

On Girth and the Parameterized Complexity of Token Sliding and Token Jumping

Valentin Bartier

Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, France
valentin.bartier@grenoble-inp.fr

Nicolas Bousquet 

CNRS, LIRIS, Université de Lyon, Université Claude Bernard Lyon 1, France
nicolas.bousquet@univ-lyon1.fr

Clément Dallard 

FAMNIT, University of Primorska, Koper, Slovenia
clement.dallard@famnit.upr.si

Kyle Lomer

Department of Computer Science, American University of Beirut, Lebanon
kjl00@mail.aub.edu

Amer E. Mouawad

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

Abstract

In the TOKEN JUMPING problem we are given a graph $G = (V, E)$ and two independent sets S and T of G , each of size $k \geq 1$. The goal is to determine whether there exists a sequence of k -sized independent sets in G , $\langle S_0, S_1, \dots, S_\ell \rangle$, such that for every i , $|S_i| = k$, S_i is an independent set, $S = S_0$, $S_\ell = T$, and $|S_i \Delta S_{i+1}| = 2$. In other words, if we view each independent set as a collection of tokens placed on a subset of the vertices of G , then the problem asks for a sequence of independent sets which transforms S to T by individual token jumps which maintain the independence of the sets. This problem is known to be PSPACE-complete on very restricted graph classes, e.g., planar bounded degree graphs and graphs of bounded bandwidth. A closely related problem is the TOKEN SLIDING problem, where instead of allowing a token to jump to any vertex of the graph we instead require that a token slides along an edge of the graph. TOKEN SLIDING is also known to be PSPACE-complete on the aforementioned graph classes. We investigate the parameterized complexity of both problems on several graph classes, focusing on the effect of excluding certain cycles from the input graph. In particular, we show that both TOKEN SLIDING and TOKEN JUMPING are fixed-parameter tractable on C_4 -free bipartite graphs when parameterized by k . For TOKEN JUMPING, we in fact show that the problem admits a polynomial kernel on $\{C_3, C_4\}$ -free graphs. In the case of TOKEN SLIDING, we also show that the problem admits a polynomial kernel on bipartite graphs of bounded degree. We believe both of these results to be of independent interest. We complement these positive results by showing that, for any constant $p \geq 4$, both problems are W[1]-hard on $\{C_4, \dots, C_p\}$ -free graphs and TOKEN SLIDING remains W[1]-hard even on bipartite graphs.

2012 ACM Subject Classification Mathematics of computing \rightarrow Paths and connectivity problems; Mathematics of computing \rightarrow Graph algorithms; Mathematics of computing \rightarrow Combinatoric problems

Keywords and phrases Combinatorial reconfiguration, Independent Set, Token Jumping, Token Sliding, Parameterized Complexity

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2020.44

Related Version A full version of the paper is available at [1], <https://arxiv.org/abs/2007.01673>.

Funding *Valentin Bartier*: This work is supported by ANR project GrR (ANR-18-CE40-0032).

Nicolas Bousquet: This work is supported by ANR project GrR (ANR-18-CE40-0032).



© Valentin Bartier, Nicolas Bousquet, Clément Dallard, Kyle Lomer, and Amer E. Mouawad; licensed under Creative Commons License CC-BY

31st International Symposium on Algorithms and Computation (ISAAC 2020).

Editors: Yixin Cao, Siu-Wing Cheng, and Minming Li; Article No. 44; pp. 44:1–44:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Clément Dallard: This work is supported in part by the Slovenian Research Agency (N1-0102).

Amer E. Mouawad: This work is supported by URB project “A theory of change through the lens of reconfiguration”

Acknowledgements The authors would like to thank the anonymous reviewer for his/her insightful comments that allowed us to improve Lemma 2 and Theorem 4.

1 Introduction

Many algorithmic questions present themselves in the following form: given the description of a system state and the description of a state we would “prefer” the system to be in, is it possible to transform the system from its current state into the more desired one without “breaking” the system in the process? Such questions, with some generalizations and specializations, have received a substantial amount of attention under the so-called *combinatorial reconfiguration framework* [7, 31, 33]. Historically, the study of reconfiguration questions predates the field of computer science, as many classic one-player games can be formulated as reachability questions [20, 23], 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 [8, 15, 16], constraint satisfaction [13, 28], computational geometry [27], and even quantum complexity theory [12]. We refer the reader to the surveys by van den Heuvel [31] and Nishimura [30] for more background on combinatorial reconfiguration.

Independent Set Reconfiguration. In this work, we focus on the reconfiguration of independent sets. Given a simple undirected graph G , a set of vertices $S \subseteq V(G)$ is an *independent set* if the vertices of this set are all pairwise non-adjacent. Finding an independent set of maximum cardinality, i.e., the INDEPENDENT SET problem, is a fundamental problem in algorithmic graph theory and is known to be not only NP-hard, but also W[1]-hard and not approximable within $\mathcal{O}(n^{1-\epsilon})$, for any $\epsilon > 0$, unless $P = NP$ [34]. Moreover, INDEPENDENT SET is known to remain W[1]-hard on graphs excluding C_4 (the cycle on four vertices) as an induced subgraph [4].

We view an independent set as a collection of tokens placed on the vertices of a graph such that no two tokens are adjacent. This gives rise to (at least) two natural adjacency relations between independent sets (or token configurations), also called *reconfiguration steps*. These two reconfiguration steps, in turn, give rise to two combinatorial reconfiguration problems. In the TOKEN JUMPING (TJ) problem, introduced by Kamiński et al. [22], a single reconfiguration step consists of first removing a token on some vertex u and then immediately adding it back on any other vertex v , as long as no two tokens become adjacent. The token is said to *jump* from vertex u to vertex v . In the TOKEN SLIDING (TS) problem, introduced by Hearn and Demaine [14], two independent sets are adjacent if one can be obtained from the other by a token jump from vertex u to vertex v with the additional requirement of uv being an edge of the graph. The token is then said to *slide* from vertex u to vertex v along the edge uv . Note that, in both the TJ and TS problems, the size of independent sets is fixed. Generally speaking, in the TOKEN JUMPING and TOKEN SLIDING problems, we are given a graph G and two independent sets S and T of G . The goal is to determine whether there exists a sequence of reconfiguration steps – a *reconfiguration sequence* – that transforms S into T (where the reconfiguration step depends on the problem).

Both problems have been extensively studied under the combinatorial reconfiguration framework, albeit under different names [3, 5, 10, 11, 18, 19, 22, 26, 29]. It is known that both problems are PSPACE-complete, even on restricted graph classes such as graphs of bounded

■ **Table 1** Parameterized complexity of TOKEN JUMPING and TOKEN SLIDING.

Graph Class	TOKEN JUMPING	TOKEN SLIDING
$\{C_3, C_4\}$ -free graphs	FPT (Section 3.1)	Open
C_4 -free graphs	W[1]-hard (Section 4.1)	W[1]-hard (Section 4.1)
Bipartite graphs	Open	W[1]-hard (Section 4.2)
Bipartite C_4 -free graphs	FPT (Section 3.1)	FPT (Section 3.3)

bandwidth (and then pathwidth) [32] and planar graphs [14]. In general TOKEN SLIDING is more complicated to decide than TOKEN JUMPING. However TOKEN SLIDING and TOKEN JUMPING can be decided in polynomial time on trees [10], interval graphs [3], bipartite permutation and bipartite distance-hereditary graphs [11] or line graphs [15]. Lokshtanov and Mouawad [25] showed that, in bipartite graphs, TOKEN JUMPING is NP-complete while TOKEN SLIDING remains PSPACE-complete. In split graphs, TOKEN JUMPING is a trivial problem while TOKEN SLIDING is PSPACE-complete [2]. In addition to the classes above, TOKEN JUMPING can be decided in polynomial time for even-hole-free graphs [21]. In this paper we focus on the parameterized complexity of the TOKEN JUMPING and TOKEN SLIDING problems on graphs where some cycles with prescribed length are forbidden. Given an NP-hard problem, parameterized complexity permits to refine the notion of hardness: does it come from the whole instance or from a small parameter? A problem Π is *FPT* (Fixed Parameterized Tractable) parameterized by k if one can solve it in time $f(k) \cdot \text{poly}(n)$. In other words, the combinatorial explosion can be restricted to a parameter k . In the rest of the paper, our parameter k will be the size of the independent set (i.e. number of tokens).

Both TOKEN JUMPING and TOKEN SLIDING are known to be $W[1]$ -hard¹ parameterized by k on general graphs [26]. On the positive side, Lokshtanov et al. showed [26] that TOKEN JUMPING is FPT on bounded degree graphs. TOKEN JUMPING is also known to be FPT on strongly $K_{\ell, \ell}$ -free graphs [17, 6], a graph being strongly $K_{\ell, \ell}$ -free if it does not contain any $K_{\ell, \ell}$ as a subgraph.

Our results. (For a complete overview of our results, see Table 1). In this paper, we focus on what happens if we consider graphs that do not admit a (finite or infinite) collection of cycles of prescribed lengths. Such graph classes contain bipartite graphs (odd-hole-free graphs), even-hole-free graphs and triangle-free graphs. Our main goal was to understand which cycles make the independent set reconfiguration problems hard. Our main technical result consists in showing that TOKEN SLIDING is $W[1]$ -hard parameterized by k on bipartite graphs with a reduction from MULTICOLORED INDEPENDENT SET. We were not able to adapt our reduction for TOKEN JUMPING and left it as an open question:

► **Question 1.** Is TOKEN JUMPING FPT parameterized by k on bipartite graphs?

On the positive side, we prove that TOKEN JUMPING admits a quadratic kernel (i.e. an equivalent instance of size $O(k^2)$ can be found in polynomial time) for $\{C_3, C_4\}$ -free graphs while it is $W[1]$ -hard if we restrict to $\{C_4, \dots, C_p\}$ -free graphs for a fixed constant p (the same hardness result also holds for TOKEN SLIDING). Note that the fact that the problem is FPT on graphs of girth² at least 5 graphs also follows from FPT algorithms for strongly $K_{3, \ell}$ -free graphs of [17], but even if a polynomial kernel can be derived from their result, the

¹ Informally, it means that they are very unlikely to admit an FPT algorithm.

² The *girth* of a graph is the length of its shortest cycle.

degree of our polynomial is better. We were not able to remove the C_4 condition in order to obtain a parameterized algorithm for triangle-free graphs. If an FPT algorithm exists for triangle-free graphs, it would, in particular, answer Question 1.

► **Question 2.** Is TOKEN JUMPING FPT parameterized by k on triangle-free graphs?

We then focus on TOKEN SLIDING. While FPT algorithms are (relatively) easy to design on sparse graphs for TOKEN JUMPING, they are much harder for TOKEN SLIDING. In particular, it is still open to determine if TOKEN SLIDING is FPT on planar graphs or H -minor free graphs while they follow for instance from [17, 6] for TOKEN JUMPING. Our main positive result is that TOKEN SLIDING on bipartite C_4 -free graphs (i.e. bipartite graphs of girth at least 6) admits a polynomial kernel. Our proof is in two parts, first we show that TOKEN SLIDING on bipartite graphs with bounded degree admits a polynomial kernel and then show that, if the graph admits a vertex of large enough degree then the answer is always positive. So TOKEN SLIDING is W[1]-hard on bipartite graphs but FPT on bipartite C_4 -free graphs. In our positive results, C_4 -freeness really plays an important role (neighborhoods of the neighbors of a vertex x are almost disjoint). It would be interesting to know if forbidding C_4 is really important or whether it is only helpful with our proof techniques. In particular, does TOKEN SLIDING admit an FPT algorithm on bipartite C_{2p} -free graphs for some $p \geq 3$? In our hardness reduction for bipartite graphs, all (even) cycles can appear and then such a result can hold. Recall that we prove that TOKEN JUMPING admits a polynomial kernel for graphs of girth at least 6. It would be interesting to see if our result on bipartite C_4 -free graphs can be extended to this class.

► **Question 3.** Is TOKEN SLIDING FPT parameterized by k on graphs of girth at least 5? Or, slightly weaker, is it FPT on graphs of girth at least p , for some constant p .

Note that the fact that the girth is at least 5 is needed since TOKEN SLIDING is W[1]-hard on bipartite graphs (which have girth at least 4). Let us finally briefly discuss some cases where we forbid an infinite number of cycles. We have already discussed the case where odd cycles are forbidden. One can wonder what happens if even cycles are forbidden. It is shown in [21] that TOKEN JUMPING can be decided in polynomial time for even-hole-free graphs (which is remarkable since computing a maximum independent set in this class is open). However, as far as we know, the complexity status of the problem is open for TOKEN SLIDING. More generally, one can wonder what happens when we forbid all the cycles of length $p \bmod q$ for every pair of integers p, q .

2 Preliminaries

We denote the set of natural numbers by \mathbb{N} . For $n \in \mathbb{N}$, we let $[n] = \{1, 2, \dots, n\}$.

Graphs. 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. The *open neighborhood* of a vertex v is denoted by $N_G(v) = \{u \mid uv \in E(G)\}$ and the *closed neighborhood* by $N_G[v] = N_G(v) \cup \{v\}$. For a set of vertices $Q \subseteq V(G)$, we define $N_G(Q) = \{v \notin Q \mid uv \in E(G), u \in Q\}$ and $N_G[Q] = N_G(Q) \cup Q$. The subgraph of G induced by Q is denoted by $G[Q]$, where $G[Q]$ has vertex set Q and edge set $\{uv \in E(G) \mid u, v \in Q\}$. We let $G - Q = G[V(G) \setminus Q]$.

A *walk* of length ℓ from v_0 to v_ℓ in G is a vertex sequence v_0, \dots, v_ℓ , such that for all $i \in \{0, \dots, \ell - 1\}$, $v_i v_{i+1} \in E(G)$. It is a *path* if all vertices are distinct. It is a *cycle* if $\ell \geq 3$, $v_0 = v_\ell$, and $v_0, \dots, v_{\ell-1}$ is a path. A path from vertex u to vertex v is also called a *uv -path*. For a pair of vertices u and v in $V(G)$, by $\text{dist}_G(u, v)$ we denote the

distance or length of a shortest uv -path in G (measured in number of edges and set to ∞ if u and v belong to different connected components). The *eccentricity* of a vertex $v \in V(G)$, $\text{ecc}(v)$, is equal to $\max_{u \in V(G)}(\text{dist}_G(u, v))$. The *radius* of G , $\text{rad}(G)$, is equal to $\min_{v \in V(G)}(\text{ecc}(v))$. The *diameter* of G , $\text{diam}(G)$, is equal to $\max_{v \in V(G)}(\text{ecc}(v))$. For $r \geq 0$, the r -*neighborhood* of a vertex $v \in V(G)$ is defined as $N_G^r[v] = \{u \mid \text{dist}_G(u, v) \leq r\}$. We write $B(v, r) = \{u \mid \text{dist}_G(u, v) \leq r\}$ and call it a *ball of radius r around v* ; for $S \subseteq V(G)$, $B(S, r) = \bigcup_{v \in S} B(v, r)$.

A graph G is *bipartite* if the vertex set of G can be partitioned into two disjoint sets L (the left part) and R (the right part), i.e. $V(G) = L \cup R$, where $G[L]$ and $G[R]$ are edgeless. Given two graphs G and H , we say that G is H -*free* if G does not contain H as an induced subgraph.

Reconfiguration. In the TOKEN JUMPING problem we are given a graph $G = (V, E)$ and two independent sets S and T of G , each of size $k \geq 1$. The goal is to determine whether there exists a sequence of k -sized independent sets in G , $\langle S_0, S_1, \dots, S_\ell \rangle$, such that $|S_i| = k$, S_i is an independent set ($\forall i$), $S = S_0$, $S_\ell = T$, and $|S_i \Delta S_{i+1}| = 2$. In other words, if we view each independent set as a collection of tokens placed on a subset of the vertices of G , then the problem asks for a sequence of independent sets which transforms S to T by individual token jumps which maintain the independence of the sets. For two independent sets S and T , we write $S \rightsquigarrow T$ in G if there exists a sequence of jumps that transforms S to T in G . For the closely related problem of TOKEN SLIDING, instead of allowing a token to jump to any vertex of the graph we instead require that a token slides along an edge of the graph. We use the same terminology for both problems as it will be clear from context which problem we are referring to. Note that both TOKEN JUMPING and TOKEN SLIDING can be expressed in terms of a *reconfiguration graph* $\mathcal{R}_Q(G, k)$, where $Q \in \{\text{TS}, \text{TJ}\}$. Both $\mathcal{R}_{\text{TJ}}(G, k)$ and $\mathcal{R}_{\text{TS}}(G, k)$ contain a node for each independent set of G of size exactly k . We add an edge between two nodes whenever the independent set corresponding to one node can be obtained from the other by a single reconfiguration step. That is, a single token jump corresponds to an edge in $\mathcal{R}_{\text{TJ}}(G, k)$ and a single token slide corresponds to an edge in $\mathcal{R}_{\text{TS}}(G, k)$. Given two nodes S and T in $\mathcal{R}_{\text{TJ}}(G, k)$ ($\mathcal{R}_{\text{TS}}(G, k)$), the TOKEN JUMPING problem (TOKEN SLIDING problem) asks whether S and T belong to the same connected component of $\mathcal{R}_{\text{TJ}}(G, k)$ ($\mathcal{R}_{\text{TS}}(G, k)$).

3 Positive results

3.1 TOKEN JUMPING on $\{C_3, C_4\}$ -free

We say that a class of graphs \mathcal{G}_ε is ε -*sparse*, for some $\varepsilon > 0$, if for every graph $G \in \mathcal{G}$ with n vertices, the number of edges in G is at most $n^{2-\varepsilon}$. By extension, G is said to be ε -sparse. Given an instance $\mathcal{I} = (G, S, T, k)$ of TOKEN JUMPING, let $H = G - N_G[S \cup T]$ and J denote the graph induced by $N_G[S \cup T]$. In the remainder of this section, we show that \mathcal{I} is a yes-instance whenever (at least) one of the following two conditions is true: (1) H is ε -sparse and contains more than $k(2k)^{1/\varepsilon}$ vertices or (2) J is $\{C_3, C_4\}$ -free and contains a vertex of degree at least $3k$.

► **Lemma 1.** *Let $\mathcal{I} = (G, S, T, k)$ be an instance of TOKEN JUMPING and let $H = G - N_G[S \cup T]$. If H is an ε -sparse graph with more than $k(2k)^{1/\varepsilon}$ vertices then \mathcal{I} is a yes-instance. Moreover, the length of the shortest reconfiguration sequence from S to T is at most $2k$.*

Proof. First, consider an ε -sparse graph H' with $n > (2k)^{1/\varepsilon}$ vertices. We claim that H' contains a vertex with degree less than $\frac{n}{k}$. Assume otherwise, i.e., suppose that the minimum degree in H' is at least $\frac{n}{k}$. Then, $|E(H')| \geq \frac{n^2}{2k}$. Moreover, since H' is ε -sparse, it holds that $|E(H')| \leq n^{2-\varepsilon}$. However, $\frac{n^2}{2k} \leq n^{2-\varepsilon}$ if and only if $n \leq (2k)^{1/\varepsilon}$, a contradiction.

Now, we shall prove, by induction on k , that H contains an independent set of size at least k . The statement holds for $k = 1$ (since H must contain at least one vertex). Now, consider the case where $k > 1$ and let z be a vertex with minimum degree in H . Following the above claim, z has degree less than $\frac{n}{k}$. Note that the graph $H' = H - N[z]$ contains at least $(k-1)\frac{n}{k} \geq (k-1)\frac{k(2k)^{1/\varepsilon}}{k} = (k-1)(2k)^{1/\varepsilon}$ vertices. By the induction hypothesis, H' contains an independent set X of size at least $k-1$. Thus, $X \cup \{z\}$ is an independent set in H of size at least k .

Hence, we can transform S to T by simply jumping all the tokens in S to an independent set $X \subseteq V(G) \setminus (S \cup T)$ and then from X we jump the tokens (one by one) to T . This completes the proof. \blacktriangleleft

► **Lemma 2.** *Let $\mathcal{I} = (G, S, T, k)$ be an instance of TOKEN JUMPING and let J denote the graph induced by $N_G[S \cup T]$. If J is $\{C_3, C_4\}$ -free and contains a vertex v of degree at least $3k$, then \mathcal{I} is a yes-instance. Moreover, the length of the shortest reconfiguration sequence from S to T is at most $2k$.*

Proof. Fix $w \in S \cup T$. First, observe that for any $u \in N(S \cup T)$, u is adjacent to w and no neighbor of w (otherwise J would contain a C_3), or u is adjacent to at most one neighbor of w (otherwise J would contain a C_4). Therefore, every vertex in $N(S \cup T)$ has degree at most $2k$ in J . As J is C_3 -free, $N_J(w)$ is an independent set. Furthermore, for any $u, v \in N_J(w)$, $u \neq v$, we have $N_J(u) \cap N_J(v) = \{w\}$, that is, w is the only common neighbor of u and v in J ; otherwise, J would contain C_4 . Hence, if w has at least $3k$ neighbors, then at least k of them only have w as a neighbor in $S \cup T$. Thus, we can jump the tokens on S to $N(w)$, starting with the token on w , if any. Then, we can jump the tokens on the vertices in T . Clearly, the length of such a reconfiguration sequence is at most $2k$. \blacktriangleleft

► **Proposition 3.** *Let $\mathcal{I} = (G, S, T, k)$ be an instance of TOKEN JUMPING, let $H = G - N_G[S \cup T]$, and let J denote the graph induced by $N_G[S \cup T]$. If H is ε -sparse, $\varepsilon > 0$, and J is $\{C_3, C_4\}$ -free then \mathcal{I} admits a kernel with $\mathcal{O}(k^2 + k^{1+1/\varepsilon})$ vertices.*

Proof. If H contains more than $k(2k)^{1/\varepsilon}$ vertices then \mathcal{I} is a yes-instance by Lemma 1. If J contains a vertex of degree $3k$ or more then, again, \mathcal{I} is a yes-instance by Lemma 2. Putting it all together, we have $|S \cup T| \leq 2k$, $|N_G(S \cup T)| \leq 2k(3k-1) = \mathcal{O}(k^2)$, and $|V(G) \setminus N_G[S \cup T]| \leq k(2k)^{1/\varepsilon} = \mathcal{O}(k^{1+1/\varepsilon})$. \blacktriangleleft

► **Theorem 4.** *TOKEN JUMPING parameterized by k admits a kernel with at most $\mathcal{O}(k^2)$ vertices on $\{C_3, C_4\}$ -free graphs (and then on bipartite C_4 -free graphs).*

Proof. Let $\mathcal{I} = (G, S, T, k)$ be an instance of TOKEN JUMPING such that G is $\{C_3, C_4\}$ -free. Let $H = G - N_G[S \cup T]$ and J denote the graph induced by $N_G[S \cup T]$. Since J is $\{C_3, C_4\}$ -free, Lemma 2 implies that if J contains more than $6k^2 - 2k$ vertices, then \mathcal{I} is a yes-instance. Kim showed that a C_3 -free graph with $\mathcal{O}(k^2/\log k)$ vertices contains an independent set of size at least k [24]. Hence, if H contains more than $\mathcal{O}(k^2/\log k)$ vertices, then \mathcal{I} is a yes-instance. Thus, G contains at most $\mathcal{O}(k^2)$ vertices. The same result holds for bipartite C_4 -free graphs since they are $\{C_3, C_4\}$ -free. \blacktriangleleft

3.2 TOKEN SLIDING on bounded-degree bipartite graphs

Unlike the case of TOKEN JUMPING, it is not known whether TOKEN SLIDING is fixed-parameter tractable (parameterized by k) on graphs of bounded degree. In this section we show that it is indeed the case for bounded-degree bipartite graphs. This result, interesting in its own right, will be crucial for proving that TOKEN SLIDING is fixed-parameter tractable on bipartite C_4 -free graphs in the next section. We start with a few definitions and needed lemmas.

Let $R(G, I) = \{v \mid v \in \bigcap_{I' \rightsquigarrow I} I'\}$ be the subset of I containing all of the tokens v such that $v \in I'$ for all I' reachable from I . In other words, the tokens on vertices of $R(G, I)$ can never move in any reconfiguration sequence starting from I . We call vertices in $R(G, I)$ *rigid with respect to G and I* . An independent set I is said to be *unlocked* if $R(G, I) = \emptyset$. Given a graph G and $r \geq 1$, a set $S \subseteq V(G)$ is called an *r -independent set*, or *r -independent* for short, if $B(v, r) \cap S = \{v\}$, for all $v \in S$. Note that a 1-independent set is a standard independent set and a r -independent set, $r > 1$, is a set where the shortest path between any two vertices of the set contains at least r vertices (excluding the endpoints).

For a vertex $v \in V(G)$ and a set $S \subseteq V(G) \setminus \{v\}$, we let $D(v, S)$ denote the set of vertices in S that are closest to v . That is, $D(v, S)$ is the set of vertices in S whose distance to v is minimum. We say $D(v, S)$ is *frozen* if $|D(v, S)| \geq 2$ and it is not possible to slide a single token in $D(v, S)$ to obtain S' such that either $v \in S'$ or $|D(v, S')| = 1$. Note that, in time polynomial in $n = |V(G)|$, it can be verified whether $D(v, S)$ is frozen by simply checking, for each vertex $u \in D(v, S)$, whether u can slide to a vertex w which is closer to v (or to v itself if u is adjacent to v).

► **Lemma 5** ([11]). *$S \rightsquigarrow T$ in G if and only if $R(G, S) = R(G, T)$ and $(S \setminus R(G, S)) \rightsquigarrow (T \setminus R(G, S))$ in $G - N[R(G, S)]$. Moreover, if G is bipartite then $R(G, S)$ and $R(G, T)$ can be computed in time linear in $|V(G)| = n$.*

► **Lemma 6** ([11]). *Let $G = (L \cup R, E)$ be a bipartite graph and let S be an unlocked independent set of G . Then, in time linear in n , we can compute a reconfiguration sequence $\langle S = I_0, I_1, \dots, I_\ell \rangle$ where $I_\ell \cap L = \emptyset$ and $\ell = |S \cap L|$.*

Let $\mathcal{I} = (G = (V, E), S, T, k)$ be an instance of TOKEN SLIDING where G is a bipartite graph of bounded degree Δ . We assume, without loss of generality, that G is connected; as otherwise we can solve the problem independently on each component of G (and there are at most k components containing tokens). Moreover, given Lemma 5, we can assume, without loss of generality, that S and T are unlocked. In other words, we assume that it has been verified that $R(G, S) = R(G, T)$ and $N[R(G, S)]$ has been deleted from G . We now give a slightly different version of a result in [11] better suited for our needs.

► **Lemma 7** ([11]). *Let G be a connected bipartite graph and let S be an unlocked independent set of G . Let v be a vertex in $V(G) \setminus S$ such that $N_G[v] \cap S = \emptyset$. Let $D(v, S) \subseteq L$ (or symmetrically $D(v, S) \subseteq R$) such that $\text{dist}_G(u, v) = d$, for all $u \in D(v, S)$. Then, in time linear in $|V(G)| = n$, we can find a reconfiguration sequence $\langle S = I_0, I_1, \dots, I_\ell \rangle$, where $I_\ell = (S \setminus \{u\}) \cup \{v\}$ for some vertex u in $D(v, S)$ and ℓ is at most $2(|S| - 1) + d$.*

Proof. There are two cases to consider:

(1) If there is a unique token $u \in D(v, S)$ which is closest to v or $D(v, S)$ is not frozen then the reconfiguration sequence obtained by repeatedly moving the token on u to a vertex which is closer to v gives us the required sequence. Since no other token is moved, we have $I_\ell = (S \setminus \{u\}) \cup \{v\}$.

(2) In the other case, we have $D(v, S) \geq 2$ and $D(v, S)$ is frozen. We assume, without loss of generality, that $D(v, S) \subseteq L$. We apply Lemma 6 which returns a reconfiguration sequence $\langle S = I_0, I_1, \dots, I_\ell \rangle$ where $I_\ell \cap L = \emptyset$ and $\ell = |S \cap L|$. There exists an index j , with $j < \ell < |S \cap L|$, where I_j has a unique token $u \in D(v, S)$ which is closest to v . Let $\alpha = \langle I_0, I_1, \dots, I_j \rangle$. Note that α slides exactly j distinct tokens (not including u) from L to R . We let M_α denote these tokens. Moreover, α is reversible. Hence, we let α^{-1} denote the sequence consisting of applying the slides of α in reverse order. Now, we construct a sequence β of slides that moves the token on u to v . Recall that this is a sequence of exactly d slides that repeatedly slides the same token. We denote the resulting independent set (after applying $\alpha \cdot \beta$) by I_β . We claim that $\gamma = \alpha \cdot \beta \cdot \alpha^{-1}$ is the required sequence that transforms S to $(S \setminus \{u\}) \cup \{v\}$. To see why γ is a valid reconfiguration sequence, it suffices to show that $N_G[M_\alpha] \cap N_G[v] = \emptyset$. Since $N_G[v] \cap S = \emptyset$, we know that $d \geq 2$ if both v and $D(v, S)$ are contained in L (or R) and $d \geq 3$ otherwise. If $\{v\}, D(v, S) \subseteq L$ (or $\{v\}, D(v, S) \subseteq R$) then every vertex in M_α is at distance at least three from v , as needed. Finally, if $v \in L$ and $D(v, S) \subseteq R$ (or $v \in R$ and $D(v, S) \subseteq L$) then every vertex in M_α is at distance at least four from v . \blacktriangleleft

► **Lemma 8.** *If G is a connected graph and S and T are any two 2-independent sets of G such that $S \cup T$ is also 2-independent then $S \leftrightarrow T$ in G .*

Proof. We proceed by induction on $|S\Delta T| = |(S \setminus T) \cup (T \setminus S)|$, i.e., the size of the symmetric difference between S and T . If $|S\Delta T| = 0$ then $S = T$ and there is nothing to prove. Hence, we assume that the statement is true for $|S\Delta T| = q > 0$. We compute a shortest path between all pairs of vertices (u, v) in G , where $u \in S \setminus T$ and $v \in T \setminus S$. We let (u, v) denote a pair where the distance is minimized and we fix a shortest path between u and v . There are two cases to consider:

(1) If $S \cap T = \emptyset$ then we can simply slide u to v along the shortest path and we are done. To see why, recall that both S and T are 2-independent. Hence, they are both unlocked and if there is more than one vertex in $S \setminus T$ that is closest to v then we can simply slide u into one of its neighbors, say w , that is closer to v to obtain a unique vertex which is closest to v ; none of those neighbors are adjacent to a vertex in S since S is 2-independent. Now, assume that there exists a vertex x along the shortest path from w to v such that $x \in N(y)$, $y \in S$. This contradicts the choice of u since y is closer to v .

(2) If $S \cap T \neq \emptyset$ then there are two cases. When the shortest path from u to v does not contain any vertex in $N_G[S \cap T]$ then we apply the same reasoning as above. Otherwise, let $W = w_1, w_2, \dots, w_q$ denote the vertices in $N_G[S \cap T]$ along the shortest path from u to v (sorted in the order in which they are visited). We divide W into three sets $X = W \cap (S \cap T)$, $Y = W \cap (N_G(X))$, and $Z = W \setminus (X \cup Y)$. In other words, X denotes the set of vertices in $S \cap T$, Y denotes the vertices used as entry and exit points for the vertices in X , and Z denotes the vertices in $N_G(S \cap T)$ visited along the shortest path without passing through a vertex $N_G(Z) \cap (S \cap T)$. Since $S \cap T$ is 2-independent, no vertex in $Y \cup Z$ can have two neighbors in $S \cap T$. Moreover, since we have a shortest path from u to v , if there exists $x \in X$ then $N_G(x) \cap Z = \emptyset$. In particular, the shortest path either visits a vertex $x \in S \cap T$ and two of its neighbors or only visits at most three neighbors of x ; as otherwise we can find a shorter path from u to v . If the shortest path visits three neighbors w, y , and z , of a vertex $x \in S \cap T$ then we can safely replace this sub-path by w, x, z . Hence, we assume in what follows that the shortest path visits at most two neighbors of any vertex in $S \cap T$. We construct, from W , the sequence $A = a_1, a_2, \dots, a_p$ of “affected” vertices in $S \cap T$. In other words, if the shortest path from u to v visits a vertex in $S \cap T$ or visits one or two

of its neighbors then we add the vertex to A (in the order in which the visits occur). We now proceed iteratively as follows. We slide a_p to v , then a_{p-1} to a_p , \dots , and then finally we slide u to a_1 . Note that between every one of those pairs of vertices we have a shortest path; since we are sliding along the shortest path from u to v . Moreover, after moving each token to its target position, we maintain a 2-independent set S' . Therefore, for each such shortest path the intersection with $N_G[S']$ remains empty. \blacktriangleleft

Let G be a graph and let $X \subseteq V(G)$. The *interior* of X is the set of vertices in X at distance at least three from $V(G) \setminus X$ (separated by at least two vertices). We say a set X is *fat* if its interior is connected and contains a 2-independent set of size at least $2k$.

► **Lemma 9.** *Let G be a graph of maximum degree Δ . Let $v \in V(G)$ and $r \in \mathbb{N}$. If $B(v, r)$ contains more than $2k(1 + \Delta + \Delta^2)^2$ vertices then $B(v, r)$ is fat.*

Proof. We only need to prove that the interior of $B(v, r)$, that is $B(v, r - 2)$, contains a 2-independent set of size at least $2k$; as $B(v, r - 2)$ is connected by construction. First, note that any graph of maximum degree Δ on more than $2k(1 + \Delta + \Delta^2)$ vertices must contain a 2-independent set of size at least $2k$. So it suffices to show that $B(v, r - 2)$ contains more than $2k(1 + \Delta + \Delta^2)$ vertices. We divide $B(v, r)$ into layers, where $L_0 = \{v\}$, $L_1 = N(v)$, \dots , and $L_r = N^r(v)$. Since G has maximum degree Δ , for every $i \geq 1$, layer L_i contains at most $(\Delta - 1)^{i-1} \Delta$ vertices. If $B(v, r - 2)$ contains more than $2k(1 + \Delta + \Delta^2)$ vertices then we are done. Otherwise, L_{r-2} must contain at most $2k(1 + \Delta + \Delta^2)$ vertices. Consequently, $L_{r-1} \cup L_r$ would contain at most $2k\Delta(1 + \Delta + \Delta^2) + 2k\Delta^2(1 + \Delta + \Delta^2) = (1 + \Delta + \Delta^2)(2k\Delta + 2k\Delta^2)$ vertices. Therefore, $B(v, r)$ contains at most $2k(1 + \Delta + \Delta^2) + (1 + \Delta + \Delta^2)(2k\Delta + 2k\Delta^2) = (1 + \Delta + \Delta^2)(2k + 2k\Delta + 2k\Delta^2)$ which is equal to $2k(1 + \Delta + \Delta^2)^2$ vertices, a contradiction. \blacktriangleleft

► **Lemma 10.** *Let $\mathcal{I} = (G, S, T, k)$ be an instance of TOKEN SLIDING where G is a bounded-degree bipartite graph. If $V(G) \setminus (S \cup T)$ contains a fat set X then \mathcal{I} is a yes-instance.*

Proof. First, recall that we assume that G is connected and both S and T are unlocked. Let I be a 2-independent set of size $2k$ in the interior of X (at distance at least three from any vertex outside of X). We prove that S can be transformed into $S' \subset I$. Similar arguments hold for transforming T into $T' \subset I$. Hence, the statement of the theorem follows by applying Lemma 8 on S' and T' .

We proceed by induction on $|S \Delta S'|$, i.e., the size symmetric difference between S and S' . If $|S \Delta S'| = 0$ then $S = S'$ and we are done. Otherwise, we reduce the size of the symmetric difference as follows. Recall that initially $S \cap S' = \emptyset$; as $X \subseteq V(G) \setminus (S \cup T)$. However, the size of the intersection will increase as more tokens are moved to S' . We pick a pair (u, v) such that $u \in S \setminus S'$ and $v \in S'$ and the distance between u and v is minimized. There are two cases 2 consider:

- (1) If v does not contain a token (or $v \in S' \setminus S$) then the shortest path from u to v does not intersect with $N_G[S' \cap S]$. We therefore invoke Lemma 7 in the graph $G - (N[S' \cap S])$. This guarantees that the token on u slides to v and every other token remains in place.
- (2) Otherwise, v already contains a token (or $v \in S' \cap S$). We invoke Lemma 8 on the graph induced by the interior of X and transform $C = S' \cap S \subset I$ into another 2-independent set $C' \subseteq I$ that does not contain v ; this is possible since $|C| = |C'| \leq k$. Now we can again invoke Lemma 7 similarly to the previous case. \blacktriangleleft

► **Theorem 11.** *TOKEN SLIDING parameterized by k admits a kernel with $\mathcal{O}(k^2 \Delta^5)$ vertices on bounded-degree bipartite graphs. Moreover, the problem can be solved in $\mathcal{O}^*(k^{2k} \Delta^{5k})$ -time.*

Proof. Let $\mathcal{I} = (G, S, T, k)$ be an instance of Token Sliding where G is a bipartite graph of maximum degree Δ . We assume, without loss of generality, that G is connected and S and T are unlocked; for otherwise we can solve connected components independently and we can return a trivial no-instance if $R(G, S) \neq R(G, T)$ (Lemma 5). Next, from Lemmas 9 and 10, we know that each connected component of $V(G) \setminus (S \cup T)$ contains at most $\mathcal{O}(k\Delta^4)$ vertices; otherwise we can return a trivial yes-instance. Since the number of components in $V(G) \setminus (S \cup T)$ is bounded by $2k\Delta$ and $|S \cup T| \leq 2k$, we get the desired bound. To solve the problem, it suffices to construct the complete reconfiguration graph and verify if S and T belong to the same connected component. This concludes the proof. \blacktriangleleft

3.3 TOKEN SLIDING on bipartite C_4 -free graphs

Equipped with Theorem 11, we are now ready to prove that TOKEN SLIDING admits a polynomial kernel on bipartite C_4 -free graphs. Our strategy will be simple. We show that if the graph contains a vertex of large degree then we have a yes-instance. Otherwise, we invoke Theorem 11 to obtain the required kernel.

We start with a few simplifying assumptions. Let $\mathcal{I} = (G, S, T, k)$ be an instance of TOKEN SLIDING where $G = (L \cup R, E)$ is a connected bipartite C_4 -free graphs. We assume that both S and T are unlocked (Lemma 5). Moreover, we assume that each vertex in G can have at most one pendant neighbor. This assumption is safe because no two tokens can occupy two pendant neighbors of a vertex; as otherwise S or T would be locked. Moreover, if a token is placed on a pendant neighbor of a vertex v then no other token can reach v .

Let $v \in V(G)$ be a vertex of degree at least $k^2 + k + 1$ in G . We let u_p denote the pendant neighbor of v (if it exists). We assume, without loss of generality, that $v \in L$. We let $N_1 = N_G(v) \setminus \{u_p\} = \{u_1, u_2, \dots, u_q\}$, $N_2 = N_G^2(v)$, and $N_3 = N_G^3(v)$. Since G is bipartite, $N_1 \subseteq R$, $N_2 \subseteq L$, and $N_3 \subseteq R$. Moreover, since G is C_4 -free, each vertex in N_2 has exactly one neighbor in N_1 . Therefore, we partition N_2 into sets $N_{u_1}, N_{u_2}, \dots, N_{u_q}$, where each set N_{u_i} contains the neighbors of u_i in N_2 , that is, $N(u_i) \setminus \{v\}$. We also partition N_3 into two sets M_{small} and M_{big} . Each vertex in M_{big} contains vertices connected to at least $k + 1$ sets in N_2 . Note that, because of C_4 -freeness, each vertex in N_3 is connected to at most one vertex of any set N_{u_i} . We let $M_{\text{small}} = N_3 \setminus M_{\text{big}}$. Each vertex in M_{small} has at most k neighbors in N_2 . In other words, each vertex in M_{small} is connected to at most k sets, each one of those sets being the neighborhood of a vertex in N_1 .

The proof proceeds in five stages. We first show how to transform S to S_1 such that $S_1 \cap B(v, 3) \subseteq N_2$. In other words, we can guarantee that all tokens in the ball of radius three around v are contained in N_2 . We then transform S_1 to S_2 such that $S_2 \cap B(v, 3) \subseteq N_1 \cup N_3$. Next, we transform S_2 to S_3 such that $S_3 \cap B(v, 3) \subseteq N_1 \cup M_{\text{small}}$. Then, we transform S_3 to S_4 such that $S_4 \cap B(v, 3) \subseteq N_1$ and finally to S_5 such that $S_5 \subseteq N_1$. By applying the same strategy starting from T , we obtain $T_5 \subseteq N_1$. We conclude our proof by showing that S_5 can be transformed to T_5 .

► **Theorem 12.** TOKEN SLIDING parameterized by k admits a kernel with $\mathcal{O}(k^{12})$ vertices on bipartite C_4 -free graphs.

4 Hardness results

4.1 TOKEN SLIDING and TOKEN JUMPING on C_4 -free graphs

In the GRID TILING problem we are given an integer $k \geq 0$ and k^2 sets $S_{i,j} \subseteq [m] \times [m]$, for $0 \leq i, j \leq k - 1$, of cardinality n called *tiles* and we are asked whether it is possible to find an element $s_{i,j}^* \in S_{i,j}$ for every $0 \leq i, j \leq k - 1$ such that $s_{i,j}^*$ and $s_{i,j+1}^*$ share the same first

coordinate while $s_{i,j}^*$ and $s_{i+1,j}^*$ share the same second coordinate for each $0 \leq i, j \leq k-1$ (including modulo k). It was proven in [9] that GRID TILING parameterized by k is $W[1]$ -hard. We prove the next theorem via a reduction from GRID TILING. Following the construction in [4] to give a graph G with the desired properties and extending it to a $\{C_4, \dots, C_p\}$ -free graph G' which gives a reduction to TOKEN SLIDING.

► **Theorem 13.** *For any $p \geq 4$, TOKEN SLIDING is $W[1]$ -hard on $\{C_4, \dots, C_p\}$ -free graphs.*

Construction of G . Given an instance of GRID TILING, $S_{i,j} \subseteq [m] \times [m]$ ($0 \leq i, j \leq k-1$) and an integer $p \geq 4$, we use the construction described in [4] to create a graph G with the following properties:

- **P1** – G can be partitioned into $8k^2(p+1)$ cliques $V_1, \dots, V_{8k^2(p+1)}$ of size n with some edges between them.
- **P2** – G is $\{C_4, \dots, C_p\}$ -free.
- **P3** – The instance of GRID TILING has a solution if and only if $\exists I \subseteq V(G)$, such that I is an independent set of size $8k^2(p+1)$.

Note that as each V_i is a clique, any maximum independent set I of G can have at most one vertex in every clique.

Construction of G' . For $k' = 8k^2(p+1)$, we construct an instance of TOKEN SLIDING $(G', S, T, k' + (3k' + 1)\frac{p}{2} + \frac{p}{2})$ by extending the graph G to a new graph G' . We label the k' cliques in G arbitrarily as $V_1, \dots, V_{k'}$. For each $1 \leq i \leq k'$ we add two vertices x_i and y_i adjacent to all vertices in V_i . These will respectively be starting and ending positions of tokens. Informally, we want to force all the tokens to be in their respective V_i at the same time to obtain an independent set in G of size k' . We do this by creating *guard paths*, which are paths on p vertices that will be alternating between starting and target positions of tokens. Note that we can assume p is even, since if p is odd we can use $p+1$ instead to create a graph which is $\{C_4, \dots, C_p\}$ -free. Let P_G be a guard path with vertices g_1, \dots, g_p and for each x_i let P_{x_i} be a guard path with vertices x_{i1}, \dots, x_{ip} such that x_i is adjacent to x_{ip} and g_p is adjacent to x_{i1} . For each y_i let P_{y_i} be a guard path with vertices y_{i1}, \dots, y_{ip} such that y_i is adjacent to y_{i1} and g_1 is adjacent to y_{ip} . Finally, for each i let P_{z_i} be a guard path between x_i and y_i with vertices z_{i1}, \dots, z_{ip} such that x_i is adjacent to z_{ip} and y_i is adjacent to z_{i1} . This completes the construction of G' (see Figure 1). The source independent set S is the set containing all of the x_i and all of the guard path vertices with odd indices:

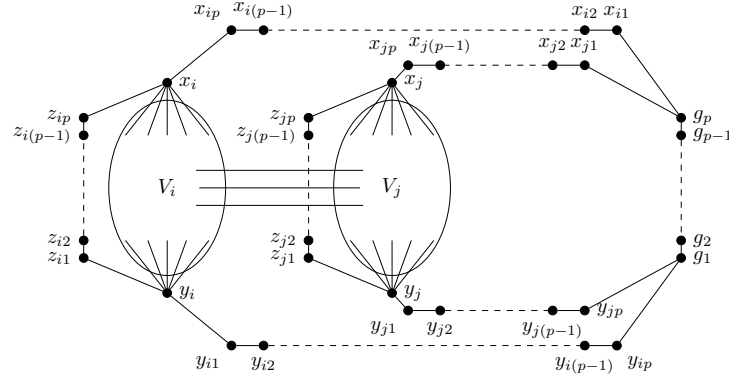
$$S = \bigcup_i (\{x_i\} \cup \{x_{ij} \mid j \text{ is odd}\} \cup \{y_{ij} \mid j \text{ is odd}\} \cup \{z_{ij} \mid j \text{ is odd}\}) \cup \{g_j \mid j \text{ is odd}\}.$$

The target independent set T consists of all of the y_i and all of the guard path vertices with even indices:

$$T = \bigcup_i (\{y_i\} \cup \{x_{ij} \mid j \text{ is even}\} \cup \{y_{ij} \mid j \text{ is even}\} \cup \{z_{ij} \mid j \text{ is even}\}) \cup \{g_j \mid j \text{ is even}\}.$$

► **Lemma 14.** *For any $p \geq 4$, G' is $\{C_4, \dots, C_p\}$ -free.*

► **Lemma 15.** *If there is a solution to the GRID TILING instance then there is a reconfiguration sequence from S to T in G' .*



■ **Figure 1** The construction of G' for two cliques V_i, V_j in G .

Proof. By **P3**, there exists an independent set I containing one vertex v_i in every V_i . This gives the following reconfiguration sequence from S to T . Move each token on x_i to v_i . Then, move the tokens along the guard paths: for all odd j starting with the greatest j values move the token on each z_{ij} to $z_{i(j+1)}$, then move the tokens on x_{ij} to $x_{i(j+1)}$, g_j to g_{j+1} , and y_{ij} to $y_{i(j+1)}$. Finally, move each token on v_i to y_i . This completes the proof. ◀

► **Theorem 16.** For any $p \geq 4$, TOKEN JUMPING and TOKEN SLIDING are $W[1]$ -hard on $\{C_4, \dots, C_p\}$ -free graphs.

4.2 TOKEN SLIDING on bipartite graphs

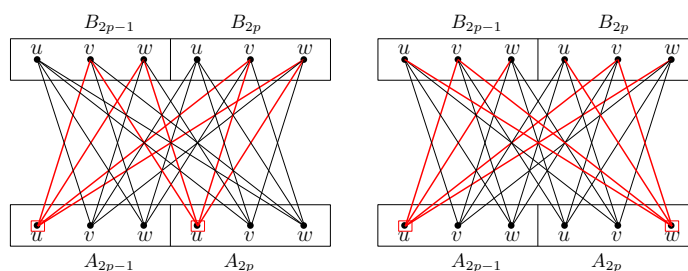
This section is devoted to proving the following theorem:

► **Theorem 17.** TOKEN SLIDING on bipartite graphs is $W[1]$ -hard parameterized by k .

The proof of Theorem 17 consists in a reduction from MULTICOLORED INDEPENDENT SET. In what follows, $\mathcal{I} := (G, k, (V_1, \dots, V_k))$ denotes an instance of MULTICOLORED INDEPENDENT SET, which is known to be $W[1]$ -hard parameterized by k [9]. We first detail the construction of the equivalent instance $\mathcal{I}' := (G', I_s, I_e, 4k + 2)$ of TOKEN SLIDING, where G' is a bipartite graph and I_s, I_e are independent sets of size $4k + 2$, and we prove that if \mathcal{I} is a yes-instance, then \mathcal{I}' is a yes-instance. The more involved proof of the converse direction is detailed in the full version of the paper [1].

Construction of G' . In what follows, $V(G') := (\mathcal{A}, \mathcal{B})$ denotes the bipartition of G' . For every $p \in \{1, \dots, k\}$, both \mathcal{A} and \mathcal{B} contain two copies of the set V_p denoted as A_{2p-1}, A_{2p} and B_{2p-1}, B_{2p} respectively, plus some additional vertices that will be described in the next subsection. Two vertices $u', v' \in V(G')$ are said to be *equivalent* and we write $u' \sim v'$ if and only if they are copies of the same vertex in G . With this definition, every vertex $u \in V_p$ has exactly four copies in G' (one in each copy of V_p). Note that the \sim relation is transitive and symmetric. We also define the sets $A := \cup_{p=1}^k A_{2p-1} \cup A_{2p}$ and $B := \cup_{p=1}^k B_{2p-1} \cup B_{2p}$. For every vertex u' of $A \cup B$, the *corresponding vertex* of u' denoted as $orr(u')$ is the unique vertex $u \in V(G)$ that u' is a copy of. With these definitions at hand, we can now explain how the copies of the sets V_1, V_2, \dots, V_k are connected in G' . For every two vertices $u' \in A_i$ and $v' \in B_j$ there is an edge connecting u' to v' in G' if and only if:

1. A_i and B_j are not copies of the same subset of $V(G)$ and $(orr(u'), orr(v')) \in E(G)$, or
2. A_i and B_j are copies of the same subset of $V(G)$ and $u' \approx v'$



■ **Figure 2** Connections between the four copies of V_p in $A \cup B$. Vertices with the same name are equivalent vertices. The red square represent tokens: two tokens are positioned on equivalent vertices at the left, and on non-equivalent vertices at the right.

In other words, if A_i and B_j are not copies of the same subset, we connect these sets in the same way there corresponding sets are connected in G . If at the contrary A_i and B_j are copies of the same subset, then $G'[A_i \cup B_j]$ induces a complete bipartite graph minus the matching consisting of every two pairs of equivalent vertices in $A_i \cup B_j$. The connection between four copies of the same subset of $V(G)$ is illustrated in Figure 2. The following observation follows directly from the definition of G' :

► **Observation 1.** Let I' be an independent set of G' such that for every $p \in \{1, 2, \dots, k\}$ we have $I' \cap A_{2p-1} = \{u_{2p-1}\}$ and $I' \cap B_{2p-1} = \{v_{2p-1}\}$. Then the set $I := \{orr(u_1), \dots, orr(u_k)\}$ is a multicolored independent set of G .

Observation 1 ensures that any independent set of a reconfiguration sequence of G' having exactly one vertex in A_{2p-1} and one vertex in B_{2p-1} for every $p \in \{1, 2, \dots, k\}$ corresponds to a multicolored independent set of G . Note that up to that point, we did not make use of the sets A_{2p} and B_{2p} . The following observation explains why we need two copies of every V_p in both sides of the bipartition:

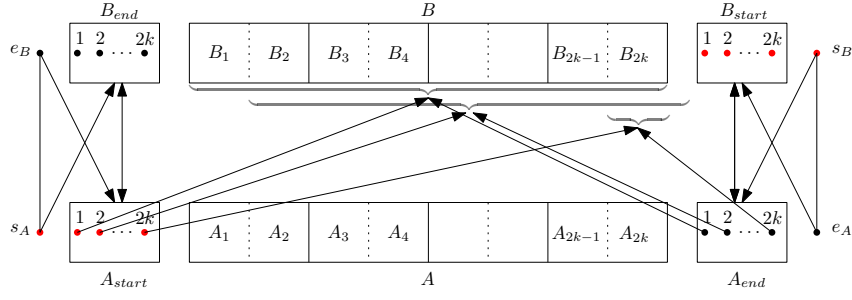
► **Observation 2.** Let I' be an independent set of G' and $p \in \{1, 2, \dots, k\}$ such that $I' \cap A_{2p-1} = \{u_{2p-1}\}$, $I' \cap A_{2p} = \{u_{2p}\}$, and $u_{2p-1} \sim u_{2p}$. Then the tokens on u_{2p-1} and u_{2p} cannot move to B .

If at some point in the reconfiguration sequence two tokens are positioned on equivalent vertices in A , then these tokens lock each other at their respective position in some sense. Note that by symmetry of the construction, the same observation can be made when two tokens are positioned on equivalent vertices in B . On the contrary, if two tokens on the same copies of V_p in A are positioned on two non-equivalent vertices we have the following:

► **Observation 3.** Let I' be an independent set of G' and $p \in \{1, 2, \dots, k\}$ such that $I' \cap A_{2p-1} = \{u_{2p-1}\}$, $I' \cap A_{2p} = \{u_{2p}\}$, and $u_{2p-1} \not\sim u_{2p}$. Then $I' \cap (B_{2p-1} \cup B_{2p}) = \emptyset$.

This observation not only ensures that $B_{2p-1} \cup B_{2p} = \emptyset$ but also ensures that no other token but the ones positioned on u_{2p-1} and u_{2p} can move to $B_{2p-1} \cup B_{2p}$. Then, by Observations 2 and 3, either there are two tokens on equivalent vertices in $A_{2p-1} \cup A_{2p}$ and then these tokens cannot move to B (and ensures that if there is a token on $B_{2p-1} \cup B_{2p}$ it must be on an equivalent vertex), or there are two tokens on non-equivalent vertices forbidding any other token to move to $B_{2p-1} \cup B_{2p}$.

Definition of the initial and target independent sets. The initial independent set I_s consists of two sets of $2k$ vertices A_{start} and B_{start} plus two vertices s_A, s_B included in \mathcal{A} , and the target independent set I_e consists in two sets of $2k$ vertices A_{end} and B_{end} plus two



■ **Figure 3** The constructed graph G' . Vertices in red are the vertices of I_s . An arrow between a vertex v and a subset of vertices indicates that v is complete to this subset. An arrow between a vertex v and a brace indicates that v is complete to the subsets included in the brace. A double arrow between two sets indicate these sets induce a complete bipartite graph. The connections between A and $B_{end} \cup B_{start}$ are symmetric and have been omitted for the sake of clarity.

vertices e_A, e_B included in \mathcal{B} . The two sets I_s and I_e are disjoint from $A \cup B$. The graph induced by $A_{start} \cup B_{end} \cup \{s_A, e_B\}$ and the graph induced by $A_{end} \cup B_{start} \cup \{s_B, e_A\}$ are complete bipartite graphs. The main goal of this section is to explain how to connect the set $A_{start} \cup B_{start}$ and the set $A_{end} \cup B_{end}$ to $A \cup B$ in order to ensure that any reconfiguration sequence transforming one into the other enforces the $2k$ tokens starting on A_{start} and the $2k$ tokens starting on B_{start} to switch sides by going through $A \cup B$. More particularly, we will show the existence of an independent set that satisfies the condition of Observation 1 in any such reconfiguration sequence, giving a multicolored independent set of G . For $p \in \{1, 2, \dots, 2k\}$, we denote by $a_{s,p}$ and $b_{s,p}$ the vertices of A_{start} and B_{start} respectively and we denote by $a_{e,p}$ and $b_{e,p}$ the vertices of A_{end} and B_{end} respectively. These vertices are connected to $A \cup B$ as follows:

1. the vertices $a_{s,p}$ and $a_{e,p}$ are complete to $B - \cup_{i=1}^{p-1} B_i$, and
2. the vertices $b_{s,p}$ and $b_{e,p}$ are complete to $A - \cup_{i=1}^{p-1} A_i$.

An illustration of the full construction is given in Figure 3. By construction, no token starting on $A_{start} \cup \{s_A\}$ can move to $B_{end} \cup \{e_B\}$ as long as there are at least two tokens on $A_{start} \cup \{s_A\}$ (and the same goes for $B_{start} \cup \{s_B\}$ and $A_{end} \cup \{e_A\}$). Since there are initially $2k + 1$ tokens on $A_{start} \cup \{s_A\}$ and since $N(s_A) \cap B = \emptyset$, the $2k$ tokens initially on A_{start} must move to B at some point in the sequence, and the same goes for B_{start} and A . The tokens initially on s_A and s_B have a special role and act as “locks”: without these token, the last token remaining on A_{start} (resp. B_{start}) would be able to move directly to B_{end} (resp. A_{end}) without never going through B (resp. A). Let us now explain the connections to $A \cup B$.

► **Observation 4.** Let I' be an independent set of G' such that $\{a_{s,p}, a_{s,p+1}, \dots, a_{s,2k}\} \subseteq I'$ for some $p < 2k$. Then the tokens on $\{a_{s,p+1}, a_{s,p+2}, \dots, a_{s,2k}\}$ are frozen. Furthermore the token on $a_{s,p}$ cannot move to $\cup_{i=p+1}^{2k} B_p$.

By symmetry, the same observation can be made for tokens on B_{start} . This shows that the tokens initially on A_{start} and B_{start} must respect a strict order to move respectively to B and A : the only tokens that can initially move are the tokens on $a_{s,1}$ and $b_{s,1}$ and these have no choice but to move to B_1 and A_1 respectively. After such a move the tokens on $a_{s,2}$ and $b_{s,2}$ are free to move to B_2 and A_2 respectively, and so on. Suppose that after the first 4 moves, there is exactly one token in each of the four subset A_1, B_1, A_2 and B_2 . Then it is not hard to see - but will be formally proved in the next section - that these tokens lie on

equivalent vertices, corresponding to a unique vertex of G . By Observation 2 these tokens cannot move to the other side of the bipartite graph and must stay at the same position while the remaining tokens on A_{start} and B_{start} moves to $A \cup B$. With the full constructions of G' , I_s and I_e at hand, the direct part of the reduction is easy to obtain by moving the tokens in the right order. The complete proof is deferred to the full version of the paper [1].

► **Lemma 18.** *If there is a multicolored independent set of size k in G then there exists a reconfiguration sequence transforming I_s to I_e in G' .*

Well-organized configurations. To simplify the tracking of tokens along the transformation, we give different colors to the tokens initially on A_{start} and B_{start} . The tokens initially on A_{start} are the *blue tokens* and the tokens initially on B_{start} are the *red tokens*. Given a configuration C , $M_A(C)$ (resp. $M_B(C)$) is the maximum integer $p \in \llbracket 1, 2k \rrbracket$ such that there is a token on A_p (resp. B_p). By convention, if there is no token on $X \in \{A, B\}$, we set $M_X(C) = 0$. A configuration C is *well-organized* if there is a token on either s_A or e_B and on either s_B or e_A and if it satisfies the following conditions:

1. For every $p \leq M_A(C)$ and every $q \leq M_B(C)$ there is exactly one token on A_p and exactly one token on B_q .
2. If $M_A(C) < 2k$ then for every $M_A(C) < p \leq 2k$ there is a token on $a_{s,p}$. If $M_B(C) < 2k$ then for every $M_B(C) < q \leq 2k$ there is a token on $b_{s,q}$.

Since the construction and the definition of well-organized configurations are symmetric, we can always suppose that $M_A(C) \leq M_B(C)$ for any well-organized configuration C . Note that the initial configuration is well-organized. We say that two configurations C and C' are *adjacent* if C can be transformed into C' by moving exactly one token.

Throughout the proof let $S := C_1, \dots, C_N$ denote a shortest reconfiguration sequence from I_s to I_e . We say that a token *moves from a set X to a set Y at time t* and we write $(t : X \rightarrow Y)$ if there exists two sets $X, Y \subseteq V(G')$ and two vertices $x \in X, y \in Y$ such that $C_{t+1} = C_t - \{x\} + \{y\}$. When the sets X and Y contain exactly one vertex we write $(t : x \rightarrow y)$ by abuse of notation. A move that transforms a well-organized configuration into a configuration that is not well-organized is a *bad move*. We aim to show the following:

► **Lemma 19.** *A shortest reconfiguration sequence from I_s to I_e contains no bad move.*

With Lemma 19 at hand, the proof of the converse part of the reduction easily follows:

► **Lemma 20.** *If there exists a reconfiguration sequence from I_s to I_e in G' , then there exists a multicolored independent set in G .*

Proof. Consider a shortest reconfiguration sequence S from I_s to I_e , which exists by supposition. By Lemma 19 this sequence contains no bad moves, therefore all the configurations of S are well-organized since the initial configuration is. Consider the configuration C just before the first token reaches $A_{end} \cup B_{end}$ (which exists since $A_{end} \cup B_{end} \subseteq I_e$). By the definition of well-organized configurations there can be no token on $A_{start} \cup B_{start}$ in C and thus we have $M_A(C) = M_B(C) = 2k$. Then by Observation 1 there exists a multicolored independent set in G . ◀

The proof of Lemma 19 is deferred to the full version of the paper [1].

References

- 1 Valentin Bartier, Nicolas Bousquet, Clément Dallard, Kyle Lomer, and Amer E. Mouawad. On girth and the parameterized complexity of token sliding and token jumping. *CoRR*, abs/2007.01673, 2020. [arXiv:2007.01673](https://arxiv.org/abs/2007.01673).
- 2 Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, Yota Otachi, and Florian Sikora. Token sliding on split graphs. In *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, pages 13:1–13:17, 2019. [doi:10.4230/LIPIcs.STACS.2019.13](https://doi.org/10.4230/LIPIcs.STACS.2019.13).
- 3 Marthe Bonamy and Nicolas Bousquet. Token sliding on chordal graphs. *CoRR*, abs/1605.00442, 2016. [arXiv:1605.00442](https://arxiv.org/abs/1605.00442).
- 4 Édouard Bonnet, Nicolas Bousquet, Pierre Charbit, Stéphan Thomassé, and Rémi Watrigant. Parameterized complexity of independent set in H-free graphs. In Christophe Paul and Michal Pilipczuk, editors, *13th International Symposium on Parameterized and Exact Computation, IPEC 2018, August 20-24, 2018, Helsinki, Finland*, volume 115 of *LIPIcs*, pages 17:1–17:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. [doi:10.4230/LIPIcs.IPEC.2018.17](https://doi.org/10.4230/LIPIcs.IPEC.2018.17).
- 5 Paul S. Bonsma, Marcin Kaminski, and Marcin Wrochna. Reconfiguring independent sets in claw-free graphs. In *Algorithm Theory - SWAT 2014 - 14th Scandinavian Symposium and Workshops, Copenhagen, Denmark, July 2-4, 2014. Proceedings*, pages 86–97, 2014.
- 6 Nicolas Bousquet, Arnaud Mary, and Aline Parreau. Token jumping in minor-closed classes. In *Fundamentals of Computation Theory - 21st International Symposium, FCT 2017, Bordeaux, France, September 11-13, 2017, Proceedings*, pages 136–149, 2017. [doi:10.1007/978-3-662-55751-8_12](https://doi.org/10.1007/978-3-662-55751-8_12).
- 7 Richard C. Brewster, Sean McGuinness, Benjamin Moore, and Jonathan A. Noel. A dichotomy theorem for circular colouring reconfiguration. *Theor. Comput. Sci.*, 639:1–13, 2016.
- 8 Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Connectedness of the graph of vertex-colourings. *Discrete Mathematics*, 308(56):913–919, 2008.
- 9 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. [doi:10.1007/978-3-319-21275-3](https://doi.org/10.1007/978-3-319-21275-3).
- 10 Erik D. Demaine, Martin L. Demaine, Eli Fox-Epstein, Duc A. Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada. Polynomial-time algorithm for sliding tokens on trees. In *Algorithms and computation*, volume 8889 of *Lecture Notes in Comput. Sci.*, pages 389–400. Springer, Cham, 2014. [doi:10.1007/978-3-319-13075-0_31](https://doi.org/10.1007/978-3-319-13075-0_31).
- 11 Eli Fox-Epstein, Duc A. Hoang, Yota Otachi, and Ryuhei Uehara. Sliding token on bipartite permutation graphs. In *Algorithms and Computation - 26th International Symposium, ISAAC 2015, Nagoya, Japan, December 9-11, 2015, Proceedings*, pages 237–247, 2015.
- 12 Sevag Gharibian and Jamie Sikora. Ground state connectivity of local hamiltonians. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 617–628, 2015. [doi:10.1007/978-3-662-47672-7_50](https://doi.org/10.1007/978-3-662-47672-7_50).
- 13 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.
- 14 Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theor. Comput. Sci.*, 343(1-2):72–96, 2005. [doi:10.1016/j.tcs.2005.05.008](https://doi.org/10.1016/j.tcs.2005.05.008).
- 15 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. [doi:10.1016/j.tcs.2010.12.005](https://doi.org/10.1016/j.tcs.2010.12.005).
- 16 Takehiro Ito, Marcin Kamiński, and Erik D. Demaine. Reconfiguration of list edge-colorings in a graph. *Discrete Applied Mathematics*, 160(15):2199–2207, 2012.

- 17 Takehiro Ito, Marcin Kamiński, and Hirotaka Ono. Fixed-parameter tractability of token jumping on planar graphs. In *Algorithms and computation*, volume 8889 of *Lecture Notes in Comput. Sci.*, pages 208–219. Springer, Cham, 2014. doi:10.1007/978-3-319-13075-0_17.
- 18 Takehiro Ito, Marcin Kaminski, Hirotaka Ono, Akira Suzuki, Ryuhei Uehara, and Katsuhisa Yamanaka. On the parameterized complexity for token jumping on graphs. In *Theory and Applications of Models of Computation - 11th Annual Conference, TAMC 2014, Chennai, India, April 11-13, 2014. Proceedings*, pages 341–351, 2014.
- 19 Takehiro Ito, Hiroyuki Nooka, and Xiao Zhou. Reconfiguration of vertex covers in a graph. *IEICE Transactions*, 99-D(3):598–606, 2016. URL: http://search.ieice.org/bin/summary.php?id=e99-d_3_598.
- 20 Wm. Woolsey Johnson and William E. Story. Notes on the “15” puzzle. *American Journal of Mathematics*, 2(4):397–404, 1879.
- 21 Marcin Kaminski, Paul Medvedev, and Martin Milanič. Complexity of independent set reconfigurability problems. *Theor. Comput. Sci.*, 439:9–15, 2012. doi:10.1016/j.tcs.2012.03.004.
- 22 Marcin Kamiński, Paul Medvedev, and Martin Milanič. Complexity of independent set reconfigurability problems. *Theoretical Computer Science*, 439:9–15, 2012.
- 23 Graham Kendall, Andrew J. Parkes, and Kristian Spoerer. A survey of NP-complete puzzles. *ICGA Journal*, pages 13–34, 2008.
- 24 Jeong Han Kim. The Ramsey number $R(3, t)$ has order of magnitude $t^2/\log t$. *Random Structures Algorithms*, 7(3):173–207, 1995. doi:10.1002/rsa.3240070302.
- 25 Daniel Lokshantov and Amer E. Mouawad. The complexity of independent set reconfiguration on bipartite graphs. *ACM Trans. Algorithms*, 15(1):7:1–7:19, 2019. doi:10.1145/3280825.
- 26 Daniel Lokshantov, Amer E. Mouawad, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. Reconfiguration on sparse graphs. In *Algorithms and Data Structures - 14th International Symposium, WADS 2015, Victoria, BC, Canada, August 5-7, 2015. Proceedings*, pages 506–517, 2015.
- 27 Anna Lubiw and Vinayak Pathak. Flip distance between two triangulations of a point set is NP-complete. *Comput. Geom.*, 49:17–23, 2015. doi:10.1016/j.comgeo.2014.11.001.
- 28 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.
- 29 Amer E. Mouawad, Naomi Nishimura, and Venkatesh Raman. Vertex cover reconfiguration and beyond. In *Algorithms and Computation - 25th International Symposium, ISAAC 2014, Jeonju, Korea, December 15-17, 2014, Proceedings*, pages 452–463, 2014.
- 30 Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018. doi:10.3390/a11040052.
- 31 Jan van den Heuvel. The complexity of change. *Surveys in Combinatorics 2013*, 409:127–160, 2013.
- 32 Marcin Wrochna. Reconfiguration in bounded bandwidth and treedepth. *CoRR*, abs/1405.0847, 2014. arXiv:1405.0847.
- 33 Marcin Wrochna. Homomorphism reconfiguration via homotopy. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, pages 730–742, 2015.
- 34 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007. doi:10.4086/toc.2007.v003a006.