

Graph operations characterizing rank-width and balanced graph expressions *

Bruno Courcelle[†] Mamadou Moustapha Kanté[‡]

May 3, 2007

Abstract

Graph complexity measures like *tree-width*, *clique-width*, *NLC-width* and *rank-width* are important because they yield *Fixed Parameter Tractable* algorithms. Rank-width is based on ranks of adjacency matrices of graphs over $GF(2)$. We propose here algebraic operations on graphs that characterize rank-width. For algorithmic purposes, it is important to represent graphs by balanced terms. We give a unique theorem that generalizes several “balancing theorems” for tree-width and clique-width. New results are obtained for rank-width and a variant of clique-width, called *m-clique-width*.

1 Introduction

Graph complexity measures like *tree-width* [17], *clique-width* [6], *NLC-width* [18] and *rank-width* [16] are important parameters for the construction of polynomial algorithms. Every graph property expressible by a formula of MS (*Monadic Second-Order*) logic has a *Fixed Parameter Linear* algorithm if tree-width is taken as parameter and a *Fixed Parameter Cubic* algorithm if clique-width (equivalently rank-width) is taken as parameter. These results are proved in the books by Downey and Fellows [10] and by Flum and Grohe [11] for tree-width, by Courcelle and al. [5] with help of results by Oum and Seymour [15, 16] for rank-width and clique-width.

Clique-width and rank-width are equivalent in the sense that the same classes of undirected graphs have bounded clique-width and bounded rank-width. Clique-width has the advantage of having a definition in terms of very simple graph operations. Furthermore this definition is the basis of the construction of algorithms

*Research supported by the french ANR-project "Graph decompositions and algorithms (GRAAL)

[†]LaBRI, Universite Bordeaux 1, CNRS, 351 cours de la liberation 33405 Talence Cedex France.
Email: courcell@labri.fr.

[‡]LaBRI, Universite Bordeaux 1, CNRS, 351 cours de la liberation 33405 Talence Cedex France.
Email: mamadou.kante@labri.fr.

for checking MS graph properties in linear time in the size of the algebraic expressions defining the input graphs. Rank-width has the advantage of a good behavior with respect to vertex-minor inclusion, so that the class of graphs of rank-width at most k is characterized by finitely many excluded vertex-minors. Furthermore, the cubic-time algorithm that constructs for a given graph an algebraic expression of clique-width at most $2^{3k} - 1$ if the graph has clique-width at most k , is based on the decomposition underlying rank-width.

In this article we propose algebraic operations on graphs that characterize rank-width as follows:

a graph G has rank-width at most k if and only if (iff for short) it is the value of a term in $T(R_k, C_k)$

where R_k is a finite set of graph operations, C_k a finite set of constants, both depending on k .

In a few words, the operations are based on coloring vertices by sets of colors $\subseteq [k] := \{1, 2, \dots, k\}$, like in the variant of clique-width called *m-clique-width* (see definitions of Section 2 and [6, 7]), but vertex colors are manipulated by linear transformations on the $GF(2)$ vector space $\{0, 1\}^k$ rather than with set union over subsets of $\{1, \dots, k\}$. Furthermore, edges are created between two disjoint graphs by means of bilinear forms, taking the vectors of colors as arguments. It is thus somewhat natural that they can generate (exactly) the set of graphs of rank-width at most k since rank-width is based on ranks of $GF(2)$ matrices.

The operation that replaces anywhere a vertex color a by the color b , and the one that adds edges between any vertex colored by a and any vertex colored by b are typical examples of quantifier-free transformations. Quantifier-free transformations modify logical structures by redefining certain relations by quantifier-free formulas (see [9, 3] for graph algebras).

For algorithmic purposes, it is useful and sometimes crucial to represent graphs by a -balanced binary terms, i.e. trees of height at most $a(\log(n) + 1)$ where n is the number of nodes and a is a constant. This is the case for instance, of the labeling schemes considered in [8, 7]. Another practical use of balanced terms is the design of parallel algorithms. This is considered for example by Bodlaender to design parallel algorithms to construct minimum-width tree-decompositions of graphs or to solve some NP-complete problems [1, 2].

Therefore it is quite natural to ask whether, every graph of “width” k admits an a -balanced binary “decomposition” of width $f(k)$ for some fixed function f . It is known that every graph of tree-width k admits a 2-balanced binary tree-decomposition of width at most $3k + 2$ [1] and every graph of m -clique-width k admits a 6-balanced m -clique-width expression of width at most $2k$ [7]. We investigate the problem of a unified framework. We prove a general theorem covering several particular cases saying that every term in $T(F, C)$ representing a *binary structure* is equivalent to a 3-balanced one in $T(F', C')$, where (F', C') is a *binary signature* and $(F, C) \subseteq (F', C')$. For that we introduce a kind of generalization of the notion of associative and commutative operation, called *flexibility*.

The main results of this article are: an algebraic characterization of rank-width, a unified framework for “balancing theorems” with application to rank-width, clique-width, NLC-width and m-clique-width.

2 Notations and definitions

We denote by $[k]$ the set $\{1, \dots, k\}$. Graphs are finite, simple, loop-free, undirected unless otherwise specified. A graph G is defined as $\langle V_G, \text{edg}_G \rangle$ where $\text{edg}_G \subseteq V_G \times V_G$ is the symmetric adjacency relation. Without loss of generality we assume that V_G is always linearly ordered. This order will be used to represent edg_G by a square matrix over $GF(2)$.

A *sub-cubic tree* is a tree such that the degree of each node is at most 3. All logarithms are in base 2.

Let t be a rooted tree and $a \in \mathbb{R}$. We say that t is a -balanced if the *height* of t , i.e., the maximal distance of a leaf to the root, is at most $a(\log(n) + 1)$ where n is the number of nodes of t .¹

Let F be a set of binary functions and C be a set of constants. We denote by $T(F, C)$ the set of well-formed terms built with $F \cup C$. They will be discussed as colored directed and rooted ordered trees in the usual way. 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 $\text{Cxt}(F, C)$ 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 .

We define two binary operations on terms and contexts: $s \circ s' = s[s'/u]$, belonging to $\text{Cxt}(F, C)$ for s, s' in $\text{Cxt}(F, C)$ and $s \bullet t = s[t/u]$, belonging to $T(F, C)$ for s in $\text{Cxt}(F, C)$ and t in $T(F, C)$.

We now recall the definition of *rank-width*, a graph complexity measure introduced by Oum and Seymour in their investigations on recognition algorithms for graphs of bounded *clique-width* [16]. For an (R, C) -matrix $M = (m_{ij} \mid i \in R, j \in C)$ over a field F , if $X \subseteq R$, $Y \subseteq C$, we let $M[X, Y]$ denote the sub-matrix $(m_{ij} \mid i \in X, j \in Y)$. For a graph G , we let A_G be its adjacency (V_G, V_G) -matrix over $GF(2)$.

Cut-rank functions. Let $G = \langle V_G, \text{edg}_G \rangle$ be a graph. We define the *cut-rank* function ρ_G of G by letting $\rho_G(X) = rk(A_G[X, V_G \setminus X])$ for $X \subseteq V_G$, where rk is the matrix rank function. We let $\rho_G(\emptyset) = \rho_G(V_G) = 0$.

Rank-width. A *layout* of a graph G is a pair (T, f) of a sub-cubic tree T and a bijective function $f: V_G \rightarrow \{t \mid t \text{ is a node of degree 1 in } T\}$.

For an edge e of T , the connected components of $T \setminus e$ induce a bipartition of the set of nodes of degree 1 of T , hence a bipartition (X_e, Y_e) of the set of vertices of G . The width of an edge e of a layout (T, f) is $\rho_G(X_e) = \rho_G(Y_e)$. The width of a layout (T, f) is the maximum width over all edges of T . The *rank-width* of G , denoted by $\text{rwd}(G)$, is the minimum width over all layouts of G .

¹This definition is meaningful in the case $n = 1$.

The notions of rank-width and of clique-width are equivalent in the sense that a class of graphs has bounded rank-width iff it has bounded clique-width. Oum has given in [15] a $O(n^3)$ -time algorithm that reports that a graph has rank-width at least $k + 1$ or outputs a layout of width at most $3k - 1$. This has been improved in [12] which gives a cubic-time algorithm that outputs a layout of width k if the graph has rank-width k . But if we want to solve problems definable in MS on graphs of bounded rank-width, we need to transform the layout into a clique-width expression (see [16]) and, after that, to use techniques by Courcelle and al. [5]. In this paper, we propose an algebraic characterization of rank-width, which will allow us to solve MS definable problems without transforming the layout into a clique-width expression. This is important because the transformation of a layout of width k may give a $(2^{k+1} - 1)$ -clique-width expression. The exponent 2^{k+1} is part of the large size of constants in FPT algorithms.

Proposition 2.1 [16, 7, 6, 13] *For every undirected graph G ,*

$$\begin{aligned} (1) \quad rwd(G) \leq cwd(G) \leq 2^{rwd(G)+1} - 1 & \quad (3) \quad mcwd(G) \leq twd(G) + 3 \\ (2) \quad mcwd(G) \leq cwd(G) \leq 2^{mcwd(G)+1} & \quad (4) \quad rwd(G) \leq 4 \times twd(G) + 2 \end{aligned}$$

Here twd , cwd and $mcwd$ denote respectively *tree-width* [17], *clique-width* [6] and *m-clique-width* (we recall below the definition of m-clique-width [7]).

M-clique-width. Let L be a finite set of colors. A *multi-colored graph* is a triple $\langle V_G, edg_G, \delta_G \rangle$ consisting of a graph $\langle V_G, edg_G \rangle$ and a mapping δ_G associating with each x in V_G the set of its colors, a subset of L . A vertex may have zero, one or several colors.

The following constants will be used: for $A \subseteq L$ we let \mathbf{A} be a constant denoting the graph G with single vertex x and $\delta_G(x) = A$. We write $\mathbf{A}(x)$ if we need to specify the vertex x . The following binary operations will be used: for $R \subseteq L \times L$, for *recolorings* $g, h : L \rightarrow 2^L$ and for multi-colored graphs G and H we define $K = G \otimes_{R,g,h} H$ if G and H are disjoint (otherwise we replace H by a disjoint copy) where

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

As in the operations by Wanke [18] these operations add edges between two disjoint graphs, that are the two arguments of (many) binary operations. 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 .

We let F_L be the set of all binary operations $\otimes_{R,g,h}$ and C_L be the set of constants $\{\mathbf{A} \mid A \subseteq L\}$. Every term t in $T(F_L, C_L)$ denotes a multi-colored graph $val(t)$ with colors in L , and every multi-colored graph G is the value of such a term for large

enough L . To simplify the notation, we will write F_k and C_k if $L = [k]$. We let $mcwd(G)$ be the minimum k such that G is the value of a term $t \in T(F_k, C_k)$ and call this number the m -clique-width of G .

3 Vectorial colorings and rank-width

Handling multiple colorings of vertices with k colors is clearly the same thing as handling colorings with colors in $\{0, 1\}^k$. Let $k \geq 1$ and $\mathbb{B} = \{0, 1\}$. A \mathbb{B}^k -coloring of a graph G is a mapping $\gamma: V_G \rightarrow \mathbb{B}^k$ with no constraint on the values of γ for neighbor vertices. 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 $G = \langle V_G, \text{edg}_G, \gamma_G \rangle$ where γ_G is a \mathbb{B}^k -coloring of $\langle V_G, \text{edg}_G \rangle$. The empty \mathbb{B}^k -colored graph is denoted by \emptyset_k . (This constant can be eliminated from expressions by Remark 3.1). We define some operations on these graphs.

A mapping $h: \mathbb{B}^k \rightarrow \mathbb{B}^\ell$ is *linear* if for some $(k \times \ell)$ -matrix and all row-vectors $u \in \mathbb{B}^k$ we have $h(u) = u.N$. We say that h is described by N . A mapping $f: \mathbb{B}^k \times \mathbb{B}^\ell \rightarrow \mathbb{B}$ is said *bilinear* if for some $(k \times \ell)$ -matrix and all row-vectors $u \in \mathbb{B}^k$, $v \in \mathbb{B}^\ell$ we have $f(u, v) = u.M.v^T$ where v^T indicates *transposition* of the row-vector v (we say that f is described by M).

With a \mathbb{B}^k -colored graph $G = \langle V_G, \text{edg}_G, \gamma_G \rangle$ we associate the $(V_G \times V_G)$ -adjacency (symmetric) matrix A_G and the $V_G \times [k]$ -color matrix Γ_G , the row vectors of which are the vectors $\gamma_G(x)$ in \mathbb{B}^k for x in V_G . We define the *color-rank* of G as the rank of Γ_G and we denote it by $crk(G)$. Clearly, $crk(G) \leq k$ if G is \mathbb{B}^k -colored.^{2 3}

Linear recolorings. For $h: \mathbb{B}^k \rightarrow \mathbb{B}^\ell$ a linear mapping and G a \mathbb{B}^k -colored graph, we let $\text{Recol}_h(G) = H = \langle V_G, \text{edg}_G, \gamma_H \rangle$ where $\gamma_H = h \circ \gamma_G$. Hence $\gamma_H = \Gamma_G.N$ and H is a \mathbb{B}^ℓ -colored graph. If h and h' are linear recolorings, described respectively by N and N' , then $h \circ h'$ is linear and is described by $N'.N$.

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

$$\begin{aligned} V_K &= V_G \cup V_H, \\ \text{edg}_K &= \text{edg}_G \cup \text{edg}_H \cup \{xy \mid x \in V_G, y \in V_H, f(\gamma_G(x), \gamma_H(y)) = 1\}, \\ \gamma_K(x) &= (g \circ \gamma_G)(x) \text{ if } x \in V_G, \quad \gamma_K(x) = (h \circ \gamma_H)(x) \text{ if } x \in V_H, \end{aligned}$$

where f, g, h are described respectively by M, N, P . Hence K is a \mathbb{B}^m -colored graph. We order the graph $K = G \otimes_{f,g,h} H$ by preserving the orderings of V_G and V_H and letting $x < y$ for $x \in V_G$ and $y \in V_H$. We will use the notation $\otimes_{M,N,P}$ instead of $\otimes_{f,g,h}$.

²The color-rank of G should not be confused with its *rank*. All ranks are relative to $GF(2)$.

³A graph $G = \langle V_G, \text{edg}_G \rangle$ is made canonically into a \mathbb{B}^k -colored graph for each k , with $\gamma_G(x) = (0, \dots, 0)$ for each x .

Constants. We will use $\mathbf{1}$ to denote the graph with a single vertex with its \mathbb{B}^1 -coloring by (1). In order to avoid the use of recolorings, and to deal only with constants and binary operations, we will also use constants for the graphs $Recol_h(\mathbf{1})$ where h ranges over linear recolorings defined by 1-row matrices $N = u \in \mathbb{B}^k$. Such constants will be denoted by \mathbf{u} . We use C_k to denote the set of constants \mathbf{u} for $u \in \mathbb{B}^\ell, \ell \leq k$.

Remark 3.1 *We have*

$$\begin{aligned} G \otimes_{M,N,P} H &= H \otimes_{M^T, P, N} G, & Recol_Q(G) \otimes_{M,N,P} Recol_{Q'}(H) &= G \otimes_{QM, Q'^T, QN, Q'^P} H, \\ G \otimes_{M,N,P} \mathbf{0}_k &= Recol_N(G), & Recol_Q(G \otimes_{M,N,P} H) &= G \otimes_{M, NQ, PQ} H \end{aligned}$$

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. We can assume with Remark 3.1 that a term t in $T(R_n, C_n)$ is written with the binary operations $\otimes_{M,N,P}$ and the constants \mathbf{u} where $u \in \mathbb{B}^1 \cup \dots \cup \mathbb{B}^n$.

Proposition 3.2 *1. The operations $Recol_N$ are quantifier-free operations.*

2. The operations $\otimes_{M,N,P}$ are expressible in terms of \oplus and quantifier-free operations.

Corollary 3.3 *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 $t \in T(R_n, C_n)$.*

Theorem 3.4 *A graph G has rank-width at most n iff it is the value of a term in $T(R_n, C_n)$.*

For the ‘‘If’’ direction, we let G be defined by a term t in $T(R_n, C_n)$ (t has its root colored by a binary operation $\otimes_{M,N,P}$). We take the syntactic tree of t as a layout of G . It is sufficient to prove the claim below to prove that the rank-width of this layout is at most n .

Claim 3.5 *([4]) 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: $A_G[V_H, V_G - V_H] = \Gamma_H \cdot B$ and $\Gamma_{G[V_H]} = \Gamma_H \cdot C$ for some matrices B and C , and, $rk(A_G[V_H, V_G - V_H]) \leq n$.*

For the converse, we prove some technical lemmas. We write $G = H \otimes_M K$ instead of $H \otimes_{M,N,P} K$ if we do not care about the coloring of G but only of its vertices and edges.

Lemma 3.6 *([4]) 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.*

Proposition 3.7 ([4]) 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 $G = \text{Recol}_N(H) \otimes_M \text{Recol}_P(K)$.

Lemma 3.8 ([4]) 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

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

We can thus prove the following proposition (the “only if” direction of Theorem 3.4).

Proposition 3.9 ([4]) Every graph of rank-width at most n is the value of a term in $T(R_n, C_n)$.

4 A general framework for establishing balancing theorems

It is known that every graph of tree-width k has a 2-balanced binary tree-decomposition of width at most $3k + 2$ [1] and every graph of m -clique-width k has a 6-balanced m -clique-width expression of width at most $2k$ [7]. We will propose a general framework for establishing *balancing theorems*. This will allow us to prove similar theorems for rank-width, clique-width and NLC-width. Our general framework combines two ideas.

The first idea, coming from [8] consists in introducing binary operations \circ and \bullet on terms and contexts representing respectively the composition of the unary functions associated with two contexts and the evaluation of such a function for an argument defined by a term. We use a result of [8] showing that every term t in $T(F, C)$ can be replaced by an equivalent 3-balanced *special term* t^b written with \circ , \bullet and the constant Id (the trivial context defining the identity). This construction makes no assumption on the algebraic properties of the signature (F, C) .

The second idea introduces a kind of generalization of the notion of an associative and commutative operation. It concerns a subsignature (F, C) of (F', C') . Roughly speaking if f in F is not associative, hence if we do not have $f(x, f(y, z)) = f(f(x, y), z)$ for $f \in F$ then we require that $f(x, f(y, z)) = f'(f(x, y), z)$ for some $f' \in F'$. We say that (F', C') is (F, C) -flexible if this condition and similar ones hold. This condition makes it possible to eliminate from a term written with F, C, \circ, \bullet and Id the operations \circ, \bullet and the constant Id and somehow, to express them in terms of operations of F' .

The idea is to associate with a context $c \in \text{Ctx}(F, C)$ an object m_c denoted by a term \tilde{c} in $T(F', C')$ and a function $f^c \in F'$ such that $c \bullet t$ is equivalent to $f^c(\tilde{c}, t)$. For two contexts c and c' we have (by the condition of flexibility) an operation $f^{c, c'}$

in F' such that $c \circ \widetilde{c'}$ is equivalent to $f^{c,c'}(\widetilde{c}, \widetilde{c'})$. It follows that a special term t over F, C can be transformed into an equivalent term in $T(F', C')$ of no larger height.

By combining the two constructions, we can transform a term $t \in T(F, C)$ into an equivalent 3-balanced term in $T(F', C')$.

In our applications to graph operations we will apply this to a signature (F, C) using k colors (e.g. (R_k, C_k)) corresponding to rank-width at most k and prove that some finite $(F', C') \supseteq (F, C)$ is (F, C) -flexible. This technique also applies to branch-width and tree-width [14].

Let \mathcal{S} be a countable set whose elements are called *sorts*. A *binary \mathcal{S} -signature* is a pair (F, C) where F is a set of binary function symbols, each of them having a type $s_1 \times s_2 \rightarrow s$ where $s_1, s_2, s \in \mathcal{S}$, and C is a set of nullary symbols, each of them having a type s in \mathcal{S} . A nullary symbol is called a *constant*. We say that a binary \mathcal{T} -signature (F, C) is a sub-signature of (F', C') if $\mathcal{T} \subseteq \mathcal{S}$, $F \subseteq F'$, $C \subseteq C'$ and the types of the elements of F and C are the same for (F', C') and for (F, C) . Let $\sigma : F \cup C \rightarrow \mathcal{S}$ where $\sigma(f) = s$ if f is a constant of type s or a binary function of type $s_1 \times s_2 \rightarrow s$. We define the type of a term $t \in T(F, C)$ as $\sigma(r_t)$, where r_t is its first symbol (the one at the root of its syntactic tree).

Special Terms. 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:

$$\begin{aligned} S_t &:= S_c \bullet S_t \cup f(S_t, S_t) \cup b \\ S_c &:= S_c \circ S_c \cup f(S_t, S_c) \cup f(S_c, S_t) \cup f(S_t, Id) \cup f(Id, S_t) \end{aligned}$$

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 $Id \notin S_t \cup S_c$. The notions of context and the operations \circ and \bullet extend in presence of sorts. We have actually several operations \circ , \bullet and several constants Id depending on sorts, but we will overlook this technical point.

For terms t in $SPE_t(F, C) \cup SPE_c(F, C)$ we denote by $|t|_{FC}$ the number of occurrences of symbols from $F \cup C$, by $|t|_0$ the number of occurrences of \circ and \bullet , and, by $|t|_{Id}$ the number of occurrences of Id .

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 $Cxt(F, C) - \{Id\}$.

A more careful proof than the one of [8, Theorem 1] gives the following result.

Theorem 4.1 ([4]) *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 $n = |t|$.*

Comb-term. Let $X_{n+1} = \{x_1, \dots, x_{n+1}\}$. A *comb-term* is a term in $T(F, X_{n+1})$ of the form $q = f_1(x_1, f_2(x_2, \dots, f_n(x_n, x_{n+1}))) \dots$. It contains no constant. We denote it also by $q(x_1, \dots, x_n, x_{n+1})$ in order to specify the list of variables, in the order in which they occur.

Commutativity. A binary \mathcal{S} -signature (F, C) is *commutative* with respect to a class of algebras \mathcal{C} (that will be implicitly assumed in most cases) if for every $f \in F$

there exists a function \tilde{f} in F such that

$$\tilde{f}_M(x, y) = f_M(y, x) \quad (1)$$

for all $M \in \mathcal{C}$, all $x, y \in D_M$.

Comb-decomposition. The *comb-decomposition* of a term $t \in T(F, C) - C$ is the unique writing of t as $q(t_1, \dots, t_n, b)$ where $q(x_1, \dots, x_{n+1})$ is a comb-term, $b \in C$ and $t_i \in T(F, C)$.

The following definition makes sense only if F is commutative. Let $c \in \text{Cxt}(F, C) - \{Id\}$. Let us define by structural induction on c a comb-term $q(x_1, \dots, x_n, u)$ for some n , and a sequence (t_1, \dots, t_n) of terms in $T(F, C)$ such that $c \simeq q(t_1, \dots, t_n, u)$ and \simeq denotes the equivalence of terms with respect to the intended class \mathcal{C} of algebras (for which F is commutative).

We define $\text{Comb}(c)$ and $\text{seq}(c)$ as follows:

1. $\text{Comb}(c) = f(x_1, u)$ and $\text{seq}(c) = (t)$ if $c = f(t, Id)$.
2. $\text{Comb}(c) = \text{Comb}(c')$ and $\text{seq}(c) = \text{seq}(c')$ if $c = f(c_1, t)$ and $c' = \tilde{f}(t, c_1)$.
3. $\text{Comb}(c) = f(x_1, q(x_2, \dots, x_{n+1}, u))$ and $\text{seq}(c) = (t).\text{seq}(c')$ if $c = f(t, c_1)$, $c_1 \neq Id$ and $\text{Comb}(c') = q(x_1, \dots, x_n, u)$.

These definitions actually extend to contexts defined as terms in $\text{SPE}_c(F, C)$. We need only add one clause to (1)-(3):

- (4) If $c = c' \circ c''$ (so that $c' \neq Id$, $c'' \neq Id$) if $\text{Comb}(c') = q'(x_1, \dots, x_p, u)$ and $\text{Comb}(c'') = q''(x_1, \dots, x_n, u)$ then we define

$$\begin{aligned} \text{Comb}(c) &\text{ as } q'(x_1, \dots, x_p, q''(x_{p+1}, \dots, x_{n+p}, u)) \\ \text{and } \text{seq}(c) &\text{ as } \text{seq}(c').\text{seq}(c''). \end{aligned}$$

In the following, we will extend the equivalence relation \simeq by letting $\text{Eval}(t) \simeq t$ and $\text{Eval}(c) \simeq c$ for terms in $\text{SPE}_t(F, C) \cup \text{SPE}_c(F, C)$.

Flexibility. We let (F', C') and (F, C) be two binary signatures such that $(F, C) \subseteq (F', C')$. We let \mathcal{C} be a set of (F', C') -algebras. All equivalences of terms and contexts denoted by \simeq will be considered with respect to \mathcal{C} . We say that (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

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

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 .

Proposition 4.2 ([4]) *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 $ht(\tilde{t}) \leq ht(t)$.*

Combining Theorem 4.1 and Proposition 4.2 we get the following theorem:

Theorem 4.3 ([4]) *Let (F', C') be an (F, C) -flexible 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 $O(\max\{|q|, |q'|\})$.*

We can apply this theorem to m -clique-width, rank-width, clique-width and NLC-width. It will suffice to check the flexibility condition for appropriate super-signatures of the signatures that define m -clique-width, rank-width, clique-width and NLC-width.

Theorem 4.4 ([4, 7])

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 $k \times 2^{k+1}$.*

References

- [1] H.L. Bodlaender, NC-algorithms for graphs with small tree-width, in *Graph-Theoretic Concepts in Computer Science (WG'88)*, volume 344 of *LNCS*, pages 1-10. Springer-Verlag, 1989.
- [2] H.L. Bodlaender and T. Hagerup, Parallel algorithms with optimal speedup for bounded tree-width, *SIAM J. Comput.*, 27(1998), 1725-1746.

- [3] A. Blumensath and B. Courcelle, Recognizability, hypergraph operations and logical types, *Information and Computation*, 204(2006), 853-919.
- [4] B. Courcelle and M.M. Kanté, Multiple Colorings : graph operations characterizing rank-width and balanced graph expressions. Available on http://www.labri.fr/perso/courcell/courcelle_kante07.pdf
- [5] B. Courcelle and J.A. Makowsky and U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, *Theory of Computing Systems*, 33(2000), 125-150.
- [6] B. Courcelle and S. Olariu, Upper bounds to the clique-width of graphs, *Discrete Applied Mathematics*, 101(2000), 77-114.
- [7] B. Courcelle and A. Twigg, Compact Forbidden-set Routing, *Proceedings of STACS'07*, Aachen, 2007, volume 4393 of LNCS, pages 37-48.
- [8] B. Courcelle and R. Vanicat, Query efficient implementation of graphs of bounded clique-width, *Discrete Applied Mathematics*, 131(2003), 129-150.
- [9] B. Courcelle and P. Weil, The recognizability of sets of graphs is a robust property, *Theor. Comput. Sci.*, 342(2005), 173-228.
- [10] R. Downey and M. Fellows, *Parameterized complexity*, Springer-verlag, 1999.
- [11] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer Verlag, 2006.
- [12] P. Hliněný and S. Oum, Finding Branch-decompositions and Rank-decompositions, *submitted*, 2007.
- [13] M.M. Kanté, Vertex-minor reductions can simulate edge contractions, *submitted*, 2006.
- [14] M.M. Kanté, Manuscript in preparation, 2007.
- [15] S. Oum, Approximating rank-width and Clique-width Quickly, preprint 2006. Expanded version of the article in *Graph-Theoretic Concepts in Computer Science* (Metz, 2005), volume 3787 of LNCS, pages 49-58. Springer, 2005.
- [16] S. Oum and P. Seymour, Approximating clique-width and branch-width, *J. Combin Theory, Ser B*, 96(2006), 514-528.
- [17] N. Robertson and P. Seymour, Graph minors V : excluding a planar graph, *J. Combin. Theory (B)*, 41(1986), 92-114.
- [18] E. Wanke, k -NLC graphs and Polynomial algorithms, *Discrete Applied Mathematics*, 54(1994), 251-266.