Vertex-minor reductions can simulate edge contractions

Mamadou Moustapha Kanté [1]

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

Abstract

We prove that *local complementation* and vertex deletion, operations from which *vertex-minors* are defined, can simulate *edge contractions*. As an application, we prove that the *rank-width* of a graph is linearly bounded in term of its *tree-width*.

Key words: Clique-width, rank-width, tree-width, tree-decomposition, local complementation, edge contraction, vertex-minor.

1 Introduction

Minor inclusion is an essential notion in graph theory because it yields the *Graph Minor Theorem* saying that every minor-closed family of graphs is characterized by finitely many excluded minors [17]. It follows together with another result [16] that every minor-closed family of graphs has a cubic-time recognition algorithm. The notion of a minor is closely related with that of *tree-width* [14].

Also very important is the notion of *vertex-minor* [10], closely linked with the notion of *rank-width*[12], another complexity measure on graphs introduced as an approximation of *clique-width* [6], but of independent interest. For a graph G and a vertex x of G, the *local complementation at x* replaces in G the subgraph induced by the neighbors of x by its complement graph. A graph H is a vertex-minor of G if H can be obtained from G by a sequence of local complementations and deletions of vertices. Oum [10] proved that for fixed k, there is a finite list of graphs such that a graph G has rank-width at most k if and only if no graph in this list is isomorphic

[1] Research supported by the french ANR-project "Graph decompositions and algorithms (GRAAL). Email: mamadou.kante@labri.fr

to a vertex-minor of G. It is not known yet whether vertex-minor inclusion well-quasi-orders the class of all graphs.

In this work we relate the two notions of minor and of vertex-minor by proving that edge contractions can be simulated by vertex deletions and local complementations : these two latter graph operations are those that define the vertex-minor relation.

The deletion of an edge can be simulated by vertex-minor operations. Let G be a simple undirected graph and e be an edge xy linking x and y in G. To simulate the deletion of e by vertex-minor operations, we introduce a new vertex x' adjacent to x and y. We obtain a graph G' which is G augmented with the vertex x' and the edges xx' and yx'. By applying a local complementation at x', we delete the edge e. By deleting the vertex x', we get the graph G-e which is G without the edge e, and G-e is a vertex-minor of G'.

Using a more complicated process, we will prove that edge contractions can be simulated by creating "*twin vertices*" (like x' for x in the above case) and taking vertex-minors. This technique allows us to prove that if G has tree-width at most k then G has rank-width at most $4k+2$. This bound is not optimal since Oum proved recently [13] that the rank-width of G is at most $k+1$, by using a different technique based on *tangles* and *branch-width* [15].

Outline. In Section 2 we will present some notations and recall the notions of tree-width, clique-width and rank-width and some existing results used throughout the paper. In Section 3 we show how to simulate edge contractions. As an application, we prove in Section 4 that rank-width is linearly bounded in term of tree-width. We conclude by some perspectives in Section 5.

2 Notation and definitions

We denote by $|C|$ the cardinality of a set C. The set of subsets of a set C is denoted by $\mathscr{P}(C)$. Our main results concern undirected simple graphs. However, for some proofs, we will use graphs with both directed and undirected edges. Directed edges will be called *arcs*. For a graph G, we let V_G denote its vertex set and E_G its set of edges and/or arcs. An edge between x and y is denoted by xy (equivalently yx). An arc from x to y is denoted by $\overrightarrow{xy}$ and y is called the target of the arc $\overrightarrow{xy}$. If G is a directed graph, we denote by $und(G)$ the simple undirected graph obtained from G by omitting the direction of the arcs. For $x \in V_G$ we denote by $neigh_G(x)$ the set of neighbors of x in G. We denote by $[m]$ the set $\{1,2,\ldots,m\}$.

We denote by $G[X]$ the subgraph of G induced by $X \subseteq V_G$ and by G^F the subgraph of G induced by $F \subseteq E_G$ ($E_{G^F} = F$ and V_{G^F} is the set of vertices incident to an edge in F). For $F \subseteq E_G$, we denote by G/F the graph possibly with loops and multiple edges obtained from G by contracting the edges of F. In order to avoid confusions in some technical lemmas, the vertices of trees will be called *nodes*.

A *forest* is a disjoint union of trees. A tree T is *rooted* if there exists a distinguished node r called the *root* of T. Then a rooted tree is directed so that all nodes are reachable from the root by a directed path. A *rooted forest* is a forest where all the trees, which are its connected components, are rooted. Let T be a rooted tree. For $u \in V_T$, we denote by $T\downarrow u$ the subtree of T rooted at u induced by the set of all descendants of u.

2.1 Tree-Width and Strong tree-width

We recall the notion of *tree-width* [14].

Definition 1 *A* tree-decomposition *of a graph $G = (V_G, E_G)$ is a pair (T, f) such that $T = (V_T, E_T)$ is a tree, f is a mapping associating with every node u of T a subset $f(u)$ of V_G such that:*

(1) $\bigcup_{u\in V_T} f(u) = V_G$,
(2) for each edge xy or arc $\overrightarrow{xy}$ of G, there exists one node u in V_T with x, y in $f(u)$,
(3) for all $u, v, w \in V_T$, if v is on the path from u to w in T then $f(u) \cap f(w) \subseteq f(v)$.

In *(2)* it is convenient, for each edge xy or arc $\overrightarrow{xy}$ to choose one node u such that $x, y \in f(u)$.

The *width* of a tree-decomposition (T, f) is $\max_{u\in V_T}\{|f(u)|\} - 1$. The *tree-width* of a graph $G = (V_G, E_G)$, denoted by $twd(G)$, is the minimum width over all tree-decompositions of G. We say that (T, f) is *rooted* if T is.

We now recall the definition of a *strong tree-decomposition* [18].

Definition 2 *A* strong tree-decomposition *of a graph $G = (V_G, E_G)$ is a pair (T, f) as in Definition 1 such that:*

(1) $\{f(u) \mid u \in V_T\}$ is a partition of V_G,
(2) for each edge xy or arc $\overrightarrow{xy}$ of G:
    (2.1) either there exists a node u in V_T with $x, y \in f(u)$,
    (2.2) or there exists an edge uv in E_T with $x \in f(u)$ and $y \in f(v)$ or vice versa.

The edges xy or arcs $\overrightarrow{xy}$ of type (2.2) are called the "shared edges or arcs of G". This notion is relative to chosen a strong tree-decomposition.

For $u \in V_T$ we call $f(u)$ the *box* of u. The *width* of a strong tree-decomposition (T, f) is $\max_{u\in V_T} |f(u)|$. The *strong tree-width* of a graph $G = (V_G, E_G)$, denoted by *stwd*(G), is the minimum width over all strong tree-decompositions of G. We say that (T, f) is *rooted* if T is.

Let (T, f) be a rooted strong tree-decomposition of a graph G. For $u \in V_T$, we denote by $G\downarrow u$ the graph $G[\bigcup_{v \in V_{T\downarrow u}} f(v)]$.

Tree-decompositions and related notions have been studied for the last two decades. See the surveys and articles [1,2] by Bodlaender. In the rest of the paper we consider rooted tree-decompositions and rooted strong tree-decompositions.

Let us recall a useful lemma and prove a technical lemma which will be used in Section 4.

Lemma 3 *(Bodlaender [2]) Suppose the tree-width of a graph G is k. Then G has a tree-decomposition (T, f) of width k such that:*

(1) For each $u \in V_T$ we have $|f(u)| = k+1$.
(2) For each $\overrightarrow{uv} \in E_T$ we have $|f(u) \cap f(v)| = k$.

Lemma 4 *Let (T, f) be a tree-decomposition of an undirected graph G of width k. There exist a graph H and a strong tree-decomposition (T, g) of H of width $k+1$ such that $G = H/F$ where F is the set of shared edges. The graph H^F is a forest.*

Proof. We let $H = (V_H, E_H)$ where:

$$\begin{aligned} V_H &= \{x_u \mid u \in V_T,\ x \in f(u)\}, \\ E_H &= \{x_u x_v \mid \overrightarrow{uv} \in E_T \ \wedge\ x \in f(u) \cap f(v)\} \cup \\ &\qquad \{x_u y_u \mid u \in V_T \ \wedge\ x, y \in f(u) \ \wedge\ xy \in E_G \text{ is in } f(u)\}. \end{aligned}$$

For each $u \in V_T$ we let $g(u) = \{x_u \mid x \in f(u)\}$. It is easy to verify that (T, g) is a strong tree-decomposition of H, the shared edges are the edges $\{x_u x_v \mid \overrightarrow{uv} \in E_T \ \wedge\ x \in f(u) \cap f(v)\}$. They form a set F that spans a forest and $G = H/F$ (by the definition of tree-decomposition). See Figure 1 for an example. The shared edges are dotted and marked by ε (because they are *in fine* contracted). □

2.2 Clique-width

We recall the definition of *clique-width* [5,6]. Here we deal with labeled undirected graphs (unlabeled graphs are considered as graphs whose vertices have all the same label). Let k be a positive integer. A k-graph is a graph whose vertices are labeled with labels from $[k]$. We define formally a k-graph as $G = (V_G, E_G, \delta_G)$ where $\delta_G(x) \in [k]$ for each $x \in V_G$. We recall the following operations:

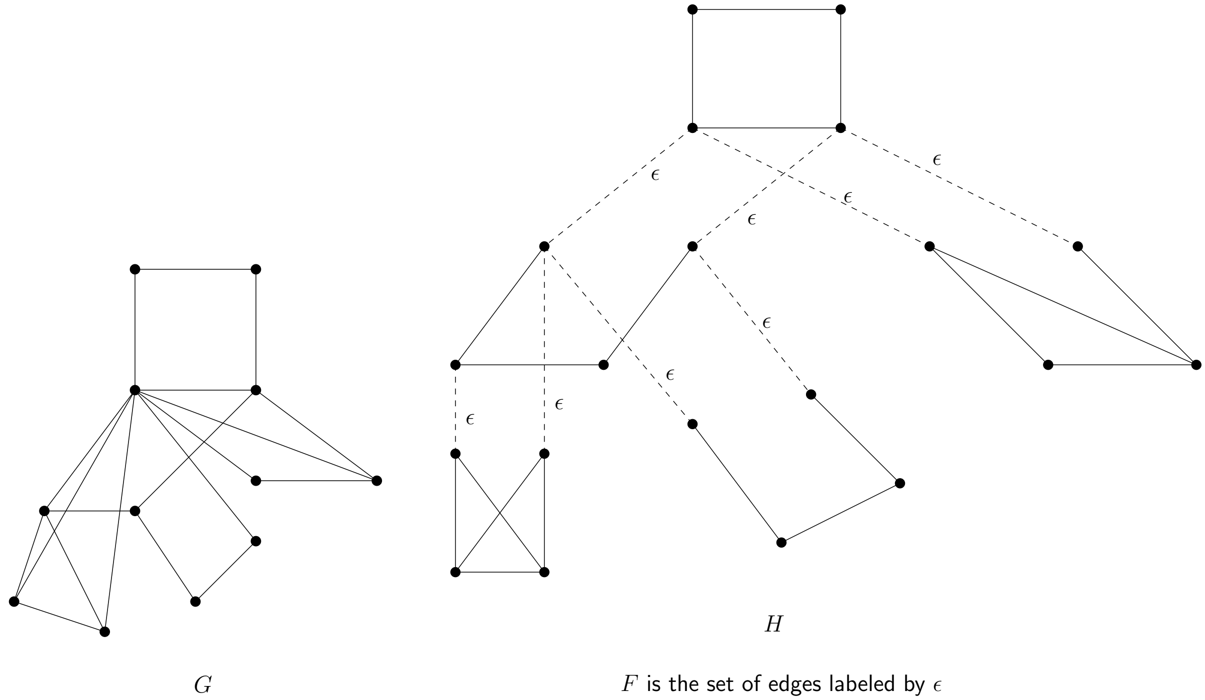


Figure 1. A graph G and the corresponding graph H.

(1) For k-graphs $G = (V_G, E_G, \delta_G)$ and $H = (V_H, E_H, \delta_H)$ such that $V_G \cap V_H = \emptyset$ (if not, we take a disjoint copy of H), we denote by $G \oplus H$ the k-graph $(V_G \cup V_H, E_G \cup E_H, \delta' = \delta_G \cup \delta_H)$ and call it the disjoint union of G and H with:

$$\delta'(x) = \begin{cases} \delta_G(x) & \text{if } x \in V_G, \\ \delta_H(x) & \text{if } x \in V_H. \end{cases}$$

(2) For a k-graph $G = (V_G, E_G, \delta_G)$, for distinct $i, j \in [k]$, we denote by $\eta_{i,j}(G)$, the k-graph (V_G, E'_G, δ_G) where

$$E'_G = E_G \cup \{xy \mid x,y \in V_G \ \wedge \ x \neq y \ \wedge \ i \in \delta_G(x),\ j \in \delta_G(y)\}.$$

(3) For a k-graph $G = (V_G, E_G, \delta_G)$, for distinct $i, j \in [k]$, we denote by $\rho_{i \to j}(G)$, the k-graph (V_G, E_G, δ'_G) where

$$\delta'_G(x) = \begin{cases} j & \text{if } \delta_G(x) = i, \\ \delta_G(x) & \text{otherwise.} \end{cases}$$

(4) For each $i \in [k]$, $\mathbf{i}$ denotes a k-graph with a single vertex labeled by i.

A *k-expression* is a well-formed term written with the symbols $\oplus$, $\eta_{i,j}$, $\rho_{i \to j}$, $\mathbf{i}$ where $i \neq j$ and $i, j \in [k]$. Every k-expression t defines, up to isomorphism, a k-graph G. The clique-width of a graph G, denoted by $cwd(G)$, is the minimum k such that there exists a k-expression t defining G.

Clique-width can be considered as more powerful than tree-width. It is known that bounded tree-width implies bounded clique-width and not vice-versa (cliques have unbounded tree-width but have clique-width 2). The following theorem gives an upper-bound on the clique-width of graphs of bounded tree-width:

Theorem 5 *(Corneil and Rotics [4]) For every simple undirected graph G,*

$$cwd(G) \leq 3.2^{twd(G)-1}.$$

Contrary to the case of tree-width, there is no known polynomial algorithm for the recognition of graphs of clique-width $\leq k$ for $k > 3$ (for $k \leq 3$ there are, see [3]) which produces the k-expression of the graph. (Note that the clique-width minimization problem is NP-Hard [9]). Oum and Seymour investigated the problem and Oum proved the following theorem:

Theorem 6 *(Oum [11]) For each $k \in \mathbb{N}$, there is an algorithm that for an input graph G, undirected, either yields correctly that $cwd(G) > k$ or outputs a $(2^{24k+1} - 1)$-expression of G. Its running time is $O(|V_G|^3)$.*

This result uses the graph complexity measure *rank-width* (see Section 2.3). It is equivalent to clique-width in the sense that the same families of undirected graphs have bounded rank-width and bounded clique-width. However, rank-width has better algebraic properties. This explains why the class of graphs of rank-width $\leq k$ is closed under taking vertex-minors.

2.3 Rank-width

In this section graphs are simple, loop-free and undirected. We now recall the definition of *rank-width* [10] and some results about it, needed throughout the paper.

For an (R,C)-matrix $M = (m_{ij} \mid i \in R, j \in C)$ over a field F, if $X \subseteq R$, $Y \subseteq C$, let $M[X,Y]$ denote the sub-matrix $(m_{ij} \mid i \in X, j \in Y)$. For a graph $G = (V_G, E_G)$, let A_G be its adjacency (V_G,V_G)-matrix over $GF(2)$.

Definition 7 *Let $G = (V_G, E_G)$ be a graph. We define the* cut-rank *function ρ_G of G by letting $\rho_G(X) = rk(A_G[X, V_G \backslash X])$, $X \subseteq V_G$, where rk is the matrix rank function.*

A *sub-cubic tree* is an undirected tree where the degree of each node is at most 3. A *rank-decomposition* of a graph $G = (V_G, E_G)$ is a pair (T, f) of a sub-cubic tree T and a bijective function $f : V_G \to \{t \mid t \text{ is a leaf of } T\}$. (Leaves are nodes of degree 1).

For an edge e of T, the connected components of $T \backslash e$ induce a bipartition of the set of leaves of T, hence a bipartition (X_e, Y_e) of the set of vertices of G. The width of an edge e of a rank-decomposition (T, f) is $\rho_G(X_e)$. The width of a rank-decomposition (T, f) is the maximum width over all edges of T. The rank-width of G, denoted by $rwd(G)$, is the minimum width over all rank-decompositions of G.

Proposition 8 *(Oum and Seymour [12]) For every graph G,*

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

This result combined with Theorem 5 gives $rwd(G) \leq 3 \times 2^{twd(G)-1}$. We will improve this bound and prove that rank-width is linearly bounded in term of tree-width in the following proposition:

Proposition 9 *For every graph G, $rwd(G) \leq 4 \times twd(G) + 2$.*

Oum improves recently this bound. Using different techniques based on tangles and branch-width [15], he proves the following:

Proposition 10 *(Oum [13]) For every graph G, $rwd(G) \leq twd(G) + 1$.*

We now introduce definitions relative to the notion of *vertex-minor*. For two sets A and B we denote by $A - B$ the set of elements in A and not in B. For two sets A and B, we let $A \triangle B = (A - B) \cup (B - A)$.

Definition 11 *Let G be a graph and $x \in V_G$. The graph obtained by applying* local complementation *at x to G is*

$$G * x = (V_G, E_G \triangle \{yz \mid xy, xz \in E_G, z \neq y\}).$$

The graph $G * x$ is obtained from G by edge-complementing the subgraph induced by the vertices adjacent to x. We say that H is *locally equivalent* to G if H can be obtained by applying a sequence of local complementations to G. We say that H is a *vertex-minor* of G if H can be obtained by applying a sequence of vertex deletions and local complementations to G.

Lemma 12 *(Courcelle and Oum [7]) Let H and G be graphs and x be a vertex of H.*

(1) If H is an induced subgraph of $G * x$, then $H * x$ is an induced subgraph of G.
(2) A graph H is a vertex-minor of G if and only if H is an induced subgraph of a graph that is locally equivalent to G.
(3) A graph locally equivalent to a vertex-minor of G is also a vertex-minor of G.

Proposition 13 *(Oum [10]) Let H and G be two graphs. If H is locally equivalent to G, then $rwd(H) = rwd(G)$. If H is a vertex-minor of G, then $rwd(H) \leq rwd(G)$.*

3 Vertex-minor reductions and edge contractions

We recall some notations and definitions. Let $G = (V_G, E_G)$ be an undirected simple loop-free graph. We say that $J \subseteq E_G$ is *good* if G^J is a forest and G/J has no loop or multiple edges. This is equivalent to saying that in G every cycle contains at least 3 edges not in J. For a vertex x of G, we denote by G-x the induced subgraph $G[V_G - \{x\}]$ of G. If a rooted forest is reduced to one arc f, we will denote it by $\{f\}$.

Let F be a rooted forest. We denote by V_F^{root} the set $\{x \in V_F \mid x \text{ is a root}\}$ and by V_F^{nroot} the set $V_F - V_F^{root}$, i.e, of vertices that are the targets of some arcs in F. We say that F is a *rooted forest in G* if $E_{und(F)} \subseteq E_G$, i.e., $G^{E_{und(F)}}$ is a subgraph of G. We say that F is a *good rooted forest in G* if F is a rooted forest in G and $E_{und(F)}$ is good.

Let us define the two following operations:

Definition 14 ($\bullet$) *Let G be a graph and $x \in V_G$. The graph obtained by applying* lc-deletion *at x to G is $G \bullet x = (G * x)$-x.*

It is clear that $G \bullet x$ is a vertex-minor of G. We note that $G \bullet x \bullet y$ is not necessarily equal to $G \bullet y \bullet x$. See Figure 2 for an illustration of Definition 14.

Definition 15 ($\boxtimes$) *Let G be a simple undirected graph and F be a rooted forest in G. The graph obtained by applying* local augmentation *at F to G is $G \boxtimes F = (V_{G\boxtimes F}, E_{G\boxtimes F})$ where:*

$$\begin{aligned}
V_{G\boxtimes F} &= V_G \cup \{x^t \mid x \in V_F^{nroot}\},\\
E_{G\boxtimes F} &= E_G \cup \{x^t y \mid x \in V_F^{nroot} \ \wedge\ xy \in E_G \ \wedge\ \overrightarrow{xy} \notin F \ \wedge\ \overrightarrow{yx} \notin F\}\\
&\quad \cup \{x^t y^t \mid x, y \in V_F^{nroot} \ \wedge\ xy \in E_G\}.
\end{aligned}$$

x^t is a new vertex called the "*twin*" of x.

We illustrate the construction of Definition 15 with an example. Figure 3 shows a graph G, a rooted forest F in G and the graph $G \boxtimes F$. The connected components of F are T_1 induced by $\{a, b, c, d\}$ with root 1 and T_2 induced by $\{e\}$ with root 6. One can verify we have $V_F^{root} = \{1, 6\}$ and $V_F^{nroot} = \{2, 3, 4, 5, 7\}$. Then

$$\begin{aligned}
V_{G\boxtimes F} &= V_G \cup \{2^t, 3^t, 4^t, 5^t, 7^t\},\\
E_{G\boxtimes F} &= E_G \cup \{\{2^t, 8\}, \{2^t, 9\}, \{4^t, 6\}, \{4^t, 10\}, \{4^t, 11\}, \{5^t, 12\}, \{5^t, 13\}, \{7^t, 12\}, \{7^t, 14\}\}\\
&\quad \cup \{\{3^t, 4^t\}, \{3^t, 5^t\}\}.
\end{aligned}$$

Our main result is the following:

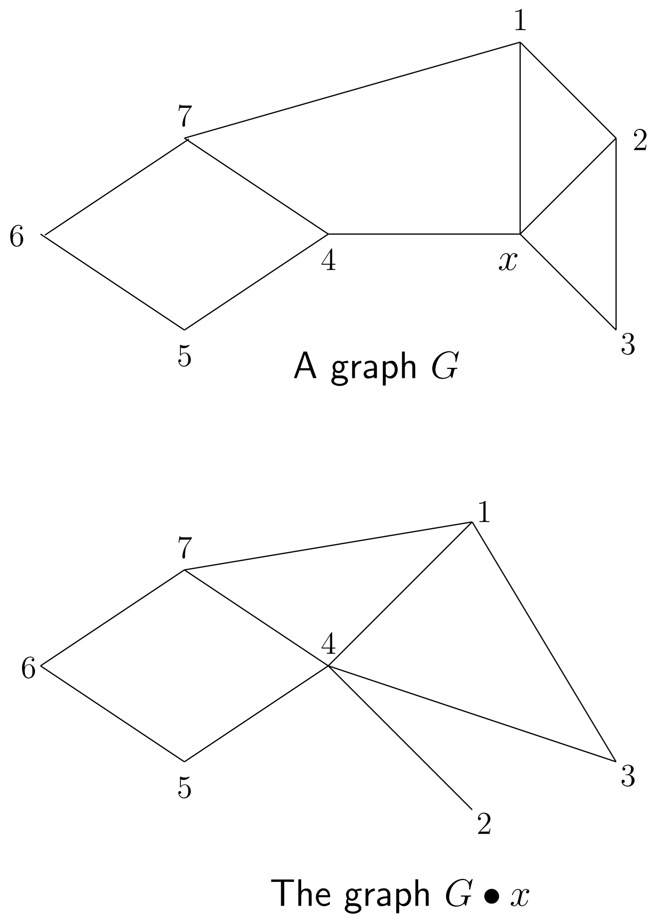


Figure 2.

Theorem 16 *Let G be a graph and F be a* good *rooted forest in G. Then $G/E_{und(F)}$ is a vertex-minor of $G \boxtimes F$.*

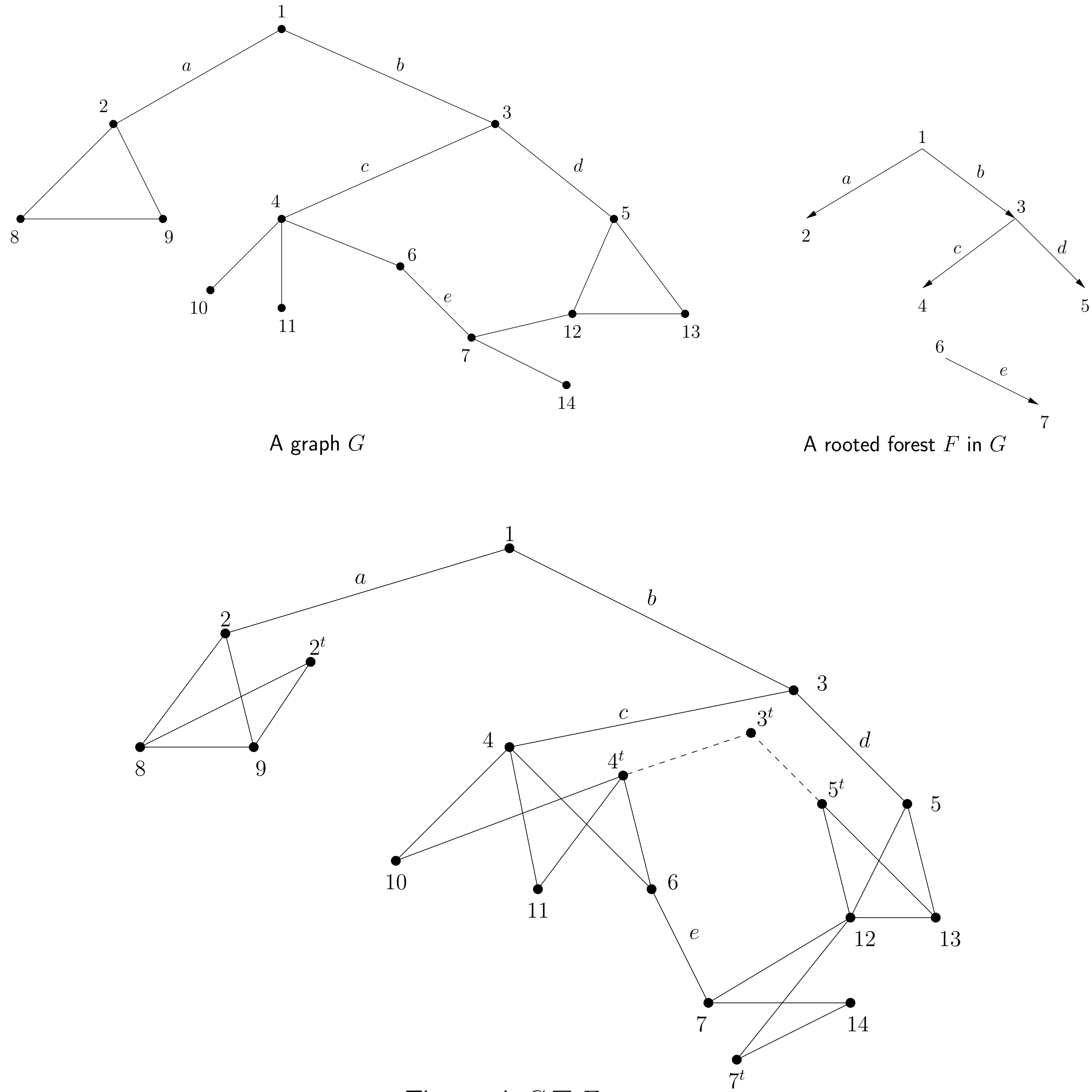
In order to prove Theorem 16, we prove how to simulate edge contractions by vertex-minor operations. For that we use the operations $\bullet$ and $\boxtimes$ defined above in this section. We begin by proving some technical lemmas.

Fact 17 *Let G be a graph and $f = \{\overrightarrow{yx}\}$ be such that $\{f\}$ is a good rooted forest in G. Then $(G \boxtimes \{f\}) \bullet x \bullet x^t = G/e$ where $e = yx$.*

Proof. We let $y, z_1, \ldots, z_m$ be the neighbors of x. The effect of contracting e can be described as follows:

(a) deletion of x and the edges incident to x,
(b) creation of edges between y and z_i for each $i \in [m]$.

Since $\{f\}$ is a good rooted forest in G, there is no edge in G between y and any z_i for any $i \in [m]$. The effect of applying lc-deletion at x to $G \boxtimes \{f\}$ is thus:



A graph G A rooted forest F in G

The graph $G \boxtimes F$

Figure 3.

(1) creation of edges between y and z_i for each $i \in [m]$ (that is (b)),
(2) creation of edges $z_i z_j$ where $z_i z_j \notin E_G,\ i \neq j$,
(3) deletion of edges $z_i z_j \in E_G,\ i \neq j$,
(4) deletion of x and the incident edges to x (that is (a)).

The lc-deletion applied at x links y to z_i, deletes x, but also deletes existing edges between the neighbors z_i of x (that is (3)) and creates edges in place of non-existing ones (that is (2)). Since $\{f\}$ is good we have $neigh_{G\boxtimes\{f\}}(x^t) = \{z_1, \ldots, z_m\}$. Therefore lc-deleting at x^t undoes (2) and (3) and deletes x^t and its incident edges. Then

$(G \boxtimes \{f\}) \bullet x \bullet x^t = G/e$. $\square$

Lemma 18 *Let G be a graph. Let F be a good rooted forest in G and $f = \overrightarrow{yx}$ be an arc in F where x is a leaf. Then $(G \boxtimes F) \bullet x \bullet x^t = (G/e) \boxtimes (F - \{f\})$ where $e = yx$.*

Proof. We distinguish two cases: either y is a root or not (see Figures 4 and 5 for an illustration).

Claim 19 *Let F be a good rooted forest in G and $f = \overrightarrow{yx}$ be an arc in F where y is a root and x is a leaf. Then $(G \boxtimes F) \bullet x \bullet x^t = (G/e) \boxtimes (F - \{f\})$ where $e = yx$.*

Proof. Let $y, z_1, \dots, z_m$ be the neighbors of x. The effect of contracting the edge yx in G can be described as follows:

(a) deletion of x and its incident edges,
(b) creation of edges between y and z_i for each $i \in [m]$.

Since F is a good rooted forest in G, y is not adjacent to any z_i in G. But in G/e, y is adjacent to all z_i. We get:

$$
\begin{aligned}
V_{(G/e)\boxtimes(F-\{f\})} &= (V_G - \{x\}) \cup \{z^t \mid z \in V_F^{nroot} \ \wedge \ z \neq x\}, \\
E_{(G/e)\boxtimes(F-\{f\})} &= (E_G - \{xz \mid xz \in E_G\}) \cup \{yz_i \mid i \in [m]\} \\
&\quad \cup \{yz_i^t \mid i \in [m] \ \wedge \ z_i \in V_F^{nroot}\} \\
&\quad \cup \{u^t z \mid u \in V_F^{nroot} \ \wedge \ u \neq x \ \wedge \ uz \in E_G \ \wedge \ \overrightarrow{uz}, \ \overrightarrow{zu} \notin F\} \\
&\quad \cup \{u^t z^t \mid u, z \in V_F^{nroot} \ \wedge \ u, z \neq x \ \wedge \ uz \in E_G\}.
\end{aligned}
$$

We have $neigh_{G\boxtimes F}(x) = \{z_1, \dots, z_m\} \cup \{z_i^t \mid i \in [m] \ \wedge \ z_i \in V_F^{nroot}\} \cup \{y\}$. Therefore the effect of applying lc-deletion at x to $G \boxtimes F$ can be described as follows:

(1) creation of edges yz_i for each $i \in [m]$ (that is (b)),
(2) creation of edges yz_i^t for each $z_i \in V_F^{nroot}$ (edges created in $(G/e) \boxtimes (F - \{f\})$),
(3) creation of edges $z_i z_j$, $z_i z_j^t$, $z_i^t z_j^t$ where $z_i z_j \notin E_G$ and $i \neq j$ and of edges $z_i z_i^t$ for each $z_i \in V_F^{nroot}$,
(4) deletion of edges $z_i z_j$, $z_i z_j^t$ and $z_i^t z_j^t$ where $z_i z_j \in E_G$ and $i \neq j$,
(5) deletion of x and its incident edges (that is (a)).

By definition, $neigh_{G\boxtimes F}(x^t) = \{z_1, \dots, z_m\} \cup \{z_i^t \mid i \in [m] \ \wedge \ z_i \in V_F^{nroot}\}$. Then the effect of applying lc-deletion at x^t to $(G \boxtimes F) \bullet x$ can be described as follows:

(3') deletion of edges $z_i z_j$, $z_i z_j^t$, $z_i^t z_j^t$ where $z_i z_j \notin E_G$ and $i \neq j$ and of edges $z_i z_i^t$ for each $z_i \in V_F^{nroot}$ (undo (3)),
(4') creation of edges $z_i z_j$, $z_i z_j^t$ and $z_i^t z_j^t$ where $z_i z_j \in E_G$ and $i \neq j$ (undo (4)),

(5') deletion of x^t and its incident edges.

Then we have:

$$
\begin{aligned}
V_{(G\boxtimes F)\bullet x \bullet x^t} &= (V_G - \{x\}) \cup \{z^t \mid z \in V_F^{nroot}\} \setminus \{x^t\},\\
E_{(G\boxtimes F)\bullet x \bullet x^t} &= (E_G - \{xz \mid xz \in E_G\}) \cup \{yz_i \mid i \in [m]\}\\
&\quad \cup \{yz_i^t \mid i \in [m] \ \wedge \ z_i \in V_F^{nroot}\}\\
&\quad \cup \{u^t z \mid u \in V_F^{nroot} \ \wedge \ u \neq x \ \wedge \ uz \in E_G \ \wedge \ \overrightarrow{uz},\ \overrightarrow{zu} \notin F\}\\
&\quad \cup \{u^t z^t \mid u,z \neq x \ \wedge \ u,z \in V_F^{nroot} \ \wedge \ uz \in E_G\}.
\end{aligned}
$$

We thus deduce that $(G\boxtimes F) \bullet x \bullet x^t = (G/e) \boxtimes (F - \{f\})$. $\square$

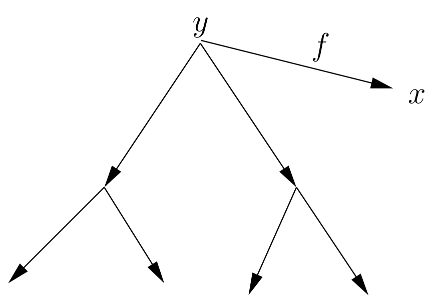


Figure 4. F and $\{f\}$ with y a root.

Claim 20 *Let F be a good rooted forest in G and $f = \overrightarrow{yx}$ be an arc in F where y is a non-root and x is a leaf. Then $(G\boxtimes F) \bullet x \bullet x^t = (G/e) \boxtimes (F - \{f\})$ where $e = yx$.*

Proof. Let $y, z_1, \ldots, z_m$ be the neighbors of x. The effect of contracting the edge yx in G can be described as follows:

(a) deletion of x and its incident edges,
(b) creation of edges between y and z_i for each $i \in [m]$.

Since F is a good rooted forest in G, y is not adjacent to any z_i in G, but it is in G/e. We get:

$$
\begin{aligned}
V_{(G/e)\boxtimes(F-\{f\})} &= (V_G - \{x\}) \cup \{z^t \mid z \in V_F^{nroot} \ \wedge \ z \neq x\},\\
E_{(G/e)\boxtimes(F-\{f\})} &= (E_G - \{xz \mid xz \in E_G\}) \cup \{yz_i \mid i \in [m]\}\\
&\quad \cup \{yz_i^t \mid i \in [m] \ \wedge \ z_i \in V_F^{nroot}\}\\
&\quad \cup \{u^t z \mid u \in V_F^{nroot} \ \wedge \ u \neq x \ \wedge \ uz \in E_G \ \wedge \ \overrightarrow{uz},\ \overrightarrow{zu} \notin F\}\\
&\quad \cup \{u^t z^t \mid u,z \in V_F^{nroot} \ \wedge \ u,z \neq x \ \wedge \ uz \in E_G\}\\
&\quad \cup \{y^t z_i \mid i \in [m]\} \cup \{y^t z_i^t \mid i \in [m] \ \wedge \ z_i \in V_F^{nroot}\}.
\end{aligned}
$$

We have $neigh_{G\boxtimes F}(x) = \{z_1,\ldots,z_m\} \cup \{z_i^t \mid i \in [m] \ \wedge \ z_i \in V_F^{nroot}\} \cup \{y\}$. Then the effect of applying lc-deletion at x to $G\boxtimes F$ is the same as in Claim 19.

By definition, $neigh_{G\boxtimes F}(x^t) = \{z_1,\ldots,z_m\} \cup \{z_i^t \mid i \in [m] \ \wedge \ z_i \in V_F^{nroot}\} \cup \{y^t\}$. So lc-deleting at x^t in $(G\boxtimes F)\bullet x$ has the same effect as in Claim 19 with two additional steps which create two types of edges:

(1') creation of edges $y^t z_i$ for each $i \in [m]$ (edges created in $(G/e)\boxtimes(F-\{f\})$),
(2') creation of edges $y^t z_i^t$ for each $z_i \in V_F^{nroot}$ (edges created in $(G/e)\boxtimes(F-\{f\})$).

Then we have:

$$\begin{aligned}
V_{(G\boxtimes F)\bullet x\bullet x^t} &= (V_G - \{x\}) \cup \{z^t \mid z \in V_F^{nroot}\} \setminus \{x^t\}, \\
E_{(G\boxtimes F)\bullet x\bullet x^t} &= (E_G - \{xz \mid xz \in E_G\}) \cup \{yz_i \mid i \in [m]\} \\
&\quad \cup \{yz_i^t \mid i \in [m] \ \wedge \ z_i \in V_F^{nroot}\} \\
&\quad \cup \{u^t z \mid u \in V_F^{nroot} \ \wedge \ u \neq x \ \wedge \ uz \in E_G \ \wedge \ \overrightarrow{uz},\ \overrightarrow{zu} \notin F\} \\
&\quad \cup \{u^t z^t \mid u,z \neq x \ \wedge \ u,z \in V_F^{nroot} \ \wedge \ uz \in E_G\} \\
&\quad \cup \{y^t z_i \mid i \in [m]\} \cup \{y^t z_i^t \mid i \in [m] \ \wedge \ z_i \in V_F^{nroot}\}.
\end{aligned}$$

We thus deduce that $(G\boxtimes F)\bullet x \bullet x^t = (G/e)\boxtimes(F-\{f\})$. $\square$

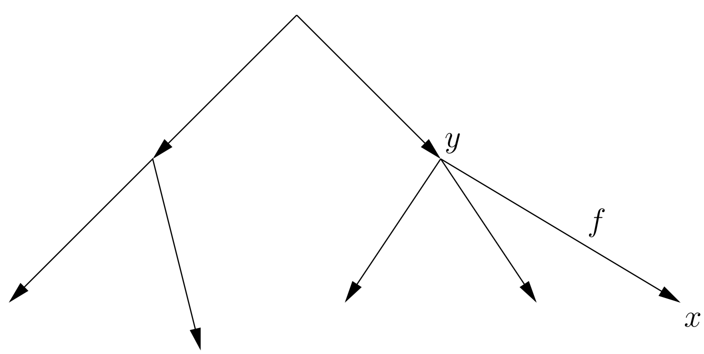


Figure 5. F and $\{f\}$ with y not a root.

End of proof of Lemma 18. We considered the two cases (y root or not) in Claim 19 and in Claim 20. In both cases, we have $(G\boxtimes F)\bullet x\bullet x^t = (G/e)\boxtimes(F-\{f\})$. $\square$

Proof of Theorem 16. We prove it by induction on the size of F. Let $V_F^{nroot} = \{x_1,\ldots,x_k\}$. Its elements are numbered from leaves to internal nodes in inverse topological order. We claim that $G/E_{und(F)} = (\ldots((G\boxtimes F)\bullet x_1 \bullet x_1^t)\bullet \ldots)\bullet x_k \bullet x_k^t$.

If $F = \{f\}$, let $f = \overrightarrow{yx_1}$. From Fact 17 we have $G/e = (G\boxtimes\{f\})\bullet x_1 \bullet x_1^t$ where $e = yx_1$.

We now assume that $|F| \geq 2$ and let $F = F_1 \cup \{f\}$ where $f = \overrightarrow{yx_1}$ and x_1, the target of f, is a leaf. We let $e = yx_1$. By definition we have $G/E_{und(F)} = (G/e)/E_{und(F_1)}$.

We observe that the edges incident to the vertices $x_1^t, \ldots, x_k^t$ in $G \boxtimes F$ are defined relatively to the pair (G,F) according to the definition of the operation $\boxtimes$. We also observe that F_1 is a good rooted forest in G/e and then the non-root vertices of F_1 are $x_2, \ldots, x_k$.

By Lemma 18 $(G \boxtimes F) \bullet x_1 \bullet x_1^t = (G/e) \boxtimes F_1$. Then the edges incident to $x_2, \ldots, x_k$ and $x_2^t, \ldots, x_k^t$ are the same in $(G \boxtimes F) \bullet x_1 \bullet x_1^t$ and in $(G/e) \boxtimes F_1$. Therefore we get

$$(\ldots((G \boxtimes F) \bullet x_1 \bullet x_1^t) \bullet \ldots) \bullet x_k \bullet x_k^t = (\ldots(((G/e) \boxtimes F_1) \bullet x_2 \bullet x_2^t) \bullet \ldots) \bullet x_k \bullet x_k^t.$$

By the inductive hypothesis we have

$$((G/e) \boxtimes F_1) \bullet x_2 \bullet x_2^t) \bullet \ldots) \bullet x_k \bullet x_k^t = (G/e)/E_{und(F_1)} = G/E_{und(F)}.$$

Then $G/E_{und(F)} = (\ldots((G \boxtimes F) \bullet x_1 \bullet x_1^t) \bullet \ldots) \bullet x_k \bullet x_k^t$. $\square$

4 Application to rank-width

In this section we prove Theorem 9. We first prove that clique-width is linear in term of strong tree-width.

Lemma 21 *Let G be a simple undirected graph, then $cwd(G) \leq 2 \times stwd(G) + 1$.*

Proof. Let (T, f) be a rooted strong tree-decomposition of width k of G. To prove the lemma, we introduce a binary operation. We first consider the particular case of the trees.

Let K and H be trees with one distinguished node labeled by 1 and all other nodes labeled by 0.
We let $K \odot H$ be obtained from $K \oplus H$, where K, H are disjoint, by a new edge from the distinguished node of K to the one of H and the distinguished node of K is made the distinguished one of the resulting tree. Clearly

$$K \odot H = \rho_{2 \to 0}(\eta_{1,2}(K \oplus \rho_{1 \to 2}(H))) \tag{1}$$

All trees can be generated from the operation $\odot$ and the constant **1**.

Let $n, m \leq k$. Assume now that K is a graph with distinguished vertices labeled from 1 to n, each label for one vertex. All other vertices are labeled by 0. Let H be similar with distinguished vertices labeled from 1 to m. Let t_K and t_H be terms that define respectively K and H as explained above. For $R \subseteq [n] \times [m]$ we define:

$$K \odot_R H = \left(\circ_{i\in[m]}\ \rho_{i'\to 0} \right) \left(\circ_{(i,j)\in R}\ \eta_{i,j'} \right) \left(t_K \oplus \left(\circ_{i\in[m]}\ \rho_{i\to i'} \right) \left(t_H \right) \right) \qquad (2)$$

Claim 22 *The simple loop-free undirected graphs of strong tree-width $\leq k$ are generated by the operations :*

- $\odot_R$ *for* $R \subseteq [k] \times [k]$,
- $\eta_{i,j}$ *for* $i, j \in [k],\ i \neq j$,
- *and the basic graphs* $\mathbf{1} \oplus \mathbf{2} \oplus \ldots \oplus \mathbf{n}$ *for* $1 \leq n \leq k$.

It follows from Claim 22 that $cwd(G) \leq 2k+1$ if $stwd(G) \leq k$. $\square$

Proof of Claim 22. We first color each box $f(u)$ with colors from 1 to $|f(u)|$ using a mapping γ_u, each label for one vertex (see Figure 6 for an example). We prove by induction on the number of nodes of T that for each $u \in V_T$, the graph $G{\downarrow}u$ labeled so that the vertices in $f(u)$ are labeled from 1 to $|f(u)|$ and all others are labeled by 0, is generated by the above operations.

Let $R_u = \{(\gamma_u(x), \gamma_u(y)) \mid x, y \in f(u)\ \wedge\ xy \in E_G\}$ and assume that $|f(u)| = n$. Let

$$t_u = (\circ_{(i,j)\in R_u}\ \eta_{i,j})(\mathbf{1} \oplus \mathbf{2} \oplus \ldots \oplus \mathbf{n}).$$

It is clear from the definition of R_u, that $val(t_u) = G[f(u)]$. If $V_T = \{u\}$, we have $G = G[f(u)]$, then the claim is verified. Now assume that $v_1, \ldots, v_p$ are the children of u (in Figure 6 $p = 2$). By the inductive hypothesis, for each child v_i of u, $G{\downarrow}v_i$, labeled as explained above, is generated by the above operations.

Let $R_i = \{(\gamma_u(x), \gamma_{v_i}(y)) \mid x \in f(u),\ y \in f(v_i)\ \wedge\ xy \in E_G\}$ for $i = 1, \ldots, p$. It is clear that $R_i \subseteq [k] \times [k]$ for $i = 1, \ldots, p$. By the definition of strong tree-decompositions and the inductive hypothesis, it only remains to add the shared edges between vertices of $f(u)$ and vertices of $f(v_i)$ for $i = 1, \ldots, p$. From the definition of R_i, if $x \in f(u),\ y \in f(v_i)$ and $xy \in E_G$, then $(\gamma_u(x), \gamma_{v_i}(y)) \in R_i$. We let

$$t = (((val(t_u) \odot_{R_1} G{\downarrow}v_1) \odot_{R_2} G{\downarrow}v_2) \ldots) \odot_{R_p} G{\downarrow}v_p.$$

It is easy to verify that the above expression defines $G{\downarrow}u$ as wanted (see Figure 6 for an example). If u is the root of T we have $G = G{\downarrow}u$. Then the claim is proved. $\square$

We illustrate the proof of Lemma 21 with an example, taking $p = 2$. Figure 6 shows a part of a strong tree-decomposition of a graph G (the sub-tree of the strong tree-decomposition rooted at u). The node u has two children v_1 and v_2. One can verify we have:

$$\begin{aligned}
R_1 &= \{(1,1),(1,2),(2,2)\},\\
R_2 &= \{(1,1),(3,1),(4,3)\},\\
G\downarrow u &= (G[f(u)] \odot_{R_1} G\downarrow v_1) \odot_{R_2} G\downarrow v_2.
\end{aligned}$$

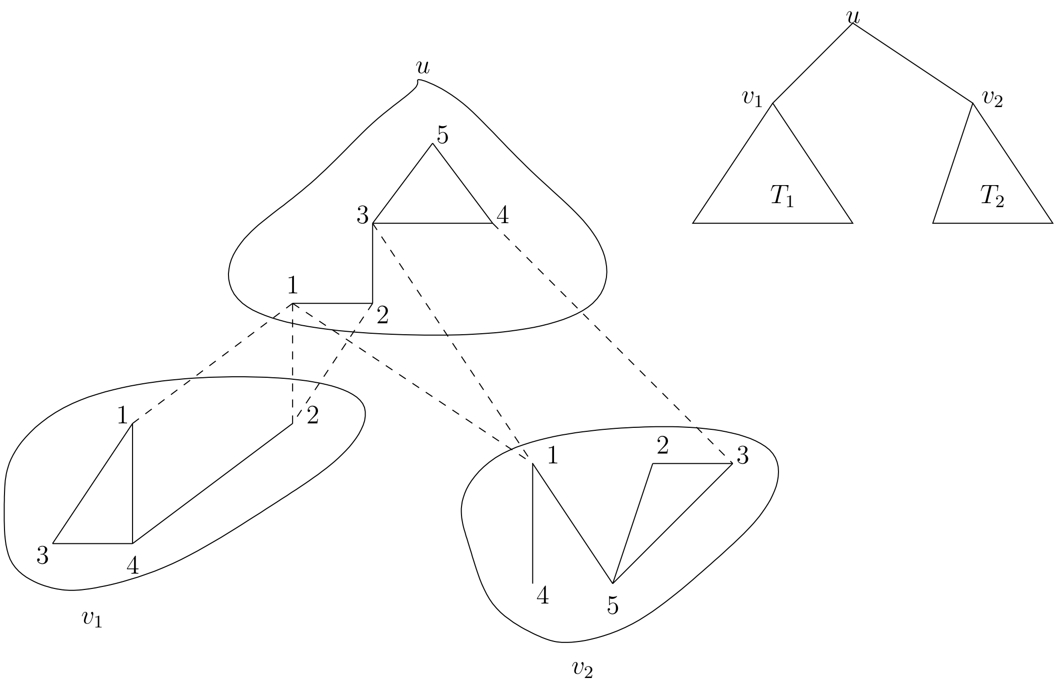


Figure 6. Illustrating the proof of Claim 22.

Remark 23 *We can note that if for each $\overrightarrow{uv} \in E_T$ the shared edges between $f(u)$ and $f(v)$ are incident to at most $k-i$ vertices in $f(v)$ then $cwd(G) \leq 2k-i+1$.*

Proof of Proposition 9. Let (T,f) be a rooted tree-decomposition of width k of G satisfying the condition of Lemma 3. By Lemma 4 we can build a graph H, a forest F and a strong tree-decomposition (T,g) of H with $G = H/E_F$. The notation g is as in Lemma 4. Let $\overrightarrow{F} = (V_{\overrightarrow{F}}, E_{\overrightarrow{F}})$ where :

$$\begin{aligned}
V_{\overrightarrow{F}} &= V_F\\
E_{\overrightarrow{F}} &= \{\overrightarrow{x_u x_v} \mid x_u x_v \in E_F \ \wedge \ \overrightarrow{uv} \in E_T\}
\end{aligned}$$

It is clear that $und(\overrightarrow{F}) = F$. By the definition of F, H/E_F has no loops or multiple edges, so E_F is good in H. Then $\overrightarrow{F}$ is a good rooted forest in H.

By Proposition 13 and Theorem 16, we have $rwd(G) \leq rwd(H \boxtimes \overrightarrow{F})$. We now prove that $rwd(H \boxtimes \overrightarrow{F}) \leq 4k+2$.

Let $h(u) = g(u) \cup \{x_u^t \mid x_u \in V_{\overrightarrow{F}}^{nroot}\}$. It is easy to prove that (T,h) is a strong tree-decomposition of $H \boxtimes \overrightarrow{F}$ of width at most $2k+1$. We have $2k+1$ instead of $2(k+1)$ because for each $u \in V_T$ the size of the set $\{x_u \mid x_u \in V_{\overrightarrow{F}}^{nroot}\}$ is at most k and then the size of the set $\{x_u^t \mid x_u \in V_{\overrightarrow{F}}^{nroot}\}$ is at most k. It is easy to verify (from the definition of (T,f)) that for each $\overrightarrow{uv}$ in (T,h) the shared edges between the vertices of $h(u)$ and the vertices of $h(v)$ are incident to at most $2k$ vertices in $h(v)$. Then by Lemma 21 and Remark 23 $cwd(H \boxtimes \overrightarrow{F}) \leq 4k+2$.

By Proposition 8 we have $rwd(H \boxtimes \overrightarrow{F}) \leq 4k+2$. Then $rwd(G) \leq 4k+2$. $\square$

5 Conclusion

In this paper we showed how to simulate edge contractions by duplications of certain vertices and vertex-minor operations. We proved that if G is a graph and F is a good rooted forest in G, then $G/E_{und(F)}$ is a vertex-minor of $G \boxtimes F$, which is a graph constructed from G and F ($und(F)$ is the undirected forest associated with F). This construction allows us to prove that the rank-width of G is linearly bounded in term of its tree-width. Even if the bound is not tight, we think that the proof method is interesting in its own because it relates several types of decompositions : tree-decomposition, strong tree-decomposition, rank-decomposition and clique-width expressions. Furthermore it relates vertex-minor reductions to edge contractions. We recall that we have also shown in the introduction how to simulate edge deletions by creation of new vertices and by vertex-minor reductions. We hope that this is a first step towards making links between minor operations and vertex-minor operations. We also recall that the problem of finding a “minor” inclusion relation for clique-width, analogous to minor inclusion for tree-width is still open.

Acknowledgment. The author would like to thank Bruno Courcelle for the valuable discussions that lead to the improvement of the proof of Theorem 9 and the referees for their critical reading.

References

[1] H. L. Bodlaender. A tourist guide through tree-width, *Acta Cybernetica*, 11(1993), 1-23.

[2] H. L. Bodlaender. A partial k-arboretum of graphs with bounded tree-width, *Theoretical Computer Science*, 209(1998), 1-45.

[3] D. G. Corneil, M. Habib, J.M. Lanlignel, B. Reed and U. Rotics. Polynomial time recognition of clique-width ≤ 3 graphs, *LATIN'00:Latin American Theoretical Informatics*, 1776(2000), 126-134.

[4] D. G. Corneil and U. Rotics. On the relationship between clique-width and tree-width, *SSIAM J. Comput.*, 34 (2005), 825-847.

[5] B. Courcelle. The expression of graph properties and graph transformations in monadic second order logic, in: *G. Rozenberg (ed) Handbook of Graph Grammars and Computing by Graph Transformations, World Scientific*, 1997, pp. 313-400.

[6] B. Courcelle and S. Olariu. Upper bounds to the clique-width of graphs, *Discrete Applied Mathematics*, 101(2000), 77-114.

[7] B. Courcelle and S. Oum. Vertex-minors, monadic second-order logic and a conjecture by Seese, *J. Comb. Theory B*, 97(2006), 91-126.

[8] R. Downey and M. Fellows. Parameterized complexity, *Springer-verlag*, 1999.

[9] M. Fellows, F. A. Rosamond, U. Rotics and S. Szeider. Clique-width minimization is NP-Hard, in *Proceedings of STOC'06: 38th ACM Symposium on Theory of Computing* (Seattle, 2006), pages 354-362. ACM Press, 2006.

[10] S. Oum. Rank-width and vertex-minors, *J. Combin. Theory, Ser B*, 95(2006), 79-100.

[11] S. Oum. Approximating rank-width and Clique-width Quickly, in *Graph-Theoretic Concepts in Computer Science* (Metz, 2005), volume 3787 of *LNCS*, pages 49-58. Springer, 2005.

[12] S. Oum and P. Seymour. Approximating clique-width and branch-width, *J. Combin Theory, Ser B*, 96(2006), 514-528.

[13] S. Oum. Rank-width is less than or equal to branch-width, Submitted, 2006.

[14] N. Robertson and P. Seymour. Graph minors V : excluding a planar graph, *J. Combin. Theory (B)*, 41(1986), 92-114.

[15] N. Robertson and P. Seymour. Graphs minors X : Obstructions to tree-decompositions, *J. Combin. Theory (B)*, 52(1991), 153-190.

[16] N. Robertson and P. Seymour. Graphs minors XIII : The Disjoint Paths Problem, *J. Combin. Theory (B)*, 63(1995), 65-110.

[17] N. Robertson and P. Seymour. Graph minors XX : Wagner's conjecture, *J. Combin. Theory (B)*, 92(2004), 325-357.

[18] D. Seese. Tree-partite graphs and the complexity of algorithms, in *Proc. Int. Conf. on Fundamentals of Computation Theory*, volume 199 of *LNCS*, pages 412-421. Springer, 1985.