

Graph Operations on Clique-Width Bounded Graphs

Frank Gurski*

February 1, 2008

Abstract

In this paper we survey the behavior of various graph operations on the graph parameters clique-width and NLC-width. We give upper and lower bounds for the clique-width and NLC-width of the modified graphs in terms of the clique-width and NLC-width of the involved graphs. Therefor we consider the binary graph operations join, co-join, sum, difference, products, corona, substitution, and 1-sum, and the unary graph operations quotient, subgraph, edge complement, bipartite edge complement, power of graphs, switching, local complementation, edge addition, edge subdivision, vertex identification, and vertex addition.

Keywords: clique-width, NLC-width, graph operations

1 Introduction

The clique-width of a graph is defined by a composition mechanism for vertex-labeled graphs [CO00]. The operations are the vertex disjoint union, the addition of edges between vertices controlled by a label pair, and the relabeling of vertices. The clique-width of a graph G is the minimum number of labels needed to define it. The NLC-width of a graph is defined by a composition mechanism similar to that for clique-width [Wan94]. Every graph of clique-width at most k has NLC-width at most k and every graph of NLC-width at most k has clique-width at most $2k$ [Joh98]. The only essential difference between the composition mechanisms of clique-width bounded graphs and NLC-width bounded graphs is the addition of edges. In an NLC-width composition the addition of edges is combined with the union operation. This union operation applied to two graphs G and J is controlled by a set S of label pairs such that for every pair $(a, b) \in S$ all vertices of G labeled by a will be connected with all vertices of J labeled by b . Both concepts are useful, because it is sometimes much more comfortable to use NLC-width expressions instead of clique-width expressions and vice versa, respectively.

Clique-width and NLC-width bounded graphs are particularly interesting from an algorithmic point of view. A lot of NP-complete graph problems can be solved in polynomial time for graphs of bounded clique-width. For example, all graph properties which are expressible in monadic second order logic with quantifications over vertices and vertex sets (MSO_1 -logic) are decidable in linear time on clique-width bounded graphs [CMR00]. Furthermore, there are also a lot of NP-complete graph problems which are not expressible in MSO_1 -logic like Hamiltonicity, partition problems, and bounded degree subgraph problems but which can also be solved in polynomial time on clique-width bounded graphs [Wan94, EGW01, KR03, Tod03, GW06].

*Heinrich-Heine Universität Düsseldorf, Institute of Computer Science, D-40225 Düsseldorf, Germany E-Mail: gurski@acs.uni-duesseldorf.de

Distance hereditary graphs have clique-width at most 3 [GR00]. The set of all graphs of clique-width at most 2 or NLC-width 1 is the set of all labeled co-graphs. Brandstädt et al. have analyzed the clique-width of graphs defined by forbidden one-vertex extensions of P_4 [BDLM02]. The clique-width and NLC-width of permutation graphs, interval graphs, grids and planar graphs is not bounded [GR00]. An arbitrary graph with n vertices has clique-width at most $n - r$, if $2^r < n - r$, and NLC-width at most $\lceil \frac{n}{2} \rceil$ [Joh98]. Every graph of tree-width¹ at most k has clique-width at most $3 \cdot 2^{k-1}$ [CR01]. In [GW00], it is shown that every graph of clique-width or NLC-width k which does not contain the complete bipartite graph $K_{n,n}$ for some $n > 1$ as a subgraph has tree-width at most $3k(n - 1) - 1$. The recognition problem for graphs of clique-width or NLC-width at most k is still open for $k \geq 4$ and $k \geq 3$, respectively. Clique-width of at most 3 is decidable in polynomial time [CHL⁺00]. NLC-width of at most 2 is decidable in polynomial time [Joh00]. Clique-width of at most 2 and NLC-width 1 is decidable in linear time [CPS85]. Minimizing NLC-width and minimizing clique-width is NP-complete [GW05, GW07, FRRS05, FRRS06]. The clique-width of tree-width bounded graphs is computable in linear time [EGW03].

Graph operation can be used to characterize sets of graphs by forbidden graphs. Graphs of bounded tree-width, which are less powerful than graphs of bounded clique-width, are closed under taking minors and characterized by a finite set of forbidden minors [RS85]. Oum and Seymour introduced in [OS06] the rank-width of graphs, which is defined independently of vertex labels, but which is shown to be as powerful as clique-width. In [Oum05] it is shown that rank-width bounded graphs are closed under taking local complementation which leads to a characterization of graphs of rank-width at most k by forbidden vertex-minors (i.e. taking induced subgraphs and local complementations). It is still open if there exists a graph operation that does not increase the NLC-width or clique-width and which can be used to characterize graphs of NLC-width at most k or clique-width at most k by a set of forbidden subgraphs. Thus we want to study the behavior of graph operations on the NLC-width and clique-width of graphs more precisely.

Further we know from [OS06] that we can compute a clique-width $f(k)$ -expression for every given graph in polynomial time. Since $f(k)$ can be exponential in the clique-width of the given graph and nearly all algorithms for hard graph problems on clique-width bounded graphs have a running time exponential in the number of used labels, it is important to find expressions using few labels. In order to deal with this problem, our survey shows for various graph operations f how to construct an expression for graph $f(G)$ from an expression for some graph G .

This paper is organized as follows. In Section 2, we recall the definitions of clique-width and NLC-width. In Section 3, we give an overview on the behavior of the binary operations join, co-join, sum, difference, products, corona, substitution, and 1-sum on the clique-width and NLC-width of a given graph. In Section 4, we consider the latter problem for the unary graph operations quotient, subgraph, edge complement, bipartite edge complement, power of graphs, switching, local complementation, edge addition, edge subdivision, vertex identification, and vertex addition. In Section 5, we show how these results can be used to give upper bounds on the clique-width and NLC-width of graph classes and we compare the shown bounds within a table.

¹See [Bod98] for definition and an overview on tree-width.

2 Preliminaries

Let $[k] := \{1, \dots, k\}$ be the set of all integers between 1 and k . We work with finite undirected labeled graphs $G = (V_G, E_G, \text{lab}_G)$, where V_G is a finite set of *vertices* labeled by some mapping $\text{lab}_G : V_G \rightarrow [k]$ and $E_G \subseteq \{\{u, v\} \mid u, v \in V_G, u \neq v\}$ is a finite set of *edges*. A labeled graph $J = (V_J, E_J, \text{lab}_J)$ is a *subgraph* of G if $V_J \subseteq V_G$, $E_J \subseteq E_G$ and $\text{lab}_J(u) = \text{lab}_G(u)$ for all $u \in V_J$. J is an *induced subgraph* of G if additionally $E_J = \{\{u, v\} \in E_G \mid u, v \in V_J\}$. The labeled graph consisting of a single vertex labeled by $a \in [k]$ is denoted by \bullet_a . For a vertex $v \in V_G$ we denote by $N_G(v)$ the set of all vertices which are adjacent to v in G , i.e. $N_G(v) = \{w \in V_G \mid \{v, w\} \in E_G\}$. $N_G(v)$ is called the set of all *neighbors* of v in G or *neighborhood* of v in G . The *degree* of a vertex $v \in V_G$, denoted by $\deg_G(v)$, is the number of neighbors of vertex v in G , i.e. $\deg_G(v) = |N_G(v)|$.

The notion of clique-width for labeled graphs is defined by Courcelle and Olariu in [CO00].

Definition 2.1 (CW_k, clique-width, [CO00]) *Let k be some positive integer. The class CW_k of labeled graphs is recursively defined as follows.*

1. *The single vertex graph \bullet_a for some $a \in [k]$ is in CW_k.*
2. *Let $G = (V_G, E_G, \text{lab}_G) \in \text{CW}_k$ and $J = (V_J, E_J, \text{lab}_J) \in \text{CW}_k$ be two vertex disjoint labeled graphs, then $G \oplus J := (V', E', \text{lab}')$ defined by $V' := V_G \cup V_J$, $E' := E_G \cup E_J$, and*

$$\text{lab}'(u) := \begin{cases} \text{lab}_G(u) & \text{if } u \in V_G \\ \text{lab}_J(u) & \text{if } u \in V_J \end{cases}, \forall u \in V'$$

is in CW_k.

3. *Let $a, b \in [k]$ be two distinct integers and $G = (V_G, E_G, \text{lab}_G) \in \text{CW}_k$ be a labeled graph, then*

(a) $\rho_{a \rightarrow b}(G) := (V_G, E_G, \text{lab}')$ defined by

$$\text{lab}'(u) := \begin{cases} \text{lab}_G(u) & \text{if } \text{lab}_G(u) \neq a \\ b & \text{if } \text{lab}_G(u) = a \end{cases}, \forall u \in V_G$$

is in CW_k and

(b) $\eta_{a,b}(G) := (V_G, E', \text{lab}_G)$ defined by

$$E' := E_G \cup \{\{u, v\} \mid u, v \in V_G, u \neq v, \text{lab}(u) = a, \text{lab}(v) = b\}$$

is in CW_k.

The clique-width of a labeled graph G is the least integer k such that $G \in \text{CW}_k$.

The notion of NLC-width² of labeled graphs is defined by Wanke in [Wan94].

Definition 2.2 (NLC_k, NLC-width, [Wan94]) *Let k be some positive integer. The class NLC_k of labeled graphs is recursively defined as follows.*

²The abbreviation NLC results from the *node label controlled* embedding mechanism originally defined for graph grammars.

1. The single vertex graph \bullet_a for some $a \in [k]$ is in NLC_k .
2. Let $G = (V_G, E_G, lab_G) \in NLC_k$ and $J = (V_J, E_J, lab_J) \in NLC_k$ be two vertex disjoint labeled graphs and $S \subseteq [k]^2$ be a relation, then $G \times_S J := (V', E', lab')$ defined by $V' := V_G \cup V_J$,

$$E' := E_G \cup E_J \cup \{\{u, v\} \mid u \in V_G, v \in V_J, (lab_G(u), lab_J(v)) \in S\},$$

and

$$lab'(u) := \begin{cases} lab_G(u) & \text{if } u \in V_G \\ lab_J(u) & \text{if } u \in V_J \end{cases}, \forall u \in V'$$

is in NLC_k .

3. Let $G = (V_G, E_G, lab_G) \in NLC_k$ and $R : [k] \rightarrow [k]$ be a function, then $\circ_R(G) := (V_G, E_G, lab')$ defined by $lab'(u) := R(lab_G(u))$, $\forall u \in V_G$ is in NLC_k .

The NLC-width of a labeled graph G is the least integer k such that $G \in NLC_k$.

An expression X built with the operations $\bullet_a, \oplus, \rho_{a \rightarrow b}, \eta_{a,b}$ for integers $a, b \in [k]$ according to Definition 2.1 is called a *clique-width k -expression*. An expression X built with the operations $\bullet_a, \times_S, \circ_R$ for $a \in [k]$, $S \subseteq [k]^2$, and $R : [k] \rightarrow [k]$ according to Definition 2.2 is called an *NLC-width k -expression*. The graph defined by expression X is denoted by $\text{val}(X)$.

Every clique-width k -expression can be transformed into an equivalent NLC-width k -expression and every NLC-width k -expression can be transformed into an equivalent clique-width $2k$ -expression [Joh98].

Example 2.3

1. The following two clique-width expressions X_1 and X_2 define the labeled graphs G_1 and G_2 in Figure 1.

$$X_1 = \eta_{1,2}((\rho_{2 \rightarrow 1}(\eta_{1,2}(\bullet_1 \oplus \bullet_2))) \oplus \bullet_2)$$

$$X_2 = \rho_{1 \rightarrow 2}(\eta_{2,3}(((\eta_{1,2}(\bullet_1 \oplus \bullet_2)) \oplus (\eta_{1,2}(\bullet_1 \oplus \bullet_2))) \oplus \bullet_3))$$

2. The following two NLC-width expressions X_3 and X_4 also define the labeled graphs G_1 and G_2 in Figure 1.

$$X_3 = (\bullet_1 \times_{\{(1,1)\}} \bullet_1) \times_{\{(1,2)\}} \bullet_2$$

$$X_4 = \circ_{\{(1,2),(2,2),(3,3)\}}(((\bullet_1 \times_{\{(1,2)\}} \bullet_2) \times_{\emptyset} (\bullet_1 \times_{\{(1,2)\}} \bullet_2)) \times_{\{(2,3)\}} \bullet_3)$$

Every NLC-width k -expression X has by its recursive definition a tree structure that is called the *NLC-width k -expression-tree* T for X . T is an ordered rooted tree whose leaves correspond to the vertices of graph $\text{val}(X)$ and the inner nodes³ correspond to the operations

³To distinguish between the vertices of (non-tree) graphs and trees, we simply call the vertices of trees *nodes*.

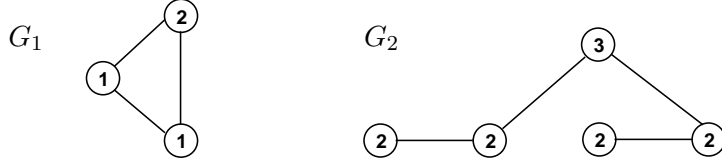


Figure 1: Two labeled graphs G_1 and G_2 defined by expressions X_1 and X_3 and by expressions X_2 and X_4 , respectively.

of X , see [GW00]. In the same way we define the clique-width k -expression-tree for every clique-width k -expression, see [EGW03].

If integer k is known from the context or irrelevant for the discussion, then we sometimes use the simplified notion *expression-tree* for the notion k -expression-tree.

For some node u of expression-tree T , let $T(u)$ be the subtree of T rooted at u . Note that tree $T(u)$ is always an expression-tree. The expression $X(u)$ defined by $T(u)$ can simply be determined by traversing the tree $T(u)$ starting from the root, where the left children are visited first. $X(u)$ defines a (possibly) relabeled induced subgraph $G(u)$ of G .

For an inner node v of some expression-tree T and a leaf u of $T(v)$ we define by $\text{lab}(u, G(v))$ the label of that vertex of graph $G(v)$ that corresponds to u .

A node u of T is called a *predecessor* of a node u' of T if u' is on a path from u to a leaf. A node u of T is called the *least common predecessor* of two nodes u_1 and u_2 if u is a predecessor of both nodes u_1, u_2 , and no child of u is a predecessor of u_1, u_2 .

3 Binary Operations

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two vertex disjoint graphs and let f be some binary graph operation which creates a new graph $f(G_1, G_2)$ from G_1 and G_2 . We next consider the NLC-width and clique-width of graph $f(G_1, G_2)$ with respect to the NLC-width or clique-width of G_1 and G_2 . See [BM76, Har69] for an overview on graph operations.

We start with preliminary results whose proofs are very easy.

3.1 Disjoint Union, Co-Join

The *disjoint union* or *co-join* $G = G_1 \cup G_2$ of G_1 and G_2 is the graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. Since NLC-width and clique-width operations both contain the disjoint union it is easy to see that

$$\text{NLC-width}(G_1 \cup G_2) = \max(\text{NLC-width}(G_1), \text{NLC-width}(G_2))$$

and

$$\text{clique-width}(G_1 \cup G_2) = \max(\text{clique-width}(G_1), \text{clique-width}(G_2)).$$

This obviously implies that the NLC-width and clique-width of a graph can be computed by the maximum NLC-width or clique-width of its connected components.

3.2 Join

The *join* $G = G_1 \times G_2$ of G_1 and G_2 is the graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2 \cup \{\{u_1, u_2\} \mid u_1 \in V_1, u_2 \in V_2\}$. It is obviously that

$$\text{NLC-width}(G_1 \times G_2) = \max(\text{NLC-width}(G_1), \text{NLC-width}(G_2))$$

and

$$\text{clique-width}(G_1 \times G_2) = \max(\text{clique-width}(G_1), \text{clique-width}(G_2), 2).$$

By the bounds of Section 4.3 this obviously implies that the NLC-width of a graph also can be computed by the maximum NLC-width of its co-connected components.

3.3 Sum

The *sum* $G = G_1 + G_2$ of G_1 and G_2 with $|V_1| = |V_2|$ is the graph defined by the adjacency matrix given by the sum of adjacency matrices of G_1 and G_2 , that is two vertices are adjacent in $G_1 + G_2$ if and only if they are adjacent in G_1 or they are adjacent in G_2 . Note that the representation of a graph by an adjacency matrix is not unique.

Let G_1 be the disjoint union of m paths P_n , each represented by a row in the adjacency matrix for G_1 , and G_2 be the disjoint union of n paths P_m , each represented by a column in the adjacency matrix for G_2 then graph $G_1 + G_2$ is a $n \times m$ grid and is of unbounded clique-width, even if G_1 and G_2 are of bounded tree-width.

3.4 Difference

The *difference* $G = G_1 - G_2$ of G_1 and G_2 with $|V_1| = |V_2|$ is the graph defined by the adjacency matrix given by the difference of adjacency matrices of G_1 and G_2 , that is two vertices are adjacent in $G_1 - G_2$ if and only if they are adjacent in G_1 and not adjacent in G_2 .

If G_1 has bounded tree-width, then since graphs of bounded tree-width are closed under taking subgraphs, G has bounded tree-width and thus G has bounded clique-width [CR01]. If G_1 is the K_n then G is the edge complement graph of G_2 and has bounded NLC-width and bounded clique-width. If G_1 is the $K_{n,n}$ then we obtain by G bipartite edge complement graph of G_2 . Thus the difference generalizes edge complement and bipartite edge complement.

3.5 Product

A graph product of G_1 and G_2 is a new graph whose vertex set is $V_1 \times V_2$ and for two vertices (u_1, u_2) and (v_1, v_2) the adjacency in the product is defined by the adjacency (or equality, or non-adjacency) of u_1, v_1 in G_1 and u_2, v_2 in G_2 . We consider some of the most famous possibilities to define graph products. Results on graph products, independently of clique-width, can be found in [Har69, IK00, JT94].

The *cartesian graph product* $G = G_1 \square G_2$ of G_1 and G_2 is the graph with vertex set $V_1 \times V_2$ and (u_1, u_2) and (v_1, v_2) are adjacent in G if and only if $(u_1 = v_1 \text{ and } \{u_2, v_2\} \in E_2)$ or $(u_2 = v_2 \text{ and } \{u_1, v_1\} \in E_1)$. If we take the cartesian product of two paths P_n and P_m we obtain an $n \times m$ -grid which does not have bounded clique-width [GR00].

The *categorical graph product* $G = G_1 * G_2$ of G_1 and G_2 is the graph with vertex set $V_1 \times V_2$ and (u_1, u_2) and (v_1, v_2) are adjacent in G if and only if $\{u_1, v_1\} \in E_1$ and $\{u_2, v_2\} \in E_2$. If we take the categorical product of two paths P_n and P_m we obtain the disjoint union of two grids which does not have bounded clique-width [GR00].

The categorical graph product is also called *edge product*.

The *normal graph product* $G = G_1 \cdot G_2$ of G_1 and G_2 is the graph with vertex set $V_1 \times V_2$ and (u_1, u_2) and (v_1, v_2) are adjacent in G if and only if $(u_1 = v_1 \text{ and } \{u_2, v_2\} \in E_2)$ or $(\{u_1, v_1\} \in E_1 \text{ and } u_2 = v_2)$ or $(\{u_1, v_1\} \in E_1 \text{ and } \{u_2, v_2\} \in E_2)$. If we take the normal product of two paths P_n and P_m we obtain a graph of unbounded clique-width.

The normal graph product is also called *strong graph product* or *AND product*.

The *co-normal graph product* $G = G_1 \circ G_2$ of G_1 and G_2 is the graph with vertex set $V_1 \times V_2$ and (u_1, u_2) and (v_1, v_2) are adjacent in G if and only if $\{u_1, v_1\} \in E_1$ or $\{u_2, v_2\} \in E_2$. If we take the co-normal product of two paths P_n and P_m we obtain an $n \times m$ -grid which does not have bounded clique-width [GR00].

The co-normal graph product is also called *disjunctive graph product* or *OR product*.

The *lexicographic graph product* $G = G_1[G_2]$ of G_1 and G_2 is the graph with vertex set $V_1 \times V_2$ and (u_1, u_2) and (v_1, v_2) are adjacent in G if and only if $(\{u_1, v_1\} \in E_1)$ or $(u_1 = v_1 \text{ and } \{u_2, v_2\} \in E_2)$. The lexicographic graph product is also called *graph composition*.

Next we show that the composition of two graphs of bounded NLC-width or bounded clique-width has bounded NLC-width and bounded clique-width.

Theorem 3.1 *For two graphs G_1 and G_2*

$$NLC\text{-width}(G_1[G_2]) = \max(NLC\text{-width}(G_1), NLC\text{-width}(G_2))$$

and

$$clique\text{-width}(G_1[G_2]) = \max(clique\text{-width}(G_1), clique\text{-width}(G_2)).$$

Proof Let G_1 be a graph of NLC-width k_1 and G_2 be a graph of NLC-width k_2 . Let T_1 be an NLC-width k_1 -expression-tree for G_1 and T_2 be an NLC-width k_2 -expression-tree for G_2 . We now construct from T_1 and T_2 an expression-tree T for $G_1[G_2]$. We start with a copy T of T_1 . We relabel every leaf of T from \bullet_a into \circ_R , $R(b) = a$ for $b \in [k_2]$. For each leaf of T we insert a copy of T_2 in T . Then we make the root of each of the copies of T_2 adjacent to one of each relabeled leaf of T . The resulting tree is an expression-tree for $G_1[G_2]$ using $\max(k_1, k_2)$ labels.

In the same way we can show the clique-width result. \square

3.6 Corona

The *corona* $G_1 \wedge G_2$ of G_1 and G_2 consists of the disjoint union of one copy of G_1 and $|V_{G_1}|$ copies of G_2 and each vertex of the copy of G_1 is connected to all vertices of one copy of G_2 (i.e. $|V_{G_1}| \cdot |V_{G_2}|$ edges are inserted in the disjoint union of the $|V_{G_1}| + 1$ graphs).

The corona of graphs was introduced by Frucht and Harary in [FH70]. We show that we can bound the NLC-width and clique-width of $G_1 \wedge G_2$ in the NLC-width or clique-width of its combined graphs.

Theorem 3.2 *Let $m_1 = \max(\text{NLC-width}(G_1), \text{NLC-width}(G_2))$, then it holds*

$$m_1 \leq \text{NLC-width}(G_1 \wedge G_2) \leq m_1 + 1$$

and for $m_2 = \max(\text{clique-width}(G_1), \text{clique-width}(G_2))$

$$m_2 \leq \text{clique-width}(G_1 \wedge G_2) \leq m_2 + 1$$

Proof First we prove the upper bound. Let G_1 be a graph of NLC-width k_1 and G_2 be a graph of NLC-width k_2 . Let T_1 be an NLC-width k_1 -expression-tree for G_1 and T_2 be an NLC-width k_2 -expression-tree for G_2 . We now construct from T_1 and T_2 an expression-tree T for graph $G_1 \wedge G_2$. We start with a copy T of T_1 .

For every $i = 1, \dots, |V_{G_1}|$, we first relabel leaf u_i of T originally labeled by \bullet_{l_i} into $\times_{\{(l_i, k+1)\}}$. Then for $i = 1, \dots, |V_{G_1}|$ we insert one new leaf v_i labeled by \bullet_{l_i} into T . Further we insert one copy $T_{2,i}$ of T_2 and one node w_i labeled by $\circ_R, R(l) = k+1, l = 1, \dots, k_2$ into T . Last we insert two edges between u_i and v_i and u_i and w_i such that v_i is the left child of u_i and w_i is the right child of u_i and one edge between the root of $T_{2,i}$ and w_i into T .

The resulting tree is an expression-tree for graph $G_1 \wedge G_2$ using $\max(k_1, k_2) + 1$ labels.

Since G_1 and G_2 are induced subgraphs of $G_1 \wedge G_2$ the NLC-width of $G_1 \wedge G_2$ is at least the maximum of $\text{NLC-width}(G_1)$ and $\text{NLC-width}(G_2)$. \square

3.7 Substitution

Let G_1 and G_2 be two graphs and let $v \in V_{G_1}$ be a vertex of G_1 . We define a new graph $G_1[v/G_2]$ which has vertex set $V_{G_1} \cup V_{G_2} - \{v\}$ and edge set $E_{G_1} \cup E_{G_2} - \{\{v, w\} \mid w \in N_{G_1}(v)\} \cup \{\{u, w\} \mid u \in V_{G_2}, w \in N_{G_1}(v)\}$. That is by $G_1[v/G_2]$ we denote the graph which we obtain by substituting vertex v in G_1 by graph G_2 . By a proof similar to that of Theorem 3.1 we obviously can show the following bounds.

Theorem 3.3 *For two graphs G_1 and G_2 and $v \in V_{G_1}$*

$$\text{NLC-width}(G_1[v/G_2]) = \max(\text{NLC-width}(G_1), \text{NLC-width}(G_2))$$

and

$$\text{clique-width}(G_1[v/G_2]) = \max(\text{clique-width}(G_1), \text{clique-width}(G_2)).$$

Vertex set V_{G_2} is also called a *module* of graph $G_1[v/G_2]$, since all vertices of V_{G_2} have the same neighbors in graph $G_1[v/G_2]$.

3.8 1-Sum

Let G_1 and G_2 be two graphs and let $v \in V_{G_1}$ and $w \in V_{G_2}$. We define a new graph $G_1 \oplus_{v,w} G_2$ which has vertex set $V_{G_1} \cup V_{G_2} - \{v, w\} \cup \{z\}$ and edge set $E_{G_1} \cup E_{G_2} - \{\{v, v_1\} \in E_{G_1} \mid v_1 \in V_{G_1}\} - \{\{w, w_1\} \in E_{G_2} \mid w_1 \in V_{G_2}\} \cup \{\{z, z_1\} \mid z_1 \in N_{G_1}(v) \cup N_{G_2}(w)\}$. That is by $G_1 \oplus_{v,w} G_2$ we denote the graph which we obtain by the disjoint union of G_1 and G_2 in which vertices v and w are identified.

By a proof similar to that of Theorem 3.1 we can show the following bounds.

Theorem 3.4 For two graphs G_1 and G_2 and $v \in V_{G_1}$ and $w \in V_{G_2}$

$$NLC\text{-width}(G_1 \oplus_{v,w} G_2) \leq \max(NLC\text{-width}(G_1), NLC\text{-width}(G_2)) + 2$$

and

$$clique\text{-width}(G_1 \oplus_{v,w} G_2) \leq (clique\text{-width}(G_1), clique\text{-width}(G_2)) + 2.$$

Vertex z is also called an *articulation vertex* of graph $G_1 \oplus_{v,w} G_2$, since $G_1 \oplus_{v,w} G_2$ without z has more connected components than $G_1 \oplus_{v,w} G_2$. The maximal connected components of some graph G without any articulation vertex are called *blocks* of G . The bounds of Theorem 3.4 obviously imply that the NLC-width and clique-width of a graph can be estimated by the maximum NLC-width or clique-width of its blocks and its number of blocks.

4 Unary Operations

In the following we consider unary graph operations f which create a new graph $f(G)$ from a graph $G = (V, E)$.

4.1 Quotient

If we substitute a module $V' \subseteq V$ of a graph G by a single vertex, e.g. by removing all but one vertices of V' from G , we call the resulting graph $G[V'/v]$ the *quotient* of graph G . Since $G[V'/v]$ is an induced subgraph of G we conclude the following bounds.

Theorem 4.1 For every graph G and every module $V' \subseteq V_G$

$$NLC\text{-width}(G[V'/v]) \leq NLC\text{-width}(G)$$

and

$$clique\text{-width}(G[V'/v]) \leq clique\text{-width}(G).$$

The substitution and quotient operation is used in [Joh98, CMR00] to show that the NLC-width and clique-width of a graph can be obtained by the maximum NLC-width or clique-width of its prime subgraphs appearing as quotient graphs in a modular decomposition.

4.2 Induced subgraph

For every induced subgraph H of a graph G we know that

$$NLC\text{-width}(H) \leq NLC\text{-width}(G)$$

and

$$clique\text{-width}(H) \leq clique\text{-width}(G).$$

An expression for graph H can easily be obtained by an expression from graph G by omitting the vertices of $V_G - V_H$. Although taking induced subgraphs does not increase the NLC-width and clique-width of a graph it is still open if there is a characterization for NLC_k for $k \geq 2$ or CW_k for $k \geq 3$ by sets of forbidden subgraphs.

For arbitrary subgraphs of graphs of bounded clique-width, and thus also not for minors the clique-width and NLC-width can not be bounded. This can easily be shown by the example of a complete graph (NLC-width 1, clique-width 2) and the subgraphs and minors have arbitrary large NLC-width and clique-width.

4.3 Edge Complement

The *edge complement graph* \overline{G} of a graph G has the same vertex set as G and two vertices in \overline{G} are adjacent if and only if they are not adjacent in G , i.e. $\overline{G} = (V, \{\{u, v\} \mid u, v \in V, u \neq v, \{u, v\} \notin E\})$. The following bounds are known from [Wan94, CO00].

$$\text{NLC-width}(\overline{G}) = \text{NLC-width}(G)$$

$$\frac{1}{2} \cdot \text{clique-width}(G) \leq \text{clique-width}(\overline{G}) \leq 2 \cdot \text{clique-width}(G)$$

4.4 Bipartite complement

If $G = (V, E)$ is a bipartite graph with vertex partition $V = V_1 \cup V_2$, such that there are no edges between two vertices of V_1 and no edges between two vertices of V_2 , then the *bipartite complement* $\overline{G}^{\text{bip}}$ of G is the graph with vertex set V and edge set $\{\{u, v\} \mid \{u, v\} \notin E, u \in V_1, v \in V_2\}$. In [LR04] it is shown that for bipartite graphs G it holds

$$\frac{1}{4} \cdot \text{clique-width}(G) \leq \text{clique-width}(\overline{G}^{\text{bip}}) \leq 4 \cdot \text{clique-width}(G).$$

In the same way one can show that

$$\frac{1}{2} \cdot \text{NLC-width}(G) \leq \text{NLC-width}(\overline{G}^{\text{bip}}) \leq 2 \cdot \text{NLC-width}(G).$$

4.5 Power of a graph

The *d-th power* G^d of a graph G is a graph with the same set of vertices as G and an edge between two vertices if and only if there is a path of length at most d between them. Todinca [Tod03] has shown the following bound.

$$\text{clique-width}(G^d) \leq 2 \cdot \text{clique-width}(G) \cdot d^{\text{clique-width}(G)}$$

Now we show bounds for the NLC-width and clique-width of graphs when applying local graph operations which are our main results.

4.6 Switching

Next we consider the *switching* operation defined by Seidel [Sei92]. Let G be a graph and x be a vertex of G . $S(G, x)$ defines the graph whose vertex set is the same as of G and whose edge set is the edge set of G but the vertices adjacent to x become exactly to those vertices which are not adjacent to x . That is, $V_{S(G, x)} = V_G$ and

$$E_{S(G, x)} = E_G - \{\{x, y\} \mid y \in V_G, \{x, y\} \in E_G\} \cup \{\{x, y\} \mid y \in V_G, x \neq y, \{x, y\} \notin E_G\}.$$

Two graphs are called *switching equivalent* if one of them can be transformed into a graph isomorphic to the other by a sequence of switching operations. It is shown in [CC80] that deciding if two graphs are switching equivalent is an isomorphism complete problem.

Next we will show that one switching operation in a graph increases or decreases its NLC-width at most by one.

Theorem 4.2 *Let G be a graph and $x \in V_G$, then*

$$NLC\text{-}width(G) - 1 \leq NLC\text{-}width(S(G, x)) \leq NLC\text{-}width(G) + 1$$

and

$$\frac{1}{2} \cdot \text{clique-width}(G) \leq \text{clique-width}(S(G, x)) \leq 2 \cdot \text{clique-width}(G).$$

Proof Let T be an NLC-width k -expression-tree that defines G . We now define a new NLC-width $k + 1$ -expression-tree that defines $S(G, x)$. We start with a copy T' of T . Let u be the leaf of T' that corresponds to vertex x of G . We relabel leaf u in T' by \bullet_{k+1} .

Now we consider the union nodes u_1 on the path from u to the root of T' in T' . If u is a left (right) child of u_1 and union node u_1 is labeled by \times_S then we relabel u_1 by $\times_{S'}$, where $S' = S \cup \{(k + 1, l) \mid (\text{lab}(u, G(u_1)), l) \notin S, l \in [k]\}$ ($S' = S \cup \{(l, k + 1) \mid (l, \text{lab}(u, G(u_1))) \notin S, l \in [k]\}$). This is necessary in order to make all vertices adjacent to x which are not adjacent to x in G , and vice versa.

The resulting tree is denoted by T'' . It is clear that T'' is an NLC-width $k + 1$ -expression-tree and it is easy to show that T'' defines graph $S(G, x)$.

The lower bound follows since by $S(S(G, x), x)$ we obtain G .

In order to show a bound on the clique-width of graph $S(G, x)$ we can not bound the clique-width by a constant independently on k , because of the different edge insertion operation in the clique-width model. Clearly we can first remove vertex x from a graph G and insert a new vertex with neighborhood $V_G - N(x)$ to obtain graph $S(G, x)$ of clique-width at most $2k$, see Section 4.11. In this case it is much more comfortable to use NLC-width operations. \square

Obviously the bound on the NLC-width given in Theorem 4.2 is best possible. Let G be the graph of NLC-width 1 in Figure 2 (which is called *paw* or *3-pan* [BLS99]). A switching operation on graph G at one of the vertices of degree 2 creates a graph H which is isomorphic to P_4 of NLC-width 2. Further by $S(H, x)$ we obtain graph G , thus the lower bound is best possible too.



Figure 2: The graph G on the left side is called *paw* or *3-pan* and has NLC-width 1 (clique-width 2), the graph H on the right side can be obtained by $S(G, x)$ and is called P_4 has NLC-width 2 (clique-width 3).

4.7 Local Complementation

For some graph G and a vertex x of G the *local complementation* $LC(G, x)$ is defined by Bouchet [Bou94]. Graph $LC(G, x)$ is obtained from graph G by replacing the subgraph of G defined by $N(x)$ by its edge complement, i.e. $V_{LC(G, x)} = V_G$ and

$$E_{LC(G, x)} = E_G - \{\{y, z\} \mid y, z \in N_G(x), \{y, z\} \in E_G\} \cup \{\{y, z\} \mid y, z \in N_G(x), \{y, z\} \notin E_G\}.$$

Two graphs are called *locally equivalent* if one of them can be transformed into a graph isomorphic to the other by a sequence of local complementations.

The example of a paw (3-pan) G , see Figure 2, shows that the local complementation can increase or decrease the NLC-width and clique-width of a graph. If we apply a local complementation on G at one of the vertices of degree 2, we obtain graph $H \cong P_4$ with NLC-width 2 (clique-width 3).

Theorem 4.3 *Let G be a graph and $x \in V_G$, then*

$$\frac{1}{2} \cdot \text{NLC-width}(G) \leq \text{NLC-width}(LC(G, x)) \leq 2 \cdot \text{NLC-width}(G)$$

and

$$\frac{1}{3} \cdot \text{clique-width}(G) \leq \text{clique-width}(LC(G, x)) \leq 3 \cdot \text{clique-width}(G).$$

Proof Let T be an NLC-width k -expression-tree that defines graph G . We now define a new NLC-width $2k$ -expression-tree that defines graph $LC(G, x)$. We start with a copy T' of T . The main idea is to separate the vertices in $N(x)$ from the vertices in $V - N(x)$. Let $l = |N(x)|$ and u_1, \dots, u_l be the leaves of T' that corresponds to vertices in $N(x)$ of G .

Now we consider the nodes u on the paths from u_i , $i = 1, \dots, l$ to the root of T' in T' .

1. If u is a leaf u_i , $i = 1, \dots, l$, labeled by \bullet_a in T' , then we relabel u by \bullet_{a+k} .
2. If u is a relabeling node labeled by \circ_R , then we relabel u by $\circ_{R'}$, such that $R'(a) = R(a)$, if $a \leq k$ and $R'(a) = R(a - k)$, if $a > k$.
3. If u is a union node labeled by \times_S , then we relabel u by $\times_{S'}$, such that $S' = S \cup S_1 \cup S_2$, where $S_1 = \{(a + k, b + k) \mid (a, b) \notin S\}$ and $S_2 = \{(a, b + k), (a + k, b) \mid (a, b) \in S\}$. S_1 creates an edge between two nodes in $N(x)$, if and only if these vertices are not adjacent in G , S_2 creates an edge between one vertex of $V_G - N(x)$ and one vertex of $N(x)$, if and only if these vertices are adjacent in G .

This is necessary in order to create the complement graph of the graph induced by $N(x)$. The resulting tree is denoted by T'' . It is clear that T'' is an NLC-width $2k$ -expression-tree and it is easy to show that T'' defines graph $LC(G, x)$.

The lower bound follows since by $L(L(G, x), x)$ we obtain G .

A similar result we can show for clique-width, but we need k additionally labels to distinguish the vertices in $N(x)$ from those in $V - N(x)$ and k further labels to create the complement graph of $G(N(x))$. \square

The proof of Theorem 4.3 implies the following bounds for the NLC-width and clique-width of graph $LC(G, x)$ using the vertex degree of x in graph G .

Corollary 4.4 *Let G be a graph of NLC-width k , and $x \in V_G$, then*

$$\text{NLC-width}(LC(G, x)) \leq k + \min(k, \deg_G(x))$$

and for every graph G of clique-width k

$$\text{clique-width}(LC(G, x)) \leq k + \min(2k, 2 \cdot \deg_G(x)).$$

In [Oum05] it is shown that for each fixed k the set of graphs of rank-width at most k is closed under local complementation which leads to a characterization of graphs of rank-width at most k by a finite set of forbidden vertex-minors.

4.8 Edge Addition and Edge Deletion

Next we analyze the NLC-width and clique-width of a graph if we add a further edge or if we remove an existing edge. Our next theorem shows that we can insert a further edge into a graph using at most 2 more labels.

Theorem 4.5 *Let H be a graph obtained from graph G by inserting or deleting one edge, then*

$$NLC\text{-width}(G) - 2 \leq NLC\text{-width}(H) \leq NLC\text{-width}(G) + 2$$

and

$$\text{clique-width}(G) - 2 \leq \text{clique-width}(H) \leq \text{clique-width}(G) + 2.$$

Proof First we want to show the upper bound. Let G be a graph of NLC-width k and let x, y be two non-adjacent vertices of G . Further, let T be an NLC-width k -expression-tree that defines G . We now define a new NLC-width $k + 2$ -expression-tree that defines H . We start with a copy T' of T . Let u and v be the leaves of T' that correspond to vertices x and y , respectively, of graph G . First, we relabel leaf u and v in T' by \bullet_{k+1} and \bullet_{k+2} , respectively.

Now we consider all union nodes u_1 on the path from u to the root of T' in T' . If u is a left (right) child of u_1 and union node u_1 is labeled by \times_S and $(\text{lab}(u, G(u_1)), l) \in S$ ($(l, \text{lab}(u, G(u_1))) \in S$) for some $l \in [k]$ then we relabel u_1 by $\times_{S'}$, where $S' = S \cup \{(k+1, l) \mid (l, \text{lab}(u, G(u_1))) \in S, l \in [k]\}$ ($S' = S \cup \{(l, k+1) \mid (l, \text{lab}(u, G(u_1))) \in S, l \in [k]\}$). This is necessary in order to make all vertices adjacent to x which are adjacent to x in G . In the same way we have to modify all union nodes on the path from v to the root of T' in order to make all vertices of G adjacent to y , if they are in G .

Last we have to relabel the least common predecessor w of u and v in T' to create the edge between x and y . Since w is always a union node in T' , w is labeled by \times_S for some $S \subseteq [k]^2$. If u is the left (right) child and v is the right (left) child of w in T' then we relabel w by $\times_{S \cup \{(k+1, k+2)\}} (\times_{S \cup \{(k+2, k+1)\}})$.

The resulting tree is denoted by T'' . It is clear that T'' is an NLC-width $k + 2$ -expression-tree and it is easy to show that T'' defines graph H .

The proof for edge deletion runs similar, we just have to leave out the above described relabeling of the least common predecessor w of u and v in T' to create the edge between x and y .

For the lower bounds assume that H is obtained from G by inserting (deleting) an edge e and $NLC\text{-width}(H) < NLC\text{-width}(G) - 2$. Then we obtain by deleting (inserting) e from (in) H a graph G' isomorphic to G with $NLC\text{-width}(G') < NLC\text{-width}(G)$, by our upper bound, and thus a contradiction.

The results for clique-width can be shown by similar augmentations. \square

If we add or delete an edge in a graph of NLC-width 1, i.e. a co-graph, then we always obtain a graph of NLC-width at most 2, since we can label both end vertices of the new edge (both end vertices of the deleted edge) by the same label 2.

The example in Figure 3 shows that the NLC-width bounds of Theorem 4.5 can not be improved for $k = 2$. Graph G has NLC-width 2 and graph H , which we obtain after inserting edge $e = \{x, y\}$ in G , has NLC-width 4. Further the example of Figure 3 also shows that the bound for edge deletion is best possible for $k = 2$. The edge complement graph \overline{G} of G has NLC-width 2 and contains edge $\{x, y\}$. If we remove edge $\{x, y\}$ from \overline{G} we obtain graph \overline{H} which has NLC-width 4.

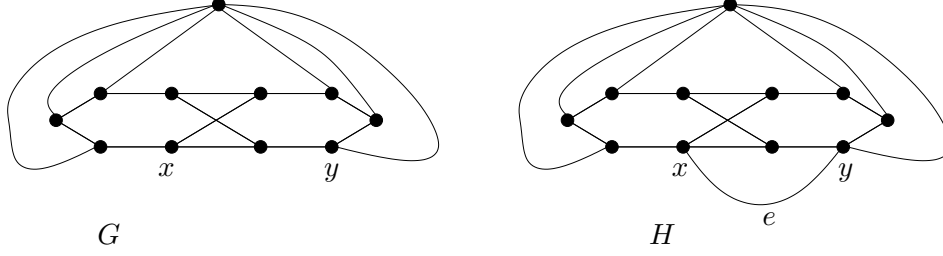


Figure 3: Graph G has NLC-width 2. Graph H can be obtained from G by adding edge e and H has NLC-width 4.

Thus, in the in last theorem we give an answer of Question 6.3 of [CO00]. It remains open to find a graph G and a graph H , which is obtained from G by adding or deleting one edge, such that $|\text{clique-width}(G) - \text{clique-width}(H)| = 2$ holds.

4.9 Edge Subdivision

Here we analyze the subdivision of an edge, i.e. for some graph $G = (V, E)$ and an edge $\{x, y\} \in E$ we define graph $G_{x,y} = (V_{G_{x,y}}, E_{G_{x,y}})$ by $V_{G_{x,y}} = V \cup \{z\}$ and $E_{G_{x,y}} = E - \{\{u, v\} \cup \{\{x, z\}, \{y, z\}\}\}$, as *subdivision* of G . The subdivision operation is also known as *elementary refinement*. At least after subdividing all edges of a graph the resulting graph is bipartite. After subdividing every edge of a graph G the resulting graph is called the *incidence graph* of the given graph.

Incidence graphs have unbounded clique-width in general, but incidence graphs of graphs of bounded tree-width have bounded clique-width, since subdivisions do not change the tree-width.

Theorem 4.6 *Let G be a graph and $\{x, y\} \in E_G$ an edge, then*

$$\text{NLC-width}(G) - 2 \leq \text{NLC-width}(G_{x,y}) \leq \text{NLC-width}(G) + 2$$

and

$$\text{clique-width}(G) - 2 \leq \text{clique-width}(G_{x,y}) \leq \text{clique-width}(G) + 2.$$

Proof First we want to show the upper bound. Let G be a graph of NLC-width k and let $\{x, y\}$ be an edge of G . Let T be an NLC-width k -expression-tree that defines G . We now define a new NLC-width $k + 2$ -expression-tree that defines $G_{x,y}$.

Let T' be defined for T as in the proof of Theorem 4.5 for edge removing. In T' we insert a new root r labeled by $\times_{\{(k+1,k+1),(k+2,k+1)\}}$ and a new node w (defining the vertex which subdivides edge e) labeled by \bullet_{k+1} and two edges, one from r to r and one from r to the root of T' such that w is the right child of r .

The resulting tree is denoted by T'' . It is clear that T'' is an NLC-width $k + 2$ -expression-tree and it is easy to show that T'' defines graph $G_{x,y}$.

For the lower bounds assume that $G_{x,y}$ is obtained from G by subdividing an edge $e = \{x, y\}$ and $\text{NLC-width}(G_{x,y}) < \text{NLC-width}(G) - 2$. Then we obtain by removing the inserted vertex and inserting e in $G_{x,y}$ a graph G' isomorphic to G with $\text{NLC-width}(G') < \text{NLC-width}(G)$, by our upper bound in Theorem 4.5, and thus a contradiction.

Since the clique-width operations do not allow edge insertions between equal labeled vertices, we have to do one additional relabeling $\rho_{k+1 \rightarrow k+2}$ (to label x and y in the proof of Theorem 4.5 by $k+2$) before inserting the new vertex in T . \square

The upper bound for $\text{NLC-width}(G_{x,y})$ and $\text{clique-width}(G_{x,y})$ of Theorem 4.6 can not be improved, since first subdividing an edge e and deleting the new vertex corresponds to edge deletion, which needs to additional labels in general.

In the appendix of [CO00] it is shown that in a graph G of clique-width of at least 4 every path of length at least 5, consisting of vertices which all have degree 2 in G and one end vertex of degree 1 in G , can be extended by subdivisions without increasing the clique-width of G .

Clearly subdivisions can increase (P_3 to P_4) the NLC-width or leave it unchanged (P_4 to P_5).

Problem 4.7 *Does there exists some graph G , such that $\text{NLC-width}(G_{x,y}) < \text{NLC-width}(G)$ or $\text{clique-width}(G_{x,y}) < \text{clique-width}(G)$?*

4.10 Vertex Identification and Edge Contraction

In this section we analyze the *identification* or *merger* of two vertices in a graph with respect to the NLC-width and clique-width. For some graph $G = (V, E)$ and two different vertices $x, y \in V$ we define $G^{x,y} = (V_{G^{x,y}}, E_{G^{x,y}})$ with $V_{G^{x,y}} = V - \{x, y\} \cup \{z\}$ and

$$E_{G^{x,y}} = E - \{\{u, v\} \mid u \in V, v \in \{x, y\}\} \cup \{\{u, z\} \mid u \notin \{x, y\} \text{ and } \{u, x\} \in E \text{ or } \{u, y\} \in E\}.$$

Theorem 4.8 *If graph $G^{x,y}$ is obtained from graph G by identifying two vertices, then*

$$\frac{1}{4} \cdot \text{NLC-width}(G) \leq \text{NLC-width}(G^{x,y}) \leq 2 \cdot \text{NLC-width}(G)$$

and

$$\frac{1}{4} \cdot \text{clique-width}(G) \leq \text{clique-width}(G^{x,y}) \leq 2 \cdot \text{clique-width}(G).$$

Proof For the upper bound we can delete x, y and insert z , see Theorem 4.10, with neighborhood $N(x) \cup N(y)$. The lower bound holds since we can obtain G from $G^{x,y}$ by removing z and inserting the two vertices x and y , each with a factor of 2. \square

If the two vertices of an identification are adjacent we call the corresponding operation *edge contraction*, which is a well known minor operation.

Problem 4.9 *Does there exists some graph G and $\{x, y\} \in E_G$, such that $\text{NLC-width}(G) < \text{NLC-width}(G^{x,y})$ or $\text{clique-width}(G) < \text{clique-width}(G^{x,y})$?*

In the appendix of [CO00] it is shown that in a graph G of clique-width of at least 4 every path of length at least 2, consisting of vertices which all have degree 2 in G and one end vertex of degree 1 in G , can be decreased by edge contractions without increasing the clique-width of G .

4.11 Vertex Addition and Vertex Deletion

Last we consider the NLC-width and clique-width of a graph if we add a further vertex.

Theorem 4.10 *If graph H is obtained from graph G by inserting a new vertex, then*

$$\text{NLC-width}(G) \leq \text{NLC-width}(H) \leq 2 \cdot \text{NLC-width}(G)$$

and

$$\text{clique-width}(G) \leq \text{clique-width}(H) \leq 2 \cdot \text{clique-width}(G).$$

Proof Let T be an NLC-width k -expression-tree that defines graph G . Let H be the graph which we obtain from G by inserting a new vertex v with an arbitrary neighborhood $N(v) \subseteq V_G$. That is $H = (V_G \cup \{v\}, E_G \cup \{\{v, v'\} \mid v' \in N(v)\})$. We now define a new NLC-width $2k$ -expression-tree that defines H . We start with a copy T' of T .

First we want to separate the neighborhood of v from the non-neighborhood by introducing k further labels $k+1, \dots, 2k$. Every leaf of T' that corresponds to a vertex from G which is not from $N(v)$ will be relabeled from label \bullet_a , $a \in [k]$, into \bullet_{a+k} .

Then we consider all nodes u on the paths from these relabeled leaves to the root of the so defined tree. If node u is a union node labeled by some \times_S , $S \subseteq [k]^2$, then we relabel u by $\times_{S'}$ where $S' = \{(a, b), (a, b+k), (a+k, b), (a+k, b+k) \mid (a, b) \in S\}$. If node u is a relabeling node labeled by some \circ_R , $R : [k] \rightarrow [k]$, then we relabel u by $\circ_{R'}$, where $R' : [2k] \rightarrow [2k]$ and $R'(a) = R(a)$, if $i \leq k$ and $R'(a) = R(a) + k$, if $k < a \leq 2k$. The resulting tree is denoted by T'' .

In a last step we insert two additional nodes t_v and t_r labeled by \bullet_1 and $\times_{\{(1,a) \mid a \in [k]\}}$, respectively and two additional arcs from t_r to t_v and from t_r to the root of T' in T' , such that t_v is the left child of t_r .

The resulting tree is denoted by T''' . It is clear that T''' is an NLC-width $2k$ -expression-tree and that T''' defines graph H .

Since G is an induced subgraph of H , the NLC-width of H cannot be less than NLC-width of G .

To prove the corresponding clique-width bound, we have to find for vertex v a label which is not used in the graph defined by $G(T'')$, since clique-width does not allow edge insertions between equal labeled vertices. This can be done by relabeling all vertices $G(T'')$ labeled by $k+1, \dots, 2k$ by e.g. $k+1$ and then we can take, for $k \geq 2$, one of the free labels e.g. label $2k$ to label the inserted vertex v . In the case $k = 1$, G consists of isolated vertices and H is the disjoint union of one $K_{1,p}$, for some p , and isolated vertices. Thus also in this case H has clique-width $2k = 2$. \square

If we insert a vertex in a path of length 2 to get a path of length 3, we insert a vertex in a graph NLC-width 1 and obtain a graph of NLC-width 2. Graph $H - x$ of Figure 3 has NLC-width 2 and graph H has NLC-width 4. Thus, for $k = 1, 2$ the NLC-width bound of Theorem 4.10 is best possible.

Further it is possible to bound the NLC-width and clique-width of $G+v$ in the NLC-width and clique-width of G and the vertex degree of v . The main idea is to label each vertex of G which should be adjacent to vertex v by a new label from $\{k+1, \dots, k+d\}$. Then, the new vertex can easily be inserted in a last step. (If we use clique-width operations we first have to relabel at least one of the used labels from $\{1, \dots, k\}$ to get a free label in order to insert the new vertex.)

Corollary 4.11 *If graph H is obtained from graph G by inserting a new vertex of degree d , then*

$$NLC\text{-width}(G) \leq NLC\text{-width}(H) \leq NLC\text{-width}(G) + d$$

and

$$\text{clique-width}(G) \leq \text{clique-width}(H) \leq \text{clique-width}(G) + d.$$

In Principle 2 of [BLM04] the clique-width of graph H , obtained by a graph G by inserting a set of vertices V' , is bounded by the clique-width of G and the number of neighborhoods of V' to the vertices of G . This is clearly possible and very useful in some cases.

Since graphs of bounded NLC-width and graphs of bounded clique-width are closed under taking induced subgraphs, we can delete vertices (including all adjacent edges) without increasing the NLC-width or clique-width of a given graph.

5 Conclusions and Overview

We gave an overview how the NLC-width and clique-width of a given graph changes if we apply certain graph operations f on this graph. In all cases in which it is possible to bound the NLC-width of the resulting graph $f(G)$ we also show how to compute the corresponding expression in linear time in the size of the corresponding expression for G . Thus our results are constructive. Although clique-width is the more famous concept, we obtain in all cases closer bounds for $NLC\text{-width}(f(G))$ for local operations f . Therefore in our results the results for NLC-width are mentioned at first.

Since the computation of NLC-width and clique-width is NP-hard [GW05, FRRS06], it seems to be difficult to find an optimal k -expression for some given graph. Our results may help to find an expression for some given graph $f(G)$ from a known expression of a similar graph G . For example, we can construct an NLC-width $k+l$ -expression for every graph which is switching equivalent to some graph with known NLC-width k -expression, where l is the number of necessary switching operations. As well, we can construct an $k+2$ -expression for every graph which differs only by one edge from a graph with known k -expression.

In nearly all cases, it remains to show that our bounds are best possible, or to improve them. Especially the clique-width bounds on the complement, bipartite complement and local complementation seem to be improvable.

Further it remains open if there are graph operations, which do not increase the clique-width or NLC-width of a given graph and make the given graph smaller, in order to define reduction rules or a characterization by forbidden graphs for graphs of bounded clique-width or graphs of bounded NLC-width. Among the considered operations only the operation induced subgraph does not increase the clique-width or NLC-width, but it is open if there is a characterization for graphs of clique-width at most k or graphs of NLC-width at most k by forbidden induced subgraphs.

It is also an open problem to find graph operations that increase or decrease the NLC-width or clique-width of some graph by a fixed constant or a fixed factor, e.g. an operations such that for every graph G there is a positive integer c such that $NLC\text{-width}(f(G)) = c + NLC\text{-width}(G)$ or $NLC\text{-width}(f(G)) = c \cdot NLC\text{-width}(G)$. This would imply a useful means in order to decrease the NLC-width or clique-width in a controlled way.

From our results we can follow that *tree-cographs* [Tin89] defined recursively as follows have bounded NLC-width.

1. Every tree is a tree-cograph.
2. If G is a tree-cograph, then the complement graph \overline{G} is a tree-cograph.
3. If G_1, \dots, G_k , $k \geq 2$ are connected tree-cographs then the disjoint union $G_1 \cup \dots \cup G_k$ is also a tree-cograph.

Since trees have NLC-width at most 3 [Wan94] and edge-complement and disjoint union do not increase the NLC-width we know that tree-cographs have NLC-width at most 3 and thus clique-width at most 6.

For two tree-cographs T_1, T_2 we can compute $\overline{T_1 \cup T_2}$ by $\overline{T_1} \times \overline{T_2}$. Thus we can equivalently define tree-cographs by the following four rules.

1. Every tree is a tree-cograph.
2. Every complement of a tree (no tree-cograph!) is a tree-cograph.
3. If G_1, \dots, G_k , $k \geq 2$ are connected tree-cographs then the disjoint union $G_1 \cup \dots \cup G_k$ is also a tree-cograph.
4. If G_1, \dots, G_k , $k \geq 2$ are connected tree-cographs then the join $G_1 \times \dots \times G_k$ is also a tree-cograph.

Since trees have clique-width at most 3 and complement of trees have clique-width at most 4 we follow from the last characterization that tree-cographs have clique-width at most 4.

Corollary 5.1 *Tree-cographs have NLC-width at most 3 and clique-width at most 4.*

In a similar way we can modify the definition of a tree to get *cograph-trees*.

1. Every tree is a cograph-tree.
2. If we substitute a vertex x of a cograph-tree G by a co-graph H , as defined in Section 3, we obtain by $G[x/H]$ a cograph-tree.

Since trees have NLC-width and clique-width at most 3 and cographs have NLC-width 1 and clique-width at most 2 and substitution does not increase the NLC-width or clique-width, we obtain the following result.

Corollary 5.2 *Cographs-trees have NLC-width at most 3 and clique-width at most 3.*

In Table 1 we compare our results concerning binary graph operations. In Table 2 we compare our results concerning unary graph operations.

References

- [BDLM02] A. Brandstädt, F. F. Dragan, H.-O. Le, and R. Mosca. New graph classes of bounded clique width. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 2573 of *LNCS*, pages 57–67. Springer, 2002.

operation \bullet	NLC-width($G_1 \bullet G_2$)	clique-width($G_1 \bullet G_2$)
disjoint union	$\max(k_1, k_2)$	$\max(k_1, k_2)$
join	$\max(k_1, k_2)$	$\max(k_1, k_2, 2)$
composition	$\max(k_1, k_2)$	$\max(k_1, k_2)$
corona	$\max(k_1, k_2) + 1$	$\max(k_1, k_2) + 1$
substitution	$\max(k_1, k_2)$	$\max(k_1, k_2)$
1-sum	$\max(k_1, k_2) + 2$	$\max(k_1, k_2) + 2$

Table 1: The table shows for binary graph operations \bullet and some given pair of graphs G_1, G_2 of NLC-width or clique-width k_1 and k_2 , respectively, the upper bounds of the NLC-width or clique-width of graph $G_1 \bullet G_2$.

operation f	NLC-width($f(G)$)	clique-width($f(G)$)
quotient	k	k
induced subgraph	k	k
edge complement	k	$2k$
bipartite complement	$2k$	$4k$
local complementation	$2k$	$3k$
switching	$k + 1$	$2k$
edge insertion	$k + 2$	$k + 2$
edge subdivision	$k + 2$	$k + 2$
vertex insertion	$2k$	$2k + 1$
edge deletion	$k + 2$	$k + 2$
edge contraction	$2k$	$2k$

Table 2: The table shows for some given graph G of NLC-width or clique-width k the upper bounds of the NLC-width or clique-width of graph G after applying unary operation f .

- [BLM04] A. Brandstädt, H.O. Le, and R. Mosca. Gem- and co-gem-free graphs have bounded clique-width. *International Journal of Foundations of Computer Science*, 15:163–185, 2004.
- [BLS99] A. Brandstädt, V.B. Le, and J.P. Spinrad. *Graph Classes: A Survey*. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, 1999.
- [BM76] J. Bondy and U. Murty. *Graph Theory with Applications*. North-Holland, 1976.
- [Bod98] H.L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209:1–45, 1998.
- [Bou94] A. Bouchet. Circle graph obstructions. *Journal of Combinatorial Theory, Series B*, 60:107–144, 1994.
- [CC80] C.J. Colbourn and D.G. Corneil. On deciding switching equivalence of graphs. *Discrete Applied Mathematics*, 2:181–184, 1980.
- [CHL⁺00] D.G. Corneil, M. Habib, J.M. Lanlignel, B. Reed, and U. Rotics. Polynomial time recognition of clique-width at most three graphs. In *Proceedings of Latin American Symposium on Theoretical Informatics*, volume 1776 of *LNCS*, pages 126–134. Springer, 2000.
- [CMR00] B. Courcelle, J.A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000.
- [CO00] B. Courcelle and S. Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101:77–114, 2000.
- [CPS85] D.G. Corneil, Y. Perl, and L.K. Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14(4):926–934, 1985.
- [CR01] D.G. Corneil and U. Rotics. On the relationship between clique-width and treewidth. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 2204 of *LNCS*, pages 78–90. Springer, 2001.
- [EGW01] W. Espelage, F. Gurski, and E. Wanke. How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 2204 of *LNCS*, pages 117–128. Springer, 2001.
- [EGW03] W. Espelage, F. Gurski, and E. Wanke. Deciding clique-width for graphs of bounded tree-width. *Journal of Graph Algorithms and Applications - Special Issue of JGAA on WADS 2001*, 7(2):141–180, 2003.
- [FH70] R. Frucht and F. Haray. On the coronas of two graphs. *Aequationes Math.*, 4:322–324, 1970.
- [FRRS05] M.R. Fellows, F.A. Rosamund, U. Rotics, and S. Szeider. Proving NP-hardness for clique-width II: non-approximability of clique-width. Technical Report TR05-081, Electronic Colloquium on Computational Complexity (ECCC), 2005.

- [FRRS06] M.R. Fellows, F.A. Rosamund, U. Rotics, and S. Szeider. Clique-width minimization is NP-hard. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, pages 354–362. ACM, 2006.
- [GR00] M.C. Golumbic and U. Rotics. On the clique-width of some perfect graph classes. *International Journal of Foundations of Computer Science*, 11(3):423–443, 2000.
- [GW00] F. Gurski and E. Wanke. The tree-width of clique-width bounded graphs without $K_{n,n}$. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 1938 of *LNCS*, pages 196–205. Springer, 2000.
- [GW05] F. Gurski and E. Wanke. Minimizing NLC-width is NP-complete (Extended Abstract). In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 3787 of *LNCS*, pages 69–80. Springer, 2005.
- [GW06] F. Gurski and E. Wanke. Vertex disjoint paths on clique-width bounded graphs. *Theoretical Computer Science*, 359(1-3):188–199, 2006.
- [GW07] F. Gurski and E. Wanke. Line graphs of bounded clique-width. *Discrete Mathematics*, to appear, 2007.
- [Har69] F. Harary. *Graph Theory*. Addison-Wesley Publishing Company, Massachusetts, 1969.
- [IK00] W. Imrich and S. Klavzar. *Product Graphs: Structure and Recognition*. Series in Discrete Mathematics and Optimization. Wiley-Interscience, Chichester, 2000.
- [Joh98] Ö. Johansson. Clique-decomposition, NLC-decomposition, and modular decomposition - relationships and results for random graphs. *Congressus Numerantium*, 132:39–60, 1998.
- [Joh00] Ö. Johansson. NLC₂-decomposition in polynomial time. *International Journal of Foundations of Computer Science*, 11(3):373–395, 2000.
- [JT94] T.R. Jensen and B. Toft. *Graph Coloring Problems*. John Wiley & Sons, New York, 1994.
- [KR03] D. Kobler and U. Rotics. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Applied Mathematics*, 126(2-3):197–221, 2003.
- [LR04] V. Lozin and D. Rautenbach. Chordal bipartite graphs of bounded tree- and clique-width. *Discrete Mathematics*, 283:151–158, 2004.
- [OS06] S.-I. Oum and P.D. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006.
- [Oum05] S.-I. Oum. Rank-width and well-quasi-ordering. Manuscript, 2005.
- [RS85] N. Robertson and P.D. Seymour. *Graph Minors – A Survey*, pages 153–171. Cambridge Univ. Press, 1985.
- [Sei92] J.J. Seidel. More about two-graphs, combinatorics, graphs and complexity. *Annals of Discrete Mathematics*, 51:297–308, 1992.

- [Tin89] G. Tinhofer. Strong tree-cographs are birkhoff graphs. *Discrete Applied Mathematics*, 22:275–288, 1988/89.
- [Tod03] I. Todinca. Coloring powers of graphs of bounded clique-width. In *Proceedings of Graph-Theoretical Concepts in Computer Science*, volume 2880 of *LNCS*, pages 370–382. Springer, 2003.
- [Wan94] E. Wanke. k -NLC graphs and polynomial algorithms. *Discrete Applied Mathematics*, 54:251–266, 1994.