

# Rank of configurations on a graph.

Robert Cori,

LIGM, 7 Mars, 2017

# Graphs

## Definition

A graph is given by a set  $X = \{x_1, x_2, \dots, x_n\}$  of vertices and a set  $E$  of  $m$  edges, consisting of pairs of vertices.

All graphs considered in this talk are connected graphs.

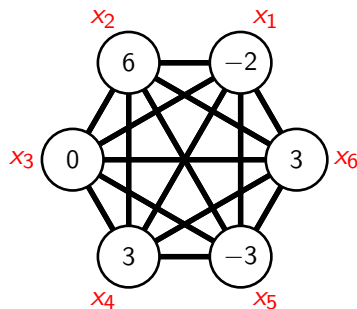
We consider often here the **complete graph**  $K_n$  having all the possible  $m = \binom{n}{2}$  edges, between vertices.

And the **wheel graph**  $W_k$  consisting of a cycle of length  $k$  and a vertex connected by an edge to all the vertices of this cycle ( $n = k + 1, m = 2k$ ).

# Configurations

## Definition

- ▶ A configuration is a mapping from  $X$  into the ring  $\mathbb{Z}$  of integers.
- ▶

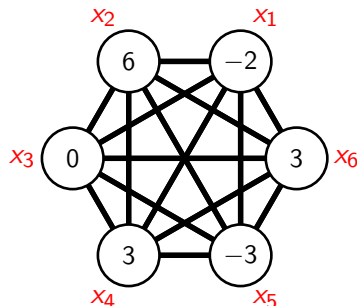


Configuration:  $u = (u_i)_{i=1,\dots,n}$   
ex:  $u = (-2, 6, 0, 3, -3, 3)$ .

# Configurations

## Definition

- ▶ A configuration is a mapping from  $X$  into the ring  $\mathbb{Z}$  of integers.
- ▶ The degree of a configuration is the algebraic sum of the values attributed to the vertices



Configuration:  $u = (u_i)_{i=1,\dots,n}$   
ex:  $u = (-2, 6, 0, 3, -3, 3)$ .

$$\deg(u) = \sum_{i=1}^n u_i = 7$$

# Topplings

In the toppling of vertex  $x_i$

- ▶ Each neighbor of  $x_i$  receives 1 from  $x_i$ .
- ▶ The value of the configuration on  $x_i$  decreases by the number of neighbors of  $x_i$

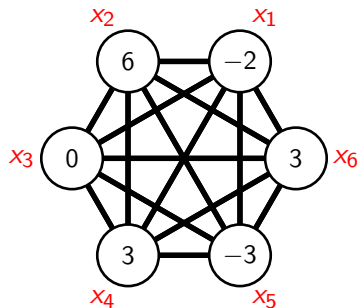
▶

$$u \xrightarrow{i} u'$$

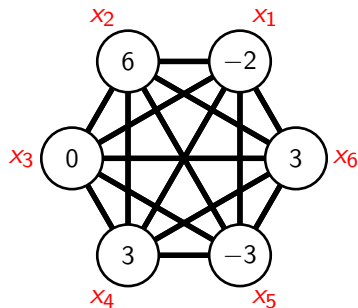
$$\begin{cases} u'_i = u_i - d_i \\ u'_j = u_j + e_{i,j} & \text{if } i \neq j \end{cases}$$

- ▶ Where  $e_{i,j}$  is the number of edges between  $x_i$  and  $x_j$ . And  $d_i = \sum_{j=1}^n e_{i,j}$ .
- ▶ The degree of the configuration is invariant.

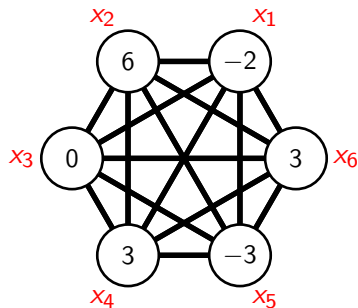
## Example of topplings



## Example of topplings

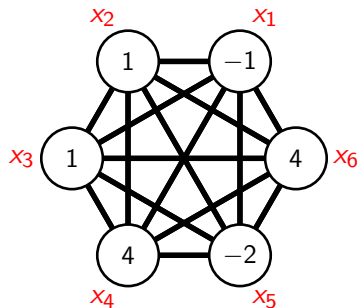


## Example of topplings

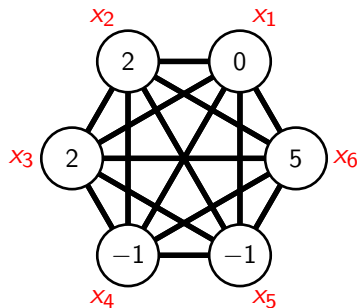




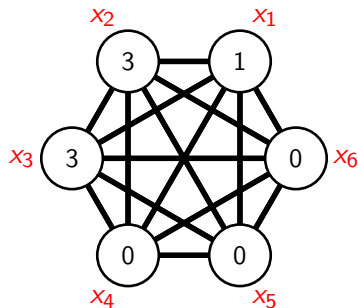
## Example of topplings



## Example of topplings



## Example of topplings



# Positive/Effective configurations

## Definition

A configuration  $u$  is **effective** if  $u_i \geq 0$  for all  $1 \leq i \leq n$ , it is a **sandpile configuration** if  $u_i \geq 0$  for all  $1 \leq i < n$ .

# Positive/Effective configurations

## Definition

A configuration  $u$  is **effective** if  $u_i \geq 0$  for all  $1 \leq i \leq n$ , it is a **sandpile configuration** if  $u_i \geq 0$  for all  $1 \leq i < n$ .

## Remark

*For any configuration  $u$  there exists a sequence of topplings leading to a sandpile configuration.*

# $\Delta$ -Effective configurations

## Definition

A configuration  $u$  is  $\Delta$ -**effective** if there exists a sequence of topplings leading from  $u$  to an effective configuration.

## Question:

Given the configuration  $u$  determine if it is  $\Delta$ -effective.

## Some linear algebra: the Laplacian matrix $\Delta$

The rows  $\Delta_i$  of the Laplacian matrix are given by:

$$\Delta_i = (e_{i,1}, \quad \cdots, \quad \underset{\substack{\uparrow \\ i}}{-d_i}, \quad \cdots, \quad e_{i,n})$$

A toppling :

$$u \xrightarrow{i} u'$$

is such that

$$u' = u + \Delta_i$$

# $\Delta$ -equivalence

Equivalence:  $u \equiv_{\Delta} v \iff u = v + \sum_{i=1}^n a_i \Delta_i$

## Question:

Given the configuration  $u$  determine if there is an effective  $v$  such that  $u \equiv_{\Delta} v$ .



# Classes of the $\Delta$ -equivalence

Main results:

- ▶ For the configurations of given degree the number of equivalence classes is equal to the number of spanning trees of the graph.
- ▶ One can test  $\Delta$ -equivalence by an algorithm in polynomial time.

# Stable configurations

## Definition

A sandpile configuration  $u$  is a stable configuration if  $u$  satisfies:

$$0 \leq u_i < d_i \text{ for all } i < n$$

# Stable configurations

## Definition

A sandpile configuration  $u$  is a stable configuration if  $u$  satisfies:

$$0 \leq u_i < d_i \text{ for all } i < n$$

## Remark

*For any configuration  $u$  there exists at least one stable configuration  $v$  such that*

$$u \equiv_{\Delta} v$$

# Superstable configurations

## Definition

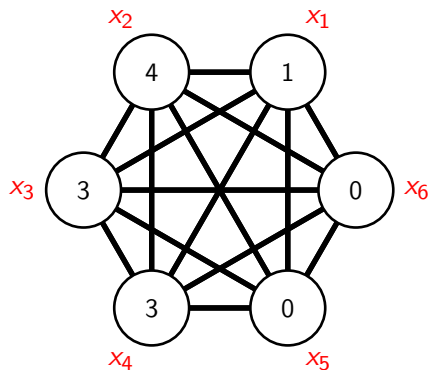
A stable configuration  $u$  is a superstable configuration if the toppling of any subset of vertices  $\{x_{i_1}, x_{i_2}, \dots, x_{i_k}\}$  which is a subset of  $\{x_1, x_2, \dots, x_{n-1}\}$  gives a vertex with negative value.

More precisely:  $\forall I \subset \{1, 2, \dots, n-1\}, I \neq \emptyset$ :

$$v = u + \sum_{i \in I} \Delta_i$$

is such that there exists  $j$  with  $v_j < 0$ .

# Example



# Main facts on superstable configurations

## **Proposition**(Dhar)

Any class of the toppling equivalence contains a unique superstable configuration.

# Algorithm to obtain the superstable configuration

- 1: **While**  $\exists i < n, u_i < 0$  **do**
- 2:     *Topple  $x_n$  and perform subsequent topplings*
- 3: **While**  $\exists$  an  $I$ -toppling leaving the  $u_i \geq 0$  **do**
- 4:     *Topple all the  $x_i$  for  $i$  in  $I$*

# Main Theorem for testing $\Delta$ -effectiveness

## Theorem

*(Baker, Norine) A superstable configuration is  $\Delta$ -effective if and only if  $u_n \geq 0$ .*



# Main Theorem for testing $\Delta$ -effectiveness

## Theorem

*(Baker, Norine) A superstable configuration is  $\Delta$ -effective if and only if  $u_n \geq 0$ .*

For any graph, there is a polynomial time algorithm determining if a configuration  $u$  is  $\Delta$ -effective (using J. Van Den Heuvel algorithm finding the superstable configuration).

# Rank of a configuration

## Definition

The rank  $\rho(u)$  of a configuration  $u$  is given by

$$\rho(u) = \text{Min}(\deg(f)) - 1$$

where the minimum is taken among all the positive configurations  $f$  such that  $u - f$  is not  $\Delta$ -effective.

# Rank of a configuration

## Definition

The rank  $\rho(u)$  of a configuration  $u$  is given by

$$\rho(u) = \text{Min}(\deg(f)) - 1$$

where the minimum is taken among all the positive configurations  $f$  such that  $u - f$  is not  $\Delta$ -effective.

An effective configuration  $\mu$  is a **proof for the rank**  $\rho(u)$  of  $u$  if  $u - \mu$  is not  $\Delta$ -effective and  $u - \lambda$  is  $\Delta$ -effective for any effective configuration  $\lambda$  such that  $\deg(\lambda) < \deg(\mu)$ .

Notice that if  $\lambda$  is a proof for  $\rho(u)$  then  $\rho(u) = \deg(\lambda) - 1$ .

# Determining the rank by exhaustive search

- 1: **For**  $k = 1, 2 \dots n$  **do**  
    **For all** compositions  $\lambda$  of  $k$  **do**  
        **If**  $u - \lambda$  is not  $\Delta$ -effective  $\rho(u) = k - 1$

## Simple facts on the rank

- ▶ If two configurations  $u$  and  $v$  are such that  $u_i \leq v_i$  for all  $i$  then  $\rho(u) \leq \rho(v)$ .

# Simple facts on the rank

- ▶ If two configurations  $u$  and  $v$  are such that  $u_i \leq v_i$  for all  $i$  then  $\rho(u) \leq \rho(v)$ .
- ▶ Moreover if  $u_i = v_i - \varepsilon^{(i)}$  then  $\rho(v) - 1 \leq \rho(u) \leq \rho(v)$

## Simple facts on the rank

- ▶ If two configurations  $u$  and  $v$  are such that  $u_i \leq v_i$  for all  $i$  then  $\rho(u) \leq \rho(v)$ .
- ▶ Moreover if  $u_i = v_i - \varepsilon^{(i)}$  then  $\rho(v) - 1 \leq \rho(u) \leq \rho(v)$
- ▶ If  $\lambda$  is a proof for the rank of  $u$ , and  $\mu$  is an effective configuration such that  $\mu_i \leq \lambda_i$  for  $i = 1, 2, \dots, n$  then  $\lambda - \mu$  is a proof for the rank of  $v = u - \mu$

# More facts on the rank

- ▶ Let  $u$  and  $v$  be two  $\Delta$ -effective configurations then

$$\rho(u + v) \geq \rho(u) + \rho(v)$$

- ▶ Moreover :

$$\rho(u + v) \leq \rho(u) + \deg(v)$$



# Baker and Norine's Theorem for graphs

M. Baker, S. Norine, *Riemann-Roch and Abel-Jacobi theory on a finite graph* (2007) *Advances in Maths*, **215**, 766-788.

## Theorem

Let  $\kappa$  be the configuration such that  $\kappa_i = d_i - 2$ , so that  $\deg(\kappa) = 2(m - n)$ . Any configuration  $u$  satisfies:

$$\rho(u) = \deg(u) - m + n + \rho(\kappa - u)$$

# Consequence of Baker and Norine's Theorem

- ▶ If the degree of  $u$  is greater than  $2(m - n)$ , then  $\rho(u) = \deg(u) - (m - n + 1)$
- ▶ For any configuration  $u$ , the rank  $\rho(u)$  satisfies

$$\rho(u) \geq \deg(u) - (m - n + 1)$$

- ▶ we will say that the configuration is of *minimal rank* if  $\rho(u) = \deg(u) - (m - n + 1)$ .
- ▶ If  $u$  has minimal rank then for any effective configuration  $\lambda$ ,  $u + \lambda$  has also minimal rank.

# The computation of the rank is an NP-complete problem

V. Kiss and L. Tóthmérész.

Chip-firing games on eulerian digraphs and NP-hardness of computing the rank of a divisor on a graph.

*Discrete Appl. Math.*, 193:48—56, 2015.

# A greedy algorithm for the rank

**Input:** A configuration  $u$  on  $G$ ,  $\hat{u}$  the superstable configuration  $\Delta$ -equivalent to  $u$ ;

- ▶ While  $\hat{u}_n \geq 0$  do
  - ▶ Find  $i \neq n$  such that  $u' = u - \varepsilon^{(i)}$  is such that  $\hat{u}'_n$  is minimal.
  - ▶  $u \leftarrow u - \varepsilon^{(i)}$ ;  $u \leftarrow \hat{u}$
  - ▶  $s \leftarrow s + 1$ ;
- ▶ od;

**Output:**  $s - 1$ .

The computation of the rank in  $K_n$  can be solved in linear time

R. C. and Y. Le Borgne.

On the computation of Baker and Norine's rank on the complete graph

*Electronic J. of Combinatorics*, **23** P1.31 (2016) .

# Main fact for the rank of a configuration on $K_n$

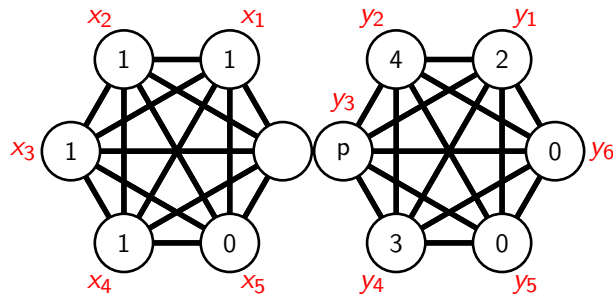
## Theorem

*Let  $u$  be a superstable configuration on  $K_n$ , then there exists  $i < n$  such that  $u_i = 0$ , and for all such  $i$ , the configuration  $v = u - \varepsilon(i)$  satisfies*

$$\rho(v) = \rho(u) - 1$$

Hence the greedy algorithm computes correctly the rank in  $K_n$ .

# Graph counterexample for the Greedy algorithm



Compare  $p = 1$  and  $p = 4$ .

## The wheel graph $W_k$

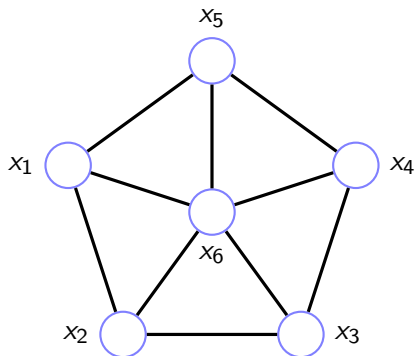


Figure: The Wheel Graph  $W_5$



# Superstable configurations on the wheel

## Proposition

*A configuration is superstable if and only if the following conditions are satisfied*

- ▶ *There exists at least one  $i < n$  such that  $u_i = 0$*
- ▶ *If  $u_i = u_j = 2$  there exists one  $p$  in the interval  $[i, j]$  such that  $u_p = 0$ .*

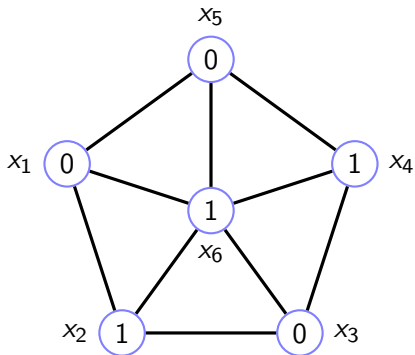


Figure: Configuration on the Wheel Graph with rank 0

Subtracting 1 from a vertex  $x_i$  such that  $u_i = 0$  may not modify the rank

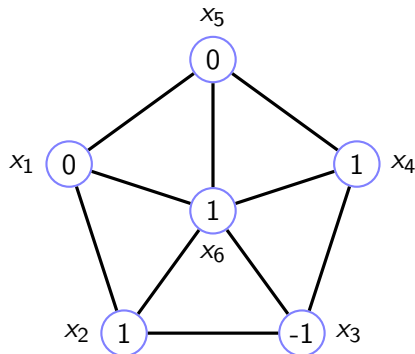


Figure: Configuration on the Wheel Graph with rank 0

Subtracting 1 from a vertex  $x_i$  such that  $u_i = 0$  may not modify the rank

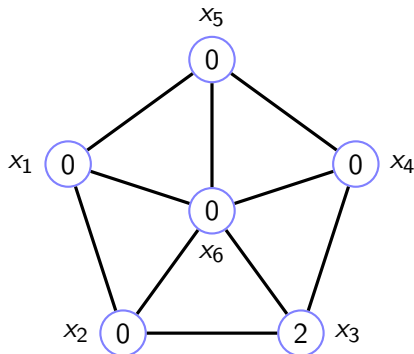


Figure: Configuration on the Wheel Graph with rank 0

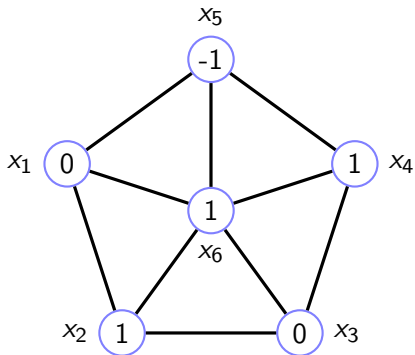


Figure: Configuration on the Wheel Graph with rank -1

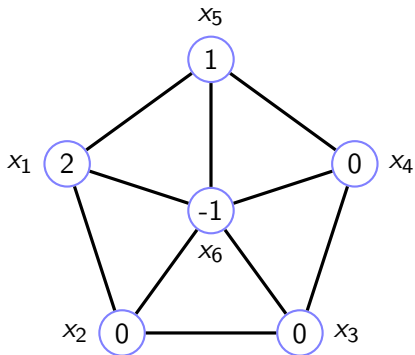


Figure: Configuration on the Wheel Graph with rank -1

# Toppling equivalence for the complete graph

## Lemma

*The configuration  $u$  is such that  $u \equiv_{\Delta} 0$  if and only if  $\deg(u) = 0$  and*

$$\forall i, j \quad u_i = u_j \pmod n$$

## Corollary

*Two configurations  $u, v$  are such that  $u \equiv_{\Delta} v$  if and only if  $\deg(u) = \deg(v)$  and*

$$\forall i, j \quad u_i - v_i = u_j - v_j \pmod n$$

# Toppling equivalence for the complete graph

To determine the parking configuration equivalent to  $u$  replace each  $u_i$  for  $i < n$  by  $u'_i = u_i \bmod n$  then set

$$u'_n := \deg(u) - \sum_{i=1}^{n-1} u'_i$$

Then perform subset topplings while the  $u_i$  with  $i < n$  are non negative.



# Superstable configurations in $K_n$

## Definition

A configuration  $u$  in  $K_n$  is superstable if and only if the sequence  $(u_1, u_2, \dots, u_{n-1})$  is a parking function.

# Superstable configurations in $K_n$

## Definition

A configuration  $u$  in  $K_n$  is superstable if and only if the sequence  $(u_1, u_2, \dots, u_{n-1})$  is a parking function.

Meaning that sorting it one obtains a sequence  $(u'_1, u'_2, \dots, u'_{n-1})$  satisfying  $0 \leq u'_i < i$  for all  $i$ .

# Dyck words

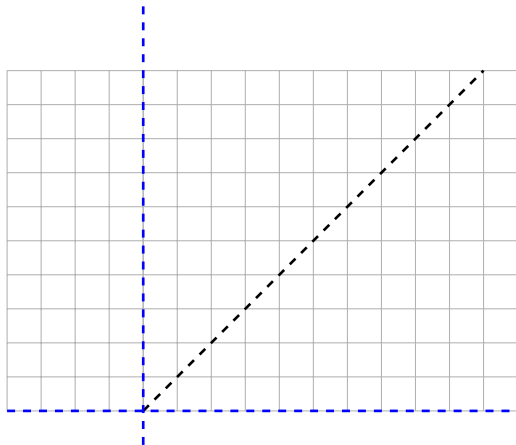
We consider the alphabet  $A = \{a, b\}$ .

- ▶  $A_n$  is the set of words with  $n$  occurrences of  $b$  and  $n - 1$  occurrences of  $a$ . The number of elements of  $A_n$  is  $\binom{2n-1}{n}$ .
- ▶ The mapping  $\delta$  is defined by:

$$\delta(f) = |f|_a - |f|_b$$

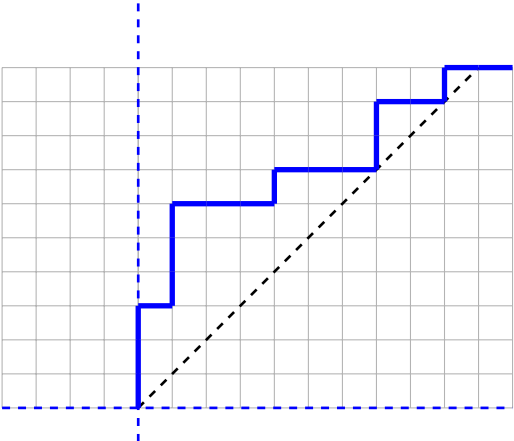
- ▶  $D_n$  is the subset of  $A_n$  consisting of the words  $f$  such that  $\delta(g) \geq 0$  for each decomposition  $f = gh$  and  $g \neq f$ .

# Drawings for words in $D_n$



*a a a b a a a b b b a b b b a a b b a b b*

## Drawings for words in $D_n$



*a a a b a a a b b b a b b b a a b b a b b*

## Word associated to a parking configuration $u$ of $K_n$

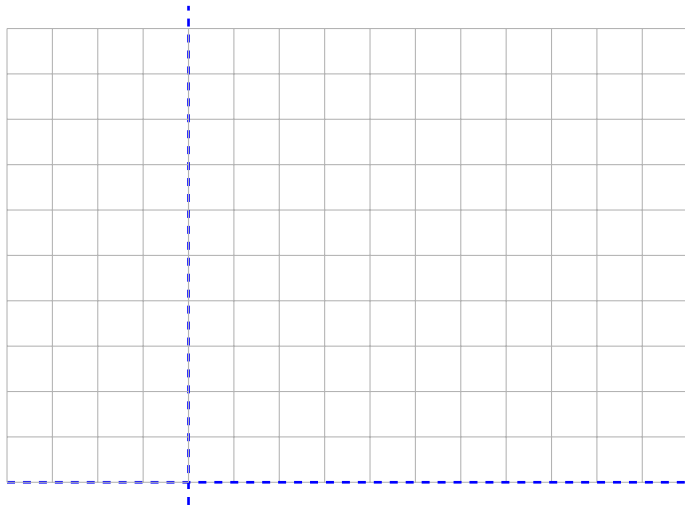
- ▶ Sort the first  $n - 1$  coefficients of  $u$  in weakly increasing order giving  $u'$ .
- ▶ The word  $f$  has  $n - 1$  occurrences of  $a$  and  $n$  occurrences of  $b$ .
- ▶ The prefix of  $f$  ending with the  $i$ -th occurrence of  $a$  contains  $u'_i$  occurrences of  $b$
- ▶ Associate also the last coefficient  $s = u_n$ .
- ▶ Example:  $u = (1, 4, 0, 0, 7, 1, 0, 9, 1, 7, 17)$ ,

$$u' = (0, 0, 0, 1, 1, 1, 4, 7, 7, 9, 5)$$

$$f = a a a b a a a b b b a b b b a a b b a b b$$

$$s = 17$$

# Numbering the cells



*a a a b a a a b b b a b b b a a b b a b b*

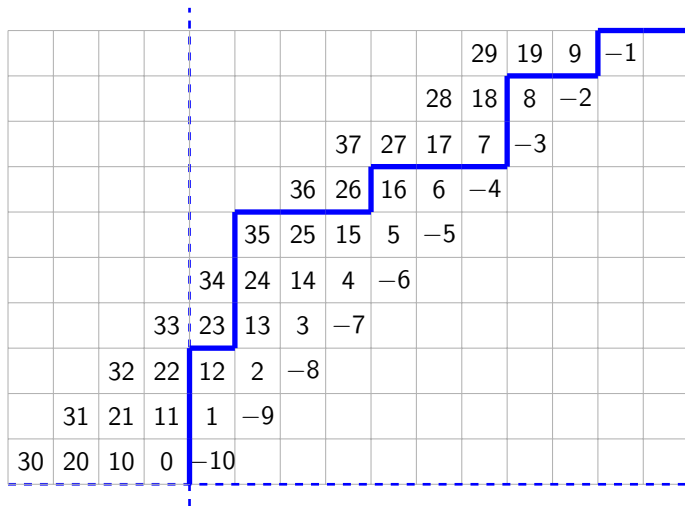
# Numbering the cells

										29	19	9	-1	
									28	18	8	-2		
							37	27	17	7	-3			
						36	26	16	6	-4				
					35	25	15	5	-5					
				34	24	14	4	-6						
			33	23	13	3	-7							
		32	22	12	2	-8								
	31	21	11	1	-9									
30	20	10	0	-10										

*a a a b a a a b b b a b b b a a b b a b b*



# Numbering the cells



*a a a b a a a b b b a b b b a a b b a b b*

## Main objects here

A word in  $D_n$  and an integer  $s \in \mathbb{Z}$ .

So that we consider the set:

$$D_n \times \mathbb{Z}$$

## Left active cells

A cell with label  $c$  is left active for  $s \in \mathbb{Z}$  and a word  $f \in D_n$ , if the following two conditions hold:

- ▶  $c \leq s$
- ▶ it lies on the left of the path determined by  $f$ .

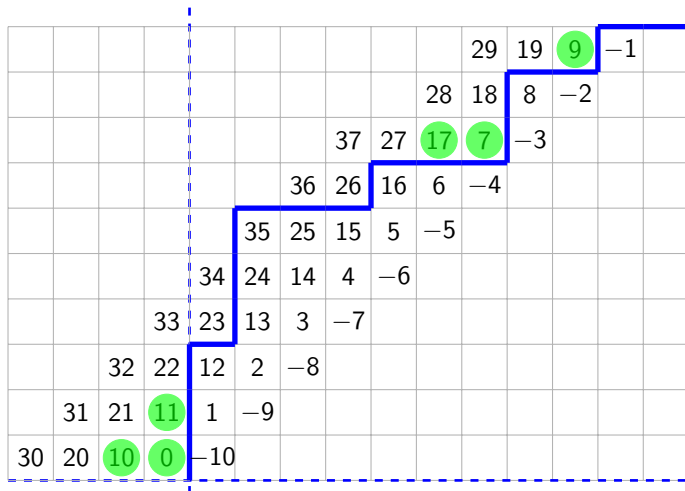
# Left active cells

A cell with label  $c$  is left active for  $s \in \mathbb{Z}$  and a word  $f \in D_n$ , if the following two conditions hold:

- ▶  $c \leq s$
- ▶ it lies on the left of the path determined by  $f$ .

The number of left active cells is denoted by  $left(f, s)$

# Left active cells



$s = 17$ ,  $\text{left}(f, 17) = 6$ ;  $\text{left}(f, 18) = 7$

# Rank and left active cells

## Theorem

*Let  $u$  be a parking configuration on  $K_n$  and  $f$  be the Dyck word associated to it, then the rank of  $u$  is given by*

$$\rho(u) = \text{left}(f, u_n) - 1$$

## Corrolary

*There is an algorithm computing the rank of a configuration in  $O(n)$ .*

# An Involution on Dyck words

- ▶ The mirror image of a word  $f = f_1 f_2 \cdots f_p$  is  $\tilde{f} = f_p f_{p-1} \cdots f_2 f_1$
- ▶ For any  $f$  in  $D_n$  define  $\Phi(f)$  as the conjugate of  $\tilde{f}$  which is an element of  $D_n$ .

## About $\Phi$

- ▶ For  $f$  in  $D_n$  let  $m = \text{maxright}(f)$  the maximal value of the label of a cell on the right of path determined by  $f$ .
- ▶ Then we have:

### Proposition

*The involution  $\Phi$  is such that:*

$$\text{maxright}(\Phi(f)) = \text{maxright}(f)$$

$$\text{left}(f, s) = \text{right}(\Phi(f), m - 1 - s)$$



# Open problems?

- ▶ Find families graphs for which the rank can be computed in polynomial time
- ▶ Approximate algorithms?
- ▶ Baker Norine's formula allowed to introduce new operations on Dyck words and Parking Functions
- ▶ better understanding of Baker Norine's formula for some other families of graphs