The NLC-width and clique-width for powers of graphs of bounded tree-width

Frank Gurski *, Egon Wanke

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

ARTICLE INFO

Article history:
Received 15 October 2007
Received in revised form 26 July 2008
Accepted 25 August 2008
Available online 24 September 2008

Keywords:
Tree powers
Leaf powers
Steiner powers
Tree-width
Clique-width
NLC-width

ABSTRACT

The $k$-power graph of a graph $G$ is a graph with the same vertex set as $G$, in that two vertices are adjacent if and only if, there is a path between them in $G$ of length at most $k$. A $k$-tree-power graph is the $k$-power graph of a tree, a $k$-leaf-power graph is the subgraph of some $k$-tree-power graph induced by the leaves of the tree.

We show that (1) every $k$-tree-power graph has NLC-width at most $k + 2$ and clique-width at most $k + 2 + \max\{\lfloor k/2 \rfloor - 1, 0\}$, (2) every $k$-leaf-power graph has NLC-width at most $k$ and clique-width at most $k + \max\{\lfloor k/2 \rfloor - 2, 0\}$, and (3) every $k$-power graph of a graph of tree-width $l$ has NLC-width at most $(k + 1)^{l+1} - 1$ and clique-width at most $2 \cdot (k + 1)^{l+1} - 2$.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

The clique-width of a graph $G$ is the least integer $k$, such that $G$ can be defined by operations on vertex-labeled graphs using $k$ labels [2]. These operations are the vertex disjoint union, the addition of edges between vertices controlled by a label pair, and the relabeling of vertices. The NLC-width of a graph $G$ is defined similarly in terms of closely related operations [3]. 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. 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 $2k$ [4]. 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 [5].

The concept of clique-width generalizes the well-known concept of tree-width defined in [6] by the existence of a tree-decomposition. Graphs of bounded clique-width and graphs of bounded tree-width are particularly interesting from an algorithmic point of view. Many NP-complete graph problems can be solved in polynomial time for graphs of bounded clique-width [7–9] and for graphs of bounded tree-width [10], respectively.

A well known concept in graph theory is the concept of graph powers [11]. The $k$-power graph $G^k$ of a graph $G$ is a graph with the same vertex set as $G$. Two vertices in $G^k$ are adjacent if and only if, there is a path between them in $G$ of length at

---

* Parts of this paper have been published in an extended abstract [F. Gurski, E. Wanke, The clique-width of tree power and leaf power graphs (Extended Abstract), in: Proceedings of Graph-Theoretical Concepts in Computer Science, in: LNCS, vol. 4769, Springer, 2007, pp. 76–85].

* Corresponding author.

E-mail addresses: gurski-dam@acs.uni-duesseldorf.de (F. Gurski), wanke-dam@acs.uni-duesseldorf.de (E. Wanke).

0166-218X/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2008.08.031
most k. Determining whether a given graph is a k-power graph is NP-complete for every fixed integer k ≥ 2 [12]. A tree-power graph is the power graph of a tree, a leaf-power graph is the subgraph of a tree-power graph induced by the leaves of the tree.

In this paper, we determine the clique-width and NLC-width of power graphs of graphs of bounded tree-width and leaf-power graphs [1]. The clique-width of k-power graphs of graphs of bounded clique-width was analyzed the first time by Suchan and Todinca in [13,14]. In [13], it is stated that the k-power graph of a graph of clique-width l has clique-width at most 2 · l · k^l. Since trees have NLC-width and clique-width 3, it follows that a k-tree-power graph has NLC-width and clique-width at most 6 · k^2. We prove in this paper that k-tree-power graphs have NLC-width at most k + 2, and clique-width at most k + 2 + max([l/2] − 1, 0). We also show that k-leaf-power graphs have NLC-width at most k and clique-width at most k + max([l/2] − 2, 0).

Corneil and Röhic have shown in [15], that every graph G of tree-width l has clique-width at most 3 · 2^{l−1}. This result implies that G^k has NLC-width and clique-width at most 3 · 2^l · k^{2^{l−1}}. We improve this bound, and show that the k-power graph of a graph of tree-width l has NLC-width at most (k + 1)^{l+1} − 1 and clique-width at most 2 · (k + 1)^{l+1} − 2. We also show that the k-power graph of a graph of path-width l has linear NLC-width at most (k + 1)^{l+1} − 1, and linear clique-width at most (k + 1)^{l+1}.

2. Preliminaries

Let [k] := {1, . . . , k} be the set of all integers between 1 and k. We work with finite undirected vertex labeled graphs G = (V G, E G, lab C), where V G is a finite set of vertices labeled by some mapping lab C : V G → [k] and E G ⊆ {u, v | u, v ∈ V G, u ̸= v}, is a finite set of edges. The labeled graph consisting of a single vertex labeled by a ∈ [k] is denoted by a.

The notion of clique-width defined by Courcelle and Olariu in [2].

Definition 1 (Clique-Width, [2]). Let k be some positive integer. The class CW_k of labeled graphs is recursively defined as follows.

(1) The single vertex graph a for some a ∈ [k] is in CW_k.

(2) Let G = (V G, E G, lab C) ∈ CW_k and J = (V J, E J, lab J) ∈ CW_k be two vertex disjoint labeled graphs, then

G ⊕ J := (V ′, E ′, lab ′)

defined by V ′ := V G ∪ V J, E ′ := E G ∪ E J, and

lab ′(u) :=
lab C(u) if u ∈ V G
lab J(u) if u ∈ V J

is in CW_k.

(3) Let a, b ∈ [k] be two distinct integers and G = (V G, E G, lab C) ∈ CW_k be a labeled graph, then

(a) ρ_{a→b}(G) := (V G, E G, lab ′) defined by

lab ′(u) :=
lab C(u) if lab C(u) ̸= a
b if lab C(u) = a

is in CW_k and

(b) η_{a,b}(G) := (V ′, E ′, lab ′) defined by

E ′ := E G ∪ {u, v | u, v ∈ V G, u ̸= v, lab C(u) = a, lab C(v) = b}

is in CW_k.

The notion of NLC-width is defined by Wanke in [3].

Definition 2 (NLC-Width, [3]). Let k be some positive integer. The class NLC_k of labeled graphs is recursively defined as follows.

(1) The single vertex graph a for some a ∈ [k] is in NLC_k.

(2) Let G = (V G, E G, lab C) ∈ NLC_k and R : [k] → [k] be a function, then

φR(G) := (V G, E G, lab ′)

defined by lab ′(u) := R(lab C(u)) is in NLC_k.

1 The operations in the definition of clique-width were first considered by Courcelle, Engelfriet, and Rozenberg in [16].
2 The abbreviation NLC results from the node label controlled embedding mechanism originally defined for graph grammars.
(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 \textit{clique-width} (\textit{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 $\bullet_o, \oplus, \rho_{a-b}, \eta_{a,b}$ for integers $a, b \in [k]$ is called a \textit{clique-width} $k$-expression. An expression built with the operations $\bullet_a, \circ_b, \times_S$ for $a \in [k]$, $R : [k] \rightarrow [k]$, and $S \subseteq [k]^2$ is called an \textit{NLC-width} $k$-expression.

The graph defined by an expression $X$ is denoted by $\text{val}(X)$. A vertex labeled graph $G$ has \textit{linear clique-width} (\textit{linear NLC-width}) at most $k$, if it can be defined by a clique-width $k$-expression (an $\text{NLC-width} k$-expression, respectively) where at least one argument of every operation $\oplus$ (of every operation $\times_S$, respectively) is a single vertex graph $\bullet_a$ for some label $a \in [k]$, see also [5].

The following example shows that every clique $K_n$, $n \geq 1$, has linear clique-width 2 and linear NLC-width 1, and that every path $P_n$ has linear clique-width at most 3 and linear NLC-width at most 3.

\textbf{Example 3.} (1) Every clique $K_n = \{(v_1, \ldots, v_n), \{(v_i, v_j) \mid 1 \leq i < j \leq n\}\}$, $n \geq 2$, has linear clique-width 2, by the following recursively defined expressions $X_{K_n}$.
$$X_{K_2} := \eta_{1,2}(\bullet_1 \oplus \bullet_2)$$
$$X_{K_n} := \eta_{1,2}(\rho_{2 \rightarrow 1}(X_{K_{n-1}}) \oplus \bullet_2), \text{ if } n \geq 3.$$  

(2) Every path $P_n = \{(v_1, \ldots, v_n), \{(v_i, v_j) \mid 1 \leq i < j \leq n\}\}$ has linear clique-width at most 3, by the following recursively defined expressions $X_{P_n}$.
$$X_{P_3} := \eta_{2,3}(\eta_{1,2}(\bullet_1 \oplus \bullet_2) \oplus \bullet_3)$$
$$X_{P_n} := \eta_{2,3}(\rho_{3 \rightarrow 2}(\rho_{2 \rightarrow 1}(X_{P_{n-1}}) \oplus \bullet_3), \text{ if } n \geq 4.$$  

(3) Every clique $K_n$, $n \geq 1$, has linear NLC-width 1, by the following recursively defined expressions $X_{K_n}$.
$$X_{K_1} := \bullet_1$$
$$X_{K_n} := X_{K_{n-1}} \times ((1,1)) \bullet_1, \text{ if } n \geq 2.$$  

(4) Every path $P_n$ has linear NLC-width at most 3, by the following recursively defined expressions $X_{P_n}$.
$$X_{P_3} := (\bullet_1 \times ((1,2))) \times ((2,3)) \bullet_3$$
$$X_{P_n} := \circ((1,1),(2,1),(3,2))(X_{P_{n-1}}) \times ((2,3)) \bullet_3, \text{ if } n \geq 4.$$  

3. \textit{Clique-width and NLC-width of tree-power graphs}

Any $k$-power graph of a tree is a chordal graph, see [17]. Determining whether a given graph $G$ is the $k$-power graph of a tree can be done in linear time for every fixed integer $k$ [18]. Fig. 1 shows an example of a 2-tree-power graph.

We next give a bound on the NLC-width and clique-width of $k$-tree-power graphs for fixed $k$.

\textbf{Theorem 4.} The $k$-power graph of a tree has NLC-width at most $k + 2$ and clique-width at most $k + 2 + \max\{|\frac{k}{2}| − 1, 0\}$. 

Fig. 1. The figure shows a tree $T$ and its 2-tree-power graph $T^2$. 
Proof. Let \( T \) be a tree with root \( v \). Then, for every vertex \( v \) of \( T \) we define an NLC-width \((k+2)\)-expression \( X_v \) for the \( k \)-power graph of the subtree of \( T \) with root \( v \). The set of labels is the set of integers\(^3\) between 0 and \( k+1 \). In \( \text{val}(X_v) \) a vertex has label \( r \leq k \) if its distance to \( v \) is \( r \). That is, the label of \( v \) is 0, the labels of the children of \( v \) are 1, and so on. All vertices with a distance to \( v \) greater than \( k \) get label \( k+1 \).

Let \( v_1, \ldots, v_m \) be the children of vertex \( v \) in \( T \). Then

\[
X_v := \begin{cases} 
\bullet_0 & \text{if } m = 0 \\
Y_m \times_S \bullet_0 & \text{if } m \geq 1 
\end{cases}
\]

and

\[
Y_m := \begin{cases} 
\circ_{R}(X_{v_1}) & \text{if } m = 1 \\
Y_{m-1} \times_S \circ_{R}(X_{v_m}) & \text{if } m > 1 
\end{cases}
\]

where

\[
S := \{(a, b) \mid a, b \in \{0, \ldots, k\} \text{ and } a + b \leq k\}
\]

and

\[
R(a) := \begin{cases} 
a + 1 & \text{if } 0 \leq a < k \\
k + 1 & \text{if } a = k + 1.
\end{cases}
\]

Then \( X_v \) defines the \( k \)-power graph of tree \( T \), using at most \( k + 2 \) labels.

To define a clique-width expression \( X_{v_i} \), the edges between the vertices of two subtrees defined by expression \( Y_m \) can be inserted with additional auxiliary labels. These labels are necessary, because we cannot insert edges between equal labeled vertices. We only need to relabel the labels 1, \( \ldots, \lfloor \frac{k}{2} \rfloor \) in one of the two combined graphs. After all succeeding edge insertions between the two subtrees the auxiliary labels can be relabeled back to 1, \( \ldots, \lfloor \frac{k}{2} \rfloor \). Since label 0 can be used as an auxiliary label, we only need \( \lfloor \frac{k}{2} \rfloor \) auxiliary labels at all. The expressions \( X_{v_i} \) can be defined similarly to those given above, using NLC-width operations as follows.

\[
X_{v_i} := \begin{cases} 
\bullet_0 & \text{if } m = 0 \\
\eta_0(Y_m \oplus \bullet_0) & \text{if } m \geq 1 
\end{cases}
\]

and

\[
Y_m := \begin{cases} 
\rho_+(X_{v_1}) & \text{if } m = 1 \\
\rho_2(\eta_2(Y_{m-1} \oplus \rho_1(\rho_+(X_{v_m}))))) & \text{if } m > 1 
\end{cases}
\]

where \( \rho_+ \) denotes the sequence of all operations \( \rho_{a \to a+1} \) applied on expression \( X \) for \( a = k, \ldots, 0 \), i.e.

\[
\rho_+ := \rho_{a \to a+1}(\ldots (\rho_{a-k-1}(\rho_{a-k}(X)) \ldots)).
\]

\( \rho_1 \) applied on some expression \( X \) changes labels \( l \in \{1, \ldots, \lfloor \frac{k}{2} \rfloor \} \) onto auxiliary labels \( f_k(l) \in \{0, k+2, \ldots, k+ \lfloor \frac{k}{2} \rfloor \} \) by some bijection \( f_k \) defined as follows.

\[
\rho_1(X) := \begin{cases} 
X & \text{if } k = 1 \\
\rho_{a \to 1}(X) & \text{if } k \in \{2, 3\} \\
\rho_{a \to k+1}(\ldots (\rho_{a+1}(X)) \ldots) & \text{if } k \geq 4.
\end{cases}
\]

After the edge insertions in expression \( Y_m \), defined below, every auxiliary label \( f_k(l) \) is labeled back to its original label \( l \), by \( \rho_2 \) defined as follows.

\[
\rho_2(X) := \begin{cases} 
X & \text{if } k = 1 \\
\rho_{k \to 1}(X) & \text{if } k \in \{2, 3\} \\
\rho_{k+1}(\ldots (\rho_{k+2}(X)) \ldots) & \text{if } k \geq 4.
\end{cases}
\]

The edge insertions used in expressions \( X_{v_i} \) and \( Y_m \) are done by three meta operations \( \eta_0, \eta_1, \) and \( \eta_2 \) defined as follows. \( \eta_0(X) \) defines the sequence of all operations \( \eta_{a \to a} \), \( a \in [k] \), applied on expression \( X \).

\( \eta_1(X) \) defines the sequence of all operations \( \eta_{a \to b} \), \( a, b \in [k], a + b \leq k, a \neq b \), applied on expression \( X \).

\( \eta_2(X) \) defines the sequence of all operations \( \eta_{a \to b} \), \( a \in [k], b \in [\lfloor \frac{k}{2} \rfloor], a + b \leq k \), applied on expression \( X \).

It is important to notice that, in contrast to the union operation of NLC-width, all clique-width edge insertion operations applied on the disjoint union of two defined subtrees, also insert edges into both involved subtrees. By the definition of relabeling operation \( \rho_+ \) it is easy to observe that between two vertices of one subtree, no new edge will be defined by \( \eta_0, \eta_1, \) or \( \eta_2 \). \( \square \)

\(^3\)In the definition of the NLC-width and clique-width, the vertex labels are always positive integers. It is easy to see that the labels we use for the vertices can be changed to conform the definition without increasing the number of labels used.
By the results of [18] and Theorem 4, it follows that for every $k$-tree power graph $G$, an NLC-width $(k + 2)$-expression and a clique-width $(k + 2 + \max\{\lfloor \frac{k}{2} \rfloor - 1, 0\})$-expression can be found in linear time for every fixed integer $k$.

4. Clique-width and NLC-width of leaf-power graphs

The notion of a leaf-power graph was introduced in [19] motivated by the reconstruction of phylogenetic trees as a certain case of tree-power graphs. The $k$-leaf-power graph $T^k$ of a tree $T$ is a graph whose vertices are the leaves of $T$. Two vertices in $T^k$ are adjacent if and only if, there is a path between them in $T$ of length at most $k$. Fig. 2 shows an example of a 3-leaf-power graph.

Every $k$-leaf-power graph is strongly chordal, see [20]. A graph is a 2-leaf-power graph if and only if, it is the disjoint union of cliques, thus they have NLC-width 1 and clique-width at most 2. A graph is a 3-leaf-power graph if and only if, it is obtained from a tree $T$ by substituting the vertices of $T$ by cliques, see [20]. That is, the NLC-width and clique-width of 3-leaf-power graphs is at most 3. 3-leaf-power and 4-leaf-power graphs can be recognized in linear time, see [20,21]. For the recognition of $k$-leaf-power graphs with $k \geq 5$ no polynomial time algorithms are known.

In [22,23] a forbidden subgraph characterization of prime 4-leaf-power graphs is given. All strictly chordal graphs, i.e. chordal graphs whose clique hypergraph is a strict hypertree, are $k$-leaf-power graphs for every $k \geq 4$. The corresponding trees can be found in linear time, see [24].

In [25] it is shown that the set of all leaf power graphs (i.e. the class of graphs which are $k$-leaf power graphs for some $k$), has unbounded clique-width.

We next give a bound on the NLC-width and clique-width of $k$-leaf-power graphs for fixed $k \geq 2$.

Theorem 5. The $k$-leaf-power graph of a tree has NLC-width at most $k$ and clique-width at most $k + \max\{\lfloor \frac{k}{2} \rfloor - 2, 0\}$.

Proof. Let $T$ be a tree with root $u$. Then for every inner vertex $v$ of $T$, we define an NLC-width $k$-expression $X_v$. This expression will define the $k$-leaf-power graph of the subtree of $T$ with root $v$. The set of labels is the set of integers between 1 and $k$. In val$(X_v)$ a vertex $u$ has label $r < k$, if its distance to $v$ is $r$. All vertices with a distance to $v$ greater than or equal to $k$ have label $k$.

Let $v_1, \ldots, v_m$ be the non-leaf children of inner vertex $v$ in $T$ and let $l$ be the number of children of $v$ that are leaves. A basic observation for leaf-power graphs is that the leaves of $v$ induce a complete graph of $T^k$, for $k \geq 2$. Let $C_l$ be an NLC-width 1-expression for a complete graph on $l$ vertices. Then

$$X_v := \begin{cases} C_l & \text{if } m = 0 \text{ and } l > 0 \\ Y_m \times_l C_l & \text{if } m \geq 1 \text{ and } l > 0 \\ Y_m & \text{if } m \geq 1 \text{ and } l = 0 \end{cases}$$

and

$$Y_m := \begin{cases} \circ_g(X_{v_1}) & \text{if } m = 1 \\ Y_{m-1} \times_l \circ_g(X_{v_m}) & \text{if } m > 1 \end{cases}$$

where

$$S := \{(a, b) \mid a, b \in \{1, \ldots, k-1\} \text{ and } a + b \leq k\}$$

and

$$R(a) := \begin{cases} a + 1 & \text{if } 1 \leq a \leq k - 1 \\ k & \text{if } a = k. \end{cases}$$

Then $X_v$ defines the $k$-leaf-power graph of $T$ using at most $k$ labels.

To define a clique-width expression $X_v$, $C_l$ is a clique-width 2-expression that defines a clique with $l$ vertices, all labeled by label 1. The edges between the leaves of two subtrees can be inserted similarly as in the proof of Theorem 4 with additional auxiliary labels. Now we only need to relabel the labels $2, \ldots, \lfloor \frac{k}{2} \rfloor$ in one of the two combined graphs. Since label 1 can be
used as auxiliary label, we only need \( \left\lfloor \frac{k}{2} \right\rfloor - 2 \) auxiliary labels at all. The expressions \( X_e \) can be defined similarly to those given above, using NLC-width operations as follows.

\[
X_e := \begin{cases} 
  C_l & \text{if } m = 0 \text{ and } l > 0 \\
  \eta_1(Y_m \oplus C_l) & \text{if } m \geq 1 \text{ and } l > 0 \\
  Y_m & \text{if } m \geq 1 \text{ and } l = 0
\end{cases}
\]

and

\[
Y_m := \begin{cases} 
  \rho_+(X_{\eta_1}) & \text{if } m = 1 \\
  \rho_2(\eta_3(\eta_2(Y_{m-1} \oplus \rho_1(\rho_+(X_{\eta_2}))))) & \text{if } m > 1
\end{cases}
\]

where \( \rho_+(X) \) denotes the sequence of all operations \( \rho_{a \rightarrow a+1} \), applied on expression \( X \) for \( a = k, \ldots, 0 \), i.e.

\[\rho_+(X) := \rho_{0 \rightarrow 1}(\ldots(\rho_{k-1 \rightarrow k}(\rho_{k \rightarrow k+1}(X))) \ldots).\]

\( \rho_1 \) applied on some expression \( X \) changes labels \( l \in \{2, \ldots, \left\lfloor \frac{k}{2} \right\rfloor \} \) onto auxiliary labels \( f_2(l) \in \{1, k + 1, \ldots, k + \left\lfloor \frac{k}{2} \right\rfloor - 2 \} \), by some bijection \( f_2 \) defined as follows

\[
\rho_1(X) := \begin{cases} 
  X & \text{if } k \in \{2, 3\} \\
  \rho_{2 \rightarrow 1}(X) & \text{if } k \in \{4, 5\} \\
  \rho_{\left\lfloor \frac{k}{2} \right\rfloor \rightarrow k+1(\left\lfloor \frac{k}{2} \right\rfloor - 2) \ldots (\rho_{3 \rightarrow k+1}(\rho_{2 \rightarrow 1}(X))) \ldots) & \text{if } k \geq 6.
\end{cases}
\]

After the edge insertions in expression \( Y_m \), defined below, every auxiliary label \( f_2(l) \) is labeled back to its original label \( l \) by \( \rho_2 \), defined as follows

\[
\rho_2(X) := \begin{cases} 
  X & \text{if } k \in \{2, 3\} \\
  \rho_{1 \rightarrow 2}(X) & \text{if } k \in \{4, 5\} \\
  \rho_{k+1(\left\lfloor \frac{k}{2} \right\rfloor - 2) \ldots (\rho_{k+1(\left\lfloor \frac{k}{2} \right\rfloor - 2)}(\rho_{1 \rightarrow 2}(X)))} & \text{if } k \geq 6.
\end{cases}
\]

The edge insertions used in expressions \( X_e \) and \( Y_m \) are done by three meta operations \( \eta_1, \eta_2, \) and \( \eta_3 \) defined as follows.

\( \eta_1(X) \) defines the sequence of all operations \( \eta_{a, 1} \), \( a \in \{2, \ldots, k-1 \} \), applied on expression \( X \).

\( \eta_2(X) \) defines the sequence of all operations \( \eta_{a, b} \), \( a, b \in \{2, \ldots, k-2 \}, a + b \leq k \), \( a \neq b \), applied on expression \( X \).

\( \eta_3(X) \) defines the sequence of all operations \( \eta_{a, f(b)} \), \( a \in \{2, \ldots, k-2 \}, b \in \{2, \ldots, \left\lfloor \frac{k}{2} \right\rfloor \}, a + b \leq k \), applied on expression \( X \).

Again, it is important to notice that, all clique-width edge insertion operations applied on the disjoint union of two defined subtrees, also insert edges into both involved subtrees. By the definition of relabeling operation \( \rho_+ \), it is easy to observe that between two vertices of one subtree no new edge will be defined by \( \eta_1, \eta_2, \text{ or } \eta_3. \)

In [13] the set of clique-width operations was extended by operations of the form \( \eta_{a, a} \) in order to define clique-width expressions that allow to insert edges between equal labeled vertices. The set of \( k \)-labeled graphs which can be defined by these extended clique-width operations using \( k \) labels, is denoted by \( \text{CW}_k^* \). A simple observation shows that \( \text{CW}_k \subseteq \text{CW}_k^* \subseteq \text{NLC}_k \subseteq \text{CW}_{2k} \). Even \( \text{CW}_1 \), which consists of the disjoint union of cliques, is less powerful than \( \text{NLC}_1 \), which is equal to the set of all co-graphs [3]. It is easy to see that every \( k \)-tree-power graph is in \( \text{CW}_{k+2} \), and every \( k \)-leaf-power graph is in \( \text{CW}_k \).

In [24] strictly chordal graphs are defined as chordal graphs whose clique hypergraph is a strict hypertree. Further in [24] it has been shown that strictly chordal graphs are 4-leaf-power graphs, which immediately implies the following NLC-width and clique-width bound.

**Corollary 6.** Every strictly chordal graph has NLC-width at most 4 and clique-width at most 4 and a corresponding 4-expression can be found in linear time.

Since strictly chordal graphs are exactly (dart, gem)-free chordal graphs [26], they are distance-hereditary. As shown by Golumbic and Rotics in [27], the clique-width of distance-hereditary graphs is at most 3, which even improves the bound of Corollary 6.

### 5. Clique-width and NLC-width for \( k \)-power graphs of graphs of bounded tree-width

Now we consider the NLC-width and clique-width of power graphs of graphs of bounded tree-width.

**Theorem 7.** The \( k \)-power graph of a graph of tree-width \( l \) has NLC-width at most \( (k + 1)^{l+1} - 1 \).

To prove Theorem 7, we need the following notations. The tree-width of a graph is defined by Robertson and Seymour in [6]. A graph of tree-width \( l \) is also known as partial \( l \)-tree, see Rose [28]. A partial \( l \)-tree is a subgraph of an \( l \)-tree, which is defined as follows:

1. A complete graph with \( l \) vertices is an \( l \)-tree.
(2) If $G$ is an $l$-tree, then the result obtained by adding a new vertex with edges to $l$ vertices of a complete subgraph of $G$ is an $l$-tree.

Let $G$ be a graph with $n$ vertices. Let $o = (v_1, \ldots, v_n)$ be an order of the $n$ vertices of $G$, i.e. every vertex of $G$ appears in sequence $o$ exactly once. Let $N^+(G, o, i)$ and $N^-(G, o, i)$ for $i = 1, \ldots, n$ be the set of neighbors $v_j$ of vertex $v_i$ with $i < j$ and $j < i$, respectively. That is,

$$N^+(G, o, i) := \{ v_j \mid \{v_i, v_j\} \in E_G \land i < j \}$$

and

$$N^-(G, o, i) := \{ v_j \mid \{v_i, v_j\} \in E_G \land j < i \}.$$ 

A vertex order $o = (v_1, \ldots, v_n)$ for graph $G$ is called a perfect elimination order (PEO) if the vertices of $N^+(G, o, i)$ for $i = 1, \ldots, n$ induce a complete subgraph of $G$.

The main idea of the following proof is as follows. Given some graph $G$ of bounded tree-width, we use a PEO of $G$ and recursively define for $i = 1, \ldots, n$ an NLC-width expression $X_i$ for the $k$-power graph of the subgraph of $G$ induced by the vertices $v_j, 1 \leq j \leq i$. The main part of the proof shows that the resulting expression $X_n$ defines the $k$-power graph of $G$.

**Proof of Theorem 7.** Let $G$ be a partial $l$-tree that is a subgraph of an $l$-tree $\hat{G}$, such that $G$ has the same vertex set as $\hat{G}$. By the recursive definition of $l$-trees, we know that each $l$-tree $\hat{G}$ has a PEO. Let $o := (v_1, \ldots, v_n)$ be a PEO of $\hat{G}$. Then, the vertices of the sets $N^+(\hat{G}, o, 1), \ldots, N^+(\hat{G}, o, n - k)$ induce $k$ vertex complete subgraphs of $\hat{G}$. The vertices of each $N^+(\hat{G}, o, i)$ for $n - k < i \leq n$ induce an $n - i$ vertex complete subgraph of $\hat{G}$.

Each $l$-tree $\hat{G}$ is $(l + 1)$-colorable, because we can assign to $v_1$ any color not used by the (at most $l$) vertices of $N^+(\hat{G}, o, i)$ for $i = 1, \ldots, n$. Let $col : V_{\hat{G}} \rightarrow [l + 1]$ be an $(l + 1)$-coloring of $l$-tree $\hat{G}$, that is, $col(v_i) \neq col(v_j)$ for all edges $\{v_i, v_j\} \in E_{\hat{G}}$. The mapping $col$ is also an $(l + 1)$-coloring of partial $l$-tree $G$.

We define $c_{\hat{G}}, 1 \leq i \leq n$, to be the set of all colors of the vertices from $N^+(\hat{G}, o, i)$. The tree structure of $l$-tree $\hat{G}$ with respect to PEO $o$ can be characterized by the following tree $T := (V_T, E_T)$ defined by

$$V_T := V_{\hat{G}}$$

$$E_T := \{ \{v_i, v_j\} \in E_{\hat{G}} \mid i < j \land \forall \tilde{v}_j', i < j' < j : \{v_i, \tilde{v}_j'\} \notin E_{\hat{G}} \}.$$ 

Graph $T$ is a tree, because for every vertex $v_i, 1 < n$, there is exactly one edge $\{v_i, v_j\} \in E_{\hat{G}}$ with $i < j$, such that $\forall v_j', i < j' < j : \{v_i, v_j'\} \notin E_{\hat{G}}$.

Let $v_0$ be the root of $T$. We recursively define for $i = 1, \ldots, n$ an NLC-width expression $X_i$ for the $k$-power graph of the subgraph of $G$ induced by the vertices $v_j, 1 \leq j \leq i$. The vertices will be labeled by sets$^4$ of pairs $(c, d)$, where $c \in [l + 1]$ is a vertex color and $d \in [k]$ is a distance. In each vertex label, for every color $c \in [l + 1]$, there is at most one pair $(c, d)$ for some $d \in [k]$. This implies that we have at most $(l + 1)^{k+1}$ different vertex labels. A color $c$ is either not used, or used together with one of $k$ distances. Since $c$ is a vertex color, we will see later that there is one label which is not used at all, this is the label $\{(1, 1), \ldots, (l + 1, 1)\}$.

For a label $C \subseteq [l + 1] \times [k]$ and a positive integer $r$ let

$$C^{+r} := \{ (c, d + r) \mid (c, d) \in C \text{ and } d + r \leq k \}.$$ 

For two labels $C_1, C_2 \subseteq [l + 1] \times [k]$ let

$$C_1 \cup C_2 \text{ and } C_1 \cap C_2$$ 

be the set of all pairs $(c, d) \in C_1 \cup C_2$, such that there is no pair $(c, d') \in C_1 \cup C_2$ for some $d' < d$.

We first define an NLC-width expression $X_n$ for a subgraph of $G$ with vertex set $V_{\hat{G}}$. After that, we will show how the missing edges are added by inserting additional pairs in the relations $S$ used by the edge insertion operations $\times_S$ of $X_n$.

Expression $X_i$ for $i = 1, \ldots, n$ is defined as follows.

1. If $N^-(T, o, i) = \emptyset$, i.e. if vertex $v_i$ is a leaf in tree $T$, then we define

   $$X_i := \bullet_{\{(c,1)\mid c \in c_{\hat{G}}\}}.$$ 

2. If $N^-(T, o, i) = \{v_j, \ldots, v_{jm}\}$, i.e. if $v_j, \ldots, v_{jm}$ are the children of vertex $v_i$ in tree $T$, then we define

   $X_{i,1} := X_1$

   $X_{i,2} := X_{i,1} \times_S X_2$

   $\vdots$

   $X_{i,m} := X_{i,m-1} \times_S X_{jm}$

   $X_i := o_R (\bullet_{d_1} \times_S \bullet_{d_m} X_{i,m})$.

$^4$ In the definition of the NLC-width and clique-width, the vertex labels are always positive integers. It is easy to see that the labels we use for the vertices can be changed to conform the definition without increasing the number of labels used.
where all $S_1, \ldots, S_{m-1}$ are (so far) empty and

$$S_m := \{(D_i, C) \mid C \subseteq [l+1] \times [k] \land (\text{col}(v_i), 1) \in C\}.$$ 

Thus by relation $S_m$ vertex $v_i$ will be connected with all vertices from $\text{val}(X_{i,m})$ whose label set contains a pair $(\text{col}(v_i), 1)$.

The label $D_i$ is defined as follows. Consider all vertices of $\text{val}(X_{i,m})$ whose labels have a pair with color $\text{col}(v_i)$. Every one of these labels has exactly one pair with color $\text{col}(v_i)$. Let $C_1, \ldots, C_r$ be the labels of these vertices and let $(\text{col}(v_i), d_1), \ldots, (\text{col}(v_i), d_r)$ be the corresponding pairs with color $\text{col}(v_i)$. Then $D_i$ is the set

$$D_i := \{(c, 1) \mid c \in C_{v_i} \} \cup \min_{1 \leq j \leq r} \left( C_j - \{ (\text{col}(v_i), d_j) \} \right)^{+ e_j}.$$ 

Vertex $v_i$ is labeled by all pairs $(c, 1)$ for all $c$ which are a color of a neighbor of vertex $v_i$.

Further vertex $v_i$ has to be connected with all vertices, to which there is a path of length at most $k$. For $k > 1$ this can be done via vertices from $\text{val}(X_{i,m})$. Therefore we insert the label pairs of the neighbors of $v_i$ in $\text{val}(X_{i,m})$ with an increased distance value $d_j$ into set $D_i$.

Finally, operation $\circ R$ relabels every label $C_i$, $1 \leq s \leq r$, as defined above into

$$R(C_i) := C_i - \{ (\text{col}(v_i), d_s) \} \cup \min_{1 \leq j \leq r} \left( C_j - \{ (\text{col}(v_i), d_j) \} \right)^{+(d_j + d_s)}.$$ 

The relabeling operation $R$ consists of three parts, which work as follows.

For every label $C_i \subseteq [l+1] \times [k]$ we first remove label pair $(\text{col}(v_i), d_s)$ since the edges between vertex $v_i$ and the vertices of $\text{val}(X_{i,m})$ have been inserted.

Next we have to connect every vertex $v_i$ from $\text{val}(X_{i,m})$ which contains a pair $(\text{col}(v_i), d_s)$ with every neighbor of $v_i$ if the distance $1 + d_s$ is less than or equal $k$. Therefore we insert for every $c$ which is a color of a neighbor of $v_i$ the pair $(c, 1 + d_s)$, if $1 + d_s \leq k$, into set $C_i$.

Furthermore, we have to connect every vertex $v_i$ from $\text{val}(X_{i,m})$ which contains a pair $(\text{col}(v_i), d_j)$ with the neighbors of every further vertex $v_j$ from $\text{val}(X_{i,m})$ which contains a pair $(\text{col}(v_j), d_j)$ if the additional distance via vertex $v_j$ is at most $k$. Therefore, we insert the label pairs of vertex $v_j$ with a distance value increased by $d_j + d_j$ into set $C_i$.

Note that operation $\circ R$ does not change label $D_i$ of $v_i$, because it has no pair with color $\text{col}(v_i)$.

A graph defined by expression $X_n$ defines the partial $l$-tree $G$, and thus defines a subgraph of $G^l$. The missing edges will be inserted by a modification of $S_1, \ldots, S_m$ shown below.

By the definition of our tree-structure $T$ the following remark is easy to prove.

**Remark 8.** Let $X_i = \circ_R (\bullet_{n_i} \times_s X_{i,m})$ some subexpression of $X_n$, such that $X_{i,m}$ is the composition of expressions $X_{i_1}, \ldots, X_{i_m}$, $m \geq 2$. Let $u \in \text{val}(X_{i,m})$ and $w \in \text{val}(X_{i,m})$, $i' \neq i''$, be two vertices. Then there is no edge $(u, w)$ in graph $G$, i.e. if there is some path from $u$ to $w$ in $G$, then it uses vertex $v_i$.

The following lemma characterizes the labelings of vertices in graph $\text{val}(X_i)$, $1 \leq i \leq n$.

**Lemma 9.** Let $u$ be a vertex of graph $\text{val}(X_i)$, $1 \leq i \leq n$, and $C = \text{lab}_{\text{val}(X_i)}(u)$ be the label of $u$ in graph $\text{val}(X_i)$. Then $C$ contains pair $(c, k')$, if and only if there is a shortest path $(u, \ldots, v, u)$ of length $k' \leq k$ in $G$ such that $u, \ldots, v \in \text{val}(X_j)$, $w \in \text{val}(X_i)$, and $\text{col}(w) = c$.

**Proof.** We prove the lemma by an induction on the composition of expression $X_n$.

**Basis.** For every expression $X_i$ which defines a leaf $v_i$ of tree structure $T$ our assumption holds true by the definition of its label set $\{(c, 1) \mid c \in C_{v_i}\}$. At $v_i$ starts a path of length 1 to some vertex $w \in \text{val}(X_i)$ if and only if, $(\text{col}(w), 1) \in \text{lab}_{\text{val}(X_i)}(v_i)$. (See Fig. 3.)
Induction: We assume that for all expressions $X_j, j < i$ the lemma is shown and show it for expression $X_i = o_R(v_{j_i} \times_{S_m} X_{i,m}).$

for some relabeling function $R$ and some label $D_i$ of vertex $v_i$ and edge insertion relation $S_m.$ Expression $X_{i,m}$ consists of $m \geq 1$ subexpressions $X_{j,k}, 1 \leq k \leq m$ which all define the subgraphs corresponding to the subtrees rooted at $v_{j_1}, \ldots, v_{j_m}$ with $j_1 < \ldots < j_m \leq i \leq i_m$ (See Fig. 4).

By our induction hypothesis, the label set of every $u \in V_{val(X_{i,k})}$ contains a pair $(c, k'),$ if and only if, there is a shortest path $(u, \ldots, v, w)$ of length $k' \leq k$ in $G,$ such that $u, \ldots, v \in V_{val(X_{i,k})}, w \in G - V_{val(X_{i,k})},$ and col$(w) = c.$

We next show that the latter property holds for the label set of every $u \in V_{val(X_i)} = \{v_i\} \cup V_{val(X_{i,m})},$ for $u, \ldots, v \in V_{val(X_i)}$ and $w \in G - V_{val(X_i)}.$

In order to show our assumption for the label set $D_i$ of the new inserted vertex $v_i,$ we consider the following two possible types of shortest paths $(v_i, \ldots, v, w), v_i, \ldots, v \in V_{val(X_i)}, w \in G - V_{val(X_i)}$.

1. First we consider shortest paths $(v_i, \ldots, v, w),$ that do not use vertices from $val(X_{i,k}).$ Obviously, every such path consists of exactly one edge $\{v_i, v\}$ and by the first part of the definition of $D_i$ we know that the label set of $v_i$ contains a pair $(\text{col}(w), 1),$ if and only if, at $v_i$ starts a path of length 1 to vertex $w \in G - V_{val(X_i)}.$

2. Next we consider shortest paths $(v_i, v_r, \ldots, v, w),$ that use at least one vertex $v_r \in V_{val(X_{i,k})}$ for some $1 \leq i \leq m.$ By our induction hypothesis the label set of $v_r$ contains a pair $(\text{col}(w), d_r),$ if and only if, there exists a shortest path $(v_r, \ldots, v, w),$ $v_r, \ldots, v \in V_{val(X_{i,m})}, w \in G - V_{val(X_{i,m})}$ of length $d_r.$ Since $v_i \in G - V_{val(X_{i,k})}$ this also holds true for $w = u_i.$ By the second part of the definition of $D_i,$ we know that for every $v_r \in V_{val(X_{i,k})}$ which contains a pair $(\text{col}(v_i), d_i),$ all the label pairs of $v_r,$ except pair $(\text{col}(v_i), d_i),$ are also inserted into the label set of $v_i$ with a distance value increased by $d_i.$

By the minimum operation within the definition of $D_i$ and since a shortest path $(v_i, \ldots, w), w \in G - V_{val(X_{i,k})},$ cannot use vertices from two different subgraphs $val(X_{i,k})$ by Remark 8, the label set of $v_i$ contains a pair $(\text{col}(w), k'),$ if and only if, at $v_i$ starts a shortest path $(v_i, \ldots, v, w)$ of length $k' \geq 2, k' \leq k,$ to vertex $w \in G - V_{val(X_i)}.$

Next, we want to show our assumption for the label sets of vertices $V_{val(X_{i,k})}.$ Therefore we consider the following three possible types of shortest paths $(u, \ldots, v, w), u, \ldots, v \in V_{val(X_{i,k})}, w \in G - V_{val(X_i)}$.

1. For paths $(u, \ldots, w)$ that do not use vertex $v_i$ our assumption holds true by our induction hypothesis.

2. Next, we consider paths $(u, \ldots, v, w), u \in V_{val(X_{i,k})},$ which use vertex $v_r,$ but that do not use vertices from $val(X_{i,k}),$ $i_1 \neq i_2.$ By induction hypothesis, the label set of every such $u$ contains a pair $(\text{col}(v_i), d_i),$ if and only if, at $u$ starts a shortest path $(u, \ldots, v_i)$ of length $d_i \leq k.$ The second part of relabeling operation $R$ inserts for every neighbor $w \in G - V_{val(X_i)}$ of $v_i$ a pair $(\text{col}(w), d_i + 1)$ into the label set of $u.$

3. Next, we consider paths $(u, \ldots, v_r, \ldots, w), u \in V_{val(X_{i,k})}, w \in G - V_{val(X_i)},$ that also uses at least one vertex $v_r$ from some $val(X_{i,k}), i \neq i'.$

By our induction hypothesis, we know the following results. The label set of $v_r$ contains a pair $(\text{col}(u), d_r),$ if and only if, there exists a shortest path $(v_r, \ldots, v, w)$ of length $d_r \leq k,$ for $v_r, v_r, \ldots, v \in V_{val(X_{i,m})}, w \in G - V_{val(X_{i,m})}.$ Since $v_r \in V_{val(X_{i,k})},$ this also holds true for $w = v_i.$ i.e. the label set of $v_r$ contains a pair $(\text{col}(v_i), d_i),$ if and only if, there exists a shortest path $(v_r, \ldots, v, v_i)$ of length $d_i \leq k$ for $v_r, \ldots, v \in V_{val(X_{i,k})}.$ Further, the label set of every $u \in V_{val(X_{i,k})}$ contains a pair $(\text{col}(v_i), d_i),$ if and only if, at $u$ starts a shortest path $(u, \ldots, v_i)$ of length $d_i \leq k$ for $u, \ldots, v_i \in V_{val(X_{i,k})}.$ By the third part of the relabeling operation $R,$ we know that for every $u \in V_{val(X_{i,k})}$ whose label set contains a pair with col$(v_i),$ for every vertex $v_r \in V_{val(X_{i,k})}$ which contains a pair with col$(v_i),$ all label pairs of $v_r,$ except the pair which contains col$(v_i),$ are also inserted into the label set of $u$ with a distance value increased by $d_i.$

By our induction hypothesis, the label set of every vertex $u \in V_{val(X_{i,k})}$ contains for every shortest path $(u, \ldots, v, v_i), u, \ldots, v \in V_{val(X_{i,k})}$ of length $d_i \leq k$ a pair $(\text{col}(v_i), d_i).$ Since in expression $X_i$ vertex $v_i$ belongs to set $V_{val(X_i)},$ and no longer to set $G - V_{val(X_{i,k})}$ these label pairs have to be removed from the label sets of the corresponding vertexes $u,$ which is done by the first part of our relabeling operation, simultaneously with the above two mentioned modifications.

By the minimum operation within the definition of $R,$ and since a shortest path $(u, \ldots, v, w), u, \ldots, v \in V_{val(X_i)}, w \in G - V_{val(X_i)},$ cannot use vertices from more than two subgraphs $val(X_{i,k})$ by Remark 8, our assumption holds for the label set of every vertex $u \in V_{val(X_{i,k})}.$ □

In order to complete the proof of Theorem 7, we next show the following two claims.
Let $p = (v_1, \ldots, v_m)$ be a path in the partial l-tree $G.$ Path $p$ is called a high-end-path, if $i_j < i_m$ for $j = 1, \ldots, m - 1,$ i.e. the end vertex of $p$ has the highest index with respect to PEO $o.$
Claim 10. If $u$ and $u'$ are two equal labeled vertices of $\text{val}(X_i)$, $1 \leq i \leq n$, such that $v_j$ is the parent node of $v_i$ in $T$, then there is a shortest high-end-path of length $k' \leq k$ in $G$ from $u$ to vertex $v_j$, if and only if, there is a shortest high-end-path from $u'$ to $v_j$ of length $k'$ in $G$.

Proof of Claim 10. Since for every expression $X_i$ which defines a leaf of tree structure $T$ Claim 10 holds true, we just consider subexpressions of the form $X_i = \text{o}_G(\bullet \cdot \times X_{i,m})$.

Let $u$ and $u'$ be two equal labeled vertices in graph $\text{val}(X_i)$, $1 \leq i \leq n$. Assume that there is a shortest high-end-path of length $k'$ for $k' \leq k$ in $G$ from $u$ to vertex $v_j$, the parent node of $v_i$ in $T$, then by Lemma 9 the label set of $u$ contains a pair $(\text{col}(v_j), k')$ and by our assumption, the label set of $u'$ contains a pair $(\text{col}(v_j), k')$ which implies by Lemma 9 that there also is a shortest high-end-path of length $k'$ in $G$ from $u'$ to vertex $v_j$. □

Next we generalize Claim 10 as follows.

Claim 11. If $u$ and $u'$ are two equal labeled vertices of $\text{val}(X_i)$, $1 < i < n$, then there is a shortest high-end-path from $u$ to some vertex $v_j$, $j > i$, of length $k' \leq k$ in $G$ if and only if, there is a shortest high-end-path from $u'$ to $v_j$ of length $k'$ in $G$.

Proof of Claim 11. Let $u$ and $u'$ be two equal labeled vertices of $\text{val}(X_i)$, for some $1 \leq i \leq n$.

It is obvious that there cannot be a high-end-path of length greater than 0 to a vertex of $G$, which corresponds to a leaf in tree structure $T$. That is, for every high-end-path $p = (v_{i_1}, \ldots, v_{i_m})$ of $G$ vertex $v_{i_m}$ is the new inserted vertex in some sub-expression $X_i = \text{o}_G(\bullet \cdot \times X_{i,m})$ of $X_n$.

That is, for every vertex $v_j$, $j > i$, we can find some expression $X_{j'}$, $j' \geq i$, such that $v_j$ is the parent node in tree $T$ of $v_{j'}$ which is the new inserted vertex in expression $X_{j'} = \text{o}_G(\bullet \cdot \times X_{j',m})$.

In expression $X_{j'}$ vertices $u$ and $u'$ are still equal labeled, and thus by Claim 10 there is a shortest high-end-path from $u$ to vertex $v_j$, $j > i$, of length $k' \leq k$ in $G$ if and only if, there is a shortest high-end-path from $u'$ to $v_j$ of length $k'$ in $G$. Hence Claim 11 is proved. □

Next we proceed the proof of Theorem 7.

Now we extend the sets $S$ of all edge insertion operations $\times_S$ of $X_n$ as follows. For every subexpression $Y \times_S Z$ of $X_n$ such that graph $\text{val}(Y)$ defined by $Y$ has a vertex $u$, graph $\text{val}(Z)$ defined by $Z$ has a vertex $w$, and there is a shortest path $5$ $p$ of length $k' \leq k$ between $u$ and $w$ in $G$, we add the label pair $(\text{lab}_{\text{val}}(Y)(u), \text{lab}_{\text{val}}(Z)(w))$ to $S$. The new expression is denoted by $X_{n'}$.

The graph $\text{val}(X_{n'})$, defined by the new expression $X_{n'}$, certainly has all edges of the $k$-power graph $G^k$. The following observation shows that it has only edges of $G^k$. Let $v_j$ be the vertex of the path $p$ with the largest index $j$ with respect to the perfect elimination order $\sigma$.

We consider the following two cases for operation $\times_S$.

(1) Operation $\times_S$ combines two defined subexpressions $\text{val}(X_{i_j})$ and $\text{val}(X_{i_p})$, such that $u$ is of graph $\text{val}(X_{i_j})$ and $w$ is of graph $\text{val}(X_{i_p})$. By the definition of our tree structure $T$ vertex $v_j$ is neither of $\text{val}(X_{i_j})$ nor of $\text{val}(X_{i_p})$ and $k' \geq 2$.

Then, the first part of $p$ from $u$ to $v_j$ is a shortest high-end-path of length $k_1 \leq k'$, and the last part of $p$ from $w$ back to $v_j$ is a shortest high-end-path of length $k_2 \leq k'$, such that $k_1 + k_2 = k'$. By Claim 11, every vertex of $\text{val}(Y)$ labeled by $\text{lab}_{\text{val}}(Y)(u)$ has a shortest high-end-path to $v_j$ of length $k_1$ in $G$, and every vertex of $\text{val}(Z)$ labeled by $\text{lab}_{\text{val}}(Z)(w)$ has a shortest high-end-path to $v_j$ of length $k_2$ in $G$. Thus, all vertices of $\text{val}(Y)$ labeled by $\text{lab}_{\text{val}}(Y)(u)$ can be connected with all vertices of $\text{val}(Z)$ labeled by $\text{lab}_{\text{val}}(Z)(w)$, because there are paths between the length $k' \leq k$ in $G$.

---

5 Obviously we obtain the $k$-power graph of a graph $G$ by adding edges between all vertices $u$ and $w$, if there is a shortest path of length at most $k$ between $u$ and $w$ in $G$. 

Fig. 4. Graph considered in the induction in the proof Lemma 9.
Fig. 5. A 2-tree $\hat{G}$, the tree $T$, a partial 2-tree $G$, and its 2-power graph $G^2$. The numbering $1, \ldots, 9$ of the vertices represents the perfect elimination order $o$, the letters $a$, $b$, $c$ represent a 3-coloring. The graphs below are the graphs defined by the NLC-width 8-expressions $X_1$ to $X_9$. The sets of pairs at the vertices of $\text{val}(X_1)$ to $\text{val}(X_9)$ are their labels. The NLC-width expressions $X_1$ to $X_9$ are shown in Table 1.

(2) Operation $\times_z$ combines one defined subexpressions $X_j$, and a new inserted vertex $\bullet_{D_i}$. Then, $v_j$ is either the new inserted vertex which implies by Claim 11 the correctness of our edge insertion.

Or $v_j$ is neither the new inserted nor of graph $\text{val}(X_j)$ and thus not of graph $\text{val}(Y \times_z Z)$. In this case, the correctness of our edge insertion follows by case (1). \hfill $\square$

Fig. 5 shows an example of the construction of an NLC-width expression for a partial 2-tree $G$. Table 1 shows the NLC-width expressions $X_1$ to $X_9$ defined by the proof of Theorem 7 for the graph $G$. Since every graph of NLC-width $k$ has clique-width at most $2k$, see [4], we get the following corollary.
Corollary 12. The k-power graph of a graph of tree-width \( l \), has clique-width at most \( 2 \cdot (k + 1)^{l+1} - 2 \).

Next we consider k-power graphs of a graphs of bounded path-width.

**Theorem 13.** The k-power graph of a graph of path-width \( l \) has linear NLC-width at most \( (k + 1)^{l+1} - 1 \).

To prove Theorem 13, we need the following notations. The path-width of a graph is defined by Robertson and Seymour in [29]. A graph of path-width \( l \) is also known as partial \( l \)-path, see [30]. A partial \( l \)-path is a subgraph of an \( l \)-path which is defined as follows:

For the definition of an \( l \)-path \( G \), we denote some vertices of \( G \) by link vertices.

(1) The complete graph with \( l \) vertices that are all link vertices, is an \( l \)-path.

(2) If \( G \) is an \( l \)-path with \( l \) link vertices \( U = \{u_1, \ldots, u_l\} \), then the graph obtained by inserting a new vertex \( u \not\in V_G \) and \( l \) edges between \( u \) and the \( l \) link vertices in \( U \) of \( G \) (which will always induce a complete subgraph of \( G \) is an \( l \)-path. The set of new link vertices of the resulting \( l \)-path is a subset of \( l \) vertices of \( U \cup \{u\} \).

**Proof of Theorem 13.** Let \( G \) be a partial \( l \)-path that is a subgraph of an \( l \)-path \( \hat{G} \). As mentioned in the proof of Theorem 7 we know that \( \hat{G} \), and thus \( G \) has a \( (l + 1) \)-coloring and a PEO \( o \). In the case of a partial \( l \)-path \( G \), tree structure \( T \) used in the proof of Theorem 7 is now a caterpillar with hairlength at most \( 1 \). Since every set \( N^- (T, o, i) \) contains at most one non-leaf vertex, the expressions \( X_i, 1 \leq i \leq n \), defined in the proof of Theorem 7 are linear NLC-width expressions.

Since every graph of linear NLC-width \( k \) has clique-width at most \( k + 1 \), see [5], we get the following corollary.

**Corollary 14.** The k-power graph of a graph of path-width \( l \) has linear clique-width at most \( (k + 1)^{l+1} \).

6. Conclusions

The proof of Theorem 7 also re-proves the well known fact from [3,15] that the NLC-width of a graph always can be bounded in its tree-width. More precisely, in Theorem 7 for \( k = 1 \) we show that a graph of tree-width \( l \) has NLC-width at most \( 2^{l+1} - 1 \).
The notation of Steiner-power graphs was researched in [31,23,24] as a more general concept than tree-powers and leaf-powers graphs with applications in computational biology. The k-Steiner-power graph G of a tree T is a graph whose vertices are a subset of V(T). The vertices of T that are not vertices of G are also denoted as Steiner points [31,24]. Two vertices in G are adjacent if and only if, there is a path between them in T of length at most k. Tree T is called the k-Steiner-root graph of G.

Let G be the k-Steiner-power graph of k-Steiner-root graph T. If V_T is equal to the set of leaves of T, then G also is the k-leaf power graph of T. Further, if V_T is equal to V_T, then G also is the k-tree power graph of T. Thus, it is easy to see that the k-Steiner-power graph G is always an induced subgraph of the k-tree power graph of T. This observation implies by Theorem 4 the following NLC-width and clique-width bounds for k-Steiner power graphs.

**Corollary 15.** The k-Steiner power graph of a tree has NLC-width at most \(k+2\) and clique-width at most \(k+2+\max\{\frac{k}{2}, 1\}\).

**Acknowledgements**

The authors want to thank the two anonymous referees for their helpful remarks.

**References**


