Graph operations characterizing rank-width and balanced graph expressions

Mamadou Moustapha KANTÉ
(joint work with Bruno Courcelle)

LaBRI, Université Bordeaux 1, CNRS
351 cours de la libération 33405 Talence Cedex.

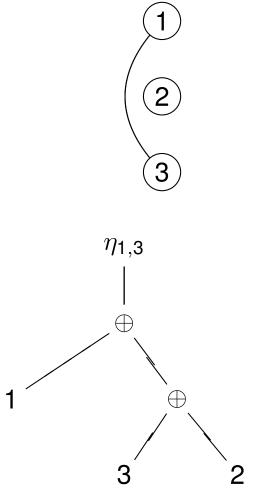
April 03 2007
ANR GRAAL Project

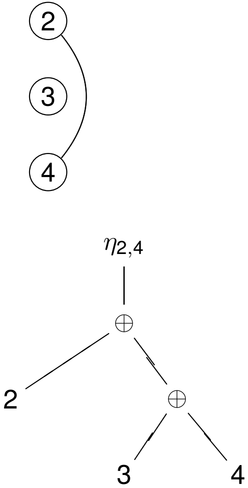
Introduction: Widths, decomposition and inductive computations

- In many cases G has width at most k means G is the value of a term $t \in T(F_k, C_k)$ where F_k is the related signature and C_k a set of constants.
- We have thus inductive computations based on the structure of the terms $\Longrightarrow$ linear time.
- For instance, when a term defining a cograph G ($cwd(G) \leq 2$) is given, we can compute its chromatic number $\chi(G)$ using the following rules:

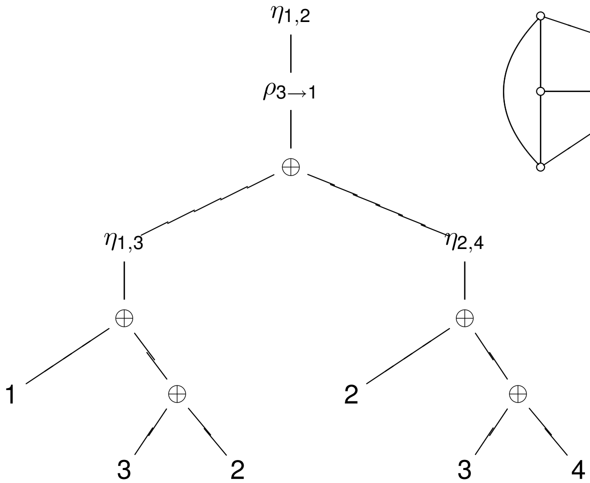
$$\begin{cases} \chi(G \oplus H) = \max\{\chi(G), \chi(H)\}, \\ \chi(G \otimes H) = \chi(G) + \chi(H), \\ \chi(\bullet) = 1 \end{cases}$$

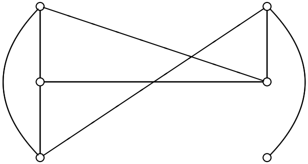
Clique-width(1)





Clique-width (2)





Overview(1)

- As tree-width, when k is part of the input, recognizing graphs of clique-width k is NP-complete (Fellows and al.)
- But contrary to tree-width, there is no polynomial algorithm that decides for fixed k, if a graph G has clique-width at most k.
- Oum and Seymour investigate the problem and define the notion of *rank-width* equivalent to clique-width.
- Rank-width is the parameter associated with a layout of a graph, which is defined as a *branch-decomposition* of a symmetric and submodular function.
- The disadvantage of rank-width is we need to transform the rank-decomposition into a clique-width expression to solve MS problems, which yields an exponential.
- We propose algebraic operations that characterize rank-width, operations based on linear transformations of the $GF(2)$-vector space.

Overview(2)

- The hidden constants yield by the general FPT algorithms based on MS logic are very huge (towers of exponents).
- It is sometimes better to use dynamic programming.
- We need sometimes balanced decompositions.
- It is the case for some applications (labeling schemes considered by Courcelle and al., Gavoille, ... and, NC-algorithms considered by Bodlaender for instance).
- Bodlaender proved that every tree-decomposition of width k can be transformed into a balanced tree-decomposition of width at most $3k+2$.
- We propose a general framework for obtaining balanced decompositions that handle many graph decompositions (rank-width, tree-width, clique-width, ...).

Plan

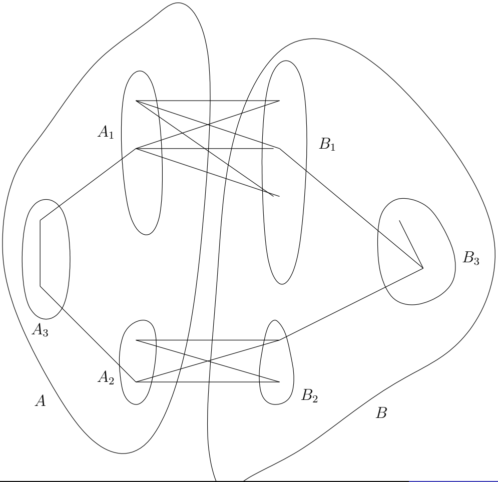
Notations

- Graphs are simple, loop-free and undirected unless specified. They are defined as $< V_G, A_G >$.
- We denote by A_G the adjacency-matrix of a graph G.
- A *sub-cubic tree* is a tree such that the degree of each node is at most 3.

Rank-width

- Let $X \subseteq V_G$. We let $\rho_G(X) = rk(A_G[X, V_G - X])$ where rk is the matrix rank function.
- A layout of a graph G is a pair (T, L) where T is a sub-cubic tree and $L : V_G \to N_T$ where N_T is the set of nodes of degree 1 in T.
- Any edge e of T induced a bipartition (X_e, Y_e) of the set N_T, thus a bipartition $(X, V - X)$ of V_G.
- the width of e is $\rho_G(X)$ and the width of the layout (T, L) is the maximum width over all edges e of T.
- The rank-width of G, denoted by $rwd(G)$, is the minimum width over all layouts of G

Example of layout of a graph(1)

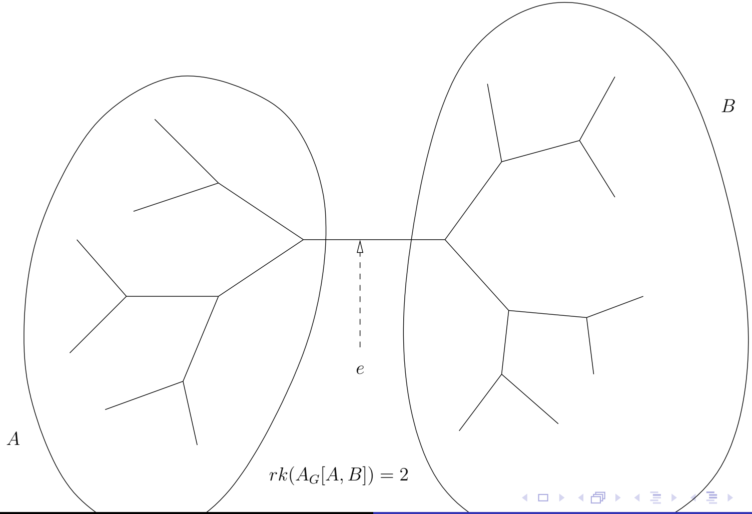


$rk(A_G[A, B]) = 2$

$A_G[A, B] =$

$A \backslash B$	B_1	B_2	B_3
A_1	1	0	0
A_2	0	1	0
A_3	0	0	0

Example of layout of a graph (2)



Properties of rank-width(1)

Clique-width

- $F_k = \{\oplus, \eta_{i,j}, \rho_{i \to j} \mid i, j \in [k], i \neq j\}$ and $C_k = \{\mathbf{i} \mid i \in [k]\}$
- The clique-width of a graph G is the minimum k such that G is the value (up to isomorphism) of a term t in $T(F_k, C_k)$.

Clique-width and Rank-width

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

Properties of rank-width(2)

- For fixed k, Oum proposed 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 small as possible).
- This is why characterization of rank-width by graph operations is interesting.

Plan

Plan

Vectorial colorings

- Handling multiple colorings of vertices with k colors is clearly the same thing as handling colorings with colors in $\{0,1\}^k$ ($GF(2)$-vectors).
- Let $k \in \mathbb{N}$, $\mathbb{B} = \{0,1\}$. A $\mathbb{B}^k$-coloring of a graph G is a mapping $\gamma : V_G \rightarrow \mathbb{B}^k$.
- We consider that $x \in V_G$ has color i (among others) iff $\gamma(x)[i]$ (the i-th component of $\gamma(x)$) is 1.
- A $\mathbb{B}^k$-colored graph is a triple $G =< V_G, edg_G, \gamma_G >$ where γ_G is a $\mathbb{B}^k$-coloring of $< V_G, edg_G >$.
- Let us now define the operations ...

Linear recolorings

Definition

A recoloring $Recol_h$ is linear if $h : \mathbb{B}^k \to \mathbb{B}^\ell$ is linear, i.e., if for some $(k \times \ell)$-matrix N and all graphs G, we have, by letting $H = Recol_h(G)$:

$$\Gamma_H = \Gamma_G.N$$

i.e., $\gamma_H(x) = \gamma_G(x).N$ for each x in V_G.
If h and h' are linear recolorings, described respectively by N and N', then $h \circ h'$ is linear and is described by $N' \times N$.

We say that h is described by N and we note it $Recol_N$.

Remark

2^k different colors are handled efficiently via $(k \times \ell)$-matrices.

Edge creation from bilinear forms

For G, $\mathbb{B}^k$-colored and H, $\mathbb{B}^\ell$-colored, edges between G and H are created in 3 steps.

- The disjoint union of G and H.
- $x \in V_G$ colored by u and $y \in V_H$ colored by v are linked iff $u.M.v^T = 1$.
- Independent recolorings of G and H defined by N and P.

We denote this operations by $\otimes_{M,N,P}$.

Some remarks

Remark

It combines $\oplus$, η and ρ of clique-width and this is done in linear algebra.

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 a single binary operation 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...

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

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

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

- We let R_n be the set of linear recolorings and bilinear products.
- 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

Main theorem

Theorem

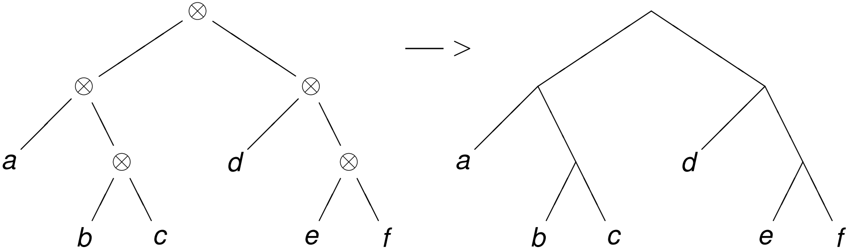
A graph G has rank-width at most n iff it is the value of a term in $T(R_n, C_n)$.

We can assume that a term in $T(R_n, C_n)$ is written with the binary operations $\otimes_{M,N,P}$ and the constants **u** where u is a row-vector in $\mathbb{B}^1 \cup \ldots \cup \mathbb{B}^n$ because it is clear that

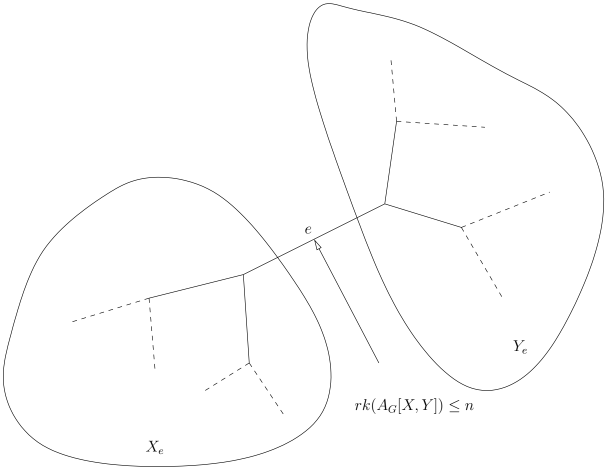
$$Recol_Q(G \otimes_{M,N,P} H) = G \otimes_{M,N',P'} H$$

where $N' = N.Q$ and $P' = P.Q$.

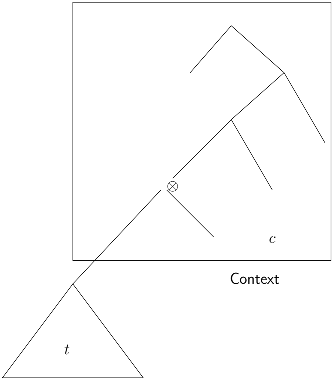
Proof of “If direction”(1)

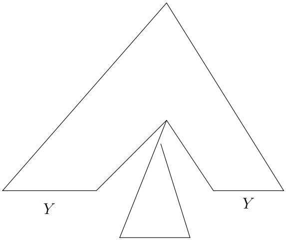


Proof of “If direction”(2)

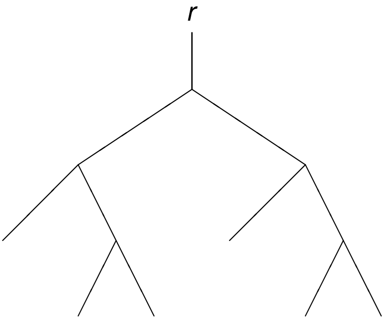


Proof of "If direction"(3)





Proof of converse direction(1)



Layout (T, L) of G

Layout (T, L) of G

- Choose a vertex r of degree 1 in T.
- Direct T with r as root.
- We claim that T directed, is 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)

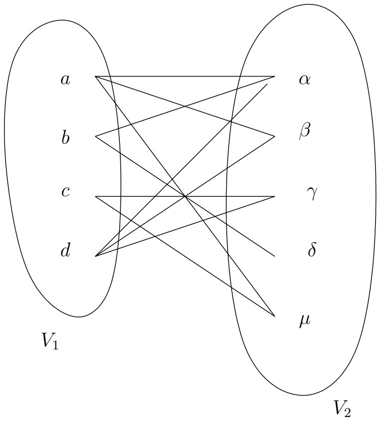
Lemma 1

Let G be a graph with a bipartition $V_G = V_1 \cup V_2$ of its vertices. 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.

Idea of proof of Lemma 1 (1)

	α	β	γ	δ	μ
a	1	1	0	0	1
b	1	0	0	1	0
c	0	0	1	0	1
d	1	1	1	0	0

$M = A_G[\{a,b,c\},\{\alpha,\beta,\gamma\}]$
$A_G[d] = A_G[a] + A_G[c]$
$A_G^T[\delta] = A_G^T[\alpha] + A_G^T[\beta]$
$A_G^T[\mu] = A_G^T[\beta] + A_G^T[\gamma]$



Idea of Proof of Lemma 1(2)

- We call $\{a, b, c\}$ a vertex-basis of H with respect to K

- We call $\{\alpha, \beta, \gamma\}$ a vertex-basis of K with respect to H

$\mathbb{B}^3$-coloring of $G[V_1]$, i.e., H

a	1	0	0
b	0	1	0
c	0	0	1
d	1	0	1

$\mathbb{B}^3$-coloring of $G[V_2]$, i.e., K

α	1	0	0
β	0	1	0
γ	0	0	1
δ	1	1	0
μ	0	1	1

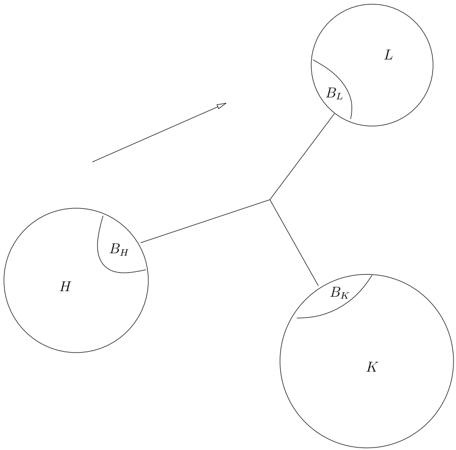
Proof of Converse direction (3)

Lemma 2

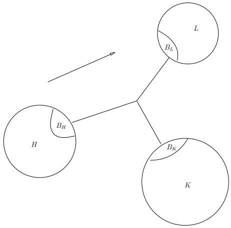
Let G be a graph, let H, K, L be induced subgraphs such that (V_H, V_K, V_L) is a 3-partition of V_G, with each component not empty. Let $h = \rho_G(V_H)$, $k = \rho_G(V_K)$, $\ell = \rho_G(V_L)$. There exist matrices of appropriate dimensions such that

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

Idea of proof of Lemma 2 (1)



Idea of proof of Lemma 2(2)



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.

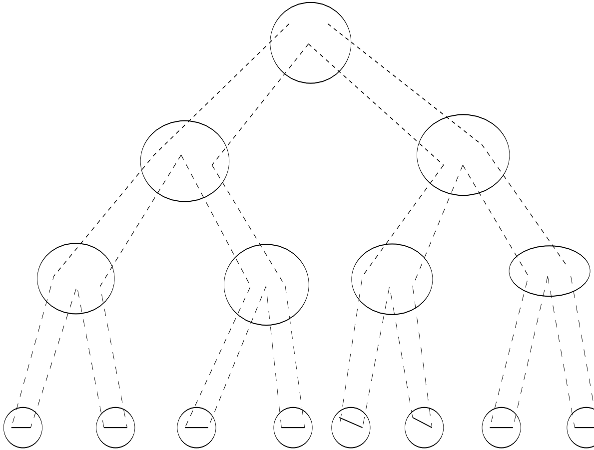
Idea of proof of Lemma 2 (3)

- 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 $G[H \cup K] = H \otimes_M K$.
- $B_H \cup B_K$ is included in a vertex-basis of $H \cup K$ with respect to G. 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

Plan

Balacing Theorems: General ideas



Generalisation

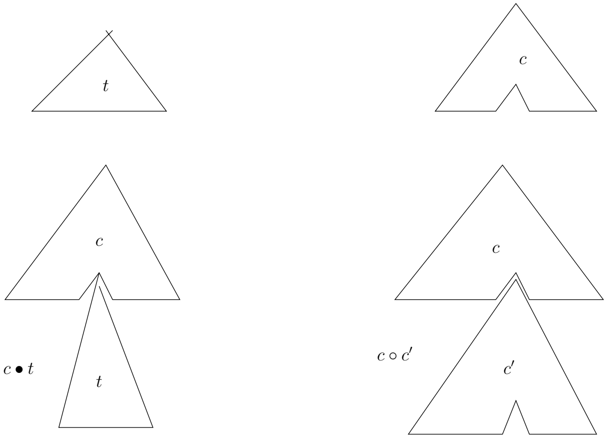
- Each tree has a balanced tree-decomposition of width 2 and of height at most $3\log(n)$.
- Bodlaender proves that every graph of tree-width k admits a tree-decomposition of width at most $3k+2$ and of height at most $\log(n)$).
- Courcelle and Vanicat proves that every graph of clique-width k admits a clique-width expression of height at most $3\log(n)$ and uses $k.2^k$ colors at most.
- Courcelle and Twigg proves that 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.
- 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

Tools - Informal definition

- 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 operations and C_k (e.g. $\{\mathbf{i} \mid i \in [k]\}$) 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.
- $c \bullet t = c[t/u] \in T(F_k, C_k)$ for c a context and t a term
- $c \circ c' = c[c'/u] \in T(F_K, C_k \cup \{u\})$ for c, c' contexts.

Tools - Some visual intuitions



Intuition of the framework

Iterated splitting

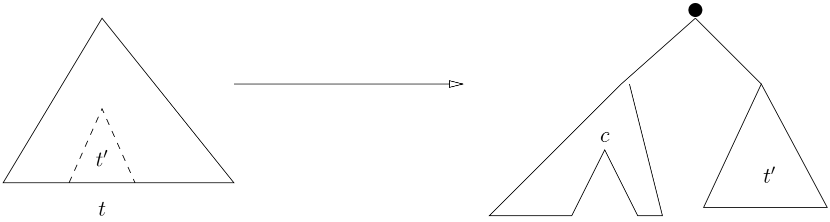
Transform the term t into a term t^b of height at most $3\log(n)$ written with symbols in $F_k \cup C_k \cup \{\bullet, \circ\}$.

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(1)

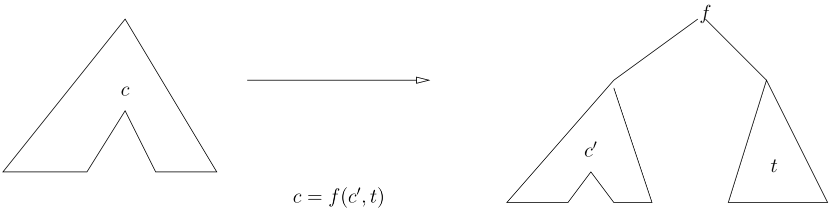
- We split a term t into a context c and a term t' such that c and t' are as equal as possible => $t = c \bullet t'$.



$$t = c \bullet t'$$

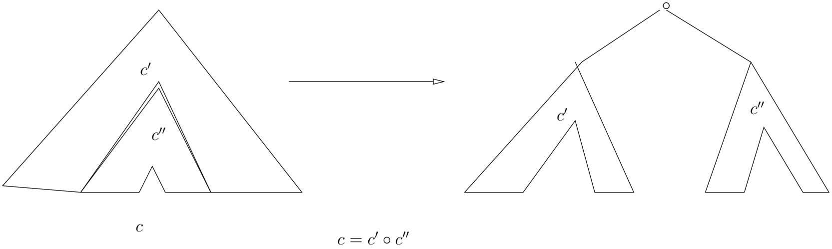
Iterated splitting(2)

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



Iterated splitting(3)

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



Elimination of • and ∘

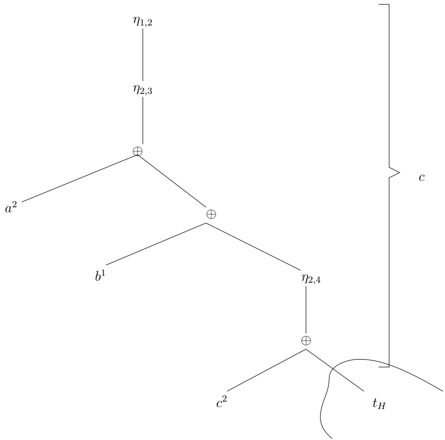
For terms and contexts over graph operations, we use the following rules to transform a term written with symbols in $F_k \cup C_k \cup \{\bullet, \circ\}$ into a term in $T(F_{k'}, C_{k'})$ with $k' \geq k$:

- 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)

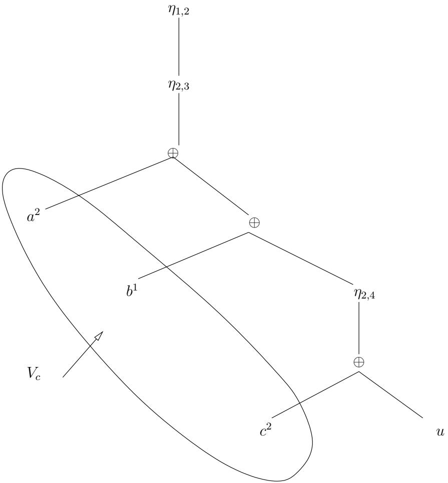
$\Longrightarrow$

With the rules above, one can eliminate • and ∘ and replace them by operations using more colors.

What is G_c: Example of clique-width over $[k]$? (1)



What is G_c: Example of clique-width over $[k]$? (2)



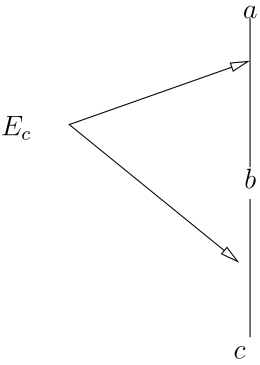
The set of vertices V_c

V_c is the set of leaves in c different from u.

What is G_c: Example of clique-width over $[k]$? (3)

The set of edges E_c

$$E_c = E_{G[V_c]}.$$

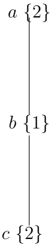


What is G_c: Example of clique-width over $[k]$? (4)

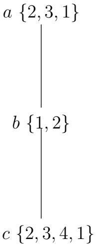
The color of a vertex

Each vertex v in V_c is colored by its color in $c[\emptyset/u]$ + a set J, which is the set of labels $j \in [k]$ such that v is linked to a vertex in H colored by j in H.

$val(c[\emptyset/u]) =$



$G_c =$



Conclusion

- This is why we have $k.2^k$ colors for clique-width.
- The same method works for m-clique-width and rank-width but we have $2k$ instead of the exponent.

Conclusion

- Rank-width is an interesting complexity measure because it is equivalent to clique-width, a powerful graph complexity measure, and also because it is increasing for the vertex-minor inclusion and has a polynomial approximation algorithm.
- We propose an algebraic characterization of the notion of rank-width by means of linear transformations of the $GF(2)$-vector space.
- Because in some applications, balanced decompositions are crucial, we propose here a general framework that handle many graph decompositions (tree-decomposition, clique-width, NLC-width, m-clique-width, Boolean-width and rank-width.

Conclusion

- Rank-width is an interesting complexity measure because it is equivalent to clique-width, a powerful graph complexity measure, and also because it is increasing for the vertex-minor inclusion and has a polynomial approximation algorithm.
- We propose an algebraic characterization of the notion of rank-width by means of linear transformations of the $GF(2)$-vector space.
- Because in some applications, balanced decompositions are crucial, we propose here a general framework that handle many graph decompositions (tree-decomposition, clique-width, NLC-width, m-clique-width, Boolean-width and rank-width.

Conclusion

- Rank-width is an interesting complexity measure because it is equivalent to clique-width, a powerful graph complexity measure, and also because it is increasing for the vertex-minor inclusion and has a polynomial approximation algorithm.
- We propose an algebraic characterization of the notion of rank-width by means of linear transformations of the $GF(2)$-vector space.
- Because in some applications, balanced decompositions are crucial, we propose here a general framework that handle many graph decompositions (tree-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 structures (not only binary ones).
- Can we do the balancing in linear time instead of $O(n\log(n))$?
- We propose a definition of rank-width for directed graphs, study it.

Perspectives

- The tightness of the bound of the balanced terms.
- Generalize the general framework of balanced terms to arbitrary structures (not only binary ones).
- Can we do the balancing in linear time instead of $O(n\log(n))$?
- We propose a definition of rank-width for directed graphs, study it.

Perspectives

- The tightness of the bound of the balanced terms.
- Generalize the general framework of balanced terms to arbitrary structures (not only binary ones).
- Can we do the balancing in linear time instead of $O(n\log(n))$?
- We propose a definition of rank-width for directed graphs, study it.

Perspectives

- The tightness of the bound of the balanced terms.
- Generalize the general framework of balanced terms to arbitrary structures (not only binary ones).
- Can we do the balancing in linear time instead of $O(n\log(n))$?
- We propose a definition of rank-width for directed graphs, study it.

Plan

Plan

Proof (if direction)

- Let $t \in T(R_k, C_k)$ be such that $G = val(t)$.
- We let t have its root colored by a binary operation $\otimes_{M,N,P}$. We take the syntactic tree of t as a layout of G.

Claim

If $t = c \bullet t'$, $t' \in T(R_n, C_n)$, $c \in Cxt(R_n, C_n) - \{Id\}$, $G = val(t)$, $H = val(t')$ then we have:

1. $A_G[V_H, V_G - V_H] = \Gamma_H.B$ *and* $\Gamma_{G[V_H]} = \Gamma_H.C$ *for some matrices* B *and* C,
2. $rk(A_G[V_H, V_G - V_H]) \leq n$.

Proof of the claim

- *(2)* is an immediate consequence of (1).
- For *(1)*, we use an induction on the size of c. We have 2 cases:
  - $c = Id \otimes_{M,N,P} t''$ (similarly $c = t'' \otimes_{M,N,P} Id$), i.e., $G = K \otimes_{M,N,P} H$ where $K = val(t'')$.
  - $c = c' \otimes_{M,N,P} t''$ (similarly $c = t'' \otimes_{M,N,P} c'$) where $c' \neq Id$, i.e., $G = G' \otimes_{M,N,P} K$ where $G' = val(c' \bullet t')$ and $K = valt(t'')$.

The proof follows from the definition and inductive hypothesis.■

Proof Scketch of the lemma 1(1)

- We select m vertices $x_1, \ldots, x_m$ in V_H with associated row-vectors independent and m vertices $y_1, \ldots, y_m$ in V_K with associated column-vectors independent.
- We denote by $x_{m+1}, \ldots, x_p$ and $y_{m+1}, \ldots, y_q$ the other vertices of H and K respectively.
- We denote by $A_G[x_i]$ the row-vector associated with x_i. For each $j = 1, \ldots, p$ we have

$$A_G[x_j] = \sum_{1 \leq i \leq m} b_{ij} A_G[x_i]$$

- Similarly for $j = 1, \ldots, q$ we let

$$A_G^T[y_j] = \sum_{1 \leq i \leq m} c_{ij} A_G^T[y_i]$$

Proof Scketch of the lemma 1(2)

- We let $M = A_G[\{x_1, \ldots, x_m\}, \{y_1, \ldots, y_m\}]$ and $\gamma_H(x_j) = (b_{1j}, \ldots, b_{mj})$ and $\gamma_K(y_j) = (c_{1j}, \ldots, c_{mj})$.
- We prove that $G = H \otimes_M K$ using the definitions.
- By the choice of $x_1, \ldots, x_m$ and $y_1, \ldots, y_m$ the matrix M is non-singular.$\blacksquare$

Remark

- The construction above can be done if $\{A[x_i] \mid i = 1, \ldots, p\}$ where $p \geq rk(A)$ generates the row-vectors and $\{A^T[y_j] \mid j = 1, \ldots, q\},\ q \geq rk(A)$ generates the column-vectors. We call $\{x_1, \ldots, x_p\}$ a *vertex-generator* of H with respect to G.
- The linear combinations expressing $A[x_i]$ and $A^T[y_j]$ are not unique. Then we get $G = H \otimes_M K$ with M of dimension $p \times q$. We have the following from the proof of the previous lemma:

Proposition

Assume $G = H \otimes_A K$ with A of dimension $p \times q$ of rank k. Let M be a $k \times k$ sub-matrix of rank k of A. Then we have N of dimension $p \times k$, P of dimension $q \times k$ such that $A = N.M.P^T$ and $G = Recol_N(H) \otimes_M Recol_P(K)$.

Example 2

The same example can be used. We check $A = N.M.P^T$.

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \left(\begin{array}{ccc|cc} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{array}\right)$$

$$= \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \times \left(\begin{array}{ccc|cc} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{array}\right) = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Proof of Lemma 2(1)

We let $V_H = \{1, 2, \ldots, 5\}$, $V_K = \{a, b, c, d, e\}$, $V_L = \{\alpha, \beta, \gamma, \delta\}$. The portion of A_G relevant to the expression of G as $(H \otimes_{M,N_1,N_2} K) \otimes_P L$ is:

$A_G =$

	a	b	c	d	e	α	β	γ	δ
1	1	0	0	1	1	1	0	0	1
2	1	1	0	0	0	1	1	0	0
3	0	1	1	1	0	0	1	1	1
4	0	1	0	1	1	0	1	0	1
5	1	0	1	1	0	1	0	1	1
a						0	1	1	1
b						1	1	1	0
c						1	0	1	1
d						1	0	0	1
e						0	0	1	0

Proof of Lemma 2(2)

We have the following linear relation between rows and columns of $A_G[V_H, V_K \cup V_L]$, $A_G[V_K, V_H \cup V_L]$ and $A_G[V_L, V_H \cup V_K]$:

$$
\begin{aligned}
4 &= 1 + 2 \\
5 &= 2 + 3 \\
\delta &= \alpha + \beta \\
d &= a + b \\
e &= a + b + c
\end{aligned}
$$

Proof of Lemma 2(3)

Vertex bases are for $H : \{1, 2, 3\}$, for $K : \{a, b, c\}$, and for $L : \{\alpha, \beta, \gamma\}$. Among $\{1, 2, 3, a, b, c\}$, one can select $\{1, 2, a\}$.

$M =$

	a	b	c
1	1	0	0
2	1	1	0
3	0	1	1

$P =$

	α	β	γ
1	1	0	0
2	1	1	0
a	0	1	1

$N_1 =$

	1	2	a
1	1	0	0
2	0	1	0
3	0	0	1

$N_2 =$

	1	2	a
a	0	0	1
b	1	0	1
c	0	1	1

Proof of the “only if” direction(1)

- We let (t, L) be a layout of with n of G ($rwd(G) = n$).
- We root t with a leaf s as root.
- The weight of a node u of t is the number of the nodes of t/u, the subtree of the directed tree t rooted at u.
- For every edge of t of the form $\overrightarrow{vu}$, we let G_u be the induced subgraph of G, the vertices of which are the leaves of t/u, and G^v the induced subgraph of G, the vertices of which are the leaves not in t/u.

Claim

One can choose for each u different from the root of t, a vertex basis B_u of G_u relative to G; there corresponds to B_u a $\mathbb{B}^{r(u)}$-coloring of G_u where $r(u)$ is the width of $\overrightarrow{vu}$, and a term t_u in $T(R_n, C_n)$ defining G_u colored in this way.

Proof of the “only if” direction(2)

- The proof of the claim is by induction on the weight $w(u)$ of u, using the technical lemmas above.■
- For the case of u where $r \rightarrow u$, r is the root of t, we have G^r defined by $\mathbf{1}$ and G defined by $t_u \otimes_{P,O,O} \mathbf{1}$, where t_u is obtained by the claim above. This terminates the proof of the main theorem.■

Plan

Special Terms(1)

Definition

We let $S = T(F \cup \{\circ, \bullet\}, C \cup \{Id\})$. We let S_c and S_t be the least subsets of S such that:

$$S_t := S_c \bullet S_t \ \cup \ f(S_t, S_t) \ \cup \ b$$
$$S_c := S_c \circ S_c \ \cup \ f(S_t, S_c) \ \cup \ f(S_c, S_t) \ \cup \ f(S_t, Id) \ \cup \ f(Id, S_t)$$

with rules for each f in F, each b in C. We denote them by $SPE_t(F, C)$ and $SPE_c(F, C)$ if we need to specify F and C. Note that $Id \notin S_t \cup S_c$.

Every term t in $SPE_t(F, C)$ evaluates into a term $Eval(t)$ in $T(F, C)$ and every term c in $SPE_c(F, C)$ evaluates into a context $Eval(c)$ in $Cxt(F, C) - \{Id\}$.

Special terms(2)

Theorem

For every term t in $T(F, C) - C$ one can construct a term t^b in $SPE_t(F, C)$ such that $|t^b|_{FC} = |t|_{FC} = |t|$, $Eval(t^b) = t$ and $ht(t^b) \leq 3\log(|t| - 1)$. This term can be constructed in time $O(n\log(n))$ if $n = |t|$.

Flexibility

(F', C') is (F, C)-*flexible* if the following conditions hold:

1. F and F' are commutative.
2. There exist three mappings: $q \mapsto \hat{q}$, $q \mapsto f^q$ and $(q, q') \mapsto f^{q,q'}$ which satisfy the following properties:
   - (2.1) For every comb-term $q(x_1, \dots, x_n, u)$ over F with $n \geq 2$, $\hat{q}$ is a comb-term $\hat{q}(x_1, \dots, x_n)$ over F'.
   - (2.2) If $q(x_1, u)$ is the comb-term $g(x_1, u)$ then $\hat{q} = x_1$ and $f^q = g$.
   - (2.3) For every q as in (2.1), we have $f^q \in F'$ and $q \simeq f^q(\hat{q}, u)$.
   - (2.4) For every two comb-terms as in (2.1) or (2.2) $q(x_1, \dots, x_p, u)$ and $q'(x_1, \dots, x_n, u)$ we have $f^{q,q'} \in F'$ and

     $$\widehat{q''} \simeq f^{q,q'}(\hat{q}(x_1, \dots, x_p), \widehat{q'}(x_{p+1}, \dots, x_{p+n}))$$

     where $q'' = q(x_1, \dots, x_p, q'(x_{p+1}, \dots, x_{p+n}, u))$.

If q is a comb-term as in (2.2), Property (2.3) also holds from the definitions of $\hat{q}$ and f^q.

General theorem

Proposition

If (F', C') is (F, C)-flexible, then for every term t in $SPE_t(F, C)$ one can define a term $\tilde{t}$ in $T(F', C')$ that is equivalent to t and such that $|\tilde{t}|_{F'C'} = |t|_{FC}$ and $ht(\tilde{t}) \leq ht(t)$.

The main theorem of this section is:

Theorem

Let (F', C') be an (F, C)-flexible $\mathcal{S}$-signature. Every term t in $T(F, C)$ of size n is equivalent to a 3-balanced term t' in $T(F', C')$. This term can be constructed in time $O(n\log(n))$, if we assume that $\hat{q}$, f^q, $f^{q,q'}$ can be constructed in time $O(\max\{|q|, |q'|\})$.

Consequence of the above theoreom

Theorem

1. *Every graph of m-clique-width k is the value of a 3-balanced term of m-clique-width at most $2k$.*
2. *Every graph of rank-width k is the value of a 3-balanced term of rank-width at most $2k$.*
3. *Every graph of clique-width or NLC-width k is the value of a 3-balanced clique-width expression of clique-width or NLC-width at most $k \times 2^{k+1}$.*

Proof Sketch

- We show the method for rank-width.
- We let (R_k, C_k) denote the signature for graphs with vectors of colors in $\mathbb{B}^k$. We let (R'_{2k}, C_k) denote the signature for graphs with vectors of colors in $\mathbb{B}^k \cup \mathbb{B}^{2k}$.
- We must prove that the signature (R'_{2k}, C_k) is (R_k, C_k)-flexible.
- For that we show only how to construct f^q and $f^{q,q'}$ and the terms $\hat{q}$.
- Let $q = \otimes_1(x_1, \otimes_2(x_2, \ldots, \otimes_n x_n)) \ldots)$ where $\otimes_i = \otimes_{M_i,N_i,P_i}$.

Construction of $\hat{q}$

We let

$$\hat{q} = x_1 \otimes'_1 (x_2 \otimes'_2 (\dots (x_n \otimes'_n \emptyset_k)) \dots)$$

where $\otimes'_i = \otimes_{M'_i, N'_i, P'_i}$ with M'_i, N'_i are $(k \times 2k)$-matrices and P'_i are $(2k \times 2k)$-matrices and such that:

$$M'_i = \begin{pmatrix} M_i & 0_k \end{pmatrix} \quad N'_i = \begin{pmatrix} N_i & M_i.Q_i^T \end{pmatrix} \quad P'_i = \begin{pmatrix} P_i & 0_k \\ 0_k & I_k \end{pmatrix}$$

Construction of f^q

We let $\otimes^q = \otimes_{M,N,P}$ where:

$$M = \begin{pmatrix} O_k \\ I_k \end{pmatrix} \qquad N = \begin{pmatrix} I_k \\ 0_k \end{pmatrix} \qquad P = P_n.P_{n-1}.\dots.P_1$$

Construction of $f^{q,q'}$:general case

Let $q = x_1 \otimes_1 (x_2 \otimes_2 (\ldots(x_p \otimes_p u))\ldots)$ and $q' = x_{p+1} \otimes_{p+1} (x_{p+2} \otimes_{p+2} (\ldots(x_n \otimes_n u))\ldots)$ so that $q'' = x_1 \otimes_1 (x_2 \otimes_2 (\ldots(x_n \otimes_n u))\ldots)$. We let $f^{q,q'} = \otimes_{M,N,P}$ where:

$$M = \begin{pmatrix} O_k & 0_k \\ I_k & 0_k \end{pmatrix} \quad N = \begin{pmatrix} I_k & 0_k \\ 0_k & Q_{q'}^T \end{pmatrix} \quad P = \begin{pmatrix} Q_q & 0_k \\ 0_k & I_k \end{pmatrix}$$

where $Q_q = P_p.P_{p-1}.\ldots.P_1$ and $Q_{q'} = P_n.P_{n-1}.\ldots.P_{p+1}$.

The operations $\otimes^{q,q'}$ in particular cases

(1) $p = 1$, $n = 2$. Then $q = x_1 \otimes_1 u$, $q' = x_2 \otimes_2 u$, $\hat{q} = x_1$, $\widehat{q'} = x_2$. In this case $q'' = x_1 \otimes_1 (x_2 \otimes_2 u)$. We let $\widehat{q''} = x_1 \otimes^{q,q'} x_2$ with $\otimes^{q,q'} = \otimes_{M,N,P}$ where :

$$M = M_1.N_2^T \quad N = \begin{pmatrix} N_1 & M_1.P_2^T \end{pmatrix} \quad P = \begin{pmatrix} P_2.P_1 & M_2 \end{pmatrix}$$

It is clear that $\widehat{q''} \simeq \hat{q} \otimes^{q,q'} \widehat{q'}$.

The operations $\otimes^{q,q'}$ in particular cases

(2) $p = 1$, $n > 2$. Then $q = x_1 \otimes_1 u$,
$q' = x_2 \otimes_2 (x_3 \otimes_3 \ldots (x_n \otimes_n u)) \ldots)$, $\hat{q} = x_1 \otimes'_1 \emptyset_k$. In this case we can apply the construction of the general case which gives $\otimes^{q,q'}$.
In particular we have $Q_q = P$.
We have thus

$$\widehat{q''} = (G_1 \otimes'_1 \emptyset_k) \otimes^{q,q'} \widehat{q}'(G_2, \ldots, G_n).$$

But we can merge $G_1 \otimes'_1 \emptyset_k$ and $\otimes_{q,q'}$ into a single operation $\bar{\otimes}^{q,q'} = \otimes_{M,N,P}$ where:

$$M = \begin{pmatrix} M_1 & 0_k \end{pmatrix} \quad N = \begin{pmatrix} N_1 & M_1.Q_{q'}^T \end{pmatrix} \quad P = \begin{pmatrix} P_1 & 0_k \\ 0_k & I_k \end{pmatrix}$$

The operations $\otimes^{q,q'}$ in particular cases

(3) $p > 1$, $n = p + 1$. This case is similar. We have $\widehat{q'} = x_2 \otimes'_2 \emptyset_k$ and we get

$$\widehat{q''} = \hat{q} \otimes^{q,q'} (x_2 \otimes'_2 \emptyset_k)$$

and we can derive an operation $f^{q,q'}$ as wanted.

(4) By using the same technique as above, we can eliminate the constant $\emptyset_k$ from every $\hat{q}$. We have thus proved that (R'_{2k}, C_k) is (R_k, C_k)-flexible. ■

Plan

Contexts

Definition

A context is a term in $T(F, C \cup \{u\})$ having a single occurrence of the variable u (a nullary symbol). We denote by $Cxt(F, C)$ the set of contexts. We denote by Id the particular context u.

- Let s be a context and t be a term or a context, we denote by $s[t/u]$ the result of the substitution of t for u in s.
- $s \circ s' = s[s'/u]$, belonging to $Cxt(F, C)$ for s, s' in $Cxt(F, C)$.
- $s \bullet t = s[t/u]$, belonging to $T(F, C)$ for s in $Cxt(F, C)$ and t in $T(F, C)$.

M-clique-width

- The notion of m-clique-width is introduced by Courcelle and Twigg (STACS'07) for constraint labeling schemes.
- Let L be a finite set of colors. A multi-labeled graph is defined as $< V_G, edg_G, \delta_G >$ where $\delta_G(x) \subseteq L$ (a vertex may have zero, one or several colors).
- For $R \subseteq L \times L$, $g, h : L \to \mathscr{P}(L)$ and for G, H disjoint multi-labeled graphs, we let $K = G \otimes_{R,g,h} H$ where $V_K = V_G \cup V_H$ and :

$$\begin{aligned}
edg_K &= edg_G \cup edg_H \\
&\quad \cup \{xy \mid x \in V_G, y \in V_H, R \cap (\delta_G(x) \times \delta_H(y)) \neq \emptyset\}, \\
\delta_K(x) &= (g \circ \delta_G)(x) = \{a \mid a \in g(b), b \in \delta_G(x)\} \text{ if } x \in V_G, \\
\delta_K(x) &= (h \circ \delta_H)(x) \text{ if } x \in V_H.
\end{aligned}$$

Recolorings

Definition

For $h : \mathbb{B}^k \rightarrow \mathbb{B}^\ell$ and G a $\mathbb{B}^k$-colored graph, we let $Recol_h(G) = G' =< V_G, edg_G, \gamma_{G'} >$ where $\gamma_{G'} = h \circ \gamma_G$. Hence G' is a $\mathbb{B}^\ell$-colored graph.

Graph products

Definition

Let $f : \mathbb{B}^k \times \mathbb{B}^\ell \to \{0, 1\}$, let $g : \mathbb{B}^k \to \mathbb{B}^m$ and $h : \mathbb{B}^\ell \to \mathbb{B}^m$ be arbitrary mappings. For G, $\mathbb{B}^k$-colored and H, $\mathbb{B}^\ell$-colored, we let $K = G \otimes_{f,g,h} H$ be defined as follows, where, as usual, we assume $V_G \cap V_H = \emptyset$:

$$
\begin{aligned}
&V_K = V_G \cup V_H,\\
&edg_K = edg_G \cup edg_H \cup \{xy \mid x \in V_G, y \in V_H, f(\gamma_G(x), \gamma_H(y)) = 1\},\\
&\gamma_K(x) = (g \circ \gamma_G)(x) \text{ if } x \in V_G,\\
&\gamma_K(x) = (h \circ \gamma_H)(x) \text{ if } x \in V_H.
\end{aligned}
$$

Hence K is a $\mathbb{B}^m$-colored graph.

Multiple colorings (suite)

- For each $u \in \mathbb{B}^n$ we let **u** be a constant denoting a graph with one vertex colored by u and no edge.
- We denote by C_n the set of this constants. We use the constant $\emptyset_k$ to denote the empty $\mathbb{B}^k$-colored graph.
- We let B_n be the finite set of operations $Recol_h$, $\otimes_{f,g,h}$ where $k, \ell, m \in [n]$. It is not a loss of generality to assume $k, l, m \neq 0$.
- Every graph G is assigned a *default* $\mathbb{B}^1$*-coloring* with $\gamma_G(x) = (1)$ for every vertex x.
- Every term $t \in T(B_k, C_k)$ denotes a $\mathbb{B}^k$-colored graph for some $k \geq 1$, or actually the family of all graphs isomorphic to such a graph.
- The *Boolean-width* of a graph G, denoted by $Bwd(G)$ is the minimum n such that G is defined, up to isomorphism, by a term in $T(B_n, C_n)$.

Multiple colorings(suite)

The operations of B_n represent the "best we can do with n vertex colors" if we only allow:

- manipulations of vertex colors by quantifier-free operations, and done only in terms of existing vertex colors,
- creations of edges between two disjoint graphs based on vertex colors, independently on both sides.

What we do not allow:

- vertex color manipulations that depend on existence of loops incident with a vertex (actually we do not consider loops at all)
- edge deletions (and creations) of edges (in place of non-existing ones), like in the edge-complement operation.

Some comparisons

Proposition

For every uncolored undirected graph G,

1. $rwd(G) \leq cwd(G) \leq 2^{rwd(G)+1} - 1$.
2. $Bwd(G) \leq mcwd(G) \leq cwd(G) \leq 2^{Bwd(G)+1} \leq 2^{mcwd(G)+1}$.
3. $mcwd(G) \leq twd(G) + 3$.
4. $rwd(G) \leq 4 \times twd(G) + 2$.

$\mathbb{B}^k$-colored graphs as relational structures

- The meaning of $c_i(x) = true$ will be "x has color i".
- A $\mathbb{B}^n$-colored graph $G =< V_G, edg_G, \gamma_G >$ for $n \leq k$ is faithfully described by the relational structure with domain V_G that we will also denote by G:

$$< V_G, edg_G, c_{1G}, \ldots, c_{kG} > .$$

- Every relational structure of this form, and such that edg_G is symmetric and anti-reflexive ($edg_G(x, x)$ never holds) represents a $\mathbb{B}^k$-colored graph G.
- The notions of quantifier-free operations are well-known from the results of Courcelle and al.

FPT-algorithms

Proposition

1. *The operations $Recol_h$ are quantifier-free operations.*
2. *The operations $\otimes_{f,g,h}$ are expressible in terms of $\oplus$ and quantifier-free operations.*

Corollary

For each n, every MS graph property of a graph G can be decided in time $O(|t|)$, if G is the value of a given term $t \in T(B_n, C_n)$.