

# Génération aléatoire d'arbres de Motzkin "à la Rémy"

Axel Bacher, Olivier Bodini, Alice Jacquot

LIPN, Université Paris 13

22 mars 2013

# Sommaire

- 1 Introduction : Arbres de Catalan
  - Algorithme de Rémy
- 2 Spécification holonome pour les arbres de Motzkin
  - Définition
  - Spécification
- 3 Génération aléatoire
  - Méthode
  - Méthode récursive
  - Algorithme "à la Rémy"
- 4 Perspectives
  - Perspectives

# Algorithme de Rémy

**Rémy**( $n$ ) :

$T :=$  une feuille.

Pour  $i = 0$  à  $n$  faire :

    Choisir uniformément un nœud

    Avec probabilité  $1/2$  :

        Étendre à droite

    Sinon :

        Étendre à gauche



[Rémy 85]

# Algorithme de Rémy

Droite

**Rémy**( $n$ ) :

$T$  := une feuille.

Pour  $i = 0$  à  $n$  faire :

    Choisir uniformément un nœud

    Avec probabilité  $1/2$  :

        Étendre à droite

    Sinon :

        Étendre à gauche



[Rémy 85]

# Algorithme de Rémy

Droite

**Rémy**( $n$ ) :

$T$  := une feuille.

Pour  $i = 0$  à  $n$  faire :

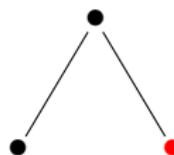
Choisir uniformément un nœud

Avec probabilité  $1/2$  :

Étendre à droite

Sinon :

Étendre à gauche



[Rémy 85]

# Algorithme de Rémy

**Rémy**( $n$ ) :

$T :=$  une feuille.

Pour  $i = 0$  à  $n$  faire :

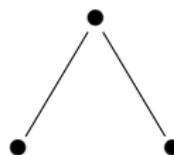
Choisir uniformément un nœud

Avec probabilité  $1/2$  :

Étendre à droite

Sinon :

Étendre à gauche



[Rémy 85]

# Algorithme de Rémy

Gauche

**Rémy**( $n$ ) :

$T$  := une feuille.

Pour  $i = 0$  à  $n$  faire :

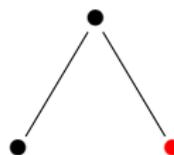
Choisir uniformément un nœud

Avec probabilité  $1/2$  :

Étendre à droite

Sinon :

Étendre à gauche



[Rémy 85]

# Algorithme de Rémy

Gauche

**Rémy**( $n$ ) :

$T :=$  une feuille.

Pour  $i = 0$  à  $n$  faire :

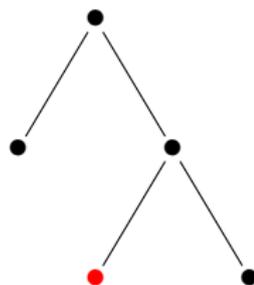
Choisir uniformément un nœud

Avec probabilité  $1/2$  :

Étendre à droite

Sinon :

Étendre à gauche



[Rémy 85]

# Algorithme de Rémy

**Rémy**( $n$ ) :

$T :=$  une feuille.

Pour  $i = 0$  à  $n$  faire :

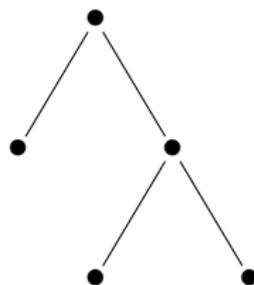
Choisir uniformément un nœud

Avec probabilité  $1/2$  :

Étendre à droite

Sinon :

Étendre à gauche



[Rémy 85]

# Algorithme de Rémy

Gauche

**Rémy**( $n$ ) :

$T$  := une feuille.

Pour  $i = 0$  à  $n$  faire :

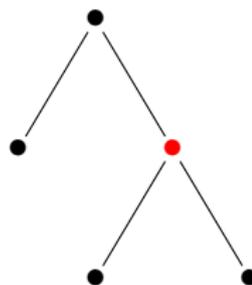
Choisir uniformément un nœud

Avec probabilité  $1/2$  :

Étendre à droite

Sinon :

Étendre à gauche



[Rémy 85]

# Algorithme de Rémy

Gauche

**Rémy**( $n$ ) :

$T :=$  une feuille.

Pour  $i = 0$  à  $n$  faire :

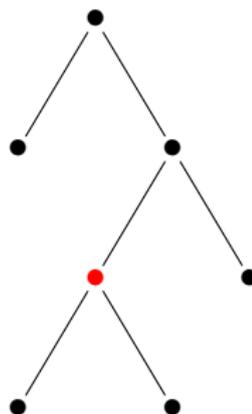
Choisir uniformément un nœud

Avec probabilité  $1/2$  :

Étendre à droite

Sinon :

Étendre à gauche



[Rémy 85]

# Algorithme de Rémy

**Rémy**( $n$ ) :

$T :=$  une feuille.

Pour  $i = 0$  à  $n$  faire :

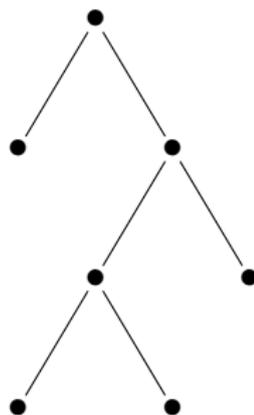
Choisir uniformément un nœud

Avec probabilité  $1/2$  :

Étendre à droite

Sinon :

Étendre à gauche



[Rémy 85]



# Algorithme de Rémy

**Rémy**( $n$ ) :

$T$  := une feuille.

Pour  $i = 0$  à  $n$  faire :

Choisir uniformément un nœud

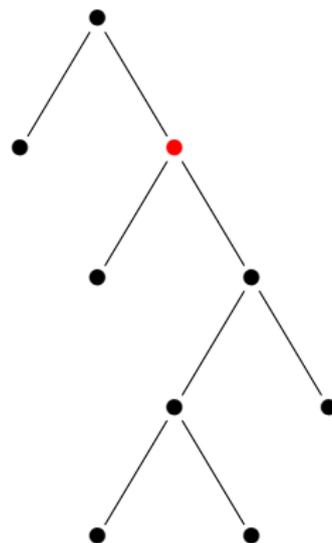
Avec probabilité  $1/2$  :

Étendre à droite

Sinon :

Étendre à gauche

Droite



[Rémy 85]

# Algorithme de Rémy

**Rémy**( $n$ ) :

$T :=$  une feuille.

Pour  $i = 0$  à  $n$  faire :

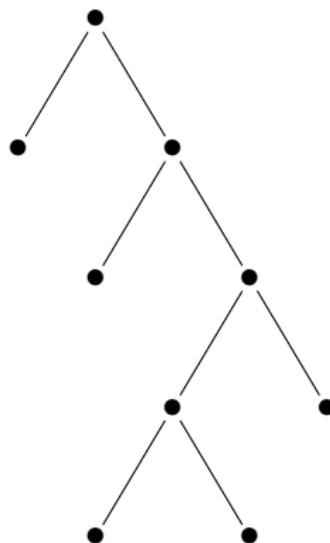
Choisir uniformément un nœud

Avec probabilité  $1/2$  :

Étendre à droite

Sinon :

Étendre à gauche



[Rémy 85]

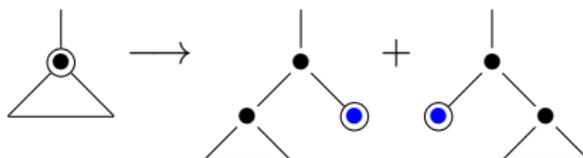
## Théorème

Rémy( $n$ ) génère un arbre de taille  $2n + 1$  uniformément parmi tous les arbres de Catalan de taille  $2n + 1$ .

## Théorème

Rémy( $n$ ) génère un arbre de taille  $2n + 1$  uniformément parmi tous les arbres de Catalan de taille  $2n + 1$ .

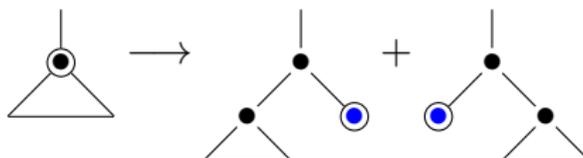
$$\mathcal{C}^{\bullet(\text{leaf})} = \mathcal{Z} + 2\mathcal{Z}^2\mathcal{C}^{\bullet}$$



## Théorème

Rémy( $n$ ) génère un arbre de taille  $2n + 1$  uniformément parmi tous les arbres de Catalan de taille  $2n + 1$ .

$$\mathcal{C}^{\bullet(\text{leaf})} = \mathcal{Z} + 2\mathcal{Z}^2\mathcal{C}^{\bullet}$$



Comme un arbre de taille  $2i + 1$  a toujours  $i + 1$  feuilles, on peut oublier le point et rester uniforme à chaque étape de l'algorithme de Remy.

# L'algorithme de Rémy pour les arbres de Motzkin

- 1 Trouver une spécification holonome pour les arbres de Motzkin
- 2 L'utiliser pour construire un générateur aléatoire
- 3 Transformer le générateur en générateur "à la Rémy"

Une équation holonome est une équation différentielle linéaire à coefficients polynomiaux.

# Sommaire

- 1 Introduction : Arbres de Catalan
  - Algorithme de Rémy
- 2 Spécification holonome pour les arbres de Motzkin
  - Définition
  - Spécification
- 3 Génération aléatoire
  - Méthode
  - Méthode récursive
  - Algorithme "à la Rémy"
- 4 Perspectives
  - Perspectives



# Spécification

Soit  $\mathcal{M}$  la classe des arbres de Motzkin.

## Theorem

$$\mathcal{M} + \mathcal{M}^\bullet = 2\mathcal{Z} + 3\mathcal{Z}^2\mathcal{M}^\bullet + \mathcal{Z}\mathcal{M}^\bullet + \mathcal{Z}(\mathcal{M} + \mathcal{M}^\bullet)$$

Ou :  $(n + 1)m_n = 3(i - 2)m_{i-2} + (i - 1)m_{i-1} + nm_{i-1}$ .

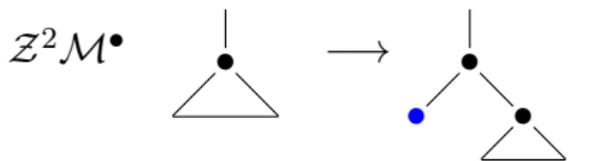
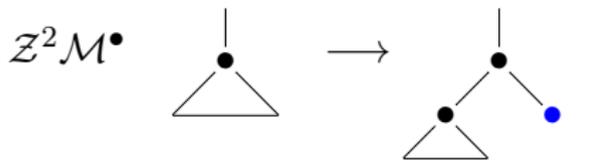
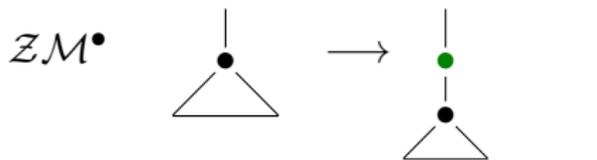
Or, on a :

$$2f + u = n + 1 \Rightarrow \mathcal{M} + \mathcal{M}^\bullet = 2\mathcal{M}^{\bullet(\text{feuille})} + \mathcal{M}^{\bullet(\text{unaire})}$$

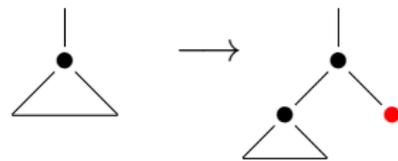
D'où

$$\mathcal{M} + \mathcal{M}^\bullet = \mathcal{M}^{\bullet(\text{feuille})} + \mathcal{M}^{\bullet(\text{feuille})} + \mathcal{M}^{\bullet(\text{unaire})}$$

$$M^{\bullet(\text{feuille})} + M^{\bullet(\text{feuille})} + M^{\bullet(\text{unaire})} = 2Z + 3Z^2M^{\bullet} + ZM^{\bullet} + Z(M^{\bullet(\text{feuille})} + M^{\bullet(\text{feuille})} + M^{\bullet(\text{unaire})})$$



$Z^2M^{\bullet}$



$ZM^{\bullet(\text{feuille})}$



$ZM^{\bullet(\text{feuille})}$



$ZM^{\bullet(\text{unaire})}$



# Sommaire

- 1 Introduction : Arbres de Catalan
  - Algorithme de Rémy
- 2 Spécification holonome pour les arbres de Motzkin
  - Définition
  - Spécification
- 3 Génération aléatoire
  - Méthode
  - Méthode récursive
  - Algorithme "à la Rémy"
- 4 Perspectives
  - Perspectives

# Méthode

- 1 Écrire un générateur aléatoire par la méthode récursive pour la spécification holonome :
  - Pré-calcul linéaire, stockage linéaire des coefficients,
  - Génération en temps linéaire,
  - Stockage de la pile de récursion.
- 2 Le modifier pour obtenir un générateur "à la Rémy" :
  - Pas de pré-calcul,
  - Génération en temps linéaire, stockage constant,
  - Récursif terminal.

# Méthode récursive

[Nijenhuis, Wilf 78]

[Flajolet, Zimmermann, Van Cutsem 94]

## Générateur( $\mathcal{A} + \mathcal{B}, n$ )

Avec Probabilité  $\frac{a_n}{a_n + b_n}$  :

**Renvoyer** Générateur( $\mathcal{A}, n$ )

Avec Probabilité  $\frac{b_n}{a_n + b_n}$  :

**Renvoyer** Générateur( $\mathcal{B}, n$ )

## Générateur( $\mathcal{C} = \mathcal{A} \times \mathcal{B}, n$ )

Tirer  $k$  tel que

$$\mathbb{P}(k = K) = \frac{a_K b_{n-K}}{c_n}$$

**Renvoyer** (Générateur( $\mathcal{A}, k$ ), Générateur( $\mathcal{B}, n - k$ ))

# Méthode récursive

[Nijenhuis, Wilf 78]

[Flajolet, Zimmermann, Van Cutsem 94]

## Générateur( $\mathcal{A} + \mathcal{B}, n$ )

Avec Probabilité  $\frac{a_n}{a_n + b_n}$  :

**Renvoyer** Générateur( $\mathcal{A}, n$ )

Avec Probabilité  $\frac{b_n}{a_n + b_n}$  :

**Renvoyer** Générateur( $\mathcal{B}, n$ )

## Générateur( $\mathcal{C} = \mathcal{A} \times \mathcal{B}, n$ )

Tirer  $k$  tel que

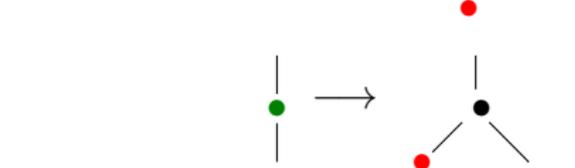
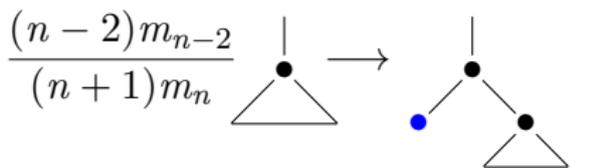
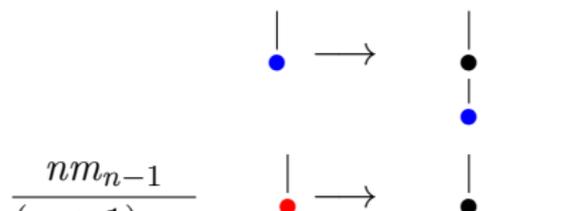
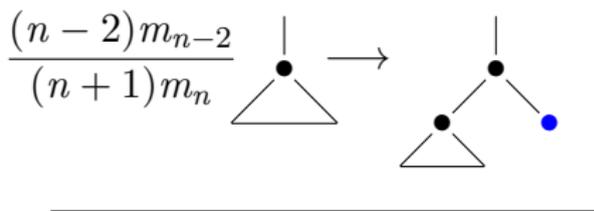
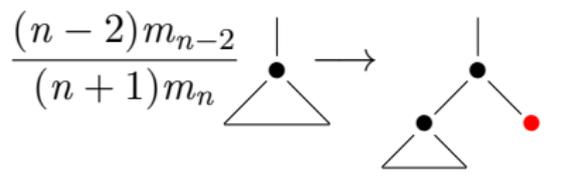
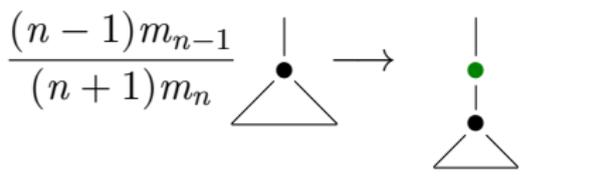
$$\mathbb{P}(k = K) = \frac{a_K b_{n-K}}{c_n}$$

**Renvoyer** (Générateur( $\mathcal{A}, k$ ), Générateur( $\mathcal{B}, n - k$ ))

- Pré-calculer et stocker tous les  $a_i$
- Complexité du tirage ?

# Méthode récursive : récursion holonome

$$(n + 1)m_n = 3(n - 2)m_{n-2} + (n - 1)m_{n-1} + nm_{n-1}$$



# Méthode récursive : récursion holonome

- Pré-calcul linéaire et stockage des coefficients  $m_i$
- Tirage linéaire
- Pile de récursion

# Méthode récursive : récursion holonome

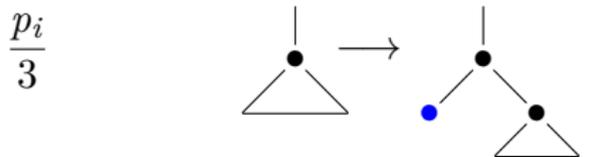
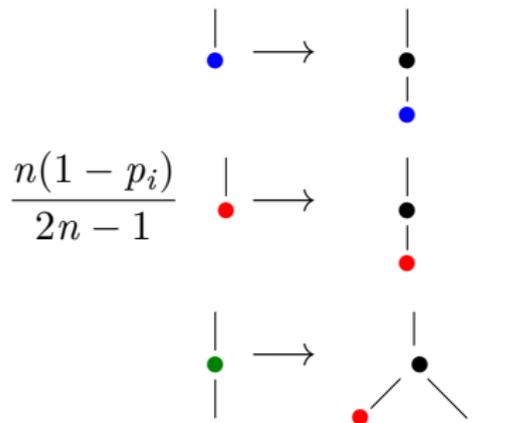
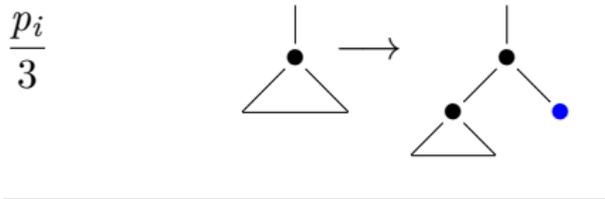
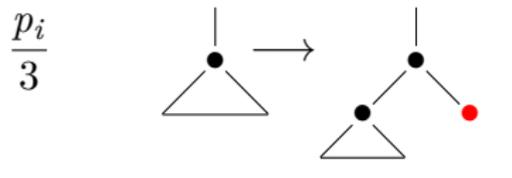
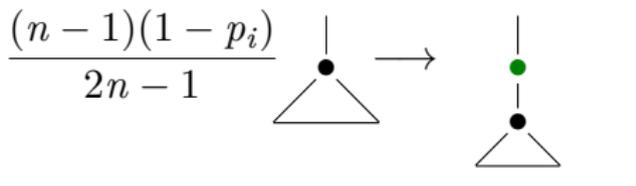
- Pré-calcul linéaire et stockage des coefficients  $m_i$
- Tirage linéaire
- Pile de récursion

Idée : On veut partir d'une feuille et faire grossir l'arbre.

# Méthode récursive : récursion holonome

$$i = |T|$$

Tant que  $i < n$  :



# Algorithme "à la Rémy"

$p_i$  : Probabilité d'un arbre de taille  $i$  de grossir de 2.

On définit  $a_i$ ,  $q_i$  et  $p_i$  comme :

$$a_i = \frac{3(i-2)m_{i-2}}{(n+1)m_i}; \quad q_i = \frac{1 - q_{i-1}}{a_i}; \quad p_i = \frac{1 - q_{n+1}}{q_n}$$

avec :

$$p_1 = \sum_{i=2}^{n-1} \frac{2(-3)^i}{i(+1)(n+2)m_i m_{i+1}}; \quad q_2 = 1 - p_1$$

L'algorithme ainsi obtenu :

- tire en taille exacte
- utilise un **stockage constant** (+ la sortie)
- a une **complexité linéaire**

# Algorithme "à la Rémy"

$p_i$  : Probabilité d'un arbre de taille  $i$  de grossir de 2.

On définit  $a_i$ ,  $q_i$  et  $p_i$  comme :

$$a_i = \frac{3(i-2)m_{i-2}}{(n+1)m_i}; \quad q_i = \frac{1 - q_{i-1}}{a_i}; \quad p_i = \frac{1 - q_{n+1}}{q_n}$$

avec :

$$p_1 = \sum_{i=2}^{n-1} \frac{2(-3)^i}{i(+1)(n+2)m_i m_{i+1}}; \quad q_2 = 1 - p_1$$

L'algorithme ainsi obtenu :

- tire en taille exacte
- utilise un **stockage constant** (+ la sortie)
- a une **complexité linéaire**  
car opérations arithmétiques en  $\Theta(1)$ ...

# Sommaire

- 1 Introduction : Arbres de Catalan
  - Algorithme de Rémy
- 2 Spécification holonome pour les arbres de Motzkin
  - Définition
  - Spécification
- 3 Génération aléatoire
  - Méthode
  - Méthode récursive
  - Algorithme "à la Rémy"
- 4 Perspectives
  - Perspectives

# Perspectives

- Optimiser l'algorithme : passer par des générateurs de Boltzmann critique
- Pondérer différemment les nœuds unaires et binaires
- En faire une méthode générique pour spécification holonome :
  - Trouver une spécification/récursion holonome et **son interprétation constructive**
  - Utiliser la méthode récursive
  - Inverser le temps

# Perspectives

- Optimiser l'algorithme : passer par des générateurs de Boltzmann critique
- Pondérer différemment les nœuds unaires et binaires
- En faire une méthode générique pour spécification holonome :
  - Trouver une spécification/récursion holonome et **son interprétation constructive**
  - Utiliser la méthode récursive
  - Inverser le temps
- Trouver des exemples