On the Parameterized Complexity of Reconfiguration of Connected Dominating Sets

Daniel Lokshtanov
University of California Santa Barbara, Santa Barbara, USA
daniello@ucsb.edu

Amer E. Mouawad
Department of Computer Science, American University of Beirut, Lebanon
aa368@aub.edu.lb

Fahad Panolan
Department of Computer Science and Engineering, IIT Hyderabad, India
fahad@iith.ac.in

Sebastian Siebertz
University of Bremen, Germany
siebertz@uni-bremen.de

Abstract
In a reconfiguration version of an optimization problem $Q$ the input is an instance of $Q$ and two feasible solutions $S$ and $T$. The objective is to determine whether there exists a step-by-step transformation between $S$ and $T$ such that all intermediate steps also constitute feasible solutions. In this work, we study the parameterized complexity of the Connected Dominating Set Reconfiguration problem (CDS-R). It was shown in previous work that the Dominating Set Reconfiguration problem (DS-R) parameterized by $k$, the maximum allowed size of a dominating set in a reconfiguration sequence, is fixed-parameter tractable on all graphs that exclude a biclique $K_{d,d}$ as a subgraph, for some constant $d \geq 1$. We show that the additional connectivity constraint makes the problem much harder, namely, that CDS-R is $W[1]$-hard parameterized by $k + \ell$, the maximum allowed size of a dominating set plus the length of the reconfiguration sequence, already on 5-degenerate graphs. On the positive side, we show that CDS-R parameterized by $k$ is fixed-parameter tractable, and in fact admits a polynomial kernel on planar graphs.

1 Introduction
In an optimization problem $Q$, we are usually asked to determine the existence of a feasible solution for an instance $I$ of $Q$. In a reconfiguration version of $Q$, we are instead given a source feasible solution $S$ and a target feasible solution $T$ and we are asked to determine whether it is possible to transform $S$ into $T$ by a sequence of step-by-step transformations such that after each intermediate step we also maintain feasible solutions. Formally, we consider a graph, called the reconfiguration graph, that has one vertex for each feasible solution and two vertices are connected by an edge if we allow the transformation between the two corresponding solutions. We are then asked to determine whether $S$ and $T$ are connected in the reconfiguration graph, or even to compute a shortest path between them.
Reconfiguration of Connected Dominating Set

Historically, the study of reconfiguration questions predates the field of computer science, as many classic one-player games can be formulated as such reachability questions [16][18], e.g., the 15-puzzle and Rubik’s cube. More recently, reconfiguration problems have emerged from computational problems in different areas such as graph theory [1][14][15], constraint satisfaction [11][23] and computational geometry [4][17][21], and even quantum complexity theory [10]. Reconfiguration problems have been receiving considerable attention in recent literature, we refer the reader to [22][26][30] for an extensive overview.

In this work, we consider the Connected Dominating Set Reconfiguration problem (CDS-R) in undirected graphs. A dominating set in a graph $G$ is a set $D \subseteq V(G)$ such that every vertex of $G$ lies either in $D$ or is adjacent to a vertex in $D$. A dominating set $D$ is a connected dominating set if the graph induced by $D$ is connected. The Dominating Set problem and its connected variant have many applications, including the modeling of facility location problems, routing problems, and many more.

We study CDS-R under the Token Addition/Removal model (TAR model). Suppose we are given a connected dominating set $D$ of a graph $G$, and imagine that a token/pebble is placed on each vertex in $D$. The TAR rule allows either the addition or removal of a single token/pebble at a time from $D$, if this results in a connected dominating set of size at most a given bound $k \geq 1$. A sequence $D_1, \ldots, D_\ell$ of connected dominating sets of a graph $G$ is called a reconfiguration sequence between $D_1$ and $D_\ell$ under TAR if the change from $D_i$ to $D_{i+1}$ respects the TAR rule, for $1 \leq i < \ell$. The length of the reconfiguration sequence is $\ell - 1$. The (Connected) Dominating Set Reconfiguration problem for TAR gets as input a graph $G$, two (connected) dominating sets $S$ and $T$ and an integer $k \geq 1$, and the task is to decide whether there exists a reconfiguration sequence between $S$ and $T$ under TAR using at most $k$ tokens/pebbles.

Structural properties of the reconfiguration graph for $k$-dominating sets were studied in [13][29]. The Dominating Set Reconfiguration problem was shown to be PSPACE-complete in [24], even on split graphs, bipartite graphs, planar graphs and graphs of bounded bandwidth. Both pathwidth and treewidth of a graph are bounded by its bandwidth, hence the Dominating Set Reconfiguration problem is PSPACE-complete on graphs of bounded pathwidth and treewidth. These hardness results motivated the study of the parameterized complexity of the problem. It was shown in [24] that the Dominating Set Reconfiguration problem is W[2]-hard when parameterized by $k+\ell$, where $k$ is the bound on the number of tokens and $\ell$ is the length of the reconfiguration sequence. However, the problem becomes fixed-parameter tractable on graphs that exclude a fixed complete bipartite graph $K_{d,d}$ as a subgraph, as shown in [20]. Such so-called biclique-free classes are very general sparse graph classes, including in particular the planar graphs, which are $K_{3,3}$-free.

In this work we study the complexity of CDS-R. The standard reduction from Dominating Set to Connected Dominating Set shows that also CDS-R is PSPACE-complete, even on graphs of bounded pathwidth (Figure 1).

We hence turn our attention to the parameterized complexity of the problem. We first show that the additional connectivity constraint makes the problem much harder, namely, that CDS-R parameterized by $k+\ell$ is W[1]-hard already on 5-degenerate graphs. As 5-degenerate graphs exclude the biclique $K_{6,6}$ as a subgraph, Dominating Set Reconfiguration is fixed-parameter tractable on much more general graph classes than its connected variant. To prove hardness we first introduce an auxiliary problem that we believe is of independent interest. In the Colored Connected Subgraph problem we are given a graph $G$, an integer $k$, and a coloring $c : V(G) \rightarrow C$, for some color set $C$ with $|C| \leq k$. The question is whether $G$ contains a vertex subset $H$ on at most $k$ vertices such that $G[H]$ is connected and $H$ contains
Figure 1 A graph $G$ with a minimum dominating set of size $k = 2$ marked in dark blue and the graph $H$ obtained in the standard reduction from Dominating Set to Connected Dominating Set. $G$ has a dominating set of size $k$ if and only if $H$ has a connected dominating set of size $k + 1$. If $p$ is equal to the pathwidth of $G$ then the pathwidth of $H$ is bounded by $2p + 1$.

The existence of a reconfiguration sequence of length at most $\ell$ with connected dominating sets of size at most $k$ can be expressed by a first-order formula of length depending only on $k$ and $\ell$. It follows from [12] that the problem is fixed-parameter tractable parameterized by $k + \ell$ on every nowhere dense graph class and the same is implied by [2] for every class of bounded cliquewidth. Nowhere dense graph classes are very general classes of uniformly sparse graphs, in particular the class of planar graphs is nowhere dense. Nowhere dense classes are themselves biclique-free, but are not necessarily degenerate. Hence, our hardness
result on degenerate graphs essentially settles the question of fixed-parameter tractability for the parameter \( k + \ell \) on sparse graph classes. It remains an interesting open problem to find dense graph classes beyond classes of bounded cliquewidth on which the problem is fixed-parameter tractable.

We then turn our attention to the smaller parameter \( k \) alone. We show that CDS-R parameterized by \( k \) is fixed-parameter tractable on the class of planar graphs. Our approach is as follows. We first compute a small domination core for \( G \), a set of vertices that captures exactly the domination properties of \( G \) for dominating sets of sizes not larger than \( k \). The notion of a domination core was introduced in the study of the Distance-
\( r \)-Dominating Set problem on nowhere dense graph classes [3]. While the classification of interactions with the domination core would suffice to solve Dominating Set Reconfiguration on nowhere dense classes, additional difficulties arise for the connected variant. In a second step we use planarity to identify large subgraphs that have very simple interactions with the domination core and prove that they can be replaced by constant size gadgets such that the reconfiguration properties of \( G \) are preserved.

Observe that CDS-R parameterized by \( k \) is trivially fixed-parameter tractable on every class of bounded degree. The existence of a connected dominating set of size \( k \) implies that the diameter of \( G \) is bounded by \( k + 2 \), which in every bounded degree class implies a bound on the size of the graph depending only on the degree and \( k \). We conjecture that CDS-R is fixed-parameter tractable parameterized by \( k \) on every nowhere dense graph class. However, resolving this conjecture remains open for future work (see Figure 1).

The rest of the paper is organized as follows. We give background on graph theory and fix our notation in Section 2. We show hardness of CDS-R on degenerate graphs in Section 3 and finally show how to handle the planar case in Section 4. Due to space constraints proofs of results marked with a * are deferred to the appendix.

## 2 Preliminaries

We denote the set of natural numbers by \( \mathbb{N} \). For \( n \in \mathbb{N} \), we let \([n] = \{1, 2, \ldots, n\} \). We assume that each graph \( G \) is finite, simple, and undirected. We let \( V(G) \) and \( E(G) \) denote the vertex set and edge set of \( G \), respectively. An edge between two vertices \( u \) and \( v \) in a graph is denoted by \( \{u, v\} \) or \( uv \). The open neighborhood of a vertex \( v \) is denoted by \( N_G(v) = \{u \mid \{u, v\} \in E(G)\} \) and the closed neighborhood by \( N_G[v] = N_G(v) \cup \{v\} \). The degree of a vertex \( v \), denoted \( d_G(v) \), is \( |N_G(v)| \). For a set of vertices \( S \subseteq V(G) \), we define \( N_G(S) = \{v \notin S \mid \{u, v\} \in E(G), u \in S\} \) and \( N_G[S] = N_G(S) \cup S \). The subgraph of \( G \) induced by \( S \) is denoted by \( G[S] \), where \( G[S] \) has vertex set \( S \) and edge set \( \{\{u, v\} \in E(G) \mid u, v \in S\} \). We let \( G - S = G[V(G) \setminus S] \). A graph \( G \) is \( d \)-degenerate if every subgraph \( H \subseteq G \) has a vertex of degree at most \( d \). For a set \( C \), we use \( K[C] \) to denote the complete graph on vertex set \( C \). For an integer \( r \in \mathbb{N} \), an \( r \)-independent set in a graph \( G \) is a subset \( U \subseteq V(G) \) such that for any two distinct vertices \( u, v \in U \), the distance between \( u \) and \( v \) in \( G \) is more than \( r \). An independent set in a graph is a 1-independent set. A subset of vertices \( U \) in \( G \) is called a separator in \( G \) if \( G - U \) is has more than one connected component. For \( s, t \in V(G) \), we say that a subset of vertices \( U \) in \( G \) is an \((s, t)\)-separator in \( G \) if there is no path from \( s \) to \( t \) in \( G - U \).


3 Hardness on degenerate graphs

In this section we prove that CDS-R and CCS-R are W[1]-hard when parameterized by \( k + \ell \) even on 5-degenerate and 4-degenerate graphs, respectively. Towards that, we first give a polynomial-time reduction from the W[1]-hard Multicolored Clique problem to CCS-R on 4-degenerate graphs with the property that for an input \((G, c, k)\) of Multicolored Clique the resulting instance of CCS-R admits either a reconfiguration sequence of length \( O(k^3) \) or no reconfiguration sequence at all. As a result, we conclude that CCS-R is W[1]-hard when parameterized by \( k + \ell \) on 4-degenerate graphs. Then, we give a parameter-preserving polynomial-time reduction from CCS-R to CDS-R. Let us first formally define the CCS problem.

| Colored Connected Subgraph (CCS) | Parameter: \( k \) |
|----------------------------------|------------------|
| **Input:** A graph \( G \), \( k \in \mathbb{N} \), and a vertex-coloring \( c \colon V(G) \to C \), where \( |C| \leq k \) | **Question:** Is there a vertex subset \( S \subseteq V(G) \) of at most \( k \) vertices with at least one vertex from every color class such that \( G[S] \) is connected? |

**Reduction from Multicolored Clique to CCS-R.** We now present the reduction from Multicolored Clique to CCS-R, which we believe to be of independent interest. We can assume, without loss of generality, that for an input \((G, c, k)\) of Multicolored Clique, \( G \) is connected and \( c \) is a proper vertex-coloring, i.e., for any two distinct vertices \( u, v \in V(G) \) with \( c(u) = c(v) \) we have \( \{u, v\} \notin E(G) \). Before we proceed let us define a graph operation.

**Definition 3.1.** Let \( G \) be a graph and let \( c \colon V(G) \to \{1, \ldots, k\} \) be a proper vertex coloring of \( V(G) \). Let \( H \) be a graph on the vertex set \( \{1, \ldots, k\} \). We define the graph \( G \upharpoonright c H \) as follows. We remove all edges \( \{u, v\} \in E(G) \) such that \( c(u) = i \) and \( c(v) = j \) and \( \{i, j\} \notin E(H) \). We subdivide every remaining edge, i.e. for every remaining edge \( \{u, v\} \) we introduce a new vertex \( s_{uv} \), remove the edge \( \{u, v\} \) and introduce instead the two edges \( \{u, s_{uv}\} \) and \( \{v, s_{uv}\} \). We write \( W(G \upharpoonright c H) \) for the set of all subdivision vertices \( s_{uv} \) (see Figure 3).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3}
\caption{Construction of \( G \upharpoonright c H \).}
\end{figure}

Let \((G, c, k)\) be the input instance of Multicolored Clique, where \( G \) is a connected graph and \( c \) is a proper \( k \)-vertex-coloring of \( G \). We construct an instance \((H, \hat{c} \colon V(H) \to [k + 1], Q_s, Q_t, 2k)\) of CCS-R.
We first construct a routing gadget. For $1 \leq i \leq k$, let $T^i$ be the star with vertex set $\{1, \ldots, k\}$ having vertex $i$ as the center. For any $1 \leq i \leq k$ and $1 \leq r \leq 20k$, we let $H^{(i,r)}$ be a copy of the graph $G[1 \leq c T^i]$. We let $c_{(i,r)}$ be the partial vertex-coloring of $H^{(i,r)}$ that is naturally inherited from $G$. For an illustration, consider the input instance $(G, c, k)$ of MULTICOLORED CLIQUE depicted in Figure 3a. Then, $T^2$ is identical to the graph $H$ in Figure 3b and Figure 3d represents $H^{(2,r)} = G[1 \leq c T^2]$, for any $1 \leq r \leq 20k$. Now, for $1 \leq i \leq k$ we define a graph $H^i$ as follows. We use $W(H^{(i,r)})$ to denote the set of subdivision vertices in $H^{(i,r)}$. For $1 \leq r \leq 20k$ and all vertices $u, v$ in $V(H^{(i,r)}) \setminus W(H^{(i,r)})$, we connect the copy of the subdivision vertex $s_{uv}$ in $H^{(i,r)}$ (if it exists) with the copies of the vertices $u$ and $v$ in $H^{(i,r+1)}$ (see Figure 4 for an illustration of a portion of $H^1$). We use $W(H^i)$ to denote the set of subdivision vertices $\bigcup_{u \in [20k]} W(H^{(i,r)})$.

For each $1 \leq i \leq k$, we use $c_i$ to denote a coloring on $V(H^i)$ that is a union of $c_{(i,1)}, c_{(i,2)}, \ldots, c_{(i,20k)}$ and we color all the copies of the subdivision vertices using a new color $k + 1$. In other words, we know that for each $u \in V(H^i)$ we have $u \in V(H^{(i,r)})$, for some $r \in \{1, \ldots, 20k\}$. Hence, if $u \in V(H^{(i,k)}) \setminus W(H^{(i,r)})$ then we set $c_i(u) = c_{(i,r)}(u)$. For all $s_{uv} \in W(H^i)$, we set $c_i(s_{uv}) = k + 1$.

Now, define a graph $R$, which is super graph of $H^1 \cup \ldots \cup H^k$, as follows. For $1 \leq i < k$ and all vertices $u$ and $v$, we connect the copy of the subdivision vertex $s_{uv}$ in $H^{(i,20k)}$ (if it exists) with the copies of the vertices $u$ and $v$ in $H^{(i+1,1)}$ (see Figure 5 for an illustration).

We additionally introduce two subgraphs $H^0$ and $H^{k+1}$. The graph $H^0$ is obtained by subdividing each edge of a star on vertex set $\{v_1, \ldots, v_k\}$ centered at $v_1$. Here we use $w_1, \ldots, w_k$ to denote the subdivision vertices. Similarly, the graph $H^{k+1}$ is obtained by subdividing each edge of star on $\{x_1, \ldots, x_k\}$ centered at $x_k$. Here $y_1, \ldots, y_{k-1}$ denote the subdivision vertices. Let $c_0$ and $c_{k+1}$ be the colorings on $\{v_1, \ldots, v_k, w_2, \ldots, w_k\}$ and $\{x_1, \ldots, x_k, y_1, \ldots, y_{k-1}\}$, respectively, defined as follows. For all $1 \leq i \leq k$, $c_0(v_i) = i$ and $c_{k+1}(x_i) = i$. For all $2 \leq i \leq k$, $c_0(w_i) = k + 1$ and for all $1 \leq i \leq k - 1$, $c_{k+1}(y_i) = k + 1$. Observe that we may interpret $H^0$ as $K[{v_1, \ldots, v_k}] \|_{c_0} T^0$ and $H^{k+1}$ as $K[{x_1, \ldots, x_k}] \|_{c_{k+1}} T^{k+1}$, where $T^0$ and $T^{k+1}$ are two trees on vertex set $\{1, \ldots, k\}$, with $E(T^0) = \{\{1, i\} : 2 \leq i \leq k\}$ and $E(T^{k+1}) = \{\{k, i\} : 1 \leq i \leq k - 1\}$.

Finally, for each $2 \leq i \leq k$, we connect the “subdivision vertex” $w_i$ (adjacent to $v_i$ and $v_i$) to all vertices $v \in V(H^{(i,1)})$ colored 1 or $i$, i.e., with $c_{(i,1)}(v) \in \{1, i\}$. For each subdivision vertex $s_{ab} \in W(H^{(k,20k)})$, we connect $s_{ab}$ to $x_k$ and $x_i$, where $k = c_k(a) = c_{(k,20k)}(a)$ and.
Figure 5 Illustration of the subgraph of $R$ induced on $V(H^{(2,20k)}) \cup V(H^{(3,1)})$ constructed from the instance $(G,c,k)$ depicted in Figure 3a. The red edge are some of the “crossing edges”.

$i = c_k(b) = c_{(k,20k)}(b)$. Recall that $s_{ab}$ is adjacent a vertex of color $k$ and a vertex of color $i$, for some $i < k$. This completes the construction of $H$ (see Figure 6). We define \( \hat{c} \colon V(H) \mapsto [k+1] \) to be the union of $c_0, \ldots, c_{k+1}$.

Figure 6 Illustration of connection between $H^0$ and $R$, and $H^{k+1}$ and $R$ from the instance $(G,c,k)$ depicted in Figure 3a. The red edge are some of the “crossing edges” between $H^0$ and $H^1$, and $H^k$ and $H^{k+1}$.

Observation 3.2. The sets \( \{v_1, \ldots, v_k, w_2, \ldots, w_k\} \) and \( \{x_1, \ldots, x_k, y_1, \ldots, y_{k-1}\} \) are solutions of size $2k - 1$ of the CCS instance $(H,\hat{c},2k)$.

We define the starting configuration $Q_s$ as the set \( \{v_1, \ldots, v_k, w_2, \ldots, w_k\} \) and the target configuration $Q_t$ as the set \( \{x_1, \ldots, x_k, y_1, \ldots, y_{k-1}\} \). We now consider the instance $(H,\hat{c},Q_s,Q_t,2k)$ of the CCS-R problem. That is, the bound on the sizes of the solutions in the reconfiguration sequence is at most $2k$. Before we analyze the reconfiguration properties of $H$, let us verify that $H$ is 4-degenerate.

Lemma 3.3 (⋆). The graph $H$ is 4-degenerate.

Lemma 3.4 (⋆). If there exists a $k$-colored clique in $G$ then there is reconfiguration sequence of length $O(k^3)$ from $Q_s$ to $Q_t$ in $(H,\hat{c},2k)$. 
Reconfiguration of Connected Dominating Set

Proof sketch. We aim to shift the connected vertices of $Q_s$ through the subgraphs $H^1, \ldots, H^k$ (in that order) to maintain connectivity and eventually shift all the tokens to $Q_t$. For each $u_i \in V(G)$, $1 \leq j \leq k$ and $1 \leq r \leq 20k$, we use $u_i^{j,r}$ to denote the copy of $u_i$ in $H^{j,r}$. Let $C = \{u_1, \ldots, u_k\}$ be a $k$-colored clique in $G$ such that $c(u_i) = i$, for all $1 \leq i \leq k$. To prove the lemma, we need to define a reconfiguration sequence starting from $Q_s$ and ending at $Q_t$ such that the cardinality of any solution in the sequence is at most $2k$. First we define $k$ “colored” trees $\hat{T}_1, \ldots, \hat{T}_k$ each on $2k - 1$ vertices, and then prove that there are reconfiguration sequences from $Q_s$ to $V(\hat{T}_1)$, $V(\hat{T}_i)$ to $V(\hat{T}_{i+1})$ for all $1 \leq i < k$, and $V(\hat{T}_k)$ to $Q_t$.

We start by defining $\hat{T}_1, \ldots, \hat{T}_k$. For each $1 \leq i \leq k$, $C_i = \{u_1^{i,1}, \ldots, u_k^{i,1}\}$ and $S_i = \{z \in V(H^{i,1}) : N_{H^{i,1}}(z) \cap C_i = 2\}$. That is, for each $1 \leq j \leq k$ and $j \neq i$, $s_{u_i^{j,1}}u_j^{i,1} \in S_i$ (the subdivision vertex on the edge $u_i^{i,1}u_j^{i,1}$ is in $S_i$), and $|S_i| = k - 1$. In other words, $C_i$ contains the copies of the vertices of the clique $C$ in $H^{i,1}$ and $S_i$ contains subdivision vertices corresponding to $k - 1$ edges in the clique incident on the $i$th colored vertex of the clique, such that $H[C_i \cup S_i]$ is a tree. Now, define $\hat{T}_1 = H[C_i \cup S_i]$. It is easy to verify that $\hat{T}_1 = H[C_i \cup S_i]$ is a solution to the CCS instance $(H, \hat{c}, 2k)$. Let $T_s = H[Q_s]$ and $T_t = H[Q_t]$. Note that $T_s$ and $T_t$ are trees on $2k - 1$ vertices.

Case 1: Reconfiguration from $Q_s$ to $V(\hat{T}_1)$. Informally, we move to $\hat{T}_1$ by adding a token on $u_i^{i,1}$ and then removing tokens from $v_i$ for $i$ in the order $2, \ldots, k, 1$ (for a total of $2k$ token additions/removals). Finally, we move the tokens from $\{w_2, \ldots, w_{k-1}\}$ to $S_1$ in $2(k - 1)$ steps. The length of the reconfiguration sequence is $2k + 2(k - 1) = 4k - 2$.

Case 2: Reconfiguration from $V(\hat{T}_i)$ to $V(\hat{T}_{i+1})$. First we define $20k$ trees $P_1, \ldots, P_{20k}$, each on $2k - 1$ vertices such that for all $1 \leq r \leq 20k$, (i) $V(P_r) \subseteq V(H^{i,r})$, and (ii) $\hat{T}_i = P_1$. Then we give a reconfiguration sequence from $V(P_r)$ to $V(P_{r+1})$ for all $r \in [20k - 1]$ and a reconfiguration sequence from $V(P_{20k})$ to $V(\hat{T}_{i+1})$.

Recall that $C = \{u_1, \ldots, u_k\}$ is a $k$-colored clique in $G$ such that $c(u_i) = i$ for all $1 \leq i \leq k$. For $1 \leq r \leq 20k$, let $C_i^r = \{u_1^{i,r}, \ldots, u_k^{i,r}\}$ and $S_i^r = \{z \in V(H^{i,r}) : N_{H^{i,r}}(z) \cap C_i^r = 2\}$. That is, for each $1 \leq j \leq k$ and $j \neq i$, $s_{u_i^{i,r}}u_j^{i,r} \in S_i^r$ (i.e, the subdivision vertex on the edge $u_i^{i,r}u_j^{i,r}$ is in $S_i^r$) and $|S_i^r| = k - 1$. Let $P_s = H[C_i^r \cup S_i^r]$. Notice that for all $r \in [20k]$, $P_r$ is a tree on $2k - 1$ vertices. Moreover, for each $1 \leq r \leq 20k$, $V(P_r)$ is a solution to the CCS instance $(H, \hat{c}, 2k)$. By arguments similar to those given for Case 1, one can prove that there is a reconfiguration sequence of length $4k - 2$ from $V(P_r)$ to $V(P_{r+1})$, for all $1 \leq r < 20k$.

For the reconfiguration sequence from $V(P_{20k})$ to $V(\hat{T}_{i+1})$ we refer the reader to the complete proof in the appendix.

Case 3: Reconfiguration from $V(\hat{T}_k)$ to $V(T_t)$. The arguments for this case are similar to those given in Case 1, we therefore omit the details.

Lemma 3.5. If there is a reconfiguration sequence from $Q_s$ to $Q_t$ then there is a $k$-colored clique in $G$.

Theorem 3.6. CCS-R parameterized by $k + \ell$ is W[1]-hard on 4-degenerate graphs.

Reduction from CCS-R to CDS-R. We give a polynomial-time parameter-preserving reduction from CCS-R to CDS-R that is fairly straightforward. Let $(G, c, S, T, k)$ be an instance of CCS-R. Let $c : V(G) \mapsto \{1, \ldots, k’\}$, where $k’ \leq k$. We construct a graph $H$ as follows. For each $1 \leq i \leq k’$, we add a vertex $d_i$ and connect $d_i$ to all the vertices in $c^{-1}(i)$. 

\[\text{Proof sketch.} \]
Next, for each $1 \leq i \leq k'$, we add $2k + 1$ pendent vertices to $d_i$. That is, we add vertices 
\{x_{i,j}: 1 \leq i \leq k', 1 \leq j \leq 2k + 1\} and edges 
\{x_{i,j}, d_i\}: 1 \leq i \leq k', 1 \leq j \leq 2k + 1\}. Let 
$D = \{d_1, \ldots, d_{k'}\}$. We output $(H, S \cup D, T \cup D, k + k')$ as the new CDS-R instance.

\section*{4 Fixed-parameter tractability on planar graphs}

This section is devoted to proving that CDS-R under TAR parameterized by $k$ is fixed-parameter tractable on planar graphs. In fact, we show that the problem admits a polynomial kernel. Recall that a kernel for a parameterized problem $Q$ is a polynomial-time algorithm that computes for each instance $(I, k)$ of $Q$ an equivalent instance $(I', k')$ with $|I'| + k' \leq f(k)$ for some computable function $f$. The kernel is polynomial if the function $f$ is polynomial.

We prove that for every instance $(G, S, T, k)$ of CDS-R, with $G$ planar, we can compute in polynomial time an instance $(G', S, T, k)$ where $|V(G')| \leq p(k)$ for some polynomial $p$, $G'$ planar, and where there exists a reconfiguration sequence under TAR from $S$ to $T$ in $G$ (using at most $k$ tokens) if and only if such a sequence exists in $G'$.

Our approach is as follows. We first compute a small domination core for $G$, that is, a set of vertices that captures exactly the domination properties of $G$ for dominating sets of sizes not larger than $k$. While the classification of interactions with the domination core would suffice to solve DOMINATING SET RECONFIGURATION, additional difficulties arise for the connected variant. In a second step we use planarity to identify large subgraphs that have very simple interactions with the domination core and prove that they can be replaced by constant size gadgets such that the reconfiguration properties of $G$ are preserved.

\subsection*{4.1 Domination cores}

\begin{definition}
Let $G$ be a graph and let $k \geq 1$ be an integer. A $k$-domination core is a subset $C \subseteq V(G)$ of vertices such that every set $X \subseteq V(G)$ of size at most $k$ that dominates $C$ also dominates $G$.
\end{definition}

\section*{5 Dominating Set Reconfiguration on planar graphs}

This completes the proof.\[\boxend\]
It is not difficult to see that DOMINATING SET is fixed-parameter tractable on all graphs that admit a \( k \)-domination core of size at most \( f(k) \) that is computable in time \( g(k) \cdot n^c \), for any computable functions \( f, g \) and constant \( c \). This approach was first used (implicitly) in [3] to solve DISTANCE-\( r \) DOMINATING SET on nowhere dense graph classes. In case \( k \) is the size of a minimum (distance-\( r \)) dominating set, one can establish the existence of a linear size \( k \)-domination core on classes of bounded expansion [5] (including the class of planar graphs) and a polynomial size (in fact an almost linear size) \( k \)-domination core on nowhere dense graph classes [7,19]. If \( k \) is not minimum, there exist classes of bounded expansion such that a \( k \)-domination core must have at least quadratic size [6]. The most general graph classes that admit \( k \)-domination cores are given in [8]. Moreover, DOMINATING SET RECONFIGURATION and DISTANCE-\( r \) DOMINATING SET RECONFIGURATION are fixed-parameter tractable on all graphs that admit small (distance-\( r \)) \( k \)-domination cores [20,28].

**Lemma 4.2.** There exists a polynomial \( p \) such that for all \( k \geq 1 \), every planar graph \( G \) admits a polynomial-time computable \( k \)-domination core of size at most \( p(k) \).

The lemma is implied by Theorem 1.6 of [19] by the fact that planar graphs are nowhere dense. We want to stress again that the polynomial size of the \( k \)-domination core results from the fact that \( k \) may not be the size of a minimum dominating set, if \( k \) is minimum we can find a linear size core. Explicit bounds on the degree of the polynomial can be derived from [25,27], but we refrain from doing so to not disturb the flow of ideas.

The following lemma is immediate from the definition of a \( k \)-domination core.

**Lemma 4.3.** If \( D \) is a dominating set of size at most \( k \) that contains a vertex set \( W \subseteq D \) such that \( N[D] \cap C = N[D \setminus W] \cap C = C \), then \( D \setminus W \) is also a dominating set.

**Definition 4.4.** Let \( G \) be a graph and let \( A \subseteq V(G) \). The projection of a vertex \( v \in V(G) \setminus A \) into \( A \) is the set \( N(v) \cap A \). If two vertices \( u, v \) have the same projection into \( A \) we write \( u \sim_A v \).

Obviously, the relation \( \sim_A \) is an equivalence relation. The following lemma is folklore, one possible reference is [9].

**Lemma 4.5.** Let \( G \) be a planar graph and let \( A \subseteq V(G) \). Then there exists a constant \( c \) such that there are at most \( c \cdot |A| \) different projections to \( A \), that is, the equivalence relation \( \sim_A \) has at most \( c \cdot |A| \) equivalence classes.

### 4.2 Reduction rules

Let \( G \) be an embedded planar graph. We say that a vertex \( v \) touches a face \( f \) if \( v \) is drawn inside \( f \) or belongs to the boundary of \( f \) or is adjacent to a vertex on the boundary of \( f \). We fix two connected dominating sets \( S \) and \( T \) of size at most \( k \). We will present a sequence of lemmas, each of which implies a polynomial-time computable reduction rule that allows to transform \( G \) to a planar graph \( G' \) that inherits its embedding from \( G \), with \( S, T \subseteq V(G') \) and that has the same reconfiguration properties with respect to \( S \) and \( T \) as \( G \). To not overload notation, after stating a lemma with a reduction rule, we assume that the reduction rule is applied until this is no longer possible and call the resulting graph again \( G \). We also assume that whenever one or more of our reduction rules are applicable, then they are applied in the order presented. We will guarantee that \( S \) and \( T \) will always be connected dominating sets of size at most \( k \), hence, after each application of a reduction rule, we can recompute a \( k \)-domination core in polynomial time. This yields only polynomial overhead
and allows us to assume that we always have marked a $k$-domination core $C$ of size at most $p := p(k)$ as described in Lemma 4.2. This allows us to state the lemmas as if $G$ and $C$ were fixed. Without loss of generality we assume that $C$ contains $S$ and $T$.

**Definition 4.6.** A set $W \subseteq V(G) \setminus C$ of vertices is *irrelevant* if there is a reconfiguration sequence from $S$ to $T$ in $G$ if and only if there is a reconfiguration sequence from $S$ to $T$ in $G - W$.

**Definition 4.7.** Let $u, v \in V(G)$ be non-equal vertices. We call the set $D(u, v) := (N(u) \cap N(v)) \cup \{u, v\}$ the *diamond* induced by $u$ and $v$. We call $|N(u) \cap N(v)|$ the *thickness* of $D(u, v)$.

**Lemma 4.8.** If $G$ contains a diamond $D(u, v)$ of thickness greater than $3k$, then at least one of $u$ or $v$ must be pebbled in every reconfiguration sequence from $S$ to $T$.

**Proof.** Assume $S = S_1, \ldots, S_t = T$ is a reconfiguration sequence from $S$ to $T$ and $u, v \notin S_i$ for some $1 \leq i \leq t$. Then every $s \in S_i$ can dominate at most 3 vertices of $N(u) \cap N(v)$: otherwise $u, v, s$ together with 3 vertices of $N(u) \cap N(v)$ different from $u, v$ and $s$ would form a complete bipartite graph $K_{3,3}$.

![Figure 7](image.png)

**Figure 7** A vertex $s \in S_i$ can dominate at most 3 vertices of $N(u) \cap N(v)$.

**Lemma 4.9.** If $G$ contains a diamond $D(u, v)$ of thickness greater than $3k$, then we can remove all internal edges in $D(u, v)$, i.e., edges with both endpoints in $N(u) \cap N(v)$.

**Proof.** Assume $S = S_1, \ldots, S_t = T$ is a reconfiguration sequence from $S$ to $T$. According to Lemma 4.8 for each $1 \leq i \leq t$, $S_i \cap \{u, v\} \neq \emptyset$. Hence all vertices of $N(u) \cap N(v)$ are always dominated by at least one of $u$ or $v$, say by $u$. Moreover, pebbling more than one vertex of $N(u) \cap N(v)$ will never create connectivity via internal edges that is not already there via edges incident on $u$. In other words, for any connected dominating set $S$ of $G$, if an edge $yz$ is used for connectivity, where $y, z \in N(u) \cap N(v)$, then this edge can be replaced by either the path $yz$ or the path $yvz$ (depending on which of $u$ or $v$ is in $S$).

As described earlier, we now apply the reduction rule of Lemma 4.9 until this is no longer possible, and name the resulting graph again $G$. As we did not make use of the properties of a $k$-domination core in the lemma, it is sufficient to recompute a $k$-domination core $C$ after applying the reduction rule exhaustively. In the following it may be necessary to recompute it after each application of a reduction rule. We will not mention these steps explicitly anymore in the following.

**Lemma 4.10 (⋆).** If $G$ contains a diamond $D(u, v)$ of thickness greater than $4|C| + 3k + 1$ then $G$ contains an irrelevant vertex.

We may in the following assume that $G$ does not contain diamonds of thickness greater than $4|C| + 3k + 1$. 
Reconfiguration of Connected Dominating Set

**Corollary 4.11.** If a vertex \( v \in V(G) \) has degree greater than \((4|C| + 3k + 1) \cdot k\), then the token on \( v \) is never lifted throughout a reconfiguration sequence.

**Proof.** Assume \( S = S_1, \ldots, S_t = T \) is a reconfiguration sequence from \( S \) to \( T \) in \( G \) and assume there is \( S_i \) with \( v \notin S_i \). The dominating set \( S_i \) has at most \( k \) vertices and must dominate \( N(v) \). Hence, there must be one vertex \( u \in S_i \) that dominates at least a \( \frac{1}{k} \) fraction of \( N(v) \), which is larger than \( 4|C| + 3k + 1 \). Then there is a diamond \( D(u, v) \) of thickness greater than \( 4|C| + 3k + 1 \), which does not exist after application of the reduction rule of Lemma 4.10.

According to Corollary 4.11, the only vertices that can have high degree after applying the reduction rules are vertices that are never lifted throughout a reconfiguration sequence. This gives rise to another reduction rule that is similar to the rule of Lemma 4.9.

**Lemma 4.12.** Assume \( v \) is a vertex of degree greater than \((4|C| + 3k + 1) \cdot k\). Then we may remove all edges with both endpoints in \( N(v) \).

**Proof.** Let \( G' \) be the graph obtained from \( G \) by removing all edges with both endpoints in \( N(v) \). We claim that reconfiguration between \( S \) and \( T \) is possible in \( G \) if and only if it is possible in \( G' \). The fact that \( S \) and \( T \) are in fact connected dominating sets in \( G' \) is implied by the argument below.

Assume \( S = S_1, \ldots, S_t = T \) is a reconfiguration sequence from \( S \) to \( T \) in \( G \). We claim that the same sequence is a reconfiguration sequence in \( G' \). According to Corollary 4.11 \( v \in S_i \) for all \( 1 \leq i \leq t \). This implies that \( S_i \) is connected in \( G' \) for all \( 1 \leq i \leq t \), as all \( x, y \in S_i \) that are no longer connected by an edge in \( G' \) but were connected in \( G \) are connected via a path of length 2 using the vertex \( v \). It is also easy to see that \( S_i \) is a dominating set in \( G' \), as all vertices that are no longer dominated by \( s \in S_i \) in \( G \) are still dominated by \( v \). Observe that this in particular implies that \( S \) and \( T \) are connected dominating sets in \( G' \). Vice versa, if \( S = S_1, \ldots, S_t = T \) is a reconfiguration sequence from \( S \) to \( T \) in \( G' \), this is trivially also a reconfiguration sequence in \( G \).

The following reduction rule is obvious.

**Lemma 4.13.** If a vertex \( v \) has more than \( k + 1 \) pendant neighbours, i.e., neighbors of degree exactly one, then it suffices to retain exactly \( k + 1 \) of them in the graph.

**Lemma 4.14.** There are at most \( c|C| \cdot (4|C| + 3k + 1) \) vertices of \( V(G) \setminus C \) that have 2 neighbours in \( C \), where \( c \) is the constant of Lemma 4.9.

**Proof.** According to Lemma 4.5, there are at most \( c|C| \) different projections to \( C \). Each projection class that has at least 3 representatives has size at most 2, as otherwise we would find a \( K_{3,3} \) as a subgraph, contradicting the planarity of \( G \). Consider a class with a projection of size 2 into \( C \). Denote these two vertices of \( C \) by \( u \) and \( v \). If this class has more than \( 4|C| + 3k + 1 \) representatives, then \( D(u, v) \) is a diamond of thickness greater than \( 4|C| + 3k + 1 \), which cannot not exist after exhaustive application of the reduction rule of Lemma 4.10.

We now come to the description of our final reduction rule. Let \( D \) denote the set of vertices containing both \( C \) and all vertices of \( V(G) \setminus C \) having at least two neighbors in \( C \). In other words, \( D \) contains all those vertices in \( V(G) \setminus C \) that have exactly one neighbor in \( C \). According to Lemma 4.14, at most \( c|C| \cdot (4|C| + 3k + 1) \) vertices have two neighbors in \( C \), hence \(|D| \leq c|C| \cdot (4|C| + 3k + 1) + |C| = p \).
Lemma 4.15 (*). Assume there are two vertices \( u \) and \( v \) with degree greater than \( 4p + (4|C| + 3k + 1) \cdot k + 1 \). Let \( \mathcal{P} \) be a maximum set of vertex-disjoint paths of length at least 2 that run between \( u \) and \( v \) using only vertices in \( V(G) \setminus D \). If \(|\mathcal{P}| > 4p + (4|C| + 3k + 1) \cdot k + 1\), then there is \( G' \) such that the instances \((G, S, T, k)\) and \((G', S, T, k)\) are equivalent, \( G' \) is planar, and \(|V(G')| < |V(G)|\).

We are ready to state the final result.

Theorem 4.16. CDS-R under TAR parameterized by \( k \) admits a polynomial kernel on planar graphs.

Proof. Our kernelization algorithm starts by computing (in polynomial time) a \( k \)-domination core \( C \) of size at most \( p := p(k) \) as described in Lemma 4.2. Without loss of generality we assume that \( C \) contains \( S \) and \( T \). After each application of a reduction rule, we recompute the core, giving a polynomial blow-up of the running time. We are left to prove that each reduction rule can be implemented in polynomial time and that we end up with a polynomial number of vertices.

It is clear that the reduction rules of Lemma 4.10, Lemma 4.12 and Lemma 4.13 can easily be implemented in polynomial time. The reduction rule of Lemma 4.15 is slightly more involved, however, we can use a standard maximum-flow algorithm to compute in polynomial time a maximum set of vertex-disjoint paths in a subgraph of \( G \).

It remains to bound the size of \( G \). Recall that we call \( D \) the set of all vertices \( C \) and of all vertices of \( V(G) \setminus C \) that have at least 2 neighbours in \( C \). It follows from Lemma 4.14 that \( D \) has size at most \( c|C| \cdot (4|C| + 3k + 1) + |C| =: p \), where \( c \) is the constant of Lemma 4.5. We are left to bound the number of vertices in \( V(G) \setminus C \) having exactly one neighbour in \( C \) (recall that each vertex in \( V(G) \setminus C \) has at least one neighbour in \( S \cup T \subseteq C \)).

Let \( p' = (4p + (4|C| + 3k + 1) \cdot k + 1) \cdot (4|C| + 3k + 1) \cdot k + k + 1 \), which is still a polynomial in \( k \). Towards a contradiction, assume that there exists an equivalence class \( Q \) in \( \sim_C \) with a projection of size one containing more than \( p' \) vertices. Let \( u \in C \) denote the projection of the aforementioned class. Due to Lemma 4.13, we know that at most \( k + 1 \) of the vertices in \( Q \) are pendant, i.e., adjacent to only one \( u \) in \( G \). Since we cannot apply the reduction rule of Lemma 4.12 any more, we know that there are no edges with both endpoints in \( Q \). Hence, all but \( k + 1 \) vertices of \( Q \) must be adjacent to at least one other vertex in \( V(G) \setminus C \). Let \( R = N_G(Q) \setminus \{u\} \) denote this set of neighbours. No vertex in \( R \) can be adjacent to more than \( 4|C| + 3k + 1 \) vertices of \( Q \), as we cannot apply the reduction rule of Lemma 4.10. The vertices of \( R \) must be dominated by \( S \), and cannot be dominated by \( u \), as otherwise two neighbours of \( u \) would be connected. Hence, there is \( v \in S \) different from \( u \) that dominates at least a \( 1/k \) fraction of \( R \). This implies the existence of at least \( 4p + (4|C| + 3k + 1) \cdot k + 1 \) vertex-disjoint paths of length at least 2 that run between \( u \) and \( v \). But in this case, the reduction rule of Lemma 4.15 is applicable. Therefore, we conclude that \( Q \) cannot exist, obtaining a bound on the size of all equivalence classes of \( \sim_C \), as needed.

References

1. Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Connectedness of the graph of vertex-colourings. Discrete Mathematics, 308(56):913–919, 2008.

2. Bruno Courcelle, Johann A Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. Theory of Computing Systems, 33(2):125–150, 2000.
Reconfiguration of Connected Dominating Set

3 Anuj Dawar and Stephan Kreutzer. Domination problems in nowhere-dense classes. In IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, pages 157–168, 2009.

4 Erik D. Demaine and Joseph O’Rourke. Geometric folding algorithms - linkages, origami, polyhedra. Cambridge University Press, 2007.

5 Pál Grünwald Drange, Markus Sortland Dregi, Fedor V. Fomin, Stephan Kreutzer, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, Felix Reidl, Fernando Sánchez Villaamil, Saket Saurabh, Sebastian Siebertz, and Somnath Sikdar. Kernelization and sparseness: the case of dominating set. In 33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, pages 31:1–31:14, 2016.

6 Eduard Eiben, Mithilesh Kumar, Amer E. Mouawad, Fahad Panolan, and Sebastian Siebertz. Lossy kernels for connected dominating set on sparse graphs. In 35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, pages 29:1–29:15, 2018.

7 Kord Eickmeyer, Archontia C. Giannopoulou, Stephan Kreutzer, O-joung Kwon, Michal Pilipczuk, Roman Rabinovich, and Sebastian Siebertz. Neighborhood complexity and kernelization for nowhere dense classes of graphs. In 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, pages 63:1–63:14, 2017.

8 Grzegorz Fabianski, Michal Pilipczuk, Sebastian Siebertz, and Szymon Toruńczyk. Progressive algorithms for domination and independence. In 36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, pages 27:1–27:16, 2019.

9 Jakub Gajarský, Petr Hlinený, Jan Obrdrzálek, Sebastian Ordyniak, Felix Reidl, Peter Rossmanith, Fernando Sánchez Villaamil, and Somnath Sikdar. Kernelization using structural parameters on sparse graph classes. J. Comput. Syst. Sci., 84:219–242, 2017.

10 Sevag Gharibian and Jamie Sikora. Ground state connectivity of local hamiltonians. In Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming, ICALP 2015, pages 617–628, 2015.

11 Parikshit Gopalan, Phokion G. Kolaitis, Elitza N. Maneva, and Christos H. Papadimitriou. The connectivity of Boolean satisfiability: computational and structural dichotomies. SIAM Journal on Computing, 38(6):2330–2355, 2009.

12 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. Journal of the ACM (JACM), 64(3):17, 2017.

13 Ruth Haas and Karen Seyffarth. The k-dominating graph. Graphs and Combinatorics, 30(3):609–617, 2014.

14 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.

15 Takehiro Ito, Marcin Kamiński, and Erik D. Demaine. Reconfiguration of list edge-colorings in a graph. Discrete Applied Mathematics, 160(15):2199–2207, 2012.

16 Wm. Woolsey Johnson and William E. Story. Notes on the “15” puzzle. American Journal of Mathematics, 2(4):397–404, 1879.

17 Iyad A. Kanj and Ge Xia. Flip distance is in FPT time o(n + k * c^k). In 32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, pages 500–512, 2015.

18 Graham Kendall, Andrew J. Parkes, and Kristian Sproer. A survey of NP-complete puzzles. ICGA Journal, pages 13–34, 2008.

19 Stephan Kreutzer, Roman Rabinovich, and Sebastian Siebertz. Polynomial kernels and wideness properties of nowhere dense graph classes. In Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, pages 1533–1545, 2017.

20 Daniel Lokshtanov, Amer E. Mouawad, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. Reconfiguration on sparse graphs. J. Comput. Syst. Sci., 95:122–131, 2018.
Anna Lubiw and Vinayak Pathak. Flip distance between two triangulations of a point set is NP-complete. *Comput. Geom.*, 49:17–23, 2015.

Amer E. Mouawad. On reconfiguration problems: Structure and tractability. 2015.

Amer E. Mouawad, Naomi Nishimura, Vinayak Pathak, and Venkatesh Raman. Shortest reconfiguration paths in the solution space of boolean formulas. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 985–996, 2015.

Amer E. Mouawad, Naomi Nishimura, Venkatesh Raman, Narges Simjour, and Akira Suzuki. On the parameterized complexity of reconfiguration problems. *Algorithmica*, 78(1):274–297, 2017.

Wojciech Nadara, Marcin Pilipczuk, Roman Rabinovich, Felix Reidl, and Sebastian Siebertz. Empirical evaluation of approximation algorithms for generalized graph coloring and uniform quasi-wideness. In *17th International Symposium on Experimental Algorithms, SEA 2018*, pages 14:1–14:16, 2018.

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

Michał Pilipczuk, Sebastian Siebertz, and Szymon Toruńczyk. On the number of types in sparse graphs. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 799–808. ACM, 2018.

Sebastian Siebertz. Reconfiguration on nowhere dense graph classes. *Electr. J. Comb.*, 25(3):P3.24, 2018.

Akira Suzuki, Amer E. Mouawad, and Naomi Nishimura. Reconfiguration of dominating sets. *Journal of Combinatorial Optimization*, 32(4):1182–1195, 2016.

Jan van den Heuvel. The complexity of change. *Surveys in combinatorics*, 409(2013):127–160, 2013.
A Details omitted from Section 3

We need the following lemma to prove Lemma 3.4.

\begin{itemize}
  \item \textbf{Lemma A.1.} Let $T_1, T_2$ be two trees on vertex set $\{1, \ldots, k\}$ and let $f_1, \ldots, f_{k-1}$ be an ordering of the edges in $T_2$. Then, in polynomial time, we can find an ordering $e_1, \ldots, e_{k-1}$ of the edges in $T_1$ such that the following holds. In the sequence of graphs $T'_0, T'_1, \ldots, T'_{k-1}$ on vertex set $\{1, \ldots, k\}$, where for each $0 \leq i < k - 2$, $T'_i = T'_i + f_i - e_i$ and $T'_0 = T_1$, we have that $T'_i$ is a tree, for all $i \in [k-1]$, and $T'_{k-1} = T_2$.
\end{itemize}

\textbf{Proof.} We proceed by induction on $\ell = |E(T_1) \setminus E(T_2)|$. In the base case, we have $\ell = 0$ and $E(T_1) = E(T_2)$. In this case $f_1, \ldots, f_{k-1}$ is also the required ordering of the edges in $T_1$ (note that the sequence of graphs consists of only $T_1 = T_2$ in this case).

Now consider the induction step, $\ell > 1$. Let $j$ be the first index in $\{1, \ldots, k - 1\}$ such that $f_j \notin E(T_1)$. We add $f_j$ to $T_1$ and this creates a cycle in $T_1$. Hence, there exists an edge $e_j \in E(T_1) \setminus E(T_2)$ whose removal results in a tree. That is, $T'_1 = T_1 + f_j - e_j$ is a tree. Notice that $|E(T'_1) \setminus E(T_2)| = \ell - 1$. By the induction hypothesis, there is a sequence $g_1, \ldots, g_{k-1}$ of edges in $E(T'_1)$ such that for the sequence of graphs $T'_1 = T''_0, T''_1, \ldots, T''_{k-1}$ on vertex set $\{1, \ldots, k\}$, we have $T''_{i+1} = T''_i + f_i - g_i$, each $T''_i$ is a tree, and $T_2 = T''_{k-1}$, $0 \leq i < k$. Since $j$ is the first index in $\{1, \ldots, k - 1\}$ such that $f_j \notin E(T_1)$, $T'_1 = T_1 + f_j - e_j$, and $T''_0, T''_1, \ldots, T''_{k-1}$ are trees, we have that $g_i = f_i$ for all $i < j$. Notice that $f_j \in E(T'_1)$ and $E(T_1) = (E(T'_1) \setminus \{f_j\}) \cup \{e_j\}$.

We claim that $e_1, \ldots, e_{j-1}, e_j, e_{j+1}, \ldots, e_{k-1}$, where $e_i = g_i$ for all $i < j$, is the required sequence of edges in $T_1$. Let $T'_0, T'_1, \ldots, T'_{k-1}$ be the sequence where, for each $0 \leq i < k - 2$, $T'_i = T'_i + f_i - e_i$ and $T'_0 = T_1$. Since $g_i = f_i = e_i$ for all $i < j$, we have that $T'_1 = T'_0 = T'_1 = \ldots = T'_{j-1}$. Moreover, $T'_j = T_1 + \{f_1, \ldots, f_j\} - \{e_1, \ldots, e_j\} = T_1 + \{f_1, \ldots, f_j\} - \{g_1, \ldots, g_j\} = T''_j$ because $E(T_1) = (E(T'_1) \setminus \{f_j\}) \cup \{e_j\}$ and $e_i = g_i$ for all $i < j$. Then, the sequence $T'_j, \ldots, T'_{k-1}$ is the same as the sequence $T''_j, \ldots, T''_{k-1}$. Therefore, the sequence $e_1, \ldots, e_{j-1}, e_j, e_{j+1}, \ldots, e_{k-1}$ of edges in $T_1$ satisfies the conditions of the lemma.

\textbf{Proof of Lemma 3.3}

\textbf{Proof.} We iteratively remove minimum degree vertices and show that we can always remove a vertex of degree at most 4 in each step.

- Every subdivision vertex $v \in W(H^r)$ for $1 \leq i \leq k$ has degree at most 4; it has 4 neighbors in $V(H^r) \cup V(H^{r+1})$.
- After removal of all subdivision vertices the degree of the remaining vertices of each $H^r$ is at most one. That is, a vertex in $H^{(1,4)}$ may have a neighbor in $\{w_2, \ldots, w_k\}$.
- After the removal of $V(H^1) \cup \ldots \cup V(H^k)$, the degree of all vertices except $v_1$ and $x_k$ is at most 2.
- Finally we remove $v_1$ and $x_k$.

\textbf{Proof of Lemma 3.4}

\textbf{Proof.} We aim to shift the connected vertices of $Q_s$ through the subgraphs $H^1, \ldots, H^k$ (in that order) to maintain connectivity and eventually shift to $Q_t$. For each $u_i \in V(G)$, $1 \leq j \leq k$ and $1 \leq r \leq 20k$, we use $u_i^{(j,r)}$ to denote the copy of $u_i$ in $H^{(j,r)}$.

Let $C = \{u_1, \ldots, u_k\}$ be a $k$-colored clique in $G$ such that $c(u_i) = i$, for all $1 \leq i \leq k$.

To prove the lemma, we need to define a reconfiguration sequence starting from $Q_s$ and
ending at \( Q_i \) such that the cardinality of any solution in the sequence is at most \( 2k \). First we define \( k \) “colored” trees \( \tilde{\mathcal{T}}_1, \ldots, \tilde{\mathcal{T}}_k \) each on \( 2k-1 \) vertices, and then prove that there are reconfiguration sequences from \( Q_s \) to \( V(\tilde{\mathcal{T}}_1) \), \( V(\tilde{\mathcal{T}}_i) \) to \( V(\tilde{\mathcal{T}}_{i+1}) \) for all \( 1 \leq i < k \), and \( V(\tilde{\mathcal{T}}_k) \) to \( Q_s \).

We start by defining \( \tilde{\mathcal{T}}_1, \ldots, \tilde{\mathcal{T}}_k \). For each \( 1 \leq i \leq k \), \( C_i = \{ u^{(i,1)}_1, \ldots, u^{(i,1)}_k \} \) and \( S_i = \{ z \in V(H^{(i,1)}): N_{H^{(i,1)}}(z) \cap C_i = 2 \} \). That is, for each \( 1 \leq j \leq k \) and \( j \neq i \), \( s_{u^{(i,1)}_j u^{(i,1)}_j} \in S_i \) (the subdivision vertex on the edge \( u^{(i,1)}_j u^{(i,1)}_j \) is in \( S_i \)), and \( |S_i| = k - 1 \). In other words, \( C_i \) contains the copies of the vertices of the clique \( C \) in \( H^{(i,1)} \) and \( S_i \) contains subdivision vertices corresponding to \( k-1 \) edges in the clique incident on the \( i \)th colored vertex of the clique, such that \( H[C_i \cup S_i] \) is a tree. Now, define \( \tilde{\mathcal{T}}_1 = H[C_i \cup S_i] \). It is easy to verify that \( \tilde{\mathcal{T}}_i = H[C_i \cup S_i] \) is a solution to the CCS instance \((H, \tilde{c}, 2k)\). Let \( T_s = H[Q_s] \) and \( T_i = H[Q_i] \). Note that \( T_s \) and \( T_i \) are trees on \( 2k-1 \) vertices each.

**Case 1: Reconfiguration from \( Q_s \) to \( V(\tilde{\mathcal{T}}_1) \).** Informally, we move to \( \tilde{\mathcal{T}}_1 \) by adding a token on \( u^{(i,1)}_1 \) and then removing tokens from \( v_i \) for \( i \) in the order \( 2, \ldots, k \) (for a total of \( 2k \) token additions/removals). Finally, we move the tokens from \( \{ w_2, \ldots, w_{k-1} \} \) to \( S_1 \) in \( 2(k-1) \) steps. The length of the reconfiguration sequence is \( 2k + 2(k-1) = 4k - 2 \).

Formally, we define \( Z_0 = Q_s \) and for each \( 1 \leq j \leq k-1 \), \( Z_{2j-1} = Z_{2j-2} \cup \{ u^{(i,1)}_{j+1} \} \) and \( Z_{2j} = Z_{2j-1} \setminus \{ v_{j+1} \} \). That is, for each \( 1 \leq j \leq k-1 \),

\[
Z_{2j-1} = \{ u^{(i,1)}_1, \ldots, u^{(i,1)}_{j+1}, v_{j+1}, \ldots, v_k, v_1 \} \cup \{ w_1, \ldots, w_{k-1} \}, \quad \text{and} \\
Z_{2j} = \{ u^{(i,1)}_1, \ldots, u^{(i,1)}_{j+1}, v_{j+2}, \ldots, v_k, v_1 \} \cup \{ w_1, \ldots, w_{k-1} \}.
\]

Next, we define \( Z_{2k-1} \) and \( Z_{2k} \) as

\[
Z_{2k-1} = \{ u^{(i,1)}_1, \ldots, u^{(i,1)}_k, v_1 \} \cup \{ w_1, \ldots, w_{k-1} \}, \quad \text{and} \\
Z_{2k} = \{ u^{(i,1)}_1, \ldots, u^{(i,1)}_k, v_1 \} \cup \{ w_1, \ldots, w_{k-1} \}.
\]

It is easy to verify that \( Z_1, \ldots, Z_{2k} \) are solutions to the CCS instance \((H, \tilde{c}, 2k)\). Thus, we now have a reconfiguration sequence \( Z_0, Z_1, \ldots, Z_{2k} \), where \( Z_0 = Q_s \).

Next, we explain how to get a reconfiguration sequence from \( Z_{2k} \) to \( V(\tilde{\mathcal{T}}_1) \). Recall that \( Z_{2k} = C_1 \cup \{ w_1, \ldots, w_{k-1} \} \) and \( V(\tilde{\mathcal{T}}_1) = C_1 \cup S_1 \). Let \( s_j = s_{u^{(i,1)}_j u^{(i,1)}_j} \), for all \( 2 \leq j \leq k \). Notice that \( S_1 = \{ s_2, \ldots, s_k \} \). To obtain a reconfiguration sequence from \( Z_{2k} \) to \( V(\tilde{\mathcal{T}}_1) \), we add \( s_j \) and then remove \( w_j \) for \( j \) in the order \( 2, \ldots, k \). Since \( w_j \) and \( s_j \) connect the same two vertices from \( C_1 \), this reconfiguration sequence will maintain connectivity. Moreover, it is easy to verify that each set in the reconfiguration sequence uses all the colors \( \{1, \ldots, k\} \). Therefore, there exists a reconfiguration sequence of length \( 4k - 2 \) from \( Q_s \) to \( V(\tilde{\mathcal{T}}_1) \).

**Case 2: Reconfiguration from \( V(\tilde{\mathcal{T}}_1) \) to \( V(\tilde{\mathcal{T}}_{i+1}) \).** First we define \( 2k \) trees \( P_1, \ldots, P_{2k} \), each on \( 2k-1 \) vertices such that for all \( 1 \leq r \leq 20k \), (i) \( V(P_r) \subseteq V(H^{(i,r)}) \), and (ii) \( \tilde{\mathcal{T}}_1 = P_1 \). Then we give a reconfiguration sequence from \( V(P_r) \) to \( V(P_{r+1}) \) for all \( r \in [20k-1] \) and a reconfiguration sequence from \( V(P_{20k}) \) to \( V(\tilde{\mathcal{T}}_{i+1}) \).

Recall that \( C = \{ u_1, \ldots, u_k \} \) is a \( k \)-colored clique in \( G \) such that \( c(u_i) = i \) for all \( 1 \leq i \leq k \). For each \( 1 \leq r \leq 20k \), let \( C^r = \{ u^{(i,r)}_1, \ldots, u^{(i,r)}_k \} \) and \( S^r_i = \{ z \in V(H^{(i,r)}): N_{H^{(i,r)}}(z) \cap C^r = 2 \} \). That is, for each \( 1 \leq j \leq k \) and \( j \neq i \), \( s_{u^{(i,r)}_j u^{(i,r)}_j} \in S^r_i \) (i.e., the subdivision vertex on the edge \( u^{(i,r)}_j u^{(i,r)}_j \) is in \( S^r_i \)) and \( |S^r_i| = k - 1 \). Let \( P_r = H[C^r \cup S^r_i] \). Notice that for all \( r \in [20k] \), \( P_r \) is a tree on \( 2k-1 \) vertices. Moreover, for each \( 1 \leq r \leq 20k \), \( V(P_r) \) is a solution to the CCS instance \((H, \tilde{c}, 2k)\).
Reconfiguration of Connected Dominating Set

Case 2(a): Reconfiguration from $V(P_r)$ to $V(P_{r+1})$. By arguments similar to those given for Case 1, one can prove that there is a reconfiguration sequence of length $4k - 2$ from $V(P_r)$ to $V(P_{r+1})$, for all $1 \leq r < 20k$. For completeness we give the details here. Fix an integer $1 \leq r < 20k$. Let $s_j = s_{u_i^r u_j^r}$ and $s_j' = s_{u_i^{r+1} u_j^{r+1}}$ for all $j \in \{1, \ldots, k\} \setminus \{i\}$.

Notice that $S'_i = \{s_j: j \in \{1, \ldots, k\} \setminus \{i\}\}$ and $S'_i^{r+1} = \{s_j': j \in \{1, \ldots, k\} \setminus \{i\}\}$. Now we define $Z_0 = V(P_r) = C'_i \cup S'_i$ and for each $1 \leq j \leq i - 1$, $Z_{2j-1} = Z_{2j-2} \cup \{u_j^{r+1}\}$ and $Z_{2j} = Z_{2j-1} \setminus \{u_j^{(r)}\}$.

That is, for each $1 \leq j \leq i - 1$,

$$Z_{2j-1} = \{u_1^{(r+1)}, \ldots, u_j^{(r+1)}\} \cup \{u_j^{(r)}, \ldots, u_k^{(r)}\} \cup S'_i,$$

and

$$Z_{2j} = \{u_1^{(1)}, \ldots, u_j^{(1)}\} \cup \{u_j^{(r)}, \ldots, u_k^{(r)}\} \cup S'_i.$$

For each $i \leq j \leq k - 1$, $Z_{2j-1} = Z_{2j-2} \cup \{u_j^{(r+1)}\}$ and $Z_{2j} = Z_{2j-1} \setminus \{u_j^{(r)}\}$.

That is, for each $i \leq j \leq k - 1$,

$$Z_{2j-1} = \{u_1^{(r+1)}, \ldots, u_i^{(r+1)}, u_{i+1}^{(r+1)}, \ldots, u_j^{(r+1)}\} \cup \{u_{j+1}^{(r)}, \ldots, u_k^{(r)}\} \cup S'_i,$$

and

$$Z_{2j} = \{u_1^{(1)}, \ldots, u_i^{(1)}, u_{i+1}^{(1)}, \ldots, u_{j-1}^{(1)}\} \cup \{u_j^{(r)}, \ldots, u_k^{(r)}\} \cup S'_i.$$

Next, we define $Z_{2k-1}$ and $Z_{2k}$ as

$$Z_{2k-1} = \{u_1^{(r+1)}, \ldots, u_i^{(r+1)}\} \cup \{u_i^{(r)}, \ldots, u_k^{(r)}\} \cup S'_i,$$

and

$$Z_{2k} = \{u_1^{(1)}, \ldots, u_i^{(1)}\} \cup \{u_i^{(r)}, \ldots, u_k^{(r)}\} \cup S'_i.$$

Next, for each $1 \leq j \leq k - 1$, let $Z_{2k+2j-1} = Z_{2k+2j-2} \cup \{s_j\}$ and $Z_{2k+2j} = Z_{2k+2j-1} \setminus \{s_j\}$. It is easy to verify that $Z_1, \ldots, Z_{4k-2}$ are solutions to the CCS instance $(H, \mathcal{C}, \hat{2k})$ and $Z_0, \ldots, Z_{4k-2}$ is a reconfiguration sequence where $Z_0 = V(P_r)$ and $Z_{4k-2} = V(P_{r+1})$.

Case 2(b): Reconfiguration from $V(P_{20k})$ to $V(\tilde{T}_{i+1})$. Next, we explain how to get a reconfiguration sequence from $V(P_{20k})$ to $V(\tilde{T}_{i+1})$ using Lemma A.1. Recall that $C_{20k}^1 = \{u_1^{(20k)}, \ldots, u_k^{(20k)}\}$ and $S_{20k}^1 = \{z \in V(H^{(20k)}) : N_{H^{(20k)}}(z) \cap C_{20k}^1 = 2\}$. Let $C_{i+1} = \{u_1^{(i+1,1)}, \ldots, u_k^{(i+1,1)}\}$ and $S_{i+1} = \{z \in V(H^{(i+1,1)}) : N_{H^{(i+1,1)}}(z) \cap C_{i+1} = 2\}$. For ease of presentation, let $s_j = s_{u_i^{(20k)} u_j^{(20k)}}$ for all $j \in \{1, \ldots, k\} \setminus \{i\}$. Also, let $s_j' = s_{u_i^{(i+1,1)} u_j^{(i+1,1)}}$ for all $j \in \{1, \ldots, k\} \setminus \{i+1\}$. That is, $S_{i+1}^1 = \{s_j : j \in \{1, \ldots, k\} \setminus \{i+1\}\}$ and $S_{i+1} = \{s_j' : j \in \{1, \ldots, k\} \setminus \{i\}\}$.

Towards proving the required reconfiguration sequence, we give a reconfiguration sequence from $C_{20k}^1 \cup S_{20k}^1$ to $C_{i+1} \cup S_{i+1}^1$ and then from $C_{i+1} \cup S_{i+1}^1$ to $C_{i+1} \cup S_{i+1}$. The reconfiguration sequence from $C_{20k}^1 \cup S_{20k}^1$ to $C_{i+1} \cup S_{i+1}^1$ is similar to the one in Case 1. That is, we add $u_j^{(i+1,1)}$ and delete $u_j^{(20k)}$ for $j$ in the order $1, \ldots, i - 1, i + 1, \ldots, k, i$. This gives a reconfiguration sequence from $C_{20k}^1 \cup S_{20k}^1$ to $Z = C_{i+1} \cup S_{i+1}^1$ of length $2k$.

Next we explain how to get a reconfiguration sequence from $Z = C_{i+1} \cup S_{i+1}^1$ to $C_{i+1} \cup S_{i+1}$. Notice that $H[Z]$ and $\tilde{T}_{i+1} = H[C_{i+1} \cup S_{i+1}]$ are trees. Recall that $T'$ is the star on the vertex set $\{1, \ldots, k\}$ with vertex $i$ being the center, and $T'^{+1}$ is the star on $\{1, \ldots, k\}$ with vertex $i$ being the center. Also, $c_j$ is a coloring on $H'$ which is inherited from the coloring $c$ of $G$. That is, $c_{i+1}(u_j^{(i+1,1)}) = j$ for $1 \leq j \leq k$. Then, $H[Z] = K[C_{i+1}] \cup T'$ and $\tilde{T}_{i+1} = H[C_{i+1} \cup S_{i+1}] = K[C_{i+1}] \cup T'^{+1}$.

Let $\hat{c}_j^{i+1}$, $\hat{e}_j^{i+1}$ be an arbitrary ordering of the edges in $T'^{+1}$. By Lemma A.1, we have a sequence $\hat{c}_1^{i+1}, \ldots, \hat{e}_{k-1}^{i+1}$ of edges in $T'$ such that for the sequence $T_0, T_1, \ldots, T_{k-1}$ on vertex set $\{1, \ldots, k\}$, where for each $0 \leq j \leq k - 2$, $T_{j+1} = T_j + \hat{c}_j^{i+1} - \hat{e}_j$ and $T_0 = T'$, the following holds.
(i) \( T^j_i \) is a tree for all \( 0 \leq j \leq k - 1 \), and
(ii) \( T^j_{k-1} = T^{j+1} \).

This implies that, from the sequences \( e^i_1, \ldots, e^i_{k-1} \) and \( e^{i+1}_1, \ldots, e^{i+1}_{k-1} \), we get a sequence \( f_1, \ldots, f^i_{k-1} \) on \( S^{20k} \) and a sequence \( f^{i+1}_1, \ldots, f^{i+1}_{k-1} \) on \( S^{i+1} \) such that the for the sequence \( L_0, \ldots, L_{2(k-1)} \), where \( L_0 = C_{i+1} \cup \{ f, f^{i+1}\} \) and for all \( 1 \leq j \leq k - 1 \), \( L_{2j-1} = (L_{2j-2} \cup \{ f_i \}) \), \( L_{2j} = L_{2j-1} \setminus \{ f_i \} \) the following holds.

1. \( H[L_0] \) is connected for all \( 0 \leq i \leq k - 1 \), and
2. \( L_{k-1} = S_{i+1} \cup C_{i+1} \).

Here, conditions (1) and (2) follow from conditions (i) and (ii), respectively. Moreover, \( \tilde{c}(L_i) = [k+1] \) for all \( 0 \leq i \leq 2(k-1) \) and \( L_0 = Z \). Thus, \( L_0, \ldots, L_{2(k-1)} \) is a valid reconfiguration sequence from \( Z \) to \( V(\tilde{T}_{i+1}) \). Note that the ordering on the edges implies an ordering by which we can move the subdivision vertices from \( S_i \) to \( S_{i+1} \) without violating connectivity. This implies that there is a reconfiguration sequence from \( V(P_{20k}) \) to \( V(\tilde{T}_{i+1}) \), of length \( 4k - 2 \). Therefore, we have a reconfiguration sequence from \( V(\tilde{T}_i) \) to \( V(\tilde{T}_{i+1}) \) of length \( O(k^3) \).

**Case 3: Reconfiguration from \( V(\tilde{T}_i) \) to \( V(T_i) \).** The arguments for this case are similar to those given in Case 1, we therefore omit the details. By summing up the lengths of reconfiguration sequences, we get that if \((G,c,k)\) is a yes-instance of Multicolored Clique then there is a reconfiguration sequence from \( Q_s \) to \( Q_t \), of length \( O(k^3) \). ▷

**Proof of Lemma 3.5**

**Proof.** For each \( 1 \leq i \leq k + 1 \), let \( Q_i \) be the set of vertices colored by the color \( i \). That is, \( Q_i = \tilde{c}^{-1}(i) \). First, we prove some auxiliary claims. The proofs of the following two claims follow from the construction of \( H \) and the definition of \( \tilde{c} \).

- **Claim 1.** (i) \( Q_1 \cup \ldots \cup Q_k \) is an independent set in \( H \), and (ii) every vertex in \( Q_{k+1} \) is connected to vertices of at most two distinct colors.

- **Claim 2.** Let \( v, w \in V(H) \setminus (V(H^0) \cup V(H^{k+1})) \) be two distinct vertices such that \( \tilde{c}(v) = \tilde{c}(w) \) and \( \tilde{c}(v) \in \{1, \ldots, k\} \). If \( v \) and \( w \) have a common neighbor in \( V(H) \setminus V(H^0) \), then \( v \) and \( w \) are copies of same vertex \( z \in V(G) \).

- **Claim 3.** Let \( Y \subseteq V(H) \) be a vertex subset such that \( \tilde{c}(Y) = \{1, \ldots, k + 1\} \) and \( H[Y] \) is connected. Then, \( |Y| \geq 2k - 1 \).

**Proof.** Let \( B = Y \setminus \tilde{c}^{-1}(k + 1) = Y \cap (Q_1 \cup \ldots \cup Q_k) \). Since \( \tilde{c}(Y) = \{1, \ldots, k + 1\} \), \( |B| \geq k \) and by Claim 1(ii), \( B \) is an independent set in \( H \). By Claim 1(ii), each vertex in \( Q_{k+1} \) is connected to vertices of at most two distinct colors. Thus, since \( H[Y] \) is connected, the claim follows.

Suppose \((H, \tilde{c}, Q_s, Q_t, 2k)\) is a yes-instance of CCS-R. Then, there is a reconfiguration sequence \( D_1, \ldots, D_\ell \) for \( \ell \in \mathbb{N} \), where \( D_1 = Q_s \) and \( D_\ell = Q_t \). Without loss of generality, we assume that the sequence \( D_1, \ldots, D_\ell \) is a minimal reconfiguration sequence. Then, by Claim 3 for each \( i \in [\ell] \), \( 2k - 1 \leq |D_i| \).

Moreover, since \( |D_1| = |D_\ell| = 2k - 1 \), we have that for each even \( i \), \( D_i \) is obtained from \( D_{i-1} \) by a token addition, and for each odd \( i \), \( D_i \) is obtained from \( D_{i-1} \) by a token removal. This also implies that for each even \( i \), \( |D_i| = 2k \), for each odd \( i \), \( |D_i| = 2k - 1 \), and \( \ell \) is odd.

D. Lokshtanov et al. 19
Reconfiguration of Connected Dominating Set

**Claim 4.** Let $i \in [\ell]$ and $|D_i| = 2k - 1$. Then, for all $1 \leq j \leq k$, $|D_i \cap Q_j| = 1$, and $|D_i \cap Q_{k+1}| = k - 1$. Moreover, each vertex in $D_i \cap Q_{k+1}$ will be adjacent to exactly two vertices in $H[D_i]$ and these vertices will be of different colors from $\{1, \ldots, k\}$.

**Proof.** By Claim 1, $Q_1 \cup \ldots \cup Q_k$ is independent and every vertex of $Q_{k+1}$ is adjacent to vertices of at most two different color classes. Hence, we need at least $k - 1$ vertices from $Q_{k+1}$ that make the connections between the vertices of $D_i$ colored with $\{1, \ldots, k\}$. The above statement along with the assumption $|D_i| = 2k - 1$ imply the claim.

**Claim 5.** Let $i \in \{2, \ldots, \ell\}$. Let $v, w \in D_{i+1}$ such that $v, w \notin V(H^0) \cup V(H^{k+1})$, at most one vertex in $\{v, w\}$ is in $V(H^{1,1})$, and $\tilde{c}(v) = \tilde{c}(w) \in \{1, \ldots, k\}$. Then, $v$ and $w$ are copies of the same vertex in $G$. Moreover, $v, w \in V(H^0) \cup V(H^{k+1})$ for some $j \in [k-1]$.

**Proof.** Suppose $v$ and $w$ are not copies of the same vertex $z \in V(G)$. We know that $|D_i| = 2k - 1$ or $|D_i| = 2k$.

**Case 1:** $|D_i| = 2k - 1$. Since $D_i$ is a solution, $D_i$ induces a connected subgraph in $H$. By Claim 3, $|D_i \cap Q_j| = 1$ for all $j \in \{1, \ldots, k\}$ and $|D_i \cap Q_{k+1}| = k - 1$. Also, by Claim 1 (i) $Q_1 \cup \ldots \cup Q_k$ is an independent set in $H$, and (ii) every vertex in $Q_{k+1}$ is connected to vertices of at most two distinct colors. Statements (i) and (ii), and the fact that $|D_i| = 2k - 1$ imply that (iii) $H[D_i]$ is a tree and each vertex in $D_i \cap Q_{k+1}$ is incident to exactly two vertices in $D_i$. Since $|D_{i+1}| = |D_i| + 1$, in reconfiguration step $i + 1$, we add a vertex to obtain $D_{i+1}$. We know that $v \in D_i$. Since, for any color $q \in [k]$, there is exactly one vertex in $D_i$ of color $q$ (i.e., $|D_i \cap Q_q| = 1$), we have that $D_{i+1} = D_i \cup \{w\}$. Moreover, in step $i + 2$, the vertex removed from $D_{i+1}$ will be from $\{v, w\}$ and that vertex will be $v$ (because of the minimality assumption of the length of the reconfiguration sequence). That is, $D_{i+2} = (D_i \cup \{w\}) \setminus \{v\}$. Notice that $|D_i| = |D_{i+2}| = 2k - 1$. Let $b$ a vertex in $D_{i+2}$ which is adjacent to $w$ in $H[D_{i+2}]$. Since $Q_{k+1} \cap D_i = Q_{k+1} \cap D_{i+2}$ and $|D_i| = |D_{i+2}| = 2k - 1$, by Claim 3, the neighbors of $b$ in $H[D_i]$ and $H[D_{i+2}]$ are of the same color. This implies that $b$ is adjacent to $v$ in $H[D_i]$. Thus, we proved that $\{b, w\} \in E(H)$. If $b \in V(H^0)$, then $v, w \in V(H^{1,1})$ which is a contradiction to the assumption. Otherwise, by Claim 2, we conclude that $v$ and $w$ are copies of same vertex.

**Case 2:** $|D_i| = 2k$. In this case $D_{i+1}$ is obtained by removing a vertex from $D_i$. Moreover, $i \geq 3$, because we have two vertices in $D_i$ from $V(H) \setminus D_i$. Since $|D_{i+1}| = 2k - 1$, because of Claim 4, $D_{i+1}$ is obtained by removing the vertex $v$ from $D_i$. That is, $D_{i+1} = D_i \setminus \{v\}$ and $v, w \in D_i$. Then, again by Claim 3 there is $v' \in \{v, w\}$ such that $D_{i-1} \cup \{v'\} = D_i$. Let $w' = \{v, w\} \setminus \{v'\}$. Since $i \geq 3$, we now apply Case 1 with respect to $w' \in D_{i-1}$ and $v' \in D_i$ to complete the proof.

**Claim 6.** For any index $j \in \{1, \ldots, k\}$ and color $q \in \{1, \ldots, k\}$, there exist an odd $i \in \{3, \ldots, \ell\}$ and $r \in \{5k, \ldots, 15k\}$ such that $D_i$ contains a vertex of color $q$ from $V(H^{j,r})$.

**Proof.** Without loss of generality, assume that $k \geq 2$. Moreover, for any odd $i \in [\ell/2]$, there is a vertex common in $D_i$ and $D_{i+2}$ (since $k \geq 2$). This implies that $H[D_1 \cup D_3 \ldots D_{\ell}]$ is a connected subgraph of $H$. Notice that for any $j \in \{1, \ldots, k\}$ and $r \in [20k]$, $V(H^{j,10k})$ is a $(v_1, x_1)$-separator in $H$. Therefore, since $H[D_1 \cup D_3 \ldots D_{\ell}]$ is connected and $v_1, x_1 \in D_1 \cup D_{\ell}$, (i) for any $j \in [k]$ and $r \in [20k]$, there is an odd $i \in \ell]$ such that $D_i$ contains a vertex from $V(H^{j,r})$. Now fix an index $j \in \{1, \ldots, k\}$ and a color $q \in \{1, \ldots, k\}$. By statement (i), there is an odd $i \in \{1, \ldots, \ell\}$ such that $D_i$ contains a vertex from $V(H^{j,10k})$. Since $H[D_1]$ is connected, $|D_1| = 2k - 1$, $D_i \cap V(H^{j,10k}) \neq \emptyset$, and any vertex in $V(H) \setminus \bigcup_{r=5k}^{15k} V(H^{j,r})$
is at distance more that $5k$ (by the construction of $H$), we have that all the vertices in $D_t$ belong to $\bigcup_{r=5k}^{15k} V(H^{(j,r)})$. Moreover, by Claim 4.9, $D_t$ contains a vertex colored $q$ and that will also be present in $\bigcup_{r=5k}^{15k} V(H^{(j,r)})$. This completes the proof of the claim.

\begin{itemize}
  \item Claim 7. For any color $q \in \{1, \ldots, k\}$, the vertices of color $q$ from $\bigcup_{r=2}^{5} V(H^r)$ used in the reconfiguration sequence $D_1, \ldots, D_t$ are copies of the same vertex $z \in V(G)$. Moreover, exactly one vertex from $V(H^j)$ of color $q$ is used in the reconfiguration for all $2 \leq j \leq k$.

  \textbf{Proof.} Fix a color $q \in \{1, \ldots, k\}$. By Claim 6 there are vertices of color $q$ from $V(H^j)$ for all $j$ is used in the reconfiguration sequence. By Claim 5 all these vertices are copies of the same vertex $z \in V(G)$.

  Now we define a $k$-size vertex subset $C \subseteq V(G)$ and prove that $C$ is a clique in $G$. We let $C = \{a_i \in V(G) : 1 \leq i \leq k, c(a_i) = i\}$, and the copy of $a_i$ in $V(H^2)$ is used in $D_1, \ldots, D_t$.

  Because of Claim 7, we have that $|C| = k$ and $C$ contains a vertex of each color in $C = \{a_1, \ldots, a_k\} \subseteq V(G)$ and for each $q \in [k]$, $c(a_q) = q$. We now prove that $C$ is indeed a clique in $G$. Towards that, we need to prove that for each $1 \leq q < j \leq k$, $\{a_q, a_j\} \in E(G)$.

  \item Claim 8. Let $1 \leq q < j \leq k$. Then, $\{a_q, a_j\} \in E(G)$.

  \textbf{Proof.} By Claim 6 we know that there exist an odd $i \in [t]$ and $r \in \{5k, \ldots, 15k\}$ such that $D_i$ contains a vertex of color $q$ in $V(H^{(j,r)})$. Thus, by Claim 7 a copy of $a_j$ and a copy of $a_q$ are present in $D_i$. Let $u_j$ and $u_q$ be the vertices in $D_i$ colored with $j$ and $q$, respectively. By Claim 7 $u_j$ is a copy of $a_j$ and $u_q$ is a copy of $a_q$. Any vertex $b$ in $V(H^r)$ colored $k+1$ is adjacent to vertices of exactly two colors, out of which one color is $j$. Moreover, by the construction of $H$, (a) if $b$ is adjacent to $x$ and $y$ in $V(H^r)$, and $x$ and $y$ are copies of $x'$ and $y'$ in $G$, respectively, then $\{x', y'\} \in E(G)$. We know that $H[D_i]$ is connected, $|Q_s \cap D_i| = 1$ for all $1 \leq s \leq k$, $D_i \setminus Q_{k+1}$ is an independent set in $H$, and each vertex in $D_i$ colored with $k+1$ is adjacent to exactly two vertices in $D_i \setminus Q_{k+1}$ with one of them being $u_j$ (see Claims 1 and 3). This implies that there is common neighbor $b$ for $u_q$ and $u_j$ and hence $\{a_q, a_j\} \in E(G)$, by statement (a) above. This completes the proof of the claim.

This completes the proof of the lemma.

\end{itemize}

\section{Details omitted from Section 4}

\begin{proof}[Proof of Lemma 4.10]
Let $H$ be the subgraph of $G$ induced by $D(u,v)$. We enumerate the vertices of $N(u) \cap N(v)$ consecutively as $x_1, \ldots, x_t$ for some $t > 4|C| + 3k + 1$. We let $X = \{x_1, \ldots, x_t\}$. Note that since we have $t$ vertex-disjoint paths between $u$ and $v$ in $H$, these paths define the boundaries of $t$ faces in the plane embedding of $H$ (after applying the reduction rule of Lemma 4.9). $H$ has all the edges $\{u, x\}$ and $\{v, x\}$ for $x \in N(u) \cap N(v)$ and no other edges). Each vertex in $C \setminus \{u, v\}$ can be adjacent in $H$ to at most two vertices in $X$, say with $y$ and $z$, and these two vertices $y$ and $z$ can touch at most 3 faces of $H$.

This leaves $|C| + 3k + 1 > |C| + 1$ faces of $H$ that are not touched by a vertex of $C \setminus \{u, v\}$. By the pigeonhole principle we can find 2 adjacent faces $f$ and $g$ of $H$ that are not touched by a vertex of $C \setminus \{u, v\}$.

We let $x_1$ and $x_2$ denote the two vertices on the boundary of face $f$ different from $u$ and $v$ and we let $x_3$ and $x_4$ denote the two vertices on the boundary of face $g$ different from $u$ and $v$. Recall that, due to Lemma 4.9, we know that there are no edges between those three vertices. Let $W$ denote the set of all vertices contained in the face of the cycle $u, x_1, v, x_3, u,$
Reconfiguration of Connected Dominating Set

Figure 8 Every vertex of $C \setminus \{u,v\}$ can touch at most 3 faces of $H$. In the figure we assume the vertices $c_1$ and $c_2$ are in $C \setminus \{u,v\}$. The faces that are touched by $c_1$ or $c_2$ are colored in blue. The uncolored faces $f$ and $g$ are not touched by vertices of $C \setminus \{u,v\}$.

In particular, $W$ contains $x_2$. We claim that the vertices of $W$ can be removed from $G$ without changing the reconfiguration properties of $G$, i.e., $W$ is a set of irrelevant vertices. Let $G' = G - W$. First observe that $W \cap (S \cup T) = \emptyset$, hence $S, T \subseteq V(G')$. We show that reconfiguration from $S$ to $T$ is possible in $G$ if and only if reconfiguration from $S$ to $T$ is possible in $G'$.

Assume $S = S_1, \ldots, S_t = T$ is a reconfiguration sequence from $S$ to $T$ in $G$. Let $S_1', \ldots, S_t'$, where for $1 \leq i \leq t$, $S_i' := S_i$ if $S_i$ does not contain a vertex of $W$ and $S_i' := (S_i \setminus W) \cup \{x_1\}$ otherwise. Note that this modification leaves $S$ and $T$ unchanged, hence, $S_1' = S_1$ and $S_t' = S_t$. We claim that $S_1', \ldots, S_t'$ is a reconfiguration sequence from $S$ to $T$ in $G'$.

Claim 1. For $1 \leq i \leq t$, $S_i'$ is a dominating set of $G$, and hence also of $G'$.

Proof. No vertex of $W$ is adjacent to a vertex of $C \setminus \{u,v\}$ and $W \cap C = \emptyset$ by construction. Hence, the only vertices of $C$ that are possibly adjacent to a vertex of $W$ are the vertices $u$ and $v$. Whenever $S_i$ contains a vertex of $W$, we have $x_1 \in S_i'$, which dominates both $u$ and $v$. Hence, $S_i'$ dominates at least the vertices of $C$ that $S_i$ dominates. We use Lemma 4.3 to conclude that $S_i'$ is a dominating set of $G$.

Claim 2. For $1 \leq i \leq t$, $S_i'$ is connected.

Proof. Let $s_1, s_2 \in S_i \setminus W$ and let $P$ be a shortest path between $s_1$ and $s_2$ in $G[S_i]$. We have to show that there exists a path between $s_1$ and $s_2$ in $G[S_i']$. If $P$ does not use a vertex of $W$, then there is nothing to show. Hence, assume $P$ uses a vertex of $W$. By definition of $W$, both $s_1$ and $s_2$ lie outside the face $h$ of the cycle $u, x_1, v, x_3$ that contains $x_2$. Hence, $P$ must enter and leave the face $h$, and as $P$ is a shortest path, it must enter and leave via opposite vertices, i.e., via $u$ and $v$, or via $x_1$ and $x_3$ (as all other pairs are linked by an edge and we could find a shorter path). If $P$ contains $u$ and $v$, then we can replace the vertices of $W$ on $P$ by $x_1$ and we are done.

Hence, assume $P$ uses $x_1$ and $x_3$. As $D(u,v)$ is a diamond of thickness greater than $4p + 3k + 1 > 3k$, according to Lemma 4.8 at least one of the vertices $u$ and $v$, say $u$, is contained in $S_i$, and by definition also in $S_i'$. Then we can replace the vertices of $W$ on $P$ by $u$ and we are again done.

Finally, the following claim is immediate from the definition of each $S_i'$. Combining Claim 1, 2, and 3, we conclude that $S_1', \ldots, S_t'$ is a reconfiguration sequence from $S$ to $T$ in $G'$.

Claim 3. $S_{i+1}'$ is obtained from $S_i'$ by the addition or removal of a single token for all $1 \leq i < t$. 

To prove the opposite direction, assume $S = S'_1, \ldots, S'_t = T$ is a reconfiguration sequence from $S$ to $T$ in $G'$. We claim that this is also a reconfiguration sequence from $S$ to $T$ in $G$. All we have to show is that $S'_i$ is a dominating set of $G$ for all $1 \leq i \leq t$. This follows immediately from the fact that $S'_i$ is a dominating set of $G'$, and hence, as $W$ is not adjacent to $C \setminus \{u, v\}$ and $W \cap C = \emptyset$, also a dominating set of $C$ in $G$. Then according to Lemma 4.3 $S'_i$ also dominates $G$. We conclude that there is a reconfiguration sequence from $S$ to $T$ in $G'$ if and only if there is a reconfiguration sequence from $S$ to $T$ in $G = G - W$.

Proof of Lemma 4.15

Proof. We first show that we can essentially establish the situation depicted in Figure 9. We may assume that the paths of $P$ are induced paths, otherwise we may replace them by induced paths. Let $H$ be the graph induced on $u, v$ and the vertices of $P$ that contains exactly the edges of the paths in $P$. In the figure, the paths of $P$ are depicted by thick edges, while the diagonal edges do not belong to the paths. This situation is similar to the situation in the proof of Lemma 4.10. Just as in the proof of Lemma 4.10 we find two adjacent faces $f, g$ of $H$ that do not touch a vertex of $D \setminus \{u, v\}$.

![Figure 9](An exemplary situation handled by Lemma 4.15)

Claim 1. The paths bounding $f$ and $g$ have length 3, i.e., they have exactly two inner vertices.

Proof. First observe that $P \in P$ cannot have length exactly 2, as then $P$ contains a vertex adjacent to both $u$ and $v$. However, the vertices with this property lie in $D$, and hence by construction not on $P$.

Assume there is $P \in P$ of length greater than 3. Let $M(u)$ denote the neighbors of $u$ that are in $V(G) \setminus D$ and are only adjacent to $u$ and to no other vertex of $C$. Similarly, let $M(v)$ denote the neighbors of $v$ that are in $V(G) \setminus D$ and are only adjacent to $v$ and to no other vertex of $C$. By construction, the faces $f$ and $g$ do not contain vertices of $D \setminus \{u, v\}$. Furthermore, $P$ contains exactly one vertex of $M(u)$ and exactly one vertex of $M(v)$. It cannot contain two vertices of one of these sets, as otherwise $P$ is not an induced path. Hence, assume that $P$ contains another vertex $x$ that is not in $M(u) \cup M(v)$. Then $x$ must be dominated by a vertex different from $u$ and from $v$. However, by construction, the faces $f$ and $g$ do not touch a vertex of $D \setminus \{u, v\} \supseteq (S \cup T) \setminus \{u, v\}$, a contradiction.

Denote by $x_f, y_f$ the two vertices that lie on the boundary of $f$ and not on the boundary of $g$ and by $x_g, y_g$ the two vertices that lie on the boundary of $g$ and not on the boundary of $f$. Assume that $x_f, x_g \in M(u)$ and $y_f, y_g \in M(v)$. Denote by $z_u, z_v$ the vertices shared by $f$ and $g$ different from $u$ and $v$ that are adjacent to $u$ and $v$, respectively. Denote by $W$
the set of all vertices that lie inside the face \( h \) of the cycle \( u, x_f, y_f, v, y_g, x_g, u \) that contains the vertices \( z_u \) and \( z_v \). Hence \( W \) contains at least the vertices \( z_u \) and \( z_v \). By Corollary 4.11 we know that \( u, v \in S_i \), for all \( 1 \leq i \leq t \) (both \( u \) and \( v \) can never be lifted). Consequently, by Lemma 4.12 we know that there are no edges with both endpoints in \( N(v) \) nor edges with both endpoints in \( N(u) \). Combining the previous fact with the fact that all vertices of \( W \) are adjacent to either \( u \) or \( v \) (but not both) and to no other vertex of \( C \supseteq S \cup T \), we conclude that \( W \) consists of exactly the two vertices \( z_u \) and \( z_v \) and that there are no edges between \( z_u \) and \( x_g, x_f \) and no edges between \( z_v \) and \( y_g, y_f \). Note that we can safely assume that none of the degree-one neighbors of \( u \) or \( v \) are inside \( W \). We claim that the vertices \( z_u \) and \( z_v \) are irrelevant and can be removed after possibly introducing an additional edge to the graph. Recall that \( S \) and \( T \) do not contain the vertices \( z_u \) and \( z_v \). We define \( G' \) as follows.

\[
\text{If } \{u, v\} \notin E(G) \text{ and } \{x_f, z_v\} \notin E(G) \text{ or } \{y_f, z_u\} \notin E(G) \text{ and } \{x_g, z_e\} \notin E(G) \text{ or } \{y_g, z_u\} \notin E(G) \text{ then } G' \text{ is obtained from } G \text{ by deleting } z_u \text{ and } z_v \text{ and introducing the edge } \{x_f, y_g\}. 
\]

\[
\text{Otherwise, } G' \text{ is obtained from } G \text{ by simply deleting } z_u \text{ and } z_v.
\]

We claim that \((G, S, T, k)\) and \((G', S, T, k)\) are equivalent instances of \( \text{CDS-R} \). Assume first that there exists a reconfiguration sequence \( S = S_1, \ldots, S_t = T \) in \( G \). We distinguish two cases. First assume that \( \{u, v\} \in E(G) \). Hence, \( G' \) is obtained from \( G \) by simply deleting \( z_u \) and \( z_v \). Let \( S_i', \ldots, S_t' \), where for \( 1 \leq i \leq t \), \( S_i' = S_i \setminus \{u, v\} \). We claim that \( S_1', \ldots, S_t' \) is a reconfiguration sequence from \( S \) to \( T \) in \( G' \).

\section*{Claim 2.} For \( 1 \leq i \leq t \), \( S_i' \) is a dominating set of \( G \), and hence also of \( G' \).

\textbf{Proof.} The vertices \( z_u \) and \( z_v \) are not adjacent to a vertex of \( C \setminus \{u, v\} \) and \( \{z_u, z_v\} \cap C = \emptyset \). Hence, the only vertices of \( C \) that are possibly adjacent to \( z_u \) or \( z_v \) are the vertices \( u \) and \( v \). According to Lemma 4.11, \( u, v \in S_i \), and moreover, \( u, v \in S_i' \), for all \( 1 \leq i \leq t \). Hence, \( S_i' \) dominates at least the vertices of \( C \) that \( S_i \) dominates. We use Lemma 4.13 to conclude that \( S_i' \) is a dominating set of \( G \).

\section*{Claim 3.} For \( 1 \leq i \leq t \), \( S_i' \) is connected.

\textbf{Proof.} Let \( s_1, s_2 \in S_i \setminus \{z_u, z_v\} \) and let \( P \) be a shortest path between \( s_1 \) and \( s_2 \) in \( G[S_i] \). We have to show that there exists a path between \( s_1 \) and \( s_2 \) in \( G[S_i'] \). If \( P \) does not use \( z_u \) nor \( z_v \), then there is nothing to prove. Hence, assume \( P \) uses \( z_u \) or \( z_v \) (or both). By definition of \( W \), both \( s_1 \) and \( s_2 \) lie outside the face \( h \) of the cycle \( u, x_f, y_f, v, y_g, x_g, u \) that contains \( z_u, z_v \). Hence, \( P \) must enter and leave the face \( h \), say it enters at \( u \) and leaves at \( y_f \). All other possibilities are handled analogously. Then we can avoid the vertices \( z_u \) and \( z_v \) by walking to \( v \) first, then \( u \) (or \( x_f \)), and then to \( y_f \).

The next claim follows from the definition of \( S_i' \) and the fact that we can remove any duplicate consecutive sets in a reconfiguration sequence.

\section*{Claim 4.} \( S_{i+1}' \) is obtained from \( S_i' \) by the addition or removal of a single token for all \( 1 \leq i < t \).

This finishes the proof in case \( \{u, v\} \notin E(G) \). Hence, we assume now that \( \{u, v\} \notin E(G) \) and \( \{x_f, z_v\} \notin E(G) \) or \( \{y_f, z_u\} \notin E(G) \) and \( \{x_g, z_e\} \notin E(G) \) or \( \{y_g, z_u\} \notin E(G) \). That is, \( G' \) is obtained from \( G \) by deleting \( z_u \) and \( z_v \) and introducing the edge \( \{x_f, y_g\} \). We now obtain \( S_i' \) from \( S_i \), for \( 1 \leq i \leq t \), by replacing

\[
\text{by } x_f \text{ and } y_g \text{ if } S_i \cap \{z_u, z_v\} = \{z_u, z_v\},
\]
We let $X$ without loss of generality (the other case is symmetric), that vertex of domination and connectivity are preserved. This completes the proof of the lemma. \hfill ▶

A subsequence we obtain the required reconfiguration sequence in remaining a valid reconfiguration sequence in to show how to modify removing the token on vertex a reconfiguration sequence in again there is nothing to prove as is connected for all $G$. Now assume that there exists a reconfiguration sequence $y$ in $G$, we have introduced the edge $\{y\}$, $\{y\}$ is connected in $G$ and $y \in S_i$, or $\{y, y\} \in E(G)$ and $y \in S_i$. We move the token $y$ to $x$. In the first case we have connectivity via the new edge $\{x, y\} \in E(G)$, and in the second case we have connectivity via the edge $\{x, y\} \in E(G)$. The case $S_i \cap \{u, v\} = \{v\}$ is symmetric. \hfill ▶

This finishes the proof that if $(G, S, T, k)$ is a positive instance then $(G', S, T, k)$ is a positive instance. Now assume that there exists a reconfiguration sequence $S = S'_1, \ldots, S'_t = T$ in $G'$. In case we do not introduce the new edge to obtain $G'$ from $G$, we do not need new arguments to see that $S'_1, \ldots, S'_t$ is a reconfiguration sequence also in $G$. Moreover, if $G''[S'_i]$ is connected for all $i$, where $G''$ is obtained from $G'$ by removing the edge $\{x, y\}$, then again there is nothing to prove as $G'$ is a subgraph of $G$ and therefore $S = S'_1, \ldots, S'_t = T$ is a reconfiguration sequence in $G$. Hence, assume that there exists at least one contiguous subsequence $\sigma$ starting at index $s$ and ending at index $f$ (with possibly $s = f$) such that $G''[S'_i], G''[S'_{i+1}], \ldots, G''[S'_f]$ are not connected. In other words, there exists a subsequence of length one or more that uses the edge $\{x, y\}$ for connectivity. Moreover, we assume, without loss of generality (the other case is symmetric), that $S'_i$ is obtained from $S'_{i-1}$ by adding a token on vertex $y$, i.e., $S'_i = S'_{i-1} \cup \{y\}$, and $S'_{i+1}$ is obtained from $S'_i$ by removing the token on vertex $x$, i.e., $S'_{i+1} = S'_i \setminus \{x\}$. We also assume that $E(G)$ contains the edges $\{x, y\}$ and $\{u, v\}$ (the remaining cases are handled identically). It remains to show how to modify $\sigma$ so that it does not use the edge $\{x, y\}$ for connectivity and remains a valid reconfiguration sequence in $G$. By applying the same arguments for any such subsequence we obtain the required reconfiguration sequence in $G$. We modify $\sigma$ as follows. We let $S'' = (S'' \setminus \{y\}) \cup \{z\}$, for $s \leq i \leq f$. Then we replace $S''_{i+1}$ by four new sets $A_1, A_2, A_3$, and $A_4$, where $A_1 = S'' \setminus \{x\}$, $A_2 = A_1 \cup \{z\}$, $A_3 = A_2 \setminus \{z\}$, $A_3 = A_3 \cup \{y\}$, and $A_4 = A_3 \setminus \{z\}$. Using the fact that the vertices $x, y, x, y$ are not adjacent to vertices of $D \setminus \{u, v\}$, it is easy to see that this yields a valid reconfiguration sequence, as both domination and connectivity are preserved. This completes the proof of the lemma. \hfill ▶