

# Graph operations characterizing rank-width and balanced graph expressions

Mamadou Moustapha KANTÉ Bruno COURCELLE

LaBRI, Université Bordeaux 1, CNRS.

June 21 2007  
WG 2007 (Dornburg)

# Introduction

- Graph decompositions parameterized by tree-width, clique-width and rank-width yield *Fixed Parameter Tractable* algorithms.
- $TWD(\leq k) \subset CWD(\leq f(k)) \equiv RWD(\leq g(k))$
- Tree-width and clique-width allow FPT algorithms for problems definable in Monadic Second Order logic.
- The “MSOL  $\implies$  FPT” results use descriptions of graphs by algebraic expressions (twd,cwd).
- Rank-width has good approximation algorithms:
  - ▶ “ $cwd(G) \leq k$ ” is NP-complete for given  $(G, k)$  and open if  $k$  is fixed.
  - ▶  $O(n^3)$ -time approximation algorithm for rank-width.

# Main result 1

- Clique-width and Rank-width are equivalent in the sense that

$$rwd(G) \leq cwd(G) \leq 2^{rwd(G)+1} - 1 \quad (\text{Oum-Seymour}).$$

- But the translation rank-decomposition  $\rightarrow$  clique-width expression may need  $k \rightarrow 2^{k+1}$ .

## Main Theorem 1

There exists a signature  $(R_k, C_k)$  such that

$$rwd(G) \leq k \iff G \text{ is defined by a term in } T(R_k, C_k).$$

## Second Main result

The transformation of a decomposition of width  $k$  into one of width  $f(k)$  and diameter  $O(\log(n))$ .

### Corollary

- tree-width:  $f(k) = 3k - 1$ ,
- clique-width:  $f(k) = 2^{O(k)}$ ,
- m-clique-width:  $f(k) = 2k$ ,
- rank-width:  $f(k) = 2k$ .

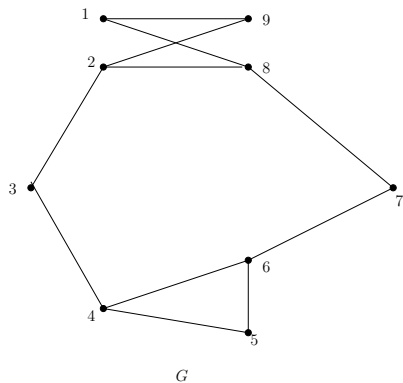
# Plan

- 1 Preliminaries
- 2 Vectorial Colorings
  - Graph Operations
  - Main theorem
- 3 General Framework for balanced decompositions
  - Balancing theorems: General ideas
  - Tools

# Rank-width

- For  $X \subseteq V_G$ , we let  $\rho_G(X) = rk(A_G[X, V_G - X])$  where  $rk$  is the matrix rank function over  $GF(2)$ .
- A layout of a graph  $G$  is a pair  $(T, \mathcal{L})$  where  $T$  is a tree of degree  $\leq 3$  and  $\mathcal{L} : V_G \rightarrow L_T$  is a bijection of the set of vertices of  $G$  onto the set of leaves of  $T$ .

# Rank-width



$$A_G[A, B] =$$

$$rk(A_G[A, B]) = 2$$

$B \backslash A$	5	6	7	8	9
1	0	0	0	1	1
2	0	0	0	1	1
3	0	0	0	0	0
4	1	1	0	0	0





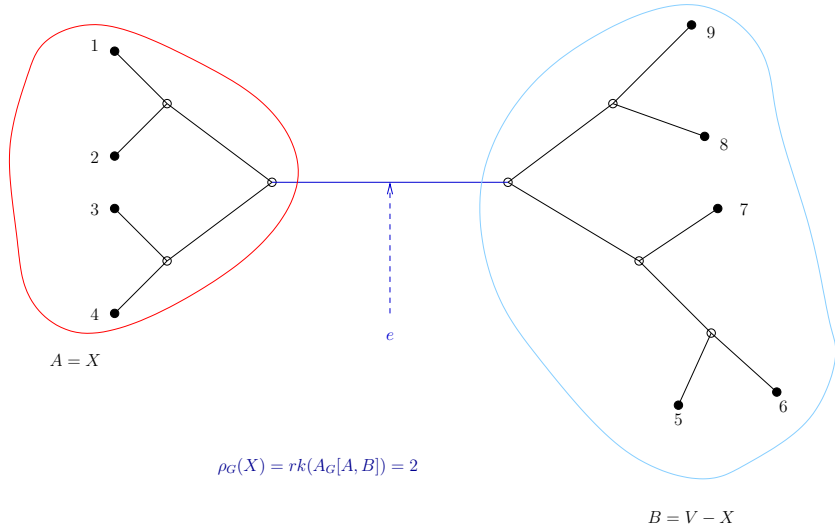
# Rank-width

- For  $X \subseteq V_G$ , we let  $\rho_G(X) = rk(A_G[X, V_G - X])$  where  $rk$  is the matrix rank function over  $GF(2)$ .
- A layout of a graph  $G$  is a pair  $(T, \mathcal{L})$  where  $T$  is a tree of degree  $\leq 3$  and  $\mathcal{L} : V_G \rightarrow L_T$  is a bijection of the set of vertices of  $G$  onto the set of leaves of  $T$ .
- Any edge  $e$  of  $T$  induces a bipartition  $(X_e, Y_e)$  of the set  $L_T$ , thus a bipartition  $(X, V - X)$  of  $V_G$ :  $wd(e) = \rho_G(X)$ .

# Rank-width

- For  $X \subseteq V_G$ , we let  $\rho_G(X) = rk(A_G[X, V_G - X])$  where  $rk$  is the matrix rank function over  $GF(2)$ .
- A layout of a graph  $G$  is a pair  $(T, \mathcal{L})$  where  $T$  is a tree of degree  $\leq 3$  and  $\mathcal{L} : V_G \rightarrow L_T$  is a bijection of the set of vertices of  $G$  onto the set of leaves of  $T$ .
- Any edge  $e$  of  $T$  induces a bipartition  $(X_e, Y_e)$  of the set  $L_T$ , thus a bipartition  $(X, V - X)$  of  $V_G$ :  $wd(e) = \rho_G(X)$ .
- $wd(T, \mathcal{L}) = \max\{wd(e) \mid e \text{ edge of } T\}$ .

## Rank-width



# Rank-width

- For  $X \subseteq V_G$ , we let  $\rho_G(X) = rk(A_G[X, V_G - X])$  where  $rk$  is the matrix rank function over  $GF(2)$ .
- A layout of a graph  $G$  is a pair  $(T, \mathcal{L})$  where  $T$  is a tree of degree  $\leq 3$  and  $\mathcal{L} : V_G \rightarrow L_T$  is a bijection of the set of vertices of  $G$  onto the set of leaves of  $T$ .
- Any edge  $e$  of  $T$  induces a bipartition  $(X_e, Y_e)$  of the set  $L_T$ , thus a bipartition  $(X, V - X)$  of  $V_G$ :  $wd(e) = \rho_G(X)$ .
- $wd(T, \mathcal{L}) = \max\{wd(e) \mid e \text{ edge of } T\}$ .
- $rwd(G) = \min\{wd(T, \mathcal{L}) \mid (T, \mathcal{L}) \text{ layout of } G\}$

# Clique-width and rank-width

## Clique-width

- $\oplus$  = disjoint union,  $\eta_{i,j}$  = edge-creation,  $\rho_{i \rightarrow j}$  = relabeling.
- $F_k = \{\oplus, \eta_{i,j}, \rho_{i \rightarrow j} \mid i, j \in [k], i \neq j\}$  and  $C_k = \{\mathbf{i} \mid i \in [k]\}$
- $cwd(G) = \min\{k \mid G = val(t) \wedge t \in T(F_k, C_k)\}$

## Clique-width and Rank-width (Oum and Seymour)

$$rwd(G) \leq cwd(G) \leq 2^{rwd(G)+1} - 1.$$

# Properties of rank-width

- For fixed  $k$ , Oum has given an  $O(n^3)$ -time “approximation” algorithm that for a graph  $G$  either outputs a layout of width at most  $3k - 1$  or outputs that  $G$  has rank-width at least  $k + 1$  (advantage over clique-width).
- But we need to transform the layout into a clique-width expression to solve FPT-problems (a disadvantage if it is important that the hidden constants stay as small as possible).
- This is why characterization of rank-width by graph operations is interesting and also from a fundamental point of view.

# Plan

- 1 Preliminaries
- 2 **Vectorial Colorings**
  - **Graph Operations**
  - Main theorem
- 3 General Framework for balanced decompositions
  - Balancing theorems: General ideas
  - Tools

# Vectorial colorings

- Let  $k \in \mathbb{N}$ ,  $\mathbb{B} = \{0, 1\}$ . A  $\mathbb{B}^k$ -coloring of a graph  $G$  is a mapping  $\gamma: V_G \rightarrow \mathbb{B}^k$ .
- $x \in V_G$  has color  $i$  (among others) iff  $\gamma(x)[i]$  (the  $i$ -th component of  $\gamma(x)$ ) is 1.
- Let  $\Gamma_G$  be the  $(V_G \times k)$ -matrix, called the **color-matrix**, the row vectors of which are the vectors  $\gamma_G(x) \in \mathbb{B}^k$  for  $x \in V_G$ .
- Let us now define the operations ...



# Linear recolorings

## Linear recolorings of $\mathbb{B}^k$ -colored graphs

- Let  $G$  be a  $\mathbb{B}^k$ -colored graph.
- Let  $N$  be a  $(k \times \ell)$ -matrix.
- $H = \text{Recol}_N(G) \iff \gamma_H(x) = \gamma_G(x) \cdot N$ . Then
  - ▶  $\Gamma_H = \Gamma_G \cdot N$ .

## Remark

- $\text{Recol}_N(\text{Recol}_{N'}(G)) = \text{Recol}_{N' \cdot N}(G)$ .
- $2^k$  different colors are handled efficiently via  $(k \times \ell)$ -matrices.

# Edge creation

## Bilinear graph products

For  $G$ ,  $\mathbb{B}^k$ -colored,  $H$ ,  $\mathbb{B}^\ell$ -colored, and for  $(k \times \ell)$ -matrices  $M, N, P$  the graph  $K = G \otimes_{M,N,P} H$  is constructed in 3 steps.

- disjoint union of  $G$  and  $H$ .
- $x \in V_G$  and  $y \in V_H$  are linked iff  $\gamma_G(x) \cdot M \cdot \gamma_H(y)^T = 1$ .
- Independent recolorings of  $G$  and  $H$  by  $Recol_N$  and  $Recol_P$ , i.e.,

$$\Gamma_K(z) = \begin{cases} \Gamma_G(z) \cdot N & \text{if } z \in V_G \\ \Gamma_H(z) \cdot P & \text{if } z \in V_H. \end{cases}$$

## Remark

We denote by  $u^T$  the transposition of the row-vector  $u$ .

## Some remarks

### Remark

*The operation  $\otimes_{M,N,P}$  combines  $\oplus$ ,  $\eta$  and  $\rho$  of clique-width and this is done in linear algebra over  $GF(2)$ .*

### Remark

*As in the operations by Wanke, these operations add edges between two disjoint graphs, that are the two arguments. This is a difference with clique-width where the single binary operation  $\oplus$  is used, and  $\eta_{i,j}$  applied to  $G \oplus H$  may add edges to  $G$  and to  $H$ .*

## In terms of products of matrices...

$$K = G \otimes_{M,N,P} H \implies$$

$$A_K = \begin{pmatrix} A_G & \Gamma_G \cdot M \cdot \Gamma_H^T \\ \Gamma_H \cdot M^T \cdot \Gamma_G^T & A_H \end{pmatrix}$$

$$\Gamma_K = \begin{pmatrix} \Gamma_G \cdot N \\ \Gamma_H \cdot P \end{pmatrix}$$

where  $M^T$  denotes the transposition of the matrix  $M$ .

- For  $u \in \mathbb{B}^n$ , we denote by  $\mathbf{u}$  the graph with a single vertex colored by  $u$  and  $C_n = \{\mathbf{u} \mid u \in \mathbb{B}^n\}$ .
- We let  $R_n$  be the set of linear recolorings and bilinear products where  $M, N, P$  are  $(k \times \ell)$ -matrices with  $k, \ell \leq n$ .
- We denote by  $val(t)$  the graph defined by a term  $t \in T(R_n, C_n)$ . This graph is the value of the term in the corresponding algebra.

# Plan

- 1 Preliminaries
- 2 **Vectorial Colorings**
  - Graph Operations
  - **Main theorem**
- 3 General Framework for balanced decompositions
  - Balancing theorems: General ideas
  - Tools

# Main theorem 1

A graph  $G$  has rank-width at most  $n$  iff it is the value of a term in  $T(R_n, C_n)$ .

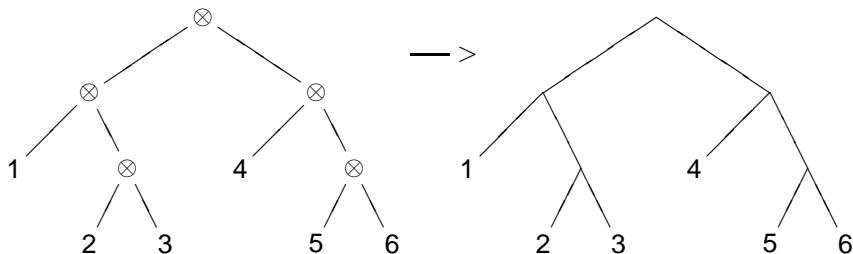
## Remark

A context  $c$  is a term in  $T(R_n, C_n \cup \{u\})$  with a single occurrence of the variable  $u$ . The context  $u$  is denoted by  $Id$ .

For  $t \in T(R_n, C_n)$  we let  $c \bullet t = c[t/u]$  (substitution)

# Proof of “If direction”(1)

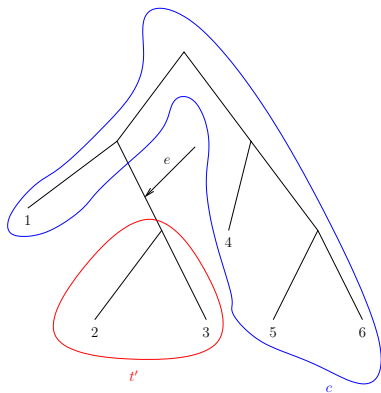
- Let  $G = \text{val}(t)$ . We let  $T$  be the syntactic tree of  $t$ . There exists a bijection  $\mathcal{L} : V_G \rightarrow L_T$  where  $L_T$  is the set of leaves of  $T$ .
- We take  $(T, \mathcal{L})$  as a layout of  $G$  and we prove that the width of  $T$  is at most  $n$ .



## “If direction”(2)

### Claim

If  $t = c \bullet t'$ ,  $c \neq Id$  a context and  $t' \in T(R_n, C_n)$  then  $A_G[V_H, V_G - V_H] = \Gamma_H \cdot B$  and  $\Gamma_{G[V_H]} = \Gamma_H \cdot C$  for some matrices  $B$  and  $C$  ( $H = val(t')$ ).

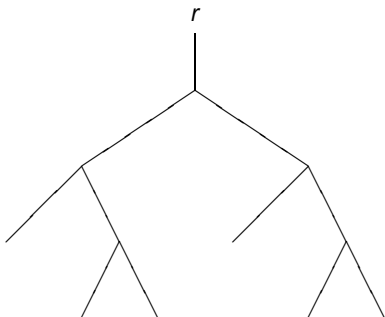


$$c = c' \otimes_{M,N,P} t''$$

- Then  $G = G' \otimes_{M,N,P} K$  where  $G' = val(c' \bullet t')$  and  $K = val(t'')$ .
- $A_G[V_H, V_G - V_H] = (A_{G'}[V_H, V_{G'} - V_H] \Gamma_{G'[V_H]} \cdot M \cdot \Gamma_K^T)$ .
- $A_{G'}[V_H, V_{G'} - V_H] = \Gamma_H \cdot B'$ ,  
 $\Gamma_{G'[V_H]} = \Gamma_H \cdot C'$
- $A_G[V_H, V_G - V_H] = \Gamma_H \cdot (B' \ C' \cdot M \cdot \Gamma_K^T)$ .
- $\Gamma_{G[V_H]} = \Gamma_{G'[V_H]} \cdot N = \Gamma_H \cdot C' \cdot N$ .



# Proof of converse direction(1)



Layout  $(T, \mathcal{L})$  of  $G$

## Layout $(T, \mathcal{L})$ of $G$

- Choose a vertex  $r$  of degree 1 in  $T$ .
- Direct  $T$  with  $r$  as root.
- $T$  directed, will be the syntactic tree of a term  $t$  that defines  $G$ .
- We must show how to label each internal node of  $T$  by an operation  $\otimes_{M,N,P}$  and how to color each vertex.

## Proof of Converse direction (2)

### Lemma 1

Let  $G$  be a graph with a bipartition  $V_G = V_1 \cup V_2$ . Let  $m = rk(A_G[V_1, V_2])$ . Then  $G = H \otimes_M K$  where  $M$  is a nonsingular  $m \times m$  matrix, for some  $\mathbb{B}^m$ -colorings  $H$  and  $K$  of  $G[V_1]$  and  $G[V_2]$  respectively.

### Note

- $G[V_1]$  denotes the induced subgraph on  $V_1$ .
- $\otimes_M = \otimes_{M,N,P}$  for some  $N$  and  $P$  we don't care about.

## Idea of proof of Lemma 1 (1)

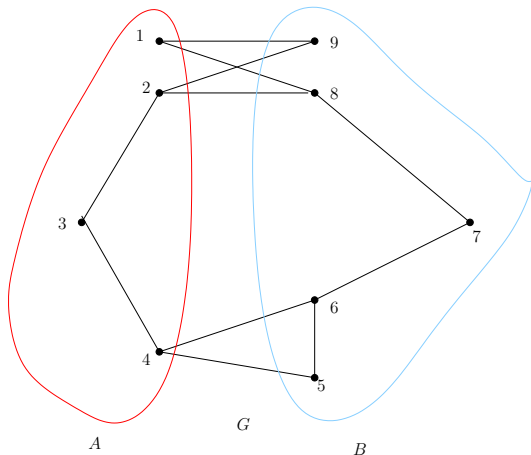
	5	8	9	6	7
1	0	1	1	0	0
4	1	0	0	1	0
2	0	1	1	0	0
3	0	0	0	0	0

$$M = A_G[\{1, 4\}, \{5, 8\}]$$

$$A_G[2] = A_G[1]$$

$$A_G^T[9] = A_G^T[8]$$

$$A_G^T[6] = A_G^T[5]$$



## Idea of Proof of Lemma 1(2)

We let  $G[A] = H$  and  $G[B] = K$ .

- We call  $\{1, 4\}$  a vertex-basis of  $H$  with respect to  $K$
- We call  $\{5, 8\}$  a vertex-basis of  $K$  with respect to  $H$

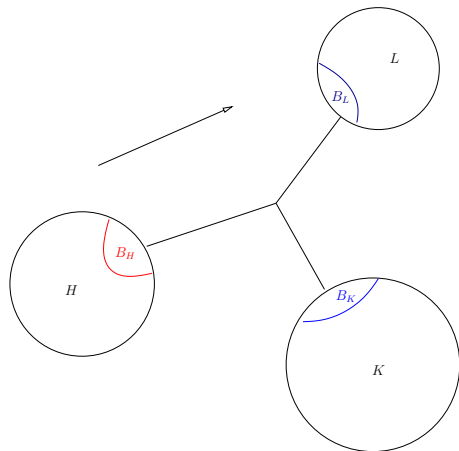
$\mathbb{B}^3$ -coloring of  $H$

1	1	0
2	1	0
3	0	0
4	0	1

$\mathbb{B}^3$ -coloring of  $K$

5	1	0
6	1	0
7	0	0
8	0	1
9	0	1

# Proof of Converse direction (3)

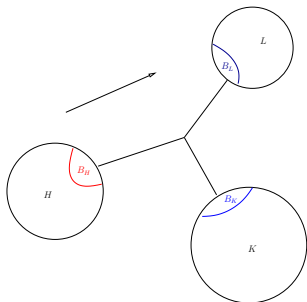


## Lemma 2

- $(V_H, V_K, V_L)$  a 3-partition of  $V_G$ ,
- $H = G[V_H], K = G[V_K]$  and  $L = G[V_L]$ ,
- $h = \rho_G(V_H), k = \rho_G(V_K), \ell = \rho_G(V_L)$

$$G = (H \otimes_{M, N_1, N_2} K) \otimes_P L.$$

# Idea of proof of Lemma 2(1)



## Choice of vertex-basis

- Choose  $B_H$  a vertex-basis of  $H$  with respect to  $K \cup L$
- Choose  $B_K$  a vertex-basis of  $K$  with respect to  $H \cup L$
- Choose  $B_L$  a vertex-basis of  $L$  with respect to  $H \cup K$
- Extract from  $B_H \cup B_K$  a vertex-basis of  $H \cup K$  with respect to  $L$ .

## Idea of proof of Lemma 2 (2)

- By Lemma 1, there exist  $\mathbb{B}^h$ -coloring of  $H$ ,  $\mathbb{B}^k$ -coloring of  $K$  and a  $(h \times k)$ -matrix  $M$  such that  $G[H \cup K] = H \otimes_M K$ .
- A vertex-basis of  $H \cup K$  with respect to  $L$  is included in  $B_H \cup B_K$ . By Lemma 1, there exists a matrix  $P$  such that  $G = G' \otimes_P L$  where  $G'$  is a  $\mathbb{B}^\ell$ -coloring of  $G[H \cup K]$ .
- We can thus find a recoloring of  $H$  by  $N_1$  and a recoloring of  $K$  by  $N_2$  such that  $G' = H \otimes_{M, N_1, N_2} K$ .
- We have then  $G = (H \otimes_{M, N_1, N_2} K) \otimes_P L$ .

# Plan

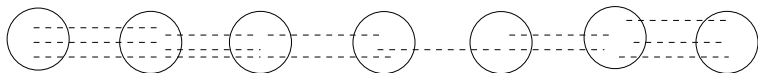
- 1 Preliminaries
- 2 Vectorial Colorings
  - Graph Operations
  - Main theorem
- 3 General Framework for balanced decompositions
  - **Balancing theorems: General ideas**
  - Tools



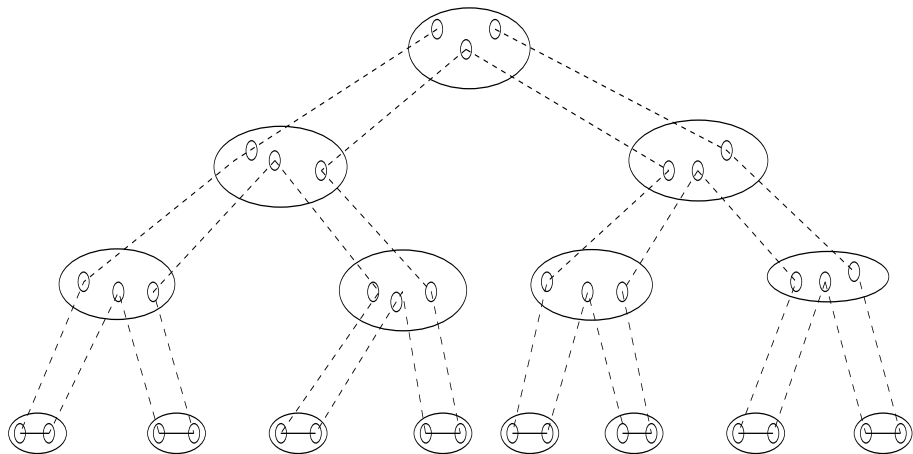
# Balancing Theorems: General ideas

- For algorithmic purposes it is sometimes crucial to use  $a$ -balanced terms, i.e., of height at most  $a(\log(n) + 1)$ .
- This is the case for instance of:
  - ▶ the labeling schemes considered by Courcelle, Vanicat, Twigg (STACS'07).
  - ▶ the NC-algorithms considered by Bodlaender.
- How to transform a path-decomposition of width  $k$  into a tree-decomposition of height  $O(\log(n))$  and of width  $O(k)$ ?
- We cut at the “middle” of the path-decomposition and we iterate the process with the smaller paths.

# Balancing Theorems: General ideas



# Balancing Theorems: General ideas



# Balancing Theorems: General ideas

- For algorithmic purposes it is sometimes crucial to use  $a$ -balanced terms, i.e., of height at most  $a(\log(n) + 1)$ .
- This is the case for instance of:
  - ▶ the labeling schemes considered by Courcelle, Vanicat, Twigg (STACS'07).
  - ▶ the NC-algorithms considered by Bodlaender.
- How to transform a path-decomposition of width  $k$  into a tree-decomposition of height  $O(\log(n))$  and of width  $O(k)$ ?
- We cut at the “middle” of the path-decomposition and we iterate the process with the smaller paths.
- We thus have a tree-decomposition of width at most  $3k$  and of height  $\log(n)$ .

# Balancing Theorems: General ideas

- For algorithmic purposes it is sometimes crucial to use  $a$ -balanced terms, i.e., of height at most  $a(\log(n) + 1)$ .
- This is the case for instance of:
  - ▶ the labeling schemes considered by Courcelle, Vanicat, Twigg (STACS'07).
  - ▶ the NC-algorithms considered by Bodlaender.
- How to transform a path-decomposition of width  $k$  into a tree-decomposition of height  $O(\log(n))$  and of width  $O(k)$ ?
- We cut at the “middle” of the path-decomposition and we iterate the process with the smaller paths.
- We thus have a tree-decomposition of width at most  $3k$  and of height  $\log(n)$ .
- How to generalize it to arbitrary tree-decompositions? And what about clique-width, rank-width and other width?

# Generalisation

- Each tree has a balanced tree-decomposition of width 2 and of height at most  $O(\log(n))$  [Bodlaender, 88].
- Every graph of tree-width  $k$  admits a tree-decomposition of width at most  $3k + 2$  and of height at most  $O(\log(n))$  [Bodlaender, 88].
- Every graph of clique-width  $k$  admits a clique-width expression of height at most  $3\log(n)$  and uses  $k \cdot 2^{k+1}$  colors at most [Courcelle-Vanicat, 03].
- Every graph of  $m$ -clique-width [variant of clique-width that uses several colors of each vertex]  $k$  has a  $m$ -clique-width expression of height at most  $3\log(n)$  and uses  $2k$  colors at most [Courcelle-Twigg, 07].
- Our result **unifies** all these results and as a corollary we get that **every graph of rank-width  $k$  admits a layout of height at most  $3\log(n)$  and of width at most  $2k$ .**

# Plan

## 1 Preliminaries

## 2 Vectorial Colorings

- Graph Operations
- Main theorem

## 3 General Framework for balanced decompositions

- Balancing theorems: General ideas
- **Tools**

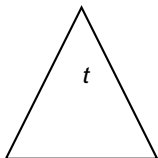
# Tools - Informal definitions

## Terms and Contexts

- We let  $F_k$  (e.g.  $\{\oplus, \eta_{i,j}, \rho_{i,j} \mid i, j \in [k], i \neq j\}$ ) be a set of binary operations and  $C_k$  (e.g.  $\{i \mid i \in [k]\}$ ) be a set of constants.
- A term is a well-formed expression in  $T(F_k, C_k)$



# Tools - Informal definitions

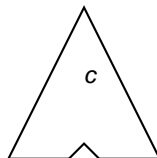
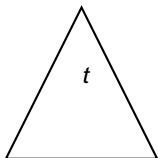


# Tools - Informal definitions

## Terms and Contexts

- We let  $F_k$  (e.g.  $\{\oplus, \eta_{i,j}, \rho_{i,j} \mid i, j \in [k], i \neq j\}$ ) be a set of binary operations and  $C_k$  (e.g.  $\{i \mid i \in [k]\}$ ) be a set of constants.
- A term is a well-formed expression in  $T(F_k, C_k)$
- A context is a well-formed expression in  $T(F_k, C_k \cup \{u\})$  with one and only one occurrence of  $u$ .

## Tools - Informal definitions



# Tools - Informal definitions

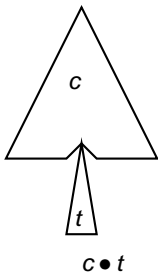
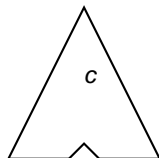
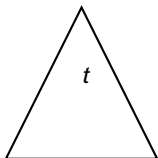
## Terms and Contexts

- We let  $F_k$  (e.g.  $\{\oplus, \eta_{i,j}, \rho_{i,j} \mid i, j \in [k], i \neq j\}$ ) be a set of binary operations and  $C_k$  (e.g.  $\{i \mid i \in [k]\}$ ) be a set of constants.
- A term is a well-formed expression in  $T(F_k, C_k)$
- A context is a well-formed expression in  $T(F_k, C_k \cup \{u\})$  with one and only one occurrence of  $u$ .

## Substitution of a term to $u$

$c \bullet t = c[t/u] \in T(F_k, C_k)$  for  $c$  a context and  $t$  a term

# Tools - Informal definitions



# Tools - Informal definitions

## Terms and Contexts

- We let  $F_k$  (e.g.  $\{\oplus, \eta_{i,j}, \rho_{i,j} \mid i, j \in [k], i \neq j\}$ ) be a set of binary operations and  $C_k$  (e.g.  $\{i \mid i \in [k]\}$ ) be a set of constants.
- A term is a well-formed expression in  $T(F_k, C_k)$
- A context is a well-formed expression in  $T(F_k, C_k \cup \{u\})$  with one and only one occurrence of  $u$ .

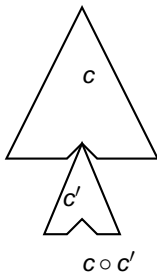
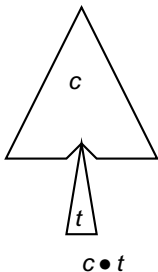
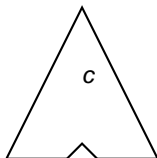
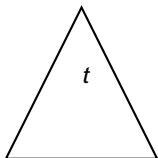
## Substitution of a term to $u$

$c \bullet t = c[t/u] \in T(F_k, C_k)$  for  $c$  a context and  $t$  a term

## Substitution of a context to $u$

$c \circ c' = c[c'/u] \in T(F_k, C_k \cup \{u\})$  for  $c, c'$  contexts.

# Tools - Informal definitions



# Intuition of the framework

We transform a term  $t$  into a 3-balanced term in 2 steps:

## Step 1: Iterated splitting

Transform the term  $t$  into a term  $t^b$  of height at most  $3\log(n)$  written with symbols in  $F_k \cup C_k \cup \{\bullet, \circ\}$ .

## Step 2: Elimination of $\bullet$ and $\circ$ by operations

We replace the operations  $\bullet$  and  $\circ$  in  $t^b$  by operations that uses more colors, but not so much.

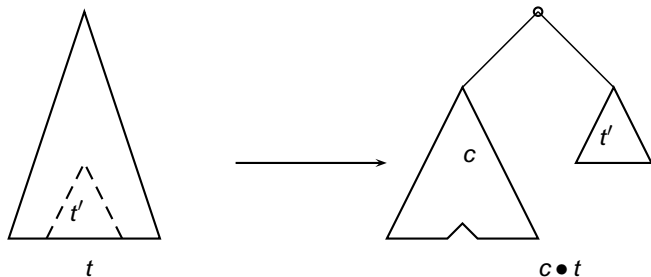


# Iterated splitting

- We split a term  $t$  into a context  $c$  and a term  $t'$  such that  $c$  and  $t'$  are as equal as possible  $\implies t = c \bullet t'$ .

# Iterated splitting

- We split a term  $t$  into a context  $c$  and a term  $t'$  such that  $c$  and  $t'$  are as equal as possible  $\implies t = c \bullet t'$ .

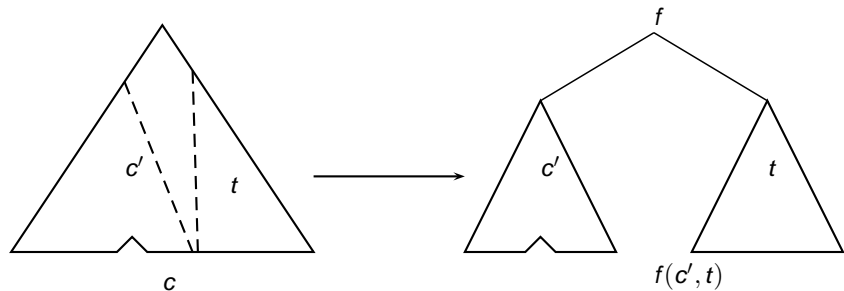


# Iterated splitting

- A context  $c$  is written  $c = f(c', t)$  if  $|t| \simeq |c'|$  or splitted (next slide).

# Iterated splitting

- A context  $c$  is written  $c = f(c', t)$  if  $|t| \simeq |c'|$  or splitted (next slide).

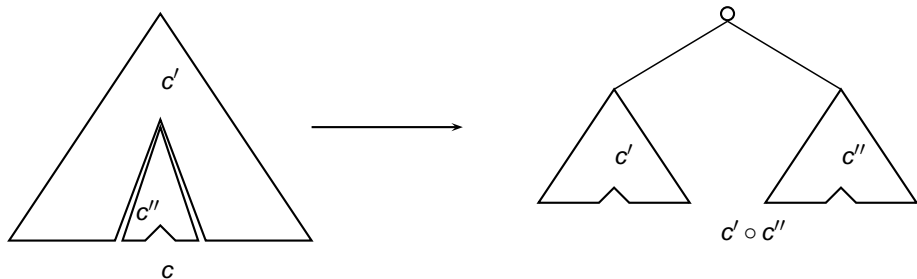


# Iterated splitting

- A context  $c$  can be splitted into contexts  $c', c''$  with  $c'$  and  $c''$  as “equal” as possible and  $c = c' \circ c''$ .

## Iterated splitting

- A context  $c$  can be splitted into contexts  $c', c''$  with  $c'$  and  $c''$  as “equal” as possible and  $c = c' \circ c''$ .



## Elimination of $\bullet$ and $\circ$

For terms and contexts over graph operations, we use the following rules to transform a term written with symbols in  $F_k \cup C_k \cup \{\bullet, \circ\}$  into a term in  $T(F_{k'}, C_{k'})$  with  $k' \geq k$ :

- A term  $t$  is replaced by  $val(t)$ .
- $c \bullet t_H$  is replaced by  $f'(G_c, H)$  where  $H = val(t_H)$  and  $G_c$  is a colored graph that represents the context  $c$ .
- $c \circ c'$  is replaced by  $f''(G_c, G_{c'})$  (inductivity of the construction)



With the rules above, one can eliminate  $\bullet$  and  $\circ$  and replace them by operations using more colors.

# Conclusion

- Rank-width is an interesting complexity measure because it is equivalent to clique-width, and also because it is increasing for the vertex-minor inclusion and has a polynomial approximation algorithm.
- We give an algebraic characterization of rank-width by means of operations based on linear transformations of the  $GF(2)$ -vector space.
- The Algebraic characterization allows us to define the notion of rank-width for directed graphs.
- Using approximations of Oum, we can extend the approximation algorithms to “directed” clique-width.
- In some applications, balanced decompositions are crucial. We give a general framework that handles many graph decompositions (tree-decomposition, branch-decomposition, clique-width, NLC-width,  $m$ -clique-width, Boolean-width and rank-width.)



# Perspectives

- The tightness of the bound of the balanced terms.
- Generalize the general framework of balanced terms to arbitrary signatures (not only binary ones)
- Can we do the balancing in linear time instead of  $O(n\log(n))$ ?