2000

Upper Bounds to the Clique Width of Graphs

Bruno Courcelle

Stephan Olariu
Old Dominion University

Original Publication Citation

Courcelle, B., & Olariu, S. (2000). Upper bounds to the clique width of graphs. *Discrete Applied Mathematics, 101*(1-3), 77-114. doi:10.1016/s0166-218x(99)00184-5

ELSEVIER

Discrete Applied Mathematics 101 (2000) 77–114

DISCRETE APPLIED MATHEMATICS

Upper bounds to the clique width of graphs ☆

Bruno Courcelle[a,1], Stephan Olariu[b,*]

[a] *LaBRI (CNRS Laboratory UMR 5800), Université Bordeaux I, Talence 33405, France*
[b] *Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162, USA*

Received 31 July 1997; revised 7 April 1999; accepted 26 April 1999

Abstract

Hierarchical decompositions of graphs are interesting for algorithmic purposes. Many NP complete problems have linear complexity on graphs with tree-decompositions of bounded width. We investigate alternate hierarchical decompositions that apply to wider classes of graphs and still enjoy good algorithmic properties. These decompositions are motivated and inspired by the study of vertex-replacement context-free graph grammars. The complexity measure of graphs associated with these decompositions is called *clique width*. In this paper we bound the clique width of a graph in terms of its tree width on the one hand, and of the clique width of its edge complement on the other.

Keywords: Hierarchical graph decompositions; Modular decomposition; Tree decompositions; Algorithms; Monadic second-order logic

1. Introduction

We investigate a hierarchical graph decomposition that refines the well-known modular decomposition. A graph complexity measure that we call *clique width* is associated in a natural way with this graph decomposition, much the same way *tree width* is associated with tree decompositions (which are actually hierarchical decompositions of graphs [23]).

Hierarchical graph decompositions are interesting for algorithmic purposes. In the following, by a *decomposition* of a graph, we mean either a tree decomposition, or the unique modular decomposition, or a decomposition of the type that we shall define below. A decomposition of a graph G can be viewed as a finite term, written with appropriate operations on graphs, that evaluates to G. Tree decompositions [23] and

☆ Work supported in part by NSF grant CCR-9522093 and by ONR grant N00014-97-1-0526.
* Corresponding author.
E-mail addresses: courcell@labri.u-bordeaux.fr (B. Courcelle); olariu@cs.odu.edu (S. Olariu)
[1] `http://dept-info.labri.u-bordeaux.fr/~courcell/ActSci.html`.

0166-218X/00/$ - see front matter
PII: S0166-218X(99)00184-5

modular decompositions are usually not defined in such an algebraic way, but they can be; see [2] for the case of tree decompositions and [11] for modular decomposition. Infinitely many operations are necessary to define all graphs. By limiting the operations in terms of some integer parameter k, one obtains complexity measures of graphs. A graph G has complexity at most k if it has a decomposition defined in terms of operations limited by this number k. (Typically the complexity of a graph is at most the number of its vertices, but there are infinitely many graphs of a fixed complexity).

Tree width is associated in this way with tree decompositions [23]. In the case of modular decomposition, the corresponding width of a graph, let us call it the *modular width*, is the largest number of vertices of a prime graph appearing at some node of the decomposition. As it turns out, the prime graphs which form the basic blocks of the modular decomposition are no longer basic with respect to the decomposition we shall introduce. It is well known that many NP-complete problems have linear algorithms on graphs of tree width or of modular width bounded by some fixed k, and the same will hold for graphs of clique width at most k. (See the special issue of Discrete Applied Mathematics [21] devoted to partial k-trees).

The graph operations upon which clique width and the related decompositions are based have been already introduced in Courcelle et al. [4,13] in relation with the description of certain context-free graph grammars in terms of systems of mutually recursive equations. These operations build graphs as gluings of complete bipartite graphs. We call *clique width* (*cwd*, for short) the corresponding graph complexity measure.

We now discuss another motivation for investigating clique width, coming from research relating logic and graph theory. We first recall that a graph can be represented by a logical structure so that logical languages can be used to express graph properties. Monadic second-order logic (namely, the extension of first-order logic with set quantifications) is of special interest: most of the NP-complete problems which are linear on hierarchically decomposed graphs correspond to graph properties expressible in MS (Monadic Second-Order logic) [1,15]; yet another reason is that many classes of graphs have decidable monadic theories [5,6].

There are actually two main ways to represent a graph by a logical structure: the domain of this structure may consist of vertices or of vertices and edges. In the latter case, quantified variables of MS formulas may denote sets of edges. We shall refer to MS logic with vertex and edge quantifications by the notation MS_2 and to MS logic with vertex quantifications only by the notation MS_1.

Seese [24] proved that if a set of finite graphs has a decidable MS_2 theory, then it has bounded tree width. He conjectured that if a set L of graphs has a decidable MS_1-theory (which is a weaker condition) then it is "interpretable in a set of trees". This condition is equivalent by results in [9,12,13] to saying that L has bounded clique width.

The result concerning MS_2 uses a result of Robertson and Seymour [23] asserting that graphs of large tree width contain large square grids as minors. At present, we lack a similar structural characterization of graphs with large clique width that would

allow us to establish the conjecture. We hope that the investigation of clique width will yield such a result and will thus make it possible to prove the conjecture.

The paper is organized as follows. Section 2 recalls basic definitions, reviews relations with graph grammars and monadic second-order logic, and introduces a kind of "normal form" as a tool for further proofs. Section 3 contains a characterization of *cwd* in terms of a sequence (and not a tree) of basic operations. It follows that going from a graph to an induced subgraph does not increase *cwd*. Section 4 shows that going from a graph to its edge complement increases *cwd* by a factor of 2, at most. In Section 5, we bound the *cwd* of a graph in term of its tree width. Section 6 summarizes the results and offers a number of open problems.

2. Basics

The set of subsets of a set C is denoted by $P(C)$. All the graphs in this work are finite with no self-loops nor multiple edges. They are directed or not. A graph is a pair $G=(V,E)$ where $E\subseteq V\times V$ if G is directed, while E is a set of unordered pairs of vertices if G is undirected. If necessary, we write $V=V(G)$ and $E=E(G)$. A directed edge will be called an *arc*. The empty graph is such that $V=E=\emptyset$. As usual, we let $|A|$ denote the cardinality of set A. An *abstract* graph is an isomorphism class of a graph.

2.1. Definitions

We now recall the notion of tree width [23]. A *tree decomposition* of a graph $G=(V,E)$ is a pair (T,f) such that T is a tree, f is a mapping associating with every vertex t in $V(T)$ a subset $f(t)$ of V such that
(1) V is the union of the sets $f(t)$, $t\in V(T)$,
(2) any two adjacent vertices of G belong to some set $f(t)$,
(3) for any two vertices t and t' of $V(T)$, $t\neq t'$, if $v\in f(t)\cap f(t')$ then v belongs to $f(w)$ for every w on the unique path in T from t to t'.

The *width* of (T,f) is defined as

$$\max\{|f(t)|\,|\,t\in V(T)\}-1.$$

The *tree width* of G denoted by $twd(G)$ is the minimal width of its tree decompositions.

All trees have a tree width of 1. A clique K_n has *twd* $n-1$ (because for any tree decomposition (T,f) of K_n, some set $f(t)$ must contain all vertices.

Tree width is independent of the directions of edges.

One can also observe that $|E|<k|V|$ (resp. $|E|<2k|V|$) for an undirected (resp. directed) graph $G=(V,E)$ of tree width at most k.

We denote by $TWD(\leqslant k)$ (resp. $T\vec{W}D(\leqslant k)$) the set of undirected graphs (resp. directed graphs) of tree width at most k.

We now introduce the notion of clique width.

Let $\mathscr{C}$ be a countable set of labels. A *labeled* graph is a pair (G,γ) where γ maps $V(G)$ into $\mathscr{C}$. A labeled graph can also be defined as a triple $G=(V,E,\gamma)$, and its labeling function is denoted by $\gamma(G)$ whenever the relevant graph G must be specified. We say that G is *C-labeled* if C is finite and $\gamma(G)(V)\subseteq C$. We denote by $\mathscr{G}(C)$ and by $\vec{\mathscr{G}}(C)$ the sets of undirected C-labeled graphs and of directed C-labeled graphs, respectively. A vertex with label a will be called an *a-port*.

We introduce the following symbols:

- a nullary symbol a for every $a\in\mathscr{C}$;
- a unary symbol $\rho_{a\to b}$ for $a,b\in\mathscr{C}$ with $a\neq b$;
- a unary symbol $\eta_{a,b}$ for $a,b\in\mathscr{C}$ with $a\neq b$;
- a unary symbol $\alpha_{a,b}$ for $a,b\in\mathscr{C}$ with $a\neq b$;
- a binary symbol $\oplus$.

These symbols are intended to denote operations on graphs: $\rho_{a\to b}$ "renames" a as b, $\eta_{a,b}$ "creates edges", $\alpha_{a,b}$ "creates arcs", $\oplus$ is the disjoint union.

For $C\subseteq\mathscr{C}$ we denote by $T(C)$ the set of finite well-formed terms written with the symbols $\oplus,a,\rho_{a\to b},\eta_{a,b}$, for $a,b\in C$, $a\neq b$. We denote by $\vec{T}(C)$ the set of those written with the symbols $\oplus,a,\rho_{a\to b},\alpha_{a,b}$, for $a,b\in\mathscr{C}$, $a\neq b$. Each term in $T(C)$ (resp. $\vec{T}(C)$) denotes a set of labeled undirected (resp. directed) graphs. Since any two graphs denoted by the same term t are isomorphic, one can also consider that t defines a unique abstract graph.

The definitions below are given by induction on the structure of t. We let $val(t)$ be the set of graphs denoted by t (or the corresponding abstract graph.).

If $t\in T(C)$ we have the following cases:

(1) $t=a\in C$: $val(t)$ is the set of graphs with a single vertex labeled by a;

(2) $t=t_1\oplus t_2$: $val(t)$ is the set of graphs $G=G_1\cup G_2$ where G_1 and G_2 are disjoint and $G_1\in val(t_1)$, $G_2\in val(t_2)$;

(3) $t=\rho_{a\to b}(t')$: $val(t)=\{\rho_{a\to b}(G)\mid G\in val(t')\}$ where for every graph G in $val(t')$, the graph $\rho_{a\to b}(G)$ is obtained by replacing in G every vertex label a by b;

(4) $t=\eta_{a,b}(t')$: $val(t)=\{\eta_{a,b}(G)\mid G\in val(t')\}$ where for every undirected labeled graph $G=(V,E,\gamma)$ in $val(t')$, we let $\eta_{a,b}(G)=(V,E',\gamma)$ such that

$$E'=E\cup\{\{x,y\}\mid x,y\in V,x\neq y,\gamma(x)=a,\gamma(y)=b\};$$

hence, we add to G all edges joining an a-port and a b-port that are not adjacent. For terms in $\vec{T}(C)$, we use the same rules with (4) replaced by

(4′) $t=\alpha_{a,b}(t')$: $val(t)=\{\alpha_{a,b}(G)\mid G\in val(t')\}$ where for every directed labeled graph $G=(V,E,\gamma)$ we let $\alpha_{a,b}(G)=(V,E',\gamma)$ such that

$$E'=E\cup\{(x,y)\mid x,y\in V,x\neq y,\gamma(x)=a,\gamma(y)=b\}.$$

Here, we add to G all arcs from an a-port x to a b-port y, such that there does not exist an arc from x to y.

Fig. 1. Examples of graphs with $cwd = 2$.

Fig. 2. Examples of graphs with $cwd = 3$.

For every labeled graph G we let

$$cwd(G) = \min\{|C| \mid G \in val(t), t \in T(C)\}$$

if G is undirected. If G is directed, we let

$$cwd(G) = \min\{|C| \mid G \in val(t), t \in \vec{T}(C)\}.$$

A term $t \in T(C) \cup \vec{T}(C)$ such that $|C| = cwd(G)$ and $G = val(t)$ is called optimal *expression of* G. Unlabeled graphs are considered as graphs all of whose vertices have the same label.

Fact 2.1. *For every graph* G, *we have* $cwd(G) \leqslant |V(G)|$.

Example 2.2.

1. All cliques $K_n, n \geqslant 2$, and all acyclic tournaments of order n, for $n \geqslant 2$, have $cwd(K_n) = 2$; for other graphs with $cwd = 2$ the reader is referred to Fig. 1.
2. The undirected cycles C_n, for $n = 5$ and $n = 6$ have $cwd(C_n) = 3$: additional graphs with $cwd = 3$ are featured in Fig. 2 below. The undirected cycles C_n with $n \geqslant 7$ have clique width equal to 4.
   The cycle C_6 (with all vertices labeled by a) is defined by the following term:

$$\rho_{b\to a}(\rho_{c\to a}(\eta_{b,c}(c \oplus (\rho_{c\to a}(\eta_{a,c}(c \oplus (\eta_{a,b}(a \oplus b) \oplus \eta_{a,b}(a \oplus b)))))))).$$

Fact 2.3. *If* G *is an undirected graph obtained from a directed graph* G' *by omitting the directions of the edges then* $cwd(G) \leqslant cwd(G')$. (*It suffices to replace* $\alpha_{a,b}$ *by* $\eta_{a,b}$ *in any expression denoting* G' *to obtain one denoting* G.) *The inequality can be strict as our examples show.*

Example 2.4.

1. All trees (with undirected edges) are of *cwd* at most 3.
2. The cographs [3] are the undirected (simple, loop-free) graphs of *cwd* at most 2. We recall that cographs are generated from isolated vertices by two binary operations: the disjoint union discussed above and the product ($G \times H$ is obtained from the disjoint union of G and H by the addition of all edges between the vertices of G and those of H.) From this definition, the characterization in term of *cwd* is easy to obtain. (Recall that the cographs are exactly the graphs without induced paths on

four vertices, but this "negative property" does not help here).All trees have tree width equal to 1, all cliques have clique width equal to 2. Our terminology (clique width) is intended to recall this fact.

3. The terms of the form

$$\eta_{a,b}(a \oplus a \oplus \cdots \oplus a \oplus b \oplus b \oplus \cdots \oplus b)$$

define complete bipartite graphs. Together with cliques, this yields examples of graphs of bounded *cwd* having a number of edges not linearly bounded in terms of the number of vertices. This should be contrasted with the case of graphs of bounded tree width.

We note here that a notion similar to our clique width, along with some algorithmic applications, is defined by Wanke in [26].

Notation.

$CWD(k)$=the set of undirected labeled graphs G with $cwd(G) = k$,
$CWD(\leqslant k)$=the set of undirected labeled graphs G with $cwd(G) \leqslant k$,
$CWD(> k)$=the set of undirected labeled graphs G with $cwd(G) > k$,
$C\vec{W}D(k)$, $C\vec{W}D(\leqslant k)$, $C\vec{W}D(> k)$ denote the similar sets of directed graphs.

2.2. Relationships with graph grammars

Several notions of context-free grammars generating sets of graphs or hypergraphs have been defined. One of them, based on the replacement of vertices by graphs that are right-hand sides of rules is discussed in detail by Engelfriet and Rozenberg [20].

We call a *VR set* a set of graphs defined by such a grammar (VR stands for Vertex Replacement). These sets can be defined equivalently (see [13]) as components of least solutions of systems of equations.

As an example of an equation one can consider

$$u = a \cup \rho_{b \to a}(\eta_{a,b}(u \oplus b)).$$

Its least solution (in the set of sets of undirected labeled graphs) is the set of all cliques K_n, all vertices of which are a-ports. We will use the following characterization.

Proposition 2.5. *A set of directed* (*undirected*) *graphs L is VR if and only if it is a component of the least solution of a system of equations written with set union and the symbols $\oplus, a, \rho_{a\to b}$, and $\alpha_{a,b}$ (resp. $\eta_{a,b}$) for $a, b \in \mathscr{C}$, $a \neq b$. There exists a finite set $C \subseteq \mathscr{C}$ such that every graph in L is the value of a term in $\vec{T}(C)$ (resp. $T(C)$).*

The set C is nothing but the finite set of labels (from $\mathscr{C}$) occurring in the operations used in the system which defines L.

Corollary 2.6. *Every VR set of graphs is included in $CWD(\leqslant k)$ or in $C\vec{W}D(\leqslant k)$ for some k.*

Proposition 2.7. *For every k, the sets $CWD(\leqslant k)$ and $C\vec{W}D(\leqslant k)$ are VR.*

Question 2.8. *Are the sets $CWD(k)$ and $C\vec{W}D(k)$ VR?*

2.3. Relationships with monadic second-order logic

We refer the reader to Courcelle [4–11] for the definition of monadic second-order logic (MS) in its two variants MS_1 and MS_2. We only recall here that MS_1 is monadic second-order logic with quantified variables denoting vertices and sets of vertices only, whereas MS_2 is the more powerful language where quantified variables can also denote edges and sets of edges. The reader who does not know MS logic can skip this subsection; the results in this paper will not use it.

We nevertheless recall that a set of graphs is *MS_i-definable* $(i = 1, 2)$, if and only if it is the set of graphs which satisfy a MS_i-formula. A set L of graphs has a *decidable MS_i-theory* if and only if there exists an algorithm that decides whether a given MS_i-formula is satisfied by every graph in L. The following results are proved in [4].

Proposition 2.9. *If L is VR and K is MS_1-definable then $L \cap K$ is VR.*

Corollary 2.10. *The MS_1-theory of a VR set of graphs is decidable.*

Proof. Given an MS_1 formula φ, one lets K be the set of graphs that do not satisfy φ. One can construct a system of equations (i.e. a VR grammar), that defines $L \cap K$. One can test whether $L \cap K = \emptyset$. And $L \cap K = \emptyset$ if and only if φ belongs to the MS_1-theory of L (i.e. if φ holds in every graph in L). □

The following conjecture by Seese [24] can be stated as follows by [12,19].

Conjecture 2.11. *If a set of graphs has a decidable MS_1-theory, then it is included in $CWD(\leqslant k)$ (or $C\vec{W}D(\leqslant k)$) for some k.*

A related result is known from Seese [24].

Theorem 2.12. *If the MS_2-theory of L is decidable then $L \subseteq TWD(\leqslant k)$ or $L \subseteq T\vec{W}D(\leqslant k)$ for some k.*

The proof of Theorem 2.12 sketched in Courcelle [9] uses the deep result of Robertson and Seymour [23] that for every $k \in \mathcal{N}$, every graph of tree width more than $f(k)$ contains a $k \times k$-grid as a minor, where f is some function. In order to prove Conjecture 2.11, one would need a similar result, telling us something about the structure of graphs of "large" clique width. Special cases of Conjecture 2.11 are known to hold.

Theorem 2.13 (Courcelle [9]). *If L is a set of graphs having a decidable MS_1-theory and if it satisfies one of the following conditions* 1–6, *then it is included in $CWD(\leqslant k)$ or $C\vec{W}D(\leqslant k)$ for some k*:

1. *the graphs in L have bounded degree*,
2. *the graphs in L are planar*,
3. *the graphs in L do not contain a fixed graph as a minor*,
4. *L is the set of all orientations of the graphs of a set of undirected graphs*,
5. *L contains all subgraphs of its members*,
6. *L is a set of (undirected) chordal graphs such that each vertex belongs to a bounded number of maximal cliques*.

In cases (1)–(5), one can even prove that L has bounded tree width, whence it follows that L has bounded *cwd* (see Section 5).

From the lemmas making it possible to prove Proposition 2.9 and Corollary 2.10, we get the following result (see [4]).

Theorem 2.14. *For every MS_1 formula ϕ, for every finite set C of labels, one can construct an algorithm that decides, for every term t in $T(C) \cup \vec{T}(C)$, in time $O(|t|)$ whether the graph $val(t)$ satisfies formula ϕ.*

Note that the algorithm is linear in $|T|$, not in the size of the graph $val(t)$. In order to obtain a polynomial algorithm in $|G|$ for $G \in CWD(\leqslant k) \cup C\vec{W}D(\leqslant k)$, we need a polynomial algorithm to "parse" G, i.e. to construct t from G (such that $G \in val(t)$). Such an algorithm exists in special cases: cographs [3], P_4-sparse graphs [14]. Furthermore, Theorem 2.14 extends to optimization problems specified by MS_1-formulas (see [1,14,15]).

2.4. Auxiliary graph operations

In this subsection, we introduce a graph operation that deletes edges, whence the possibility of denoting graphs by more complex terms. However, as it turns out, this new operation can be eliminated from terms, without adding new labels. It follows that the corresponding complexity measure is still the clique width.

For every $a, b \in \mathscr{C}$ with $a \neq b$, we let $\eta^-_{a,b}$ be the unary operation that deletes all edges of the argument graph with one endpoint labeled by a and the other by b. Formally, $\eta^-_{a,b}(G)$ is the graph G' such that:

- $V(G') = V(G)$;
- $E(G') = E(G) - \{\{x, y\} \in E(G) \mid \{\gamma(x), \gamma(y)\} = \{a, b\}\}$.

Proposition 2.15. *For every $a, b, c, d \in \mathscr{C}$ with $a \neq b, c \neq d$*:

(1) $\eta^-_{a,b}(G) = G$, *whenever G has only one vertex*;
(2) $\eta^-_{a,b}(G \oplus G') = \eta^-_{a,b}(G) \oplus \eta^-_{a,b}(G')$, *whenever G and G' are disjoint*;
(3) $\eta^-_{a,b}(\eta_{a,b}(G)) = \eta^-_{a,b}(G)$;

(3′) $\eta^-_{a,b}(\eta_{c,d}(G)) = \eta_{c,d}(\eta^-_{a,b}(G))$ *if* $\{a,b\} \neq \{c,d\}$;
(4) $\eta^-_{a,b}(\rho_{a\to c}(G)) = \rho_{a\to c}(G)$ *if* $a \neq c$ (*we allow* $b = c$);
(4′) $\eta^-_{a,b}(\rho_{c\to a}(G)) = \rho_{c\to a}(\eta^-_{a,b}(\eta^-_{b,c}(G)))$ *if* $c \neq a$ *and* $c \neq b$;
(4″) $\eta^-_{a,b}(\rho_{c\to d}(G)) = \rho_{c\to d}(\eta^-_{a,b}(G))$ *if* $\{a,b\} \cap \{c,d\} = \emptyset$.

Proof. Straightforward verification from the definitions. □

We let $T^+(\mathscr{C})$ denote the set of finite terms constructed with the operations $\oplus$, $\eta_{a,b}$, $\eta^-_{a,b}$, $\rho_{a\to b}$ and the nullary symbols a, for all a, b in $\mathscr{C}$, with $b \neq a$. Two terms are said to be equivalent if they denote the same sets of concrete graphs (or, equivalently, the same abstract graph).

Corollary 2.16. *Every term in* $T^+(C)$ *can be transformed into an equivalent one in* $T(C)$.

Proof. We first prove the result for terms of the particular form $E(t)$ where $t \in T(C)$ and E is a sequence of operations of the form $\eta^-_{a,b}$.

We let $N(E(t))$ be the term in $T(C)$, intended to be equivalent to $E(t)$, and defined by induction on the structure of t as follows:
(1) $N(E(t)) = t$ if $t \in C$ (i.e. t denotes a graph with a single vertex);
(2) $N(E(t_1 \oplus t_2)) = N(E(t_1)) \oplus N(E(t_2))$;
(3) $N(E(\eta_{a,b}(t_1))) = N(E(t_1))$ if $\eta^-_{a,b}$ occurs in E;
(3′) $N(E(\eta_{a,b}(t_1))) = \eta_{a,b}(N(E(t_1)))$ if $\eta^-_{a,b}$ does not occur in E;
(4) $N(E(\rho_{a\to b}(t_1))) = \rho_{a\to b}(N(E'(t_1)))$ where E' is obtained from E as follows:
- one deletes all operations of the form $\eta^-_{a,c}$ for $c \in C$ (note that $\eta^-_{a,c}$ and $\eta^-_{c,a}$ are just two notations for the same operation),
- one adds all operations of the form $\eta^-_{a,c}$ for c such that $\eta^-_{b,c}$ belongs to E.

Hence, for example

$$N(\eta^-_{a,d}(\eta^-_{c,d}(\eta^-_{b,d}(\eta^-_{e,b}(\rho_{a\to b}(t)))))) = \rho_{a\to b}(N(\eta^-_{c,d}(\eta^-_{b,d}(\eta^-_{a,d}(\eta^-_{e,b}(\eta^-_{e,a}(t))))))).$$

One can prove by induction on the structure of t that:

(a) $N(E(t))$ is well-defined for every $t \in T(C)$ and every sequence E, and
(b) $N(E(t))$ is equivalent to $E(t)$.

(The proof of (b) relies on properties of the operation $\eta^-_{a,b}$ stated in Proposition 2.15.)

We next extend N to arbitrary terms in $T^+(C)$ as follows:
- $N(t) = t$ if t is a nullary symbol;
- $N(t_1 \oplus t_2) = N(t_1) \oplus N(t_2)$;
- $N(f(t_1)) = f(N(t_1))$ if f is $\eta_{a,b}$ or $\rho_{a\to b}$;
- $N(\eta^-_{a,b}(t_1)) = N(\eta^-_{a,b}(N(t_1)))$.

Note that the last case uses two "calls" to N; by assuming inductively that the internal one, that computes $N(t_1)$, terminates (so that $N(t_1) \in T(C)$), we conclude that $N(\eta^-_{a,b}(N(t_1)))$ is well-defined by the first part of the proof.

One proves also by the same induction that $N(t)$ is equivalent to t for every $t \in T^+(C)$.

Remark. Corollary 2.16 guarantees that the alternative complexity measure cwd' on graphs in $\mathscr{G}(\mathscr{C})$ defined by $cwd'(G) = \min\{|C'| \mid G \in val(t), t \in T^+(C')\}$ is the same as $cwd(G)$.

A term t in $T(C)$ is said to be *irredundant* if for every subterm of t of the form $\eta_{a,b}(t')$, no a-port of $val(t')$ is adjacent to a b-port.

Corollary 2.17. *Every term in $T(C)$ is equivalent to some irredundant term in $T(C)$.*

Proof. Let $t \in T(C)$. We let $I(t)$ be the term in $T(C)$ constructed by induction on the structure of t as follows:

$I(t) = t$ if $t \in C$;
$I(t_1 \oplus t_2) = I(t_1) \oplus I(t_2)$;
$I(\rho_{a\to b}(t)) = \rho_{a\to b}(I(t))$;
$I(\eta_{a,b}(t)) = \eta_{a,b}(N(\eta^-_{a,b}(I(t))))$, where N is as in Corollary 2.16.
It is clear that $I(t)$ is well-defined and belongs to $T(C)$ for every $t \in T(C)$.

Claim 2.18.
1. *$I(t)$ is irredundant*;
2. *$I(t)$ is equivalent to t.*

Both assertions can be proved by induction on the structure of t, by using Corollary 2.16. The only fact to verify is the following one stated in the notation of this corollary:

Claim 2.19. *If $t \in T(C)$ is irredundant and E is an arbitrary sequence of operations of the form $\eta^-_{a,b}$ then $N(E(t))$ is irredundant.*

Proof. We use the same induction as in the definition of N, in the proof of Corollary 2.16. All cases are trivial except possibly case $(3')$:

$$N(E(\eta_{a,b}(t_1))) = \eta_{a,b}(N(E(t_1))).$$

But we assume here that $\eta_{a,b}(t_1)$ is irredundant. Hence no a-port of $val(t_1)$ is adjacent to a b-port. The same holds a fortiori in $val(N(E(t_1))) = val(E(t_1))$ since this graph has no more edges than $val(t_1)$. Hence, together with the induction hypothesis, saying that $N(E(t))$ is irredundant, we get that $\eta_{a,b}(N(E(t_1)))$ is also irredundant, as desired. □

For directed graphs, we have an operation similar to $\eta^-_{a,b}$ and defined as follows:

$\alpha^-_{a,b}(G) = G'$ whenever $V(G') = V(G)$ and
$E(G') = E(G) - \{(x, y) \in E(G) \mid (\gamma(x), \gamma(y)) = (a, b)\}$.

The results of Propositions 2.15–2.17 extend in an obvious way. (Note that $\alpha^{-}_{a,b}$ and $\alpha^{-}_{b,a}$ are not the same operation.)

3. Two characterizations of clique width

Let C be a finite subset of $\mathscr{C}$ and let G, G' be graphs in $\mathscr{G}(C)$. We write $G \to G'$ whenever $V(G') = V(G)$ and one of the following conditions holds:
1. $G = G_1 \oplus G_2$, $G' = G_1 \oplus \rho_{a\to b}(G_2)$ or
2. $G = G_1 \oplus G_2$, $G' = G_1 \oplus \eta_{a,b}(G_2)$

for some graphs G_1 and G_2 and for some labels $a, b \in C$ with $b \neq a$.

We write $G \to_i G'$ if $G \to G'$ and, whenever condition (2) applies, no a-port of G_2 is adjacent to a b-port.

A *C-construction* of G is a sequence $(G_0, G_1, \ldots, G_n)$ of graphs in $\mathscr{G}(C)$ such that
1. G_0 has no edge;
2. $G_j \to G_{j+1}$ for every $j = 0, 1, \ldots, n-1$;
3. $G_n = G$.

Clearly, $V(G_j) = V(G)$ for every $j = 0, \ldots, n$. Such a construction is *irredundant* if $G_j \to_i G_{j+1}$ for every $j = 0, \ldots, n-1$. Similar definitions can be given for directed graphs: one simply uses $\alpha_{a,b}$ instead of $\eta_{a,b}$, $\vec{\mathscr{G}}(C)$ instead of $\mathscr{G}(C)$, etc.

Proposition 3.1. *For every $k \in \mathcal{N}$ and for every undirected (resp. directed) labeled graph G, the following conditions are equivalent*:
1. *G has clique width at most k*;
2. *G has a C-construction for some set C of cardinality k*;
3. *G has an irredundant C-construction for some set C of cardinality k.*

Proof. We will only consider undirected graphs. The case of directed graphs is similar. We will prove the following implications: $(1) \Rightarrow (3) \Rightarrow (2) \Rightarrow (1)$. The implication $(3) \Rightarrow (2)$ is trivial.

To prove the implication $(1) \Rightarrow (3)$, let $G \in val(t)$ for some irredundant term t in $T(C)$. We define an irredundant C-construction of G by induction on the structure of t.

If $t = a \in C$ then (G) is a C-construction of G.

If $t = \rho_{a\to b}(t')$ or if $t = \eta_{a,b}(t')$, then we let $(G_0, G_1, \ldots, G_n)$ be an irredundant C-construction of $val(t')$ and then, $(G_0, G_1, \ldots, G_n, G)$ is an irredundant C-construction of G.

If $t = t' \oplus t''$ then $G = G' \oplus G''$ where $G' \in val(t')$, $G'' \in val(t'')$ and G', G'' are disjoint. We let $(G'_0, \ldots, G'_n)$ be an irredundant C-construction of G', we let $(G''_0, \ldots, G''_m)$ be an irredundant C-construction of G''. Then the sequence

$$(G'_0 \oplus G''_0, G'_1 \oplus G''_0, \ldots, G'_n \oplus G''_0, G'_n \oplus G''_1, G'_n \oplus G''_2, \ldots, G'_n \oplus G''_m)$$

is an irredundant C-construction of $G = G' \oplus G'' = G'_n \oplus G''_m$.

Finally, to prove the implication $(2) \Rightarrow (1)$ let $(G_0, \ldots, G_n)$ be a C-construction of G_n.

Claim 3.2. *Every connected component of G_n is the value of some term in $T(C)$.*

Observe that Claim 3.2 implies the desired result because G_n is the disjoint union of its connected components, say $H_1,\ldots,H_m$. If t_i is a term in $T(C)$ denoting H_i then $t_1 \oplus t_2 \oplus \cdots \oplus t_m$ is a term in $T(C)$ denoting G_n.

Proof of Claim 3.2. We shall proceed by induction on n.

Case $n = 0$: Clear, because the connected components are graphs with one vertex and no edge, denoted by symbols from C.

Case $n > 0$: We consider two subcases.

Subcase 1: $G_{n-1}=H\oplus K, G_n=\rho_{a\to b}(H)\oplus K$. A connected component L of G_n is either a connected component of K, hence of G_{n-1} and the result follows by the induction hypothesis, or a graph of the form $\rho_{a\to b}(H')$ where H' is a connected component of H, hence of G_{n-1}. By the induction hypothesis, $H' = val(t)$ for some term $t \in T(C)$ and the term $\rho_{a\to b}(t)$ denotes L.

Subcase 2: $G_{n-1} = H \oplus K, G_n = \eta_{a,b}(H) \oplus K$. We may assume that H has at least one a-port and one at least b-port (otherwise $G_n = G_{n-1}$ and the results holds by the induction hypothesis).

A connected component L of G_n is:

- either $\eta_{a,b}(H')$ where H' is the union of the connected components of H, say $H_1,\ldots,H_m$, that contain at least one a-port or b-port,
- or a connected component of H without a-port or b-port,
- or a connected component of K.

In the first case the graph L is defined by the term $\eta_{a,b}(t_1\oplus\cdots\oplus t_m)$ where $t_1,\ldots,t_m$ are terms in $T(C)$ denoting respectively $H_1,\ldots,H_m$. They exist by the induction hypothesis. In the last two cases, the result follows from the induction hypothesis as in Subcase 1. This completes the proof of Claim 3.2, whence Proposition 3.1. □

Proposition 3.1 implies that a graph of *cwd* at most k can be "marked" in a way that witnesses this fact: this marking "encodes" on the graph one of its irredundant C-constructions, where C has cardinality at most k.

Let $(G_0, G_1,\ldots, G_n)$ be an irredundant C-construction of $G=G_n$. We define a marking as follows:

1. Every edge created by a step $G_{i-1} \to G_i$ is marked i; since G_0 has no edge, every edge has a mark in $\{1,\ldots,n\}$; since the construction is irredundant, every edge is "created only once" hence has only one mark; the set of edges marked i forms a complete bipartite subgraph of each G_j for $i \leqslant j \leqslant n$.
2. Every vertex is marked a_0 where a is its label in G_0; if at step $G_{i-1} \to G_i$, a vertex gets a new label b, then we add to its marks the new mark b_i; hence, every vertex of G_n has a sequence of marks say $a_0 b_{i_1} c_{i_2} \ldots d_{i_m}$ where $0 < i_1 < i_2 < \cdots < i_m$ recording the history of the vertex: initially marked by a, it got the successive labels b in G_{i_1}, c in $G_{i_2},\ldots$, and finally d in G_{i_m}. The label of this vertex in G_j is

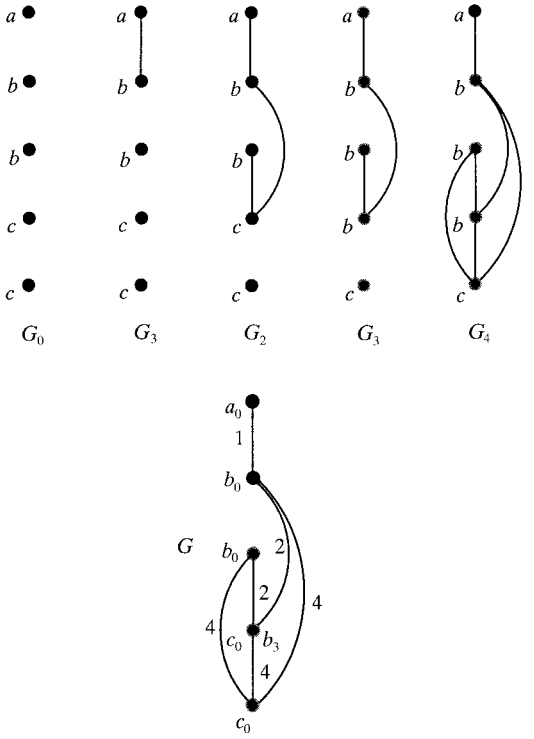


Fig. 3. Illustrating a construction and the corresponding marking.

$f \in C$ if it has mark f_{i_k} for some $k \in \{0, 1, \ldots, m\}$ (where we let $i_0 = 0$) and no mark with subscript j' for $i_k < j' \leqslant j$.

It is clear that each irredundant construction is specified unambiguously by the above described marking of the graph it constructs.

Example. Fig. 3 shows a construction $(G_0, G_1, G_2, G_3, G_4)$ defining the graph $G = G_4$, along with the marking of G recording the steps of this construction. The corresponding term reads

$$\eta_{b,c}(\rho_{c\to b}(\eta_{b,c}(\eta_{a,b}(a \oplus b) \oplus b \oplus c)) \oplus c).$$

Corollary 3.3. *If G is an induced subgraph of H then $cwd(G)$ is at most $cwd(H)$.*

Proof. Let $H_0, H_1, \ldots, H_n$ be a C-construction of H. For each i, let G_i be the induced subgraph of G with set of vertices $V(G)$. Then $G_0, G_1, \ldots, G_n$ is a C-construction of G, except that two consective graphs may be equal. After deletion of repeated elements in the list, we get a C-construction of G. □

These definitions and results extend in a straightforward way to directed graphs. We will improve Corollary 3.3 by showing that the *cwd* of a graph G is the least upper bound of the *cwds* of its prime induced subgraphs. It will follow that an optimal expression of a graph G can be obtained as a refinement of the unique modular decomposition of G. This latter decomposition can be constructed in time $O(|V| + |E|)$, where $G = (V, E)$ [17].

We recall the definition of the substitution of a vertex in a graph by another graph. Let $G = (V, E)$ and $H = (V', E')$ be disjoint directed graphs, and let x be a vertex in G. The graph $G[H \backslash x]$ called the result of the *substitution in G of H for x* is defined as follows. Its set of vertices is $V \cup V' - \{x\}$; its set of edges is $E' \cup (E \cap ((V - \{x\}) \times (V - \{x\}))) \cup \{(y,z) \mid y \in V', z \in V - \{x\}$ and $(x,z) \in E\} \cup \{(z,y) \mid y \in V', z \in V - \{x\}$ and $(z,x) \in E\}$.

This definition extends to undirected graphs by handling E and E' as symmetric relations.

A graph $P = (V, E)$ is *prime* if $|V| \geqslant 2$ and P is not equal to $G[H \backslash x]$ unless G or H is reduced to a single vertex. Examples of prime graphs include P_2, $\vec{P}_2$, $2K_1$, $\vec{P}_3$, P_4, and the graphs C_n for $n \geqslant 5$.

For every set of graphs $\mathscr{C}$ we denote by $\mathscr{F}(\mathscr{C})$ the least set of graphs that contains $\mathscr{C}$ and is closed under isomorphism and substitution. The set of cliques with at least two vertices can thus be characterized as $\mathscr{F}(\{K_2\})$.

Lemma 3.4. *Let G and H be disjoint C-labeled graphs denoted respectively by terms t and s in $T(C)$. Let x be a vertex of G. Let u be the occurrence in T of the nullary symbol a ($a \in C$) that corresponds to x. Let us assume that all vertices of H are labeled by a. Then the term $t[s \backslash u]$ in $T(C)$ obtained from t by substituting s for u denotes the graph $G[H \backslash x]$.*

Proof. Straightforward induction on the structure of the term t. □

Proposition 3.5. *If $\mathscr{C}$ is a set of graphs and $G \in \mathscr{F}(\mathscr{C})$ then $cwd(G) \leqslant \sup(\{cwd(H) \mid H \in \mathscr{C}\})$.*

Proof. Let $G = G'[G'' \backslash x]$ where G', G'', are denoted by t and s, respectively, all vertices of G' are labeled by b and all vertices of G'' are labeled by c, u is an occurrence in t of the nullary symbol a, ($a \in C$), that corresponds to the vertex x.

The term $t[\rho_{c \to a}(s) \backslash u]$ denotes G if $c \neq a$ and the term $t[s \backslash u]$ denotes G if $c = a$. Hence, $cwd(G) \leqslant \max\{cwd(G'), cwd(G'')\}$. The result follows from the definition of $\mathscr{F}(\mathscr{C})$ since isomorphism of graphs preserves *cwd*. □

Every graph has a unique *modular decomposition* that is an algebraic term using the substitution and (isomorphic copies of) its induced subgraphs, see [11,18], and hence belongs to $\mathscr{F}(Prime(G))$, where $\mathscr{F}(Prime(G))$ denotes the set of prime induced subgraphs of G.

Corollary 3.6. *For every graphs* G, $cwd(G) = \max\{cwd(H) \mid H \in Prime(G)\}$.

Proof. The inequality $\geqslant$ follows from Corollary 3.3. The inequality $\leqslant$ follows from Proposition 3.5 and the fact that every graph G belongs to $\mathscr{F}(Prime(G))$. □

The modular decomposition of a graph can be constructed efficiently [17,22]. It follows from Lemma 3.4 that one obtains an optimal expression of a graph by combining optimal expressions of its prime induced subgraphs. The combination is based on the modular decomposition. The technique is used in [14]. It follows that the (open) problem of characterizing the complexity of the problem "$cwd(G) \leqslant k$" reduces to the special case of prime graphs G.

4. Graph transformations that do not increase the clique width too much

For every graph G we denote by $\bar{G}$ its *edge complement*, that is, the graph with $V(\bar{G}) = V(G)$ and $E(\bar{G}) = (V(G) \times V(G)) - E(G) - \{(x,x) \mid x \in V(G)\}$ if G is directed and with $E(\bar{G}) = \mathscr{P}_2(V(G)) - E(G)$ if G is undirected. We denote by $\mathscr{P}_2(V)$ the set of unordered pairs of elements of a set V.

Theorem 4.1. *For every graph* G, *we have* $cwd(\bar{G}) \leqslant 2 * cwd(G)$.

Note that if G has several vertices and no edges, then $cwd(G) = 1$ but $cwd(\bar{G}) = 2$ regardless of whether $\bar{G}$ is undirected or directed. We ask the following question.

Question 4.2. *Does there exist for every value of* n *larger than* 1 *a graph* G *such that* $cwd(G) = n$ *and such that* $cwd(\bar{G}) = 2n$?

We will first consider the case of undirected graphs. We will actually prove a more general form of Theorem 4.1. Let C be a finite subset of $\mathscr{C}$ and let P be a subset of $\mathscr{P}_{\leqslant 2}(C) = C \cup \mathscr{P}_2(C)$. We let K_P be the operation on graphs in $\mathscr{G}(C)$ such that $G' = K_P(G)$ whenever

- $V(G') = V(G)$, and
- $E(G') = (E(G) - \{\{x,y\} \in E(G) \mid \{\gamma(G)(x), \gamma(G)(y)\} \in P\}) \cup \{\{x,y\} \mid \{\gamma(G)(x), \gamma(G)(y)\} \in P, \{x,y\} \notin E(G)\}$.

The following proposition yields Theorem 4.1 for undirected graphs because if we let C be the set of labels of a term denoting G and $P = \mathscr{P}_{\leqslant 2}(C)$ then $\bar{G} = K_P(G)$.

Proposition 4.3. *For every graph* G *in* $\mathscr{G}(C)$, *and for every* $P \subseteq \mathscr{P}_{\leqslant 2}(C)$ *we have* $cwd(K_P(G)) \leqslant 2 * cwd(G)$.

The proof of Proposition 4.3 will use terms in $T(C)$ of a special form to denote graphs.

Definition 4.4 (*Separated terms*). The *type* of a graph G is the set $\gamma(V(G))$ of labels of its vertices. The *type* of a term $t \in T(C)$ is the type of the graph $val(t)$, that we shall denote by $\tau(t)$. Clearly, $\tau(t) \subseteq C$. A term t is said to be *separated* if for every subterm of t of the form $t_1 \oplus t_2$, we have $\tau(t_1) \cap \tau(t_2) = \emptyset$.

Lemma 4.5. *Every term in $T(C)$ is equivalent to a separated term in $T(D)$ for some set D of cardinality at most $2 * |C|$.*

Proof. We let $C' = \{c' \mid c \in C\}$ and $D = C \cup C'$. Notice that $C \cap C' = \emptyset$. We denote by R the operation:

$$R(G) = \rho_{a \to a'}(\rho_{b \to b'}(\ldots(\rho_{d \to d'}(G))\ldots))$$

where $\{a, b, \ldots, d\} = C$, and we denote by R' the operation

$$R'(G) = \rho_{a' \to a}(\rho_{b' \to b}(\ldots(\rho_{d' \to d}(G))\ldots)).$$

We let S be the transformation:

$$T(C) \cup T(C') \to T(C \cup C')$$

defined below by induction on the structure of terms as follows. The inductive definition will use a mapping

$$t \mapsto \tilde{t} : T(C \cup C') \to T(C \cup C')$$

that replaces everywhere in t every label $a \in C$ by the corresponding label $a' \in C'$, and every label $a' \in C'$ by the corresponding label $a \in C$. We let, similarly, $\tilde{G}$ be obtained from G where $G \in \mathscr{G}(C \cup C')$. It is clear that we have

$$val(\tilde{t}) = \widetilde{val(t)}.$$

It should be noted that the mapping $\sim$ applied to a term t modifies the labels used at intermediate steps of the construction it defines.

We now define the mapping S by setting:

- $S(a) = a$ if $a \in C \cup C'$,
- $S(\rho_{a \to b}(t)) = \rho_{a \to b}(S(t))$,
- $S(\rho_{a' \to b'}(t)) = \rho_{a' \to b'}(S(t))$,
- $S(\eta_{a,b}(t)) = \eta_{a,b}S(t))$,
- $S(\eta_{a',b'}(t)) = \eta_{a',b'}S(t))$,
- $S(t_1 \oplus t_2) = R'(S(t_1) \oplus S(\tilde{t}_2))$ if $t_1, t_2 \in T(C)$ (and hence $\tilde{t}_2 \in T(C')$),
- $S(t_1 \oplus t_2) = R(S(t_1) \oplus S(\tilde{t}_2))$ if $t_1, t_2 \in T(C')$ (and hence $\tilde{t}_2 \in T(C)$).

Claim 4.6. *For every $t \in T(C) \cup T(C')$, $S(t)$ is a separated term in $T(C \cup C')$ that is equivalent to t.*

Proof. We prove by induction on the structure of t that for every $t \in T(C) \cup T(C')$:

(1) $S(t)$ is separated,

(2) $S(t)$ is equivalent to t,

(3) $\tau(S(t)) \subseteq C$ if $t \in T(C)$ and $\tau(S(t)) \subseteq C'$ if $t \in T(C')$.
(4) $S(\tilde{t}) = \widetilde{S(t)}$.

(1) The claim that $S(t_1 \oplus t_2)$ is separated is clear from the definition and the inductive hypothesis that (3) holds for t_1 and t_2 and that $S(t_1)$ and $S(\tilde{t}_1)$ are separated. All other cases follow from the inductive hypothesis that $S(t)$ is separated.

(2) The fact that $S(t_1 \oplus t_2)$ is equivalent to $t_1 \oplus t_2$ follows from the inductive hypothesis that $S(t_1)$ is equivalent to t_1 and that $S(t_2)$ is equivalent to t_2 so that, by (4), we also have $S(\tilde{t}_2)$ equivalent to $\tilde{t}_2$, and the remark that for $G_1, G_2 \in \mathscr{G}(C)$ (resp. $\mathscr{G}(C')$)

$$G_1 \oplus G_2 = R'(G_1 \oplus \tilde{G}_2) \quad (\text{resp.} \quad R(G_1 \oplus \tilde{G}_2)).$$

All other cases are straightforward. Properties (3) and(4) are also straightforward to verify by induction. □

Proof of Lemma 4.5. (*Conclusion*). Lemma 4.5 follows then immediately from Claim 4.6 since $|C \cup C'| = 2 * |C|$. □

The next proposition states some properties of the operations K_P. We refer the reader to Section 2 for the definitions of the operation $\eta^-_{a,b}$ and of the set of terms $T^+(C)$.

Proposition 4.7. *Let $C \subseteq \mathscr{C}$ be finite and let $P \subseteq \mathscr{P}_2(C)$.* For every $G, G' \in \mathscr{G}(C)$ *such that G, G' are disjoint and $\tau(G) \cap \tau(G') = \emptyset$, and for every $a, b \in C$ with $a \neq b$:*

1. $K_P(G) = G$ *if G has only one vertex,*
2. $K_P(G \oplus G') = \cdots = (\eta_{c,d}(\ldots(K_P(G) \oplus K_P(G'))\ldots))$ *where the composition extends to all $c \in \tau(G)$ and $d \in \tau(G')$ such that $\{c,d\} \in P$,*
3. $K_P(\rho_{a\to b}(G)) = \rho_{a\to b}(K_{P'}(G))$ *where $P' = \{p \in P \mid a \notin p\} \cup \{\{a,c\} \mid \{b,c\} \in P\}$,*
4. $K_P(\eta_{a,b}(G)) = \eta^-_{a,b}(K_{P'}(G))$ *if $\{a,b\} \in P$ and $P' = P - \{\{a,b\}\}$,*
5. $K_P(\eta_{a,b}(G)) = \eta_{a,b}(K_P(G))$ *if $\{a,b\} \notin P$,*
6. $K_P(\eta^-_{a,b}(G)) = \eta_{a,b}(K_{P'}(G))$ *if $\{a,b\} \in P$ and $P' = P - \{\{a,b\}\}$,*
7. $K_P(\eta^-_{a,b}(G)) = \eta^-_{a,b}(K_P(G))$ *if $\{a,b\} \notin P$.*

Proof. Straightforward verification from the definitions. □

Proof of Proposition 4.3. Let G be undirected and defined by a term t in $T(C)$ where $|C| = cwd(G)$. By Lemma 4.5 one can transform t into an equivalent separated term $u \in T(D)$ with $|D| = 2cwd(G)$.

For every $P \subseteq \mathscr{P}_{\leqslant 2}(D)$, one can construct from u a term $w(u,P)$ in $T^+(D)$ that is equivalent to $K_P(u)$: it is easy to transform the equalities of Proposition 4.7 into a definition of $w(u,P)$ that is inductive on the structure of u.

By Corollary 2.16, one can transform $w(u,P)$ into an equivalent term $\bar{w}(u,P)$ in $T(D)$. Hence $\bar{w}(u,P)$ defines the graph $K_P(G)$, as desired. □

We have already observed that Proposition 4.3 yields Theorem 4.1 for undirected graphs. We now consider the case of directed graphs. The proof is essentially the same. We only indicate the modifications in the definitions.

If $P \subseteq C \times C$ we let $\vec{K}_P$ be the operation on $\vec{\mathcal{G}}(G)$ such that $\vec{K}_P(G) = G'$ whenever
- $V(G') = V(G)$, and
- $E(G') = (E(G) - \{(x,y) \in E(G) \mid (\gamma(x), \gamma(y)) \in P\}) \cup \{(x,y) \in V(G) \times V(G) \mid x \neq y, (\gamma(x), \gamma(y)) \in P \text{ and } (x,y) \notin E(G)\}$.

Proposition 4.8. *For every graph G in $\mathcal{G}(C)$ and for every $P \subseteq C \times C$, we have*

$$cwd(\vec{K}_P(G)) \leqslant 2 * cwd(G)$$

From this proposition follows immediately Theorem 4.1 for directed graphs. Lemma 4.5 works for terms denoting directed graphs as well. In the proof, one uses $\alpha_{a,b}$ instead of $\eta_{a,b}$.

5. Comparison with tree width

In this section, given a graph G we compare its clique width, $cwd(G)$, with its tree width, $twd(G)$. We will prove that for every undirected graph G, $cwd(G) \leqslant 2^{twd(G)+1}+1$ and that for every directed graph G $cwd(G) \leqslant 2^{2twd(G)+2} + 1$

There is no hope to get a relation of the form

$$twd(G) \leqslant f(cwd(G)) \tag{1}$$

valid for all graphs since the complete graphs have unbounded tree width yet their clique width is at most 2. However, we will obtain inequalities of the form (1) for graphs belonging to particular classes, like the class of planar graphs. For our proofs we need some definitions, that we borrow from [4,6].

Definition 5.1 (*Sourced graphs*)**.** Let $k \in \mathcal{N}$, $k > 0$. We let $[k] = \{1,2,3,\ldots,k\}$ and $[0,k] = \{0,1,2,3,\ldots,k\}$. A *k-sourced* graph is a tuple $G = (V,E,s)$ where V,E are as in Section 2 (graphs may be directed or not) and s is an injective map : $[k] \to V$. Vertex $s(i)$ is called a *source* and is referred to as the *i-source* of G. We shall consider such a graph as labeled by $\gamma : V \to [0,k]$ with $\gamma(s(i)) = i$ for $i = 1,\ldots,k$ and $\gamma(v) = 0$ if $v \notin s([k])$. We now define some operations on k-sourced graphs. (They are borrowed from [2].)

5.1. Parallel composition

If G and G' are k-sourced graphs $G = (V,E,s)$ and $G' = (V',E',s')$, we let $H = G//G' = (V'',E'',s'')$ if the following conditions hold:
- $V'' = V \cup V'$,
- $V \cap V' = s([k]) \cap s'([k])$,
- $s = s' = s''$,
- $E'' = E \cup E'$.

We say that H is the *parallel composition* of G and G'. Note that $G//G'$ is not always defined. However, for any two graphs G and G', one can find a graph G'' isomorphic to G' such that $G//G''$ is well-defined. The graphs $G//G''$ defined in this way are all isomorphic.

5.2. Series composition

The following k-ary operation is a generalization of the usual series composition of graphs with two sources. If $G_1,\ldots,G_k$ are k-sourced graphs we let

$$S(G_1,\ldots,G_k) = fg(i_1(G_1)//\ldots//i_k(G_k)//e_1//\ldots//e_k)$$

where fg, i_n, e_n are defined as follows.

- The mapping fg maps $(k+1)$-sourced graphs to k-sourced graphs in such a way that

  $$fg(V,E,s) = (V,E,s')$$

  where s' is the restriction of s to the set $[k]$; in other words one "forgets" the last source, whence the notation.
- The mapping i_n, $1 \leqslant n \leqslant k$, transforms a k-sourced graph into a $(k+1)$-sourced graph by inserting a new isolated vertex at the nth position in the sequence of sources. In other words:

  $$i_n(V,E,s) = (V',E,s')$$

  whenever

  $$\begin{aligned} &V' = V \cup \{s'(n)\}, \quad (s'(n) \text{ is "new"}),\\ &s(i) = s'(i) \quad \text{if } 1 \leqslant i \leqslant n-1\\ &s(i) = s'(i+1) \quad \text{if } n \leqslant i \leqslant k. \end{aligned}$$

- The $(k+1)$-sourced graph e_n, $1 \leqslant n \leqslant k$, has $k+1$ sources, no other vertex and one edge between the nth source and the $(k+1)$th one.

We let I be the graph without edges consisting of k isolated vertices that are the sources numbered arbitrarily from 1 to k. We let K be the k-sourced graph consisting of one edge between any two sources.

We let T and U be the sets of k-sourced graphs defined recursively by

$$\begin{aligned} &T = \{K\}//U,\\ &U = U//U \cup S(U,U,\ldots,U) \cup \{I\}. \end{aligned} \tag{2}$$

Then T is the set of *k-sourced k-trees*. To take a concrete example, with $k=2$ the 2-sourced 2-tree in Fig. 4 below can be described by the expression:

$$K//S(S(I,I),S(I,I))//S(S(S(I,I),I)),I).$$

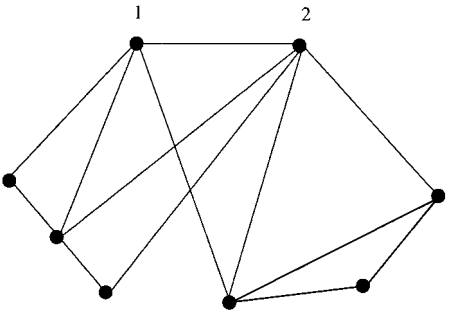


Fig. 4.

The mapping S can be described concretely as follows: $G = S(H,K)$ if and only if G is obtained from the union of two disjoint copies of H and K modified as follows:

- $s_H(2)$ and $s_K(2)$ are fused into a new vertex u which is not a source of G,
- $s_G(1) = s_K(1)$ and this vertex is linked to u by a new edge,
- $s_G(2) = s_H(1)$ and this vertex is linked to u by a new edge.

For every subset P of $[k]$, we let

$$S_P(G_1,\ldots,G_k) = fg(i_1(G_1)//\ldots//i_k(G_k)//e_{j_1}//\ldots//e_{j_n}) \tag{3}$$

where $P = \{j_1,\ldots,j_n\}$. We let also D denote the set of graphs obtained from K by edge deletions. (Hence D has 2^k elements and contains I and K which are the minimal and maximal elements with respect to the number of edges).

We let T', U' be the sets of k-sourced graphs defined recursively by

$$\begin{aligned} &T' = D//U',\\ &U' = U'//U' \cup \bigcup\{S_P(U',U',\ldots,U') \mid P \subseteq [k]\} \cup \{I\}. \end{aligned} \tag{4}$$

The set T' is the set of *k-sourced partial k-trees*.

Lemma 5.2. *An undirected graph G with at least k vertices has tree width at most k iff $(G,s) \in T'$ for some source mapping s.*

Proof. See van Leeuwen [25, Lemma 1.24]. □

We need additional concepts that we introduce next.

Definition 5.3 (*Multiply labeled graphs*)**.** We let C be a finite set of vertex labels. A *multiply labeled graph* (m.l. graph, for short) is a graph given with a labeling mapping δ such that $\delta(v) \subseteq C$ for every vertex v. Each label in $\delta(v)$ is a label of the vertex v. Hence a vertex may have zero, one, or several labels chosen in C.

We will use the operations $\oplus$, the disjoint union, and $\eta_{a,b}$, which creates an edge between x and y if $a \in \delta(x)$ and $b \in \delta(y)$ and if no such edge already exists. We will

also use the following extension of the renaming operation: $G' = \rho_{a\to P}(G)$ is obtained from G by replacing the labeling function δ by δ' such that

$$\begin{aligned} &\delta'(v) = \delta(v) && \text{if } a \notin \delta(v),\\ &\delta'(v) = (\delta(v) - a) \cup P && \text{otherwise.} \end{aligned} \tag{5}$$

We may have $a \in P$ (i.e. new labels are "added" to a) and $P = \emptyset$ (i.e. label a is "deleted"). We will use the notation $\rho_{a\to b,c,\ldots,d}$ for $\rho_{a\to P}$, if $P = \{b, c, \ldots, d\}$ and $\rho_{a\to}$ if $P = \emptyset$. We can thus denote m.l. graphs by finite terms built with $\oplus$, $\eta_{a,b}$, $\rho_{a\to P}$, and the nullary symbols a for $a \in C$. We will denote by $M(C)$ the set of these terms. Clearly, $T(C) \subseteq M(C)$. The m.l. graph (or the set of isomorphic m.l. graphs) defined by $t \in M(C)$ is denoted by $val(t)$. For every m.l. graph G, we let

$$\triangle(G) := \{\delta(v) \mid v \in V(G)\} \subseteq \mathscr{P}(C).$$

For every term t in $M(C)$ we let

$$\triangle(t) := \triangle(val(t))$$

and

$$\triangle^*(t) := \bigcup \{\triangle(t') \mid t' \text{ is a subterm of } t\}.$$

For every m.l. graph G, we let $\ell(G)$ be the corresponding labeled graph where we take $\gamma(G) = \delta(G)$: the vertex labels of $\ell(G)$ are thus elements of $\mathscr{P}(C)$. We will use $\mathscr{P}(C)$ as set of vertex labels.

Proposition 5.4. *For every m.l. graph G defined by $t \in M(C)$ we have*

$$cwd(\ell(G)) \leqslant |\triangle^*(t)|.$$

In particular

$$cwd(G) \leqslant 2^k$$

for every unlabeled graph G defined by $t \in M(C)$ where $k = |C|$.

Proof. We define a transformation $m : M(C) \to T(\mathscr{P}(C))$ by the following induction:

- $m(a) = \{a\}$ (the graph with a single vertex labeled by $\{a\}$),
- $m(t_1 \oplus t_2) = m(t_1) \oplus m(t_2)$,
- $m(\eta_{a,b}(t)) = (\eta_{P_1,P'_1}(\eta_{P_i,P'_j}(\ldots \eta_{P_n,P'_m}(m(t)))\ldots))$
- $m(\rho_{a\to P}(t)) = \rho_{P_1\to Q_1}(\rho_{P_2\to Q_2}(\ldots(\rho_{P_n\to Q_n}(m(t)))\ldots)$

where $P_1, \ldots, P_n$ is an enumeration of the sets of $\triangle(t)$ that contain a, where $P'_1, \ldots, P'_m$ is an enumeration of those containing b, and $Q_i = (P_i - \{a\}) \cup P$ for every $i = 1, \ldots, n$. Note that $\{Q_1, \ldots, Q_n\} \subseteq \triangle(\rho_{a\to P}(t))$. We omit the operation $\rho_{P_i\to Q_i}$ if $Q_i = P_i$.

Claim. *For every $t \in M(C)$ we have*

1. $m(t) \in T(\triangle^*(t))$,
2. $val(m(t)) = \ell(val(t))$.

Proof. Straightforward induction on the structure of t. Assertion (1) uses the fact that

$$\triangle^*(t_1 \oplus t_2) = \triangle^*(t_1) \cup \triangle^*(t_2)$$

and that the sets P_i, P'_j (resp. P_i, Q_i of the definition of m) are in $\triangle^*(\eta_{a,b}(t))$ (resp. in $\triangle^*(\rho_{a\to P}(t))$). □

Proof of Proposition 5.4. (*Conclusion*). Proposition 5.4 follows immediately from the claim. □

Theorem 5.5. (1) *For every undirected graph G we have $cwd(G) \leqslant 2^{twd(G)+1} + 1$.* (2) *For every directed graph G we have $cwd(G) \leqslant 2^{2twd(G)+1} + 1$.*

Proof. (1) Let $k = twd(G)$. If G has $k-1$ vertices then $cwd(G) \leqslant k-1$ and the result holds since $k-1 \leqslant 2^{k+1}+1$ for $k \geqslant 1$. Otherwise, by Lemma 5.2, one can make G into a k-source partial k-tree and we only need prove that

$$cwd(G) \leqslant 2^{k+1} + 1. \tag{6}$$

The k-sourced partial k-trees are of the form $G = H//G'$ where $G' \in U'$, $H \in D$ and U' is defined by the recursive equation (4) that we recall:

$$U' = U'//U' \cup \{I\} \cup \bigcup \{S_P(U', \ldots, U')/P \subseteq [k]\}.$$

It is clear that the graphs in U' have pairwise non-adjacent sources and that G as above satisfies

$$G = \ldots(\eta_{i,j}(\ldots(G')\ldots))$$

where we have one operation $\eta_{i,j}$ for every pair i, j with $1 \leqslant i < j \leqslant k$ such that the i-source and the j-source of H are adjacent. Hence it is enough to prove that the graphs in U' have clique width at most $2^{k+1}+1$.

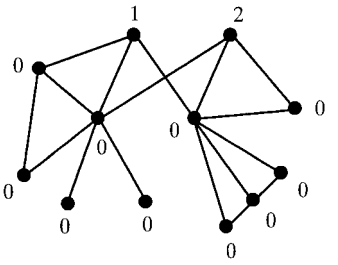
For every graph G in U', we let $Int(G)$ be its *interior*, i.e. the m.l. graph G' constructed as follows:

- $V(G')$ is the set of *internal* (non-source) vertices of G,
- $G' = G[V(G')]$ (the induced subgraph of G with set of vertices $V(G')$),
- $\delta(G')(v) = \{0\} \cup \{i \in [k] / v$ is adjacent to the ith source in $G\}$.

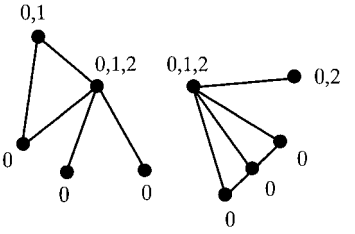
The vertex labels are thus elements of $\{0, 1, \ldots, k\}$. It follows then from this construction that

$$G = \rho_{1'\to 1}(\cdots(\rho_{k'\to k}(\rho_{1\to}(\cdots(\rho_{k\to}(\eta_{1,1'}(\cdots(\eta_{k,k'}(Int(G) \oplus 1' \oplus \cdots \oplus k'))\cdots). \tag{7}$$

The right-hand side of (7) is a term in $M(\{0, 1, \ldots, k, 1', \ldots, k'\})$. This equality is still valid if $G = I$, in which case $Int(G)$ is the empty graph. It is illustrated in Fig. 5 with $k = 2$.



A graph G from U



The graph Int(G)

Fig. 5.

Fact 5.6. *For G, $G' \in U'$, we have $Int(G//G') = Int(G) \oplus Int(G')$.*

Fact 5.7. *For $G_1, \ldots, G_k \in U'$ and $G = S_P(G_1, \ldots, G_k)$ we have*

$$Int(G) = \rho_{*\to P\cup\{0\}}(\rho_{\$\to}(\eta_{\$,*}(R_1(Int(G_1)) \oplus \cdots \oplus R_k(Int(G_k)) \oplus *))\ldots)), \qquad (8)$$

where for every graph H and for every $i = 1, \ldots, k$ we write

$$R_i(H) := \rho_{i\to i+1}(\ldots((\rho_{k-1\to k}(\rho_{k\to\$}(H))))).$$

The operations appearing in (8) use the labels $0, 1, 2, \ldots, k, \$, *$.

Proof of Fact 5.7. The graph $Int(G)$ is the disjoint union of the graphs $Int(G_1), \ldots, Int(G_k)$, augmented with a new vertex, say, v, with edges as described below, and modified by some changes of labels also described below. In a k-sourced partial k-tree G, we say that a vertex v is a *pre-i-source* for $i \in \{1, \ldots, k\}$ if it is not a source but is adjacent to the i-source, i.e. if it has label i in $Int(G)$. The edges in $Int(G)$ that are not in $Int(G_1), \ldots, Int(G_k)$ join v and the pre-k-sources of $G_1, \ldots, G_k$. They are created in $Int(G)$ by the operation $\eta_{\$,*}$ with the help of the operations $\rho_{k\to}$ of R_i $(1 \leqslant i \leqslant k)$.

The new vertex v is a pre-i-source of $Int(G)$ for $1 \leqslant i \leqslant k$ if and only if $i \in P$: the corresponding labeling is done in (7) by $\rho_{*\to P\cup\{0\}}$. Note that the vertex v is created in Eq. (8) with label $*$.

A pre-j-source of G_i for $1 \leqslant j \leqslant k-1$, where $1 \leqslant i \leqslant k$ becomes a pre-j'-source of G as follows:

- if $j < i$ then $j' = j$,
- if $j \geqslant i$ then $j' = j+1$

This is done correctly by the operations $\rho_{n \to n+1}$ in R_i (for $i \leqslant n \leqslant k-1$) and $\rho_{\$ \to}$ in (8). This completes the proof of Fact 5.7. □

Proof of Theorem 5.5. (*Conclusion*). (1) Let $G \in U'$. Its interior $Int(G)$ is the empty graph in the case $G = I$ and it follows from Facts 5.6 and 5.7 that it can be defined by a term t in $M(C)$ where $C = \{0, 1, 2, \ldots, k, *, \$\}$. We can observe that only the following sets can appear in $\triangle^*(t)$: the set $\{*\}$ and the set $P \cup \{0\}$ for every subset P of $[k] \cup \{\$\}$.

It follows from Proposition 5.4 that $Int(G)$ can be defined by a term in $T(C')$ for some set C' in bijection with $\triangle^*(t)$, where

$$Card(\triangle^*(t)) \leqslant Card(C') = 2^{k+1} + 1.$$

We now get G from $Int(G)$ by (7). But we can use $\{i, \$\}$ instead of i' for $1 \leqslant i \leqslant k$. It follows that G can be defined by a term in $T(C')$. Hence, $cwd(G) \leqslant 2^{k+1} + 1$.

(2) We now consider the case of directed graphs. We shall mainly adapt the previous proof. We denote by u the mapping that transforms a directed graph into an undirected graph by replacing every directed edge (x, y) by the undirected one $\{x, y\}$. In this transformation two opposite directed edges (x, y) and (y, x) get fused into a single undirected one.

For every undirected graph G, the graphs in the set $u^{-1}(G)$ are obtained from G by the replacement of every undirected edge $\{x, y\}$ by either the unique directed edge (x, y) or the unique directed edge (y, x) or the two directed edges (x, y) and (y, x). Going back to the proof of (6) one can see that it is enough to prove that

$$cwd(H) \leqslant 2^{2k+2} + 1$$

for every graph $H \in u^{-1}(G)$ where $G \in U'$. Recall that the set U' was defined in Eq. (4).

For every $P, P' \subseteq [k]$ we let $S_{P,P'}$, be the k-ary mapping on directed k-sourced graphs defined by

$$S_{P,P'}(G_1, \ldots, G_k) = fg(i_1(G_1) \ldots // i_k(G_k) // f_{l_1} // \ldots // f_{l_n} // f'_{j_1} // \ldots // f'_{j_m})$$

where $P = \{l_1, \ldots, l_n\}$, $P' = \{j_1, \ldots, j_m\}$ and for $i = 1, \ldots, k$:

- f_i is the $(k+1)$-sourced graph with no internal vertex and an edge from the i-source to the $(k+1)$-source,
- f'_i is the same graph with its edge in the reverse direction.

The set $W = u^{-1}(U')$ is characterized by the recursive equation

$$W = W // W \cup \cdots \cup S_{P,P'}(W, \ldots, W) \ldots \cup \{I\}, \tag{9}$$

where the union extends to all subsets P, P' of $[k]$. We define the interior $Int(G)$ of a graph G in W as in the proof of (7) except that we let $\{0\} \cup [k] \cup \{i'/i \in [k]\}$ be the

set of vertex labels (of cardinality $2k+1$) and that a vertex v of $Int(G)$ has always label 0 and has label i, $i=1,\ldots,k$, if and only if there is an arc in G from the i-source to v and label i', $i=1,\ldots,k$ if and only if there is an arc in G from v to the i-source.

It follows then that

$$G=\ldots(\rho_{i''\to i}(\ldots(\rho_{i\to}(\ldots(\rho_{i'\to}(\ldots\alpha_{i'',i}\ldots(\alpha_{i',i''}(\ldots Int(G)\oplus 1''\oplus\cdots\oplus k'')\ldots),\tag{10}$$

where i ranges from 1 to k, so that i' ranges over $\{1',\ldots,k'\}$ and i'' ranges over $\{1'',\ldots,k''\}$.

Fact 5.6 is valid for graphs in W, and Fact 5.7 is modified as follows.

Fact 5.7′. *For $G_1,\ldots,G_k\in W$ and $G=S_{P,P'}(G_1,\ldots,G_k)$ we have*

$$Int(G)=\rho_{*\to P\cup Q\cup\{0\}}(\rho_{\$\to}(\rho_{\$'\to}(\alpha_{\$',*}(\alpha_{*,\$}(R_1(Int(G_1))\oplus\cdots\oplus R_k(Int(G_k))\oplus *))\ldots))$$

where $Q=\{i'/i\in P'\}$ and for every graph H and $i\in[k]$, we let

$$R_i(H):=\rho_{i\to i+1}(\ldots(\rho_{k-1\to k}(\rho_{i'\to(i+1)'}(\ldots(\rho_{(k-1)'\to k'}(\rho_{k\to\$}(\rho_{k'\to\$'}(H))\ldots)).$$

For every graph $G\in W$, the graph $Int(G)$ can be denoted by a term t in $M(\{0,1,\ldots,k,1',\ldots,k',\$,\$',*\})$. By observing that only the set $\{*\}$ and the subsets of $\{0,1,\ldots,k,1',\ldots,k',\$,\$'\}$ containing 0 can appear in $\triangle^*(t)$, we get that

$$cwd(\ell(Int(G)))\leqslant 2^{2k+2}+1,$$

completing the proof. □

Corollary 5.8. *For every undirected graph G we have $cwd(G)\leqslant 2^{\min\{twd(G)+1,twd(\bar{G})\}}+2$.*

Proof. By Theorem 4.1, we have $cwd(G)\leqslant 2cwd(\bar{G})$. By Theorem 5.1 applied to $\bar{G}$ we have $2cwd(\bar{G})\leqslant 2^{twd(\bar{G})+2}+2$. By Theorem 5.1, again, $cwd(G)\leqslant 2^{twd(G)+1}+1$, whence the result. □

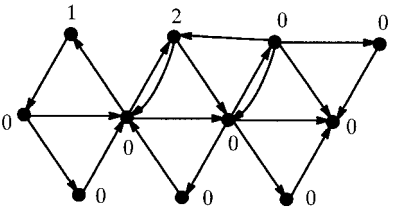
We illustrate these constructions with an example, taking $k=2$. Fig. 6 shows a graph G from W and its interior $Int(G)$. One can verify Eq. (10) namely, we have

$$G=\rho_{1''\to 1}(\rho_{2''\to 2}(\rho_{1\to}(\rho_{2\to}(\rho_{1'\to}(\rho_{2'\to}(\alpha_{1'',1}(\alpha_{2'',2}(\alpha_{1',1''}(\alpha_{2',2''}(Int(G)\oplus 1''\oplus 2''))\ldots)).$$

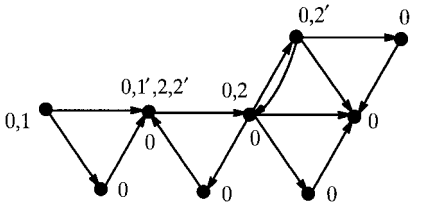
We have $G=S_{P,Q}(G_1,G_2)$ where $P=\{2\}$, $Q=\{1,2\}$ and the graphs G_1 and G_2 are shown in Fig. 7. We also show in this figure the graphs $Int(G_1)$ and $Int(G_2)$.

We can verify Fact 5.7', namely, that the following holds:

$$Int(G)=\rho_{*\to\{0,2,1',2'\}}(\rho_{\$\to}(\rho_{\$'\to}(\alpha_{\$',*}(\alpha_{*,\$}((\rho_{1\to 2}(\rho_{1'\to 2'}(\rho_{2\to\$}(\rho_{2'\to\$'}(Int(G_1)))))\oplus\rho_{2\to\$}(\rho_{2'\to\$'}(Int(G_2))\oplus *))\ldots)).$$

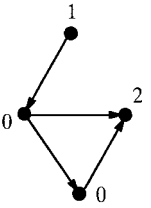


A graph G belonging to U′

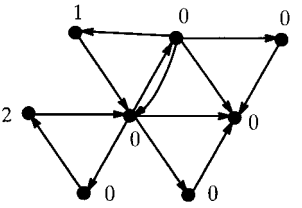


The graph Int(G)

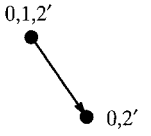
Fig. 6.



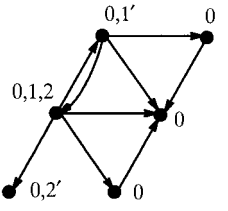
The graph G_2



The graph G_1



The graph Int(G_2)



The graph Int(G_1)

Fig. 7.

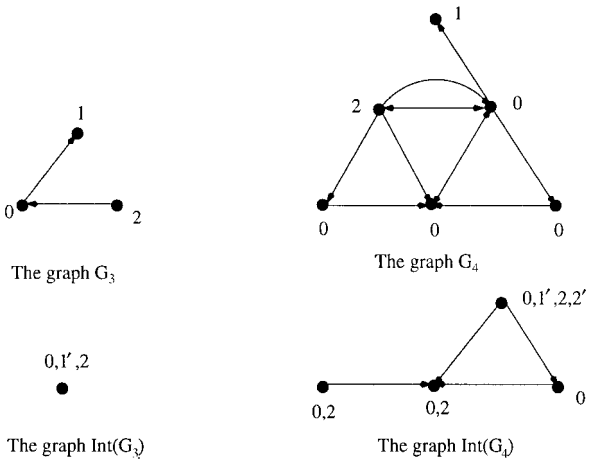


Fig. 8.

We now proceed and consider G_1. We have

$$G_1 = S_{P',Q'}(G_3, G_4)$$

where $P' = \{1,2\}$, $Q' = \emptyset$ and G_3, G_4 are illustrated in Fig. 8. We can verify Fact 5.7′ which gives

$$Int(G_1) = \rho_{*\to\{0,1,2\}}(\rho_{\$\to}(\rho_{\$'\to}(\alpha_{\$',*}(\alpha_{*,\$}(\rho_{1\to 2}(\rho_{1'\to 2'}(\rho_{2\to\$}(\rho_{2'\to\$'}(Int(G_3)))$$
$$\oplus\rho_{2\to\$}(\rho_{2'\to\$'}(Int(G_4))) \oplus *)))\ldots)).$$

By proceeding in this way, one would get terms defining $Int(G_4), Int(G_3)$, whence $Int(G_1)$, then $Int(G_2)$, whence $Int(G)$ and G. □

As observed at the beginning of this section, one cannot bound the tree width of an arbitrary graph in terms of its clique width. However, this is possible under additional conditions.

For a set of graphs L, we let

$twd(L) := Sup\{twd(G)/G \in L\} \in \mathcal{N} \cup \{\infty\}$,
$cwd(L) := Sup\{cwd(G)/G \in L\} \in \mathcal{N} \cup \{\infty\}$,
$deg(L) := Sup\{deg(G)/G \in L\} \in \mathcal{N} \cup \{\infty\}$

where $deg(G)$ is the maximal degree of a vertex of G,

$bip(L) := Sup\{bip(G)/G \in L\} \in \mathcal{N} \cup \{\infty\}$

where $bip(G)$ is the largest integer m such that $K_{m,m} \subseteq G$, (where $\subseteq$ denotes subgraph inclusion).

By an *integer function* we mean a monotone total function $(\mathcal{N} \cup \{\infty\})^m \to (\mathcal{N} \cup \{\infty\})$ where $m = 1, 2$ such that the result is ∞ iff at least one argument is ∞.

Theorem 5.9. *There exists integer functions $f_1,\dots,f_4,f_H$ where f_H is associated with each undirected graph H, such that for every set of undirected graphs L the following hold*:

1. $twd(L)\leqslant f_1(cwd(u^{-1}(L)))$,
2. $twd(L)\leqslant f_2(cwd(L))$, *if L contains all subgraphs of its graphs*,
3. $twd(L)\leqslant f_3(deg(L),cwd(L))$,
4. $twd(L)\leqslant f_4(bip(L),cwd(L))$,
5. $twd(L)\leqslant f_H(cwd,(L))$, *if L does not contain a fixed graph H as a minor.*

Proof. (1) Let L be such that $u^{-1}(L)\subseteq C\vec{W}D(\leqslant k)$. It is proved in Courcelle [9] that if a set of directed graphs of the form $u^{-1}(L)$ has a decidable MS_1-theory, then it has bounded tree width. The proof actually gives slightly more: it is enough to assume that $u^{-1}(L)$ is a subset of a set having a decidable MS_1-theory in order to get that $u^{-1}(L)$ (equivalently L) has bounded tree width. This gives the desired result since $C\vec{W}D(\leqslant k)$ has a decidable MS_1-theory (see [8,13]).

(2) Let $L\subseteq CWD(\leqslant k)$ contain all subgraphs of its graphs. Consider $L\cap$PLANAR. It is contained in $L'=CWD(\leqslant k)\cap$PLANAR which is VR because the set of planar graphs is MS_1-definable [7]. But, by Proposition 3.7 of [12], L' is of bounded tree width, and we obtain the existence of $f_2(k)$.

(3) Let $n,k\in\mathcal{N}$. Let L be such that $deg(L)\leqslant n$ and $cwd(L)\leqslant k$. Let $L_{k,n}=CWD(\leqslant k)\cap DEG(\leqslant n)$; hence $L\subseteq L_{k,n}$. Since $CWD(\leqslant k)$ is VR and $DEG(\leqslant n)$ is MS_1-definable, the set $L_{k,n}$ is VR (by Proposition 3.2.1). Since every VR set of bounded degree has bounded tree width (Lemma 3.6 of [12]), the set $L_{k,n}$ has its graphs of tree width bounded, say, by m. Hence $L\subseteq TWD(\leqslant m)$. This proves the existence of $f_3(k,n)$.

(4) Let $n,k\in\mathcal{N}$. Let $L\subseteq L'_{k,n}=CWD(\leqslant k)\cap BIP(\leqslant n)$ where $BIP(\leqslant n)$ is the set of graphs that do not contain the complete bipartite graph $K_{n,n}$ as a subgraph. This last set is MS_1-definable, hence $L'_{k,n}$ is VR. Whence $L'_{k,n}$ has bounded tree width (because it is a VR set that does not contain some fixed graph $K_{m,m}$ as a subgraph and by Theorem 4.1 [10]). Again, we obtain the existence of $f_4(k,n)$.

(5) Let $k\in\mathcal{N}$ and H be an undirected graph. Let $L\subseteq L_{k,H}=CWD(\leqslant k)\cap FORB(H)$ where $FORB(H)$ is the set of graphs that do not contain H as a minor. It is MS_1-definable [7]. The proof of case (3) works with $FORB(H)$ instead of $DEG(\leqslant m)$ and yields the existence of $f_H(k)$. It is proved in [12] that every VR set that is included in $FORB(H)$ for some graph H has bounded tree width.

6. Open questions

We are sure that there exist graphs of arbitrary high cwd, otherwise the MS_1-theory of the set of graphs would be decidable, and we know that this is not the case.

At this point, we lack the techniques for establishing that a given graph has *cwd* at least some given integer. In particular, it would be useful to have rules that reduce the size of a given graph but preserve its *cwd*. We state without proof a preliminary result in this direction.

A *string* in an undirected graph G is a path P joining a vertex x to a vertex y ($y \neq x$), such that all intermediate vertices on P have degree 2. We say that a string P in a graph G labeled by $\gamma: V(G) \to C$ is *reducible* if its length is at least 2 (the length is the number of edges) and the graph G' obtained from G by contracting the edge of P incident with x, say $\{x, v\}$, has a coloring γ' that coincides with γ on $(V(G) - V(P)) \cup \{x, y\}$ and is such that $\gamma'(V(P) - \{x, y, v\}) \subseteq \gamma(V(P) - \{x, y\})$ and $cwd(G', \gamma') \leqslant cwd(G, \gamma)$. (We let x be the vertex of G' resulting from the fusion of x and v.)

Proposition 6.1. *Let G be a labeled undirected graph of cwd at least* 4. *Every string of length* 11 *is reducible.*

The proof being long is not included in this paper. The reader can find it in the authors' homepages [16].

Here are other open questions:

Question 6.2. *Find a function f such that for every graph $G=(V,E)$, we have $cwd(G)$ at most $f(|V|)$. The identity function is a trivial upper bound. It remains to find a good one.*

Question 6.3. *How different can be $cwd(G)$ and $cwd(G')$ when G and G' differ by exactly one edge*?

Question 6.4. *What is the complexity of recognizing the class of graphs of cwd at most k? Is it NP-complete for fixed k, say $k=4$? Note that for $k=2$ the complexity is linear because the cographs are recognizable in linear time* [3].

Even if we have no answer to this last question, even if the problem of exhibiting a C-construction in the sense of Section 3 of a given graph for a minimal set C is untractable, the hierarchical decompositions of graphs that we have investigated are of algorithmic interest.

If $\mathscr{C}$ is a class of graphs for which there exists a polynomial algorithm constructing for every graph G in $\mathscr{C}$ an expression of width at most k for some fixed value k, then every MS_1 property is decidable in polynomial time for graphs in $\mathscr{C}$. This technique is used in Courcelle et al. [14] for P_4-sparse graphs (for these graphs $k = 4$). It is certainly applicable to other classes, as well.

Another approach would be to define a polynomial algorithm that constructs for every graph G a decomposition that would be of width "close to" $cwd(G)$.

Appendix Some graph transformations that preserve clique width

We prove that certain edges can be subdivided or contracted without increasing the cwd.

Let G be an undirected graph. A path P in G joining a vertex x and a vertex $y \neq x$ is called a *string* if every vertex in P different from x and y has degree 2. The length of the string is the number of its edges, i.e. the number of vertices minus 1. In this terminology, an edge is a string of length 1.

Let G be labeled by $\gamma : V(G) \to \mathscr{C}$. We say that a string P in G is *reducible* if its length is at least 2 and the graph G' obtained from G by contracting the edge of P incident with x, say, $\{x, v\}$ has a coloring γ' that coincides with γ on the set $V(G) - V(P) \cup \{x, y\}$, and is such that $\gamma'(V(P) - \{x, y, v\}) \subseteq \gamma(V(P) - \{x, y\})$ and $cwd(G', \gamma') \leqslant cwd(G, \gamma)$. We let x be the vertex of G' resulting from the fusion of x and v: hence $V(G') = V(G) - \{v\}$.

We say that a string P is *extendible* if it has length at least 2 and the graph G' obtained from G by subdividing one edge of P by inserting a new vertex, say v, has a coloring γ' that coincides with γ on the set $V(G) - V(P) \cup \{x, y\}$ and is such that $\gamma'(V(P) - \{x, y\}) \cup \{v\} \subseteq \gamma(V(P) - \{x, y\})$ and $cwd(G', \gamma') \leqslant cwd(G, \gamma)$. (Here, $V(G') = V(G) \cup \{v\}$).

A string P is *dangling* if one of its ends has degree 1 and has a label different from all others in the graph.

Proposition A.1. *Let G be a labeled undirected graph of cwd at least* 4.

(1) *Every dangling string of length at least* 2 *is reducible.*

(2) *Every dangling string of length at least* 5 *is extendible.*

Proof. (1) Let G be obtained by an irredundant C-construction $(G_0, G_1, \ldots, G_n = G)$. Our aim is to prove that G' obtained by reduction of a string P has *cwd* at most $Card(G)$.

The proof is by induction on n. We consider the last step $G_{n-1} = L \to G$ of the construction.

Case 1: $L = K \oplus H, G = K \oplus \rho_{a,b}(H)$. P is a dangling string in G: let c be the label of an end of P of degree 1 that does not occur anywhere else in G. Call v this end.

Subcase 1.1: *P is in K*. Then K has a construction of length at most $n-1$ and P can be reduced (in K), giving K'. The graph $G' = K' \oplus \rho_{a,b}(H)$ has *cwd* at most

$$Max\{cwd(K'), cwd(\rho_{a,b}(H))\} \leqslant Max\{cwd(K), cwd(G)\} \leqslant cwd(G)$$

and is obtained from G by reduction of P as desired.

Subcase 1.2: *P is in $\rho_{a,b}(H)$*. We cannot have $a = c$ (because a does not label any vertex of $\rho_{a,b}(H)$). If we have $b = c$ this means that v has label a or c in H and that no other vertex of H has label a or c. Hence P is dangling in H. The same is true if $c \neq b$. The string P can thus be reduced giving H'. The graph $G' = K \oplus \rho_{a,b}(H')$

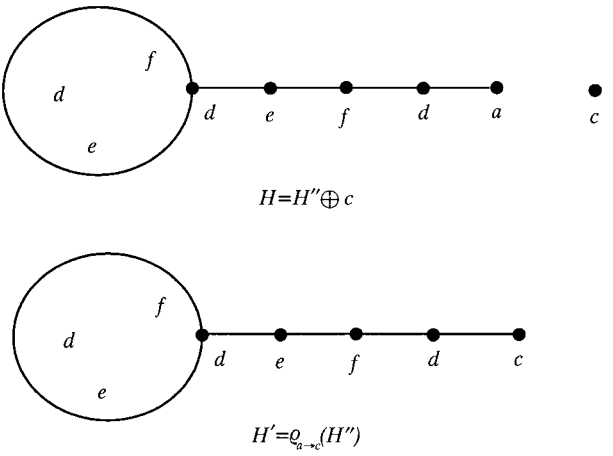


Fig. 9.

satisfies the desired properties and

$$cwd(K \oplus \rho_{a,b}(H')) = Max\{cwd(K), cwd(\rho_{a,b}(H'))\} \leqslant cwd(G).$$

Case 2: $G'' = K \oplus H, G = K \oplus \eta_{a,b}(H)$. Let P and v be as in Case 1. We assume that H has at least one a-port and at least one b-port. Furthermore, no a-port is adjacent to a b-port since the construction is irredundant.

Subcase 2.1: P is in K. Similar to Subcase 1.1.

Subcase 2.2: *P is in $\eta_{a,b}(H)$ but no edge of P is created by $\eta_{a,b}$: P is actually already in H and $c \notin \{a,b\}$*. We reduce P in H, obtaining H' and we can take $G' = K \oplus \eta_{a,b}(H')$.

Subcase 2.3: *P is in $\eta_{a,b}(H)$ and one or two edges of P is (or are) created by $\eta_{a,b}$*. Again we have several cases to consider

Subsubcase 2.3.1: *$H = H'' \oplus c, c = b$, a is the label of the end of a dangling string in H'', no other vertex in H'' is labeled a or b. We take $G' = K \oplus H'$ where $H' = \rho_{a\to c}(H'')$*. See Fig. 9 which shows H and H'.

For the following cases, we shall denote by w where $w = a_1 a_2, \ldots, a_n \in C^+$ (with $n \neq 1, a_1, a_2, \ldots, a_n \in$ C) any graph with n vertices $x_1, x_2, \ldots, x_n$, linked by edges between x_i and x_{i+1}, and such that x_i has label a_i. A word and its mirror image denote the same graphs.

Subsubcase 2.3.2: *$H = H'' \oplus bc, c \notin \{a,b\}$ and a is the label of the end of a dangling string in H'', no other vertex in H'' is labeled a or b or c*. We take $G' = K \oplus H'$, where $H' = \eta_{a,c}(H'' \oplus c)$.

Fig. 10 illustrates the graphs H and H'.

Subsubcase 2.3.3: *$H = H'' \oplus bwc, c \notin \{a,b\}$ and $w \in (C - \{a,b,c\})*$ (recall that C has cardinality at least 4); we let n be the length of w; we let d be a label occurring in w. The graph $bd^{n-1}c$ can be constructed with at most 4 labels (2 if $n = 1, 3$ if*

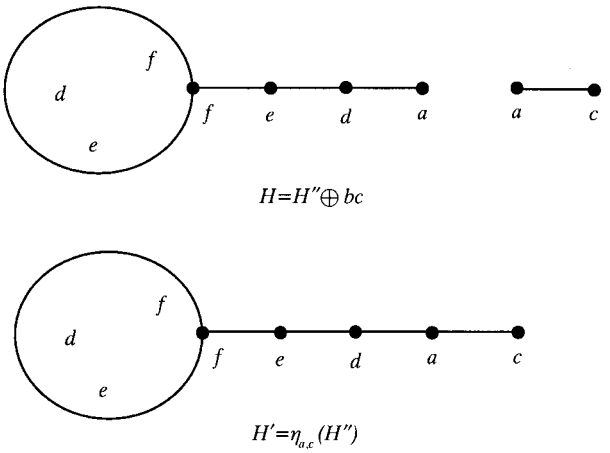


Fig. 10.

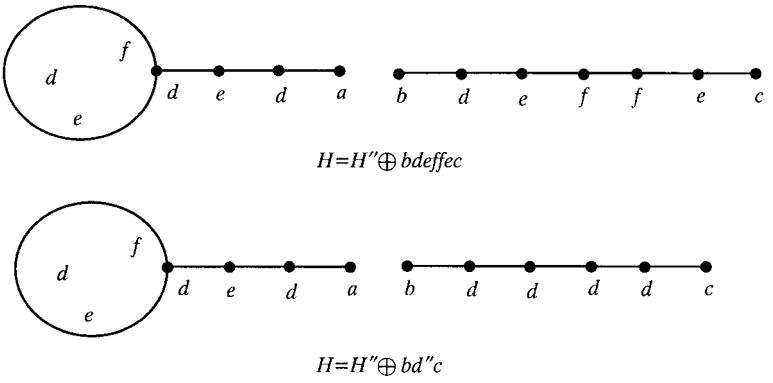


Fig. 11.

$n=2,4$ *if* $n\geqslant 3$). We let then $G'=K\oplus\eta_{a,c}(H')$, where $H'=H''\oplus bd^{n-1}c$. We have:

$$cwd(G')\leqslant \max\{cwd(K),\max\{cwd(H''),4\}\}\leqslant cwd(G)$$

since $cwd(G)\geqslant 4$. See Fig. 11 with $n=5, w=deffe$ shows H and H'.

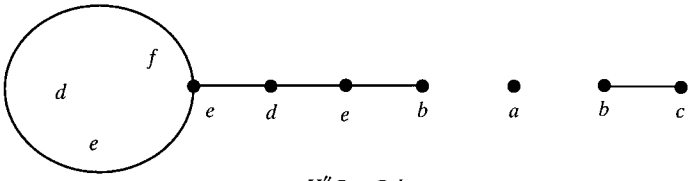
Subcase 2.3.4: $H=H''\oplus a\oplus bc, c\notin\{a,b\}$ *and* H'' *has a dangling string with end vertex labeled by b.* (*here, two edges of P are created simultaneously by* $\eta_{a,b}$). Clearly, H'' has no vertex labeled by a or c. We let

$$G'=K\oplus\eta_{a,b}(\eta_{a,c}(H''\oplus a\oplus c)).$$

For an illustration of $H''\oplus a\oplus bc$ we refer the reader to Fig. 12. Clearly (cf. 1.2)

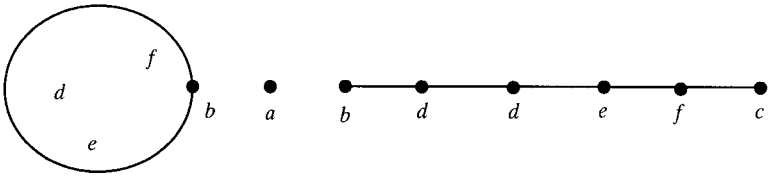
$$cwd(G')\leqslant Max\{cwd(K),cwd(H'')\}\leqslant cwd(G).$$

Subcase 2.3.5: $H=H''\oplus a\oplus bwc, c\notin\{a,b\}, w\in(C-\{a,b\})^{+}$ *and* H'' *has a dangling string with end vertex labeled by b and no a-port or c-port.* We let d occur

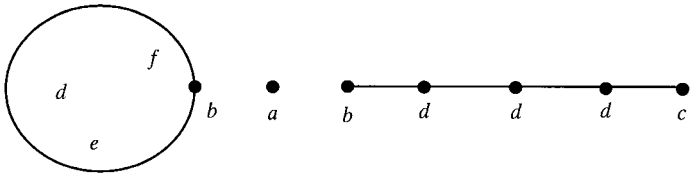


$H'' \oplus a \oplus bc$

Fig. 12.



$H'' \oplus a \oplus bddefc$



$H'' \oplus a \oplus bdddc$

Fig. 13.

in w and $n = |w|$. As in Subsubcase 2.3.3, we let

$$G' = K \oplus \eta_{a,b}(H'' \oplus a \oplus bd^{n-1}c).$$

See Fig. 13 where $w = ddef, n = 4$. It illustrates $H'' \oplus a \oplus bwc$ and $H'' \oplus a \oplus bd^3c$. Clearly

$$cwd(G') \leqslant Max\{cwd(K), cwd(H''), 4\} \leqslant cwd(G).$$

(See Case 2.3.3). This completes the proof of assertion (1).

(2) We now consider the extension of strings. We use the same case analysis as in (1) but we assume that P has length at least 5.

Case 1: Same proof as in Part 1.

Case 2:

Subcases 2.1 *and* 2.2: Same proofs as in Part 1.

Subcase 2.3: $G = K \oplus \eta_{a,b}(H), P$ *is in* $\eta_{a,b}(H)$ *and one or two edges of* P *are created by* $\eta_{a,b}$.

Subsubcase 2.3.1: $H = H'' \oplus c$. Since P has length larger than 3, the dangling string in H'' has an intermediate vertex. Let d its label: $d \notin \{a, c\}$ (recall that $b = c$). We let $G' = K \oplus H'$ where

$$H' = \eta_{a,c}(\rho_{c \to a}(\rho_{a \to d}(\eta_{a,c}(H'' \oplus c)))) \oplus c).$$

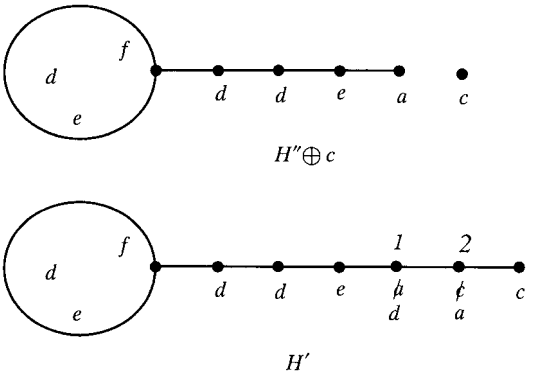


Fig. 14.

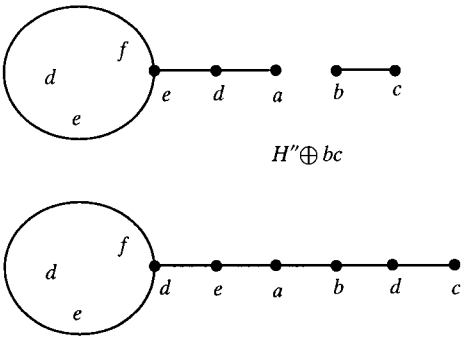


Fig. 15.

See Fig. 14 showing $H'' \oplus c$ and H'.

The edge marked 1 is created by the innermost operation $\eta_{a,c}$ and the one marked 2 by the outermost one. The changes of vertex label are recorded from top to bottom.

Subsubcase 2.3.2: $H = H'' \oplus bc$. Since P has length at least 5, the dangling string in H'' has an intermediate vertex. Let d its label. Then $d \notin \{a, b, c\}$. We let $G' = K \oplus H'$ where $H' = \eta_{a,b}(H'' \oplus bdc)$ See Fig. 15 showing $H'' \oplus bc$ and H'. The proof is as in Case 2.3.3 of part 1.

Subsubcase 2.3.3: $H = H'' \oplus bwc$ *with* $w \in (C - \{a, b, c\})^+$. We assume that d occurs in w. The construction is as in 2.3.3 of part 1 with $bd^{n+1}c$ instead of $bd^{-1}c$.

Subsubcase 2.3.4: $H = H'' \oplus a \oplus bc, c \notin \{a, b\}, H''$ *has a dangling string* P'' *with end vertex labeled by* b. Since P has length at least 5, P'' has length at least two,

hence has an intermediate vertex labeled by, say, d.($d \notin \{a,b,c\}$). We let

$$G' = K \oplus \eta_{a,b}(H'' \oplus a \oplus bdc).$$

Subsubcase 2.3.5: $H = H'' \oplus a \oplus bwc, c \notin \{a,b\}, w \in (C - \{a,b,c\})^+$. We let d occur in w, n be the length of w and

$$G' = K \oplus \eta_{a,b}(H'' \oplus a \oplus bd^{n+1}c).$$

We have $cwd(G') \leqslant cwd(G)$ by the same argument as in Case 2.3.5 of part 1. This completes the proof of Proposition A.1. □

A string P is *separating* if its two ends are not linked by any other path than P. After removal of one edge of P its two ends belong to different connected components.

Proposition A.2. *Let G be a labeled undirected graph of cwd at least* 4.
(1) *Every separating string of length at least* 5 *is reducible.*
(2) *Every separating string of length at least* 11 *is extendible.*

Proof. (1) We let G be given by a C-construction $(G_0, G_1, \ldots, G_n)$. We let P be a string of length at least 5.

We consider the last step $G_{n-1} = L \to G$ of this construction.

Case 1: $L = K \oplus H, G = K \oplus \rho_{a,b}(H)$. If P is in K, we transform K into K' (induction on n) and we take $G' = K' \oplus \rho_{a\to b}(H)$. If P is in $\rho_{a\to b}(H)$, then we also have a string P' in H. We transform H into H' and we take $G' = K \oplus \rho_{a\to b}(H')$.

Case 2: $L = K \oplus H, G = K \oplus \eta_{a,b}(H)$. If P is in K, the proof is as in the first subcase of Case 1. If P is in $\eta_{a,b}(H)$ but $\eta_{a,b}$ creates no edge of P this means that P is already in H. We transform H into H' and we let $G' = K \oplus \eta_{a,b}(H')$.

Finally, we consider cases where P is in $\eta_{a,b}(H)$ and $\eta_{a,b}$ creates one or two edges in P. There are several cases; see Fig. 16 showing H.

Subcase 2.1: H has two connected components, each with a dangling string. Since P has length larger than 4, at least one of these dangling strings has length at least 2: A dangling string of length at least 2 is reducible by Proposition 9.1, which gives H' whence $G' = K \oplus \eta_{a,b}(H)$ as desired.

Subcase 2.2: Again since P has length at least 5, H has at least one connected component with a dangling string of length at least 2. This string can be reduced and we conclude the proof as in Subcase 2.1.

Subcase 2.3: Here the operation $\eta_{a,b}$ creates an edge in P but simultaneously one or more edges not in P. Since P has length larger than 3, one connected component has a dangling string of length at least 2, which can be reduced and we conclude as in the preceding two cases.

(2) We want to extend the considered string. The argument is the same. We need only assume that P has length at least 11 in order to apply Proposition A.1 in Subcase 2.2. We omit the details. □

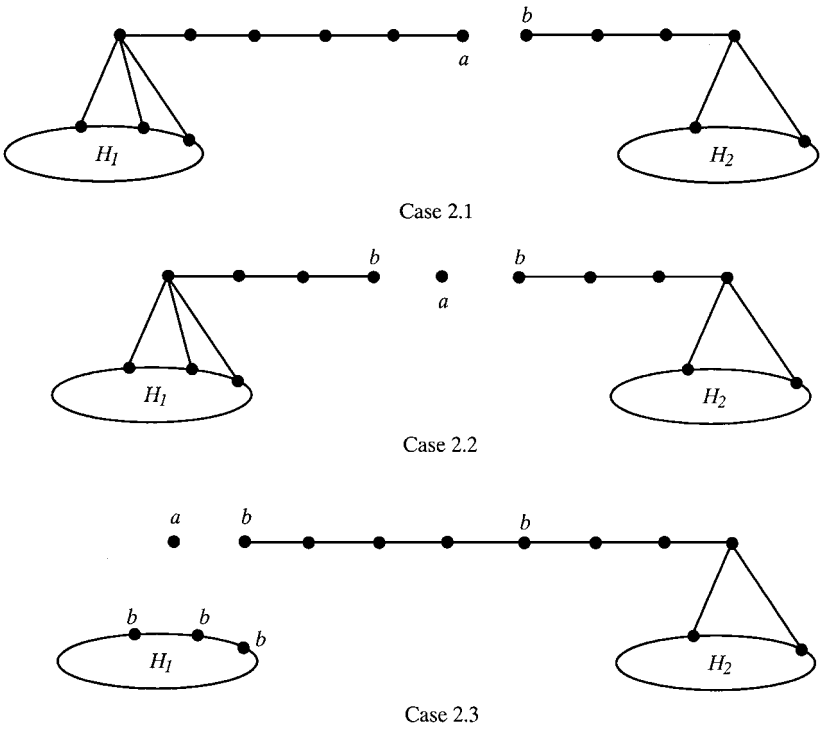


Fig. 16.

Proposition A.3. *Let G be a labeled undirected graph of cwd at least* 4. *The following conditions hold*:

(1) *Every string of length at least* 11 *is reducible.*

(2) *Every string of length at least* 23 *is extendible.*

Proof. (1) Similar to that of Proposition A.2. We only review the various subcases.

Subcase 2.1: We are like in Case 2.1 of Fig. 16 except that H_1 and H_2 may be connected by a path (that does not cross P). Since a and b have no other occurrence in H than the ends of the two strings, H has a dangling string of length at least 2 and the proof goes on as in Proposition A.2.

Subcase 2.2: We are like in Case 2.2 of Fig. 16 with possibly a path connecting H_1 and H_2. Since b may have two occurrences in a same connected components, the two strings with end labeled by b are not necessarily dangling. However, they are separating. Since P has length larger than 10, at least one of them has length at least 5. They are both separating hence, at least one of them can be reduced by Proposition A.2. The proof continues as usual.

Subcase 2.3: Same argument as in Subcase 2.2. Since P has length larger than 5 the strings linked to H_2 (see Fig. 16, Case 2.3) is separating and can be reduced by Proposition A.2.

(2) Same proofs as in Part 1. Since we assume that the length of P is larger than 22, we can find in Cases 2.1–2.3 a separating string of length at least 11 that is extendible. □

Corollary A.4. *Let G be a graph of cwd at least* 4 *having a string P of length at least* 23. *All graphs obtained from G by repeated subdivisions of the edges of P have the same cwd as G.*

Open Question:. *Is* 23 *the optimal lower bound*?

We conjecture that it is not.

References

[1] S. Arnborg, J. Lagergren, D. Seese, Easy problems for tree-decomposable graphs, J. Algorithms 12 (1991) 308–340.

[2] S. Arnborg, B. Courcelle, A. Proskurowski, D. Seese, An algebraic theory of graph reductions, J. ACM 40 (1993) 1134–1164.

[3] D.G. Corneil, Y. Perl, L.K. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput. 14 (1985) 926–934.

[4] 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, New Jersey, 1997, pp. 313–400.

[5] B. Courcelle, An axiomatic definition of context-free rewriting and its application to NLC graph grammars, Theoret. Comput. Sci. 55 (1987) 141–181.

[6] B. Courcelle, The monadic second-order logic of graphs I, Recognizable sets of finite graphs, Inform. and Comput. 85 (1990) 12–75.

[7] B. Courcelle, he monadic second-order logic of graphs III, Tree-decompositions, minors and complexity issues, Inform. Théorique Appl. 26 (1992) 257–286.

[8] B. Courcelle, The monadic second order logic of graphs VII: Graphs as relational structures, Theoret. Comput. Sci. 101 (1992) 3–33.

[9] B. Courcelle, The monadic second order logic of graphs VIII: Orientations, Ann. Pure Appl. Logic 72 (1995) 103–143.

[10] B. Courcelle, Structural properties of context-free sets of graphs generated by vertex-replacement, Inform. and Comput. 116 (1995) 275–293.

[11] B. Courcelle, The monadic second-order logic of graphs X: linear orderings, Theoret. Comput. Sci. 160 (1996) 87–143.

[12] B. Courcelle, J. Engelfriet, A logical characterization of hypergraph languages generated by hyperedge replacement grammars, Math. System Theory 28 (1995) 515–552.

[13] B. Courcelle, J. Engelfriet, G. Rozenberg, Handle-rewriting hypergraph grammars, J. Comput. System Sci. 46 (1993) 218–270.

[14] B. Courcelle, J. Makowsky, U. Rotics, Linear-time solvable optimization problems on certain structured graphs, Proceedings of WG'98, to appear.

[15] B. Courcelle, M. Mosbah, Monadic second-order evaluations on tree-decomposable graphs, Theoret. Comput. Sci. 109 (1993) 49–82.

[16] B. Courcelle, S. Olariu, Some graph transformations that preserve clique width (an appendix to "Upper Bounds to the Clique-Width of Graphs"), `http://dept-info.labri.u-bordeaux.fr/~courcell/ActSci.ht ml` or `//http://www.cs.odu.edu/~olariu/cwd.html`.

[17] A. Cournier, M. Habib, A new linear algorithm for modular decomposition, preprint 1995; preliminary version in Proceedings of CAAP'94, Lecture Notes in Computer Science, Vol. 787, 1994, pp. 68–84.

[18] A. Ehrenfeucht, T. Harju, G. Rozenberg, 2-structures, a framework for decomposition and transformation of graphs, in: G. Rozenberg (Ed.), Handbook of Graph Grammars and Computing by Graph Transformations, World Scientific, New Jersey, 1997, pp. 401–478.
[19] J. Engelfriet, A characterization of context-free NCE graph languages by monadic second-order logic on trees, in Fourth International Workshop on Graph Grammars and their Applications to Computer Science, Lecture Notes in Computer Science, Vol. 532, 1991, pp. 311–327.
[20] J. Engelfriet, G. Rozenberg, Node replacement graph grammars, in: G. Rozenberg (Ed.), Handbook of Graph Grammars and Computing by Graph Transformations, World Scientific, New Jersey, 1997, pp. 1–94.
[21] S. Hedetniemi, Efficient algorithms and partial k-trees, Discrete Appl. Math. 54 (1994) 281–290.
[22] J. Muller, J. Spinrad, Incremental modular decomposition, J. ACM 36 (1989) 1–19.
[23] N. Robertson, P. Seymour, Graph minors V, Excluding a planar graph, J. Combin. Theory (B) 41 (1986) 92–114.
[24] D. Seese, The structure of the models of decidable monadic theories of graphs Ann. Pure Appl. Logic 53 1991 169–195
[25] J. Van Leeuwen, Graph algorithms, Handbook of Theoretical Computer Science, Elsevier, Amsterdam, 1990, pp. 525–631 (Chapter 10).
[26] E. Wanke, k-NLC graphs and polynomial algorithms, Discrete Appl. Math. 54 (1994) 251–266.