Order Reconfiguration Under Width Constraints

Emmanuel Arrighi
University of Bergen, Norway

Henning Fernau
University of Trier, Germany

Mateus de Oliveira Oliveira
University of Bergen, Norway

Petra Wolf
University of Trier, Germany

Abstract

In this work, we consider the following order reconfiguration problem: Given a graph G together with linear orders ω and ω' of the vertices of G, can one transform ω into ω' by a sequence of swaps of adjacent elements in such a way that at each time step the resulting linear order has cutwidth (pathwidth) at most k? We show that this problem always has an affirmative answer when the input linear orders ω and ω' have cutwidth (pathwidth) at most $k/2$. Using this result, we establish a connection between two apparently unrelated problems: the reachability problem for two-letter string rewriting systems and the graph isomorphism problem for graphs of bounded cutwidth. This opens an avenue for the study of the famous graph isomorphism problem using techniques from term rewriting theory.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability; Mathematics of computing → Combinatorial optimization; Theory of computation → Equational logic and rewriting

Keywords and phrases Parameterized Complexity, Order Reconfiguration, String Rewriting Systems

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.8

Funding *Emmanuel Arrighi*: Research Council of Norway (274526), IS-DAAD (309319).
Henning Fernau: DAAD PPP (57525246).
Mateus de Oliveira Oliveira: Research Council of Norway (288761), IS-DAAD (309319).
Petra Wolf: DFG project FE 560/9-1, DAAD PPP (57525246).

1 Introduction

In the field of reconfiguration, one is interested in studying relationships among solutions of a problem instance [17, 24, 27]. Here, by reconfiguration of one solution into another, we mean a sequence of steps where each step transforms a feasible solution into another. Three fundamental questions in this context are: (1) Is it the case that any two solutions can be reconfigured into each other? (2) Can any two solutions be reconfigured into each other in a polynomial number of steps? (3) Given two feasible solutions X and Y, can one find in polynomial time a reconfiguration sequence from X to Y?

In this work, we study the reconfiguration problem in the context of linear arrangements of the vertices of a given graph G. The space of feasible solutions is the set of all linear orders of cutwidth (pathwidth) at most k for some given $k \in \mathbb{N}$. We say that a linear order ω can be reconfigured into a linear order ω' in width k if there is a sequence $\omega_1, \ldots, \omega_m$ of linear orders of width at most k such that $\omega_1 = \omega$, $\omega_m = \omega'$ and for each $i \in \{2, \ldots, m\}$, ω_i is obtained from ω_{i-1} by swapping two adjacent vertices. Our main result (Theorem 3) states that if ω and ω' are linear orders of cutwidth at most k, then ω can be reconfigured into ω' in width at most $2k$. Additionally, reconfiguration in width at most $2k$ can be done using at most $\mathcal{O}(n^2)$ swaps. Finally, a reconfiguration sequence can be found in polynomial time.

46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021).
Editors: Filippo Bonchi and Simon J. Puglisi; Article No. 8; pp. 8:1–8:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Our results on reconfiguration of linear arrangements can be used to establish an interesting connection between two apparently unrelated computational problems: reachability for two-letter string rewriting and graph isomorphism.

A two-letter rewriting rule over a given alphabet Σ is a rewriting rule of the form $ab \to cd$ for letters $a, b, c, d \in \Sigma$. A two-letter string rewriting system is a collection R of two-letter string rewriting rules. The reachability problem for such a rewriting system R is the problem of determining whether a given string $x \in \Sigma^n$ can be transformed into a given string $y \in \Sigma^n$ by the application of a sequence of two-letter rewriting rules. On the other hand, in the graph isomorphism problem, we are given graphs G and G' and the goal is to determine whether there exists a bijection φ from the vertex set of G to the vertex set of G' in such a way that an edge $\{u, v\}$ belongs to G if and only if the edge $\{\varphi(u), \varphi(v)\}$ belongs to G'.

In order to describe more precisely the connections between two-letter term rewriting and graph isomorphism, we briefly discuss the notion of slices and unit decompositions. A *slice* is a graph $\mathbf{S}$ where the vertices are partitioned into a center C and special in-frontier I and out-frontier O that can be used for composition. A slice $\mathbf{S}_1$ can be glued to a slice $\mathbf{S}_2$ if the out-frontier of $\mathbf{S}_1$ can be coherently matched with the in-frontier of $\mathbf{S}_2$. In this case, the gluing gives rise to a bigger slice $\mathbf{S}_1 \circ \mathbf{S}_2$ which is obtained by matching the out-frontier of $\mathbf{S}_1$ with the in-frontier of $\mathbf{S}_2$. A *unit slice* is a slice with a unique vertex in the center. Any slice $\mathbf{S}$ can be decomposed into a sequence of unit slices. More specifically, a *unit decomposition* is a sequence $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2 \ldots \mathbf{S}_n$ of unit slices with the property that for each $i \in [n-1]$, $\mathbf{S}_i$ can be glued to the slice $\mathbf{S}_{i+1}$. The result of gluing the unit slices in $\mathbf{U}$ is a slice $\mathring{\mathbf{U}}$ with n center vertices. Conversely, any slice $\mathbf{S}$ with n center vertices can be written as a unit decomposition $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2 \ldots \mathbf{S}_n$ with the property that $\mathring{\mathbf{U}}$ is isomorphic to $\mathbf{S}$.

An important remark connecting unit decompositions and the notion of cutwidth is that if a slice $\mathbf{S}$ has cutwidth k, then $\mathbf{S}$ can be decomposed into a unit decomposition $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2 \ldots \mathbf{S}_n$ where for each $i \in [n]$, $\mathbf{S}_i$ has at most k vertices in each frontier. Therefore, if we let $\mathbf{\Sigma}(k)$ denote the set of all unit slices with frontiers of size at most k, then any graph G with n vertices of cutwidth at most k can be written as a word (unit decomposition) of length n over the alphabet $\mathbf{\Sigma}(k)$. In this work, for each $k \in \mathbb{N}$, we introduce a suitable two-letter string rewriting system $R(k)$ over the alphabet $\mathbf{\Sigma}(k)$ with the following property: if $\mathbf{U}$ and $\mathbf{U}'$ are two unit decompositions over $\mathbf{\Sigma}(k)$ and if $\mathbf{U}$ can be transformed into $\mathbf{U}'$ using the rewriting rules in $R(k)$, then the graphs $\mathring{\mathbf{U}}$ and $\mathring{\mathbf{U}}'$ are isomorphic. Our second main result is a partial converse for this property. More precisely, we show that given two unit decompositions $\mathbf{U}$ and $\mathbf{U}'$ over $\mathbf{\Sigma}(k)$, if the graphs $\mathring{\mathbf{U}}$ and $\mathring{\mathbf{U}}'$ are isomorphic, then each of these unit decompositions can be transformed into one another by the application of rewriting rules from the string rewriting system $R(2k)$ (Theorem 11).

The proof of Theorem 11 is heavily based on Theorem 3. An important feature of this proof is that, given an isomorphism from $\mathring{\mathbf{U}}$ to $\mathring{\mathbf{U}}'$, one can construct a sequence of rewriting steps transforming $\mathbf{U}$ into $\mathbf{U}'$. Conversely, given any such a sequence, we are able to construct an isomorphism from $\mathring{\mathbf{U}}$ to $\mathring{\mathbf{U}}'$. This result, together with the fact that unit decompositions of minimum cutwidth can be approximated in FPT time, implies that the graph isomorphism problem for graphs of cutwidth at most k is FPT-equivalent to the reachability problem for $R(2k)$ (Theorem 13).

Related Work. The reachability problem for a given string rewriting system R consists in determining whether a given string x can be transformed into a given string y by the application of rewriting rules from R. Reachability is a central problem in the field of string rewriting [6] and can also be studied under the light of term rewriting theory [19, 5, 1, 6].

The complexity of the reachability problem is highly dependent on the rewriting system R. For general rewriting systems, the problem becomes undecidable [6]. In the case of two-letter rewriting, reachability can be solved in PSPACE since in this case, strings never grow in size. It is also not difficult to design two-letter rewriting systems for which the reachability problem is PSPACE-complete. Nevertheless, our results imply that for each $k \in \mathbb{N}$, the $R(2 \cdot k)$-reachability problem for unit decompositions of length n and width at most k is reducible to the graph isomorphism problem. Therefore, it can be solved in time $n^{\text{polylog}(n)}$, independently of k, using Babai's quasi-polynomial time algorithm for graph isomorphism [2]. An interesting question we leave unsolved is the complexity of $R(\alpha \cdot k)$-reachability for unit decompositions of width at most k when α is a rational number with $1 \leq \alpha < 2$. In particular, we do not know if there is such an α for which the reachability problem becomes PSPACE-hard.

In the field of parameterized complexity theory [7, 6], a computational problem is said to be *fixed-parameter tractable (FPT)* with respect to a parameter k if it can be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$ on inputs of size n. Here $f : \mathbb{N} \to \mathbb{N}$ is a computable function depending only on the parameter k, but not on the size n of the input. The Graph Isomorphism problem (GI for short) has been shown to be solvable in time $f(k) \cdot n^{\mathcal{O}(1)}$ (that is, FPT time) whenever the parameter k stands for eigenvalue multiplicity [3], treewidth [21], feedback vertex-set number [20], or size of the largest color class [9] of the involved graphs. On the other hand, GI can be solved in time $f_1(k) \cdot n^{f_2(k)}$ (that is, in XP time), whenever the parameter k stands for genus [23], rankwidth [15], maximum degree [22], size of an excluded topological subgraph [12], or size of an excluded minor [11]. We note that, in particular, Babai's algorithm and techniques have been recently used to improve the fastest FPT algorithm for graphs of treewidth at most k from $2^{\mathcal{O}(k^5 \cdot \log k)} \cdot n^{\mathcal{O}(1)}$ [21] to $2^{O(k \cdot \text{polylog}(k))} \cdot n^{\mathcal{O}(1)}$ [14], and for graphs of maximum degree d, the fastest XP-algorithm has been improved from $n^{\mathcal{O}(d/\log d)}$ [4] to $n^{\text{polylog}(d)}$ [13]. In particular, it is worth noting that graphs of cutwidth k have maximum degree at most k and treewidth $O(k)$. Therefore, isomorphism of graphs of cutwidth k can be solved in time $2^{O(k \cdot \text{polylog}(k))} \cdot n^{\mathcal{O}(1)}$ [14]. This implies that $R(2 \cdot k)$-reachability can be solved in $2^{O(k \cdot \text{polylog}(k))} \cdot n^{\mathcal{O}(1)}$ time when restricted to unit decompositions of width at most k. Showing that isomorphism for graphs of cutwidth k can be solved in time $2^{O(k)} \cdot n^{\mathcal{O}(1)}$ is still an open problem.

Another width parameter for linear orders that has been studied in the context of graph theory is the vertex separation number of a graph [8]. This parameter may be seen as a order theoretic interpretation of the notion of pathwidth. The techniques used to prove Theorem 3 can be generalized to prove that reconfiguration of linear orders of vertex separation number k can always be achieved in width at most $2 \cdot k$ (Theorem 16). While we do not provide a string-rewriting interpretation of this result, we do state it formally in Section 5 since this result may be of independent interest in the field of reconfiguration.

2 Preliminaries

Basics. We let $\mathbb{N}$ denote the set of natural numbers, including 0, and $\mathbb{N}_+$ denote the set of positive natural numbers. For each $n \in \mathbb{N}_+$, we let $[n] = \{1, \ldots, n\}$. As a degenerate case, we let $[0] = \emptyset$. Given a finite set S, we let $\mathcal{P}(S)$ be the set of all subsets of S. For each $k \in \mathbb{N}$, we let $\mathcal{P}(S, k)$ and $\mathcal{P}(S, \leq k)$ be the sets of subsets of S of size exactly k and at most k, respectively.

Graphs. In this work, graphs are simple and undirected. Given a graph G we let $V(G)$ denote the vertex set of G and $E(G)$ denote the edge set of G. Given a subset $S \subseteq V(G)$, we let $G[S]$ be the subgraph of G induced by S. More precisely, $V(G[S]) = S$ and $E(G[S]) = E(G) \cap \mathcal{P}(S, 2)$. An *isomorphism* from a graph G to a graph G' is a bijection $\varphi : V(G) \to V(G')$ such that for each $v, u \in V(G)$, $\{v, u\} \in E(G)$ if and only if $\{\varphi(v), \varphi(u)\} \in E(G')$. If such an isomorphism exists, we say that G is *isomorphic* to G'.

Order. Let V be a set with $|V| = n$. A linear order on V is a bijection $\omega : [n] \to V$. Intuitively, for each $j \in [n]$ and $v \in V$, $\omega(j) = v$ indicates that v is the j-th element of ω. If $S \subseteq [n]$, then we let $\omega(S) = \{\omega(j) \ : \ j \in S\}$ be the image of S under ω. Given linear orders $\omega, \omega' : [n] \to V$ of V and a number $i \in [n-1]$, we write $\omega \xrightarrow{i} \omega'$ to indicate that ω' is obtained from ω by swapping the order of the vertices at positions i and $i+1$. More precisely, $\omega'(j) = \omega(j)$ for every $j \in [n] \setminus \{i, i+1\}$, $\omega'(i) = \omega(i+1)$, and $\omega'(i+1) = \omega(i)$.

Let $\omega : [n] \to V$ be a linear order on a set V. Let $S \subseteq V$. We let $\omega^S : [|S|] \to S$ be the linear order induced by ω on S. More precisely, if we write the elements of S in increasing order according to ω, then for each $i \in [|S|]$, $\omega^S(i)$ is the i-th element in this sequence.

Order Reconfiguration. We say that ω can be reconfigured into ω' in one swap, and denote this fact by $\omega \to \omega'$, if there exists some $i \in [n]$ such that $\omega \xrightarrow{i} \omega'$. We say that ω can be reconfigured into ω' in at most r swaps, and denote this fact by $\omega \to_r \omega'$, if there are numbers $r' \in [r]$, $i_1, \dots i_{r'} \in [n]$, and linear orders $\omega_0, \dots, \omega_{r'}$ such that

$$\omega = \omega_0 \xrightarrow{i_1} \omega_1 \xrightarrow{i_2} \dots \xrightarrow{i_{r'}} \omega_{r'} = \omega'.$$

We call this sequence a *reconfiguration sequence* from ω to ω'. The mere existence of a (possibly empty) reconfiguration sequence from ω to ω' is also written as $\omega \to^* \omega'$.

Composition of Linear Orders. Let $i \in \{0, \dots, n\}$, and $\omega, \omega' : [n] \to V$. We let $\omega \oplus_i \omega' : [n] \to V$ be the linear order that orders the vertices in the subset $\omega([i]) \subseteq V$ according to ω followed by the vertices in the subset $V \setminus \omega([i])$, ordered according to ω'. More precisely, $\omega \oplus_i \omega'$ is defined as follows for each $j \in [n]$.

$$\omega \oplus_i \omega'(j) = \begin{cases} \omega(j) & \text{if } j \leq i, \\ \omega'^{V \setminus \omega([i])}(j - i) & \text{if } j > i. \end{cases} \tag{1}$$

We note that in particular, $\omega \oplus_0 \omega' = \omega'$ and $\omega \oplus_n \omega' = \omega$.

String Rewriting. A two-letter string rewriting systems is a pair (Σ, R) where Σ is a finite, non-empty set of symbols (an alphabet), and $R \subseteq \Sigma^2 \times \Sigma^2$ is a set of rewriting rules of the form $ab \to cd$. Let x and y be strings in Σ^n and $i \in [n-1]$. We say that x can be transformed into y by applying a rewriting rule $ab \to cd$ at position i if $x_i x_{i+1} = ab$, $y_i y_{i+1} = cd$ and $x_j = y_j$ for $j \notin \{i, i+1\}$. We write $x \xrightarrow{i} y$ to denote that x can be transformed into y by the application of some rewriting rule at position i. We write $x \to y$ to denote that x can be transformed into y by the application of some rewriting rule at some position $i \in [n-1]$. We say that y is reachable from x if there is a sequence of strings $x = x_0, x_1, \dots, x_m = y$ such that $x_{i-1} \to x_i$ for each $i \in [m]$. We write $x \to_* y$ to denote that y is reachable from x. We say that x and y are R-equivalent if $x \to_* y$ and $y \to_* x$.

3 Linear Order Reconfiguration

Let G be an n-vertex graph. Given sets $S, S' \subseteq V(G)$, we let $E(G,S,S') = \{\{u,v\} \in E(G) : u \in S,\ v \in S'\}$ be the set of edges with one endpoint in S and the other endpoint in S'. As a special case, we define $E(G,S) = E(G,S,V(G) \setminus S)$. We will often make use of the following monotonicity property without explicit mentioning: If $T \subseteq S$ and $T' \subseteq S'$, then $|E(G,T,T')| \leq |E(G,S,S')|$.

▶ **Definition 1** (Cutwidth). *Let G be an n-vertex undirected graph. Let $\omega : [n] \to V(G)$ be a linear order on the vertices of G. For each $p \in [n]$, we let $\mathrm{cw}(G,\omega,p) = |E(G,\omega([p-1]))|$ be the number of edges that have one endpoint in the first $p-1$ vertices of the linear order ω and the other endpoint in the remaining vertices. The* cutwidth *of the linear order ω is defined as $\mathrm{cw}(G,\omega) = \max_{p\in[n]} \mathrm{cw}(G,\omega,p)$. The* cutwidth *of the graph G is defined as $\mathrm{cw}(G) = \min_\omega \mathrm{cw}(G,\omega)$, where ω ranges over all linear orders on the vertex set $V(G)$.*

For each $k \in \mathbb{N}$, and each n-vertex graph G, we let $\mathrm{CW}(G,k) = \{\omega : [n] \to V(G) : \mathrm{cw}(G,\omega) \leq k\}$ be the set of linear orders of $V(G)$ of cutwidth at most k. We say that ω can be *reconfigured* into ω' in cutwidth at most k if there is a *reconfiguration sequence*

$$\omega = \omega_0 \xrightarrow{i_1} \omega_1 \xrightarrow{i_2} \cdots \xrightarrow{i_r} \omega_r = \omega'$$

such that for each $j \in \{0,\ldots,r\}$, $\omega_j \in \mathrm{CW}(G,k)$.

▶ **Problem 2** (Bounded Cutwidth Order Reconfiguration). *Let G be an n-vertex graph, $\omega,\omega' : [n] \to V(G)$ be linear orders on the vertex set of G, and $k \in \mathbb{N}$. Is it true that ω can be reconfigured into ω' in cutwidth at most k?*

It should be clear that if k is smaller than the cutwidth of the graph G, then the answer for Problem 2 is trivially NO since in this case neither ω nor ω' are in $\mathrm{CW}(G,k)$. On the other hand, we will show in Theorem 3 below that the answer is always YES if k is at least twice the cutwidth of the thickest input linear order.

▶ **Theorem 3.** *Let G be an n-vertex graph and $\omega,\omega' : [n] \to V(G)$ be linear orders of $V(G)$ of cutwidth at most k. Then, ω can be reconfigured into ω' in cutwidth at most $\mathrm{cw}(G,\omega) + \mathrm{cw}(G,\omega') \leq 2k$.*

To prove this theorem, we need the following three lemmas.

▶ **Lemma 4.** *Let G be an n-vertex graph, $S \subseteq V(G)$ and $\omega : [n] \to V(G)$ be a linear order on $V(G)$. Then, ω^S is a linear order on $V(G[S])$. Additionally, $\mathrm{cw}(G[S],\omega^S) \leq \mathrm{cw}(G,\omega)$.*

Proof. As $S = V(G[S])$, ω^S is a linear order on $V(G[S])$. Let $p \in [|S|]$ and let $p' \in [n]$ be the unique number such that $\omega^S(p) = \omega(p')$. Then,

$$\begin{aligned}\mathrm{cw}(G[S],\omega^S,p) &= |E(G[S],\omega^S([p-1]))|\\ &= |E(G[S],\omega^S([p-1]),\{\omega^S(r) : r \geq p\})|\\ &= |E(G,\omega^S([p-1]),\{\omega^S(r) : r \geq p\})|\\ &\leq |E(G,\omega([p'-1]))|\\ &= \mathrm{cw}(G,\omega,p')\\ &\leq \mathrm{cw}(G,\omega)\,,\end{aligned}$$

as $\omega^S([p-1]) \subseteq \omega([p'-1])$ and $\{\omega^S(r) : r \geq p\} \subseteq \{\omega(r') : r' \geq p'\} = V(G)\setminus\omega([p'-1])$. ◀

▶ **Lemma 5.** *Let G be an n-vertex graph and $\omega, \omega' : [n] \to V(G)$ be linear orders of $V(G)$ with cutwidth of at most k. Then, for each $i \in [n]$, $\omega \oplus_i \omega'$ has cutwidth at most* $\mathrm{cw}(G, \omega) + \mathrm{cw}(G, \omega') \leq 2k$.

Proof. Let $i, p \in [n]$. There are two cases to be analyzed. By definition, we have that

$$\mathrm{cw}(G, \omega \oplus_i \omega', p) = |E(G, \omega \oplus_i \omega'([p-1]))| = |E(G, \omega \oplus_i \omega'([p-1]), V(G) \setminus \omega \oplus_i \omega'([p-1]))|\,.$$

First, if $p \leq i$, then we have

$$\mathrm{cw}(G, \omega \oplus_i \omega', p) = |E(G, \omega([p-1]), V(G) \setminus \omega([p-1]))| = \mathrm{cw}(G, \omega, p) \leq \mathrm{cw}(G, \omega)\,.$$

Secondly, if $p > i$, then we have

$$\begin{aligned}
\mathrm{cw}(G, \omega \oplus_i \omega', p) &= |E(G, \omega \oplus_i \omega'([p-1]), V(G) \setminus \omega \oplus_i \omega'([p-1]))| \\
&\overset{(a)}{=} |E(G, (\omega \oplus_i \omega'([p-1])) \cap \omega([i]), V(G) \setminus \omega \oplus_i \omega'([p-1]))| \\
&\qquad + |E(G, (\omega \oplus_i \omega'([p-1])) \setminus \omega([i]), V(G) \setminus \omega \oplus_i \omega'([p-1]))| \\
&\overset{(b)}{\leq} \mathrm{cw}(G, \omega, i+1) + \mathrm{cw}(G[V(G) \setminus \omega([i])], \omega'^{V(G)\setminus\omega([i])}, p-i) \\
&\leq \mathrm{cw}(G, \omega) + \mathrm{cw}(G, \omega').
\end{aligned}$$

For Equality (a), observe that $\{(\omega \oplus_i \omega'([p-1])) \cap \omega([i]), (\omega \oplus_i \omega'([p-1])) \setminus \omega([i])\}$ is a partition of $\omega \oplus_i \omega'([p-1])$. To understand Inequality (b), we need two arguments. As $\omega([i]) \subseteq \omega \oplus_i \omega'([p-1])$,

$$E(G, (\omega \oplus_i \omega'([p-1])) \cap \omega([i]), V(G) \setminus (\omega \oplus_i \omega'([p-1]))) \subseteq E(G, \omega([i]), V(G) \setminus \omega([i]))\,,$$

which shows that the cardinality of the first set is upper-bounded by $\mathrm{cw}(G, \omega, i+1)$. As the edges in $E(G, (\omega \oplus_i \omega'([p-1])) \setminus \omega([i]), V(G) \setminus (\omega \oplus_i \omega'([p-1]))$ only connect vertices with positions beyond i within $\omega \oplus_i \omega'$, after an index shift, we see that only the linear order ω' really matters, which explains the inequality

$$|E(G, (\omega \oplus_i \omega'([p-1])) \setminus \omega([i]), V(G) \setminus (\omega \oplus_i \omega'([p-1]))| \leq \mathrm{cw}(G[V(G) \setminus \omega([i])], \omega'^{V(G)\setminus\omega([i])}, p-i)\,.$$

For the last inequality, apply Lemma 4 to derive $\mathrm{cw}(G[V(G) \setminus \omega([i])], \omega'^{V(G)\setminus\omega([i])}) \leq \mathrm{cw}(G, \omega')$. As p is arbitrary, $\mathrm{cw}(G, \omega \oplus_i \omega') \leq \mathrm{cw}(G, \omega) + \mathrm{cw}(G, \omega') \leq 2k$ follows for each $i \in [n]$. ◀

▶ **Lemma 6.** *Let G be an n-vertex graph, $\omega, \omega' : [n] \to V(G)$ be linear orders on $V(G)$ of cutwidth at most k and $i \in \{0, \ldots, n-1\}$ be an integer. Then, $\omega \oplus_i \omega'$ can be reconfigured into $\omega \oplus_{i+1} \omega'$ in cutwidth at most* $\mathrm{cw}(G, \omega) + \mathrm{cw}(G, \omega') \leq 2k$.

Proof. By Lemma 5, $\omega \oplus_i \omega'$ and $\omega \oplus_{i+1} \omega'$ have cutwidth at most $\mathrm{cw}(G, \omega) + \mathrm{cw}(G, \omega') \leq 2k$. Let $j \in [n]$ such that $\omega \oplus_i \omega'(j) = \omega(i+1)$, i.e., j is the position of $\omega(i+1)$ in $\omega \oplus_i \omega'$. As for each $p \in [i]$, $\omega \oplus_i \omega'(p) = \omega \oplus_{i+1} \omega'(p) = \omega(p)$, we have $j > i$. Let us consider the following sequence of swaps:

$$\omega \oplus_i \omega' = \omega_0 \xrightarrow{j-1} \omega_1 \xrightarrow{j-2} \cdots \xrightarrow{i+1} \omega_{j-i-1} = \omega \oplus_{i+1} \omega'.$$

If $j = i+1$, this sequence is empty and $\omega \oplus_i \omega' = \omega \oplus_{i+1} \omega'$. At each step of this sequence, we swap $\omega(i+1)$ with its left neighbor. This brings $\omega(i+1)$ from position j to position $i+1$. By doing this, we transform $\omega \oplus_i \omega'$ into $\omega \oplus_{i+1} \omega'$.

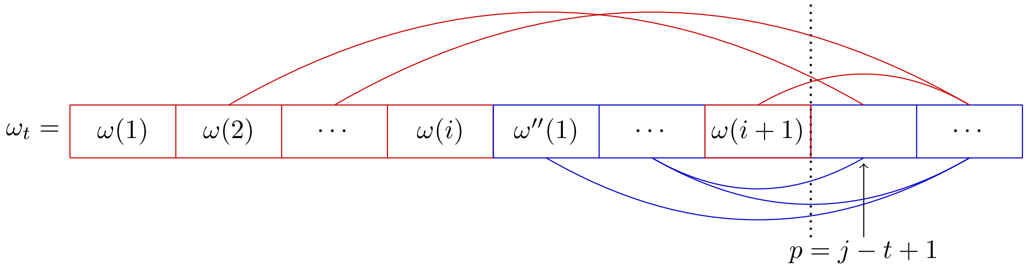

Figure 1 Illustration of the key part in Lemma 6. In this figure $\omega'' = \omega'^{V \setminus \omega([i+1])}$. The red part of the linear order follows the linear order ω for the first $i+1$ elements, and the blue part of the linear order follows ω' for the remaining elements. Then, the set of edges crossing the cut at position $p = j - t + 1$ can be split in two, the set of edges that start from the red part and the set of edge that start from the green part. The number of red edges is bounded by the cutwidth of ω and the number of green edges is bounded by the cutwidth of ω'.

Inductively, we show that each element ω_t in the sequence has cutwidth upper-bounded by $\mathrm{cw}(G, \omega) + \mathrm{cw}(G, \omega') \leq 2k$. By Lemma 5, $\mathrm{cw}(G, \omega_0) = \mathrm{cw}(G, \omega \oplus_i \omega') \leq \mathrm{cw}(G, \omega) + \mathrm{cw}(G, \omega') \leq 2k$, which proves the induction basis. Let $t \in [j-i-1]$ and $p \in [n]$ be two integers. As induction hypothesis, we have $\mathrm{cw}(G, \omega_{t-1}) \leq \mathrm{cw}(G, \omega) + \mathrm{cw}(G, \omega') \leq 2k$. If $p \leq j - t$ or $p > j - t + 1$, then we have $\omega_{t-1}([p-1]) = \omega_t([p-1])$, so $\mathrm{cw}(G, \omega_t, p) = \mathrm{cw}(G, \omega_{t-1}, p) \leq \mathrm{cw}(G, \omega) + \mathrm{cw}(G, \omega') \leq 2k$ by induction hypothesis. Otherwise, $p = j - t + 1 \in \{i, \ldots, j\}$ (Figure 1) and we have

$$
\begin{aligned}
\mathrm{cw}(G, \omega_t, p) &= |E(G, \omega_t([p-1]))| \\
&= |E(G, \omega_t([p-1]), \{\omega_t(r) \;:\; r \geq p\})| \\
&= |E(G, \omega_t([i] \cup \{p-1\}), \{\omega_t(r) \;:\; r \geq p\})| \\
&\quad + |E(G, \{\omega_t(l) \;:\; i < l < p-1\}, \{\omega_t(r) \;:\; r \geq p\})|.
\end{aligned}
$$

By definition of ω_t and p, we have $\omega_t(p-1) = \omega_t(j-t) = \omega(i+1)$. Therefore, we have $|E(G, \omega_t([i] \cup \{p-1\}), \{\omega_t(r) \;:\; r \geq p\})| = |E(G, \omega([i+1]), \{\omega_t(r) \;:\; r \geq j-t+1\})|$. As we are swapping $\omega(i+1)$ leftwards, $\{\omega_t(r) \;:\; r \geq j-t+1\} \subseteq \{\omega(r) \;:\; r \geq i+2\} = V(G) \setminus \omega([i+1])$. Again by definition of ω_t and p, the elements in $\{\omega_t(l) \;:\; i < l < p-1\}$ are ordered according to ω', which is also true for $\{\omega_t(r) \;:\; r \geq p\}$. More formally, $\{\omega_t(l) \;:\; i < l < p-1\} = \{\omega \oplus_i \omega'(l) \;:\; i+1 \leq l \leq p-2\} = \{\omega'^{V(G) \setminus \omega([i+1])}(l') \;:\; l' \leq p-2-i\}$ and $\{\omega_t(r) \;:\; r \geq p\} = \{\omega \oplus_i \omega'(r) \;:\; r \geq p\} = \{\omega'^{V(G) \setminus \omega([i+1])}(r') \;:\; r' \geq p-i-1\}$. Therefore,

$$
\begin{aligned}
\mathrm{cw}(G, \omega_t, p) &\leq \mathrm{cw}(G, \omega, i+2) + \mathrm{cw}(G[V(G) \setminus \omega([i+1])], \omega'^{V(G) \setminus \omega([i+1])}, p-i-1) \\
&\leq \mathrm{cw}(G, \omega) + \mathrm{cw}(G[V(G) \setminus \omega([i+1])], \omega'^{V(G) \setminus \omega([i+1])}) \\
&\leq \mathrm{cw}(G, \omega) + \mathrm{cw}(G, \omega') \\
&\leq 2k.
\end{aligned}
$$

To achieve the penultimate inequality, we again apply Lemma 4. ◀

Proof of Theorem 3. Consider the following sequence: $\omega = \omega' \oplus_0 \omega \to^* \omega' \oplus_1 \omega \to^* \cdots \to^* \omega' \oplus_n \omega = \omega'$. By Lemma 6, one can realize each step in cutwidth at most $\mathrm{cw}(G, \omega) + \mathrm{cw}(G, \omega') \leq 2k$, which then also upper-bounds the whole reconfiguration sequence. ◀

4 Slice Rewriting System

Slices. Let $\mathcal{I} = \{[a] \ : \ a \in \mathbb{N}\}$ denote the set of intervals of the form $[a] = \{1, \ldots, a\}$ for $a \in \mathbb{N}$ (recall that $[0] = \emptyset$). We let $\mathcal{I}_0 = \{\{0\} \times [a] : [a] \in \mathcal{I}\}$, and $\mathcal{I}_1 = \{\{1\} \times [a] : [a] \in \mathcal{I}\}$. A *slice* $\mathbf{S} = (I, C, O, E)$ is a (multi-)graph where the vertex set $V = I \,\dot\cup\, C \,\dot\cup\, O$ is partitioned into an *in-frontier* $I \in \mathcal{I}_0$, a *center* $C \in \mathcal{I}$ and an *out-frontier* $O \in \mathcal{I}_1$, and E is a multiset of unordered pairs from $I \cup C \cup O$ in such a way that vertices of $I \cup O$ have degree exactly 1, there is no edge between any two vertices in I, and no edge between any two edges in O. We depict slices as in Figure 2. We define slices using multigraphs, as the gluing operation, defined below, can take slices which are simple graphs, and create a slice which is a multigraph (see Figure 5). Given a slice $\mathbf{S}$, we define $I(\mathbf{S})$ as the in-frontier of $\mathbf{S}$, $O(\mathbf{S})$ as the out-frontier of $\mathbf{S}$, and $C(\mathbf{S})$ as the center vertices of $\mathbf{S}$. The *width* of a slice $\mathbf{S}$ is defined as $\mathbf{w}(S) = \max(|I(\mathbf{S})|, |O(\mathbf{S})|)$.

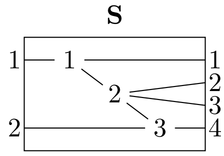

Figure 2 Slices are drawn as tiles. This figure depicts the slice $\mathbf{S} = (I, C, O, E)$ where $I = \{(0,1), (0,2)\}$, $C = \{1,2,3\}$, $O = \{(1,1), (1,2), (1,3), (1,4)\}$ and $E = \{\{(0,1), 1\}, \{(0,2), 3\}, \{1,2\}, \{2,3\}, \{1,(1,1)\}, \{2,(1,2)\}, \{2,(1,3)\}, \{3,(1,4)\}\}$. We omit the first element of the pair for frontier vertices and use the following convention. The in-frontier vertices are on the left of the tile and the out-frontier vertices are on the right of the tile. If the frontier vertices are not explicitly mentioned in the drawing, we assume that frontier vertices are ordered from top to bottom as in this drawing.

Gluing Slices. A slice $\mathbf{S}_1 = (I_1, C_1, O_1, E_1)$ can be glued to $\mathbf{S}_2 = (I_2, C_2, O_2, E_2)$ if for some interval $[a] \in \mathcal{I}$, $O_1 = \{1\} \times [a]$ and $I_2 = \{0\} \times [a]$. In this case, the gluing gives rise to the slice $\mathbf{S}_1 \circ \mathbf{S}_2 = (I_1, C_1 \cup (|C_1| \oplus C_2), O_2, E)$ where $|C_1| \oplus C_2 = \{|C_1|+1, |C_1|+2, \ldots, |C_1|+|C_2|\}$,

$$
\begin{aligned}
E =& \{\{x, y\} \in E_1 \ : \ x, y \in I_1 \cup C_1\} \\
& \cup \{\{x, y\} \in E_2 \ : \ x, y \in O_2\} \\
& \cup \{\{x, y + |C_1|\} \ : \ \{x, y\} \in E_2 \wedge x \in O_2 \wedge y \in C_2\} \\
& \cup \{\{x + |C_1|, y + |C_1|\} \ : \ \{x, y\} \in E_2 \wedge x, y \in C_2\} \\
& \cup \{\{x, y\} \ : \ \exists i, \{x, (1, i)\} \in E_1 \wedge y \in O_2 \wedge \{(0, i), y\} \in E_2\} \\
& \cup \{\{x, y\} \ : \ \exists i, \{x, (1, i)\} \in E_1 \wedge y \in |C_1| \oplus C_2 \wedge \{(0, i), y - |C_1|\} \in E_2\}.
\end{aligned}
$$

Note that the gluing operation is associative. Therefore, we will not write parentheses for the gluing of more than two slices. Figure 3 illustrates the gluing of two slices.

Unit Slices and Unit Decompositions. We say that a slice is a *unit slice* if it has a unique vertex in its center. A *unit decomposition* is a sequence $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2 \ldots \mathbf{S}_n$, where $\mathbf{S}_i$ are unit slices and $\mathbf{S}_i \circ \mathbf{S}_{i+1}$ is well defined for each $i \in [n-1]$. The *slice associated to a unit decomposition* $\mathbf{U}$ is defined as $\mathring{\mathbf{U}} = \mathbf{S}_1 \circ \mathbf{S}_2 \circ \ldots \circ \mathbf{S}_n$ (Figure 4). Note that if the in-frontier of $\mathbf{S}_1$ and the out-frontier of $\mathbf{S}_n$ are empty, then $\mathring{\mathbf{U}}$ is just a multigraph with vertex set $[n]$ (Figure 5). For each $k \in \mathbb{N}$, we define the alphabet $\mathbf{\Sigma}(k)$ as the set of all unit slices of width at most k. We let $\mathbf{\Sigma}(k)^{\circledast}$ denote the set of all unit decompositions over $\mathbf{\Sigma}(k)$.

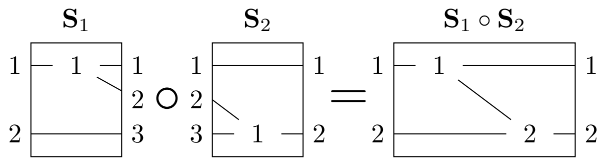

Figure 3 Gluing of two slices $\mathbf{S}_1$ and $\mathbf{S}_2$. The gluing operation is a way to merge two slices into one. In this example, the edge from the center vertex 1 from $\mathbf{S}_1$ to the out-frontier vertex $(1, 2)$ is stitched to the edge from the in-frontier vertex $(0, 2)$ to the center vertex 1 from $\mathbf{S}_2$ to form the edge between the center vertices 1 and 2 in $\mathbf{S}_1 \circ \mathbf{S}_2$. The stitching of edges is done following the order of the frontier vertices.

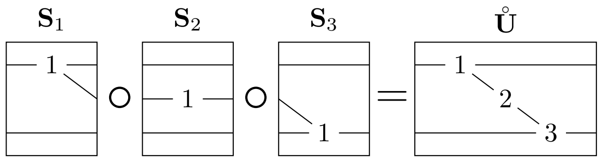

Figure 4 Slice associated to the unit decomposition $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2\mathbf{S}_3$. The gluing operation is associative, therefore parentheses are not needed.

The order of the unit slices in a unit decomposition $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2 \dots \mathbf{S}_n$ induces a linear order $\omega_{\mathbf{U}}$ on the vertices of the slice $\mathring{\mathbf{U}}$. This linear order sets $\omega_{\mathbf{U}}(i) = (0, i)$ for each $(0, i) \in I(\mathbf{S}_1)$, $\omega_{\mathbf{U}}(|I(\mathbf{S}_1)| + i) = i$ for each $i \in \{1, \dots, n\}$ and $\omega_{\mathbf{U}}(|I(\mathbf{S}_1)| + n + i) = (1, i)$ for each $(1, i) \in O(\mathbf{S}_n)$.

Given a unit decomposition $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2 \dots \mathbf{S}_n$ in $\mathbf{\Sigma}(k)^{\circledast}$, we let $\mathbf{w}(\mathbf{U}) = \max_{i \in [n]} \mathbf{w}(\mathbf{S}_i)$.

▶ **Proposition 7.** *Let $k \in \mathbb{N}$, and $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2 \dots \mathbf{S}_n$ be a unit decomposition in $\mathbf{\Sigma}(k)^{\circledast}$, and $\omega_{\mathbf{U}}$ be the linear order induced by $\mathbf{U}$ on $\mathring{\mathbf{U}}$. Then,* $\mathrm{cw}(\mathring{\mathbf{U}}, \omega_{\mathbf{U}}) = \mathbf{w}(\mathbf{U})$.

Proof. This follows by noticing that for each vertex p in $\{1, \dots, |I(\mathbf{S}_1)|\}$, $\mathrm{cw}(\mathring{\mathbf{U}}, \omega_{\mathbf{U}}, p) \leq |I(\mathbf{S}_1)|$, for each p in $\{|I(\mathbf{S}_1)| + n, \dots, |I(\mathbf{S}_1)| + n + |O(\mathbf{S}_n)|\}$, $\mathrm{cw}(\mathring{\mathbf{U}}, \omega_{\mathbf{U}}, p) \leq O(\mathbf{S}_n)$, and for each $p \in \{|I(\mathbf{S}_1)| + 1, \dots, |I(\mathbf{S}_1)| + n\}$, $\mathrm{cw}(\mathring{\mathbf{U}}, \omega_{\mathbf{U}}, p) \leq \mathbf{w}(\mathbf{S}_{i-|I(\mathbf{S}_1)|})$. ◀

▶ **Proposition 8.** *Let G be an n-vertex graph and ω be a linear order on the vertices of G of cutwidth k. Then, we can construct in time $\mathcal{O}(kn)$ a unit decomposition $\mathbf{U}$ such that $\omega = \omega_{\mathbf{U}}$.*

Proof. We will do this construction by first drawing the graph G in the plane. G does not need to be planar for this construction to work. First, we will place the vertices of G on a straight line L isomorphic to $\mathbb{R}$. The ith vertex of G with respect to the linear order ω is placed at the coordinate i on the line. Then edges are drawn as curves between their endpoints. Now, we will draw $n+1$ lines perpendicular to L at coordinates $\{-0.5, 0.5, 1.5, \dots, n-0.5, n+0.5\}$. We call these lines cut-lines. The cutwidth of ω is k, therefore each cut-line intersects at most k edges in the drawing of G. We put a vertex at the intersection of a cut-line and an edge. The graph between two consecutive cut-lines defines a unit slice of width at most k. Taking all those slices in the order induced by ω on the line L gives a unit decomposition $\mathbf{U}$ of width k such that $\omega = \omega_{\mathbf{U}}$. Figure 6 illustrates this construction. ◀

Equivalence of Slices. Let $\mathbf{S}_1 = (I_1, C_1, O_1, E_1)$ and $\mathbf{S}_2 = (I_2, C_2, O_2, E_2)$ be two slices. We say that $\mathbf{S}_1$ is *equivalent* to $\mathbf{S}_2$, denoted by $\mathbf{S}_1 \sim \mathbf{S}_2$, if and only if $I_1 = I_2$, $O_1 = O_2$, $C_1 = C_2$, and there is an isomorphism ϕ from $\mathbf{S}_1$ to $\mathbf{S}_2$ such that the restriction of ϕ to I_1 and O_1 is the identity function. In other words, $\mathbf{S}_1$ and $\mathbf{S}_2$ are equivalent if they are equal up to the renaming of the center vertices.

(a)

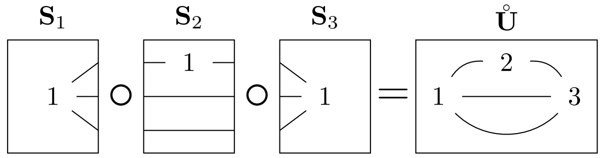

Figure 5 Slice associated to the unit decomposition $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2\mathbf{S}_3$. The resulting slice does not have any vertex in its frontier. It can therefore be seen as a multigraph on 3 vertices.

(b)

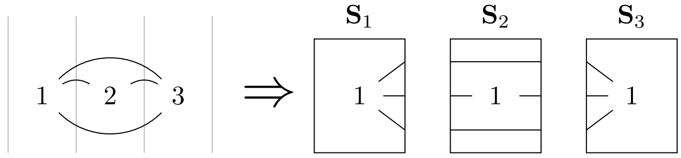

Figure 6 Slicing of the graph G on the left into a unit decomposition $\mathbf{U}$ on the right.

We let $\mathcal{R}(k) \subseteq \mathbf{\Sigma}(k)^2 \times \mathbf{\Sigma}(k)^2$ be the set of all rewriting rules of the form $\mathbf{S}_1\mathbf{S}_2 \to \mathbf{S}'_1\mathbf{S}'_2$ such that $\mathbf{S}_1 \circ \mathbf{S}_2 \sim \mathbf{S}'_1 \circ \mathbf{S}'_2$. Call two unit decompositions $\mathbf{U}, \mathbf{U}' \in \mathbf{\Sigma}(k)^{\circledast}$ *locally* $\mathcal{R}(k)$*-equivalent*, and denote this fact by $\mathbf{U} \overset{k}{\sim} \mathbf{U}'$, if there exist $\mathbf{W}, \mathbf{W}' \in \mathbf{\Sigma}(k)^{\circledast}$ and $\mathbf{S}_1, \mathbf{S}'_1, \mathbf{S}_2, \mathbf{S}'_2 \in \mathbf{\Sigma}(k)$ with $\mathbf{S}_1 \circ \mathbf{S}_2 \sim \mathbf{S}'_1 \circ \mathbf{S}'_2$ such that $\mathbf{U} = \mathbf{W}\mathbf{S}_1\mathbf{S}_2\mathbf{W}'$ and $\mathbf{U}' = \mathbf{W}\mathbf{S}'_1\mathbf{S}'_2\mathbf{W}'$ (Figure 7).

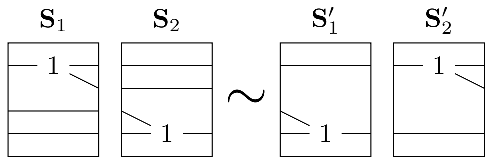

Figure 7 Local Equivalence. $\mathbf{S}_1\mathbf{S}_2$ is (locally) $\mathcal{R}(4)$-equivalent to $\mathbf{S}'_1\mathbf{S}'_2$.

We let $\overset{k}{\equiv} \subseteq \mathbf{\Sigma}(k)^{\circledast} \times \mathbf{\Sigma}(k)^{\circledast}$ be the equivalence relation defined on unit decompositions by taking the reflexive, symmetric and transitive closure of $\overset{k}{\sim}$. If $\mathbf{U} \overset{k}{\equiv} \mathbf{U}'$, then we say that $\mathbf{U}'$ is $\mathcal{R}(k)$*-equivalent* to $\mathbf{U}$. We note that there may exist unit decompositions in $\mathbf{\Sigma}(k)^{\circledast}$ that are not $\mathcal{R}(k)$-equivalent but that are $\mathcal{R}(k')$-equivalent for some $k' > k$.

▶ **Lemma 9.** *Let $k \in \mathbb{N}$ and $\mathbf{U}$ and $\mathbf{U}'$ be unit decompositions in $\mathbf{\Sigma}(k)^{\circledast}$. If $\mathbf{U}$ is $\mathcal{R}(k)$-equivalent to $\mathbf{U}'$, then $\mathring{\mathbf{U}}$ is isomorphic to $\mathring{\mathbf{U}}'$.*

Proof. It is enough to show that that if $\mathbf{U}$ can be transformed into $\mathbf{U}'$ in one $\mathcal{R}(k)$-rewriting step then $\mathring{\mathbf{U}}$ is isomorphic to $\mathring{\mathbf{U}}'$. Therefore, assume that $\mathbf{U} \to \mathbf{U}'$. Then there exist unit decompositions $\mathbf{W}, \mathbf{W}' \in \mathbf{\Sigma}(k)^{\circledast}$ and a rewriting rule $\mathbf{S}_1\mathbf{S}_2 \to \mathbf{S}'_1\mathbf{S}'_2$ in $\mathcal{R}(k)$ such that $\mathbf{U} = \mathbf{W}\mathbf{S}_1\mathbf{S}_2\mathbf{W}'$ and $\mathbf{U}' = \mathbf{W}\mathbf{S}'_1\mathbf{S}'_2\mathbf{W}'$. Since $\mathbf{S}_1 \circ \mathbf{S}_2 \sim \mathbf{S}'_1 \circ \mathbf{S}'_2$, we have an isomorphism φ from $\mathbf{S}_1 \circ \mathbf{S}_2$ to $\mathbf{S}'_1 \circ \mathbf{S}'_2$ that acts as the identity map on frontier vertices. This implies that $\mathring{\mathbf{U}}{=}\mathring{\mathbf{W}} \circ \mathbf{S}_1 \circ \mathbf{S}_2 \circ \mathring{\mathbf{W}}'$ is isomorphic to $\mathring{\mathbf{U}}'{=}\mathring{\mathbf{W}} \circ \mathbf{S}'_1 \circ \mathbf{S}'_2 \circ \mathring{\mathbf{W}}'$. ◀

Twisting. Let $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2 \cdots \mathbf{S}_n$ and $\mathbf{U}' = \mathbf{S}'_1\mathbf{S}'_2 \cdots \mathbf{S}'_n$ be two unit decompositions. We say that $\mathbf{U}$ is a *twisting* of $\mathbf{U}'$ if $\mathring{\mathbf{U}}{=}\mathring{\mathbf{U}}'$. Note that we are not equating slices up to isomorphism. In other words, we are really requiring that the slices $\mathring{\mathbf{U}}$ and $\mathring{\mathbf{U}}'$ are syntactically identical.

Let $\mathbf{S}_1$ and $\mathbf{S}_2$ be unit slices in $\mathbf{\Sigma}(k)$ such that the out-frontier of $\mathbf{S}_1$ and the in-frontier of $\mathbf{S}_2$ have k' vertices for some $k' \leq k$. Given a permutation $\pi : [k'] \to [k']$, let $\mathbf{S}_1^{\pi}$ be the slice obtained by renaming each vertex $(1, i)$ in the out-frontier of $\mathbf{S}_1$ to $(1, \pi(i))$, and let ${}^{\pi}\mathbf{S}_2$

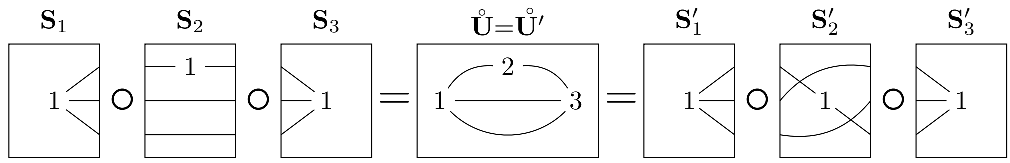

Figure 8 The unit decomposition $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2\mathbf{S}_3$ is a twisting of the unit decomposition $\mathbf{U}' = \mathbf{S}'_1\mathbf{S}'_2\mathbf{S}'_3$. Note that $\mathbf{S}_2 \circ \mathbf{S}_3 = \mathbf{S}'_2 \circ \mathbf{S}'_3$. Note that if we let π be the permutation that sets $\pi(1) = 2$, $\pi(2) = 3$ and $\pi(3) = 1$, then $\mathbf{S}'_2$ is obtained by permuting the out-frontier of $\mathbf{S}_2$ according to π and $\mathbf{S}'_3$ is obtained by permuting the in-frontier of $\mathbf{S}_3$ according to π.

be the slice obtained by renaming each vertex $(0, i)$ in the in-fronter of $\mathbf{S}_2$ to $(0, \pi(i))$. Then it should be clear that $\mathbf{S}_1 \circ \mathbf{S}_2 = \mathbf{S}_1^{\pi} \circ {}^{\pi}\mathbf{S}_2$. Additionally, for each two unit slices $\mathbf{S}'_1$ and $\mathbf{S}'_2$ such that $\mathbf{S}_1 \circ \mathbf{S}_2 = \mathbf{S}'_1 \circ \mathbf{S}'_2$, it should be clear that there is some permutation π such that $\mathbf{S}'_1 = \mathbf{S}_1^{\pi}$ and $\mathbf{S}'_2 = {}^{\pi}\mathbf{S}_2$. Note also that for each two such slices $\mathbf{S}'_1$ and $\mathbf{S}'_2$, the rewriting rule $\mathbf{S}_1\mathbf{S}_2 \to \mathbf{S}'_1\mathbf{S}'_2$ belongs to $\mathcal{R}(k)$. This implies that if a unit decomposition $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2 \ldots \mathbf{S}_n$ is a twisting of a unit decomposition $\mathbf{U}' = \mathbf{S}'_1\mathbf{S}'_2 \ldots \mathbf{S}'_n$, then $\mathbf{U}$ and $\mathbf{U}'$ are $\mathcal{R}(k)$-equivalent and can be transformed into each other by applying a sequence of rewriting rules that "twists" for each $i \in [n-1]$ the out-frontier of $\mathbf{S}_i$ and the in-frontier of $\mathbf{S}_{i+1}$ according to some permutation π_i. This process is illustrated in Figure 8.

▶ **Proposition 10** (Twisting). *Let $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2 \cdots \mathbf{S}_n$ and $\mathbf{U}' = \mathbf{S}'_1\mathbf{S}'_2 \cdots \mathbf{S}'_n$ be two unit decompositions in $\mathbf{\Sigma}(k)^{\circledast}$ such that $\mathbf{U}$ is a twisting of $\mathbf{U}'$. Then, $\mathbf{U}$ can be transformed into $\mathbf{U}'$ by the application of $n-1$ rewriting rules from $\mathcal{R}(k)$.*

An interesting question is whether, for each $k \in \mathbb{N}$, there is some $k' \in \mathbb{N}$ such that any two unit decompositions $\mathbf{U}$ and $\mathbf{U}'$ in $\mathbf{\Sigma}(k)$ are $\mathcal{R}(k')$-equivalent if and only if $\mathring{\mathbf{U}}$ is isomorphic to $\mathring{\mathbf{U}}'$. The answer turns out to be yes, as shown in Theorem 11 below.

▶ **Theorem 11.** *Let $\mathbf{U}$ and $\mathbf{U}'$ be unit decompositions in $\mathbf{\Sigma}(k)^{\circledast}$. Then, $\mathring{\mathbf{U}}$ is isomorphic to $\mathring{\mathbf{U}}'$ if and only if $\mathbf{U}$ and $\mathbf{U}'$ are $\mathcal{R}(2k)$-equivalent.*

Proof. Let $\mathbf{U} = \mathbf{S}_1\mathbf{S}_2 \cdots \mathbf{S}_n$ and $\mathbf{U}' = \mathbf{S}'_1\mathbf{S}'_2 \cdots \mathbf{S}'_n$. Suppose that $\mathbf{U}$ and $\mathbf{U}'$ are $\mathcal{R}(2k)$-equivalent. Then by Lemma 9, $\mathring{\mathbf{U}}$ is isomorphic to $\mathring{\mathbf{U}}'$.

For the converse, suppose that $\mathring{\mathbf{U}}$ is isomorphic to $\mathring{\mathbf{U}}'$ and let φ be an isomorphism from $\mathring{\mathbf{U}}$ to $\mathring{\mathbf{U}}'$. We show that $\mathbf{U}$ and $\mathbf{U}'$ are $\mathcal{R}(2k)$-equivalent.

Given a position $i \in [n-1]$ in the unit decomposition $\mathbf{U}$, a *swap* between $\mathbf{S}_i$ and $\mathbf{S}_{i+1}$ is a rewriting rule in $\mathcal{R}(k')$ for some k' that rewrites $\mathbf{U}$ into the unit decomposition

$$\mathbf{U}_i = \mathbf{S}_1\mathbf{S}_2 \cdots \mathbf{S}_{i-1}\mathbf{S}''_i\mathbf{S}''_{i+1}\mathbf{S}_{i+2} \cdots \mathbf{S}_n$$

such that, the function $\psi : [n] \to [n]$ that sets $\psi(p) = p$ for all $p \notin \{i, i+1\}$, $\psi(i) = i+1$ and $\psi(i+1) = i$ is an isomorphism from $\mathring{\mathbf{U}}$ to $\mathring{\mathbf{U}}_i$.

Intuitively, we swap the center vertex of $\mathbf{S}_i$ with the center vertex of $\mathbf{S}_{i+1}$. Note that there may be several rewriting rules corresponding to such a swap. Now, a swap in the unit decomposition $\mathbf{U}$ corresponds to a swap in $\omega_{\mathbf{U}}$ as defined for linear orders in Section 2. The isomorphism φ defines a transformation of $\omega_{\mathbf{U}}$ into $\omega_{\mathbf{U}'}$.

By Proposition 7, $\mathrm{cw}(\mathring{\mathbf{U}}, \omega_{\mathbf{U}}) \leq k$ and $\mathrm{cw}(\mathring{\mathbf{U}}', \omega_{\mathbf{U}'}) \leq k$. Now, our result in Section 3 can be used for the slice rewriting system $\mathcal{R}(2k)$. More precisely, it follows from Theorem 3 that we can transform $\omega_{\mathbf{U}}$ into $\omega_{\mathbf{U}'}$ by a sequence of $O(n^2)$ swaps and at each step, the cutwidth is at most $2k$. By using the rewriting rules from $\mathcal{R}(2k)$, we can replicate these swaps into the unit decomposition $\mathbf{U}$, obtaining in this way a unit decomposition $\mathbf{U}''$ such that $\omega_{\mathbf{U}''} = \omega_{\mathbf{U}'}$.

Since $\mathring{\mathbf{U}}''=\mathring{\mathbf{U}}'$, we have that $\mathbf{U}''$ is a twisting of $\mathbf{U}'$. Therefore, it follows from Proposition 10 that $\mathbf{U}''$ can be further transformed into $\mathbf{U}'$ by applying a sequence of $n-1$ rewriting rules from $\mathcal{R}(k) \subseteq \mathcal{R}(2k)$.

Hence, $\mathbf{U}$ can be rewritten into $\mathbf{U}''$ by applying $O(n^2)$ rewriting rules from $\mathcal{R}(2k)$. ◀

Theorem 11 allows us to establish connections between the graph isomorphism problem for graphs of cutwidth at most k and the reachability problem in $\mathcal{R}(2k)$.

▶ **Theorem 12** ([10]). *Let G be an n-vertex graph of cutwidth k. We can compute a linear order ω of the vertices of G of width k in time $2^{\mathcal{O}(k^2)} \cdot n$.*

▶ **Theorem 13.** *Graph isomorphism for n-vertex graphs of cutwidth at most k can be reduced in time $2^{\mathcal{O}(k^2)} \cdot n$ to $\mathcal{R}(2k)$-reachability.*

Proof. Given n-vertex graphs G and G' of cutwidth at most k, we first compute in time $2^{\mathcal{O}(k^2)} \cdot n$ linear orders ω and ω' of the vertex sets of G and G', respectively, of cutwidth at most k. Then, from Proposition 8, we construct unit decompositions $\mathbf{U}$ and $\mathbf{U}'$ such that $\omega_{\mathbf{U}} = \omega$, $\omega_{\mathbf{U}'} = \omega'$, G is isomorphic to $\mathring{\mathbf{U}}$ and G' is isomorphic to $\mathring{\mathbf{U}}'$. By Proposition 8, those decompositions belong to $\mathbf{\Sigma}(k)^{\circledast}$. By Theorem 11, we have that $\mathring{\mathbf{U}}$ and $\mathring{\mathbf{U}}'$ are isomorphic if and only if $\mathbf{U}$ and $\mathbf{U}'$ are $\mathcal{R}(2k)$-equivalent. ◀

5 Order Reconfiguration Parameterized by Vertex Separation Number

In this section, we show that the techniques employed in Section 3 for total orders of bounded cutwidth can be generalized to the context of orders of bounded vertex-separation number (Theorem 16). We consider that this generalization may be of independent interest in the theory of reconfiguration, since vertex separation number is a width measure for graphs that is strictly more expressive than cutwidth.

Let G be an n-vertex graph. Given sets $S, S' \subseteq V(G)$, we let $V(G,S,S') = \{u \in S : \exists v \in S' \colon \{u,v\} \in E(G)\}$ be the set of vertices in S that are adjacent to some vertex in S'. As a special case, we define $V(G,S) = V(G,S,V(G) \setminus S)$. We will often make use of the following monotonicity property without explicitly mentioning: If $T \subseteq S$ and $T' \subseteq S'$, then $|V(G,T,T')| \leq |V(G,S,S')|$.

▶ **Definition 14** (Vertex Separation Number). *Let G be an n-vertex undirected graph with vertex set $V(G)$ and edge set $E(G)$. Let $\omega : [n] \to V(G)$ be a linear order on the vertices of G. For each $p \in [n]$, we let*

$$\mathrm{vsn}(G,\omega,p) = |V(G,\omega([p-1]))| = |\{l \in [p-1] \ : \ \exists r \geq p \text{ such that } \{\omega(l),\omega(r)\} \in E(G)\}|.$$

The vertex separation number *of ω is defined as $\mathrm{vsn}(G,\omega) = \max_{p\in[n]} \mathrm{vsn}(G,\omega,p)$. The* vertex separation number *of G is defined as $\mathrm{vsn}(G) = \min_{\omega} \mathrm{vsn}(G,\omega)$, where ω ranges over all linear orders on the vertex set V.*

For each $k \in \mathbb{N}$ and each n-vertex graph G, let $\mathrm{VSN}(G,k) = \{\omega : [n] \to V(G) \ : \ \mathrm{vsn}(G,\omega) \leq k\}$ be the set of linear orders of $V(G)$ of vertex separation number at most k. We say that ω can be *reconfigured* into ω' in vertex separation number at most k if there is a *reconfiguration sequence* $\omega = \omega_0 \xrightarrow{i_1} \omega_1 \xrightarrow{i_2} \cdots \xrightarrow{i_r} \omega_r = \omega'$ such that for each $j \in [r]$, $\omega_j \in \mathrm{VSN}(G,k)$.

▶ **Problem 15** (Bounded Vertex Separation Number Reconfiguration). *Let G be an n-vertex graph, $\omega, \omega' : [n] \to V(G)$ be linear orders on the vertex set of G, and $k \in \mathbb{N}$. Is it true that ω can be reconfigured into ω' in vertex separation number at most k?*

The proof of Theorem 16 below is analogous to the proof of Theorem 3. More precisely, this proof follows by restating Lemma 4, Lemma 5 and Lemma 6 in terms of the vertex separation number of a graph instead of cutwidth, and then by using this last restated lemma to conclude the proof, as done in Theorem 3.

▶ **Theorem 16.** *Let G be an n-vertex graph and $\omega, \omega' : [n] \to V(G)$ be linear orders of $V(G)$ of vertex separation number at most k. Then, ω can be reconfigured into ω' in vertex separation number at most* $\mathrm{vsn}(G, \omega) + \mathrm{vsn}(G, \omega') \leq 2k$.

6 Conclusion

In this work, we have studied the order reconfiguration problem under the framework of the theory of fixed-parameter tractability. In particular, in our main technical result, we have shown that the order reconfiguration problem for orders of cutwidth at most k can always be achieved in cutwidth at most $2k$ (Theorem 3). Using this result, we have established new connections between the graph isomorphism problem and the reachability problem for a special two-letter string rewriting system operating on unit slices. In particular, we have proven that unit decompositions $\mathbf{U}$ and $\mathbf{U}'$ of width k are $R(2k)$-equivalent if and only if the graphs $\mathbf{U}$ and $\mathbf{U}'$ are isomorphic (Theorem 11).

Theorem 11 opens up the possibility of studying the graph isomorphism problem under the perspective of term rewriting theory. The most immediate question in this direction is the complexity of deciding $R(2k)$-reachability for unit decompositions of width k. By a reduction to isomorphism of graphs of cutwidth k, this problem can be solved in time $2^{O(k \cdot \mathrm{polylog} k)} n^{O(1)}$ using the results from [14]. Can techniques that are intrinsic to string/term rewriting theory be used to improve this running time? Can such techniques be used to obtain algorithms running in time $f(k) \cdot n^{O(1)}$ for some computable function $f : \mathbb{N} \to \mathbb{N}$? Note that a positive answer to this question would be conceptually relevant even if the function $f(k)$ is substantially worse than the current $2^{O(k \cdot \mathrm{polylog}(k))}$, since techniques based on rewriting may carry over to contexts where group theoretic techniques do not. One interesting line of attack for this question would be to study connections between $R(2k)$ and techniques related to Knuth-Bendix completion and their generalizations [25, 26, 18, 16].

A natural question that arises in the context of reconfiguration of linear orders is the following: given two linear orders ω and ω', what is the minimum cutwidth of a linear order ω'' occurring in a reconfiguration sequence from ω to ω'? Is this problem NP-hard, or hard to approximate? Is it solvable in FPT-time for certain parameters? We thank one of the reviewers for bringing these interesting questions to our attention.

References

1 Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1999.

2 László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 684–697. ACM, 2016.

3 László Babai, D. Yu. Grigoryev, and David M. Mount. Isomorphism of graphs with bounded eigenvalue multiplicity. In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, STOC '82*, pages 310–324. ACM, 1982.

4 László Babai, William M. Kantor, and Eugene M. Luks. Computational complexity and the classification of finite simple groups. In *24th Annual Symposium on Foundations of Computer Science, FOCS '83*, pages 162–171. IEEE, 1983.

5 Erik Barendsen. *Term Rewriting Systems.* Cambridge University Press, 2003.

6 Ronald V. Book and Friedrich Otto. String-rewriting systems. In *String-Rewriting Systems*, pages 35–64. Springer, 1993.

7 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Paramterized Complexity.* Springer, 2013.

8 John A. Ellis, Ivan H. Sudborough, and Jonathan S. Turner. The vertex separation and search number of a graph. *Information and Computation*, 113(1):50–79, 1994.

9 Merrick L. Furst, John E. Hopcroft, and Eugene M. Luks. Polynomial-time algorithms for permutation groups. In *21st Annual Symposium on Foundations of Computer Science, FOCS '80*, pages 36–41. IEEE Computer Society, 1980.

10 Archontia C. Giannopoulou, Michał Pilipczuk, Jean-Florent Raymond, Dimitrios M. Thilikos, and Marcin Wrochna. Cutwidth: obstructions and algorithmic aspects. *Algorithmica*, 81(2):557–588, 2019.

11 Martin Grohe. Fixed-point definability and polynomial time on graphs with excluded minors. *Journal of the ACM*, 59(5):27, 2012.

12 Martin Grohe and Dániel Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. *SIAM Journal on Computing*, 44(1):114–159, 2015.

13 Martin Grohe, Daniel Neuen, and Pascal Schweitzer. A faster isomorphism test for graphs of small degree. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018*, pages 89–100. IEEE, 2018.

14 Martin Grohe, Daniel Neuen, Pascal Schweitzer, and Daniel Wiebking. An improved isomorphism test for bounded-tree-width graphs. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, pages 67:1–67:14, 2018.

15 Martin Grohe and Pascal Schweitzer. Isomorphism testing for graphs of bounded rank width. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*, pages 1010–1029. IEEE, 2015.

16 Nao Hirokawa, Aart Middeldorp, Christian Sternagel, and Sarah Winkler. Abstract completion, formalized. *Logical Methods in Computer Science*, 15(3), 2019.

17 Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12-14):1054–1065, 2011.

18 Deepak Kapur and Paliath Narendran. The Knuth-Bendix completion procedure and Thue systems. *SIAM Journal on Computing*, 14(4):1052–1072, 1985.

19 Jan Willem Klop. *Term Rewriting Systems.* Centrum voor Wiskunde en Informatica, 1990.

20 Stefan Kratsch and Pascal Schweitzer. Isomorphism for graphs of bounded feedback vertex set number. In Haim Kaplan, editor, *Algorithm Theory - SWAT 2010, 12th Scandinavian Symposium and Workshops on Algorithm Theory*, volume 6139 of *Lecture Notes in Computer Science*, pages 81–92. Springer, 2010.

21 Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. Fixed-parameter tractable canonization and isomorphism test for graphs of bounded treewidth. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014*, pages 186–195. IEEE, 2014.

22 Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42–65, 1982.

23 Gary Miller. Isomorphism testing for graphs of bounded genus. In Raymond E. Miller, Seymour Ginsburg, Walter A. Burkhard, and Richard J. Lipton, editors, *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, STOC '80*, pages 225–235. ACM, 1980.

24 Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018.

25 Christian Sternagel and René Thiemann. Formalizing Knuth-Bendix orders and Knuth-Bendix completion. In Femke van Raamsdonk, editor, *24th International Conference on Rewriting Techniques and Applications, RTA 2013*, volume 21 of *LIPIcs*. Schloss Dagstuhl, Leibniz-Zentrum für Informatik, 2013.

26 Ian Wehrman, Aaron Stump, and Edwin Westbrook. Slothrop: Knuth-Bendix completion with a modern termination checker. In Frank Pfenning, editor, *Term Rewriting and Applications, 17th International Conference, RTA 2006*, volume 4098 of *Lecture Notes in Computer Science*, pages 287–296. Springer, 2006.

27 Marcin Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *Journal of Computer and System Sciences*, 93:1–10, 2018.