Line graphs of bounded clique-width

Frank Gurski, Egon Wanke

Heinrich-Heine-Universität Düsseldorf, Institute of Computer Science, D-40225 Düsseldorf, Germany

Received 16 August 2006; received in revised form 29 September 2006; accepted 22 January 2007
Available online 16 February 2007

Abstract

We show that a set of graphs has bounded tree-width or bounded path-width if and only if the corresponding set of line graphs has bounded clique-width or bounded linear clique-width, respectively. This relationship implies some interesting algorithmic properties and re-proves already known results in a very simple way. It also shows that the minimization problem for NLC-width is NP-complete. © 2007 Elsevier B.V. All rights reserved.

Keywords: Clique-width; NLC-width; Line graphs; Incidence graphs; Tree-width

1. Introduction

The clique-width of a graph is defined by a composition mechanism for vertex-labeled graphs [11]. The operations are the creation of a new labeled vertex, 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 [39]. 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 $2k$ [25]. 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. We also consider restricted forms of clique-width and NLC-width operations. A graph $G$ has linear clique-width (linear NLC-width) at most $k$ if it can be defined by a clique-width $k$-expression (an NLC-width $k$-expression, respectively) where at least one argument of every disjoint union operation (of every union operation, respectively) is a single labeled vertex [23].

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

---

* Parts of this paper have been published in an extended abstract [22].
E-mail addresses: gurski-dm@acs.uni-duesseldorf.de (F. Gurski), wanke-dm@acs.uni-duesseldorf.de (E. Wanke).
(MSO$_1$-logic) are decidable in linear time on clique-width bounded graphs [10] if a corresponding decomposition for the graph is given as input. This MSO$_1$-logic has been extended by counting mechanisms which allow the expressibility of optimization problems concerning maximal or minimal vertex sets [10]. All graph problems expressible in extended MSO$_1$-logic can be solved in polynomial time on clique-width bounded graphs. Furthermore, there are a lot of NP-complete graph problems which are not expressible in extended 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 [39,13,27,38,21].

If a graph $G$ has clique-width (NL width) at most $k$ then the edge complement $\overline{G}$ has clique-width at most $2k$ (NL width at most $k$) [11,39]. Distance hereditary graphs have clique-width at most 3 [18]. 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. [6] have analyzed the clique-width of graphs defined by forbidden one-vertex extensions of $P_4$. The clique-width and NLC-width of permutation graphs, unit interval graphs, grids and thus planar graphs are not bounded [18]. An arbitrary graph with $n$ vertices has clique-width at most $n-r$, if $2^r < n-r$, and NLC-width at most $\lceil n/2 \rceil$ [25]. Every graph of tree-width at most $k$ has clique-width at most $3 \cdot 2^{k-1}$ [9]. In [20], it is shown that every graph of clique-width or NLC-width at most $k$ 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. Deciding whether a graph has clique-width at most 3 can be done in polynomial time [7]. NLC-width of at most 2 is decidable in polynomial time [26]. Clique-width of at most 2 and NLC-width 1 is decidable in linear time [8]. The clique-width of tree-width bounded graphs is also computable in linear time [14]. Oum and Seymour [32,31] have found polynomial time approximation algorithms for computing a clique-width $f(k)$-expression of a given graph of clique-width at most $k$, where $f(k)$ depends exponentially only on $k$. Fellows et al. [15–17] have shown that minimizing linear clique-width and clique-width is NP-complete.

The paper is organized as follows. In Section 2, we recall the definition of clique-width, NLC-width, tree-width and line graph. In Section 3, we recall the proof of [24] that the line graph$^1$ of a graph of tree-width $k$ has NLC-width at most $k+2$ and clique-width at most $2k+2$. Then we show that a graph of path-width $k$ and maximum vertex degree $r$ has linear NLC-width at most $k+2+\min\{\max\{k-2,0\}, \max\{r-2,0\}\}$ and linear clique-width at most $k+2+\min\{\max\{k-1,0\}, \max\{r-1,0\}\}$. In Section 4, we show that the root graph$^2$ of line graphs of clique-width or NLC-width at most $k$ has tree-width at most $4k-1$. Then we prove that the root graph of line graphs of linear clique-width or linear NLC-width at most $k$ has path-width at most $4k-1$. This shows a nice and new characterization of line graphs of bounded clique-width. A set of graphs has bounded tree-width or bounded path-width if and only if its set of line graphs has bounded clique-width or bounded linear clique-width. In Section 5, we improve the bounds given in Section 4 for the case of incidence graphs.$^3$ We show in Section 5 the following: (1) if the line graph of an incidence graph has clique-width or NLC-width at most $k$, then its root graph has tree-width at most $k$ and (2) if the line graph of an incidence graph has linear clique-width or linear NLC-width at most $k$, then its root graph has path-width at most $2k-1$. In Section 6 we show how these bounds can be used to show that NLC-width minimization is NP-complete. Approximation results for NLC-width and clique-width minimization are also discussed in Section 6.

2. Preliminaries

In this section, we recall the definitions of clique-width, NLC-width, tree-width, line graphs, and incidence graphs.

Let $[k] := \{1, \ldots, k\}$ be the set of all integers between 1 and $k$. We work with finite undirected vertex labeled graphs (labeled graphs for short) $G = (V_G, E_G, \text{lab}_G)$, where $V_G$ is a finite set of vertices labeled by some mapping $\text{lab}_G : V_G \to [k]$ 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

---

1 The line graph $L(G)$ of a graph $G$ has a vertex for every edge of $G$ and an edge between two vertices if the corresponding edges of $G$ are adjacent [40].

2 For some line graph $L(G)$, graph $G$ is called the root graph of $L(G)$.

3 The incidence graph $I(G)$ of a graph $G$ is the graph with vertex set $V_G \cup E_G$ and all edges joining $v \in V_G$ and $e \in E_G$ if and only if $v$ is incident to $e$ in $G$. 
additionally \(E_J = \{(u, v) \in E_G \mid u, v \in V_J\}\). The labeled graph consisting of a single vertex labeled by some \(a \in [k]\) is denoted by \(\bullet_a\).

The notion of clique-width for labeled graphs is defined by Courcelle and Olariu in [11].

**Definition 1 (Clique-width, Courcelle and Olariu [11]).** Let \(k\) be some positive integer. The class \(\text{CW}_k\) of labeled graphs is recursively defined as follows:

1. The single vertex \(\bullet_a\) labeled by some \(a \in [k]\) is in \(\text{CW}_k\).
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

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

is in \(\text{CW}_k\).
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
   
   \[
   \text{lab}'(u) := \begin{cases} 
   \text{lab}_G(u) & \text{if } \text{lab}_G(u) \neq a, \\
b & \text{if } \text{lab}_G(u) = a
   \end{cases}
   \]

is in \(\text{CW}_k\) and
   
   (b) \(\eta_{a, b}(G) := (V_G, E', \text{lab}_G)\) defined by
   
   \[
   E' := E_G \cup \{(u, v) \mid u, v \in V_G, u \neq v, \text{lab}_G(u) = a, \text{lab}_G(v) = b\}
   \]

is in \(\text{CW}_k\).

The notion of NLC-width\(^4\) of labeled graphs is defined by Wanke in [39].

**Definition 2 (NLC-width, Wanke [39]).** Let \(k\) be some positive integer. The class \(\text{NLC}_k\) of labeled graphs is recursively defined as follows:

1. The single vertex \(\bullet_a\) labeled by some \(a \in [k]\) is in \(\text{NLC}_k\).
2. Let \(G = (V_G, E_G, \text{lab}_G) \in \text{NLC}_k\) and \(R : [k] \rightarrow [k]\) be a mapping, then \(\circ_R(G) := (V_G, E_G, \text{lab}')\) defined by \(\text{lab}'(u) := R(\text{lab}_G(u))\) is in \(\text{NLC}_k\).
3. Let \(G = (V_G, E_G, \text{lab}_G) \in \text{NLC}_k\) and \(J = (V_J, E_J, \text{lab}_J) \in \text{NLC}_k\) be two vertex disjoint labeled graphs and \(S \subseteq [k]^2\) be a set of label pairs, then \(G \times S J := (V', E', \text{lab}')\) defined by \(V' := V_G \cup V_J, E' := E_G \cup E_J \cup \{(u, v) \mid u \in V_G, v \in V_J, (\text{lab}_G(u), \text{lab}_J(v)) \in S\}\), and

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

is in \(\text{NLC}_k\).

The _clique-width (NLC-width)_ of a labeled graph \(G\) is the least integer \(k\) such that \(G \in \text{CW}_k\) (\(G \in \text{NLC}_k\), respectively). An expression built with the operations \(\bullet, \oplus, \rho_{a \rightarrow b}, \eta_{a, b}\) for integers \(a, b \in [k]\) is called a _clique-width k-expression_. An expression built with the operations \(\bullet, \circ_R, \times_S\) for \(a \in [k]\), \(R : [k] \rightarrow [k]\), and \(S \subseteq [k]^2\) is called an _NLC-width k-expression_. The graph defined by an expression \(X\) is denoted by \(\text{val}(X)\). A vertex labeled graph \(G\) has _linear clique-width_ (linear NLC-width) at most \(k\) if it can be defined by a clique-width \(k\)-expression (an NLC-width \(k\)-expression,

\(^4\) The abbreviation NLC results from the _node label controlled_ embedding mechanism originally defined for graph grammars [12].
Table 1
Some clique-width and NLC-width expressions and the vertex labeled graphs defined by them

<table>
<thead>
<tr>
<th>Clique-width 2-expression X</th>
<th>Graph val(X)</th>
<th>NLC-width 2-expression X</th>
<th>Graph val(X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1 = \bullet_1$</td>
<td>$\bullet_1$</td>
<td>$G_1 = \bullet_1$</td>
<td>$\bullet_1$</td>
</tr>
<tr>
<td>$G_2 = \bullet_2$</td>
<td>$\bullet_2$</td>
<td>$G_2 = \bullet_2$</td>
<td>$\bullet_2$</td>
</tr>
<tr>
<td>$G_3 = G_1 \oplus G_2$</td>
<td>$\bullet_1 \bullet_2$</td>
<td>$G_3 = G_1 \times (1,2) G_2$</td>
<td>$\bullet_1 \bullet_2$</td>
</tr>
<tr>
<td>$G_4 = \eta_{1,2}(G_3)$</td>
<td>$\bullet_1 \bullet_2$</td>
<td>$G_4 = G_2 \times \eta G_3$</td>
<td>$\bullet_1 \bullet_2$</td>
</tr>
<tr>
<td>$G_5 = \rho_{2-1}(G_4)$</td>
<td>$\bullet_1 \bullet_1$</td>
<td>$G_5 = G_4 \times (2,1) G_1$</td>
<td>$\bullet_1 \bullet_1$</td>
</tr>
<tr>
<td>$G_6 = \eta_{1,2}(G_3 \oplus G_3)$</td>
<td>$\bullet_1 \bullet_2$</td>
<td>$G_6 = \rho_{(1,1),(2,1)}(G_3)$</td>
<td>$\bullet_1 \bullet_2$</td>
</tr>
</tbody>
</table>

respectively) where at least one argument of every operation $\oplus$ (of every operation $\times$, respectively) is a single labeled vertex $\bullet_a$ for some label $a \in [k]$, see also [23].

Every clique-width expression (NLC-width expression) has by its recursive definition a tree structure which we call the cli-que-width expression tree (NLC-width expression tree, respectively). Every complete subtree of an expression tree for some expression $X$ defines a subexpression of $X$. Table 1 shows an example of clique-width and NLC-width expressions and the vertex labeled graphs defined by them.

The notion of tree-width and path-width is defined by Robertson and Seymour in [34,33], respectively.

**Definition 3 (Tree-width and path-width, Robertson and Seymour [34,33])**. A tree decomposition of a graph $G = (V_G, E_G)$ is a pair $(\mathcal{X}, T)$ where $T = (V_T, E_T)$ is a tree and $\mathcal{X} = \{X_u \mid u \in V_T\}$ is a family of subsets $X_u \subseteq V_G$ one for each node $u$ of $T$ such that

1. $\bigcup_{u \in V_T} X_u = V_G$,
2. for every edge $\{v_1, v_2\} \in E_G$, there is some node $u \in V_T$ such that $v_1 \in X_u$ and $v_2 \in X_u$, and
3. for every vertex $v \in V_G$ the subgraph of $T$ induced by the nodes $u \in V_T$ with $v \in X_u$ is connected.

The width of a tree decomposition $(\mathcal{X} = \{X_u \mid u \in V_T\}, \ T = (V_T, E_T))$ is $\max_{u \in V_T} |X_u| - 1$. A tree decomposition $(\mathcal{X}, T)$ is called a path decomposition if $T$ is a path. The tree-width (path-width) of a graph $G$ is the smallest integer $k$ such that there is a tree decomposition (a path decomposition, respectively) $(\mathcal{X}, T)$ for $G$ of width $k$.

Fig. 1 shows a graph $G$ and a tree decomposition of width 2.

The notion of a line graph is introduced by Whitney [40]. The line graph $L(G)$ of a graph $G$ has a vertex for every edge of $G$ and an edge between two vertices if the corresponding edges in $G$ have a common vertex. Graph $G$ is called the root graph of $L(G)$. Whitney has shown that there are only two distinct5 graphs which define the same line graph, these are the cycle $C_3$ with three vertices and the claw $(K_{1,3})$. For a given line graph $L(G)$ the root graph $G$ can be found in linear time [36,28]. Line graphs can also be characterized by a finite number of forbidden induced subgraphs [3]. Fig. 2 shows a graph $G$, its line graph $L(G)$, the cycle $C_3$, and the claw.

The incidence graph $I(G)$ of a graph $G = (V_G, E_G)$ is the graph with vertex set $V_G \cup E_G$ and edge set $\{(u, e) \mid u \in V_G, e \in E_G, u \in e\}$. The incidence graph of $G$ is the graph we get, if we replace every edge $\{u, v\}$ of $G$ by a new vertex $w$ and two edges $\{u, w\}, \{w, v\}$. In an incidence graph every cycle has at least 6 vertices, and on every path every second vertex is of degree two. Fig. 2 also shows an example of an incidence graph $I(G)$ for some graph $G$.

---

5 Here we mean two graphs are distinct (equal) if they are non-isomorphic (isomorphic, respectively).
Fig. 1. A graph $G$ of tree-width 2 and a tree decomposition $(\mathcal{T}, T)$ for $G$ of width 2.

Fig. 2. A graph $G$, its line graph $L(G)$, its incidence graph $I(G)$, the cycle $C_3$, and the claw.

3. The clique-width of line graphs

In this section, we show that the line graph of a graph of tree-width at most $k$ has NLC-width at most $k + 2$ and clique-width at most $2k + 2$. After that, we show that the line graph of a graph of path-width at most $k$ has linear NLC-width at most $2k + 1$ and linear clique-width at most $2k + 2$. For graphs $G$ of path-width at most $k$ and maximum vertex degree $r$ we show that the line graph of $G$ has linear NLC-width at most $k + r + 1$ and linear clique width at most $k + r + 2$.

Graphs of tree-width at most $k$ are also characterized as partial $k$-trees [35]. A partial $k$-tree is a subgraph of a $k$-tree. A $k$-tree can be defined recursively by the following two instructions: (1) The complete graph with $k$ vertices is a $k$-tree and (2) if $G$ is a $k$-tree then the graph obtained by inserting a new vertex $u$ and $k$ edges between $u$ and all vertices of a $k$ vertex complete subgraph of $G$ is a $k$-tree. (A complete graph (also called a clique) is a graph with all possible edges.)

The following theorem is already shown in [24]. We prove this theorem here again, because we will modify the proof to achieve results about the relationship between the path-width of a graph $G$ and the linear NLC-width of line graph $L(G)$.

**Theorem 4** (Gurski and Wanke [24, Theorem 3]). The line graph of a partial $k$-tree (a graph of tree-width at most $k$) has NLC-width at most $k + 2$.

**Proof.** It suffices to show that the line graph of a $k$-tree $G$ has NLC-width at most $k + 2$, because the line graph of every subgraph of $G$ is an induced subgraph of the line graph of $G$, and the class NLC$_k$ is closed under taking induced subgraphs for every $k \geq 1$. 
Let \( G = (V_G, E_G) \) be a \( k \)-tree with \( n \) vertices. Let \( o = (u_1, \ldots, u_n) \) be an order of the \( n \) vertices of \( G \), i.e., every vertex of \( V_G \) appears in sequence \( o \) exactly once. Let \( N(G, o, i) \) for \( i = 1, \ldots, n \) be the set of neighbors \( u_j \) of vertex \( u_i \) with \( i < j \). That is,

\[
N(G, o, i) := \{ u_j | (u_j, u_i) \in E_G \land i < j \}.
\]

A vertex order \( (u_1, \ldots, u_n) \) for \( G \) is called a \textit{perfect elimination order} (PEO) if the vertices of \( N(G, o, i) \) for \( i = 1, \ldots, n \) induce a complete subgraph of \( G \).

There is always a vertex order \( o = (u_1, \ldots, u_n) \) for \( k \)-tree \( G \) such that the vertices of every \( N(G, o, i) \) for \( i = 1, \ldots, n-k \) induce a \( k \) vertex complete subgraph and the vertices of every \( N(G, o, i) \) for \( i = n-k+1, \ldots, n-1 \) induce an \( n-i \) vertex complete subgraph of \( G \). Here we can use, for example, the reverse order of the vertices from the recursive definition of \( k \)-tree \( G \). For the rest of the proof, let \( o \) be a PEO for \( G \).

Let \( col : V_G \to [k+1] \) be a \((k+1)\)-coloring of \( k \)-tree \( G \), that is, \( col(u_i) \neq col(u_j) \) for all edges \( (u_i, u_j) \in E_G \). It is easy to see that each \( k \)-tree is \( k+1 \) colorable, because we can assign to \( u_i \) any color not used by the vertices of \( N(G, o, i) \) for \( i = 1, \ldots, n-1 \).

Finally, let

\[
M(G, o) = \{ (u_j, u_i) | (u_j, u_i) \in E_G \land j < i \land \forall i', j < i' < i : \{ u_j, u_{i'} \} \notin E_G \}
\]

and

\[
M(G, o, i) = \{ u_j | (u_j, u_i) \in M(G, o) \land j < i \}
\]

for \( i = 1, \ldots, n \). The edges of \( M(G, o) \) with vertex set \( V_G \) define a tree, because for every vertex \( u_j, 1 \leq j < n \), there is exactly one vertex \( u_i \), where \( j < i \) and \( \{ u_j, u_i \} \in M(G, o) \), see also Fig. 3. The vertices of \( M(G, o, i) \) are the sons of vertex \( u_i \) in tree \( (V_G, M(G, o)) \) with root \( u_n \).

We next recursively define for \( i = 1, \ldots, n \) an NLC-width \((k+2)\)-expression \( X_i \) which defines the line graph of the \( k \)-tree \( G \).

1. Let \( M(G, o, i) = \{ u_{j_1}, \ldots, u_{j_m} \} \).
   
   (a) If \( m = 1 \) then let
   
   \[
   Y_i = X_{j_1}.
   \]
   
   (b) If \( m > 1 \) then let
   
   \[
   Y_i = X_{j_1} \times I \cdots \times I X_{j_m},
   \]
where \( I = \{(s, s) | s \in [k+1]\} \) is the identity between the labels \( 1, \ldots, k+1 \). Here graph \( val(Y_i) \) defined by expression \( Y_i \) is the disjoint union of \( m \) graphs \( val(X_{j_1}), \ldots, val(X_{j_m}) \) where equal labeled vertices from different graphs are joined by an edge. These connections concern only the labels \( 1, \ldots, k+1 \). The label \( k+2 \) will exclusively be used for vertices that will not be connected with other vertices in any further composition step.

2. Let \( N(G, o, i) = \{ u_{l_1}, \ldots, u_{l_r} \} \). If \( r > 0 \) then let \( Z_i \) be an NLC-width \((k+1)\)-expression that defines a complete graph with \( r \) vertices labeled by \( col(u_{l_1}), \ldots, col(u_{l_r}) \). Note that by the definition of the PEO the vertices \( \{ u_{l_1}, \ldots, u_{l_r} \} \) induce a complete subgraph of \( G \) and thus their colors are pairwise distinct and do not include the color of \( u_i \).

3. Next let

\[
X_i = \begin{cases} 
Z_i & \text{if } m = 0, \\
\circ_R(Y_i \times S Z_i) & \text{if } m > 0 \text{ and } r > 0, \\
\circ_R(Y_i) & \text{if } r = 0,
\end{cases}
\]

where

\[
S = \{(s, s) | s \in [k+1] - \{col(u_i)\}\} \cup \{\{col(u_i), s\} | s \in [k+1]\}
\]
and

\[
R(s) = \begin{cases} 
  s & \text{if } s \neq \text{col}(u_i), \\
  k + 2 & \text{if } s = \text{col}(u_i).
\end{cases}
\]

Fig. 3 shows a complete example of such a composition.

It remains to show that the NLC-width \((k + 2)\)-expression \(X_n\) defines the line graph of \(k\)-tree \(G\).
Assume $N(G, o, i) = \{u_{i1}, \ldots, u_{ir}\}$ and $r > 0$. In this case, we define an expression $Z_i$. The graph $\text{val}(Z_i)$ has exactly $r$ vertices labeled by the $r$ distinct colors of $u_{i1}, \ldots, u_{ir}$. For a vertex $v$ of $f(Z_i)$ let $\pi(v)$ be the unique edge $\{u_i, u_j\}$ where $u_j \in N(G, o, i)$ is the vertex whose color is the label of $v$, i.e., $\text{col}(u_j) = \text{lab}_{\text{val}(Z_i)}(v)$. Since all vertices of graph $\text{val}(X_n)$ result from subexpressions of the form $Z_i$ for some $i$, $1 \leq i < n$, we have a one-to-one mapping

$$\pi : \text{val}(X_n) \to E_G$$

between all vertices of the graph $\text{val}(X_n)$ and the edges of $G$. We will now show that two vertices $v_1, v_2$ of $\text{val}(X_n)$ are adjacent in $\text{val}(X_n)$ if and only if the two edges $\pi(v_1), \pi(v_2)$ of $G$ are adjacent in $G$.

Let $F_i \subseteq E_G, 1 \leq i \leq n$, be the set of all edges $\{u_{i1}, u_{i2}\}$ where $i_1, i_2 \leq i$, let $(V_i, E_i)$ be the connected component of $(V_G, F_i)$ to which vertex $u_i$ belongs, and let $\tilde{E}_i$ be the set of edges $\{u_{i1}, u_{i2}\}$ where $u_i \in V_i$ or $u_{i2} \in V_i$. The edges $\{u_{i1}, u_{i2}\}$ of $\tilde{E}_i$ for which one of the end vertices has an index greater than $i$ are called active edges. For the example of Fig. 3, we have $F_6 = \{(1, 2), (1, 3), (2, 3), (4, 5), (4, 6), (5, 6)\}, V_6 = \{4, 5, 6\}$, and $\tilde{E}_6 = \{(4, 5), (4, 6), (5, 6), (5, 9), (6, 7), (6, 9)\}, \{5, 9\}, \{6, 7\}$, and $\{6, 9\}$ are active edges.

A simple induction on $i$ shows that $\text{val}(X_i)$ is the line graph of $\tilde{G}_i = (V, \tilde{E}_i)$. Since $\tilde{E}_n = E_G$, because $G$ is connected, we finally get that $\text{val}(X_n)$ is the line graph of $G$. Additionally, we will see that a vertex $v$ of $\text{val}(X_i)$ is labeled by some label from $[k + 1]$ if and only if edge $\pi(v)$ is an active edge of $\text{val}(X_i)$. These vertices $v$ of $\text{val}(X_i)$ will be called the active vertices of $\text{val}(X_i)$. We will also see that active vertices with the same label are all mutually adjacent in every $\text{val}(X_i)$.

**Basis:** Let $i = 1$.

Let $M(G, o, i) = \{u_{j1}, \ldots, u_{jm}\}$ and $N(G, o, i) = \{u_{i1}, \ldots, u_{ir}\}$.

For $i = 1$, we have $m = 0$. The graph $\tilde{G}_1$ has $r$ edges $\{u_{i1}, u_{i1}\}, \ldots, \{u_{ir}, u_{ir}\}$. Since $m = 0$, we have $X_1 = Z_1$ and thus $\text{val}(X_1)$ is a complete graph with $r$ vertices $v_1, \ldots, v_r$ where $\pi(v_1) = \{u_{i1}, u_{i1}\}, \ldots, \pi(v_r) = \{u_{ir}, u_{ir}\}$, and the labels of $v_1, \ldots, v_r$ from $\text{val}(X_1)$ are the colors of the vertices $u_{i1}, \ldots, u_{ir}$ from $G$. All edges of $\tilde{G}_1$ are active edges and all vertices of $\text{val}(G_1)$ are active vertices. Graph $\text{val}(X_1)$ is obviously the line graph of $\tilde{G}_1$.

**Induction:** Let $i > 1$.

Let $M(G, o, i) = \{u_{j1}, \ldots, u_{jm}\}$ and $N(G, o, i) = \{u_{i1}, \ldots, u_{ir}\}$.

If $m = 0$, then in case $i = 1$ graph $\tilde{G}_1$ has $r$ active edges $\{u_{i1}, u_{i1}\}, \ldots, \{u_{ir}, u_{ir}\}, X_i = Z_i$, and $\text{val}(X_i)$ is a complete graph with $r$ vertices $v_1, \ldots, v_r$, where $\pi(v_1) = \{u_{i1}, u_{i1}\}, \ldots, \pi(v_r) = \{u_{ir}, u_{ir}\}$, and the labels of $v_1, \ldots, v_r$ from $\text{val}(X_i)$ are the colors of the vertices $u_{i1}, \ldots, u_{ir}$ from $G$.

If $m \geq 1$ then first some expression $Y_j$ is defined. The graph $\text{val}(Y_j)$ is the disjoint union of all $\text{val}(X_{j1}), \ldots, \text{val}(X_{jm})$ where all equal labeled active vertices from different graphs are connected. These vertices represent the active edges of $\tilde{G}_{j1}, \ldots, \tilde{G}_{jm}$. If two of these vertices $v_1, v_2$ have the same label, for example $k'$, then the two edges $\pi(v_1)$ and $\pi(v_2)$ have the common vertex $u_{j}, j \geq i$, whose color in $G$ is $k'$. Additionally, index $j$ is the least index greater than or equal to $i$ such that $u_j$ has color $k'$ in $G$.

If $r > 0$ then $Y_j$ will be combined with $Z_i$. In this step (1) all equal labeled vertices from $\text{val}(Y_j)$ and $\text{val}(Z_i)$ are connected by edges (here a vertex of $\text{val}(Y_j)$ will be connected with a vertex from $\text{val}(Z_i)$ if and only if it is also an active vertex in $\text{val}(X_j)$) and (2) all vertices of $\text{val}(Y_j)$ labeled by $\text{col}(u_i)$ will be connected with all vertices of $\text{val}(Z_i)$. (These vertices of $\text{val}(Y_j)$ are not active vertices of $\text{val}(X_j)$.)

The final relabeling ensures that the active vertices of the graphs $\tilde{G}_{j1}, \ldots, \tilde{G}_{jm}$ which are not active vertices of $\text{val}(X_i)$ get label $k + 2$. Thus, graph $\text{val}(X_i)$ is the line graph of $\tilde{G}_i$.

Since every graph of NLC-width at most $k$ has clique-width at most $2k$, Theorem 4 implies that the line graph $L(G)$ of a graph $G$ of tree-width at most $k$ has clique-width at most $2k + 4$. However, on closer examination of the construction in the proof of Theorem 4 this bound can be improved to $2k + 2$.

**Theorem 5.** The line graph of a partial $k$-tree (a graph of tree-width at most $k$) has clique-width at most $2k + 2$.

**Proof.** Consider the NLC-width $(k + 2)$-expressions $X_i, 1 \leq i \leq n$, defined for a $k$-tree $G$ with $n$ vertices as in the proof of Theorem 4. A simple induction on $i$ shows that for every NLC-width $(k + 2)$-expression $X_i$ there is an equivalent clique-width $(2k + 2)$-expression $X_i'$. 


2754
For $i = 1$ there is even a clique-width $(k + 1)$-expression $X'_1$ equivalent to $X_1$, because $\val(X_1)$ has at most $k$ vertices labeled by $k$ labels from $[k + 1]$.

For $i > 1$, an equivalent clique-width expression $Y'_i$ for $Y_i = X_{j_1} \times I_1 \times \cdots \times I_{j_m}$ can easily be defined by the clique-width expressions $X'_{j_1}, \ldots, X'_{j_m}$ defined for $X_{j_1}, \ldots, X_{j_m}$ and $k$ auxiliary labels, because for $t = 1, \ldots, m$ the vertices of every $\val(X'_{j_t})$ are labeled by $k + 1$ labels from $[k + 2]$. Label $\col(u_{j_t}) \in [k + 1]$ is not used by the vertices of $\val(X'_{j_t})$ and label $k + 2$ is not involved in any edge creation.

The clique-width expression $X'_i$ for $X_i = G \circ_R (Y_1 \times S Z_i)$ can finally be defined by clique-width expression $Y'_i$ defined for $Y_i$ and $k$ auxiliary labels, because $\val(Z_i)$ has at most $k$ vertices. □

A $k$-path can recursively be defined by the following two instructions. For the definition of a $k$-path $G$, we denote some vertices of $G$ by link vertices: (1) The complete graph with $k$ vertices that are all link vertices is a $k$-path and (2) if $G$ is a $k$-path then the graph obtained by inserting a new vertex $u$ and $k$ edges between $u$ and $k$ link vertices $u_1, \ldots, u_k$ of $G$ (which will always induce a complete subgraph of $G$) is a $k$-path. The new link vertices of the resulting $k$-path are $u, u_1, \ldots, u_k$. A partial $k$-path is a subgraph of a $k$-path. A graph is a partial $k$-path if and only if it has path-width at most $k$ [19].

**Theorem 6.** The line graph of a partial $k$-path (a graph of path-width at most $k$) with maximum vertex degree $r$ has

1. linear NLC-width at most $k + 2 + \min\{\max\{k - 2, 0\}, \max\{r - 2, 0\}\}$ and
2. linear clique-width at most $k + 2 + \min\{\max\{k - 1, 0\}, \max\{r - 1, 0\}\}$.

**Proof.** Let $\hat{G}$ be a partial $k$-path with $n$ vertices and maximum vertex degree $r$. Let $G$ be a $k$-path with the same vertex set as $\hat{G}$ such that $\hat{G}$ is a subgraph of $G$. Consider the NLC-width $(k + 2)$-expressions $X_i$, $1 \leq i \leq n$, defined for the line graph of $k$-path $G$ as in the proof of Theorem 4 in which $G$ is a $k$-tree. This works in the same way, because a $k$-path is always a $k$-tree. Fig. 4 shows a complete example of such a construction for a $k$-tree $G$.

(1) A simple induction on $i$ shows that for every such NLC-width $(k + 2)$-expression $X_i$ there is an equivalent linear NLC-width $(k + 2 + \max\{k - 2, 0\})$-expression $\hat{X}'_i$. We also show that every of these $(k + 2 + \max\{k - 2, 0\})$-expressions $\hat{X}'_i$ can easily be changed into an $(k + 2 + \max\{r - 2, 0\})$-expression $\hat{X}'_i$ such that $\hat{X}'_i$ defines only the edges which are in the line graph of the partial $k$-path $\hat{G}$. For $i = 1$ there is always a linear NLC-width $(k + 1)$-expression $X'_1$ equivalent to $X_1$, because $\val(X_1)$ has at most $k$ vertices labeled by $k$ distinct labels from $[k + 1]$. We get $\hat{X}'_1$ from $X'_1$ by removing all single vertex expressions which define vertices that do not belong to the line graph of $\hat{G}$.

Let $i > 0$. If $m > 1$ then all vertices $u_{j_1}, \ldots, u_{j_m}$ of $G$ have the same color. This color is not used as a vertex label in the graphs $\val(X'_{j_1}), \ldots, \val(X'_{j_m})$. Additionally, $m - 1$ of the graphs $\val(X'_{j_1}), \ldots, \val(X'_{j_m})$ are cliques with exactly $k$ vertices. Let $\val(X'_{j_2}), \ldots, \val(X'_{j_m})$ be these cliques with $k$ vertices.

An equivalent linear NLC-width expression $Y'_i$ for $Y_i$ can simply be defined with at most $k - 2$ auxiliary labels, if $k > 2$, because one of the labels from $[k + 1]$ is not used by the labeled graphs $\val(X'_{j_2}), \ldots, \val(X'_{j_m})$ and the single vertex of the last composition step can always get its final label. For the definition of $\hat{Y}'_i$, we know that the cliques $\val(\hat{X}'_{j_1}), \ldots, \val(\hat{X}'_{j_m})$ have at most $r$ vertices. That is, the linear NLC-width expression $\hat{Y}'_i$ can simply be defined with at most $r - 2$ auxiliary labels by the same argumentation.

The linear NLC-width expressions $X'_i$ ($\hat{X}'_i$) can now be defined from $Y'_i$ ($\hat{Y}'_i$, respectively) with at most $k - 2$ auxiliary labels if $k > 2$ (at most $r - 2$ auxiliary labels if $r > 2$, respectively), because $\val(Z'_i)$ has at most $k$ vertices (at most $r$ vertices, respectively), one of the labels from $[k + 1]$ is not used by $\val(Y'_i)$, and the single vertex of the last composition step always gets its final label.

(2) For the definition of the linear clique-width expressions $X'_i$ and $\hat{X}'_i$ we need one additional auxiliary label, because we cannot use for the single vertex of the last composition steps its final label, as we can do for linear NLC-width expressions. □
Fig. 4. A 2-path \( G \) and its line graph \( H \). The numbers at the vertices of \( G \) represent a perfect elimination order. The letters represent a 3-coloring \( V_G \to \{a, b, c\} \). The figure additionally shows the edge set \( M(G, o) \) and all labeled graphs defined by the linear NLC-width 4-expressions \( X_1 \) to \( X_7 \). The letters at the vertices of \( \text{val}(X_1) \) to \( \text{val}(X_7) \) are their labels. Vertices without such an index have label \( d \), which is not used for the definition of further edges. In the composition of \( \text{val}(X_6) \) two of the three graphs \( \text{val}(X'_j) \), \( \text{val}(X''_j) \), \( \text{val}(X'''_j) \) are cliques. All three graphs do not use the label \( e = \text{col}(u_3) = \text{col}(u_4) = \text{col}(u_5) \).

Note that the proofs of the Theorems 4–6 are all constructive, i.e., a (linear) NLC-width expression and a (linear) clique-width expression can simply be constructed in polynomial time from a given partial \( k \)-tree (partial \( k \)-path) \( G \), if a tree decomposition (path decomposition) for \( G \) is given.

4. The tree-width of root graphs

It is well known that tree-width and path-width bounded graphs can also be defined by a merging procedure of so-called terminal graphs, which are also called sourced graphs, see also [2]. We will define terminal graphs with edge labels, because this will allow us to define in an easy way the edge labeled root graphs of vertex labeled line graphs.

Let \( k, l \) be two positive integers. A \( k \)-terminal \( l \)-labeled graph is a system

\[
G = (V_G, E_G, P_G, \text{lab}_G),
\]

where \((V_G, E_G)\) is a graph, \( P_G = (u_1, \ldots, u_k) \) is a sequence of \( k \geq 0 \) distinct vertices of \( V_G \), and \( \text{lab}_G : E_G \to [l] \) is an edge labeling. The vertices in sequence \( P_G \) are called terminal vertices or terminals for short. The vertex \( u_i, \ 1 \leq i \leq k \), is the \( i \)th terminal of \( G \). The other vertices in \( V_G - P_G \) are called inner vertices. The \( k \)-terminal \( l \)-labeled graph consisting of \( r, 1 \leq r \leq k \), isolated terminals is denoted by \( \bullet' \).
Definition 7. Let \( k, l \) be two positive integers. The class \( \text{TM}_{k,l} \) of \( k \)-terminal \( l \)-labeled graphs is recursively defined as follows:

1. The \( k \)-terminal \( l \)-labeled graph \( \bullet^k \), \( 1 \leq r \leq k \), is in \( \text{TM}_{k,l} \).
2. The \( k \)-terminal \( l \)-labeled graph \( \bullet - \bullet \), \( a \in [l] \), consisting of two terminals \( u \) and \( v \) and an edge \( \{ u, v \} \) labeled by \( a \) is in \( \text{TM}_{k,l} \) for \( k \geq 2 \).
3. Let \( G = (V_G, E_G, P_G, \text{lab}_G) \in \text{TM}_{k,l} \), \( P = (u_1, \ldots, u_r) \), and \( f : [r] \rightarrow [r] \), be a bijection. Then the \( r \)-terminal \( l \)-labeled graph \( G^r = (V_G, E_G, P', \text{lab}_G) \) with \( P' = (u_{f(1)}, \ldots, u_{f(r)}) \) is in \( \text{TM}_{k,l} \).
4. Let \( G = (V_G, E_G, P_G, \text{lab}_G) \in \text{TM}_{k,l} \). \( P = (u_1, \ldots, u_r) \), and \( s \in [r] \). Integer \( s \) is also called a decrement. Then the \((r-1)\)-terminal \( l \)-labeled graph \( G_{s'} = (V_G, E_G, P', \text{lab}_G) \) with \( P' = (u_1, \ldots, u_{r-s}) \) is in \( \text{TM}_{k,l} \).
5. Let \( G = (V_G, E_G, P_G, \text{lab}_G) \in \text{TM}_{k,l} \) and \( R : [l] \rightarrow [l] \) be a relabeling mapping. Then the \( k \)-terminal \( l \)-labeled graph \( \circ_R(G) = (V_G, E_G, P', \text{lab}') \) with \( \text{lab}'(e) = R(\text{lab}(e)) \) for all \( e \in E_G \) is in \( \text{TM}_{k,l} \).
6. Let \( H = (V_H, E_H, P_H, \text{lab}_H) \in \text{TM}_{k,l} \), \( J = (V_J, E_J, P_J, \text{lab}_J) \in \text{TM}_{k,l} \), and \( |P_H| \leq |P_J| \). Then \( k \)-terminal \( l \)-labeled graph \( H \times J \) defined as follows is in \( \text{TM}_{k,l} \):
   a) Take the disjoint union of \((V_H, E_H, \text{lab}_H)\) and \((V_J, E_J, \text{lab}_J)\), and identify the \( i \)th terminal from \( H \) with the \( i \)th terminal from \( J \).
   b) An edge \( e \) from \( H \times J \) is labeled by \( \text{lab}(H \times J)(e) = \text{lab}_H(e) \) if it is from \( H \) and by \( \text{lab}_H \times J(e) = \text{lab}_J(e) \) if it is from \( J \).
   c) The \( i \)th terminal of \( H \times J \) is the \( i \)th terminal of \( J \).
   d) Multiple edges are eliminated by removing the corresponding edges from \( H \).

An expression built with the operations \( \bullet^k \), \( \bullet - \bullet \), \( |f| \), \( \circ_R \), and \( \times \) is called a terminal \( k \), \( l \)-expression. The terminal graph defined by a terminal \( k \), \( l \)-expression \( X \) is denoted by \( \text{val}(X) \). The class \( \text{PTM}_{k,l} \subseteq \text{TM}_{k,l} \) is the set of \( k \)-terminal \( l \)-labeled graphs defined by terminal \( k \), \( l \)-expressions where for every \( H \times J \) operation one of the terminal graphs \( H \) or \( J \) has no inner vertices. It is easy to see that \( \text{TM}_{k+1,l} \) and \( \text{PTM}_{k+1,l} \) define exactly the sets of graphs of tree-width at most \( k \) and path-width at most \( k \), respectively, see also Fig. 5. An alternative proof can be found in [2].

Proposition 8. A graph \( G = (V_G, E_G) \) has tree-width (path-width) at most \( k \) if and only if \((V_G, E_G, \emptyset, \text{lab}_G) \in \text{TM}_{k+1,l} \) ((\( V_G, E_G, \emptyset, \text{lab}_G \) \in \( \text{PTM}_{k+1,l} \)), respectively).

Proof. Let \( (X_u \mid u \in V_T) \), \( T = (V_T, E_T) \) be a tree decomposition for \( G \) of width at most \( k \). Consider \( T \) as a rooted tree, i.e., choose one node of \( T \) to be the root. For a node \( u \) of \( T \) let \( T_u \) be the complete subtree of \( T \) with root \( u \).

Let \( H_u = (V_{H_u}, E_{H_u}, P_{H_u}, \text{lab}_{H_u}) \) be a terminal graph where all edges are labeled by \( 1 \) such that \((V_{H_u}, E_{H_u})\) is the subgraph of \( G \) induced by the vertices of \( X_u \), and \( P_{H_u} \) is any but fixed arrangement of all vertices of \( V_{H_u} \). Then \( H_u \in \text{TM}_{k+1,l} \), because it has at most \( k + 1 \) terminals an no inner vertices.

Let \( G_u = (V_{G_u}, E_{G_u}, P_{G_u}, \text{lab}_{G_u}) \) be a terminal graph where all edges are labeled by \( 1 \) such that \((V_{G_u}, E_{G_u})\) is the subgraph of \( G \) induced by the vertices of all \( X_u \), \( w \in V_{T_u} \), and \( P_{G_u} \) is any but fixed arrangement of all vertices of \( X_u \). It is easy to see that every \( G_u \) is of \( \text{TM}_{k+1,l} \).

1. For every leaf \( u \) of \( T \), terminal graph \( G_u \) can be defined by \( G_u = (H_u)^{|f|} \), for some bijection \( f \).
2. For every inner node \( u \) of \( T \) with sons \( v_1, \ldots, v_r \) the terminal graph \( G_u \) can be defined by

\[
((G_{v_1})^{|f'_1|} |_{s_1})^{|f'_2|} \times \cdots \times (((G_{v_l})^{|f'_r|} |_{s_1})^{|f'_r|} \times H_u) \cdots,
\]

for bijections \( f_1, \ldots, f_r, f'_1, \ldots, f'_r \) and decrements \( s_1, \ldots, s_r \).

If \( (X_u \mid u \in V_T) \), \( T = (V_T, E_T) \) is a path decomposition for \( G \) of width at most \( k \), then \( H_u, G_u \in \text{PTM}_{k+1,l} \), because \( H_u \) has no inner vertices and every inner node \( u \) of \( T \) has at most one son.

Conversely, every expression that defines a terminal graph \( G = (V_G, E_G, P_G, \text{lab}_G) \in \text{TM}_{k+1,l} \) (terminal graph \( G = (V_G, E_G, P_G, \text{lab}_G) \in \text{PTM}_{k+1,l} \)) immediately defines a tree decomposition (path decomposition, respectively)
Let $G = (V_G, E_G, P_G, lab_G)$ be an edge labeled terminal graph, $\mathcal{G} = (V_G, E_{\mathcal{G}}, lab_{\mathcal{G}})$ be a vertex labeled graph, and $\pi : E_G \to V_G$ be a bijection such that (1) for every $e_1, e_2 \in E_G$, $e_1$ and $e_2$ have a common vertex if and only if $\pi(e_1)$ and $\pi(e_2)$ are adjacent in $\mathcal{G}$, and (2) for every $e \in E_G$, $lab_G(e) = lab_{\mathcal{G}}(\pi(e))$. Then $\mathcal{G}$ is called the labeled line graph of $G$, and $G$ is called a labeled terminal root graph of $\mathcal{G}$.

The next theorem shows a very tight connection between the tree-width of a graph and the NLC-width of its line graph.
Theorem 9. For every NLC-width $k$-expression $X$ that defines a line graph there is a mapping $\sigma$ that associates with every subexpression $X'$ of $X$ a terminal 4k, $k$-expression $\sigma(X')$ such that graph $\text{val}(X')$ is the labeled line graph of $\text{val}(\sigma(X'))$.

Proof. Let us first observe what happens if we insert edges between two vertex labeled line graphs by an NLC-width operation. Let $G = (V_G, E_G, \text{lab}_G)$ be an edge labeled graph with at least two edges. Let $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G}, \text{lab}_\mathcal{G}) \in \text{NLC}_k$ be the vertex labeled line graph of $G$ defined by some bijection $\pi : E_G \rightarrow V_\mathcal{G}$.

Every induced subgraph of $\mathcal{G}$ defines by bijection $\pi$ a unique subgraph of $G$ where every vertex is incident with at least one edge. Assume $G = \mathcal{H} \times \mathcal{J}$ for some $S \subseteq [k]^2$ and two non-empty vertex labeled graphs $\mathcal{H}$ and $\mathcal{J}$. Since $\mathcal{H}$ and $\mathcal{J}$ are induced subgraphs of $\mathcal{G}$, we know that they are line graphs of two subgraphs $H$ and $J$ of $G$. Since $\mathcal{H}$ and $\mathcal{J}$ are vertex disjoint, we know that $H$ and $J$ are edge disjoint. Since $\mathcal{H}$ and $\mathcal{J}$ have at least one vertex, we know that $H$ and $J$ have at least one edge. Assume further that every pair $(a, b) \in S$ defines at least one edge between a vertex of $\mathcal{H}$ and a vertex of $\mathcal{J}$, otherwise we remove $(a, b)$ from $S$. If $S$ is non-empty, then in $G$ at least one edge of $H$ has a common vertex with at least one edge of $J$.

We now show that $G$ can be defined by a vertex disjoint union of $H$ and $J$ and then identifying at most 4k vertices from $H$ with at most 4k vertices from $J$. A simple example of such a composition $\mathcal{H} \times S \mathcal{J}$ is shown in Fig. 6.

For a label $a \in [k]$ let $G_a$, $H_a$, and $J_a$ be the subgraphs $G$, $H$, and $J$, respectively, defined by the edges $e$ (and their end vertices) labeled by $a$. Let $(a, b) \in S$ be a pair of $S$. Then the operation $\times S$ connects every vertex of $\mathcal{H}$ labeled by $a$ with every vertex of $\mathcal{J}$ labeled by $b$. Thus, in root graph $G$ every edge from $H_a$ has a common vertex with every edge from $J_b$. Let $e = \{u, v\}$ be any edge from $H_a$. Then every edge from $J_b$ either contains vertex $u$ or vertex $v$. If $J_b$ has three or more edges, then at least two of them must have a common vertex. By the same argumentation, if $H_a$ has three or more edges then at least two of them must have a common vertex. Thus, $H_a$ and $J_b$ have at most two connected components. If $H_a$ has two connected components, then all edges of every connected component have exactly one common vertex, because an edge from $J_b$ can only contain one vertex from every of the two connected components of $H_a$. If $H_a$ is connected then it contains no simple path with 6 vertices and no simple cycle with 3 or 5 vertices. The simple path with 6 vertices and the simple cycle with 5 vertices do not contain two non-adjacent vertices $u, v$ such that every edge either contains $u$ or $v$. The cycle with 3 vertices not even contains two non-adjacent vertices.

This observation leads to a case distinction which divides all subgraphs $H_a$, $a \in [k]$, of $H$ into 8 distinct types as illustrated in Fig. 7. (The same holds for all subgraphs $J_b$, $b \in [k]$, of $J$.) Type 8 of Fig. 7 represents all graphs that have neither a vertex $u$ such that all edges are incident with $u$ nor two non-adjacent vertices $u, v$ such that every edge
is incident with $u$ or $v$. The subgraphs $H_a$ and $J_b$ of our example cannot be of Type 8, because the pair $(a, b)$ is used by operation $\times_S$ to create at least one edge between $\mathcal{H}$ and $J$, and $\mathcal{F} = \mathcal{H} \times J$ is the line graph of $G$.

Graphs of Type 1, 2, 3, and 5 have one connected component. Graphs of Type 4 and 6 have two connected components. Every graph of Type 1–7 has at most 4 vertices.

Graphs of Type 1, 2, 3, and 5 have one connected component. Graphs of Type 4 and 6 have two connected components.

Since the labels of $G$ are labeled by at most $k$ labels, it follows that at most 4$k$ vertices of $H$ are contained in $J$. That is, at most 4$k$ vertices of $H$ and at most 4$k$ vertices of $J$ have to be identified to define $G$ from a vertex disjoint union of $H$ and $J$. Graph $G$ itself has also at most 4$k$ vertices which can be identified with other vertices during further composition steps.

This allows us to define for an arbitrary NLC-width $k$-expression $X$ that defines a line graph a mapping $\sigma$ that associates for every subexpression $X'$ of $X$ a terminal 4$k$, $k$-expression $\sigma(X')$ such that $\text{val}(\sigma(X'))$ is the edge labeled terminal root graph of $\text{val}(X')$. We call a vertex $u$ of $\text{val}(\sigma(X'))$ incomplete if it is not yet incident with all edges of $\text{val}(\sigma(X))$.

1. If $X = \bullet_a$ for some $a \in [k]$ then let $\sigma(X) = \bullet_a^r \bullet_r$.
2. If $X = o_R(X')$ for some relabeling $R : [k] \to [k]$ then let $\sigma(X) = o_R(\sigma(X'))$.
3. If $X = X_1 \times S X_2$ for some $S \subseteq [k]^2$ then $\sigma(X)$ can be defined by

   $$\sigma(X) = ((\sigma(X_1) \times (\sigma(X_2) \times \bullet^*)|f_1)|f_2)|s$$

   with two bijections $f_1, f_2$, a decrement $s$, and some $r \leq 4k$. $\sigma(X)$ can be defined as above with some $r \leq 4k$, although not all terminals of $\text{val}(\sigma(X_1))$ need to be identified with terminals of $\text{val}(\sigma(X_2))$ via $\text{val}(\bullet^*)$, or vice versa. Let $a \in [k]$ be a label such that $\text{val}(\sigma(X_1))_a$ or $\text{val}(\sigma(X_2))_a$ has a terminal not identified with a terminal of $\text{val}(\sigma(X_2))_a$ or $\text{val}(\sigma(X_1))_a$, respectively, via $\text{val}(\bullet^*)$. Then the subgraph $(\sigma(X_1) \times (\sigma(X_2) \times \bullet^*)|f_1)_a$ is of Type 1 to 7, and at most 4 terminals of $\text{val}(\bullet^*)$ are identified with terminals of $\text{val}(\sigma(X_1))_a$ or $\text{val}(\sigma(X_2))_a$. Let $L \subseteq [k]$ be the set of all the remaining labels. Then for every $a \in L$ all terminals of $\text{val}(\sigma(X_1))_a$ are identified with terminals of $\text{val}(\sigma(X_2))_a$ via $\text{val}(\bullet^*)$, and all terminals from $\text{val}(\sigma(X_2))_a$ are identified with terminals of $\text{val}(\sigma(X_1))_a$ via $\text{val}(\bullet^*)$.

   That is, at most $4 \cdot |L|$ additional terminals are necessary to identify terminals of $\text{val}(\sigma(X_1))$ and $\text{val}(\sigma(X_2))$ via $\text{val}(\bullet^*)$.

   The subsequently applied decrement $s$ removes all vertices from the terminal vertex list that are no longer incomplete. □

A similar result can be shown for linear NLC-width $k$-expressions and terminal 4$k$, $k$-expressions that define the class $\text{PTM}_{4k,k}$.

**Theorem 10.** For every linear NLC-width $k$-expression $X$ that defines a line graph there is a mapping $\sigma$ that associates with every subexpression $X'$ of $X$ a terminal 4$k$, $k$-expression $\sigma(X')$ such that graph $\text{val}(X')$ is the labeled line graph of $\text{val}(\sigma(X'))$ and $\text{val}(\sigma(X')) \in \text{PTM}_{4k,k}$. 

![Fig. 7. Eight types for the subgraphs $H_a$ and $J_b$ of $H$ and $J$, respectively. The specific vertices are framed by squares.](Image)
Proof. For a linear NLC-width \( k \)-expression \( X = X_1 \times X_2, S \subseteq [k]^2 \), either \( X_1 = \bullet_a \) or \( X_2 = \bullet_a \) holds true for some \( a \in [k] \). Let \( \sigma \) be the mapping defined in Theorem 9. If \( \sigma(X) = (((\sigma(X_1) \times (\sigma(X_2) \times \bullet'))[f])^1)[f]) \) for two bijections \( f_1, f_2 \) and a decrement \( s \), then either \( \sigma(X_1) \) or \( \sigma(X_2) \) is of the form \( \bullet_a \), and thus either \( \text{val}(\sigma(X_1)) \) or \( \text{val}((\sigma(X_2) \times \bullet'))[f]) \) has no inner vertices.  □

Since the NLC-width (linear NLC-width) of a graph is always less than or equal to its clique-width (linear clique-width, respectively) [25,23], Proposition 8 in connection with Theorems 9 and 10 yields the following corollary.

Corollary 11.

(1) If a line graph has NLC-width or clique-width at most \( k \), then its root graph has tree-width at most \( 4k - 1 \).

(2) If a line graph has linear NLC-width or linear clique-width at most \( k \), then its root graph has path-width at most \( 4k - 1 \).

5. Line graphs of incidence graphs

The next proposition improves the bound of Theorem 9 for line graphs of incidence graphs.

Proposition 12. For every NLC-width \( k \)-expression \( X \) that defines a line graph of an incidence graph there is a mapping \( \sigma \) that associates with every subexpression \( X' \) of \( X \) a terminal \( 2k \)-expression \( \sigma(X') \) such that \( \text{graph } \text{val}(X') \) is the labeled line graph of \( \text{graph } \text{val}(\sigma(X')) \).

Proof. Let us now observe what happens if we insert edges between two vertex labeled line graphs by an NLC-width operation \( \mathcal{G} = \mathcal{H} \times \mathcal{J} \), \( S \subseteq [k]^2 \) where the root graphs \( G, H, J \) of \( \mathcal{G}, \mathcal{H}, \) and \( \mathcal{J} \), respectively, are incidence graphs.

The following discussion frequently uses the facts that an incidence graph (and also any subgraph of an incidence graph) has one end vertex of degree at most 2. If an incidence graph has no cycle of length \( a \) labeled by \( \mathcal{J} \), then only two of its four vertices need to be terminals of \( G \), and a decrement \( s \) would then create a cycle of length three or four. If \( G_a \) is of Type 6, then all four vertices of \( G_a \) need to be terminals of \( G \). Here again any additional edge adjacent with all edges of \( G_a \) would then create a cycle of length three.

This discussion shows that every subgraph \( G_a, a \in [k] \), of \( G \) can be divided into four types as illustrated in Fig. 8. Type 4 of Fig. 8 represents all incidence graphs with two non-adjacent vertices \( u, v \) and an edge not incident with \( u \) or \( v \). If \( G_a \) is of Type 4, then no vertex of \( G_a \) needs to be a terminal of \( G \). A similar argumentation as in the proof of Theorem 9 now shows that for an arbitrary NLC-width \( k \)-expression \( X \) that defines the line graph of an incidence graph there is a mapping \( \sigma \) that associates for every subexpression \( X' \)

![Fig. 8. Four types for the subgraphs \( G_a \) of a terminal incidence graph \( G \). The specific vertices are framed by squares.](image-url)
of $X$ a terminal $2k, k$-expression $\sigma(X')$ such that $\text{val}(\sigma(X'))$ is the edge labeled terminal root graph of $\text{val}(X')$. □

Analogously to the proof of Theorem 10 we get the following proposition.

**Proposition 13.** For every linear NLC-width $k$-expression $X$ that defines a line graph of an incidence graph there is a mapping $\sigma$ that associates with every subexpression $X'$ of $X$ a terminal $2k, k$-expression $\sigma(X')$ such that graph $\text{val}(X')$ is the labeled line graph of $\text{val}(\sigma(X'))$ and $\text{val}(\sigma(X')) \in \text{PTM}_{2k,k}$.

Propositions 12 and 13 in connection with Proposition 8 yield the following corollary.

**Corollary 14.**

(1) If the line graph of an incidence graph $G$ has NLC-width or clique-width at most $k$, then $G$ has tree-width at most $2k - 1$.

(2) If the line graph of an incidence graph $G$ has linear NLC-width or linear clique-width at most $k$, then $G$ has path-width at most $2k - 1$.

In Theorem 17 below we will improve Corollary 14.(1).

**Definition 15** (Well-connected terminal $k, l$-expression). Let $k, l$ be two positive integers. A well-connected terminal $k, l$-expression is a terminal $k, l$-expression, if any subexpressions is of the form

(1) $Y = \bullet \circ \bullet$ for some $a \in [l]$,

(2) $Y = Y_1^f$ for some bijection $f$,

(3) $Y = Y_1^s$ for some decrement $s$,

(4) $Y = \circ R(Y_1)$ for some relabeling $R$, or

(5) $Y = (Y_1 \times (Y_2 \times \bullet')^{|f|})^{|f|}$ for bijections $f_1, f_2$ and some $r \leq k$,

where every graph $\text{val}(Y)$ is connected.

**Proposition 16.** Let $k, l$ be two positive integers. For every terminal $k, l$-expression that defines a connected terminal graph, there is an equivalent well-connected terminal $k, l$-expression.

**Proof.** Assume a terminal graph $G = (V_G, E_G, P_G, \text{lab}_G)$ has $r > 1$ connected components $(V_1, E_1), \ldots, (V_r, E_r)$ with at least one edge. Let $P_i$ be the sequence $P_G$ where all vertices not in $V_i$ are removed. That is, if $P_G = (u_1, \ldots, u_r)$, then $P_i, 1 \leq i \leq r$, is the sequence $(u_{i_1}, \ldots, u_{i_s})$ with $i_j < i_{j+1}$ for $j = 1, \ldots, s - 1$ and $\{u_{i_1}, \ldots, u_{i_s}\} = V_i \cap \{u_1, \ldots, u_r\}$. Let $\text{lab}_i$ be the mapping lab restricted to the set of edges $E_i$. The terminal graphs $G_i = (V_i, E_i, P_i, \text{lab}_i), 1 \leq i \leq r$, are called the terminal edge connected components of $G$.

We decompose every terminal graph with at least one edge into its terminal edge connected components as follows. Terminal vertices not incident to edges will be ignored, because they will be inserted later when the edges to these vertices are inserted.

(1) Let $G = \bullet'$. Then $G$ has no terminal edge connected components.

(2) Let $G = \bullet \circ \bullet$. Then $G$ is a terminal edge connected component.

(3) Let $G = H_i^f$ for $H \in \text{TM}_{k,l}$ and some bijection $f$, and let $H_1, \ldots, H_r$ be the terminal edge connected components of $H$. Then the edge connected components of $G_i, 1 \leq i \leq r$, of $G$ can be defined by $G_i = H_i^{|f|}$ with bijections $f_i, 1 \leq i \leq r$, obtained from $f$.

(4) Let $G = H_i^s$ for $H \in \text{TM}_{k,l}$ and some decrements $s$, and let $H_1, \ldots, H_r$ be the terminal edge connected components of $H$. Then every terminal edge connected component $G_i, 1 \leq i \leq r$, of $G$ can be defined by $G_i = (H_i^{|f|})_{s_i}$ with bijections $f_i$ and decrements $s_i, 1 \leq i \leq r$.

(5) Let $G = \circ R(H)$ for $H \in \text{TM}_{k,l}$ and some relabeling $R$, and let $H_1, \ldots, H_r$ be the terminal edge connected components of $H$. Then every terminal edge connected component $G_i, 1 \leq i \leq r$, of $G$ can be defined by $G_i = \circ R(H_i)$. 


Let $G = H \times J$ for $H, J \in \text{TM}_{k,l}$. Then every terminal edge connected component $G_i$, $1 \leq i \leq r$, of $G$ can be defined as follows. Let $G'_1, \ldots, G'_r$ be the terminal edge connected components of $H$ and $J$ that have at least one edge of $G_i$. Now it is easy to define terminal graphs $G''_1, \ldots, G''_r$, bijections $f_1, \ldots, f_{r-1}$, $f'_1, \ldots, f'_{r-1}$, and integers $r_1, \ldots, r_{r-1} \leq k$ such that

(a) $G''_1 = G'_1$,

(b) $G''_j = (G'_j \times (G''_{j-1} \times \bullet^{f_{j-1}})|^{f_{j-1}})$

for $j = 2, \ldots, r$, and

(c) $G_i = G''_{r_i}$.

By the connectivity structure of $G$ it is always possible to order the terminal edge connected components $G'_1, \ldots, G'_{r-1}$ such that every $G''_j$, $1 \leq j \leq r$, is connected.

Every connected terminal graph $G$ can now be defined by subexpressions of the required form which define connected terminal graphs. The graphs $\text{val}(\bullet^a)$ are only used in the composition steps defined for Case 6. □

**Theorem 17.** If the line graph of an incidence graph $G$ has NLC-width at most $k$, then $G$ has tree-width at most $k$.

**Proof.** Let $X$ be an NLC-width $k$-expression for a (without loss of generality connected) line graph of an incidence graph. Let $\sigma$ be the mapping of Theorem 12 that associates with every subexpression $X'$ of $X$ a terminal $2k$, k-expression $\sigma(X')$ such that graph $\text{val}(X')$ is the labeled line graph of $\text{val}(\sigma(X'))$.

We first transform $\sigma(X)$ as explained in the proof of Proposition 16 into a well-connected terminal $2k$, k-expression $Y$. This is possible, because the final root graph $\sigma(X)$ is connected.

Now every subexpression $Y'$ of $Y$ is of the form

1. $Y' = \bullet^a \bullet$ for some $a \in [k]$,
2. $Y' = Y'_1|^{f}$ for some bijection $f$,
3. $Y' = Y'_1|_{s}$ for some decrement $s$,
4. $Y' = \circ_{R}(Y'_1)$ for some relabeling $R$, or
5. $Y' = ((Y'_1 \times (Y'_2 \times \bullet^a)|^{f_1})|^{f_2})|_{s}$ for bijections $f_1, f_2$, some $r \leq 2k$, and a decrement $s$.

These subexpressions define connected terminal graphs. For every of these subexpressions $Y'$ there is an NLC-width $k$-expression $X'$ such that $\text{val}(Y')$ is the edge labeled root graph of the vertex labeled line graph $\text{val}(X')$.

Now we will show that $Y$ can be transformed into an equivalent terminal $k+1$, k-expression. Let $Y'$ be a subexpressions of $Y$ of the form stated above and let $G = \text{val}(Y')$. Let again $G_a$ for some $a \in [k]$ be the terminal subgraph of $G$ defined by the edges (and their end vertices) labeled by $a$.

1. If all subgraphs $G_a$, $a \in [k]$, of $G$ are of Type 1 of Fig. 8, then $G$ has at most $k$ edges. Since $G$ is connected, it has at most $k + 1$ terminals.
2. If all subgraphs $G_a$, $a \in [k]$, of $G$ are of Type 1, 2, or 4 of Fig. 8, and at least one of these subgraphs is of Type 2 or 4, then $G$ has at least one inner vertex. In this case $G$ has at most $k$ terminals. This is easy to see by the following observation. Order the edges of $G$ in a sequence $e_1, \ldots, e_m$ such that every subgraph $\hat{G}_i$ of $G$ induced by the vertices of the edges $e_1, \ldots, e_i$ for $i = 1, \ldots, m$ is connected, and one of the end vertices of the first edge $e_1$ is an inner vertex. Then the number of vertices of $\hat{G}_i$, $2 \leq i \leq m$, that are terminals in $G$ is the number of labels $a \in [k]$ such that $\hat{G}_i$ has an edge from $G_a$. If the label of $e_{i+1}$ is already an edge label of $\hat{G}_i$, then $\hat{G}_i$ and $\hat{G}_{i+1}$ have the same terminals of $G$. If the label of $e_{i+1}$ is not an edge label of $\hat{G}_i$, then $\hat{G}_{i+1}$ has at most one additional terminal.
3. If some subgraph $G_a$, $a \in [k]$, of $G$ is of Type 3, then two vertices of $G_a$ are terminals of $G$. These two vertices $u_a, v_a$ are adjacent in the root graph $\text{val}(Y)$, otherwise they would be complete and thus not terminals of $G$. We also know that during any further composition these two vertices will get incident only with the missing edge $\{u_a, v_a\}$. We now modify the expression as follows.
A subgraph of Type 3 can only be created in the following two cases:

(a) Let

\[ G = \circ_R(H) \]

be a graph such that \( G \) has a subgraph \( G_a, a \in [k] \) of Type 3, but \( H \) has no subgraph of Type 3. Then \( H \) is connected and at least one inner vertex, and thus \( H \) has at most \( k \) terminals. We insert the edge between \( u_a \) and \( v_a \) now by

\[ G = ((a \times \circ_R(H)|f_1)|f_2)|s \] with three bijections \( f_1, f_2, f_3 \) and a decrement \( s = 2 \). (This can be done for all subgraphs \( G_a, a \in [k], \) of \( G \) of Type 3 step by step.) The decrement \( s = 2 \) removes the two vertices \( u_a, v_a \) from the terminal vertex list, because these vertices are now complete. The composition step which originally inserts the edge between \( u_a \) and \( v_a \) will be omitted. In any succeeding composition step the vertices \( u_a \) and \( v_a \) do not get any further connection with other edges, thus \( u_a \) and \( v_a \) can be removed from all terminal vertex lists of the terminal graphs which will be combined with \( G \) later in the composition.

(b) Let

\[ G = (H \times (J \times \bullet'))|f_2 \]

be a graph such that \( G \) has a subgraph \( G_a \) of Type 3, but \( H \) and \( J \) have no subgraphs of Type 3. Then \( H \) and \( J \) are connected and have at least one inner vertex, thus \( H \) and \( J \) have at most \( k \) terminals. Let \( u_a \) from \( H \) and \( v_a \) from \( J \). We insert the edge between \( u_a, v_a \) of \( G_a \) by

\[ G = ((H|f_3 \times ((J|f_2 \times (\bullet \times (\bullet'))|s_1 \times \bullet')|s_2)))|f_5 \]

with bijections \( f_1, f_2, f_3, f_4, f_5 \) and decrements \( s_1 = 1, s_2 = 1 \). If \( J \) has \( k' \) terminals then \( r' = k' + 1 \). Let \( u_a \) be from \( H \) and \( v_a \) be from \( J \). One end vertex of edge \( \bullet \circ \bullet \) will be identified with the terminal \( v_a \) of \( J \). Decrement \( s_1 = 1 \) will remove this vertex from the vertex list. The other end vertex of edge \( \bullet \circ \bullet \) will then be identified with \( u_a \) from \( H \). The final restriction \( s_2 = 1 \) will remove this vertex from the vertex list, see also Fig. 9. (This can be done for all subgraphs \( G_a, a \in [k], \) of \( G \) of Type 3 step by step in the same way.) In any succeeding composition step the vertices \( u_a \) and \( v_a \) will be removed from the terminal vertex lists of the graphs. The composition step which originally inserts the edge between \( u_a \) and \( v_a \) will be omitted.

Note that the edges inserted in the two cases above in many cases do not get the labeling they have in the original final root graph, but this is not important for the statement of the theorem.

![Diagram](image_url)
Now the resulting composition is set up with terminal graphs that have at most \( k + 1 \) terminals. \( \Box \)

Since the NLC-width of a graph is always less than or equal to its clique-width [25], Theorem 17 also holds for line graphs of incidence graphs of clique-width at most \( k \).

**Corollary 18.** If the line graph of an incidence graph \( G \) has clique-width at most \( k \), then \( G \) has tree-width at most \( k \).

### 6. The NP-completeness of NLC-width minimization

A very well known property of incidence graphs is that a graph \( G \) has tree-width \( k \) if and only if its incidence graph \( I(G) \) has tree-width \( k \). This follows immediately from the fact that neither an edge contraction nor a subdivision, the replacement of an edge \( \{u, v\} \) by a new vertex \( w \) and two edges \( \{u, w\}, \{w, v\} \), increases the tree-width of a graph, see for example [29]. It is also easy to see that an incidence graph \( I(G) \) has path-width at most \( k + 1 \) if \( G \) has path-width at most \( k \).

Theorems 4–6, 17, and Corollaries 11, 14, and 18 together now imply the following bounds.

\[
\begin{align*}
(1) \quad \text{tree-width}(G) + 1 & \leq \text{NLC-width}(L(G)) \leq \text{tree-width}(G) + 2, \\
(2) \quad \text{tree-width}(G) + 1 & \leq \text{clique-width}(L(G)) \leq 2 \cdot \text{tree-width}(G) + 2, \\
(3) \quad \text{path-width}(G) + 1 & \leq \text{linear NLC-width}(L(G)) \leq 2 \cdot \text{path-width}(G), \\
(4) \quad \text{path-width}(G) + 1 & \leq \text{linear clique-width}(L(G)) \leq 2 \cdot \text{path-width}(G) + 1, \\
(5) \quad \text{tree-width}(G) & \leq \text{NLC-width}(L(I(G))) \leq \text{tree-width}(G) + 2, \\
(6) \quad \text{tree-width}(G) & \leq \text{clique-width}(L(I(G))) \leq 2 \cdot \text{tree-width}(G) + 2, \\
(7) \quad \text{path-width}(G) + 1 & \leq \text{linear NLC-width}(L(I(G))) \leq 2 \cdot \text{path-width}(G) + 2, \\
(8) \quad \text{path-width}(G) + 1 & \leq \text{linear clique-width}(L(I(G))) \leq 2 \cdot \text{path-width}(G) + 3.
\end{align*}
\]

Inequality (5) can be used to show that NLC-width minimization is NP-complete.

**Theorem 19.** Given a graph \( G \) and an integer \( k \), the problem to decide whether \( G \) has NLC-width at most \( k \) is NP-complete.

**Proof.** The problem to decide whether a given graph has NLC-width at most \( k \) is obviously in NP.

For a graph \( G = (V, E) \) and some integer \( r > 1 \) let \( G' \) be the graph \( G \) where every vertex \( u \) is replaced by a clique \( C_u \) with \( r \) vertices and every edge \( \{u, v\} \) is replaced by all edges between the vertices of \( C_u \) and \( C_v \). That is, \( G' = (V_r, E_r) \) has vertex set

\[ V_r = \{u_{i,j} | u_i \in V, j \in \{1, \ldots, r\}\} \]

and edge set

\[ E_r = \{\{u_{i,j}, u_{i',j'}\} | u_{i,j}, u_{i',j'} \in V_r \text{ and } (i = i' \lor \{u_i, u_{i'}\} \in E)\}. \]

Bodlaender et al. have shown in [5], that \( G \) has tree-width \( k \) if and only if \( G' \) has tree-width \( r(k + 1) - 1 \).

Arnborg et al. have shown in [1] that given a graph \( G \) and an integer \( k \), the problem to decide whether \( G \) has tree-width at most \( k \) is NP-complete.

For a given graph \( G \), we first construct the graph \( G^3 \), then the incidence graph \( I(G^3) \), and then the line graph \( L(I(G^3)) \). This can be done in polynomial time. If \( G \) has tree-width \( k \), then \( G^3 \) has tree-width \( 3k + 2 \), and \( I(G^3) \) has tree-width \( 3k + 2 \). By Theorem 17, \( L(I(G^3)) \) has NLC-width at least \( 3k + 2 \), and by Theorem 4, NLC-width at most \( 3k + 4 \). That is,

\[ \text{tree-width}(G) = \left\lfloor \frac{\text{NLC-width}(L(I(G^3))) - 2}{3} \right\rfloor. \]
Thus, a graph $G$ has tree-width at most $k$ if and only if $L(I(G^3))$ has NLC-width at most $3k + 4$ which completes our proof. □

In [5] it is also shown that there is no polynomial time approximation algorithm for tree-width with constant difference guarantee, unless $P = NP$. That is, for every positive integer $c$ there is no polynomial time algorithm that computes for a given graph $G$ a tree decomposition of width $k$ such that $k − \text{tree-width}(G) \leq c$, unless $P = NP$. By inequality (5), an approximation algorithm for NLC-width with difference guarantee $c$ yields an approximation algorithms for tree-width with difference guarantee $c + 2$.

**Corollary 20.** For every positive integer $c$ there is no polynomial time algorithm that computes for a given graph $G$ an NLC-width $k$-expression such that $k − \text{NLC-width}(G) \leq c$, unless $P = NP$.

Bodlander et al. have also shown in [5] that for every $\varepsilon, 0 < \varepsilon < 1$, there is no polynomial time algorithm that computes for a given graph $G$ a tree decomposition of width $k$ such that $k − \text{tree-width}(G) \leq |V_G|^\varepsilon$, unless $P = NP$, and following the proof of Theorem 23 of [5], there is also no such algorithms such that $k − \text{tree-width}(G) \leq 2 + |V_G|^\varepsilon$, unless $P = NP$. By our discussion above, any algorithm that computes for a given graph $G = (V, E)$ an NLC-width decomposition of width $k$ such that $k − \text{NLC-width}(G) \leq |V_G|^\varepsilon$ yields an algorithms that computes for a given graph $G' = (V', E')$ a tree decomposition of width $2 + (2|E'|)^\varepsilon$. Since $|E'| < |V'|^2 / 2$, the algorithm also computes a tree decomposition of width $2 + |V'|^{2\varepsilon}$.

**Corollary 21.** For every $\varepsilon, 0 < \varepsilon < \frac{1}{2}$, there is no polynomial time algorithm that computes for a given graph $G$ an NLC-width $k$-expression such that $k − \text{NLC-width}(G) \leq |V_G|^\varepsilon$, unless $P = NP$.

On the other hand, any polynomial time approximation algorithm for clique-width with performance $c$, i.e., any algorithm that decides whether $G$ has clique-width at most $k$ and $k / \text{clique-width}(G) \leq c$, respectively, can simply be transformed into a polynomial time approximation algorithm for tree-width with performance $4c$. Such a polynomial time approximation algorithm for tree-width is not known up to now.

Theorems 4 and 5 can also be used to prove the NP-completeness of graph problems on NLC-width and clique-width bounded graphs. For example, the edge disjoint paths problem is NP-complete for graphs of tree-width 2, if the number of paths is unfixed, see [30]. Theorems 4 and 5 now imply that the vertex disjoint paths problem is NP-complete for graphs of NLC-width at most 4 and clique-width at most 6, if the number of paths is unfixed. Note that the vertex disjoint paths problem can be solved in polynomial time for tree-width bounded graphs [37] and co-graphs (graphs of clique-width at most 2) [21]. This is the first problem that separates co-graphs and clique-width bounded graphs from a complexity point of view.

Since the chromatic number for NLC-width and clique-width bounded graphs is computable in polynomial time [13], Theorems 4 and 5 imply that the chromatic index of a graph of bounded tree-width can be computed in polynomial time. This re-proves a result by Bodlaender [4] in a very simple way.

Corollary 11 shows that certain classes of line graphs are of unbounded clique-width. Since the set of all complete graphs, all complete bipartite graphs and all grids have unbounded tree-width, it follows that the set of all line graphs of complete graphs (as mentioned in [27]), the set of all line graphs of complete bipartite graphs (which was open up to now), and the set of all line graphs of grids have unbounded clique-width. More precisely, since a complete graph $K_n$ with $n$ vertices has tree-width $n − 1$, a complete bipartite graph $K_{n,m}$ with $n + m$ vertices has tree-width $\min(n, m)$, and an $n \times m$ grid $G_{n,m}$ with $n \cdot m$ vertices has tree-width $\min(n, m)$, it follows that the lines graphs of these graphs have clique-width $\geq n/4$ and $\min(n, m) + 1)/4$, respectively. The bound for complete graphs improves the bound $n/24$ of [27].

**References**


