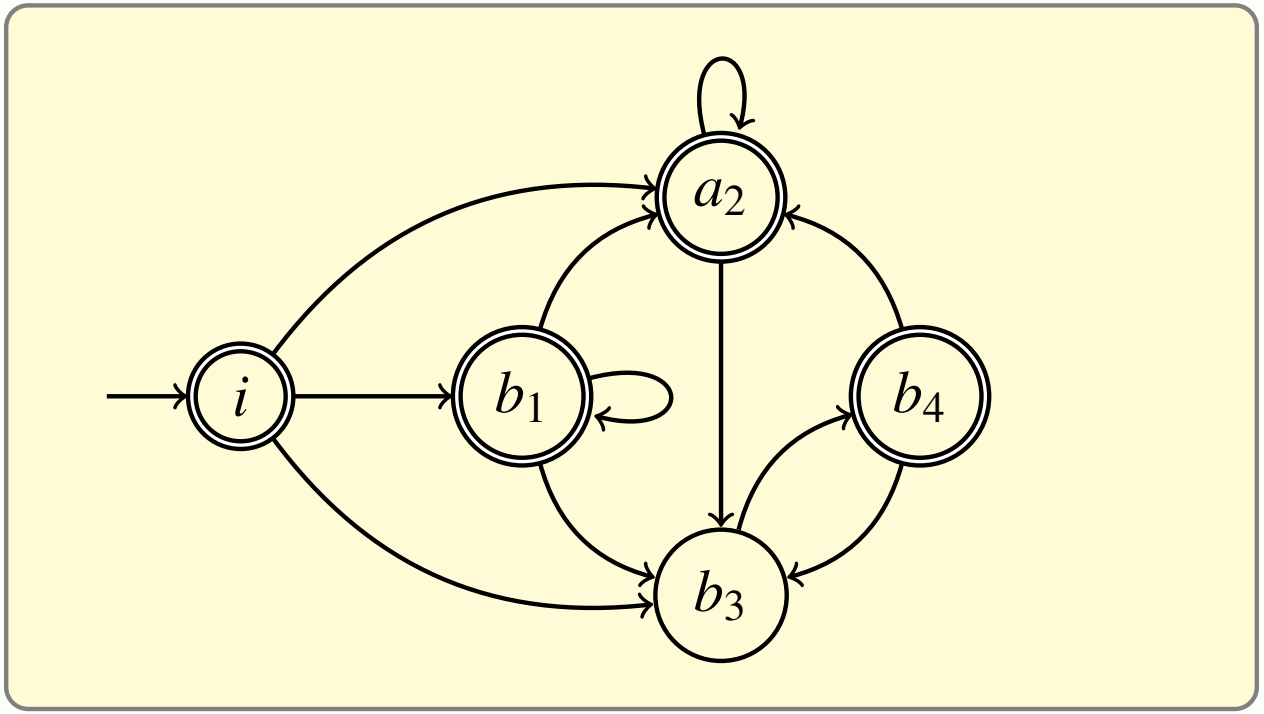


Ensembles

- $First(\tilde{E}) = \{\alpha \mid \alpha \text{ commence un mot de } \tilde{L}\}$
- $Last(\tilde{E}) = \{\alpha \mid \alpha \text{ termine un mot de } \tilde{L}\}$
- $Follow(\tilde{E}) = \{\alpha \to \beta \mid \alpha\beta \text{ est facteur d'un mot de } \tilde{L}\}$

Automates de Glushkov

$E = b^* \cdot (a + b \cdot b)^*$, on distingue les lettres : $\tilde{E} = b_1^* \cdot (a_2 + b_3 \cdot b_4)^*$

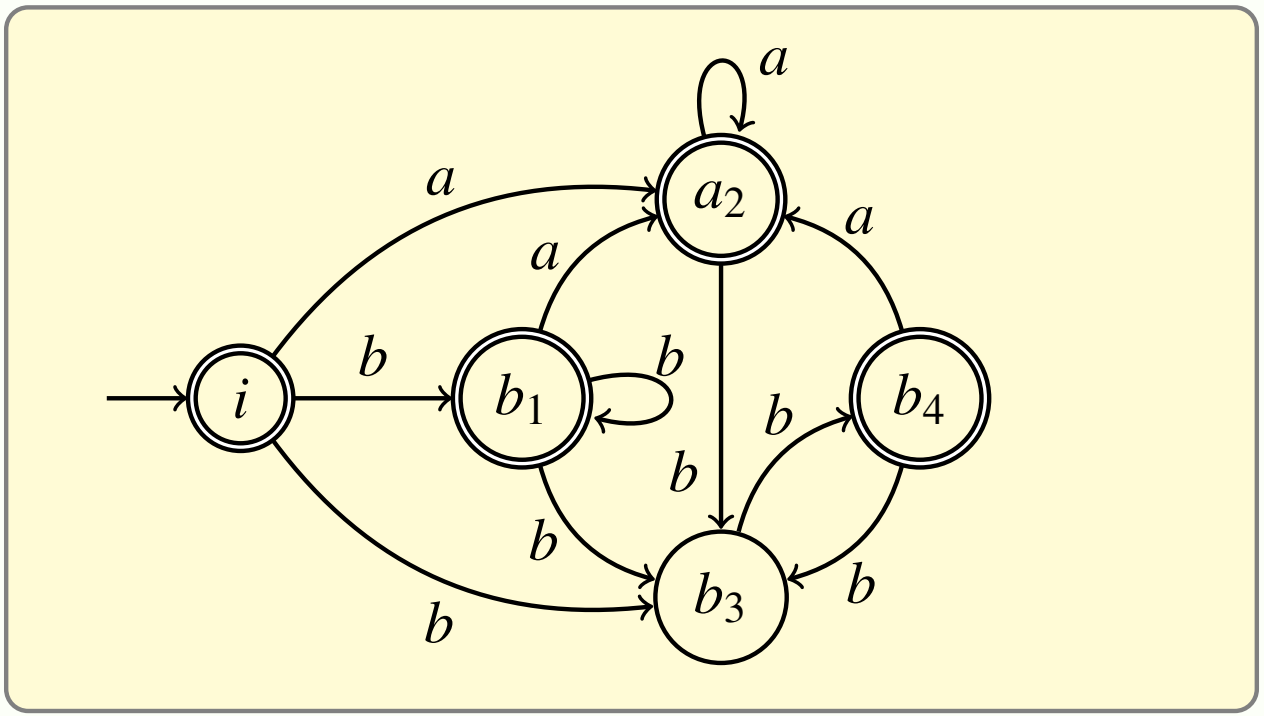


Ensembles

- $First(\tilde{E}) = \{\alpha \mid \alpha \text{ commence un mot de } \tilde{L}\}$
- $Last(\tilde{E}) = \{\alpha \mid \alpha \text{ termine un mot de } \tilde{L}\}$
- $Follow(\tilde{E}) = \{\alpha \to \beta \mid \alpha\beta \text{ est facteur d'un mot de } \tilde{L}\}$

Automates de Glushkov

$E = b^* \cdot (a + b \cdot b)^*$, on distingue les lettres : $\tilde{E} = b_1^* \cdot (a_2 + b_3 \cdot b_4)^*$



Ensembles

- $First(\tilde{E}) = \{\alpha \mid \alpha \text{ commence un mot de } \tilde{L}\}$
- $Last(\tilde{E}) = \{\alpha \mid \alpha \text{ termine un mot de } \tilde{L}\}$
- $Follow(\tilde{E}) = \{\alpha \to \beta \mid \alpha\beta \text{ est facteur d'un mot de } \tilde{L}\}$

Nombre de transitions d'un automates de Glushkov

Pire des cas

Dans le pire des cas, le nombre de transitions est en $\Theta(n^2)$.

Nombre de transitions d'un automates de Glushkov

Pire des cas

Dans le pire des cas, le nombre de transitions est en $\Theta(n^2)$.

Théorème [Nicaud 2009]

Pour la distribution uniforme sur les expressions, le nombre moyen de transitions est en $\Theta(n)$.

Nombre de transitions d'un automates de Glushkov

Pire des cas

Dans le pire des cas, le nombre de transitions est en $\Theta(n^2)$.

Théorème [Nicaud 2009]

Pour la distribution uniforme sur les expressions, le nombre moyen de transitions est en $\Theta(n)$.

Théorème [Nicaud, Pivoteau, Razet 2010]

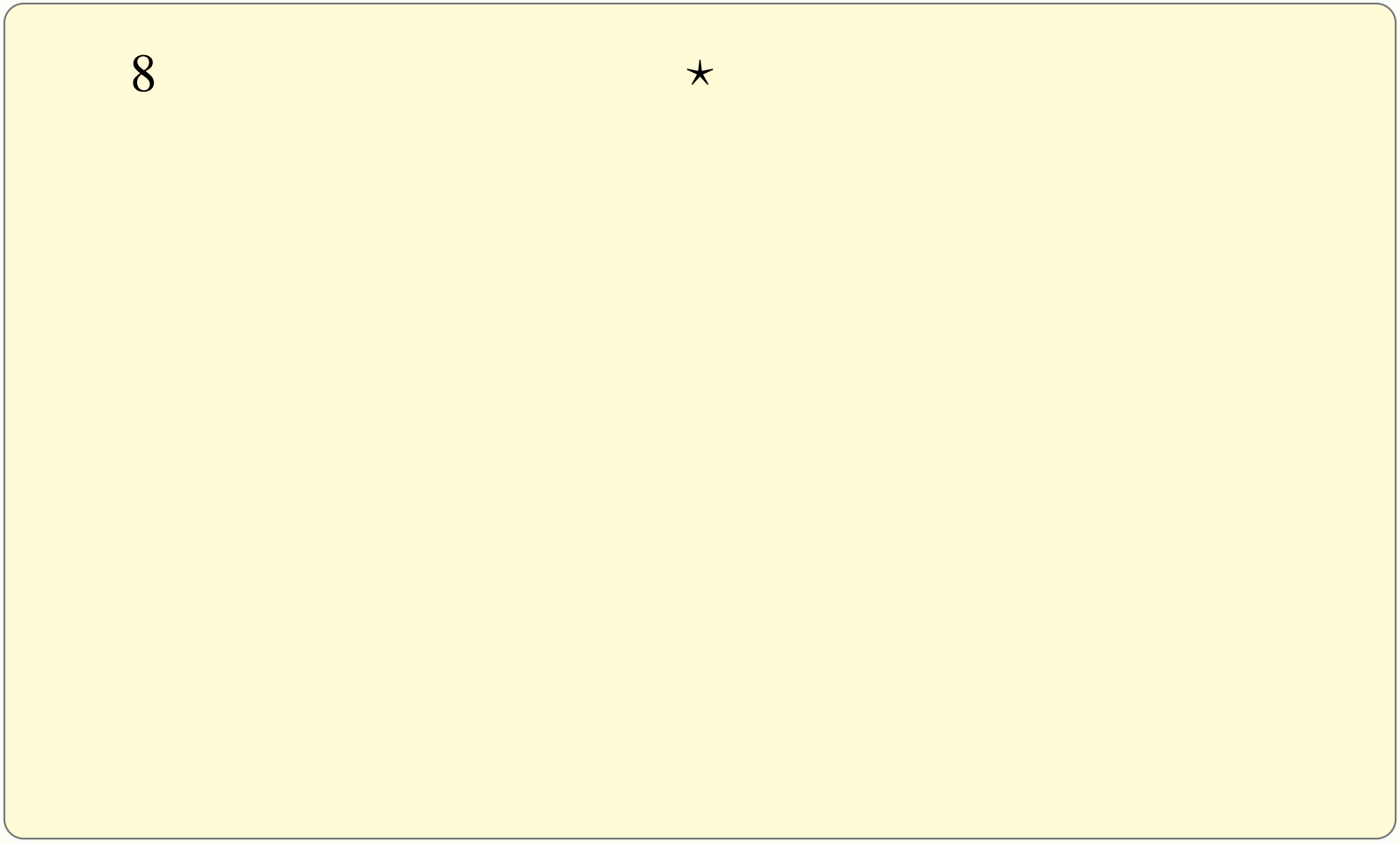
Pour la distribution "ABR", le nombre moyen de transitions est en $\Theta(n^2)$.

Distributions ”ABR”

Une feuille a probabilité p_ε d’être le mot vide. Un nœud interne, probabilité $p_\star$ d’être une étoile.

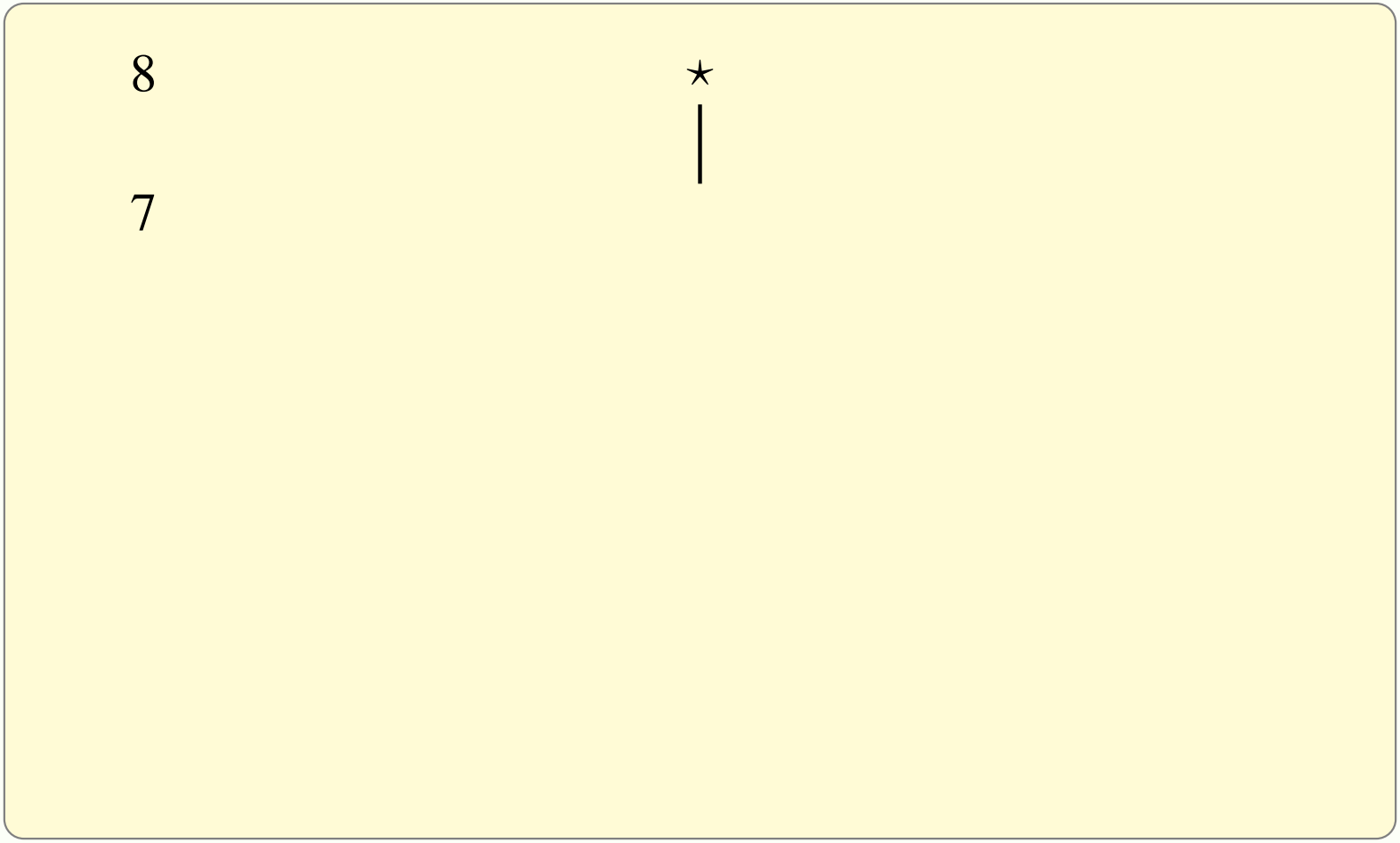
Distributions ”ABR”

Une feuille a probabilité p_ε d’être le mot vide. Un nœud interne, probabilité $p_\star$ d’être une étoile.



Distributions ”ABR”

Une feuille a probabilité p_ε d'être le mot vide. Un nœud interne, probabilité $p_\star$ d'être une étoile.

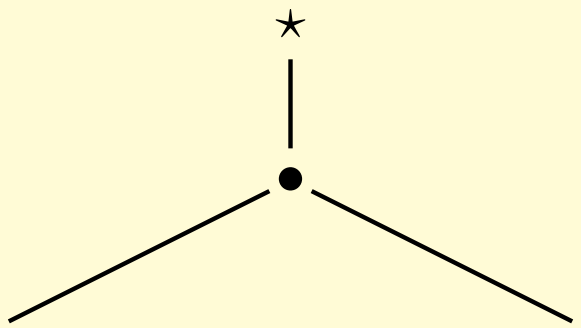


Distributions ”ABR”

Une feuille a probabilité p_ε d’être le mot vide. Un nœud interne, probabilité $p_\star$ d’être une étoile.

8

7



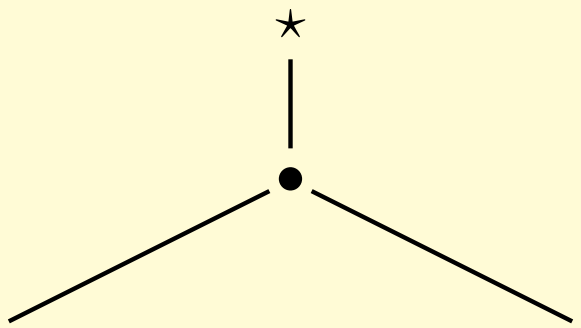
Distributions ”ABR”

Une feuille a probabilité p_ε d’être le mot vide. Un nœud interne, probabilité $p_\star$ d’être une étoile.

8

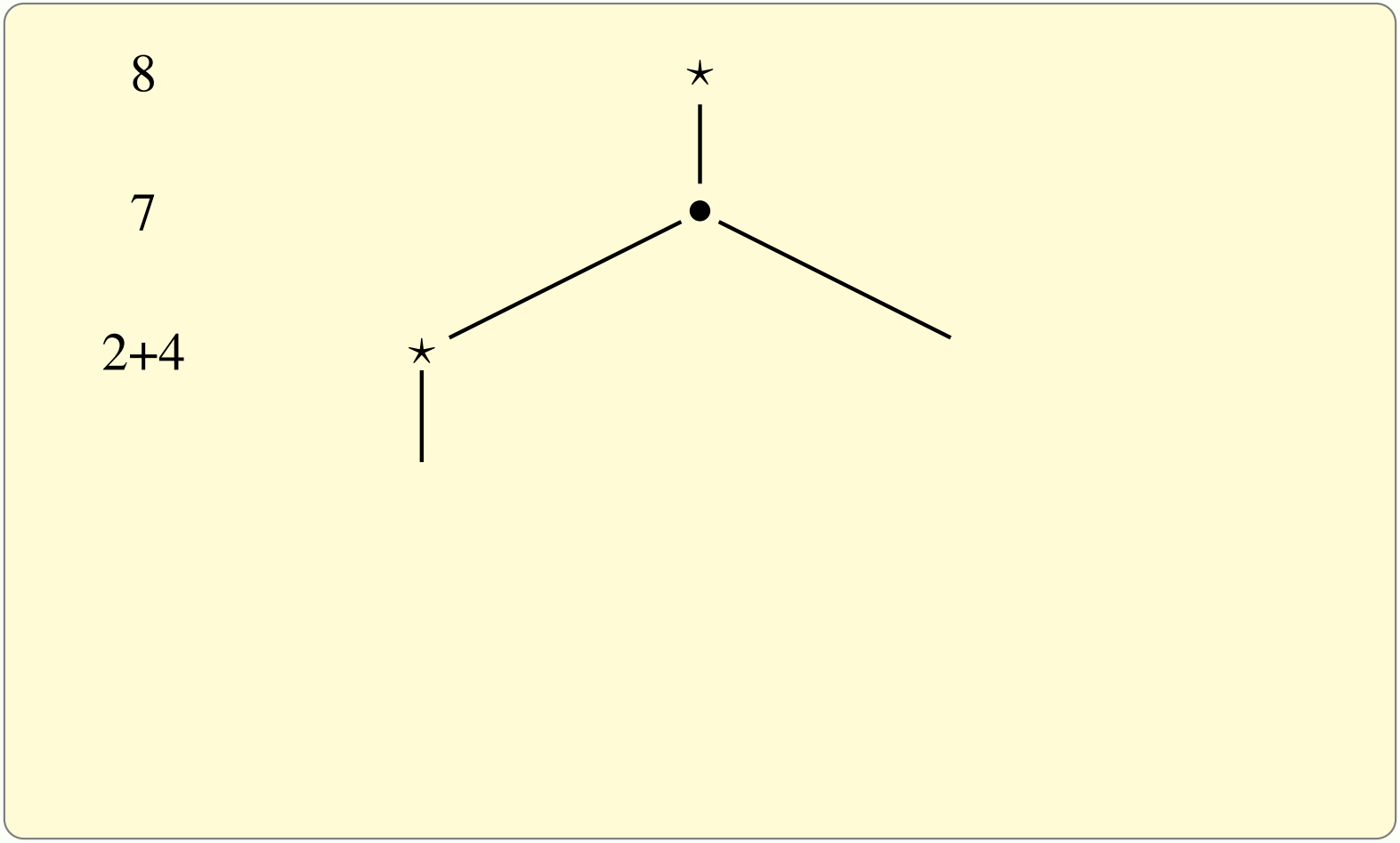
7

2+4



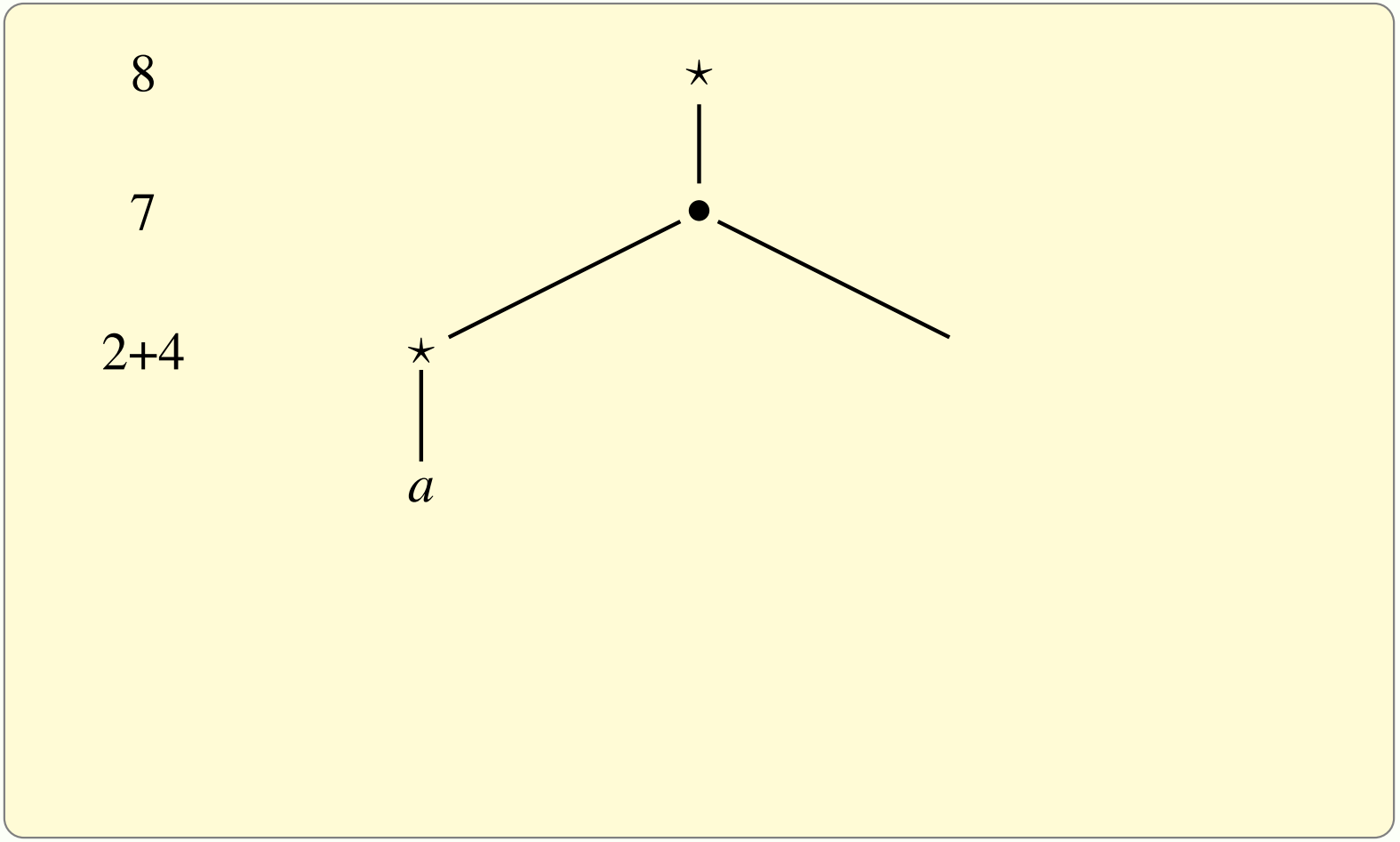
Distributions ”ABR”

Une feuille a probabilité p_ε d’être le mot vide. Un nœud interne, probabilité $p_\star$ d’être une étoile.



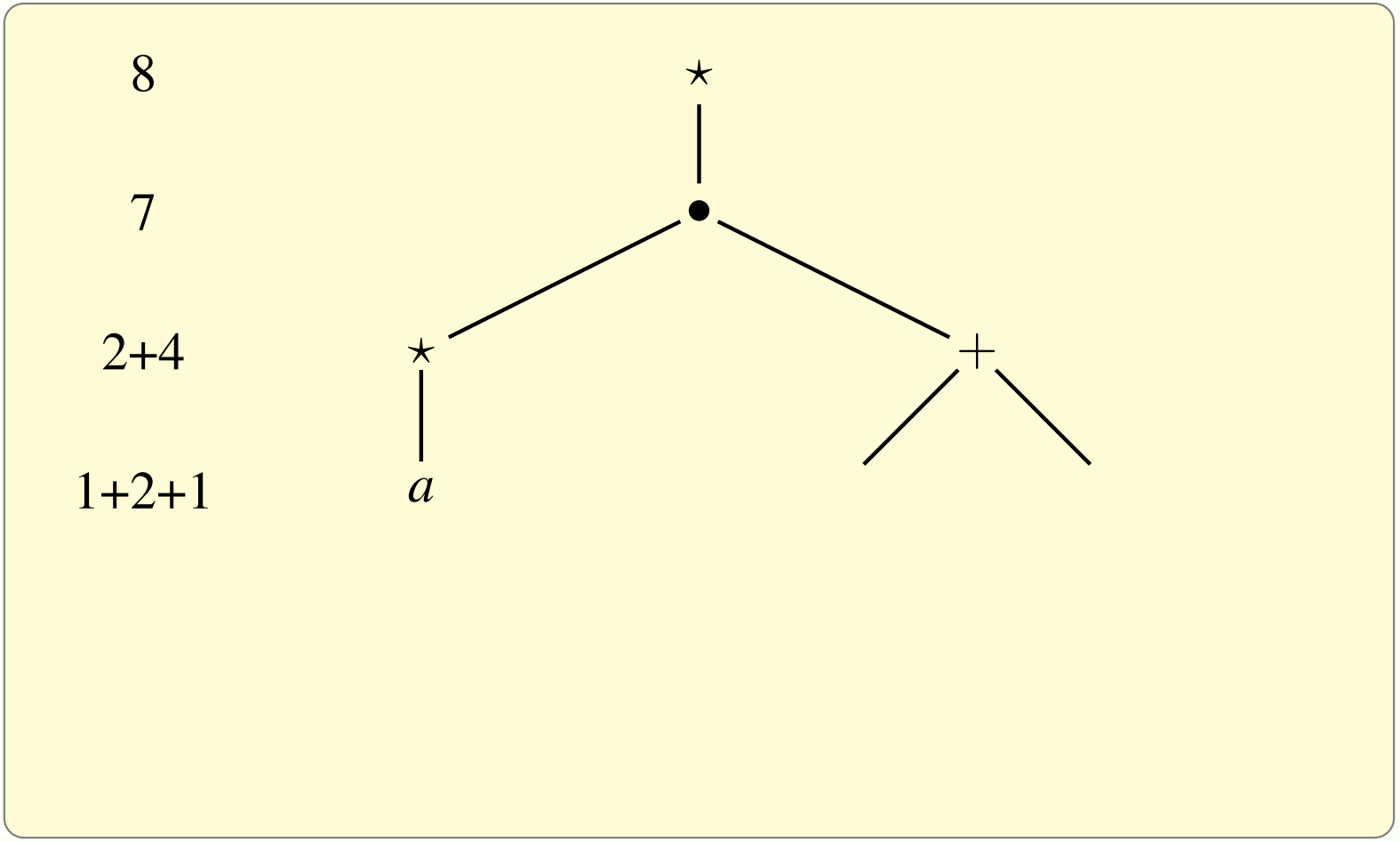
Distributions ”ABR”

Une feuille a probabilité p_ε d’être le mot vide. Un nœud interne, probabilité $p_\star$ d’être une étoile.



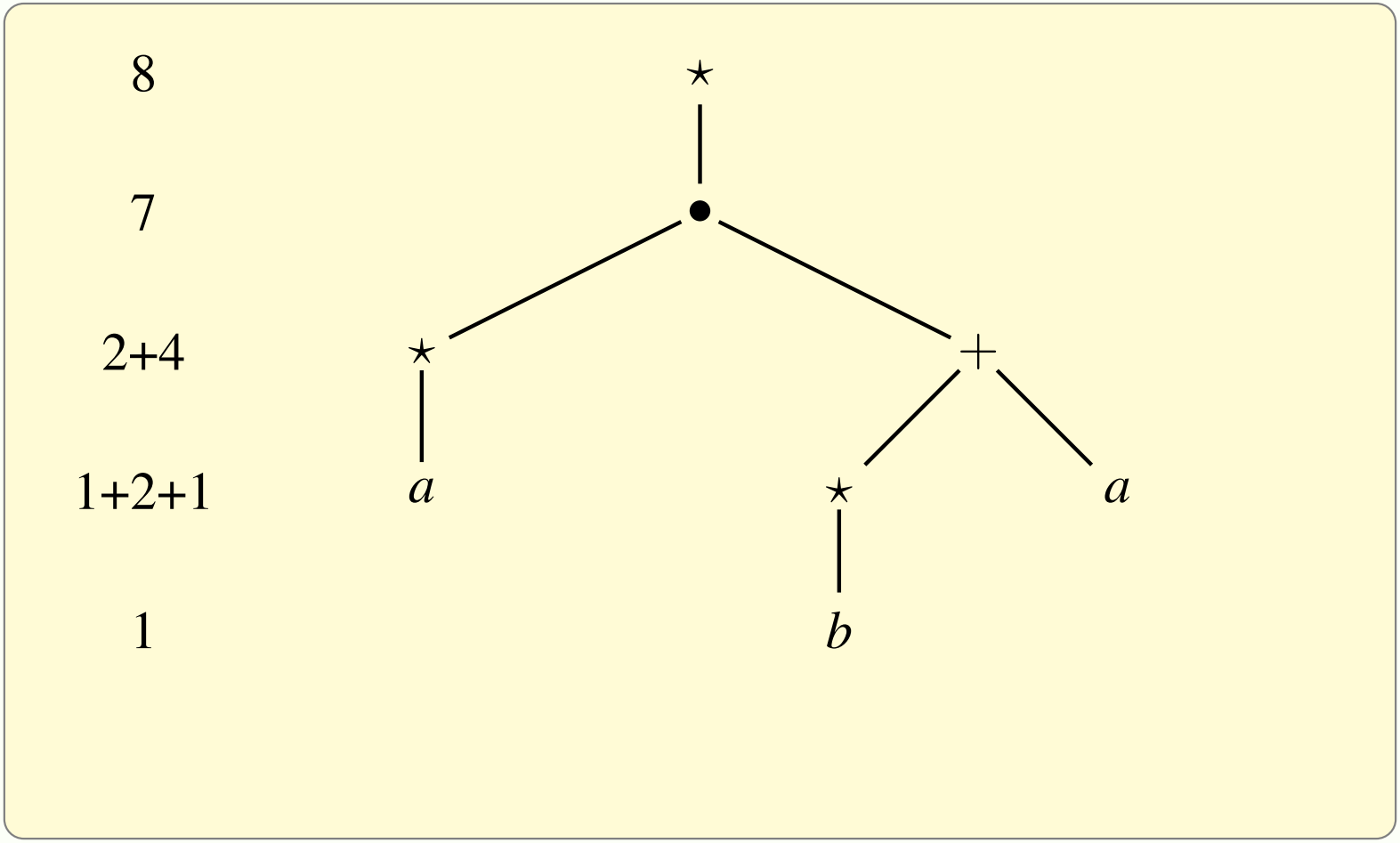
Distributions ”ABR”

Une feuille a probabilité p_ε d'être le mot vide. Un nœud interne, probabilité $p_\star$ d'être une étoile.



Distributions ”ABR”

Une feuille a probabilité p_ε d'être le mot vide. Un nœud interne, probabilité $p_\star$ d'être une étoile.



Uniforme
(Nicaud 09)

ABR
(Nicaud, Pivoteau, Razet 10)

Extension : Etude du nombre moyen d'états d'un automate d'Antimirov [Broda et al. 2010]

Uniforme
(Nicaud 09)

Hauteur $\Theta(\sqrt{n})$

ABR
(Nicaud, Pivoteau, Razet 10)

Hauteur $\Theta(\log n)$

Extension : Etude du nombre moyen d'états d'un automate d'Antimirov [Broda et al. 2010]

Uniforme (Nicaud 09)	ABR (Nicaud, Pivoteau, Razet 10)
Hauteur $\Theta(\sqrt{n})$	Hauteur $\Theta(\log n)$
Proba$(\varepsilon \notin L) = \Theta(1)$ Taille de First $= \Theta(1)$ Nbr transitions $= \Theta(n)$	Proba$(\varepsilon \notin L) = o(1)$ Taille de First $= \Theta(n)$ Nbr transitions $= \Theta(n^2)$

Extension : Etude du nombre moyen d'états d'un automate d'Antimirov [Broda et al. 2010]

En guise de conclusion

Quelques problèmes

- Automate minimaux : que peut-on dire de la taille moyenne de l'automate minimal de l'union de deux langages ?

En guise de conclusion

Quelques problèmes

- Automate minimaux : que peut-on dire de la taille moyenne de l'automate minimal de l'union de deux langages ?
- A quoi ressemble un langage aléatoire dans les différents modèles ?
- Quelles propriétés algébriques peut-on capter de manière combinatoire ?

En guise de conclusion

Quelques problèmes

- Automate minimaux : que peut-on dire de la taille moyenne de l'automate minimal de l'union de deux langages ?
- A quoi ressemble un langage aléatoire dans les différents modèles ?
- Quelles propriétés algébriques peut-on capter de manière combinatoire ?
- D'autres machines à états finis pourraient être étudiées d'un point de vue combinatoire.

Merci de votre attention !