

Sur le calcul d'un rang dans les graphes complets

Yvan Le Borgne
CNRS, LaBRI, Université Bordeaux 1
(en collaboration avec Robert Cori)

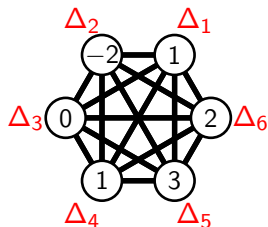
Journées ALÉA, CIRM, 19 Mars 2013

1/3 Introduction au résultat principal

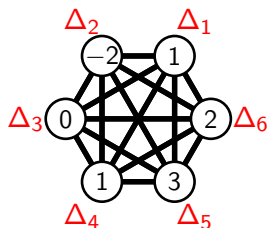
Un problème d'optimisation définissant le rang.

Configuration: $u = (u_i)_{i=1,\dots,n}$

ex: $u = (1, -2, 0, 1, 3, 2)$.



Un problème d'optimisation définissant le rang.



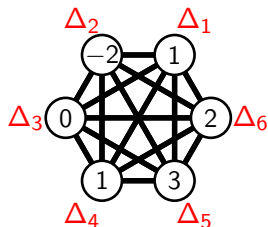
Configuration: $u = (u_i)_{i=1, \dots, n}$

ex: $u = (1, -2, 0, 1, 3, 2)$.

Éboulement du sommet i : $u \longrightarrow u + \Delta_i$

ex: $\Delta_2 = (1, -5, 1, 1, 1, 1)$.

Un problème d'optimisation définissant le rang.



Configuration: $u = (u_i)_{i=1, \dots, n}$

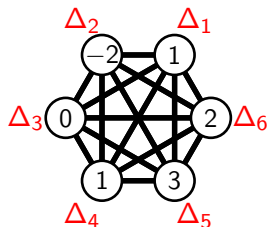
ex: $u = (1, -2, 0, 1, 3, 2)$.

Éboulement du sommet i : $u \longrightarrow u + \Delta_i$

ex: $\Delta_2 = (1, -5, 1, 1, 1, 1)$.

Équivalence: $u \equiv_{\Delta} v \iff u = v + \sum_i a_i \Delta_i$
avec $(a_i)_i \in \mathbb{Z}^n$.

Un problème d'optimisation définissant le rang.



Configuration: $u = (u_i)_{i=1, \dots, n}$

ex: $u = (1, -2, 0, 1, 3, 2)$.

Éboulement du sommet i : $u \longrightarrow u + \Delta_i$

ex: $\Delta_2 = (1, -5, 1, 1, 1, 1)$.

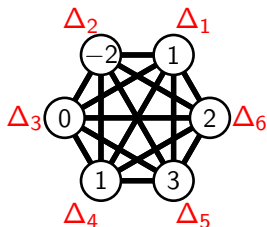
Équivalence: $u \equiv_{\Delta} v \iff u = v + \sum_i a_i \Delta_i$

avec $(a_i)_i \in \mathbb{Z}^n$.

Identité: $\sum_i \Delta_i = 0$

et donc $-\Delta_i = \sum_{i \neq j} \Delta_j$.

Un problème d'optimisation définissant le rang.



Configuration: $u = (u_i)_{i=1, \dots, n}$

ex: $u = (1, -2, 0, 1, 3, 2)$.

Éboulement du sommet i : $u \longrightarrow u + \Delta_i$

ex: $\Delta_2 = (1, -5, 1, 1, 1, 1)$.

Équivalence: $u \equiv_{\Delta} v \iff u = v + \sum_i a_i \Delta_i$

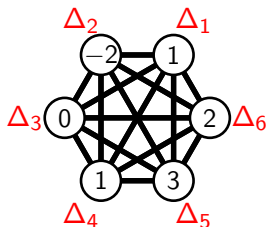
avec $(a_i)_i \in \mathbb{Z}^n$.

Identité: $\sum_i \Delta_i = 0$

et donc $-\Delta_i = \sum_{i \neq j} \Delta_j$.

u est effective si $u \equiv_{\Delta} v \geq 0$.

Un problème d'optimisation définissant le rang.



Configuration: $u = (u_i)_{i=1, \dots, n}$

ex: $u = (1, -2, 0, 1, 3, 2)$.

Éboulement du sommet i : $u \longrightarrow u + \Delta_i$

ex: $\Delta_2 = (1, -5, 1, 1, 1, 1)$.

Équivalence: $u \equiv_{\Delta} v \iff u = v + \sum_i a_i \Delta_i$

avec $(a_i)_i \in \mathbb{Z}^n$.

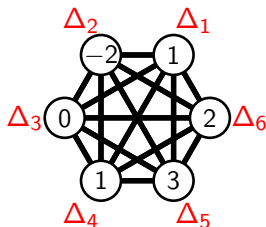
Identité: $\sum_i \Delta_i = 0$

et donc $-\Delta_i = \sum_{i \neq j} \Delta_j$.

u est effective si $u \equiv_{\Delta} v \geq 0$.

$\text{degree}(u) = \sum_{i=1}^n u_i$

Un problème d'optimisation définissant le rang.



Configuration: $u = (u_i)_{i=1, \dots, n}$

ex: $u = (1, -2, 0, 1, 3, 2)$.

Éboulement du sommet i : $u \rightarrow u + \Delta_i$

ex: $\Delta_2 = (1, -5, 1, 1, 1, 1)$.

Équivalence: $u \equiv_{\Delta} v \iff u = v + \sum_i a_i \Delta_i$

avec $(a_i)_i \in \mathbb{Z}^n$.

Identité: $\sum_i \Delta_i = 0$

et donc $-\Delta_i = \sum_{i \neq j} \Delta_j$.

u est effective si $u \equiv_{\Delta} v \geq 0$.

$\text{degree}(u) = \sum_{i=1}^n u_i$

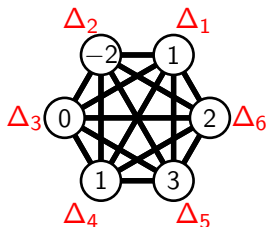
pas effective

effective

$\text{degree}(u) < 0$

$\text{degree}(u) > m - n$

Un problème d'optimisation définissant le rang.



Configuration: $u = (u_i)_{i=1, \dots, n}$

ex: $u = (1, -2, 0, 1, 3, 2)$.

Éboulement du sommet i : $u \rightarrow u + \Delta_i$

ex: $\Delta_2 = (1, -5, 1, 1, 1, 1)$.

Équivalence: $u \equiv_{\Delta} v \iff u = v + \sum_i a_i \Delta_i$

avec $(a_i)_i \in \mathbb{Z}^n$.

Identité: $\sum_i \Delta_i = 0$

et donc $-\Delta_i = \sum_{i \neq j} \Delta_j$.

u est effective si $u \equiv_{\Delta} v \geq 0$.

$\text{degree}(u) = \sum_{i=1}^n u_i$

pas effective

effective

$\text{degree}(u) < 0$

$\text{degree}(u) > m - n$

$$\text{rank}(u) = \min_{f \geq 0 | u-f \text{ pas effective}} \text{degree}(f) - 1.$$

[Baker, Norine 2007]

Une définition directe du rang sur le graphe complet I

Supposons

$$u = (u_1, \dots, u_{n-1}; s)$$

où $0 \leq u_i \leq i - 1$ et $u_i \leq u_{i+1}$.

ex:

$$u = (0, 1, 2, 3, 4, 5, 6; 18)$$

$$\text{rank}(u) = -1 + \sum_{i=1}^{n-1} \max(0, Q - i + u_i + \delta_{i < R})$$

où $s = (n - 1)Q + R$ est une division euclidienne.

Une définition directe du rang sur le graphe complet I

0 0 0 1 1 1 4 7 7 9 (26)

Supposons

$$u = (u_1, \dots, u_{n-1}; s)$$

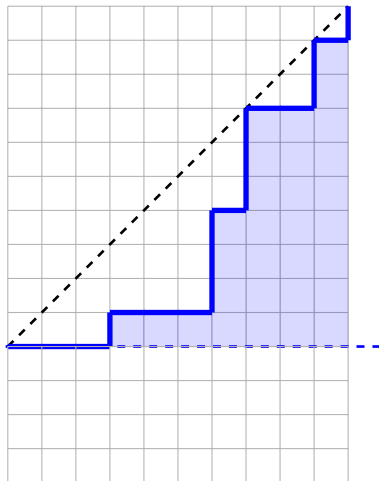
où $0 \leq u_i \leq i - 1$ et $u_i \leq u_{i+1}$.

ex:

$$u = (0, 1, 2, 3, 4, 5, 6; 18)$$

$$\text{rank}(u) = -1 + \sum_{i=1}^{n-1} \max(0, Q - i + u_i + \delta_{i < R})$$

où $s = (n - 1)Q + R$ est une division euclidienne.



Une définition directe du rang sur le graphe complet I

0 0 0 1 1 1 4 7 7 9 **26**

Supposons

$$u = (u_1, \dots, u_{n-1}; s)$$

où $0 \leq u_i \leq i - 1$ et $u_i \leq u_{i+1}$.

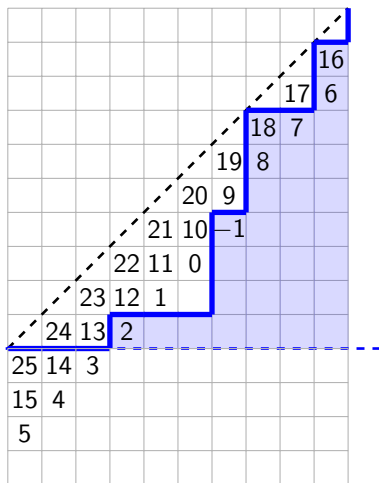
ex:

$$u = (0, 1, 2, 3, 4, 5, 6; 18)$$

$$\text{rank}(u) = -1 +$$

$$\sum_{i=1}^{n-1} \max(0, Q - i + u_i + \delta_{i < R})$$

où $s = (n - 1)Q + R$ est une division euclidienne.



Une définition directe du rang sur le graphe complet I

Supposons

$$u = (u_1, \dots, u_{n-1}; s)$$

où $0 \leq u_i \leq i - 1$ et $u_i \leq u_{i+1}$.

ex:

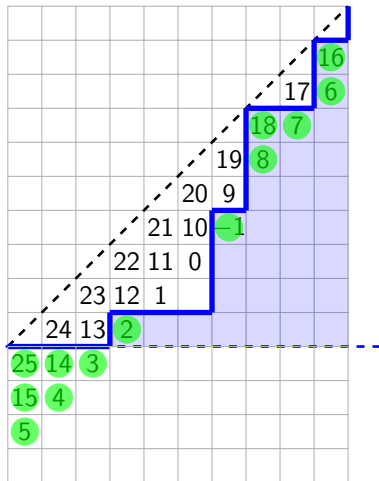
$$u = (0, 1, 2, 3, 4, 5, 6; 18)$$

$$\text{rank}(u) = -1 +$$

$$\sum_{i=1}^{n-1} \max(0, Q - i + u_i + \delta_{i < R})$$

où $s = (n - 1)Q + R$ est une division euclidienne.

0 0 0 1 1 1 4 7 7 9 **26**



$$3 \ 2 \ 1 \ 1 \ 0 \ 0 \ 1 \ 2 \ 1 \ 2 \ = 14$$

Une définition directe du rang sur le graphe complet I

Supposons

$$u = (u_1, \dots, u_{n-1}; s)$$

où $0 \leq u_i \leq i - 1$ et $u_i \leq u_{i+1}$.

ex:

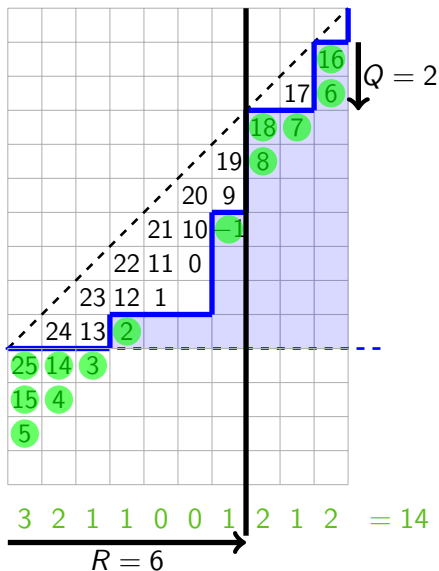
$$u = (0, 1, 2, 3, 4, 5, 6; 18)$$

$$\text{rank}(u) = -1 +$$

$$\sum_{i=1}^{n-1} \max(0, Q - i + u_i + \delta_{i < R})$$

où $s = (n - 1)Q + R$ est une division euclidienne.

0 0 0 1 1 1 4 7 7 9 **26**



Une définition directe du rang sur le graphe complet II

Pour toute configuration u et combinaison d'écroulements $\sum_i a_i \Delta_i$:

$$\text{rank}(u + \sum_i a_i \Delta_i) = \text{rank}(u).$$

Une définition directe du rang sur le graphe complet II

Pour toute configuration u et combinaison d'éboulements $\sum_i a_i \Delta_i$:

$$\text{rank}(u + \sum_i a_i \Delta_i) = \text{rank}(u).$$

Dans la classe d'équivalence de u il y a une unique n -fonction de parking $\text{parking}(u)$.

Une définition directe du rang sur le graphe complet II

Pour toute configuration u et combinaison d'éboulements $\sum_i a_i \Delta_i$:

$$\text{rank}(u + \sum_i a_i \Delta_i) = \text{rank}(u).$$

Dans la classe d'équivalence de u il y a une unique n -fonction de parking $\text{parking}(u)$.

Pour toute permutation $\sigma \in S_{n-1}$, $\sigma.u = (u_{\sigma(1)}, \dots, u_{\sigma(n-1)}, s)$.

Une définition directe du rang sur le graphe complet II

Pour toute configuration u et combinaison d'éboulements $\sum_i a_i \Delta_i$:

$$\text{rank}(u + \sum_i a_i \Delta_i) = \text{rank}(u).$$

Dans la classe d'équivalence de u il y a une unique n -fonction de parking $\text{parking}(u)$.

Pour toute permutation $\sigma \in S_{n-1}$, $\sigma.u = (u_{\sigma(1)}, \dots, u_{\sigma(n-1)}, s)$.

$$\text{rank}(\sigma.u) = \text{rank}(u).$$

Une définition directe du rang sur le graphe complet II

Pour toute configuration u et combinaison d'éboulements $\sum_i a_i \Delta_i$:

$$\text{rank}(u + \sum_i a_i \Delta_i) = \text{rank}(u).$$

Dans la classe d'équivalence de u il y a une unique n -fonction de parking $\text{parking}(u)$.

Pour toute permutation $\sigma \in S_{n-1}$, $\sigma.u = (u_{\sigma(1)}, \dots, u_{\sigma(n-1)}, s)$.

$$\text{rank}(\sigma.u) = \text{rank}(u).$$

Pour toute configuration u , soit σ une permutation triant $\text{parking}(u)$, on a donc

$$\text{rank}(u) = \text{rank}(\sigma.\text{parking}(u))$$

où $\sigma.\text{parking}(u)$ satisfait les hypothèses nécessaires.

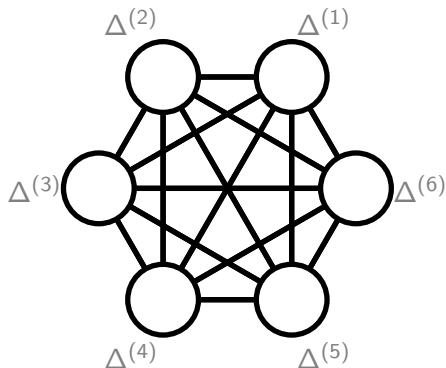
Plan de la suite de l'exposé

- ▶ Cheminement vers la formule directe du rang.
 - ▶ Calcul de $parking(u)$ et test du caractère effectif.
 - ▶ Algorithme glouton pour la suppression de grains.
 - ▶ Interprétation géométrique de la suppression d'un grain.
- ▶ *Interprétation géométrique de toutes les suppressions de grains nécessaire au calcul du rang. (fait.)*
- ▶ Étude énumérative (Optionnelle).
 - ▶ Symétrie sur la bivariate ($degree, rank$) induite par un théorème de Baker et Norine.
 - ▶ Reparamétrisation ($xpara, ypara$).
 - ▶ Chemins binomiaux.
 - ▶ Calculs par empilements de dimères sur une ligne.
- ▶ Remarques.

2/3 Cheminement vers la formule directe du rang.

Calcul naif de la fonction de parking $\text{parking}(u)$ sur K_n .

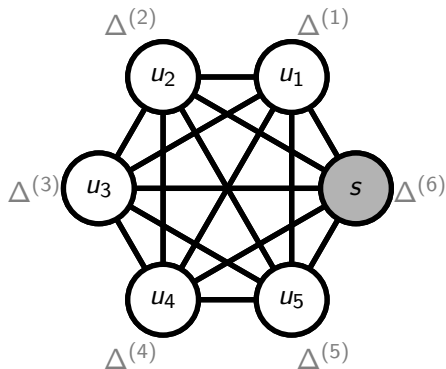
Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.



$\text{parking}(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Calcul naif de la fonction de parking $\text{parking}(u)$ sur K_n .

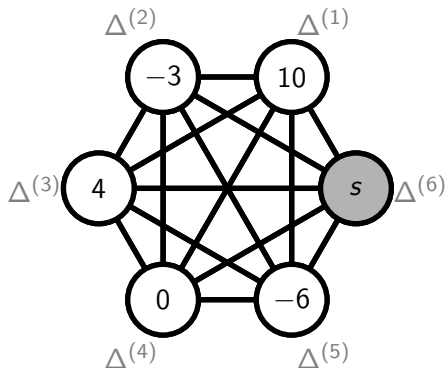
Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.



$\text{parking}(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Calcul naif de la fonction de parking $\text{parking}(u)$ sur K_n .

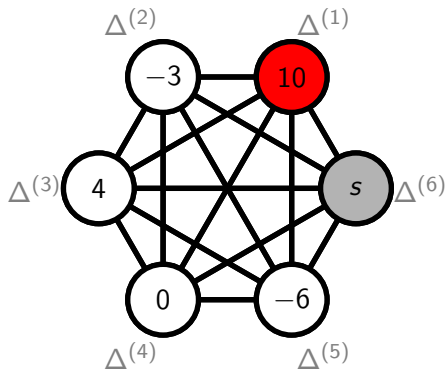
Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.



$\text{parking}(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Calcul naif de la fonction de parking $\text{parking}(u)$ sur K_n .

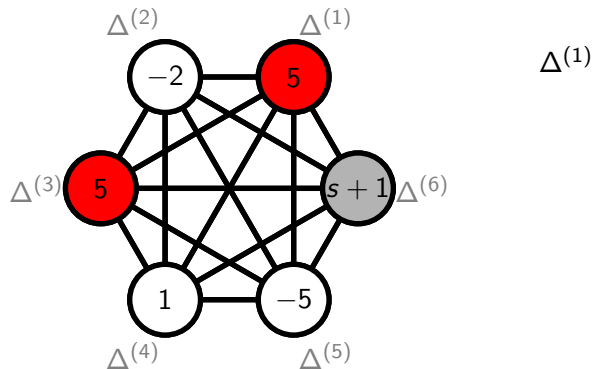
Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.



$\text{parking}(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Calcul naif de la fonction de parking $\text{parking}(u)$ sur K_n .

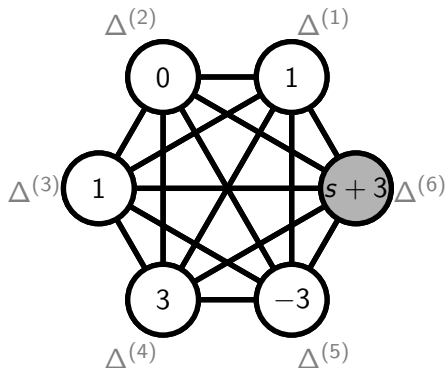
Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.



$\text{parking}(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Calcul naif de la fonction de parking $\text{parking}(u)$ sur K_n .

Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.

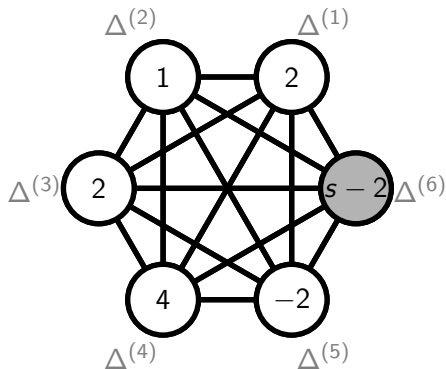


$\Delta^{(1)}$
 $\Delta^{(1)}; \Delta^{(3)}$
Stable

$\text{parking}(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Calcul naif de la fonction de parking $\text{parking}(u)$ sur K_n .

Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.

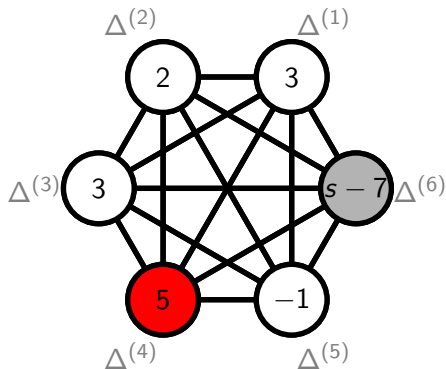


$\Delta^{(1)}$
 $\Delta^{(1)}; \Delta^{(3)}$
Stable
 $\Delta^{(6)}$

$\text{parking}(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Calcul naif de la fonction de parking $\text{parking}(u)$ sur K_n .

Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.

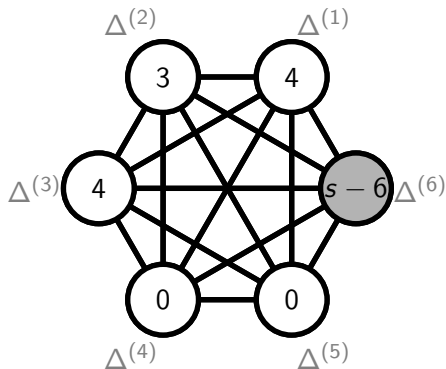


$\Delta^{(1)}$
 $\Delta^{(1)}; \Delta^{(3)}$
 Stable
 $\Delta^{(6)}$
 $\Delta^{(6)}$

$\text{parking}(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Calcul naif de la fonction de parking $parking(u)$ sur K_n .

Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.

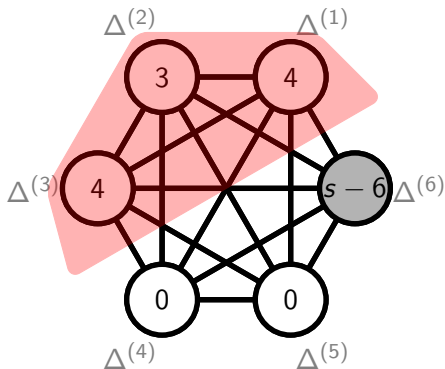


$\Delta^{(1)}$
 $\Delta^{(1)}; \Delta^{(3)}$
 Stable
 $\Delta^{(6)}$
 $\Delta^{(6)}; \Delta^{(4)}$
 Positive

$parking(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Calcul naif de la fonction de parking $\text{parking}(u)$ sur K_n .

Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.

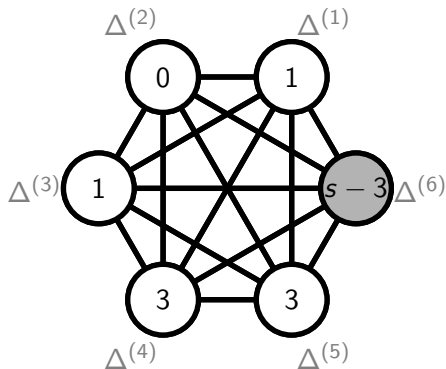


$\Delta^{(1)}$
 $\Delta^{(1)}; \Delta^{(3)}$
Stable
 $\Delta^{(6)}$
 $\Delta^{(6)}; \Delta^{(4)}$
Positive

$\text{parking}(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Calcul naif de la fonction de parking $parking(u)$ sur K_n .

Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.

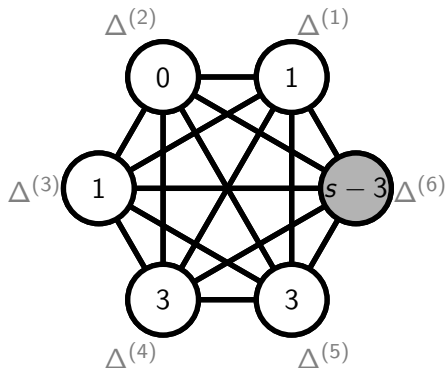


$\Delta^{(1)}$
 $\Delta^{(1)}; \Delta^{(3)}$
 Stable
 $\Delta^{(6)}$
 $\Delta^{(6)}; \Delta^{(4)}$
 Positive
 $\Delta^{(1)}; \Delta^{(2)}; \Delta^{(3)}$
 Superstable

$parking(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Calcul naif de la fonction de parking $parking(u)$ sur K_n .

Une configuration est de parking relativement à un puits si:
elle est stable, positive hors du puits et superstable.



$\Delta^{(1)}$
 $\Delta^{(1)}; \Delta^{(3)}$
Stable
 $\Delta^{(6)}$
 $\Delta^{(6)}; \Delta^{(4)}$
Positive
 $\Delta^{(1)}; \Delta^{(2)}; \Delta^{(3)}$
Superstable

Pour K_n , un calcul plus arithmétique détermine $parking(u)$ en temps linéaire.

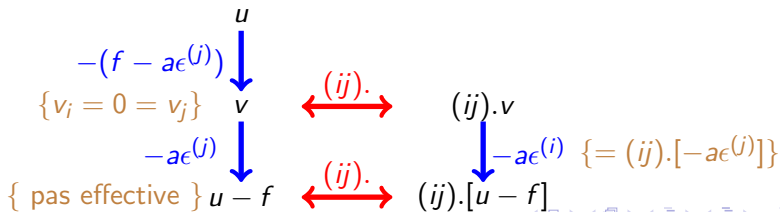
$parking(u) = (u'_1, \dots, u'_{n-1}; s')$ effective si et seulement si $s' \geq 0$.

Un algorithme glouton calculant le rang sur K_n

$$\text{rank}(u) = \min_{f \geq 0 \mid u - f \text{ pas effective}} \text{deg}(f) - 1$$

- 1: $u \leftarrow \text{parking}(u)$; $\text{rank} \leftarrow -1$
- 2: **Tant que** $s \geq 0$:
- 3: $u \leftarrow \text{enlever un grain sur un sommet sans grain de } u$
- 4: $u \leftarrow \text{parking}(u)$
- 5: $\text{rank} \leftarrow \text{rank} + 1$
- 6: Retourner rank

Pour une configuration $u \geq 0$, telle que $u_i = 0$, il existe une suppression optimale de grains g telle que $g_i > 0$.



Une interprétation géométrique d'une itération.

Configuration parking maximale.

$$(0, 1, 2, 3, 4, 5, 6, 7; s) \geq$$

Configuration parking initiale.

$$(0, 0, 1, 1, 4, 4, 5, 6; s)$$

Enlever un grain sur le premier 0.

$$(-1, 0, 1, 1, 4, 4, 5, 6; s)$$

Ebouler le puits pour positiver.

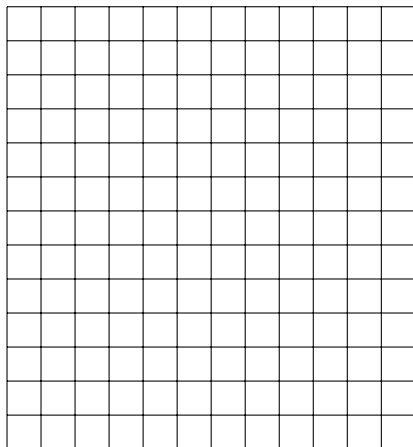
$$(0, 1, 2, 2, 5, 5, 6, 7; s - 8)$$

Ebouler ailleurs pour parking.

$$(4, 5, 6, 6, 0, 0, 1, 2; s - 4)$$

Trier pour représenter l'orbite.

$$(0, 0, 1, 2, 4, 5, 6, 6; s - 4)$$



Une interprétation géométrique d'une itération.

Configuration parking maximale.

$$(0, 1, 2, 3, 4, 5, 6, 7; s) \geq$$

Configuration parking initiale.

$$(0, 0, 1, 1, 4, 4, 5, 6; s)$$

Enlever un grain sur le premier 0.

$$(-1, 0, 1, 1, 4, 4, 5, 6; s)$$

Ebouler le puits pour positiver.

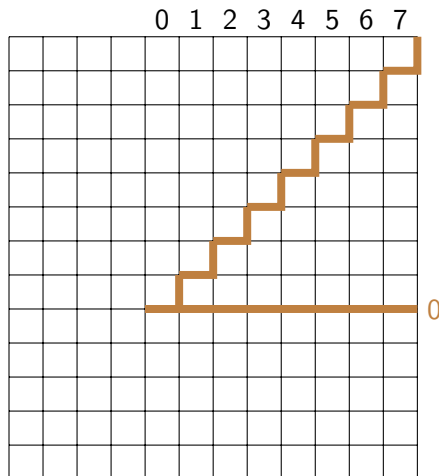
$$(0, 1, 2, 2, 5, 5, 6, 7; s - 8)$$

Ebouler ailleurs pour parking.

$$(4, 5, 6, 6, 0, 0, 1, 2; s - 4)$$

Trier pour représenter l'orbite.

$$(0, 0, 1, 2, 4, 5, 6, 6; s - 4)$$



Une interprétation géométrique d'une itération.

Configuration parking maximale.

$$(0, 1, 2, 3, 4, 5, 6, 7; s) \geq$$

Configuration parking initiale.

$$(0, 0, 1, 1, 4, 4, 5, 6; s)$$

Enlever un grain sur le premier 0.

$$(-1, 0, 1, 1, 4, 4, 5, 6; s)$$

Ebouler le puits pour positiver.

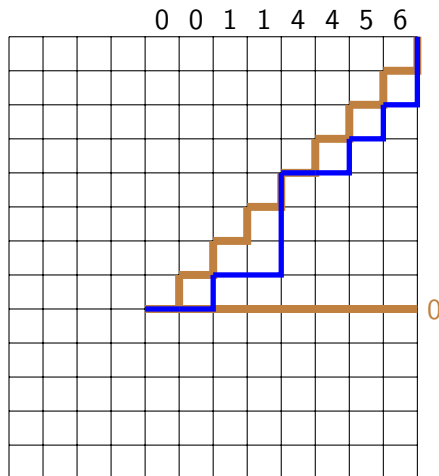
$$(0, 1, 2, 2, 5, 5, 6, 7; s - 8)$$

Ebouler ailleurs pour parking.

$$(4, 5, 6, 6, 0, 0, 1, 2; s - 4)$$

Trier pour représenter l'orbite.

$$(0, 0, 1, 2, 4, 5, 6, 6; s - 4)$$



Une interprétation géométrique d'une itération.

Configuration parking maximale.

$$(0, 1, 2, 3, 4, 5, 6, 7; s) \geq$$

Configuration parking initiale.

$$(0, 0, 1, 1, 4, 4, 5, 6; s)$$

Enlever un grain sur le premier 0.

$$(-1, 0, 1, 1, 4, 4, 5, 6; s)$$

Ebouler le puits pour positiver.

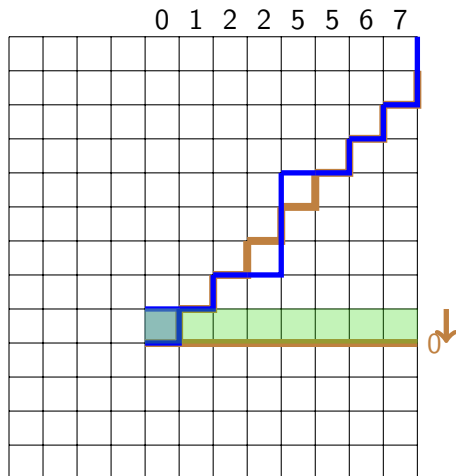
$$(0, 1, 2, 2, 5, 5, 6, 7; s - 8)$$

Ebouler ailleurs pour parking.

$$(4, 5, 6, 6, 0, 0, 1, 2; s - 4)$$

Trier pour représenter l'orbite.

$$(0, 0, 1, 2, 4, 5, 6, 6; s - 4)$$



Une interprétation géométrique d'une itération.

Configuration parking maximale.

$$(0, 1, 2, 3, 4, 5, 6, 7; s) \geq$$

Configuration parking initiale.

$$(0, 0, 1, 1, 4, 4, 5, 6; s)$$

Enlever un grain sur le premier 0.

$$(-1, 0, 1, 1, 4, 4, 5, 6; s)$$

Ebouler le puits pour positiver.

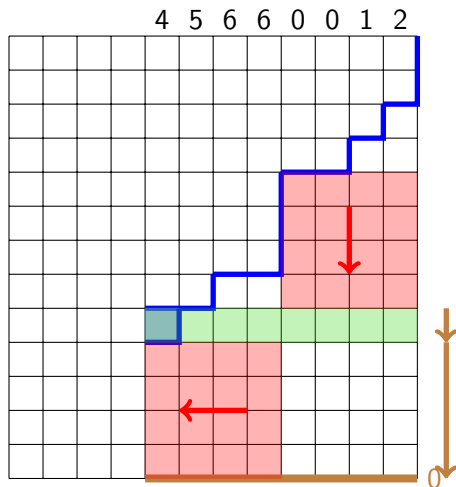
$$(0, 1, 2, 2, 5, 5, 6, 7; s - 8)$$

Ebouler ailleurs pour parking.

$$(4, 5, 6, 6, 0, 0, 1, 2; s - 4)$$

Trier pour représenter l'orbite.

$$(0, 0, 1, 2, 4, 5, 6, 6; s - 4)$$



Une interprétation géométrique d'une itération.

Configuration parking maximale.

$$(0, 1, 2, 3, 4, 5, 6, 7; s) \geq$$

Configuration parking initiale.

$$(0, 0, 1, 1, 4, 4, 5, 6; s)$$

Enlever un grain sur le premier 0.

$$(-1, 0, 1, 1, 4, 4, 5, 6; s)$$

Ebouler le puits pour positiver.

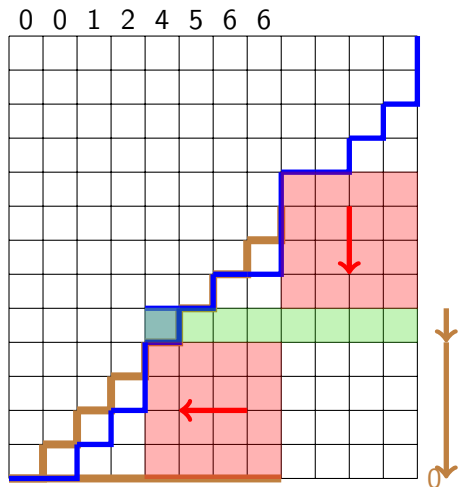
$$(0, 1, 2, 2, 5, 5, 6, 7; s - 8)$$

Ebouler ailleurs pour parking.

$$(4, 5, 6, 6, 0, 0, 1, 2; s - 4)$$

Trier pour représenter l'orbite.

$$(0, 0, 1, 2, 4, 5, 6, 6; s - 4)$$



Une interprétation géométrique d'une itération.

Configuration parking maximale.

$$(0, 1, 2, 3, 4, 5, 6, 7; s) \geq$$

Configuration parking initiale.

$$(0, 0, 1, 1, 4, 4, 5, 6; s)$$

Enlever un grain sur le premier 0.

$$(-1, 0, 1, 1, 4, 4, 5, 6; s)$$

Ebouler le puits pour positiver.

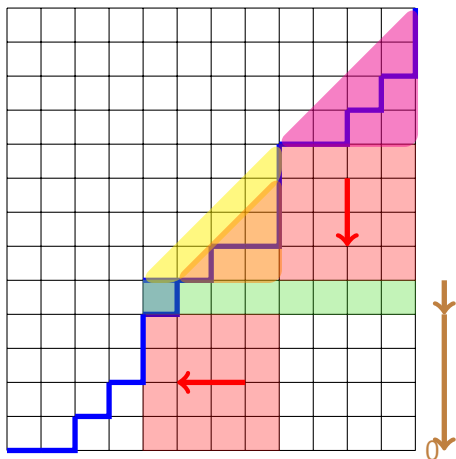
$$(0, 1, 2, 2, 5, 5, 6, 7; s - 8)$$

Ebouler ailleurs pour parking.

$$(4, 5, 6, 6, 0, 0, 1, 2; s - 4)$$

Trier pour représenter l'orbite.

$$(0, 0, 1, 2, 4, 5, 6, 6; s - 4)$$



Une interprétation géométrique d'une itération.

Configuration parking maximale.

$$(0, 1, 2, 3, 4, 5, 6, 7; s) \geq$$

Configuration parking initiale.

$$(0, 0, 1, 1, 4, 4, 5, 6; s)$$

Enlever un grain sur le premier 0.

$$(-1, 0, 1, 1, 4, 4, 5, 6; s)$$

Ebouler le puits pour positiver.

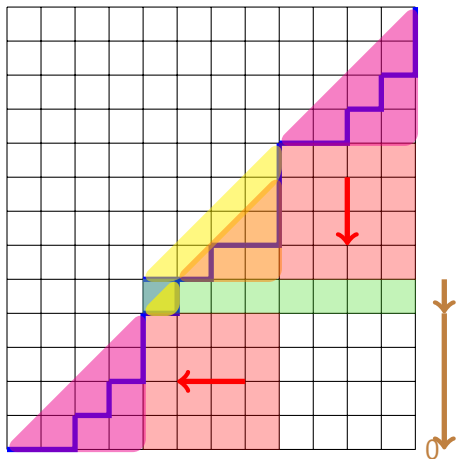
$$(0, 1, 2, 2, 5, 5, 6, 7; s - 8)$$

Ebouler ailleurs pour parking.

$$(4, 5, 6, 6, 0, 0, 1, 2; s - 4)$$

Trier pour représenter l'orbite.

$$(0, 0, 1, 2, 4, 5, 6, 6; s - 4)$$



$$\text{match } d \text{ with } | \quad ad'bd'' \longrightarrow d''abd'$$

Une interprétation géométrique d'une itération.

Configuration parking maximale.

$$(0, 1, 2, 3, 4, 5, 6, 7; s) \geq$$

Configuration parking initiale.

$$(0, 0, 1, 1, 4, 4, 5, 6; s)$$

Enlever un grain sur le premier 0.

$$(-1, 0, 1, 1, 4, 4, 5, 6; s)$$

Ebouler le puits pour positiver.

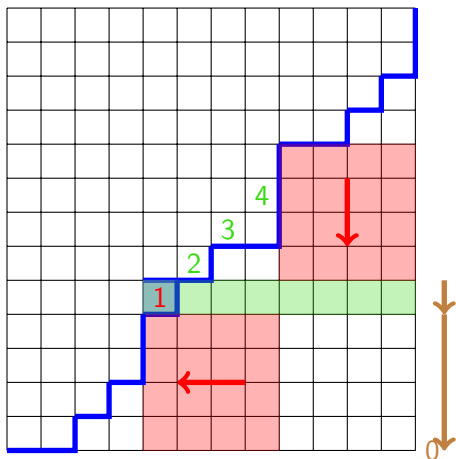
$$(0, 1, 2, 2, 5, 5, 6, 7; s - 8)$$

Ebouler ailleurs pour parking.

$$(4, 5, 6, 6, 0, 0, 1, 2; s - 4)$$

Trier pour représenter l'orbite.

$$(0, 0, 1, 2, 4, 5, 6, 6; s - 4)$$



Une interprétation géométrique d'une itération.

Configuration parking maximale.

$$(0, 1, 2, 3, 4, 5, 6, 7; s) \geq$$

Configuration parking initiale.

$$(0, 0, 1, 1, 4, 4, 5, 6; s)$$

Enlever un grain sur le premier 0.

$$(-1, 0, 1, 1, 4, 4, 5, 6; s)$$

Ebouler le puits pour positiver.

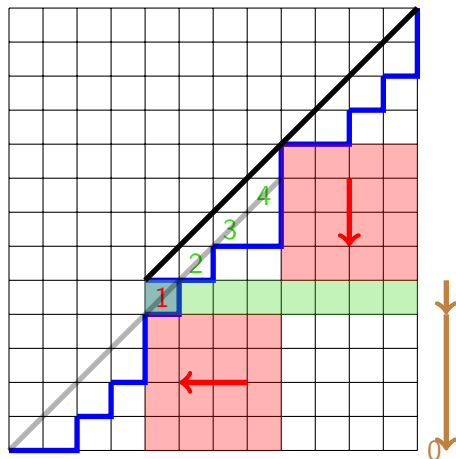
$$(0, 1, 2, 2, 5, 5, 6, 7; s - 8)$$

Ebouler ailleurs pour parking.

$$(4, 5, 6, 6, 0, 0, 1, 2; s - 4)$$

Trier pour représenter l'orbite.

$$(0, 0, 1, 2, 4, 5, 6, 6; s - 4)$$



3/3 Étude énumérative.

Distributions de la bistatistique (*degree, rank*).

$$\mathbb{K}_n(d, r) = \sum_{u \text{ parking}} d^{\text{degree}(u)} r^{\text{rank}(u)}.$$

$\{\text{configurations parking de } K_n\} \simeq \{\text{arbres couvrants de } K_n\} \times \mathbb{Z}$

Distributions de la bistatistique (*degree, rank*).

$$\mathbb{K}_n(d, r) = \sum_{u \text{ parking}} d^{\text{degree}(u)} r^{\text{rank}(u)}.$$

$\{\text{configurations parking de } K_n\} \simeq \{\text{arbres couvrants de } K_n\} \times \mathbb{Z}$

L'énumération des configuration qui ne sont pas effectives satisfait

$$[r^{-1}] \mathbb{K}_n(d, r) = \frac{d^{-1}}{1 - d^{-1}} \mathbb{T}_n(i : 1, e : d)$$

où $\mathbb{T}_n(i, e)$ est le polynôme de Tutte de K_n .

Cette égalité est valable pour tout graphe G .

Distributions de la bistatistique (*degree, rank*).

$$\mathbb{K}_n(d, r) = \sum_{u \text{ parking}} d^{\text{degree}(u)} r^{\text{rank}(u)}.$$

$\{\text{configurations parking de } K_n\} \simeq \{\text{arbres couvrants de } K_n\} \times \mathbb{Z}$

L'énumération des configuration qui ne sont pas effectives satisfait

$$[r^{-1}]\mathbb{K}_n(d, r) = \frac{d^{-1}}{1 - d^{-1}} \mathbb{T}_n(i : 1, e : d)$$

où $\mathbb{T}_n(i, e)$ est le polynôme de Tutte de K_n .

Cette égalité est valable pour tout graphe G .

Puisque $\text{rank}(\sigma.u) = \text{rank}(u)$ on considère aussi

$$K_n(d, r) = \sum_{u \text{ parking triée}} d^{\text{degree}(u)} r^{\text{rank}(u)}.$$

$\{\text{parking triées de } K_n\} \simeq \{\text{chemins de Dyck de taille } n - 1\} \times \mathbb{Z}$

Symétrie héritée d'un théorème de Baker et Norine.

r^6																	14
r^5														1	14		
r^4										2	13						
r^3									4	12							
r^2							1	2	7	10							
r^1						2	3	8	7								
r^0		1	2	4	7	8	9	4									
r^{-1}	14	14	13	12	10	7	4	1									
	d^{-2}	d^{-1}	d^0	d^1	d^2	d^3	d^4	d^5	d^6	d^7	d^8	d^9	d^{10}	d^{11}	d^{12}		

y^6	14	1											
y^5	13	2											
y^4	12	4											
y^3	10	7	2	1									
y^2	7	8	3	2									
y^1	4	9	8	7	4	2	1						
y^0	1	4	7	10	12	13	14						
	x^0	x^1	x^2	x^3	x^4	x^5	x^6						

Symétrie héritée d'un théorème de Baker et Norine.

r^6																	14
r^5																1	14
r^4													2	13			
r^3													4	12			
r^2									1	2	7	10					
r^1									2	3	8	7					
r^0			1	2	4	7	8	9	4								
r^{-1}	14	14	13	12	10	7	4	1									
	d^{-2}	d^{-1}	d^0	d^1	d^2	d^3	d^4	d^5	d^6	d^7	d^8	d^9	d^{10}	d^{11}	d^{12}		

y^6	14	1															
y^5	13	2															
y^4	12	4															
y^3	10	7	2	1													
y^2	7	8	3	2													
y^1	4	9	8	7	4	2	1										
y^0	1	4	7	10	12	13	14										
	x^0	x^1	x^2	x^3	x^4	x^5	x^6										

Les configurations $u_{\max} = (0, 0, \dots, 0; s)$ et $u_{\min} = (0, 1, 2, \dots, n-1; s)$ définissent le support.

Symétrie héritée d'un théorème de Baker et Norine.

r^6																			14	
r^5																	1	14		
r^4																		2	13	
r^3																		4	12	
r^2																	1	2	7	10
r^1																	2	3	8	7
r^0																				
r^{-1}	14	14	13	12	10	7	4	1												
	d^{-2}	d^{-1}	d^0	d^1	d^2	d^3	d^4	d^5	d^6	d^7	d^8	d^9	d^{10}	d^{11}	d^{12}					

y^6	14	1																		
y^5	13	2																		
y^4	12	4																		
y^3	10	7	2	1																
y^2	7	8	3	2																
y^1	4	9	8	7	4	2	1													
y^0	1	4	7	10	12	13	14													
	x^0	x^1	x^2	x^3	x^4	x^5	x^6													

Les configurations $u_{\max} = (0, 0, \dots, 0; s)$ et

$u_{\min} = (0, 1, 2, \dots, n - 1; s)$ définissent le support.

Pour la configuration fixe $K = (d_i - 2)_i$ et toute configuration u :

$$\text{rank}(u) - \text{rank}(K - u) = \text{degree}(u) - (m - n).$$

Symétrie héritée d'un théorème de Baker et Norine.

r^6															14
r^5														1	14
r^4													2	13	
r^3												4	12		
r^2								1	2	7	10				
r^1								2	3	8	7				
r^0			1	2	4	7	8	9	4						
r^{-1}	14	14	13	12	10	7	4	1							
	d^{-2}	d^{-1}	d^0	d^1	d^2	d^3	d^4	d^5	d^6	d^7	d^8	d^9	d^{10}	d^{11}	d^{12}

y^6	14	1													
y^5	13	2													
y^4	12	4													
y^3	10	7	2	1											
y^2	7	8	3	2											
y^1	4	9	8	7	4	2	1								
y^0	1	4	7	10	12	13	14								
	x^0	x^1	x^2	x^3	x^4	x^5	x^6								

Les configurations $u_{\max} = (0, 0, \dots, 0; s)$ et

$u_{\min} = (0, 1, 2, \dots, n - 1; s)$ définissent le support.

Pour la configuration fixe $K = (d_i - 2)_i$ et toute configuration u :

$$\text{rank}(u) - \text{rank}(K - u) = \text{degree}(u) - (m - n).$$

Soit $xpara(u) = \text{rank}(u) - \text{degree}(u)$ et $ypara(u) = \text{rank}(u) + 1$

$K_n(x, y) = \sum_u \text{parking triée } x^{xpara(u)} y^{ypara(u)}$ satisfait

$$K_n(x, y) = x^{\binom{n}{2}} y K_n(d : 1/x, r : xy) \text{ et } K_n(x, y) = K_n(y, x).$$

Reformulation de $K_n(x, y)$ en terme de chemins binomiaux

Pour interpréter $K_n(x, y)$ on utilise l'ensemble de mots binomiaux

$$W_n = \{w \mid w = w'' . EE, w'' \in N^n \sqcup E^{n-2}\}$$

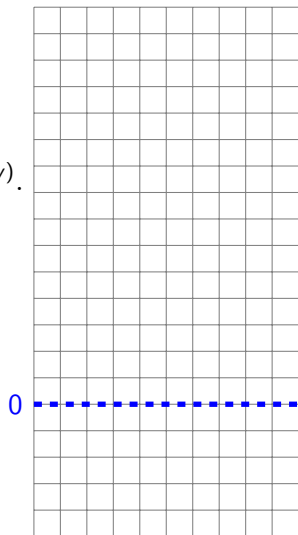
$$K_n(x, y) = \sum_{(Q, w) \in \mathbb{Z} \times W_n} x^{\text{para}(Q, w)} y^{\text{para}(Q, w)}.$$

Le poids de la lettre E dans $(Q, p.E.s)$ est

$$\begin{cases} y^h & \text{si } h > 0 \\ 1 & \text{si } h = 0 \\ x^{-h} & \text{si } h < 0 \end{cases}$$

où $h = Q - 1 + |p|_N - |p|_E$.

Le poids de (Q, w) est le produit des poids des lettres E sauf la dernière.



Reformulation de $K_n(x, y)$ en terme de chemins binomiaux

Pour interpréter $K_n(x, y)$ on utilise l'ensemble de mots binomiaux

$$W_n = \{w \mid w = w'' \cdot EE, w'' \in N^n \sqcup E^{n-2}\}$$

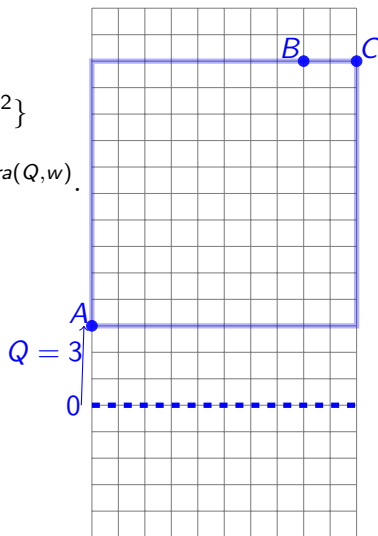
$$K_n(x, y) = \sum_{(Q, w) \in \mathbb{Z} \times W_n} x^{\text{para}(Q, w)} y^{\text{para}(Q, w)}.$$

Le poids de la lettre E dans $(Q, p.E.s)$ est

$$\begin{cases} y^h & \text{si } h > 0 \\ 1 & \text{si } h = 0 \\ x^{-h} & \text{si } h < 0 \end{cases}$$

où $h = Q - 1 + |p|_N - |p|_E$.

Le poids de (Q, w) est le produit des poids des lettres E sauf la dernière.



Reformulation de $K_n(x, y)$ en terme de chemins binomiaux

Pour interpréter $K_n(x, y)$ on utilise l'ensemble de mots binomiaux

$$W_n = \{w \mid w = w'' \cdot EE, w'' \in N^n \sqcup E^{n-2}\}$$

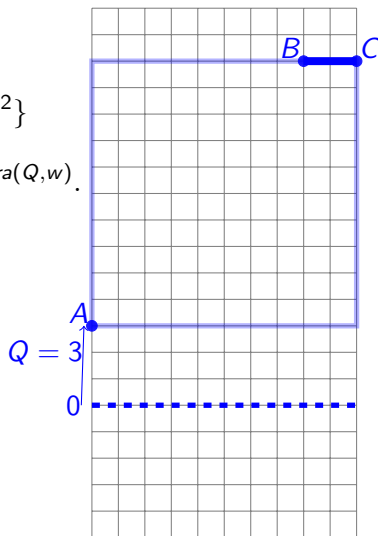
$$K_n(x, y) = \sum_{(Q, w) \in \mathbb{Z} \times W_n} x^{\text{para}(Q, w)} y^{\text{para}(Q, w)}.$$

Le poids de la lettre E dans $(Q, p.E.s)$ est

$$\begin{cases} y^h & \text{si } h > 0 \\ 1 & \text{si } h = 0 \\ x^{-h} & \text{si } h < 0 \end{cases}$$

où $h = Q - 1 + |p|_N - |p|_E$.

Le poids de (Q, w) est le produit des poids des lettres E sauf la dernière.



Reformulation de $K_n(x, y)$ en terme de chemins binomiaux

Pour interpréter $K_n(x, y)$ on utilise l'ensemble de mots binomiaux

$$W_n = \{w \mid w = w'' . EE, w'' \in N^n \sqcup E^{n-2}\}$$

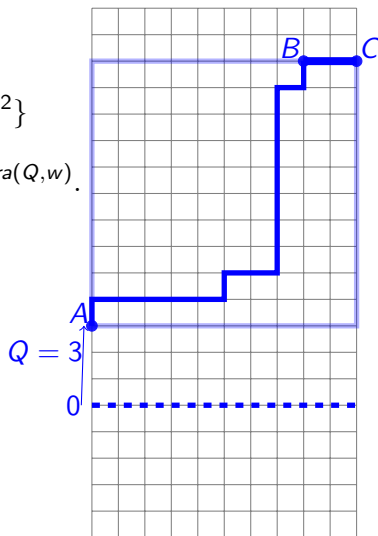
$$K_n(x, y) = \sum_{(Q, w) \in \mathbb{Z} \times W_n} x^{\text{para}(Q, w)} y^{\text{para}(Q, w)}.$$

Le poids de la lettre E dans $(Q, p.E.s)$ est

$$\begin{cases} y^h & \text{si } h > 0 \\ 1 & \text{si } h = 0 \\ x^{-h} & \text{si } h < 0 \end{cases}$$

où $h = Q - 1 + |p|_N - |p|_E$.

Le poids de (Q, w) est le produit des poids des lettres E sauf la dernière.



Reformulation de $K_n(x, y)$ en terme de chemins binomiaux

Pour interpréter $K_n(x, y)$ on utilise l'ensemble de mots binomiaux

$$W_n = \{w \mid w = w'' . EE, w'' \in N^n \sqcup E^{n-2}\}$$

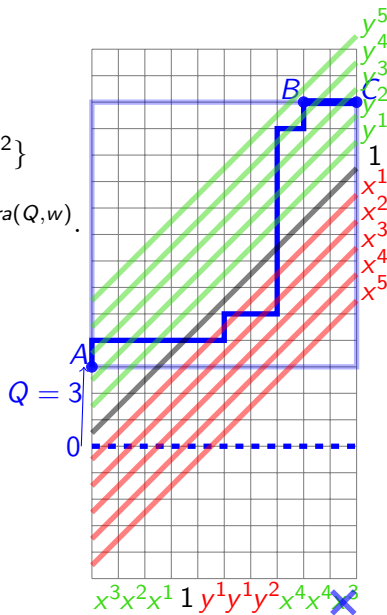
$$K_n(x, y) = \sum_{(Q, w) \in \mathbb{Z} \times W_n} x^{\text{para}(Q, w)} y^{\text{para}(Q, w)}$$

Le poids de la lettre E dans $(Q, p.E.s)$ est

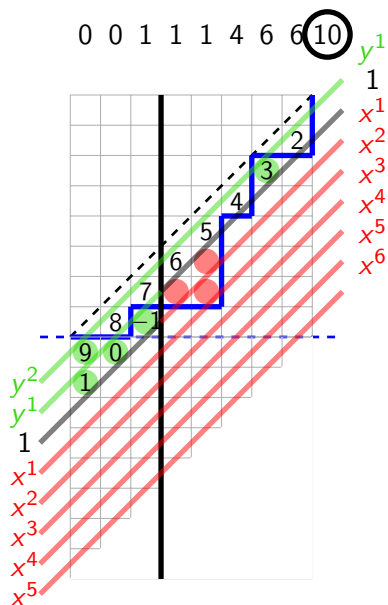
$$\begin{cases} y^h & \text{si } h > 0 \\ 1 & \text{si } h = 0 \\ x^{-h} & \text{si } h < 0 \end{cases}$$

où $h = Q - 1 + |p|_N - |p|_E$.

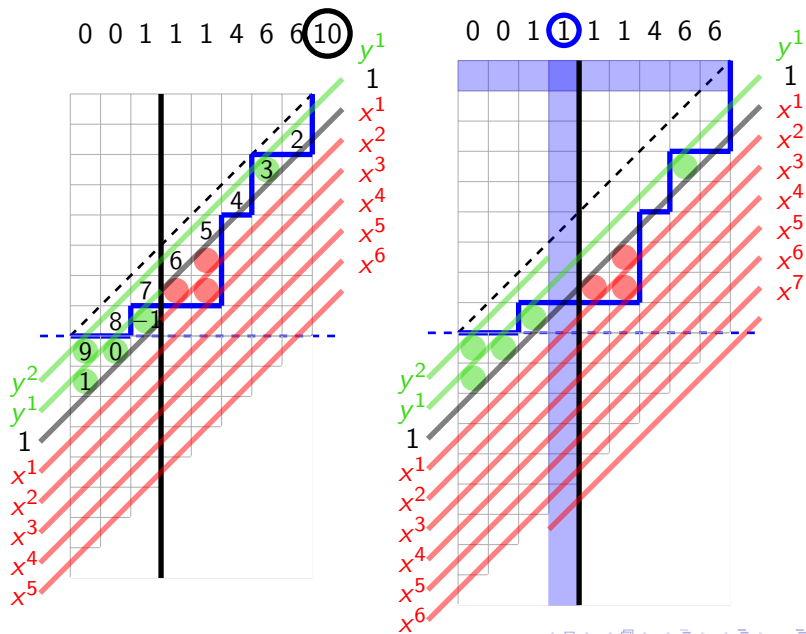
Le poids de (Q, w) est le produit des poids des lettres E sauf la dernière.



Preuve de la reformulation en terme de chemins binomiaux



Preuve de la reformulation en terme de chemins binomiaux



Une expression pour $K(x, y; z)$

$$K(x, y; z) = \sum_{n \geq 1} K_n(x, y) z^n =$$

$$\frac{1}{(1-z)(1-x)(1-y)} \sum_{0 \leq i, j \leq 2} a_{i,j}(x, y; z) c(x; xz)^i c(y; yz)^j$$

$$(c(y; yz)z + 1)(c(x; xz)z + 1)(c(x; xz)c(y; yz)z - 1)$$

où les $(a_{i,j}(x, y; z))_{i,j}$ sont des polynômes explicites,

$$c(q; z) = \frac{T(q; qz)}{T(q; z)}$$

est un q -analogue des nombres de Catalan comptant l'aire sous les chemins de Dyck. où

$$T(q; z) = \sum_{n \geq 0} \frac{q^{n(n+1)} (-z)^n}{(q; q)_n}$$

où

$$(q; a)_n = (1-a)(1-qa) \dots (1-q^{n-1}a).$$

Identité intermédiaire principale

On rappelle que $T(q; z)$ est défini par

$$\begin{cases} T(q; z) = T(q; qz) - zT(q; q^2z) \\ T(q; 1) = 1 \end{cases}$$

On pose pour tout $i \geq 0$

$$Q_i(z) = \frac{q^i z T(q; q^{i+2}z)}{T(q; q^i z)}$$

sachant que

$$c(q; z) = \frac{T(q; qz)}{T(q; z)}.$$

$$Q_1 + \sum_{i \geq 1} (Q_{i+2} + 1) \prod_{j=i}^0 Q_j = \frac{1}{1-q} (c(q; z) - c(q; qz)) - Q_0(1 + Q_2)$$

Remarques

Remarques

- ▶ Ces distributions (*degree*, *rank*) ont peut-être déjà été étudiées.

Remarques

- ▶ Ces distributions (*degree*, *rank*) ont peut-être déjà été étudiées.
- ▶ D'autres échelles pour décrire cet algorithme:

Remarques

- ▶ Ces distributions (*degree*, *rank*) ont peut-être déjà été étudiées.
- ▶ D'autres échelles pour décrire cet algorithme:

Remarques

- ▶ Ces distributions (*degree*, *rank*) ont peut-être déjà été étudiées.
- ▶ D'autres échelles pour décrire cet algorithme:
 - ▶ Mots de Dyck : $ad'bd'' \rightarrow d''abd'$
Une conjugaison par une involution τ préservant le *div*, associe cette réécriture à un flip du dernier maximum.
L'image par τ de *area* est *prerank* qui satisfait parfois

$$\text{prerank}(u_{\text{begin}}) - \text{prerank}(u_{\text{end}}) = \text{rank}(u_{\text{begin}}).$$

Remarques

- ▶ Ces distributions (*degree*, *rank*) ont peut-être déjà été étudiées.
- ▶ D'autres échelles pour décrire cet algorithme:
 - ▶ Mots de Dyck : $ad'bd'' \rightarrow d''abd'$
Une conjugaison par une involution τ préservant le *div*, associe cette réécriture à un flip du dernier maximum. L'image par τ de *area* est *prerank* qui satisfait parfois

$$\text{prerank}(u_{\text{begin}}) - \text{prerank}(u_{\text{end}}) = \text{rank}(u_{\text{begin}}).$$

- ▶ Matrice de transfert "circulante".