Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 81 (2005)

Proving NP-hardness for clique-width II: non-approximability of clique-width

Michael R. Fellows[*], Frances A. Rosamond[*], Udi Rotics[†], and Stefan Szeider[‡]

August 24, 2005

Abstract

Clique-width is a graph parameter that measures in a certain sense the complexity of a graph. Hard graph problems (e.g., problems expressible in Monadic Second Order Logic with second-order quantification on vertex sets, that includes NP-hard problems) can be solved efficiently for graphs of certified small clique-width. It is widely believed that determining the clique-width of a graph is NP-hard; in spite of considerable efforts, no NP-hardness proof has been found so far. We give the first hardness proof. We show that the clique-width of a given graph cannot be absolutely approximated in polynomial time unless P = NP. We also show that, given a graph G and an integer k, deciding whether the clique-width of G is at most k is NP-complete. This solves a problem that has been open since the introduction of clique-width in the early 1990s.

1 Introduction

The clique-width of a graph is the smallest number of labels that suffices to construct the graph using the operations: creation of a new vertex v with label i, disjoint union, insertion of edges between vertices of certain labels, and relabeling of vertices. Such a construction of a graph by means of these four operations using at most k different labels can be represented by an algebraic expression called a *k-expression* (more exact definitions are provided in Section 1.2). This composition mechanism was first considered by Courcelle, Engelfriet, and Rozenberg [4] in 1990; the term clique-width was introduced later.

By a general result of Courcelle, Makowsky, and Rotics [5], any graph problem that can be expressed in Monadic Second Order Logic with second-order quantification on vertex sets (that includes NP-hard problems) can be solved in linear time for graphs of clique-width bounded by some constant k if the k-expression is provided as input to the algorithm (albeit the running time involves a constant which is exponential in k). A main limit for applications of this result is that it is not known how to obtain efficiently k-expressions for graphs with clique-width k. Is it possible to compute the clique-width of a graph in polynomial time? This question has been open since the introduction of clique-width. In the present paper we answer this question negatively: We show that the clique-width of a graph cannot be

[*]School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan 2308 NSW, Australia, [mfellows|fran]@cs.newcastle.edu.au.

[†]School of Computer Science and Mathematics, Netanya Academic College, Netanya, Israel, rotics@mars.netanya.ac.il.

[‡]Department of Computer Science, Durham University, Durham, England, UK, stefan.szeider@durham.ac.uk.

ISSN 1433-8092

computed in polynomial time unless P = NP, and given a graph G and an integer k, deciding whether the clique-width of G is at most k is NP-complete.

With considerable efforts, polynomial-time algorithms could be developed for recognizing graphs of clique-width at most 3 in polynomial time (see Corneil, Habib, Lanlignel, Reed, and Rotics [3]). Recently, Oum and Seymour [10] obtained an algorithm that, for any fixed k, runs in time $O(n^9 \log n)$ and computes $(2^{3k+2} - 1)$-expressions for graphs of clique-width at most k. This result renders the notion "class of bounded clique-width" feasible; however, since the running time of algorithms as suggested by Courcelle et al. [5] crucially depends on k, closer approximations are desirable. The graph parameter "NLC-width" introduced by Wanke [11] is defined similarly as clique-width using a single operation that combines disjoint union and insertion of edges. Recently Gurski and Wanke [7] have reported that computing the NLC-width is NP-hard. Since NLC-width and clique-width can differ by a factor of 2 (see Johansson [8]), non-approximability with an absolute error guarantee for one of the two parameters does not imply a similar result for the other parameter.

The main results of our paper are the following.

Theorem 1. *The clique-width of graphs with n vertices of degree greater than 2 cannot be approximated by a polynomial-time algorithm with an absolute error guarantee of n^ε for any $\varepsilon \in (0,1)$, unless* P = NP.

In particular, there is no polynomial-time absolute approximation algorithm for clique-width unless P = NP.

Theorem 2. *The problem* cwd-MINIMIZATION *(that is, given a graph G and an integer k, is the clique-width of G at most k?) is* NP*-complete.*

In the first part of this series of papers [6] we have shown results similar to Theorems 1 and 2 for a weaker notion of clique-width, termed *sequential clique-width* (or linear clique-width). The sequential clique-width of a graph is defined similarly as clique-width, except that only k-expressions are considered where at least one of any two k-graphs put together by disjoint union is an initial k-graph. The parse trees of such sequential k-expressions are path-like (every node is either a leaf or adjacent to a leaf). Hence one can consider the relation between sequential clique-width and clique-width as an analogue to the relation between pathwidth and treewidth. The natural 2-expressions of complete graphs (see Section 1.2) are sequential.

1.1 Proof outline

In what follows, let α be an integer-valued graph parameter. We consider the following decision problem.

α-MINIMIZATION
Instance: A graph G and a positive integer k.
Question: Is $\alpha(G)$ at most k?

In [6] we have shown the following lemma using results of Karpinski and Wirtgen [9], Arnborg, et al. [1], and Bodlaender, et al. [2].

Lemma 1. *Assume that there is a constant c such that $|\alpha(G) - \mathrm{pwd}(G)| \le c$ holds for every cobipartite graph G with minimum degree at least 3. Then the following statements are true.*

1. *For a graph G with n vertices and minimum degree at least 3, $\alpha(G)$ cannot be approximated in polynomial-time with an absolute error guarantee of n^ε for any $\varepsilon \in (0,1)$ unless* P = NP.

2. α-MINIMIZATION *is* NP*-hard.*

We shall use the following two constructions.

Let G be a graph. We obtain a graph G' from G by replacing each edge xy of G by three paths $x - p_i - q_i - y$, $i = 1, 2, 3$, where p_i, q_i are new vertices. Similarly, we obtain from G a graph G'' by replacing each edge xy of G by one path $x - s - y$ where s is a new vertex.

In the companion paper [6] we have shown the following inequation ($\text{pwd}(G)$ and $\text{cwd}_1(G)$ denote the *pathwidth* and the sequential clique-width of G, respectively).

$$\text{pwd}(G) \leq \text{cwd}_1(G') \leq \text{pwd}(G) + 4. \tag{1}$$

In this paper we establish for cobipartite graphs of minimum degree at least 2 the following inequation ($\text{cwd}(G)$ denotes the clique-width of G).

$$\text{cwd}(G') \leq \text{cwd}_1(G') \leq \text{cwd}(G') + 18. \tag{2}$$

The non-trivial part of inequation (2) is obtained by means of the second construction G''. We show by Lemma 2, Theorem 3, and Lemma 5, respectively, that for every cobipartite graph G we have

$$\text{cwd}_1(G') \leq \text{cwd}_1(G'') + 9 \leq \text{cwd}(G'') + 15 \leq \text{cwd}(G') + 18. \tag{3}$$

The hardest task for showing (3) is to bound the sequential clique-width of G'' in terms of the clique-width of G'' plus a small constant; this is established in Theorem 3.

Consider now the graph parameter $\alpha(G) = \text{cwd}(G')$; i.e., $\alpha(G)$ is the clique-width of the graph G' obtained from G by the first of the two construction given above. The inequations (1) and (2) yield $|\alpha(G) - \text{pwd}(G)| \leq 22$, hence the assumption of Lemma 1 is met. It is now easy to establish Theorems 1 and 2 as follows.

Assume that for a constant $\varepsilon \in (0, 1)$ there exists a polynomial-time algorithm $\mathcal{A}$ that outputs for a given graph G with n vertices of degree at least 3 an integer $\mathcal{A}(G)$ with $|\mathcal{A}(G) - \text{cwd}(G)| \leq n^\varepsilon$. For a graph G with n vertices and minimum degree at least 3, G' has exactly n vertices of degree at least 3; applying $\mathcal{A}$ to G' gives now $|\mathcal{A}(G') - \text{cwd}(G')| = |\mathcal{A}(G') - \alpha(G)| \leq n^\varepsilon$. Hence, by the first part of Lemma 1 such algorithm $\mathcal{A}$ cannot exist unless P = NP. A similar reasoning applies if the approximation error is bounded by some fixed constant. Thus Theorem 1 is established.

The second part of Lemma 1 implies that α-MINIMIZATION is NP-hard. We reduce α-MINIMIZATION to cwd-MINIMIZATION by taking for an instance (G, k) of the former problem the instance (G', k) of the latter problem; obviously $\alpha(G) \leq k$ if and only if $\text{cwd}(G') \leq k$. Thus cwd-MINIMIZATION is NP-hard as well. The problem is in NP since, given a graph G, we can guess a k-expression and check in polynomial time whether it is indeed a k-expression defining G. Thus Theorem 2 is established as well.

1.2 Definitions and preliminaries

All graphs considered in this paper are undirected and simple. Let k be a positive integer. A *k-graph* is a graph whose vertices are labeled by integers from $\{1, \ldots, k\}$. We consider an arbitrary graph as a k-graph with all vertices labeled by 1. We call the k-graph consisting of exactly one vertex v (say, labeled by $i \in \{1, \ldots, k\}$) an *initial k-graph* and denote it by $i(v)$.

The *clique-width* $\text{cwd}(G)$ of a graph G is the smallest integer k such that G can be constructed from initial k-graphs by means of repeated application of the following three operations.

- *Disjoint union* (denoted by $\oplus$);
- *Relabeling*: changing all labels i to j (denoted by $\rho_{i\to j}$);

- *Edge insertion*: connecting all vertices labeled by i with all vertices labeled by j, $i \neq j$ (denoted by $\eta_{i,j}$).

A construction of a k-graph using the above operations can be represented by an algebraic term composed of $\oplus$, $\rho_{i\to j}$, and $\eta_{i,j}$, ($i,j \in \{1,\dots,k\}$, and $i \neq j$). Such a term is called a cwd-*expression* defining G.

For example, the complete graph on the vertices u, v, w, x is defined by the cwd-expression

$$\rho_{2\to 1}(\eta_{1,2}(\rho_{2\to 1}(\eta_{1,2}(\rho_{2\to 1}(\eta_{1,2}(2(u)\oplus 1(v)))\oplus 2(w)))\oplus 2(x))).$$

In general, every complete graph K_n, $n \geq 2$, has clique-width 2.

For convenience, we assume that $\eta_{i,j}$ and $\eta_{j,i}$ denote the same operation.

For a cwd-expression t, we denote by $\mathsf{val}(t)$ the labeled graph defined by t. We denote a cwd-expression which uses at most k labels as a k-expression; for convenience we assume that the k labels are the integers $1,\dots,k$. Often when it is clear from the context we shall use the term expression instead of cwd-expression or k-expression. For a labeled graph G we denote by $\mathsf{labels}(G)$ the number of labels used in G.

For a cwd-expression t defining a graph G, we denote by $\mathsf{tree}(t)$ the parse tree constructed from t in the usual way. The leaves of this tree are the vertices of G with their initial labels, and the internal nodes correspond to the operations of t and can be either binary corresponding to $\oplus$, or unary corresponding to η or ρ. For a node a of $\mathsf{tree}(t)$, we denote by $\mathsf{tree}(t)\langle a\rangle$ the subtree of $\mathsf{tree}(t)$ rooted at a. We denote by $t\langle a\rangle$ the cwd-expression corresponding to $\mathsf{tree}(t)\langle a\rangle$; i.e., $\mathsf{tree}(t)\langle a\rangle = \mathsf{tree}(t\langle a\rangle)$. Note that in $t\langle a\rangle$ (and similarly in $\mathsf{tree}(t\langle a\rangle)$) we assume that the operation a is already established.

For a vertex x of $\mathsf{val}(t\langle a\rangle)$, we say that x is *dead at* a (or *dead at* $\mathsf{val}(t\langle a\rangle)$) if all the edges incident to x in $\mathsf{val}(t)$ are included in $\mathsf{val}(t\langle a\rangle)$. Otherwise we say that x is *active at* a (or *active at* $\mathsf{val}(t\langle a\rangle)$). We say that label ℓ is a *dead* in t if it is not involved in any η-operation in t. In other words, ℓ is dead in t if there is no η-operation in t of the form $\eta_{\ell,\ell'}$ for any label ℓ'.

Let a be a $\oplus$-operation of a cwd-expression t. If z is a vertex of $\mathsf{val}(t\langle a\rangle)$ and has label ℓ in $\mathsf{val}(t\langle a\rangle)$ we say that z *occurs at* a *with label* ℓ. Let b and c be the left and right children of a, respectively. We say that vertex x occurs on the *left (right) side* of a if it occurs at b (c).

Let r be a positive integer. We say that a is an r-$\oplus$-*operation* if there are at most r vertices occurring on the left side of a or there are at most r vertices occurring on the right side of a. We say that a is a ($> r$)-$\oplus$-*operation* if it is not an r-$\oplus$-operation. We say that t is an r-*sequential* cwd-*expression* (or *sequential* cwd-*expression* for $r = 1$) if all $\oplus$-operations in t are r-$\oplus$-operations. We say that t is a *sequential* k-*expression* if t is a sequential cwd-expression which uses k labels. For a graph G, $\mathrm{cwd}_r(G)$ denotes the smallest number k such that G can be defined by an r-sequential k-expression. For example, the above 2-expression defining K_4 is sequential. In general, we have $\mathrm{cwd}_1(K_n) = \mathrm{cwd}(K_n)$ for every $n \geq 1$.

For a graph G, we denote by G' the graph obtained from G by replacing each edge xy of G by three paths $x - p_i - q_i - y$, $i = 1, 2, 3$, where p_i, q_i are new vertices. Similarly, we denote by G'' the graph obtained from G by replacing each edge xy of G by one path $x - s - y$ where s is a new vertex which is denoted as $s_{x,y}$. We call the vertices of G' and G'' which are also vertices of G *regular vertices*. We call the vertices of G' and G'' which are not vertices of G *special vertices*.

2 From G'' to G' and back

For this section let G denote a graph with minimum degree at least 2. We show that the clique-width of G'' is bounded by the clique-width of G' plus a small constant, and that the converse is true for sequential clique-width.

2.1 From G'' to G'

Lemma 2. $\mathrm{cwd}_1(G') \leq \mathrm{cwd}_1(G'') + 9$.

For the proof we shall use the following definition and lemmas.

Property 1. Let t be a sequential cwd-expression defining G''. We say that t has *Property 1* if for every two regular vertices x and y there is no node a in $\mathsf{tree}(t)$ such that x and y are active at a and have the same label at a.

Lemma 3. *Let t be a sequential k-expression defining G''. Then there exists a sequential $(k+2)$-expression defining G'' which has Property 1.*

Proof. Let t be a sequential k-expression defining G''. Let x and y be two regular vertices such that there exists a node a in t such that x and y have the same label at a and are active at a. Let b the lowest node in $\mathsf{tree}(t)$ corresponding to an operation which unifies the labels of x and y. Clearly b corresponds to either a ρ or a 1-$\oplus$-operation. Suppose b corresponds to a 1-$\oplus$-operation. This operation introduces either x or y (say that it introduces x). Since x and y have the same label at b it follows that each neighbor of x is also a neighbor of y. However, since G has minimum degree at least 2, there is a neighbor of x in G'' which is not a neighbor of y, a contradiction.

Let b_1 be the child of b in $\mathsf{tree}(t)$. Clearly x and y are active at b. Since $s_{x,y}$ is the unique vertex in G'' which is adjacent to both x and y, it follows that if we add the edges connecting x and y to $s_{x,y}$ immediately above b_1, then x and y will not be active at b. We show below how to construct an expression t_1 which achieves this goal.

Let t_1' be the expression obtained by removing $s_{x,y}$ from t. Let t_1 be the expression obtained from t_1' by adding immediately above b_1 the vertex $s_{x,y}$ with label $k+2$, then adding two η-operations which connect $s_{x,y}$ to both x and y and then renaming the label of $s_{x,y}$ to $k+1$. (Note that $k+1$ will be a dead label, i.e., no edges will be added to a vertex having label $k+1$.) Since both edges connecting $s_{x,y}$ to x and y already exists at $\mathsf{val}(t_1\langle b\rangle)$, it follows that x and y are not active at $\mathsf{val}(t_1\langle b\rangle)$.

Repeating the above construction for every pair of regular vertices x and y which have the same label at a node a of $\mathsf{tree}(t)$ and are active at a, we finally get a sequential $(k+2)$-expression t' which defines G'' and satisfies Property 1.

Note that whenever vertex $s_{x,y}$ gets label $k+2$ at node a of t' it is the unique vertex having this label in $\mathsf{val}(t'\langle a\rangle)$ and thus, it is possible to connect it to x and y using two η-operations. □

Lemma 4. *Let t be a sequential k-expression defining G'' that has Property 1. Then there exists a sequential $(k+7)$-expression defining G'.*

Proof. Let t be a sequential k-expression defining G'' that has Property 1. Let $s = s_{x,y}$ be a special vertex of G''. Let e_1 and e_2 denote the edges connecting s to x and y, respectively. If the edges e_1 and e_2 are established in t by the same η-operation, then there is a node a in t such that both x and y have the same label at a and active at a, a contradiction. Thus, we can assume without loss of generality that the edge e_1 is established before e_2 in t. Let a denote the lowest node in $\mathsf{tree}(t)$ corresponding to the η-operation which establishes the edge e_1 in t. We can assume that node a is the only η-operation in t which connects x to s. Otherwise, we can remove from t all the η-operations above a which connect x to s. Let t'_1 denote the expression obtained by removing s from t. Let t_1 denote the expression obtained from t'_1 by replacing the node a with the following sequence of operations:

1. Add vertices $s_1, \ldots, s_6$ with labels $k+2, \ldots, k+7$, respectively.
2. Add η-operations connecting s_1, s_2, and s_3 to x.
3. Add η-operations connecting s_1 to s_4, s_2 to s_5, and s_3 to s_6.
4. Add ρ-operations which rename the labels of s_1, s_2, and s_3 to $k+1$ ($k+1$ is used as a dead label).
5. Add ρ-operations which rename the labels of s_4, s_5. and s_6 to ℓ, where ℓ is the label that s has in $\mathsf{val}(t\langle a\rangle)$.

It is easy to check that t_1 defines the graph obtained from G'' by replacing the path of length two $x-s-y$ with the 3 paths of length 3, $x-s_i-s_{i+3}-y$, $i=1,2,3$.

Repeating the above construction for every special vertex s of G'', we finally obtain a sequential $(k+7)$-expression t' which defines G'.

Note that whenever vertices $s_1, \ldots, s_6$ get labels $k+2, \ldots, k+7$ at node a of t' they are the unique vertices having these labels in $\mathsf{val}(t'\langle a\rangle)$ and thus, it is possible to establish all the connections and renamings mentioned in steps 2–5 above.

This completes the proof of the lemma. □

Proof of Lemma 2. Suppose $\mathrm{cwd}_1(G'') = k$, there there exists a sequential k-expression t which defines G''. By Lemma 3 there exists a sequential $(k+2)$-expression t_1 which defines G'' and has Property 1. By Lemma 4 there exists a sequential $(k+9)$-expression t_2 which defines G'. Thus $\mathrm{cwd}_1(G') \le k+9$. □

2.2 From G' to G''

Lemma 5. $\mathrm{cwd}(G'') \le \mathrm{cwd}(G') + 3$.

For proving this lemma we shall use the following definitions and lemma.

Let G be a graph and let $D(G)$ denote the set of graphs which can be obtained from G by replacing each edge of G either with a path of length two or with a path of length three. Clearly, the graph G'' belongs to $D(G)$ and is obtained by replacing all edges of G with a path of length two. For each graph G^* in $D(G)$ we call the vertices of G^* which are also vertices of G *regular vertices* and we call the other vertices of G^* *special vertices.*

Property 2. Let t be a k-expression defining a graph G^* in $D(G)$. We say that t has *Property 2* if the following conditions hold:

Condition 2.1: there is no η-operation in t which uses label 1, i.e, there is no $\eta_{1,\ell}$-operation in t for any label ℓ. In other words, 1 is a dead label.

Condition 2.2: if label 2 is used in t, then it is used as follows: a special vertex (say s) is introduced with label 2 using a 1-$\oplus$-operation say a, such that s is the only vertex having label 2 at a. Above a in $\mathsf{tree}(t)$ there is a sequence of one or more η-operations followed by a $\rho_{2\to\ell}$-operation where ℓ is any label different from 2 and 3.

Condition 2.3: if label 3 is used in t then it is used as follows: a regular vertex (say r) is introduced with label 3 using a 1-$\oplus$-operation, say a, such that r is the only vertex having label 3 at a. Above a in $\mathsf{tree}(t)$ there is a sequence of operations which can be either η, ρ, or 1-$\oplus$-operations introducing special vertices, followed by a $\rho_{3\to\ell}$-operation where ℓ is any label different from 2 and 3.

Condition 2.4: no regular vertex ever gets label 2 and no special vertex ever gets label 3.

Observation 1. *Let G^* be a graph in $D(G)$ and let $\mathrm{cwd}(G^*) = k$. Then there is a $(k+3)$-expression t' defining G^* which has Property 2.*

Proof. Let t be a k-expression defining G^*. Let t' be the $k+3$-expression obtained from t by replacing all occurrences of the labels $1,2$ and 3 with the labels $k+1$,$k+2$ and $k+3$, respectively. Clearly t' defines G^*. Since the labels $1,2$ and 3 are not used in t', it is obvious that t' has Property 2. $\square$

The following is the key lemma for proving Lemma 5.

Lemma 6. Let G^* be a graph in $D(G)$ and let t be a k-expression which defines G^* and has Property 2. Let a be a lowest node in $\mathsf{tree}(t)$ such that there exists an induced path $x-p-q-y$ in G'' (x,y are regular vertices) and x,p,q,y occur at a. Then there exists a k-expression t_1 which has Property 2 and defines the graph G_1^* obtained from G^* by replacing the path $x-p-q-y$ with a path $x-s-y$ where s is a new special vertex.

Proof. Let a and x,p,q,y as in the statement of the lemma. In each of the following cases we obtain a k-expression t_1 which defines G_1^* and has Property 2. In all cases it is easy to see that the expression t_1 obtained has Property 2.

Case 1: suppose x and y occur on different sides of a. Assume without loss of generality that x is on the left side of a and y is on the right side of a.

Case 1.1: suppose that p and q occur on the same side of a. Assume without loss of generality that both p and q occur on the left side of a. Let a_1 denote the lowest node in $\mathsf{tree}(t)$ such that both x and p are in $t\langle a_1\rangle$. Let a_2 denote the lowest node in $\mathsf{tree}(t)$ such that both x and q are in $t\langle a_2\rangle$. By the above assumptions both a_1 and a_2 are descendants of a in $\mathsf{tree}(t)$.

Case 1.1.1: suppose a_1 is a proper descendant of a_2 in $\mathsf{tree}(t)$. If x and q have the same label at a_2 it follows that y must be in $t\langle a_2\rangle$, a contradiction. Thus p and q must have unique labels at a_2. Let ℓ_p and ℓ_q denote the labels of p and q at a_2, respectively.

Case 1.1.1.1: suppose x has a unique label (say ℓ_x) at a_2. In this case, t_1 is obtained from t as follows:

1. Add the following sequence operations immediately above a_2:

1.1. An η_{ℓ_x,ℓ_p}-operation which connects x to p.

1.2. A $\rho_{\ell_p\to\ell_q}$-operation which renames the label of p to the label of q.

2. Omit q.

Case 1.1.1.2: Suppose x does not have unique label at a_2. Thus the edge connecting x to p already exists at $\mathsf{val}(t\langle a_2\rangle)$. In this case, t_1 is obtained from t as follows:

1. Add immediately above a_2 a $\rho_{\ell_p\to\ell_q}$-operation which renames the label of p to the label of q.

2. Omit q.

In both cases 1.1.1.1 and 1.1.1.2, p is connected to y since after p gets the label of q, the η-operation above a which connects q to y will connect p to y. Thus, p can be considered as the new special vertex s in G_1^* and the expression t_1 defines G_1^*.

Case 1.1.2: suppose a_1 is equal to a_2. In this case x and p must have unique labels at a_2. This case is handled the same way as case 1.1.1.1.

Case 1.1.3: suppose a_2 is a proper descendant of a_1 in $\mathsf{tree}(t)$. Since y is not in $t\langle a_1\rangle$, x, p, and q must have unique labels at a_1. Let ℓ_x, ℓ_p, and ℓ_q denote the labels of x, p and q at a_1, respectively. In this case, t_1 is obtained from t as follows:

1. Add the following sequence operations immediately above a_1:

1.1. An η_{ℓ_x,ℓ_p}-operation which connects x to p.

1.2. A $\rho_{\ell_p\to\ell_q}$-operation which renames the label of p to the label of q.

2. Omit q.

As in the previous cases it is easy to see that t_1 defines G_1^* and p is the new special vertex s.

Case 1.2: suppose that p and q occur on different sides of a.

Case 1.2.1: suppose p occurs on the left side of a and q occurs on the right side of a. It is easy to see that at least one of p and q must have a unique label at a. Assume without loss of generality that q has a unique label (say ℓ_q) at a. Let ℓ_p and ℓ_y denote the labels that p and y have at a, respectively. Note that y is the only vertex which can have the same label as p at a. In this case, t_1 is obtained from t as follows:

1. Make changes to t such that y will have label ℓ_q at a. In particular let c be the lowest $\oplus$-operation in $\mathsf{tree}(t)$ which contains both y and q. Add a ρ-operation immediately above c which renames the label of y at c to the label of q at c (say ℓ_q). Then follow the path from c to a in $\mathsf{tree}(t)$ and for each node d corresponding to an η_{ℓ_1,ℓ_2}-operation such that y has label ℓ_1 at d, add an η_{ℓ_q,ℓ_2}-operation immediately above d. Thus, after this step y is connected to all the vertices (except q) which it was connected in $\mathsf{val}(t\langle a\rangle)$ and has label ℓ_q at a.

2. Omit q.

3. After the above changes to y, the label ℓ_p of p at a is unique. Add the following sequence of operations immediately above a:

3.1. An η_{ℓ_p,ℓ_q}-operation which connects y to p.

3.2. A $\rho_{\ell_q\to\ell_y}$-operation which renames y to the label it has in $\mathsf{val}(t\langle a\rangle)$.

By steps 1 and 3.2 above it is clear that all the vertices (except q) which are connected to y in t are also connected to y in t_1. Thus, t_1 defines G_1^* and p is the new special vertex s.

Case 1.2.2: suppose p occurs on the right side of a and q occurs on the left side of a. Since p is adjacent just to x and q, it follows that either x and q have unique labels at a or have the same label at a. If x and q have the same label at a, then there is no way to connect y to q without connecting it also to x, a contradiction. We conclude that the labels at a of p, q, x, and y (say ℓ_p, ℓ_q, ℓ_x and ℓ_y, respectively) are unique. In this case t_1 is obtained from t by omitting q and adding an η_{ℓ_p,ℓ_y}-operation immediately above a.

Case 2: suppose x and y occur on the same side of a. Assume without loss of generality that x and y occur on the left side of a.

Case 2.1: suppose p and q occur on the same side of a. Since a is the lowest node in $\mathsf{tree}(t)$ which contains x, y, p, and q, it follows that p and q must occur on the right side of a. As in case 1.2.2 it is easy to see that the labels at a of p, q, x and y (say ℓ_p, ℓ_q, ℓ_x, and ℓ_y) are unique. In this case t_1 is obtained from t by omitting q and adding an η_{ℓ_p,ℓ_y}-operation immediately above a.

Case 2.2: suppose p and q occur on different sides of a. Assume without loss of generality that p occurs on the left side of a and q occurs on the right side of a. Let a_1 denote the lowest node in $\mathsf{tree}(t)$ which contains both x and p. Let a_2 denote the lowest node in $\mathsf{tree}(t)$ which contains x and y.

Case 2.2.1: suppose a_1 is equal to a_2 or a_2 is a proper descendant of a_1. In this case it is easy to see that x, y and p must have unique labels at a_1 (say ℓ_x, ℓ_y, and ℓ_p, respectively). In this case t_1 is obtained from t by omitting q and adding an η_{ℓ_p,ℓ_y}-operation immediately above a_1.

Case 2.2.2: suppose a_1 is a proper descendant of a_2.

Case 2.2.2.1: suppose y has unique label at a_2 (say ℓ_y). In this case p must have unique label at a_2 (say ℓ_p) and t_1 is obtained from t by omitting q and adding an η_{ℓ_p,ℓ_y}-operation immediately above a_2.

Case 2.2.2.2: suppose y does not have unique label at a_2. Let ℓ_p and ℓ_y denote the labels of p and y at a_2, respectively. Since q is adjacent just to y and p, it follows that p is the only vertex which can share the label of y at a_2. Thus, $\ell_p = \ell_y$. Assume without loss of generality that y is on the right side of a_2 and x and p are on the left side of a_2. Let b_2 denote the right child of a_2 in $\mathsf{tree}(t)$. Note that the complicated handling of this case (as described below) is needed when x is active at a_2 and has the same label as another vertex which is on the right side of a_2. Since q is the only vertex which is adjacent to y and p, it follows that all the vertices which are adjacent to y (except q) must be in $\mathsf{val}(t\langle b_2\rangle)$. Let U

denote the set of all vertices (except q) which are adjacent to y. Since y is regular vertex, all vertices in U must be special and have degree exactly 2. For each vertex u in U, let $\mathsf{other}(u)$ denote the neighbor of u which is not y. Let U_1 denote the set of all vertices u in U such that $\mathsf{other}(u)$ is in $\mathsf{val}(t\langle b_2\rangle)$ and let $U_2 = U \setminus U_1$. Let U_{11} denote the set of all vertices u in U_1 such that the lowest node in $\mathsf{tree}(t)$ which contains u and $\mathsf{other}(u)$ does not contain y. Let $U_{12} = U_1 \setminus U_{11}$.

In this case t_1 is obtained from t as follows:

1. Omit q and all vertices of U_2.

2. Let c denote the lowest node in $\mathsf{tree}(t)$ which contains y. Follow the path from c to b_2 in $\mathsf{tree}(t)$ and omit any η_{ℓ_1,ℓ_2}-operation such that the label of y at that point is ℓ_1.

3. Repeat the following step for each u in U_{11}: let c denote the lowest node in $\mathsf{tree}(t)$ which contains u and $\mathsf{other}(u)$. Let d denote the lowest node in $\mathsf{tree}(t)$ which contains y and u. Since u is in U_{11}, c is a descendant of d. Thus, u and $\mathsf{other}(u)$ have unique labels at c (say ℓ_u and ℓ, respectively). Add an $\eta_{\ell_u,\ell}$-operation immediately above c which connects u and $\mathsf{other}(u)$. Add a ρ-operation immediately above d which renames the label of u to the label of y at d. Thus, after step 3 each vertex u in U_{11} is connected to $\mathsf{other}(u)$ and has label ℓ_y at a_2.

4. Repeat the following step for each u in U_{12}: let c denote the lowest node in $\mathsf{tree}(t)$ which contains u and $\mathsf{other}(u)$.

4.1. Suppose $\mathsf{other}(u)$ is a special vertex. If $\mathsf{other}(u)$ does not have a unique label at c then its label at c must be equal to the label of y at c, a contradiction, since q distinguishes y and $\mathsf{other}(u)$. Thus, $\mathsf{other}(u)$ must have unique label at c. If u does not have unique label at c, then the label of u at c must be equal to the label of the unique regular vertex (say z) which is adjacent to $\mathsf{other}(u)$. But then vertices of the induced path $z - \mathsf{other}(u) - u - y$ of G'' occur at a_2, and since a_2 is a descendent of a, we have a contradiction to the selection of a as a lowest such node with that property. We conclude that both u and $\mathsf{other}(u)$ have unique labels at c. Thus, in this case add an η-operation immediately above c connecting u and $\mathsf{other}(u)$ and above it add a ρ-operation which renames the label of u to the label that y has at that point.

4.2. Suppose $\mathsf{other}(u)$ is a regular vertex. Since t has Property 2, it follows that label 2 is not used at c. In this case omit u from t and add the following sequence of operations immediately above c:

4.2.1. A 1-$\oplus$-operation introducing u with label 2.

4.2.2. An $\eta_{2,\ell}$-operation, where ℓ is the unique label that $\mathsf{other}(u)$ has at c.

4.2.3. A $\rho_{2\to\ell'}$-operation where ℓ' is the unique label that y has at c.

Thus, after step 4 each vertex u in U_{12} is connected to $\mathsf{other}(u)$ and has label ℓ_y at a_2.

5. Omit y from t and add the following sequence of operations immediately above a_2:

5.1. A 1-$\oplus$-operation which introduces y with label 3. Note that since t has Property 2 label 3 is not used at a_2.

5.2. An η_{3,ℓ_y}-operation connecting y to p and all the vertices in U_1.

5.3. A $\rho_{\ell_y\to 1}$-operation renaming p and all the vertices in U_1 to a dead label.

5.4. For each vertex u in U_2 add the following sequence of operations:

5.4.1. A 1-$\oplus$-operation introducing u with label 2.

5.4.2. An $\eta_{2,3}$-operation connecting u and y.

5.4.3. A $\rho_{2\to\ell}$-operation where ℓ is the label that u has in t at a_2.

Thus after step 5.4 all the vertices in U_2 are connected to y and have the same label as they have in t at a_2.

5.5. A $\rho_{3\to 1}$-operation renaming the label of y to a dead label.

Each vertex u in U_1 is connected to $\mathsf{other}(u)$ in step 3 or in step 4 and is connected to y in step 5.2. Each vertex u in U_2 is connected to y at step 5.4.2 and the η-operation in t

above a_2 which connects u to $\mathsf{other}(u)$ also exists in t_1 and connects u to $\mathsf{other}(u)$ since after step 5.4 the label of u is the same as its label at a_2 in t.

Thus, t_1 defines G_1^* and p is the new special vertex s.

This completes the proof of Lemma 6. □

Proof of Lemma 5. Suppose $\mathrm{cwd}(G') = k$. Let G'_1 denote the induced subgraph of G' obtained by removing from G' for every edge $e = xy$ of G, the two pairs of vertices p_i, q_i, $i = 1, 2$ where $x - p_i - q_i - y$, $i = 1, 2$ are two of the three paths of length 3 between x and y. Since G'_1 is an induced subgraph of G', it follows that $\mathrm{cwd}(G'_1) \leq k$. Clearly, G'_1 belongs to $D(G)$. Let t be a k-expression which defines G'_1. By Observation 1, there is a $(k+3)$-expression t' defining G'_1 which has Property 2. Let a be a lowest node in $\mathsf{tree}(t')$ such that for an induced path $x - p - q - y$ of G'' (x and y are regular vertices) the vertices x, p, q, y occur at a. By Lemma 6 there exists a $(k+3)$-expression t'_1 which has Property 2 and defines the graph G_1^* obtained from G'_1 by replacing the path $x - p - q - y$ with a path $x - s - y$ where s is a new special vertex. We can repeat this process until we finally get a $(k+3)$-expression t'' which defines the graph G'' that is obtained from G'_1 by replacing all induced paths of length 3 (with regular end vertices and special internal vertices) by induced paths of length 2. This completes the proof of Lemma 5. □

3 Cwd-expressions for G''

Theorem 3. *If G is a cobipartite graph with minimum degree at least* 2, *then* $\mathrm{cwd}_1(G'') \leq \mathrm{cwd}(G'') + 6$.

For the proof of Theorem 3 we shall use the following definitions and lemmas.

In this section we assume that G is a cobipartite graph with minimum degree at least 2. Since G is cobipartite the vertices of G can be partitioned into two cliques A and B. The regular vertices of G'' which belong to A, B are called *A-regular* vertices, *B-regular vertices*, respectively.

Let t be a cwd-expression defining G''. Let a be a $\oplus$-operation of t. We say that there is a *separation* at a between the A-regular vertices and the B-regular vertices if all A-regular vertices of $\mathsf{val}(t\langle a\rangle)$ occur on one side of a (say, on the left side of a) and all the B-regular vertices of $\mathsf{val}(t\langle a\rangle)$ occur on the other side of a (say, on the right side of a).

Proposition 1. *Let t be a* cwd*-expression defining G''. For each $\oplus$-operation a of t there is at most one pair of A-regular (B-regular) vertices which occur on different sides of a and have the same label at a.*

Proof. Suppose there are two different pairs $\{x_1, y_1\}$ and $\{x_2, y_2\}$ of A-regular vertices such that for $i = 1, 2$, x_i and y_i occur at different sides of a and have the same label at a. Assume without loss of generality that x_1 and x_2 occur on the left side of a and y_1 and y_2 occur on the right side of a. Clearly, either $x_1 \neq x_2$ or $y_1 \neq y_2$. Assume without loss of generality that $x_1 \neq x_2$. Consider the special vertex s_{y_1,x_2}. If s_{y_1,x_2} is not in $\mathsf{val}(t\langle a\rangle)$, then when later on the edge connecting s_{y_1,x_2} to y_1 will be establish, also the edge connecting it to x_1 will be established, a contradiction. Thus s_{y_1,x_2} is in $\mathsf{val}(t\langle a\rangle)$. If s_{y_1,x_2} occurs on the left side of a then when the edge connecting it to y_1 will be established, it will be connected also to x_1, a contradiction. If s_{y_1,x_2} is on the right side of a, then when the edge connecting it to x_2 will be established, it will be connected also to y_2. Since the degree of s_{y_1,x_2} in G'' is exactly 2, it follows that y_1 must be equal to y_2. Thus, the three vertices x_1, x_2 and y_1 have the same label at a, which implies that the η-operation above a which connect s_{y_1,x_2} to x_2 connect it also to x_1, a contradiction. The argument for two different pairs of B-regular vertices is symmetric. □

Proposition 2. *Let t be a* cwd*-expression defining G''. Let a be a $\oplus$-operation of t and let $\{x_1, y_1\}$ be a pair of A-regular (B-regular) vertices which occur on different sides of a and have the same label at a. Then both x_1 and y_1 are active at a and for every other vertex (say z) occurring at a the label of z is different from the label of x_1 and y_1 at a.*

Proof. Since x_1 and y_1 have the same label at a, either they are both dead at a or they are both active at a. Suppose x_1 and y_1 are dead at a. Consider s_{x_1,y_1}. If s_{x_1,y_1} is not in $\mathsf{val}(t\langle a\rangle)$, then it is not possible to connect it to x_1 and y_1 (as they are dead at a), a contradiction. Assume without loss of generality that x_1 and s_{x_1,y_1} are on the same side of a. Since y_1 is on the other side of a, and y_1 is dead at a, it is not possible to connect s_{x_1,y_1} to y_1, a contradiction. We have shown that both x_1 and y_1 are active at a. If there is another vertex z with the same label as x_1 and y_1 at a, then, when the edges connecting some vertex of G'' (say, w) to x_1 and y_1 will be established (such edges must be established since x_1 and y_1 are active at a), also the edge connecting it to z will be established, a contradiction (no vertex of G'' is adjacent to x_1, y_1 and z). ☐

Proposition 3. *Let t be a* cwd*-expression defining G''. Let a be an $\oplus$-operation of t and let $\{x_1, y_1\}$ be a pair of regular vertices which occur on different sides of a and have the same label at a. Then all the edges connecting x_1 (y_1) to its neighbors in $G'' - s_{x_1,y_1}$ exist in $\mathsf{val}(t\langle a\rangle)$.*

Proof. Let s be a vertex which is adjacent to x_1 in $G'' - s_{x_1,y_1}$. Clearly s must be a special vertex of the form $s_{x_1,z}$ for $z \neq y_1$. If s is not connected to x_1 in $\mathsf{val}(t\langle a\rangle)$, then it is not possible to connect s to x_1 without connecting it also to y_1, a contradiction. ☐

3.1 Property 3

Property 3. We say that t has *Property 3* if the following conditions hold for t:

Condition 3.1: The label 1 is dead in t.

Condition 3.2: For each (> 1)-$\oplus$-operation a in t, there is no pair of A-regular (B-regular) vertices which occur on different sides of a and have the same label at a.

Lemma 7. *Let t be a k-expression defining G''. Then there exists a $(k + 4)$-expression t' defining G'' such that t' has Property 3.*

Proof. Let t be a k-expression defining G''. Let t_1 denote the $(k + 1)$-expression obtained from t by replacing each occurrence of the label 1 with the label $k+1$. Clearly, t_1 defines G'' and label 1 is dead in t_1. Let a be a (> 1)-$\oplus$-operation in t_1 such that there exist at least one pair of regular vertices that violate Condition 3.2. We define below a $(k + 4)$-expression t_2 which defines G'' and has the additional property that there is no pair of regular vertices of the same type which occur on different sides of a and have the same label in $\mathsf{val}(t_2\langle a\rangle)$. Let b denote the left child of a in $\mathsf{tree}(t)$.

Case 1: Suppose there is exactly one pair (say $\{x_1, y_1\}$) of regular vertices of the same type which occur on different sides of a and have the same label in $\mathsf{val}(t_1\langle a\rangle)$. Assume without loss of generality that x_1 occurs on the left side of a. By Proposition 2, both x_1 and y_1 must be active at a and their label at a (say ℓ) is different from the labels of all the other vertices at a. In this case t_2 is obtained from t_1 as follows:

1. Add a $\rho_{\ell\to k+2}$-operation immediately above b.
2. Omit s_{x_1,y_1}.
3. Add the following sequence of operations immediately above a:

3.1. A 1-$\oplus$-operation introducing s_{x_1,y_1} with label $k + 4$.
3.2. An $\eta_{k+4,\ell}$-operation which connects s_{x_1,y_1} to y_1.
3.3. An $\eta_{k+4,k+2}$-operation which connects s_{x_1,y_1} to x_1.

3.4 A $\rho_{k+4\to 1}$-operation renaming the label of s_{x_1,y_1} to a dead label.
3.5 A $\rho_{k+2\to 1}$-operation renaming the label of x_1 to a dead label.
3.6 A $\rho_{\ell\to 1}$-operation renaming the label of y_1 to a dead label.

Case 2: Suppose there are exactly two pairs (say $\{x_1, y_1\}$ and $\{x_2, y_2\}$) of regular vertices of the same type which occur on different sides of a and have the same label in $\mathsf{val}(t_1\langle a\rangle)$. Assume without loss of generality that x_1 and x_2 occur on the left side of a. By Proposition 2, both x_1 and y_1 must be active at a and their label at a (say ℓ_1) is different from the labels of all the other vertices at a. Similarly, x_2 and y_2 have the same unique label at a (say ℓ_2). It follows that all the vertices x_1, x_2, y_1, y_2 are distinct.

In this case t_2 is obtained from t_1 as follows:
1. Add the following sequence of operations immediately above b:
1.1 A $\rho_{\ell_1\to k+2}$-operation renaming the label of x_1 to to $k+2$.
1.1 A $\rho_{\ell_2\to k+3}$-operation renaming the label of x_2 to to $k+3$.
2. Omit s_{x_1,y_1} and s_{x_2,y_2}.
3. Add the following sequence of operations immediately above a:
3.1. A 1-$\oplus$-operation introducing s_{x_1,y_1} with label $k+4$.
3.2. An η_{k+4,ℓ_1}-operation which connects s_{x_1,y_1} to y_1.
3.3. An $\eta_{k+4,k+2}$-operation which connects s_{x_1,y_1} to x_1.
3.4 A $\rho_{k+4\to 1}$-operation renaming the label of s_{x_1,y_1} to a dead label.
3.5. A 1-$\oplus$-operation introducing s_{x_2,y_2} with label $k+4$.
3.6. An η_{k+4,ℓ_2}-operation which connects s_{x_2,y_2} to y_2.
3.7. An $\eta_{k+4,k+3}$-operation which connects s_{x_2,y_2} to x_2.
3.8 A sequence of ρ-operations renaming all labels $\ell_1, \ell_2, k+2, k+3, k+4$, to the dead label 1.

In both cases 1 and 2 it follows from Proposition 3 that the expression t_2 defines G''.

Repeating the above procedure for every (>1)-$\oplus$-operation in t_2 we finally get a $(k+4)$-expression t' defining G'' such that t' has Property 3. ☐

3.2 Property 4

The following property is similar to Property 2.

Property 4. Let t be a k-expression defining G'' which has Property 3. We say that t has *Property 4*, if the following conditions hold:

Condition 4.1: if label 2 is used in t, then it is used as follows: a special vertex (say s) is introduced with label 2 using a 1-$\oplus$-operation say a, such that s is the only vertex having label 2 at a. Above a in $\mathsf{tree}(t)$ there is a sequence of one or more η-operations followed by a $\rho_{2\to\ell}$-operation where ℓ is any label different from 2 and 3.

Condition 4.2: if label 3 is used in t then it is used as follows: a regular vertex (say r) is introduced with label 3 using a 1-$\oplus$-operation, say a, such that r is the only vertex having label 3 at a. Above a in $\mathsf{tree}(t)$ there is a sequence of operations which can be either η, ρ, or 1-$\oplus$-operations introducing special vertices, followed by a $\rho_{3\to\ell}$-operation where ℓ is any label different from 2 and 3.

Condition 4.3: no regular vertex ever gets label 2 and no special vertex ever gets label 3.

Lemma 8. *Let t be a k-expression defining G'' such that t has Property 3. Then there exists a $(k+2)$-expression t' defining G'' such that t' has Property 4.*

Proof. Let t be a k-expression defining G'' such that t has Property 3. Let t' denote the $(k+2)$-expression obtained from t by replacing each occurrence of the label 2 with the label $k+1$ and replacing each occurrence of the label 3 with the label $k+2$. Clearly, t' defines G''. Since labels 2 and 3 are not used in t', it is obvious that t' has Property 4. ☐

3.3 Property 5

Property 5. Let t be a k-expression defining G'' which has Property 4. We say that t has *Property 5*, if the following condition holds:

Condition 5: For each (>1)-$\oplus$-operation a in t, there is no regular vertex which occurs at a and has a unique label at a which is different from label 1.

Lemma 9. *Let t be a k-expression defining G'' such that t has Property 4. Then there exists a k-expression t' defining G'' such that t' has Property 5.*

For proving this lemma we use the following definitions and auxiliary results. Let t be a k-expression defining G''. For each (>1)-$\oplus$-operation a in t let $n(t\langle a\rangle)$ denote the number of regular vertices which occur at a and have unique labels at a which are different from label 1. Let $n(t)$ denote the sum of $n(t\langle a\rangle)$ over all (>1)-$\oplus$-operations in t. Clearly, if a k-expression t defines G'' and has Property 4, then $n(t)=0$ implies that t has also Property 5.

Lemma 10. *Let t be a k-expression defining G'' such that t has Property 4 and $n(t)>0$. Then there exists a k-expression t' defining G'' such that t' has Property 4 and $n(t')<n(t)$.*

Proof. Let t be a k-expression defining G'' such that t has Property 4 and $n(t)>0$. Since $n(t)>0$, there exists a (>1)-$\oplus$-operation a in t and a regular vertex x such that x has unique label (say ℓ_x) in $\mathsf{val}(t\langle a\rangle)$. We will construct below a k-expression t' defining G'', such that in t', x is introduced by a 1-$\oplus$-operation above a. We shall use the following notation and proceed similarly as in the proof of Lemma 6. Let b denote the child of a in $\mathsf{tree}(t)$ such that x is in $\mathsf{val}(t\langle b\rangle)$. Let U denote the set of all vertices which are adjacent to x and occur in $\mathsf{val}(t\langle b\rangle)$. Since x is a regular vertex, all vertices in U must be special and have degree exactly 2. For each vertex $u\in U$, let $\mathsf{other}(u)$ denote the neighbor of u which is not x. Let U_1 denote the set of all vertices $u\in U$ such that $\mathsf{other}(u)$ is in $\mathsf{val}(t\langle b\rangle)$ and let $U_2=U\setminus U_1$. Let U_{11} denote the set of all vertices $u\in U_1$ such that the lowest node in $\mathsf{tree}(t)$ which contains u and $\mathsf{other}(u)$ does not contain x. Let $U_{12}=U_1\setminus U_{11}$. The k-expression t' is obtained from t as follows:

1. Omit all vertices of U_2.

2. Let c denote the lowest node in $\mathsf{tree}(t)$ which contains x. Follow the path from c to b in $\mathsf{tree}(t)$ and omit any η_{ℓ_1,ℓ_2}-operation such that the label of x at that point is ℓ_1.

3. Repeat the following step for each $u\in U_{11}$: let d denote the lowest node in $\mathsf{tree}(t)$ which contains u and $\mathsf{other}(u)$. Let e denote the lowest node in $\mathsf{tree}(t)$ which contains x and u. Since u is in U_{11}, d is a descendant of e. Thus, u and $\mathsf{other}(u)$ have unique labels at d (say ℓ_u and ℓ, respectively). Add an $\eta_{\ell_u,\ell}$-operation immediately above d which connects u and $\mathsf{other}(u)$. Add a ρ-operation immediately above e which renames the label of u to the label of x at e. Thus, after step 3 each vertex $u\in U_{11}$ is connected to $\mathsf{other}(u)$ and has label ℓ_x at a.

4. Repeat the following step for each $u\in U_{12}$: let d denote the lowest node in $\mathsf{tree}(t)$ which contains u and $\mathsf{other}(u)$. Since t has Property 4, and u and $\mathsf{other}(u)$ occur on different sides of d it follows that the only vertex which can have label 2 at d is u. Omit u from t and add the following sequence of operations immediately above d:

4.1. A 1-$\oplus$-operation introducing u with label 2.

4.2. An $\eta_{2,\ell}$-operation connecting u and $\mathsf{other}(u)$, where ℓ is the unique label that $\mathsf{other}(u)$ has at d.

4.2.3. A $\rho_{2\to\ell'}$-operation where ℓ' is the unique label that x has at d.

Thus, after step 4 each vertex $u\in U_{12}$ is connected to $\mathsf{other}(u)$ and has label ℓ_x at a.

5. Omit x from t and add the following sequence of operations immediately above a:

5.1. A 1-$\oplus$-operation which introduces x with label 3. Note that since t has Property 4 and a is a (>1)-$\oplus$-operation label 3 is not used at a.

5.2. An η_{3,ℓ_x}-operation connecting x to all the vertices in U_1.

5.3. A $\rho_{\ell_x \to 1}$-operation renaming the label of all the vertices in U_1 to a dead label.

5.4. For each vertex $u \in U_2$ add the following sequence of operations:

5.4.1. a 1-$\oplus$-operation introducing u with label 2;

5.4.2. an $\eta_{2,3}$-operation connecting u to x;

5.4.3. a $\rho_{2\to\ell}$-operation where ℓ is the label that u has in t at a.

Thus after step 5.4 all the vertices in U_2 are connected to x and have the same label as they have in t at a.

5.5. A $\rho_{3\to\ell_x}$-operation renaming the label of x to the label it has in $\mathsf{val}(t\langle a\rangle)$.

Each vertex $u \in U_1$ is connected to $\mathsf{other}(u)$ in step 3 or in step 4 and is connected to x in step 5.2. Each vertex $u \in U_2$ is connected to x at step 5.4.2 and the η-operation in t above a which connects u to $\mathsf{other}(u)$ also exists in t' and connects u to $\mathsf{other}(u)$. Since after step 5.5. the label of x is the same as its label in $\mathsf{val}(t\langle a\rangle)$, it follows that all the vertices which are adjacent to x and are not in U will be connected to x in t' by the same η-operations which connect them to x in t.

Thus, t' defines G''. Since the above changes to t did not violate the rules of Property 4, it follows that t' has Property 4. Finally, since in t', x is introduced by a 1-$\oplus$-operation above a, and all other regular vertices are not moved, it follows that $n(t') < n(t)$. This completes the proof of Lemma 10. □

Proof of Lemma 9. Follows easily by applying Lemma 10 (at most) $n(t)$ times until a k-expression t' is obtained such that t' defines G'' and $n(t') = 0$. □

Proposition 4. *Let t be a k-expression defining G'' such that t has Property 5. Let a be a (> 1)-$\oplus$-operation in t such that at least one regular vertex occurs on the left side of a and at least one regular vertex occurs on the right side of a. Then there is a separation at a between the A-regular and the B-regular vertices.*

Proof. Let a be a (> 1)-$\oplus$-operation in t and let x and y be two regular vertices occurring on different sides of a. Assume without loss of generality that x occurs on the left side of a and y occurs on the right side of a. Suppose x and y are both A-regular vertices. By Condition 3.2, x and y do not have the same label at a. Suppose x or y (say x) has label 1 at a. By Condition 5, there exists vertex z which have the same label as y at a. The special vertex $s = s_{x,y}$ must occur on the left side of a, or else no η-operation connect s and x in t, a contradiction. Thus, the η-operation above a in $\mathsf{tree}(t)$ which connects s to y connects it also to z, a contradiction. We conclude that both x and y do not have label 1 at a. By Condition 5, there are two vertices w and z which have the same label as x and y at a, respectively. Let $s = s_{x,y}$. If s does not occur at a, then the η-operation in t which connects s to x, connects it also to w, a contradiction. If s occurs on the left side of a, then the η-operation which connects s to y connects it also to z, a contradiction. If s occurs on the right side of a, then the η-operation which connects s to x connects it also to w, a contradiction. Thus x and y can not be both A-regular vertices.

Similarly, x and y cannot be both B-regular vertices. Thus, one of x and y (say, x) must be A-regular and the other (say, y) must be B-regular. If there is a B-regular vertex (say, z) on the left side then there are two B-regular vertices (z and y) occurring on different sides of a, which is not possible by the above argument. Thus all the A-regular vertices occur on the left side of a and all the B-regular vertices occur on the right side of a. □

3.4 Property 6

Property 6. Let t be a k-expression defining G''. We say that t has *Property 6* if it has Property 5 and the following condition holds:

Condition 6: Either there are no (>1)-$\oplus$-operations in t or there is just one (>1)-$\oplus$-operation in t (say, a) and there is a separation at a between the A-regular and the B-regular vertices.

Lemma 11. *Let t be a k-expression defining G'' such that t has Property 5. Then there exists a k-expression t' which defines G'' and has Property 6.*

Proof. Let t be a k-expression which defines G'' and has Property 5. Let a be a (>1)-$\oplus$-operation in t such that one side of a (say, the left side) contains just special vertices (say, $s_1, \dots, s_m$). Clearly, $s_1, \dots, s_m$ are isolated vertices in $\mathsf{val}(t\langle a\rangle)$ and have unique labels in $\mathsf{val}(t\langle a\rangle)$. Let $\ell_1, \dots, \ell_m$ denote the labels of $s_1, \dots, s_m$ in $\mathsf{val}(t\langle a\rangle)$, respectively. Let b be the right child of a. Let t_1 be the expression obtained from t by replacing $t\langle a\rangle$ with

$$t\langle b\rangle \oplus \ell_1(s_1) \oplus \cdots \oplus \ell_m(s_m).$$

It is easy to verify that t_1 also defines G'' and has Property 5.

Let t' denote the expression obtained from t_1 by repeating the above process for each (>1)-$\oplus$-operation a in t_1 such that one side of a contains just special vertices. Let a be a (>1)-$\oplus$-operation in t'. By the above construction, each side of a contains at least one regular vertex. By Proposition 4, since Property 5 holds for t', there is a separation at a in t' between the A-regular vertices and the B-regular vertices. Suppose there is another (>1)-$\oplus$-operation (say a') in t'. By the above argument each side of a' contains at least one regular vertex and there is a separation at a' in t' between the A-regular and the B-regular vertices. If a is a descendant of a' in $\mathsf{tree}(t')$, then there cannot be a separation at a' between the A-regular and the B-regular vertices, a contradiction. Similarly, a' is not a descendant of a in $\mathsf{tree}(t')$. Let a'' be the lowest node in $\mathsf{tree}(t')$ which contains both a and a'. Clearly a'' must be a (>1)-$\oplus$-operation. By Proposition 4 there is a separation at a'' in t' between the A-regular and the B-regular vertices. Since a occurs on one side of a'', this side of a'' contains both A-regular and B-regular vertices, a contradiction. We conclude that a is a unique (>1)-$\oplus$-operation in t'. Thus t' is a k-expression which defines G'' and has Property 6. □

3.5 Property 7

Property 7. Let t be a k-expression defining G''. We say that t has *Property 7* if it has *Property 6* and either t is sequential or the following condition holds:

Condition 7: Let a be the unique (>1)-$\oplus$-operation in t. Then for each A-regular (B-regular) vertex x, which is active at a and occurs on one side (say left side) of a, there is a unique B-regular (A-regular) vertex y which is active at a and occurs on the other side (say right side) of a and has the same label as x in $\mathsf{val}(t\langle a\rangle)$.

Lemma 12. *Let t be a k-expression defining G'' such that t has Property 6. Then there exists a k-expression t' which defines G'' and has Property 7.*

Proof. Let a be the unique (>1)-$\oplus$-operation in t. Assume without loss of generality that all the A-regular vertices of $\mathsf{val}(t\langle a\rangle)$ occur on the left side of a and all the B-regular vertices of $\mathsf{val}(t\langle a\rangle)$ occur on the right side of a. Let x be a regular vertex which is active at a. Let ℓ denote the label of x at a. Since Condition 5 holds for t, the label of x at a is not unique. Suppose there are two vertices u and v which are distinct from x and have label ℓ at a. Since x is active at a, there is an η-operation above a in $\mathsf{tree}(t)$ which connects some special vertex (say, s) to x. This η-operation connects s also to u and v, a contradiction (since s is adjacent in G'' to exactly two vertices). Thus, for each regular vertex x which has label ℓ at a and is active at a there is a unique second vertex (say y) which is active at a and has label ℓ at a.

By a similar argument no η-operation above a in $\mathsf{tree}(t)$ connects a vertex other than $s_{x,y}$ to x or to y. Thus, all edges incident to x or y in G'', except $xs_{x,y}$ and $ys_{x,y}$, already exist in $\mathsf{val}(t\langle a\rangle)$.

We now define the cwd-expression t_1 depending on the following cases:

Case 1: One of the vertices x, y is A-regular and one is B-regular. Since Condition 7 holds in this case for x and y we set $t_1 = t$.

Case 2: Both x and y are A-regular. Let b denote the left child of a. In this case t_1 is obtained from t as follows:

1. Omit $s_{x,y}$ from t.

2. Add immediately above b the following sequence of operations:

2.1. A 1-$\oplus$-operation which introduces $s_{x,y}$ with label 2. Note that since t has Property 2, and a is a (> 1)-$\oplus$-operation, label 2 is not used in $\mathsf{val}(t\langle a\rangle)$.

2.2. An $\eta_{2,\ell}$-operation which connects $s_{x,y}$ to x and y, where ℓ is the label that x and y have in $\mathsf{val}(t\langle b\rangle)$.

2.3. A $\rho_{2\to 1}$-operation renaming the label of $s_{x,y}$ to the dead label 1.

2.4. A $\rho_{\ell\to 1}$-operation renaming the label of x and y to the dead label 1.

Case 3: Both x and y are B-regular. This case is symmetric to Case 2.

Let t' denote the expression obtained by repeating the above process for each regular vertex which is active at a. It is easy to see that t' defines G'' and has Property 7, as required. □

3.6 Sequential expressions for G''

In the the proof of Lemma 13 we shall use the following definition and Proposition.

Let t be an expression which defines G'', let a be any node of $\mathsf{tree}(t)$ and let $s_{x,y}$ be any special vertex in $\mathsf{val}(t\langle a\rangle)$. The label of $s_{x,y}$ at a is called an *x-connecting label* at a (a *y-connecting label* at a) if $\mathsf{val}(t\langle a\rangle)$ includes the edge connecting $s_{x,y}$ to y (x) but does not include the edge connecting $s_{x,y}$ to x (y).

Proposition 5. *Let t be an expression which defines G'', let a be any node of $\mathsf{tree}(t)$, and let y_1, y_2 be two distinct regular vertices of G''. Suppose that there is a y_1-connecting label and a y_2-connecting label at a. Then these two labels are different.*

Proof. Let s_1 and s_2 be two special vertices that have a y_1-connecting label and a y_2-connecting label at a, respectively. By definition, s_1 is a special vertex of the form s_{x_1,y_1} where s_1 is connected to x_1 and is not connected to y_1 in $\mathsf{val}(t\langle a\rangle)$. Similarly, s_2 is a special vertex of the form s_{x_2,y_2} where s_2 is connected to x_2 and is not connected to y_2 in $\mathsf{val}(t\langle a\rangle)$. Suppose that the labels of s_1 and s_2 are the same in $\mathsf{val}(t\langle a\rangle)$. The η-operation above a which connects s_1 to y_1 connects also s_2 to y_1. Thus s_2 is connected to x_2, y_2 and y_1. Since $y_1 \neq y_2$ and $x_2 \neq y_2$ and s_2 has degree 2, it follows that $x_2 = y_1$. By a symmetric argument we get that x_1 is equal to y_2. We conclude that $s_1 = s_2$. But this is not possible since $s_1 = s_2$ is connected to x_1 and is not connected to $y_2 = x_1$. □

Lemma 13. *Let t be a k-expression defining G'' such that t has Property 7. Then there is a sequential k-expression which defines G''.*

Proof. If there is no (> 1)-$\oplus$-operation in t, the claim follows immediately. Let a be the unique (> 1)-$\oplus$-operation in t. Let b and c denote the left child and the right child of a in $\mathsf{tree}(t)$, respectively. Assume without loss of generality that all the regular vertices in $\mathsf{val}(t\langle b\rangle)$ are A-regular and all regular vertices in $\mathsf{val}(t\langle c\rangle)$ are B-regular.

First we introduce the following notation. Let A_1 (B_1) denote the set of A-regular (B-regular) vertices of $\mathsf{val}(t\langle b\rangle)$ ($\mathsf{val}(t\langle c\rangle)$) and put $A_2 = A \setminus A_1$ and $B_2 = B \setminus B_1$. Let $\mathsf{Active}(A_1)$ ($\mathsf{Active}(B_1)$) denote the set of vertices of A_1 (B_1) which are active at a. Let

$\mathsf{Dead}(A_1)$ ($\mathsf{Dead}(B_1)$) denote the set of vertices of A_1 (B_1) which are dead at a. Clearly, $A_1 = \mathsf{Active}(A_1) \cup \mathsf{Dead}(A_1)$ and $B_1 = \mathsf{Active}(B_1) \cup \mathsf{Dead}(B_1)$. By Condition 7, $|\mathsf{Active}(A_1)| = |\mathsf{Active}(B_1)|$. For each B-regular vertex $u \in \mathsf{Active}(B_1)$ we denote by $\mathsf{mate}(u)$ the unique A-regular vertex (guaranteed by Condition 7) which is in $\mathsf{Active}(A_1)$ and has the same label as u in $\mathsf{val}(t\langle a \rangle)$. Let $|\mathsf{Dead}(A_1)| = q$. Let x_i, $1 \leq i \leq q$, be the ith vertex in $\mathsf{Dead}(A_1)$ which gets a non-unique label or label 1 in $t\langle b \rangle$ (if there is more than one such vertex, choose one of them arbitrarily) and let w_i be the highest node in $\mathsf{tree}(t\langle b \rangle)$ such that x_i has a unique label (which is different from label 1) in $t\langle w_i \rangle$. Note that w_i is well defined since each regular vertex in G'' is a leaf of $\mathsf{tree}(t)$ having a unique initial label (which is different from label 1).

Let $X_i = \{x_1, \ldots, x_i\}$, $1 \leq i \leq q$. Let NX_i, $1 \leq i \leq q$, denote the set of B-regular vertices which have a neighbor (in G) in the set X_i. For convenience we set $NX_0 = \emptyset$.

Observation 2. *Let v be a vertex which is adjacent to x_i (in G) and is not in $\mathsf{val}(t\langle w_i \rangle)$. Then the special vertex $s_{x_i,v}$ has the v-connecting label at w_i.*

Proof of Observation 2. Suppose the vertex $s = s_{x_i,v}$ is not adjacent to x_i in $\mathsf{val}(t\langle w_i \rangle)$. Let w'_i denote the parent of w_i in $\mathsf{tree}(t)$. The label of x_i at w'_i is either 1 or the label of another vertex (say u)). If the label of x_i at w'_i is 1 then no η-operation in t connects s and x_i, a contradiction. Thus, the label of x_i is is the same as the label of u at w'_i. If $u \neq v$ then the η-operation above w'_i which connects s to x_i connects it also to u, a contradiction. If $u = v$ then w'_i must correspond to a 1-$\oplus$-operation which introduces v with the label of x_i. Since v and x_i have the same label at w'_i it follows that each neighbor of v is also a neighbor of x_i. However, since G has minimum degree at least 2, there is a neighbor of v in G'' which is not a neighbor of x_i, a contradiction. □

Observation 3. *For $1 \leq i \leq q$, $\mathsf{labels}(\mathsf{val}(t\langle w_i \rangle)) \geq |A| + |NX_i| + 1 - i$.*

Proof of Observation 3. Let v be a vertex in $\mathsf{Active}(A_1)$. If v occurs at w_i, then v has a unique label at $\mathsf{val}(t\langle w_i \rangle)$. If v does not occur at w_i, then by Observation 2 the vertex $s_{x_i,v}$ has a v-connecting label at w_i. Thus, so far we have $|\mathsf{Active}(A_1)|$ different labels in $\mathsf{val}(t\langle w_i \rangle)$. Let v be a vertex in $\mathsf{Dead}(A_1) \setminus X_i$. If v occurs at w_i, then by definition v must have a unique label at w_i. If v does not occur at w_i, then by Observation 2 the vertex $s_{x_i,v}$ has a v-connecting label at w_i. Thus, by Proposition 5, we have additional $|\mathsf{Dead}(A_1) \setminus X_i| = q - i$ labels in $\mathsf{val}(t\langle w_i \rangle)$. Let v be a vertex in A_2. By Observation 2, the vertex $s_{x_i,v}$ has the v-connecting label in $\mathsf{val}(t\langle w_i \rangle)$. Thus, additional $|A_2|$ labels exists in $\mathsf{val}(t\langle w_i \rangle)$. Let v be a vertex in NX_i. By definition there exists a vertex in X_i (say x_j) such that v is adjacent to x_j in G. By Observation 2, vertex $s_{x_j,v}$ has the v-connecting label at w_j. Since v is not in $\mathsf{val}(t\langle w_i \rangle)$, the vertex $s_{x_j,v}$ also has the v-connecting label in $\mathsf{val}(t\langle w_i \rangle)$. Thus, additional $|NX_i|$ labels exists in $\mathsf{val}(t\langle w_i \rangle)$. Finally, by definition x_i has a unique label at w_i. Summarizing all the labels counted so far gives $|\mathsf{Active}(A_1)| + |A_2| + |NX_i| + 1 + q - i = |A| + |NX_i| + 1 - i$. □

Since t has Properties 3 and 4 we may assume that the labels 1, 2, and 3 are already considered in the counting of the k labels of t. Since the labels 1, 2, and 3 are not counted in the formula of Observation 3, the next observation follows.

Observation 4. *For $1 \leq i \leq q$, $k \geq |A| + |NX_i| + 4 - i$.*

Observation 5. $k \geq |A| + 3$.

Proof of Observation 5. If $\mathsf{Dead}(A_1) \neq \emptyset$ the claim follows from Observation 4 for $i = 1$. Suppose $\mathsf{Dead}(A_1) = \emptyset$. Let x be any vertex of $\mathsf{Active}(A_1)$. For each vertex v in A_2 the vertex $s_{x,v}$ must have an x-connecting label at a. Thus, so far we have $|A_2|$ different labels at a. Since all the vertices in $\mathsf{Active}(A_1)$ have different labels at a we get $|A_2| + |\mathsf{Active}(A_1)| = |A|$ different labels at a. Since we did not count labels $1, 2$, and 3, the claim follows. □

Observation 6. $\mathsf{labels}(\mathsf{val}(t\langle a\rangle)) \geq |\mathsf{Active}(A_1)| + |A_2| + |B_2|$.

Proof of Observation 6. By Property 7, each vertex $v \in \mathsf{Active}(A_1)$ has a unique label in $\mathsf{val}(t\langle b\rangle)$. Thus there are at least $|\mathsf{Active}(A_1)|$ labels in $\mathsf{val}(t\langle a\rangle)$. Let v be a vertex in A_2 and let u be any vertex in A_1. First assume $u \in \mathsf{Dead}(A_1)$. If $s_{u,v}$ is not connected to u in $\mathsf{val}(t\langle a\rangle)$, there is no η-operation above a that will connect it to u, a contradiction. Now assume $u \in \mathsf{Active}(A_1)$. If $s_{u,v}$ is not connected to u in $\mathsf{val}(t\langle a\rangle)$, then an η-operation above a that connects $s_{u,v}$ to u connects it also to the vertex $x \in \mathsf{Active}(B_1)$ such that $u = \mathsf{mate}(x)$, a contradiction. Hence, in any case $s_{u,v}$ is connected to u and has the v-connecting label in $\mathsf{val}(t\langle a\rangle)$. Thus additional $|A_2|$ labels must exists in $\mathsf{val}(t\langle a\rangle)$. By symmetry, additional $|B_2|$ vertices must exists in $\mathsf{val}(t\langle a\rangle)$. ☐

Since labels 1, 2, and 3 are not counted in the formula of Observation 6 the next observation follows.

Observation 7. $k \geq |\mathsf{Active}(A_1)| + |A_2| + |B_2| + 3$.

Now we start the process of constructing a sequential k-expression which defines G''. At each step we show that no more than k labels are used. Moreover, the η-operations added at each step connect special vertices of the form $s_{x,y}$ to x and y, which implies that all edges added in the process belong to G''. Finally, we show in a sequence of observations that for each regular vertex x of G'' the edges which connect x to all its neighbors in G'' exist in the sequential cwd-expression that we construct. Thus this expression satisfies the conditions of the lemma.

Let e_1 denote the expression obtained from $t\langle c\rangle$ as follows:

1. Omit all the special vertices of the form $s_{x,y}$ such that both x and y do not occur in $\mathsf{val}(t\langle c\rangle)$.

2. Add immediately above c the following sequence of η-operations: for each special vertex $s = s_{x,y}$ such that s and x (y) occur in $\mathsf{val}(t\langle c\rangle)$ but are not adjacent in $\mathsf{val}(t\langle c\rangle)$, add an η-operation which connects s and x (y).

Observation 8. *For each vertex $u \in \mathsf{Dead}(B_1)$, $\mathsf{val}(e_1)$ includes all the edges connecting u to all its neighbors in G''.*

Proof of Observation 8. Let u be a vertex in $\mathsf{Dead}(B_1)$ and let s be a neighbor of u in G''. Clearly, s is a special vertex of the form $s = s_{u,v}$ where v is a regular vertex which is a neighbor of u in G. Suppose u is not adjacent to s in $\mathsf{val}(t\langle c\rangle)$. Since u has a dead label in $\mathsf{val}(t\langle c\rangle)$, it follows that u is not adjacent to s in $\mathsf{val}(t)$, a contradiction. Thus, u is adjacent to s in $\mathsf{val}(t\langle c\rangle)$, and therefore the special vertex s is not omitted in step 1 of the construction of e_1. Thus, u is adjacent to s in e_1. ☐

Let e_2 denote the expression obtained from e_1 as follows:

1. For each vertex x such that $\mathsf{val}(e_1)$ includes all the edges connecting x to all its neighbors in G'', add a ρ-operation which renames the label of x to the dead label 1.

2. Omit all the special vertices of the form $s_{x,y}$ such that $x \in \mathsf{Active}(B_1)$ and $y = \mathsf{mate}(x)$.

3. For each regular vertex $u \in \mathsf{Active}(B_1)$ add the following sequence of operations:

3.1. A ρ-operation which introduces $\mathsf{mate}(u)$ with label 3. Note that since t has Property 2, label 3 is not used in $\mathsf{val}(t\langle a\rangle)$, which implies that this label is not used at the root of e_1.

3.2. A 1-$\oplus$-operation which introduces $s = s_{u,\mathsf{mate}(u)}$ with label 2. Note that since t has Property 2, label 2 is not used in $\mathsf{val}(t\langle a\rangle)$, which implies that this label is not used at the root of e_1.

3.3. An $\eta_{2,3}$-operation which connects $\mathsf{mate}(u)$ and s.

3.4. An $\eta_{2,\ell}$-operation which connects u and s, where ℓ is the label that u has in $\mathsf{val}(t\langle a\rangle)$.

3.5. A $\rho_{2\to 1}$-operation renaming the label of s to the dead label 1.

3.6. A $\rho_{\ell\to 1}$-operation renaming the label of u to the dead label 1.

3.7. A $\rho_{3\to \ell}$-operation renaming the label of $\mathsf{mate}(u)$ to the label it has in $\mathsf{val}(t\langle a\rangle)$.

Observation 9. *For each vertex $u \in \mathsf{Active}(B_1)$, $\mathsf{val}(e_2)$ includes all the edges connecting u to all its neighbors in G''.*

Proof of Observation 9. Let $u \in \mathsf{Active}(B_1)$ and let s be a neighbor of u in G''. Clearly, s is a special vertex of the form $s = s_{u,v}$ where v is a regular vertex which is a neighbor of u in G. Suppose $v \neq \mathsf{mate}(u)$. If s is not in $\mathsf{val}(t\langle c\rangle)$ then the η-operation above c in $\mathsf{tree}(t)$ which connects s to u connects it also to $\mathsf{mate}(u)$, a contradiction. Thus, both s and u are in $\mathsf{val}(t\langle c\rangle)$. By step 2 of the construction of e_1, u and s are adjacent in $\mathsf{val}(e_2)$. Suppose $v = \mathsf{mate}(u)$. By step 3.4 of the construction of e_2, s and u are adjacent in $\mathsf{val}(e_2)$. ☐

Let e_3 denote the expression obtained from e_2 by adding the following sequence of operations immediately above the root of $\mathsf{tree}(e_2)$:

1. For each vertex $u \in A_2 \cup B_2$, if there is no u-connecting label in $\mathsf{val}(e_2)$, add a 1-$\oplus$-operation which introduces u with a unique label ℓ_u (distinct from 1, 2, and 3). Otherwise, let ℓ denote the u-connecting label in $\mathsf{val}(e_2)$ (note that we assume that the label ℓ is unique, otherwise we can add ρ-operations which unify all the u-connecting labels to a unique label), and add the following sequence of operations:

1.1. A 1-$\oplus$-operation which introduces u with label 3.

1.2. An $\eta_{3,\ell}$-operation which connects u to all the vertices having a u-connecting label in $\mathsf{val}(e_2)$.

1.3. A $\rho_{\ell\to 1}$-operation renaming label ℓ to the dead label 1.

1.4. A $\rho_{3\to \ell}$-operation renaming the label of u to ℓ.

2. For each special vertex $s = s_{x,y}$ such that both x and y are in $\mathsf{Active}(A_1) \cup A_2 \cup B_2$, add the following sequence of operations:

2.1. A 1-$\oplus$-operation which introduces s with label 2.

2.2. An η_{2,ℓ_x}-operation, which connects s to x, where ℓ_x is the (unique) label of x at that point.

2.3. An η_{2,ℓ_y}-operation, which connects s to y, where ℓ_y is the (unique) label of y at that point.

2.4. A $\rho_{2\to 1}$-operation renaming the label of s to the dead label 1.

3. For each regular vertex $u \in B_2 \setminus NX_q$, add a $\rho_{\ell_u\to 1}$-operation renaming the label of u to the dead label 1, where ℓ_u is the (unique) label that u has at that point.

Observation 10. *e_3 is a k-expression, and* $\mathsf{labels}(\mathsf{val}(e_3)) \leq |\mathsf{Active}(A_1)| + |NX_q| + |A_2| + 1$.

Proof of Observation 10. The expression e_1 is constructed from $t\langle c\rangle$ without adding new labels. The expression e_2 is constructed from e_1 using the labels of e_1 in addition to the labels 1, 2, and 3 which are already considered in counting the k labels of t. Thus, e_2 is a k-expression.

In the construction of e_3 from e_2 (described above) the highest number of labels used is immediately before the completion of step 2 (which is the same as the number of labels used immediately before the completion of step 1). At that point all the vertices in $\mathsf{Active}(A_1) \cup A_2 \cup B_2$ have unique labels, the vertices in B_1 have label 1, the last special vertex considered has label 2 and all the other special vertices have label 1. Thus the total number of labels used at that point is at most $|\mathsf{Active}(A_1)| + |A_2| + |B_2| + 2$ which, by Observation 7, is less than k. When step 2 is completed the number of labels is reduced by one, since the last special vertex considered gets label 1. After step 3 is completed the number of labels is reduced by $|B_2 \setminus NX_q|$. ☐

Let $f_0 = e_3$ and for $1 \le i \le q$ let f_i be the expression obtained by adding the following sequence of operations immediately above the root of $\mathsf{tree}(f_{i-1})$:

1. A 1-$\oplus$-operation which introduces $x_{q-(i-1)}$ with a unique label, denoted by $\ell(x_{q-(i-1)})$.

2. For each special vertex $s = s_{x,y}$ such that $x = x_{q-(i-1)}$ and y is in $NX_{q-(i-1)}$ add the following sequence of operations:

2.1. A 1-$\oplus$-operation which introduces s with label 2.

2.2. An $\eta_{2,\ell(x_{q-(i-1)})}$-operation, which connects s to $x_{q-(i-1)}$.

2.3. An η_{2,ℓ_y}-operation, which connects s to y, where ℓ_y is the (unique) label of y at that point.

2.4 A $\rho_{2\to 1}$-operation renaming the label of s to the dead label 1.

3. For each regular vertex $u \in NX_{q-(i-1)} \setminus NX_{q-i}$, add a $\rho_{\ell_u \to 1}$-operation renaming the label of u to the dead label, where ℓ_u is the (unique) labels that u has at that point.

Observation 11. *For each vertex $u \in B_2$, $\mathsf{val}(f_q)$ includes all the edges connecting u to all its neighbors in G''.*

Proof of Observation 11. Let u be a vertex in B_2 and let s be a neighbor of u in G''. Clearly, s is a special vertex of the form $s = s_{u,v}$ where v is a regular vertex which is a neighbor of u in G. If $v \in \mathsf{Active}(A_1) \cup A_2 \cup B_2$, then the s is connected to u by one of the two η-operations added in steps 2.2 and 2.3 of the construction of e_3. Suppose $v \in B_1$. By Observations 8 and 9, s is connected to v in $\mathsf{val}(e_2)$. Thus, s has a u-connecting label in $\mathsf{val}(e_2)$ and is connected to u in step 1.2 of the construction of e_3. The last case to consider is when v is in $\mathsf{Dead}(A_1)$. In this case $v = x_{q-(i-1)}$ for some $i \in \{1, \dots, q\}$ and u must be in $NX_{q-(i-1)}$. Thus, u (denoted as y) is connected to s in step 2.3 of the construction of f_i. ☐

Observation 12. *For $0 \le i \le q$, the f_i is a k-expression, and $\mathsf{labels}(\mathsf{val}(f_i)) \le |\mathsf{Active}(A_1)| + |A_2| + |NX_{q-i}| + 1 + i = |A| + |NX_{q-i}| + 1 - (q - i)$.*

Proof of Observation 12. The proof is by induction on i. For $i = 0$ the claim follows from Observation 10, hence assume $i > 0$. It follows by Observation 10 that the number of labels used in e_3 is at most k. The highest number of labels used in the construction of f_i from f_{i-1} is immediately after step 2.1 is completed. At that point the number of labels used is equal to $\mathsf{labels}(\mathsf{val}(f_{i-1}))$ plus one new label for $x_{q-(i-1)}$ plus the label 2 used for introducing the special vertex at step 2.1. By the inductive hypothesis this number is at most $|A| + |NX_{q-(i-1)}| + 3 - (q - (i - 1))$ which by Observation 4 is less than k. At the completion of step 2 of the construction of f_i the number of labels is reduced by one since the label 2 is renamed to 1. At the completion of step 3. the number of labels is reduced by $|NX_{q-(i-1)} \setminus NX_{q-(i)}|$ which gives the claimed formula for $\mathsf{labels}(\mathsf{val}(f_i))$. ☐

Let t' denote the expression obtained from f_q by adding the following sequence of operations immediately above the root of $\mathsf{tree}(f_q)$:

1. For each special vertex $s = s_{x,y}$ such that $x \in \mathsf{Dead}(A_1)$ and $y \in A$ add the following sequence of operations:

1.1. A 1-$\oplus$-operation which introduces s with label 2.

1.2. An η_{2,ℓ_x}-operation, which connects s to x, where ℓ_x is the unique label of x in $\mathsf{val}(f_q)$.

1.3. An η_{2,ℓ_y}-operation, which connects s to y, where ℓ_y is the unique label of y in $\mathsf{val}(f_q)$.

1.4. A $\rho_{2\to 1}$-operation renaming the label of s to the dead label 1.

Observation 13. *For each vertex $u \in A$, $\mathsf{val}(t')$ includes all the edges connecting u to all its neighbors in G''.*

Proof of Observation 13. Let u be a vertex in A and let s be a neighbor of u in G''. Clearly, s is a special vertex of the form $s = s_{u,v}$ where v is a regular vertex which is a neighbor of u in G. We consider the following cases:

Case 1: Suppose $u \in \mathsf{Active}(A_1)$. If $v \in \mathsf{Active}(A_1) \cup A_2 \cup B_2$, then u is connected to s in step 2.2 or step 2.3 of the construction of e_3. If $v \in \mathsf{Active}(B_1)$, then u must be equal to $\mathsf{mate}(v)$ and is connected to s in step 3.3 of the construction of e_2. If $v \in \mathsf{Dead}(A_1)$, then u (denoted as y) is connected to s in step 1.3 of the construction of t'. The last case to consider is when v is in $\mathsf{Dead}(B_1)$. In this case s must occur at c which implies that the η-operation above a in $\mathsf{tree}(t)$ which connects s to u also connects s to the vertex z such that $u = \mathsf{mate}(z)$, a contradiction. Thus, the case when v is in $\mathsf{Dead}(B_1)$ is not possible.

Case 2: Suppose $u \in A_2$. If $v \in \mathsf{Active}(A_1) \cup A_2 \cup B_2$, then u is connected to s in step 2.2 or step 2.3 of the construction of e_3. If $v \in B_1$, then s must have a u-connecting label in $\mathsf{val}(e_2)$ and is connected to u in step 1.2 of the construction of e_3. If $v \in \mathsf{Dead}(A_1)$, then u (denoted as y) is connected to s in step 1.3 of the construction of t'.

Case 3: Suppose $u \in \mathsf{Dead}(A_1)$. If $v \in A$, then u (denoted as x) is connected to s in step 1.2. of the construction of t'. If $v \in \mathsf{Active}(B_1)$, then s must occur at b, which implies that the η-operation above a in $\mathsf{tree}(t)$ which connects s to v also connects s to $\mathsf{mate}(v)$, a contradiction. If $v \in \mathsf{Dead}(B_1)$ then, since s must occur at b, s is not connected to v in $\mathsf{val}(t)$, a contradiction. The last case to consider is $v \in B_2$. Since $u \in \mathsf{Dead}(A_1)$, $u = x_{q-(i-1)}$ for some $i \in \{1, \ldots, q\}$, and $v \in NX_{q-(i-1)}$. Thus, u is connected to s in step 2.2 of the construction of f_i. □

Observation 14. *The expression t' defines G''.*

Proof of Observation 14. From the construction of t', it is clear that all the η-operations of t'' add edges which belong to G''. To complete the proof we show that all edges of G'' exist in $\mathsf{val}(t')$. Let $e = uv$ be an edge of G''. By definition of G'' one of the two endpoints of e (say u) is a regular vertex. If $u \in A$, then e is present in $\mathsf{val}(t'')$ by Observation 13. If $u \in B_1$, then e is present in $\mathsf{val}(t'')$ by Observations 8 and 9. If $u \in B_2$, then e is present in $\mathsf{val}(t'')$ by Observation 11. □

Observation 15. *The expression t' is a sequential k-expression.*

Proof of Observation 15. Since t has Property 6, a is the unique (> 1)-$\oplus$-operation in t, which implies that $t\langle c\rangle$ is sequential. The expression t' is constructed by adding to $t\langle c\rangle$ a sequence of operations which are either η, ρ, or 1-$\oplus$-operations. Thus, t' is a sequential expression. To complete the proof we show that at most k labels are used in t'. By Observation 12, the number of labels used in f_q is at most k. The highest number of labels used in the construction of t' from f_q is equal to $\mathsf{labels}(\mathsf{val}(f_q))$ plus one new label which is used to introduce special vertices (with label 2). By Observation 12 this number is at most $|A| + |NX_0| + 1$ which, by Observation 5, is less than k. □

Lemma 13 follows now from Observations 14 and 15. □

Combining the previous lemmas we now get a proof of Theorem 3.

Proof of Theorem 3. Let t be a k-expression defining G''.

By Lemma 7, there exists a $(k+4)$-expression t_1 defining G'' such that t_1 has Property 3.
By Lemma 8, there exists a $(k+6)$-expression t_2 defining G'' such that t_2 has Property 4.
By Lemma 9, there exists a $(k+6)$-expression t_3 defining G'' such that t_3 has Property 5.
By Lemma 11, there exists a $(k+6)$-expression t_4 defining G'' such that t_4 has Property 6.
By Lemma 12, there exists a $(k+6)$-expression t_5 defining G'' such that t_5 has Property 7.
By Lemma 13, there exists a sequential $(k + 6)$-expression t' which defines G''. This completes the proof of Theorem 3. □

4 Final remarks

We have shown that the clique-width of a graph cannot be computed in polynomial time unless $P = \mathrm{NP}$, and we are left with the question on the *parameterized complexity* of clique-width: what is the complexity of deciding whether the clique-width of a graph does not exceed a fixed parameter k? In particular, the following questions remain open:

Question 1. Is it possible to recognize graphs of clique-width at most 4 in polynomial time?

Question 2. If k is a fixed constant, is it possible to recognize graphs of clique-width at most k in polynomial time?

Question 3. Is the recognition of graphs of clique-width at most k *fixed-parameter tractable*? I.e., is it possible to recognize graphs of clique-width at most k in time $O(f(k)n^c)$, where n denotes the size of the given graph, f is a computable function, and c is a constant which does not depend on k.

Obviously, a positive answer to Question 1 is a necessary pre-condition for a positive answer to Question 2, and a positive answer to Question 2 is a necessary pre-condition for a positive answer to Question 3.

Acknowledgement

We thank Derek Corneil for introducing us to the problem and for all the inspiring discussions.

References

[1] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic Discrete Methods*, 8(2):277–284, 1987.

[2] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *J. Algorithms*, 18(2):238–255, 1995.

[3] D. G. Corneil, M. Habib, J.-M. Lanlignel, B. A. Reed, and U. Rotics. Polynomial time recognition of clique-width ≤ 3 graphs (extended abstract). In G. H. Gonnet, D. Panario, and A. Viola, editors, *Theoretical Informatics, 4th Latin American Symposium (LATIN 2000)*, volume 1776 of *Lecture Notes in Computer Science*, pages 126–134, 2000.

[4] B. Courcelle, J. Engelfriet, and G. Rozenberg. Context-free handle-rewriting hypergraph grammars. In H. Ehrig, H.-J. Kreowski, and G. Rozenberg, editors, *Graph-Grammars and their Application to Computer Science, 4th International Workshop, Bremen, Germany, March 5–9, 1990, Proceedings*, volume 532 of *Lecture Notes in Computer Science*, pages 253–268, 1991.

[5] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33:125–150, 2000.

[6] M. R. Fellows, F. A. Rosamond, U. Rotics, and S. Szeider. Proving NP-hardness for clique-width I: non-approximability of sequential clique-width. Electronic Colloquium on Computational Complexity, Technical Report TR05-081, **Revision 01**, 2005.

[7] F. Gurski and E. Wanke. Minimizing NLC-width is NP-complete. Extended abstract accepted for WG 2005.

[8] Ö. Johansson. Clique-decomposition, NLC-decomposition, and modular decomposition—relationships and results for random graphs. In *Proceedings of the Twenty-ninth Southeastern International Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 1998)*, volume 132 of *Congr. Numer.*, pages 39–60, 1998.

[9] M. Karpinski and J. Wirtgen. On approximation hardness of the bandwidth problem. Technical Report TR97-041, ECCC, Electronic Colloquium on Computational Complexity, 1997.

[10] S. Oum and P. Seymour. Approximating clique-width and branch-width. Submitted, Oct. 2004.

[11] E. Wanke. k-NLC graphs and polynomial algorithms. *Discr. Appl. Math.*, 54(2-3):251–266, 1994. Efficient algorithms and partial k-trees.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'