Sur la complexité moyenne de l'algorithme de Moore [1]

Frédérique Bassino **Julien David** Cyril Nicaud

ALEA 2009

[1] STACS'09

Automate déterministe complet

Un **automate déterministe complet** $\mathcal{A}$ est

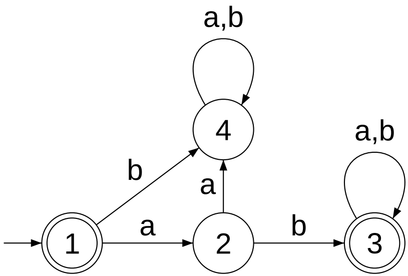
- un graphe fini orienté
- dont les transitions (ou arêtes) sont étiquetées sur un alphabet fini
- avec un ensemble F d'états (ou sommets) terminaux
- un unique état initial.
- pour tout état p et pour toute lettre a de l'alphabet, il existe exactement une transition sortant de p étiquettée par a.

Un automate est **accessible** si

- pour tout état de l'automate, il existe un chemin partant de l'état initial et passant par cet état.

Le **langage reconnu** par un automate est l'ensemble des étiquettes des chemins allant d'un état initial à un état terminal. Les **langages rationnels** sont les langages reconnus par un automate fini.

Automate



- Alphabet de l'automate : $A = \{a, b\}$
- État initial : 1
- Ensemble des états terminaux : $F = \{1, 3\}$

Automate minimal

- Pour tout langage rationnel, il n'y a pas d'unicité de l'automate reconnaissant ce langage.
- Pour tout langage rationnel, il existe un unique automate déterministe accessible complet reconnaissant ce langage, tel que le nombre d'états soit minimal.
  On le nomme **automate minimal**
- La plupart des algorithmes de minimisation d'automates calculent la relation d'équivalence de Myhill-Nerode entre les états afin de fusionner les états équivalents.

Automate minimal

- Pour tout langage rationnel, il n'y a pas d'unicité de l'automate reconnaissant ce langage.
- Pour tout langage rationnel, il existe un unique automate déterministe accessible complet reconnaissant ce langage, tel que le nombre d'états soit minimal.
  On le nomme **automate minimal**
- La plupart des algorithmes de minimisation d'automates calculent la relation d'équivalence de Myhill-Nerode entre les états afin de fusionner les états équivalents.

Automate minimal

- Pour tout langage rationnel, il n'y a pas d'unicité de l'automate reconnaissant ce langage.
- Pour tout langage rationnel, il existe un unique automate déterministe accessible complet reconnaissant ce langage, tel que le nombre d'états soit minimal.
  On le nomme **automate minimal**
- La plupart des algorithmes de minimisation d'automates calculent la relation d'équivalence de Myhill-Nerode entre les états afin de fusionner les états équivalents.

Complexité dans le pire cas

- Moore (1956) : $\mathcal{O}(n^2)$
- Hopcroft (1971) : $\mathcal{O}(n \log n)$

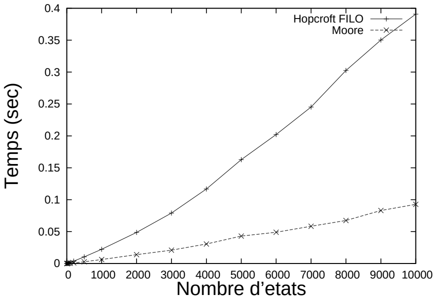
Résultats principaux

- La complexité moyenne de l'algorithme de Moore est $O(n \log n)$
- Cette majoration est optimale pour le cas des automates unaires.

REGAL[Bassino, D, Nicaud 2007] [2]

Random and **E**xhaustive **G**enerators for **A**utomata **L**ibrary

Bibliothèque en C++, permettant d'engendrer aléatoirement et exhaustivement des automates déterministes accessibles.



[2] http ://regal.univ-mlv.fr

Partition de Myhill-Nerode

$p \sim q \iff$ les mots reconnus en prenant p et q comme états initiaux sont les mêmes.

$\sim_i$ est une partition de l'ensemble des états :

$p \sim_i q \iff$ les mots de longueur $\leq i$ reconnus en prenant p et q comme états initiaux sont les mêmes.

Partition de Myhill-Nerode

$p \sim q \iff$ les mots reconnus en prenant p et q comme états initiaux sont les mêmes.

$\sim_i$ est une partition de l'ensemble des états :

$p \sim_i q \iff$ les mots de longueur $\leq i$ reconnus en prenant p et q comme états initiaux sont les mêmes.

Algorithme de Moore

Basé sur le calcul de la **partition de Myhill-Nerode** :

- $\sim_0$: partition en deux sous-ensembles d'états terminaux et d'états non terminaux.
- On calcule la partition $\sim_i$ en raffinant $\sim_{i-1}$
- On répète l'opération jusqu'à ce que $\sim_i = \sim_{i-1}$

Algorithme de Moore

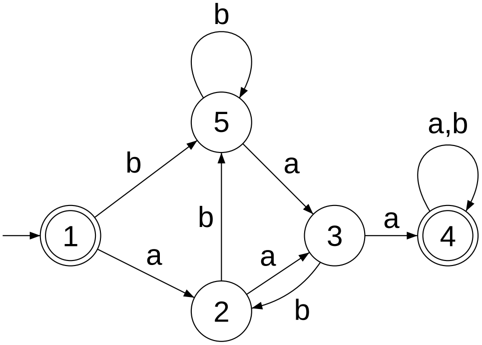


FIG.: Automate à minimiser

	$\sim_0$
1	1
2	0
3	0
4	1
5	0

$$\{1,4\}\{2,3,5\}$$

FIG.: Algorithme de Moore

Algorithme de Moore

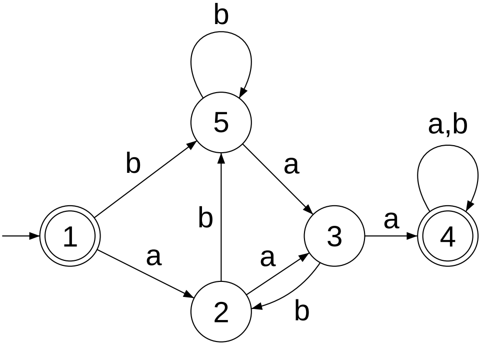


FIG.: Automate à minimiser

	$\sim_0$	a	b
1	1	0	0
2	0	0	0
3	0	1	0
4	1	1	1
5	0	0	0

$$\{1,4\}\{2,3,5\}$$

FIG.: Algorithme de Moore

Algorithme de Moore

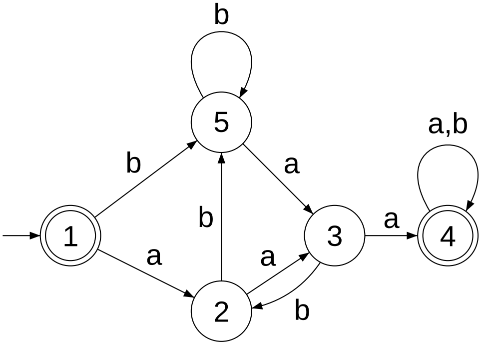


FIG.: Automate à minimiser

	$\sim_0$	a	b	$\sim_1$
1	1	0	0	0
2	0	0	0	1
3	0	1	0	2
4	1	1	1	3
5	0	0	0	1

$$\{1\}\{2,5\}\{3\}\{4\}$$

FIG.: Algorithme de Moore

Algorithme de Moore

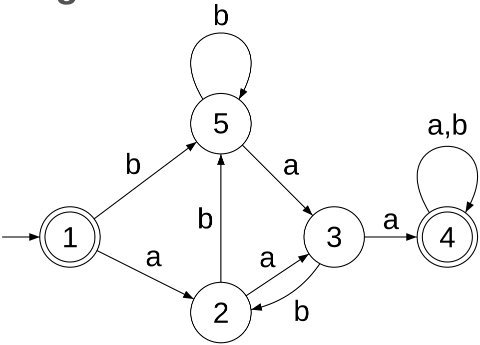


FIG.: Automate à minimiser

	$\sim_0$	a	b	$\sim_1$	a	b	$\sim_2$
1	1	0	0	0	1	1	0
2	0	0	0	1	2	1	1
3	0	1	0	2	3	1	2
4	1	1	1	3	3	3	3
5	0	0	0	1	2	1	1

$$\{1\}\{2,5\}\{3\}\{4\}$$

FIG.: Algorithme de Moore

Algorithme de Moore

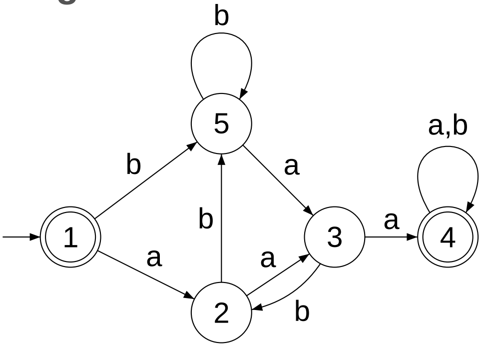


FIG.: Automate à minimiser

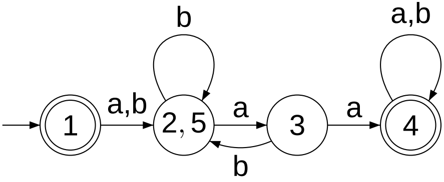


FIG.: Automate Minimal

Remarque sur la complexité

La complexité de l'algorithme dépend :

- Du nombre de raffinements de partition effectués par l'algorithme
- Du coût d'un raffinement de partition
  $\longrightarrow O(n)$

La complexité moyenne dépend du nombre moyen de raffinements de partition.

La borne supérieure

Théorème

Pour la distribution uniforme sur l'ensemble des automates déterministes accessibles complets de taille n, la complexité moyenne de l'algorithme de Moore est $\mathcal{O}(n \log n)$.

Idée rapide de la preuve

Le nombre d'automates minimisés en au moins $5 \log n$ raffinements de partition est négligeable.

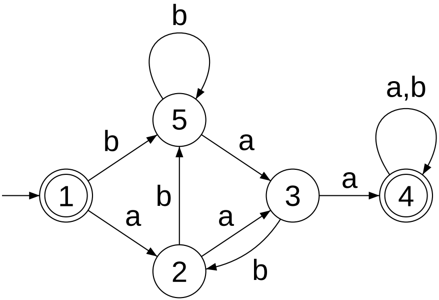
La borne supérieure

Théorème

Pour la distribution uniforme sur l'ensemble des automates déterministes accessibles complets de taille n, la complexité moyenne de l'algorithme de Moore est $\mathcal{O}(n \log n)$.

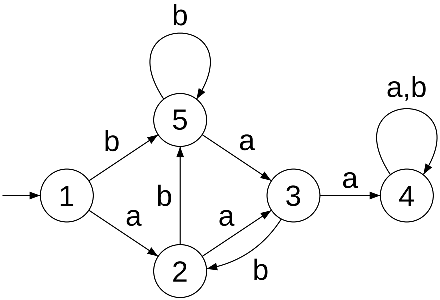
Idée rapide de la preuve

Le nombre d'automates minimisés en au moins $5 \log n$ raffinements de partition est négligeable.



Automate

$\mathcal{A} =< A, Q, \cdot, q_0, F >$



Structure de transitions

$\mathcal{T} =< A, Q, \cdot, q_0 >$

Soient $\mathcal{A}_n$ l'ensemble des automates et $\mathcal{T}_n$ l'ensemble des structures de transitions de taille n.

$$|\mathcal{A}_n| = |\mathcal{T}_n| \times 2^n$$

Plan de la preuve

Pour une structure de transitions fixée de taille $n \geq 1$, on prouve que le nombre moyen de raffinements de partition est $\mathcal{O}(\log n)$:

1. On caractérise les ensembles d'états terminaux tels que l'algorithme, appliqué à $(\mathcal{T}, F)$, requiert au moins ℓ itérations.
2. On obtient une majoration du nombre de tels ensembles
3. On prouve que ce nombre est négligeable pour $\ell \geq 5 \log n$.

Caractérisation des automates minimisés en au moins ℓ itérations

Il existe deux états p et q, qui sont séparés durant le ℓ-ème raffinement de partition :

$$\iff p \sim_{\ell-1} q : \forall w \in A^{<\ell}, p \cdot w \sim_{\ell-|w|-1} q \cdot w$$
$$(p \cdot w \in F \iff q \cdot w \in F)$$

$$p \nsim_{\ell} q : u \in A^{\ell}, p \cdot u = p', q \cdot u = q'$$
$$(p' \in F \iff q' \notin F)$$

Caractérisation des automates minimisés en au moins ℓ itérations

Pour une structure de transitions $\mathcal{T}$ fixée, ℓ, p, q, p', q' fixés,

on définit l'ensemble $\mathcal{F}_\ell \subset 2^Q$, tel que $\forall F \in \mathcal{F}_\ell$, $(\mathcal{T}, F)$ est minimisé en au moins ℓ raffinements de partition.

En fonction des paramètres, cet ensemble peut être vide.

S'il existe plusieurs mots de longueur ℓ étiquetant des chemins de p, q vers p', q', on notera u le plus petit mot dans l'ordre lexicographique.

Caractérisation des automates minimisés en au moins ℓ itérations

Pour une structure de transitions $\mathcal{T}$ fixée, ℓ, p, q, p', q' fixés,

on définit l'ensemble $\mathcal{F}_\ell \subset 2^Q$, tel que $\forall F \in \mathcal{F}_\ell$,
$(\mathcal{T}, F)$ est minimisé en au moins ℓ raffinements de partition.

En fonction des paramètres, cet ensemble peut être vide.

S'il existe plusieurs mots de longueur ℓ étiquetant des chemins de p, q vers p', q', on notera u le plus petit mot dans l'ordre lexicographique.

Caractérisation des automates minimisés en au moins ℓ itérations

L'ensemble des ensembles F d'états terminaux, tels que $(\mathcal{T}, F)$ est minimisé en au moins ℓ itérations, est égal à :

$$\bigcup_{\substack{p,q,p',q' \in \{1,\cdots,n\} \\ p \neq q,\ p' \neq q'}} \mathcal{F}_\ell$$

Un automate $(\mathcal{T}, F)$ peut être dans plusieurs ensembles $\mathcal{F}_\ell$

$$\left| \bigcup \mathcal{F}_\ell \right| \leq \sum |\mathcal{F}_\ell|$$

Le graphe de dépendance

On définit le **graphe de dépendance** $\mathcal{G}$, qui **modélise les contraintes** sur un ensemble $\mathcal{F}_\ell$.

- L'ensemble de ses sommets est l'ensemble des états de $\mathcal{T}$,
- Pour tout $w \in A^*$, il existe une arête $(p \cdot w, q \cdot w)$ ssi pour tout $F \in \mathcal{F}_\ell$:

$$p \cdot w \in F \iff q \cdot w \in F$$

Le graphe de dépendance implique un ensemble de contraintes qui permet d'obtenir une majoration sur l'ensemble $\mathcal{F}_\ell$ associé.
On ne considère qu'un sous-ensemble de ces contraintes.

Lemme
Il existe un sous-graphe acyclique du graphe de dépendance $\mathcal{G}$ qui contient exactement ℓ arêtes.

$\Longrightarrow$ **Il existe au moins ℓ contraintes distinctes sur l'ensemble $\mathcal{F}_\ell$**

Le graphe de dépendance implique un ensemble de contraintes qui permet d'obtenir une majoration sur l'ensemble $\mathcal{F}_\ell$ associé.
On ne considère qu'un sous-ensemble de ces contraintes.

Lemme

Il existe un sous-graphe acyclique du graphe de dépendance $\mathcal{G}$ qui contient exactement ℓ arêtes.

$\Longrightarrow$ **Il existe au moins ℓ contraintes distinctes sur l'ensemble $\mathcal{F}_\ell$**

Le sous-graphe acyclique $G_{\ell-1}$

Il existe $|u| = \ell$ tel que $p \cdot u = p'$ et $q \cdot u = q'$
$\Longrightarrow$ On utilise les contraintes données par les préfixes stricts de u.

Pour tout $i \in \{0, \cdots, \ell - 1\}$, on définit G_i comme étant un sous-graphe de $\mathcal{G}$ tel que :

- Une arête $(p \cdot v, q \cdot v)$ est dans G_i ssi v est un préfixe u de longueur $\leq i$.
- G_{i+1} est obtenu en ajoutant une arête $(p \cdot w, q \cdot w)$ à G_i, où w est le préfixe de u longueur $i + 1$.

Le sous-graphe acyclique $G_{\ell-1}$

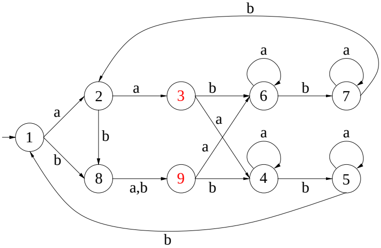
Il existe $|u| = \ell$ tel que $p \cdot u = p'$ et $q \cdot u = q'$
$\Longrightarrow$ On utilise les contraintes données par les préfixes stricts de u.

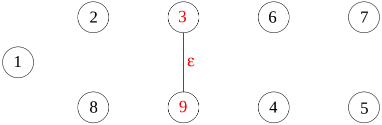
Pour tout $i \in \{0, \cdots, \ell - 1\}$, on définit G_i comme étant un sous-graphe de $\mathcal{G}$ tel que :

- Une arête $(p \cdot v, q \cdot v)$ est dans G_i ssi v est un préfixe u de longueur $\leq i$.
- G_{i+1} est obtenu en ajoutant une arête $(p \cdot w, q \cdot w)$ à G_i, où w est le préfixe de u longueur $i + 1$.

Le sous-graphe $\mathcal{G}_0$

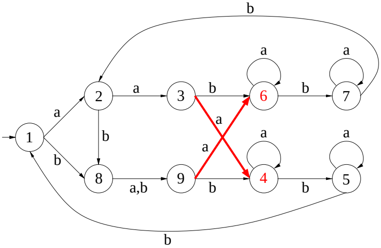
$\ell = 5$	$p = 3$	$q = 9$	$p' = 3$	$q' = 4$	$u = abbaa$

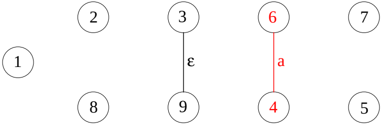




Le sous-graphe $\mathcal{G}_1$

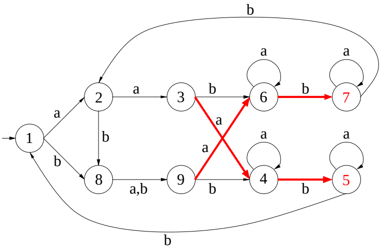
$\ell = 5$	$p = 3$	$q = 9$	$p' = 3$	$q' = 4$	$u = abbaa$

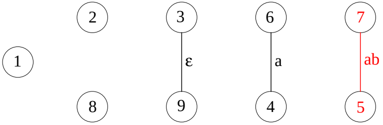




Le sous-graphe $\mathcal{G}_2$

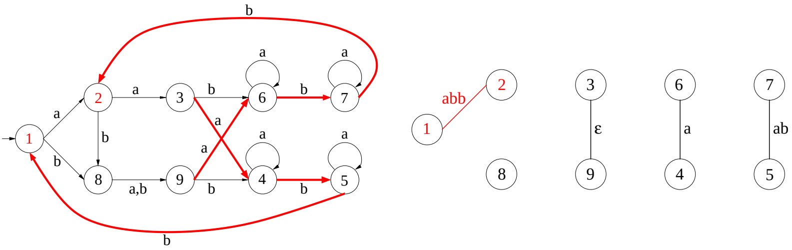
$\ell = 5$	$p = 3$	$q = 9$	$p' = 3$	$q' = 4$	$u = abbaa$





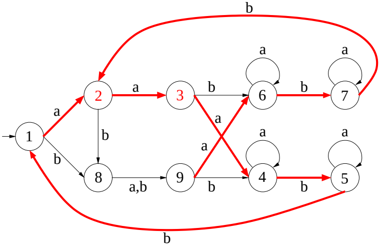
Le sous-graphe $\mathcal{G}_3$

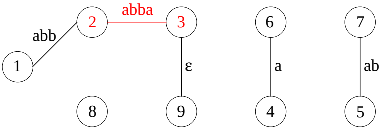
$\ell = 5$	$p = 3$	$q = 9$	$p' = 3$	$q' = 4$	$u = abbaa$



Le sous-graphe $\mathcal{G}_4$

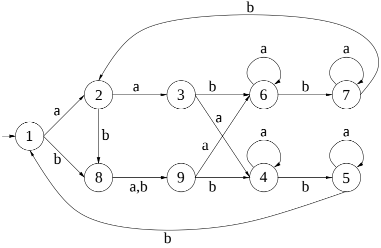
$\ell = 5$	$p = 3$	$q = 9$	$p' = 3$	$q' = 4$	$u = abbaa$

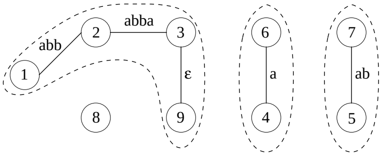




Le sous-graphe $\mathcal{G}_4$

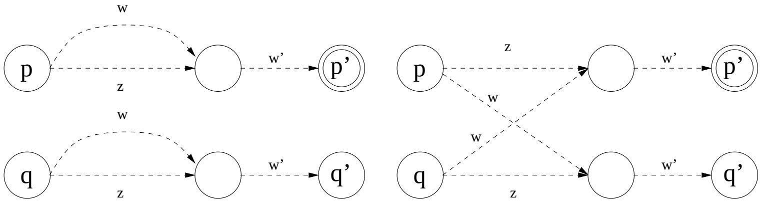
$\ell = 5$	$p = 3$	$q = 9$	$p' = 3$	$q' = 4$	$u = abbaa$





Le sous-graphe $G_{\ell-1}$ contient exactement ℓ arêtes

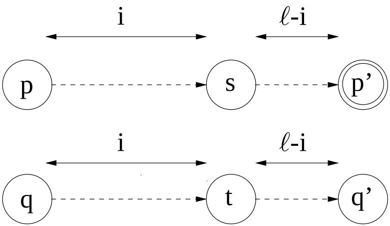
- G_{i+1} est obtenu en ajoutant une arête $(p \cdot w, q \cdot w)$ à G_i. Cette arête n'appartient pas à G_i

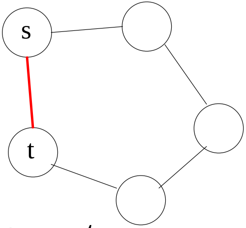


z est un préfixe strict de w
Soit $u = ww'$, $|zw'| < \ell$
Contradiction

Le sous-graphe $G_{\ell-1}$ est acyclique

- G_i est acyclique





$s \sim_{\ell-i} t$

Majoration du nombre d'ensembles d'états terminaux

Lemme

Étant donné une structure de transition $\mathcal{T}$ et un entier ℓ, avec $1 \leq \ell < n$:

$$\sum |\mathcal{F}_\ell| = \mathcal{O}(n^4 \times 2^{n-\ell})$$

- n^4 vient du choix de p, q, p', q'
- $2^{n-\ell}$ est la majoration du cardinal d'un ensemble $\mathcal{F}_\ell$, donné par le graphe de dépendance.

Corollaire

Pour une structure de transitions fixée, le nombre d'automates minimisés en au moins $5 \log n$ itérations est négligeable

Théorème

Pour la distribution uniforme sur l'ensemble des automates déterministes accessibles complets, la complexité moyenne de l'algorithme de Moore est de $\mathcal{O}(n \log n)$

Corollaire

Pour une structure de transitions fixée, le nombre d'automates minimisés en au moins $5 \log n$ itérations est négligeable

Théorème

Pour la distribution uniforme sur l'ensemble des automates déterministes accessibles complets, la complexité moyenne de l'algorithme de Moore est de $\mathcal{O}(n \log n)$

Automates unaires

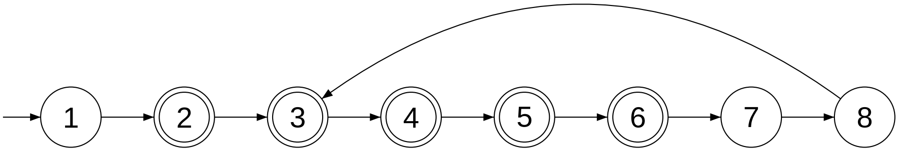
Théorème

Pour la distribution uniforme sur l'ensemble des automates unaires de taille n, la complexité moyenne de l'algorithme de Moore est $\Theta(n \log n)$

Conséquence

Quelle que soit la taille de l'alphabet, il existe des structures de transitions dont les automates associés sont minimisés en $\Theta(n \log n)$ en moyenne, pour la distribution uniforme sur l'ensemble des ensembles d'états terminaux.

Caractérisation



L'ensemble des états terminaux est codé par 01111100.

Si l'automate contient ℓ états terminaux consécutifs, les deux premiers états de cette suites seront séparé lors de la ℓ-ème itération.

Proposition [Knuth 78]

Pour la distribution uniforme sur les mots binaires de longueur n, la probabilité que la plus longue suite de 1 soit plus grande que $\lfloor \frac{1}{2} \log_2 n \rfloor$ tend vers 1.

Extension

Durant la présentation, la probabilité pour un état d'être terminal était de $\frac{1}{2}$. Les résultats énoncés sont toujours valables pour une probabilité $p \in]0, 1[$ fixée.

Conclusion

Problème ouvert : une meilleure majoration

Pour un alphabet de taille > 1, on conjecture que la complexité moyenne est $\Theta(n \log \log n)$.