Introduction à la combinatoire énumérative et bijective : autour des nombres de Catalan

Mathilde Bouvel

4 octobre 2006

Séminaire thésards

Plan

Plan

Les classes d'objets combinatoires

$\mathcal{C}$ = un ensemble d'objets **discrets**, muni d'une **taille**
Pour chaque n, il y a un nombre **fini** d'objets de taille n dans $\mathcal{C}$

Exemples :

- $\mathcal{B} = \{0,1\}^*$, taille = longueur du mot
- $S = \uplus S_n$, $S_n = \{\text{permutations de } \{1,\dots,n\}\}$
  taille de $\pi \in S = n$ tel que $\pi \in S_n$

Notations :

- $\mathcal{C}_n$ = l'ensemble des objets de $\mathcal{C}$ de taille n
- c_n = cardinalité de $\mathcal{C}_n$
- $\mathcal{C} = \biguplus_{n\geq 0} \mathcal{C}_n$

Les enjeux

Problématiques :

- **Énumérer** les objets selon leur taille :

↪ Trouver la suite $(c_n)_{n\geq 0}$, ou des propriétés de cette suite (forme close, récurrence, asymptotique, ...)

- Générer tous les objets de $\mathcal{C}_n$ pour n donné :

↪ **Génération exhaustive** des objets par récurrence sur leur taille

- **Générer aléatoirement** uniformément un objet de $\mathcal{C}_n$ pour n donné :

↪ Génération de Boltzmann : décomposition des objets en utilisant des constructeurs de base

Les enjeux

Problématiques :

- **Énumérer** les objets selon leur taille :

↪ Trouver la suite $(c_n)_{n\geq 0}$, ou des propriétés de cette suite (forme close, récurrence, asymptotique, ...)

- Générer tous les objets de $\mathcal{C}_n$ pour n donné :

↪ **Génération exhaustive** des objets par récurrence sur leur taille

- **Générer aléatoirement** uniformément un objet de $\mathcal{C}_n$ pour n donné :

↪ Génération de Boltzmann : décomposition des objets en utilisant des constructeurs de base

Les enjeux

Problématiques :

- **Énumérer** les objets selon leur taille :

↪ Trouver la suite $(c_n)_{n\geq 0}$, ou des propriétés de cette suite (forme close, récurrence, asymptotique, ...)

- Générer tous les objets de $\mathcal{C}_n$ pour n donné :

↪ **Génération exhaustive** des objets par récurrence sur leur taille

- **Générer aléatoirement** uniformément un objet de $\mathcal{C}_n$ pour n donné :

↪ Génération de Boltzmann : décomposition des objets en utilisant des constructeurs de base

Les enjeux

Problématiques :

- **Énumérer** les objets selon leur taille :

↪ Trouver la suite $(c_n)_{n\geq 0}$, ou des propriétés de cette suite (forme close, récurrence, asymptotique, ...)

- Générer tous les objets de $\mathcal{C}_n$ pour n donné :

↪ **Génération exhaustive** des objets par récurrence sur leur taille

- **Générer aléatoirement** uniformément un objet de $\mathcal{C}_n$ pour n donné :

↪ Génération de Boltzmann : décomposition des objets en utilisant des constructeurs de base

Les courants

Plusieurs branches au sein de la combinatoire :

- La combinatoire algébrique
- La combinatoire énumérative
- La combinatoire bijective

Pour une utilisation en algorithmique.

Combinatoire algébrique

Étude de la **structure algébrique** des ensembles d'objets combinatoires.

Exemples

- le groupe symétrique S_n, généré par les transpositions
- les fonctions symétriques
- certains polynômes, comme le polynôme chromatique, le polynôme de Tutte

Exemples d'utilisation :

- génération d'objets de la classe en connaissant un ensemble de générateurs
- modélisation de phénomènes physique (modèle du tas de sable)

Combinatoire énumérative

Suite $(c_n)_{n\geq 0}$. **Série génératrice** ordinaire $C(z)$ associée : $\sum_{n\geq 0} c_n z^n$

Outils d'analyse pour trouver des propriétés des c_n

Les classes d'objets combinatoires se décomposent en utilisant des opérateurs de base (union, produit cartésien, ...) : traduction sur les séries génératrices (somme, produit, ...)

Applications :

- génération aléatoire, de Boltzmann en particulier
- équivalent asymptotique de c_n
- formules de récurrence sur les c_n

Combinatoire bijective

Combinatoire énumérative $\longrightarrow$ formule pour c_n
Formule **simple** contre preuve **compliquée** ...

La combinatoire bijective veut expliquer **simplement** une formule simple : trouver une bijection **qui conserve la taille** entre $\mathcal{C}$ et une classe plus simple énumérée par $(c_n)_{n\geq 0}$.

Exemple : cartes unicursales à n arêtes, énumérées par $2^{n-1}\binom{2n-1}{n}$

Combinatoire et algorithmique

Constat : les objets discrets apparaissent naturellement en chimie, biologie, image, ... et on voudrait pouvoir résoudre des problèmes algorithmiques sur ces objets (ex : recherche de sous-objets communs)

Étude combinatoire des objets pour en comprendre la **structure** ; puis utilisation de cette structure pour concevoir des **algorithmes efficaces** répondant au problème de départ.

Minidictionnaire de la combinatoire énumérative

Série génératrice $C(z) = \sum_{n \geq 0} c_n z^n$ associée à une classe combinatoire $\mathcal{C}$

Constructeur	Taille des objets	Opération sur les S.G.
Objet vide $\cdot$	0	1
Atome $\bullet$	1	z
$\mathcal{C} = \mathcal{A} \uplus \mathcal{B}$	taille dans $\mathcal{A}$ ou dans $\mathcal{B}$	$C(z) = A(z) + B(z)$
$\mathcal{C} = \mathcal{A} \times \mathcal{B}$	$\lvert(a, b)\rvert = \lvert a\rvert + \lvert b\rvert$	$C(z) = A(z) \cdot B(z)$
$\mathcal{C} = Seq(\mathcal{A})$	$\lvert(a_1, \ldots, a_n)\rvert = \sum \lvert a_i\rvert$	$C(z) = \frac{1}{1-A(z)}$

Outils d'analyse pour le calcul des coefficients

- Retrouver c_n à partir de $C(z)$: théorème d'**inversion de Lagrange**

$$C(z) = z\phi(C(z)) \Rightarrow [z^n]C(z) = \frac{1}{n}[y^{n-1}]\phi(y)^n$$

- Résultat en provenance d'analyse complexe, pour calculer un **équivalent asymptotique** de c_n :
  $R =$ rayon de convergence de $C(z)$, fini
  c_n est du même ordre de grandeur exponentiel que $\left(\frac{1}{R}\right)^n$

Plan

Les nombres de Catalan

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

$1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, \ldots$

Quelques objets comptés par les nombres de Catalan :

- Permutations triables par une pile
- Arbres plans
- Arbres binaires
- Chemins de Dyck

Eugène Catalan (1814–1894)

Les nombres de Catalan

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

$1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, \ldots$

Quelques objets comptés par les nombres de Catalan :

- Permutations triables par une pile
- Arbres plans
- Arbres binaires
- Chemins de Dyck

Eugène Catalan (1814–1894)

Les nombres de Catalan

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

$1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, \ldots$

Quelques objets comptés par les nombres de Catalan :

- Permutations triables par une pile
- Arbres plans
- Arbres binaires
- Chemins de Dyck

Eugène Catalan (1814–1894)

Permutations triables par une pile

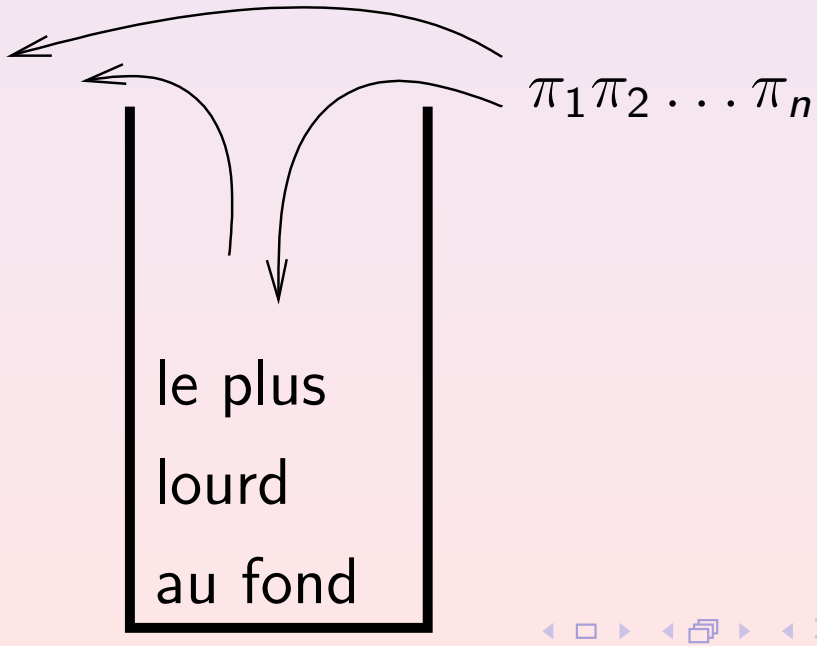
Représentation linéaire des permutations : $\pi =$ mot $\pi_1\pi_2\dots\pi_n$

$\pi = \pi_1\dots\pi_n$ est triable par pile ssi on peut transformer $\pi_1\dots\pi_n$ en $12\dots n$ en utilisant une pile dans laquelle les éléments sont toujours en ordre croissant (le plus petit en haut de pile)

Permutations triables par une pile

Représentation linéaire des permutations : $\pi =$ mot $\pi_1\pi_2\dots\pi_n$

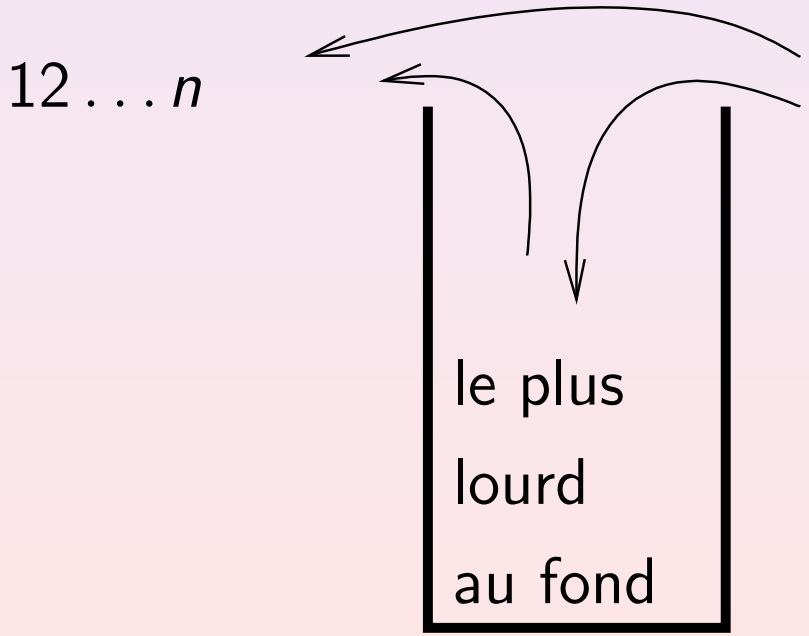
$\pi = \pi_1\dots\pi_n$ est triable par pile ssi on peut transformer $\pi_1\dots\pi_n$ en $12\dots n$ en utilisant une pile dans laquelle les éléments sont toujours en ordre croissant (le plus petit en haut de pile)



Permutations triables par une pile

Représentation linéaire des permutations : $\pi =$ mot $\pi_1\pi_2\dots\pi_n$

$\pi = \pi_1\dots\pi_n$ est triable par pile ssi on peut transformer $\pi_1\dots\pi_n$ en $12\dots n$ en utilisant une pile dans laquelle les éléments sont toujours en ordre croissant (le plus petit en haut de pile)

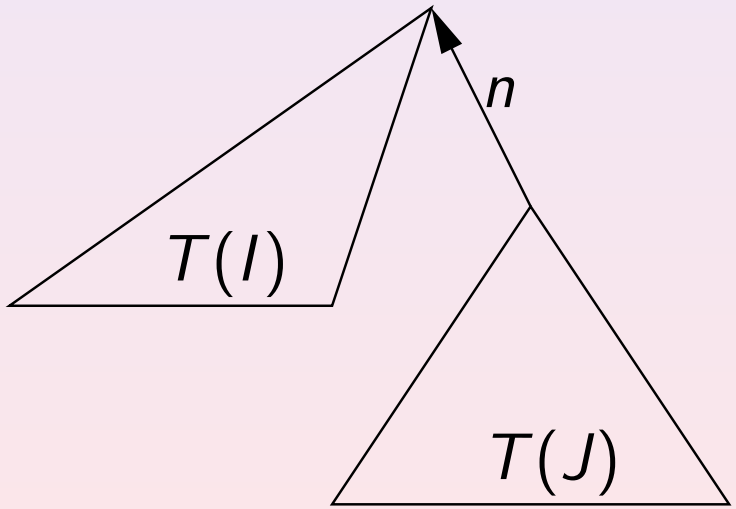


Arbres plans

Arbre plan = arbre plongé dans le plan, enraciné en une arête.

Arbres plans à n arêtes : en bijection avec les permutations triables par pile de taille n

- numérotation postfixe des arêtes, et lecture dans l'ordre préfixe
- permutation triable par pile = $InJ \rightarrow$ arbre ci-contre, récursivement



Arbres binaires

Arbre binaire = arbre plan où chaque noeud (sauf les feuilles) a arité 2

Correspondance bijective entre

- les arbres plans à n arêtes
- les arbres binaires à $n+1$ feuilles

Codage frère-fils :

- premier fils = branche droite
- frère droit = branche gauche

+ Complétion par $n+1$ feuilles

Arbres binaires

Arbre binaire = arbre plan où chaque noeud (sauf les feuilles) a arité 2

Correspondance bijective entre

- les arbres plans à n arêtes
- les arbres binaires à $n+1$ feuilles

Codage frère-fils :

- premier fils = branche droite
- frère droit = branche gauche

+ Complétion par $n+1$ feuilles

Arbres binaires

Arbre binaire = arbre plan où chaque noeud (sauf les feuilles) a arité 2

Correspondance bijective entre

- les arbres plans à n arêtes
- les arbres binaires à $n+1$ feuilles

Codage frère-fils :

- premier fils = branche droite
- frère droit = branche gauche

\+ Complétion par $n+1$ feuilles

Chemins de Dyck, mots de parenthèses

Un **chemin de Dyck** de taille (ou demi-longueur) n est un chemin dans le quart de plan positif, partant de $(0,0)$, arrivant en $(2n,0)$, en faisant des pas $u=(1,1)$ et $d=(1,-1)$

Point de vue mot de parenthèses : $u=($ et $d=)$

En bijection avec les arbres binaires :

- arête gauche = pas montant u
- arête droite = pas descendant d

Chemins de Dyck, mots de parenthèses

Un **chemin de Dyck** de taille (ou demi-longueur) n est un chemin dans le quart de plan positif, partant de $(0,0)$, arrivant en $(2n,0)$, en faisant des pas $u=(1,1)$ et $d=(1,-1)$

Point de vue mot de parenthèses : $u=($ et $d=)$

En bijection avec les arbres binaires :

- arête gauche = pas montant u
- arête droite = pas descendant d

Équations en séries génératrices

Arbres binaires : $[z^n]B(z) =$ nombre d'arbres binaires à n feuilles

$$\mathcal{B} = \bullet \uplus \mathcal{B} \times \mathcal{B} \Rightarrow B(z) = z + B(z)^2$$

Résolution de l'équation : $B(z) = \frac{1-\sqrt{1-4z}}{2}$

Chemins de Dyck : $[z^n]D(z) =$ nombre de chemins de Dyck de demi-longueur n

$$\mathcal{D} = \cdot \uplus \mathcal{Z} \times \mathcal{D} \times \mathcal{D} \Rightarrow D(z) = 1 + zD(z)^2$$

Résolution de l'équation : $D(z) = \frac{1-\sqrt{1-4z}}{2z}$

$C(z) = \frac{1-\sqrt{1-4z}}{2z} \Rightarrow$ l'ordre de grandeur exponentiel de C_n est 4^n

Équations en séries génératrices

Arbres binaires : $[z^n]B(z) =$ nombre d'arbres binaires à n feuilles

$$\mathcal{B} = \bullet \uplus \mathcal{B} \times \mathcal{B} \Rightarrow B(z) = z + B(z)^2$$

Résolution de l'équation : $B(z) = \frac{1-\sqrt{1-4z}}{2}$

Chemins de Dyck : $[z^n]D(z) =$ nombre de chemins de Dyck de demi-longueur n

$$\mathcal{D} = \cdot \uplus \mathcal{Z} \times \mathcal{D} \times \mathcal{D} \Rightarrow D(z) = 1 + zD(z)^2$$

Résolution de l'équation : $D(z) = \frac{1-\sqrt{1-4z}}{2z}$

$C(z) = \frac{1-\sqrt{1-4z}}{2z} \Rightarrow$ l'ordre de grandeur exponentiel de C_n est 4^n

Énumération des chemins de Dyck par le lemme cyclique

$\omega \in \{u, d\}^*$ mot "cyclique" avec n u et $n+1$ d $\Rightarrow$ il existe un unique représentant de ω qui est un mot de Dyck suivi d'un d

Réciproquement, un mot de Dyck définit un unique tel ω

Conséquence : le nombre de mot de Dyck de demi-longueur n est

$$\frac{1}{2n+1}\binom{2n+1}{n} = \frac{1}{n+1}\binom{2n}{n} = C_n$$

Pour conclure sur Catalan

Preuve par méthode bijective et énumérative que

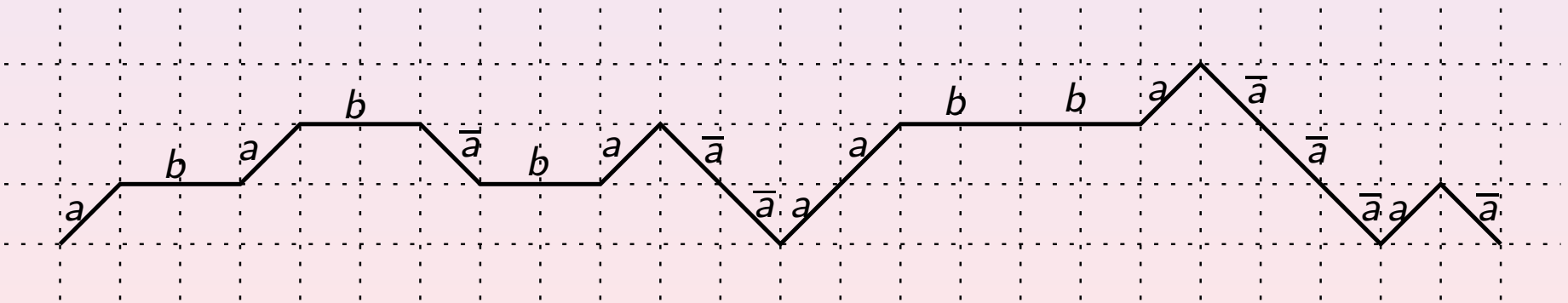
- Les permutations triables par une pile de taille n,
- les arbres plans à n arêtes,
- les arbres binaires à $n+1$ feuilles,
- les chemins de Dyck de demi-longueur n

sont en nombre $C_n = \frac{1}{n+1}\binom{2n}{n}$

Plan

Définition des chemins de Schröder

Un chemin de Schröder de taille n est un chemin dans le quart de plan positif, commençant en $(0,0)$ et arrivant en $(2n,0)$, faisant des pas montants a, de coordonnées $(1,1)$, des pas descendants $\overline{a}$, de coordonnées $(1,-1)$ et des "doubles" pas horizontaux b, de coordonnées $(2,0)$.



Énumération des chemins de Schröder

Grands nombres de Schröder : $S_n = \sum_{i=0}^{n} \binom{2n-i}{i} C_{n-i}$

$$1, 2, 6, 22, 90, 394, 1806, 8558, 41586, 206098$$

Il y a S_n chemins de Schröder de taille (= demi-longueur) n

Série génératrice :

$$\mathcal{S} = \cdot \uplus \mathcal{Z} \times \mathcal{S} \uplus \mathcal{Z} \times \mathcal{S} \times \mathcal{S}$$

$$\Rightarrow S(z) = 1 + zS(z) + zS(z)^2$$

$$\Rightarrow S(z) = \frac{1 - z - \sqrt{z^2 - 6z + 1}}{2z}$$

Arbres de Schröder (non-signés)

Arbre de Schröder = arbre plan où l'arité de tout noeud interne est supérieure ou égale à 2

Nombre d'arbres de Schröder à n feuilles : le $(n-1)$-ième **petit** nombre de Schröder T_{n-1}
$S_n = 2T_n$ dès que $n \geq 1$

Bijection entre les arbres de Schröder et la **moitié** des chemins de Schröder : sur le modèle de la bijection entre mots de Dyck et arbres binaires

Arbres de Schröder signés

Signe $+$ ou $-$ à la racine de l'arbre : il y a S_{n-1} arbres de Schröder signés à n feuilles.

On propage les signes à tous les noeuds internes : un noeud a le signe opposé de son père.

À un arbre de Schröder signé à n feuilles, on associe une permutation de taille n. Les permutations ainsi obtenues sont les **séparables**.

Permutations séparables

Celles qui possèdent un **arbre de séparation** = qui sont le support d'un arbre de Schröder signé.

Arbre de séparation = arbre de Schröder signé.

Bijection entre permutations séparables et arbres de Schröder signés. D'où preuve bijective du résultat d'énumération des séparables :

Il y a S_{n-1} permutations séparables de taille n.

Caractérisation : l'arbre de décomposition

À toute permutation on peut associer un arbre : **décomposition en "intervalles communs"**. Analogue de la décomposition modulaire des graphes.

Deux types de noeuds dans cet arbre : noeuds **linéaires** et noeuds **premiers**.

Caractérisation des permutations **séparables** : celles dont l'arbre de décomposition n'a **pas de noeuds premiers**.
En outre, pour une séparable : arbre de séparation = arbre de décomposition

Plan

Liens entre graphes et permutations

- Problème algorithmique : recherche de plus grande "sous-structure" commune
  - Plus grand sous-graphe commun : *NP*-complet
  - Plus grand sous-arbre commun : dans *P*
  - Plus grand motif commun à deux permutations : *NP*-dur
  - Occurrence d'un motif dans une permutation : *NP*-complet

- Plus grand sous-arbre commun = plus grand motif commun à deux permutations triables par pile

Liens entre graphes et permutations

- Problème algorithmique : recherche de plus grande "sous-structure" commune
  - Plus grand sous-graphe commun : *NP*-complet
  - Plus grand sous-arbre commun : dans *P*
  - Plus grand motif commun à deux permutations : *NP*-dur
  - Occurrence d'un motif dans une permutation : *NP*-complet

- Plus grand sous-arbre commun = plus grand motif commun à deux permutations triables par pile

Définition d'un motif

Représentation linéaire des permutations : π = mot $\pi_1\pi_2\dots\pi_n$

$\pi \in S_n$, $\tau \in S_k$ avec $k \leq n$

- La permutation π *contient* une occurrence du motif τ ssi $\exists$ $1 \leq i_1 < i_2 < \dots < i_k \leq n$ tels que $\pi_{i_1}\pi_{i_2}\dots\pi_{i_k}$ est isomorphe en ordre à τ : $\pi_{i_p} < \pi_{i_q}$ ssi $\tau_p < \tau_q$

- Par exemple, $\mathbf{1}25\mathbf{6}3\mathbf{4}$ contient 132 et évite 321

Deux problèmes algorithmiques sur les motifs dans les permutations

- Recherche d'occurrence de motif dans une permutation :
  en entrée : un motif τ et une permutation π
  en sortie : un booléen, Vrai ssi il existe une occurrence de τ dans π

- Recherche de plus grand motif commun à deux permutations :
  en entrée : deux permutations π_1 et π_2
  en sortie : un motif qui possède une occurrence dans π_1, une dans π_2, et qui est de longueur maximale

Deux problèmes algorithmiques sur les motifs dans les permutations

- Recherche d'occurrence de motif dans une permutation :
  en entrée : un motif τ et une permutation π
  en sortie : un booléen, Vrai ssi il existe une occurrence de τ dans π

- Recherche de plus grand motif commun à deux permutations :
  en entrée : deux permutations π_1 et π_2
  en sortie : un motif qui possède une occurrence dans π_1, une dans π_2, et qui est de longueur maximale

Complexité de la recherche d'une occurrence d'un motif

- Motif quelconque : problème *NP*-complet
- Motif **séparable** : problème résolu en temps **polynomial**
  - Algorithme de Bose, Buss et Lubiw (BBL) : $O(kn^6)$ en temps, $O(kn^4)$ en espace
  - Algorithme d'Ibarra : $O(kn^4)$ en temps, $O(kn^3)$ en espace
- Exploiter la structure d'**arbre de séparation** (précalcul)
- Outil clé = **programmation dynamique**

Complexité de la recherche d'une occurrence d'un motif

- Motif quelconque : problème NP-complet
- Motif **séparable** : problème résolu en temps **polynomial**
  - Algorithme de Bose, Buss et Lubiw (BBL) : $\mathcal{O}(kn^6)$ en temps, $\mathcal{O}(kn^4)$ en espace
  - Algorithme d'Ibarra : $\mathcal{O}(kn^4)$ en temps, $\mathcal{O}(kn^3)$ en espace

- Exploiter la structure d'**arbre de séparation** (précalcul)
- Outil clé = **programmation dynamique**

Complexité de la recherche d'une occurrence d'un motif

- Motif quelconque : problème NP-complet
- Motif **séparable** : problème résolu en temps **polynomial**
  - Algorithme de Bose, Buss et Lubiw (BBL) : $\mathcal{O}(kn^6)$ en temps, $\mathcal{O}(kn^4)$ en espace
  - Algorithme d'Ibarra : $\mathcal{O}(kn^4)$ en temps, $\mathcal{O}(kn^3)$ en espace
- Exploiter la structure d'**arbre de séparation** (précalcul)
- Outil clé = **programmation dynamique**

Complexité de la recherche de plus grand motif commun

- Pas d'hypothèse sur les deux permutations en entrée : problème *NP*-difficile, complexité exacte encore inconnue
- Une des permutations est **séparable** : problème résolu en temps **polynomial**
  - Adaptation de l'algorithme de Bose, Buss et Lubiw
- Si **l'arbre de décomposition** d'une des deux permutations a tous ses **noeuds premiers d'arité bornée par une constante** d : problème résolu en temps **polynomial**
  - Extension de l'algorithme de Bose, Buss et Lubiw adapté

Complexité de la recherche de plus grand motif commun

- Pas d'hypothèse sur les deux permutations en entrée : problème *NP*-difficile, complexité exacte encore inconnue
- Une des permutations est **séparable** : problème résolu en temps **polynomial**
  - Adaptation de l'algorithme de Bose, Buss et Lubiw
- Si **l'arbre de décomposition** d'une des deux permutations a tous ses **noeuds premiers d'arité bornée par une constante** d : problème résolu en temps **polynomial**
  - Extension de l'algorithme de Bose, Buss et Lubiw adapté

Complexité de la recherche de plus grand motif commun

- Pas d'hypothèse sur les deux permutations en entrée : problème *NP*-difficile, complexité exacte encore inconnue
- Une des permutations est **séparable** : problème résolu en temps **polynomial**
  - Adaptation de l'algorithme de Bose, Buss et Lubiw
- Si **l'arbre de décomposition** d'une des deux permutations a tous ses **noeuds premiers d'arité bornée par une constante** d : problème résolu en temps **polynomial**
  - Extension de l'algorithme de Bose, Buss et Lubiw adapté

Conclusion

- Les problèmes qu'on se pose en combinatoire
- Différentes techniques utilisées pour résoudre ces problèmes
- Exemple "historique" : les nombres de Catalan
- Problème actuel : autour des nombres de Schröder, en lien avec l'algorithmique