Paths to Trees and Cacti

Akanksha Agrawal[a], Lawqueen Kanesh[b], Saket Saurabh[b,c], Prafullkumar Tale[b,*]

[a] *Ben-Gurion University of the Negev, Beersheba, Israel*
[b] *The Institute of Mathematical Sciences, HBNI, Chennai, India*
[c] *Department of Informatics, University of Bergen, Bergen, Norway*

Abstract

For a family of graphs $\mathcal{F}$, the $\mathcal{F}$-CONTRACTION problem takes as an input a graph G and an integer k, and the goal is to decide whether there exists $F \subseteq E(G)$ of size at most k such that G/F belongs to $\mathcal{F}$. When $\mathcal{F}$ is the family of paths, trees or cacti, then the corresponding problems are PATH CONTRACTION, TREE CONTRACTION and CACTUS CONTRACTION, respectively. It is known that TREE CONTRACTION and CACTUS CONTRACTION do not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$, while PATH CONTRACTION admits a kernel with $\mathcal{O}(k)$ vertices. The starting point of this article is the following natural questions: *What is the structure of the family of paths that allows* PATH CONTRACTION *to admit a polynomial kernel? Apart from the size of the solution, what other additional parameters should we consider so that we can design polynomial kernels for these basic contraction problems?* To design polynomial kernels, we consider the family of trees with the bounded number of leaves (note that the family of paths are trees with at most two leaves). In particular, we study BOUNDED TREE

☆Preliminary version of this article has appeared in 10^{th} International Conference on Algorithms and Complexity (CIAC 2017).

☆☆The research leading to these results has received funding from the European Research Council (ERC) via grant PARAPPROX, reference 306992. The first and the third authors are supported by the PBC Program of Fellowships for Outstanding Post-doctoral Researchers from China and India (No. 5101479000); and Horizon 2020 Framework program, ERC Consolidator Grant LOPPRE (No. 819416), respectively.

*Corresponding author

Email addresses: `agrawal@post.bgu.ac.il` (Akanksha Agrawal), `lawqueen@imsc.res.in` (Lawqueen Kanesh), `saket@imsc.res.in` (Saket Saurabh), `pptale@imsc.res.in` (Prafullkumar Tale)

Contraction (Bounded TC). Here, an input is a graph G, integers k and ℓ, and the goal is to decide whether or not, there exists a subset $F \subseteq E(G)$ of size at most k such that G/F is a tree with at most ℓ leaves. We design a kernel for Bounded TC with $\mathcal{O}(k\ell)$ vertices and $\mathcal{O}(k^2 + k\ell)$ edges. Finally, we study Bounded Cactus Contraction (Bounded CC) which takes as input a graph G and integers k and ℓ. The goal is to decide whether there exists a subset $F \subseteq E(G)$ of size at most k such that G/F is a cactus graph with at most ℓ leaf blocks in the corresponding block decomposition. For Bounded CC we design a kernel with $\mathcal{O}(k^2 + k\ell)$ vertices and $\mathcal{O}(k^2 + k\ell)$ edges. We complement our results by giving kernelization lower bounds for Bounded TC, Bounded OTC and Bounded CC by showing that unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ the size of the kernel we obtain is optimal.

Keywords: Kernel, Graph Contraction, Kernel Lower Bound

1. Introduction

Graph editing problems are one of the central problems in graph theory that have received a lot of attention in the realm of parameterized complexity. Some of the important graph editing operations are vertex deletion, edge deletion, edge addition, and edge contraction. For a family of graphs $\mathcal{F}$, the $\mathcal{F}$-Editing problem takes as an input a graph G and an integer k, and the objective is to decide if at most k edit operations can result in a graph that belongs to the graph family $\mathcal{F}$. In fact, the $\mathcal{F}$-Editing problem, where the edit operations are restricted to vertex deletion or edge deletion or edge addition or edge contraction alone have also been studied extensively in parameterized complexity. When we just focus on deletion operation (vertex/edge deletion) then the corresponding problem is called $\mathcal{F}$-Vertex (Edge) Deletion problem. For instance, the $\mathcal{F}$-Editing problems encompasses several NP-hard problems such as Vertex Cover, Feedback vertex set, Planar $\mathcal{F}$-Deletion, Interval Vertex Deletion, Chordal Vertex Deletion, Odd cycle transversal, Edge Bipartization, Tree Contraction, Path Contraction, Split Contraction, Clique Contraction etc. However, most of the study in paramterized complexity or classical complexity, have been restricted to combination of vertex deletion, edge deletion or edge addition [9, 7, 8, 6, 18, 20, 17, 25, 27, 29, 32, 13, 14, 15, 21, 2, 3, 34]. Only recently, edge contraction as an edit operation has started to gain attention in the realm of parameterized complexity. In this paper we study three edge-contraction problems from the perspective of kernelization complexity – one of the established subarea in parameterized complexity.

In parameterized complexity each problem instance is accompanied by a parameter k. A central notion in this field is the one of *fixed parameter tractable* (FPT). This means, for a given instance (I, k), solvability in time $\mathcal{O}(f(k)|I|^{\mathcal{O}(1)})$ where f is some function of k. Other important notion in parameterized complexity is *kernelization*, which captures the efficiency of data reduction techniques. A parameterized problem Π admits a kernel of size $g(k)$ (or $g(k)$-kernel) if there is a polynomial time algorithm (called *kernelization algorithm*) which takes as an input (I, k), and in time $\mathcal{O}(|I|^{\mathcal{O}(1)})$ returns an equivalent instance (I', k') of Π such that $|I'| + k' \leq g(k)$. Here, $g(\cdot)$ is a computable function whose value depends only on k. Depending on whether the function $g(\cdot)$ is *linear, polynomial* or *exponential*, the problem is said to admit a *linear, polynomial* or *exponential kernel*, respectively. It turns out that linear and polynomial kernels are most interesting from the kernelization perspective, because any problem that is fixed-parameter tractable admits an exponential kernel [10]. In this paper whenever we say *kernel*, we will refer to polynomial or linear kernels.

For several families of graphs $\mathcal{F}$, early papers by Watanabe et al. [35, 36] and Asano and Hirata [1] showed that $\mathcal{F}$-Edge Contraction is NP-complete. In the framework of parameterized complexity (or even the classical complexity), these problems exhibit properties that are quite different than those of problems where we only delete or add vertices and edges. For instance, deleting k edges from a graph such that the resulting graph is a tree is polynomial-time solvable. On the other hand, Asano and Hirata showed that Tree Contraction is NP-hard [1]. Furthermore, a well-known result by Cai [4] states that in a case $\mathcal{F}$ is a hereditary family of graphs with a finite set of forbidden induced subgraphs, then the graph modification problem defined by $\mathcal{F}$ and the edit operations restricted to vertex deletion, edge deletion, and edge addition admits a simple FPT algorithm. Indeed, for these problems, the result by Cai [4] does not hold when the edit operation is edge contraction. In particular, Lokshtanov et al. [31] and Cai and Guo [5] independently showed that if $\mathcal{F}$ is either the family of P_ℓ-free graphs for some $\ell \geq 5$ or the family of C_ℓ-free graphs for some $\ell \geq 4$, then $\mathcal{F}$-Edge Contraction is W[2]-hard. To the best of our knowledge, Heggernes et al. [24] were the first to explicitly study $\mathcal{F}$-Edge Contraction from the viewpoint of Parameterized Complexity. They showed that in case $\mathcal{F}$ is the family of trees, $\mathcal{F}$-Edge Contraction is FPT but does not admit a polynomial kernel, while in case $\mathcal{F}$ is the family of paths, the corresponding problem admits a faster algorithm and an $\mathcal{O}(k)$-vertex kernel. Golovach et al. [22] proved that if $\mathcal{F}$ is the family of planar graphs, then $\mathcal{F}$-Edge Contraction is again FPT. Moreover, Cai and Guo [5] showed that in case $\mathcal{F}$ is the family of cliques, $\mathcal{F}$-Edge Contraction is solvable in time $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$, while in case $\mathcal{F}$ is the family of chordal graphs, the problem is W[2]-hard. Heggernes et al. [26] developed an FPT algorithm for the case where $\mathcal{F}$ is the family of bipartite graphs. Later, a faster algorithm was proposed by Guillemot and Marx [23].

It is evident from our discussion that the complexity of the graph editing problem when restricted to edge contraction seems to be more difficult than their vertex or edge deletion counterparts. The starting point of our research is the following result by Heggernes et al. [24] who showed that Tree Contraction does not admit a polynomial kernel unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [24] and Path Contraction admits a linear vertex kernel.

We wanted to understand the structure of the family of paths that allows Path Contraction to admit a polynomial kernel. Apart from the size of the solution, what other additional parameters should we consider so that we can design polynomial kernels for these basic contraction problems? One of the natural candidates for such an extension is to consider the family of trees with the bounded number of leaves. With the goal to apprehend the understanding of the role the number of leaves plays in the kernelization complexity for contracting to the "path-like" graph, we study the problem which we call as Bounded Tree Contraction (Bounded TC). Formally, the problem is defined below.

> Bounded Tree Contraction **Parameter:** $k + \ell$
> **Input:** A graph G and integers k, ℓ
> **Question:** Does there exist $F \subseteq E(G)$ of size at most k such that G/F is a tree with at most ℓ leaves?

We give a kernel for Bounded TC with $\mathcal{O}(k\ell)$ vertices and $\mathcal{O}(k^2 + k\ell)$ edges. The approach we follow is similar to the one Heggernes et al. [24] used to obtain a linear kernel for Path Contraction. We observe that our algorithm works even when the input is a directed graph. In particular, we consider Bounded Out-Tree Contraction (Bounded OTC), which is defined as follows.

Bounded Out-Tree Contraction	**Parameter:** $k + \ell$
Input: A digraph D and integers k, ℓ	
Question: Does there exist $F \subseteq A(D)$ of size at most k such that D/A is an out-tree with at most ℓ leaves?	

By incorporating direction appropriately into our algorithm for Bounded TC, we get a kernel for Bounded OTC with $\mathcal{O}(k^2 + k\ell)$ vertices and arcs.

We also study the contraction problem for a class of graphs which generalizes trees – the family of cactus. Formally, the problem we study is defined as follows.

Bounded Cactus Contraction	**Parameter:** $k + \ell$
Input: A graph G and integers k, ℓ	
Question: Does there exist $F \subseteq E(G)$ of size at most k such that G/F is a cactus with at most ℓ leaves?	

For Bounded CC we give a kernel with $\mathcal{O}(k^2 + k\ell)$ vertices and edges. Finally, we give kernelization lower bound results. We complement all our kernelization algorithms by giving a matching lower bounds. In particular, we show that Bounded TC, Bounded OTC and Bounded CC do not admit better kernels unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

2. Preliminaries

Graph Theory. We consider graphs with finite number of vertices. For an *undirected graph* G, by $V(G)$ and $E(G)$ we denote the set of vertices and edges of G respectively. For a *directed graph (or digraph)* D, by $V(D)$ and $A(D)$ we denote the sets of vertices and directed edges (arcs) in D, respectively. Two vertices u, v are said to be adjacent in G (or in D) if there is an edge (arc) $uv \in E(G)$ (or in $A(D)$) and u, v are said to be endpoints of the edge (arc) uv. The neighbourhood of a vertex v, denoted by $N_G(v)$, is the set of vertices adjacent to v. For a vertex $v \in V(D)$, $N^-_D(v)$ denotes the set $\{u \in V(D) \mid uv \in A(D)\}$ of its in-neighbors and $N^+_D(v)$ denotes the set $\{u \in V(D) \mid vu \in A(D)\}$ of its out-neighbors. The neighbourhood of a vertex $v \in V(D)$ is the set $N_D(v) = N^+_D(v) \cup N^-_D(v)$. The closed neighbourhood of a vertex is $N_G[v] = N_G(v) \cup \{v\}$. Degree of a vertex $\mathsf{deg}_G(u)$, is the cardinality of the set $N_G(v)$. In case of digraphs, the in-degree and out-degree of a vertex v, denoted by $\mathsf{deg}^-_D(v)$, $\mathsf{deg}^+_D(v)$, is $|N^-_D(v)|$ and $|N^+_D(v)|$ respectively. The (total) degree of v, denoted by $\mathsf{deg}_G(v)$, is the sum of its in-degree and out-degree. The subscripts in the notation for neighbourhood and degree will be omitted if the context is clear. For every digraph we associate an underdirected graph, called *underlying graph* obtained by forgetting the direction of arcs. Formally, for a digraph D, underlying graph G_D is defined as $V(G_D) = V(D)$ and $E(G_D) = \{uv|\ uv \in A(D) \text{ or } vu \in A(D)\}$. For $F \subseteq E(G)$, $V(F)$ denotes the set of endpoints of edges (or arcs) in F. For a subset $S \subseteq V(G)$, by $G - S$ and $G[S]$ we denote the graph obtained by deleting vertices in S from G and the graph obtained by removing vertices in $V(G) \setminus S$ from G, respectively. For $F \subseteq E(G)$, $G - F$ is graph obtained by deleting edges in F from G. For $X, Y \subseteq V(G)$, we say X, Y are adjacent if there exist an edge with one end point in X and other in Y. A *subdivision* of an edge $uv \in E(G)$ is an operation that deletes an edge uv, adds a vertex w to $V(G)$, and makes it adjacent to u and v.

A graph G is called *connected* if there is a path between every pair of distinct vertices in G. It is called *disconnected* otherwise. A *component* of a graph is a maximal connected subgraph. A *cut-vertex* in G is a vertex v such that the number of components in $G \setminus \{v\}$ is strictly more than the number of components in G. A graph that has no cut-vertex is called a *2-connected* graph. An edge uv of a graph G is called a *cut-edge* if the number of connected components in $G - \{uv\}$ is more than the number of connected components in G. We note that the number of connected components after removal of an edge can increase by at most 1. A *directed graph* (*digraph*) is connected (disconnected, 2-connected) if its underlying undirected graph is connected (disconnected, 2-connected).

A *path* $P = (v_1, \ldots, v_q)$ is an ordered collection of distinct vertices where every consecutive pair of vertices are adjacent. The vertices of P is the set $\{v_1, \ldots, v_q\}$ and is denoted by $V(P)$. A *cycle* is a path $P = (v_1, \ldots, v_q)$ such that $(v_1, v_q) \in E(G)$. A *leaf* is a vertex with $\mathsf{deg}_G(v) = 1$. A *tree* is a connected graph without a cycle. An *out-tree* T is a digraph where each vertex has in-degree at most 1 and underlying undirected graph is a tree. A vertex v of an out-tree is called a *leaf* if $\mathsf{deg}^-(v) = 1$ and $\mathsf{deg}^+(v) = 0$. The *root* of an out-tree is the unique vertex that has no in-neighbour. The number of leaves in a tree (or out-tree), denoted by $L(T)$, is number of vertices whose degree is one. A *cactus* is an undirected graph such that every edge is contained in at most one cycle. We use following result to bound the summation of degrees of vertices with degree 3 or more in a tree. Following proposition also implies that in a tree, the number of vertices with degree at least 3 is upper bounded by number of vertices with degree 1.

Proposition 1 (Lemma 3 [30]). *For a tree T, if V_1, V_2, V_3 are the set of vertices of degree 1, degree 2 and at least 3 respectively, then $\sum_{v \in V_3} \mathsf{deg}_T(v) \leq 3|V_1|$.*

Proof. By definition, $|V(T)| = |V_1| + |V_2| + |V_3|$. Since there are no isolated vertices, $\sum_{v \in V(T)} \mathsf{deg}_T(v) = 2|E(T)|$. Since T is a tree, $|E(T)| < |V(T)|$. This implies $\sum_{v \in V(T)} \mathsf{deg}_T(v) < 2(|V_1| + |V_2| + |V_3|)$. Substituting lower bounds of degrees for each set, we get $|V_1| + 2|V_2| + 3|V_3| \leq \sum_{v \in V_1} \mathsf{deg}_T(v) + \sum_{v \in V_2} \mathsf{deg}_T(v) + \sum_{v \in V_3} \mathsf{deg}_T(v) = \sum_{v \in V(T)} \mathsf{deg}_T(v)$. Using the two equations we get $|V_1| + 2|V_2| + 3|V_3| \leq 2(|V_1| + |V_2| + |V_3|)$ which implies $|V_3| \leq |V_1|$. Adding the degree of vertices only in V_3 we get $\sum_{v \in V_3} \mathsf{deg}_T(v) = 2|V(T)| - (\sum_{v \in V_1} \mathsf{deg}_T(v) + \sum_{v \in V_2} \mathsf{deg}_T(v)) = 2(|V_1| + |V_2| + |V_3|) - (|V_1| + 2|V_2|) \leq |V_1| + 2|V_3|$. Using the bound of $|V_3|$, $\sum_{v \in V_3} \mathsf{deg}_T(v) \leq 3|V_1|$. $\square$

A *block* is a connected maximal connected subgraph which is 2-connected. A block in a graph is either an induced maximal 2-connected subgraph or an edge or an isolated vertex. Two distinct blocks in the graph can intersect in at most one vertex. A vertex contained in at least two blocks must be a cut-vertex in the graph. Let K be the set of cut-vertices and $\mathcal{B}$ be the set of blocks in G. A *block-decomposition* of G is a bipartite graph $\mathcal{D}$ with the vertex set $K \uplus \mathcal{B}$. Furthermore, $aB \in E(\mathcal{D})$ for $a \in K$ and $B \in \mathcal{B}$ if and only if $a \in V(B)$. Here, we slightly abuse the notation and use $\mathcal{B}$ to denote the set of blocks in G as well as vertices corresponding to the blocks of G in $\mathcal{D}$. It is known that a block decomposition of a connected graph is unique and is a tree [11, Proposition 3.1.2]. For the sake of clarity, we call vertices in $\mathcal{D}$ as nodes. See Figure 1. The *number of leaves* of cactus is defined as the number of leaves in its block decomposition. Since every edge in cactus is part of at most one cycle, if G is a cactus then a block of G is either a cycle or an edge.

Graph Contraction. A *contraction of an edge* is an operation that merges its two end points and removes self loop, parallel edges created in the process. In graph G, we can contract an edge uv by deleting the vertices u, v in G followed by adding a vertex w to $V(G)$ and making it adjacent to vertices that were adjacent to either u or v. The resulting graph is denoted by G/uv. Formally, $V(G/uv) = (V(G) \setminus \{u, v\}) \cup \{w\}$ and $E(G/uv) = \{xy \mid x, y \in V(G) \setminus \{u, v\}, xy \in E(G)\} \cup \{wx \mid x \in (N_G(u) \cup N_G(v)) \setminus \{u, v\}\}$. For a digraph D, contracting an arc uv results in a digraph D/uv on the vertex set $V' = V(D) \setminus \{u, v\} \cup \{w\}$ with $A(D/uv) = \{xy \mid xy \in A(D)$ and $x, y \in$

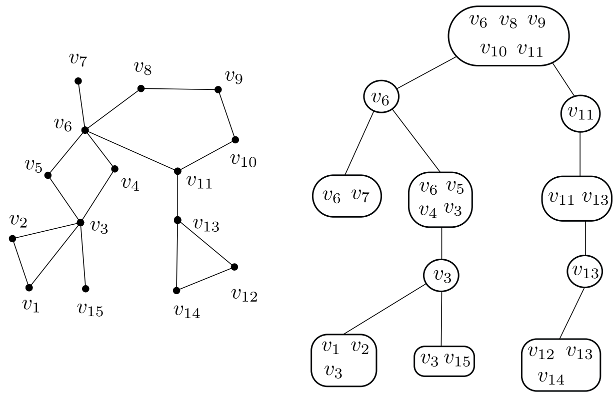

Figure 1: Cactus graph and its block decomposition.

$V'\} \cup \{xw \mid xu \in A(D)\} \cup \{wy \mid uy \in A(D)\} \cup \{xw \mid xv \in A(D)\} \cup \{wy \mid vy \in A(D)\}$.

We can generalize the above definition of contraction of an edge to *contraction of a connected subgraph*. Consider a subset U of $V(G)$ such that $G[U]$ is connected. Fix a spanning tree of $G[U]$ and let T_U be the set of edges in that spanning tree. We use G/T_U to denote the graph obtained from G by merging all vertices in U and removing self-loops, parallel edges created in the process. Recall that for a set of edges $F \subseteq E(G)$, set $V(F)$ denotes the union of endpoints of edges in F. We use G/F to denote the graph obtained from G by contracting each connected component of $G[V(F)]$ into a vertex.

A graph G is *isomorphic* to a graph H if there exists a *one-to-one* and *onto* function $\varphi : V(G) \to V(H)$ such that for $u, v \in V(G)$, $uv \in E(G)$ if and only if $\varphi(u)\varphi(v) \in E(H)$. In this article, we do not distignuish between isomorphic graphs. With slight abuse of notation, we say graph G/F is obtained from G by sequentially contracting the edges in F. The graph G/F is oblivious to the order in which edges in F are contracted. For a set of edges $F \subseteq E(G)$, G/F denotes the graph obtained from G by sequentially contracting the edges in F. A graph G is *contractible* to a graph H, if there exists $F \subseteq E(G)$ such that G/F is *isomorphic* to H. In other words, G is contractible to H if there exists a *onto* function $\psi : V(G) \to V(H)$ such that the following properties hold.

- For any vertex $h \in V(H)$, graph $G[W(h)]$ is connected where $W(h) := \{v \in V(G) \mid \psi(v) = h\}$.

- For any pair of vertices $h, h' \in V(H)$, $hh' \in E(H)$ if and only if $W(h)$ and $W(h')$ in G are adjacent.

For digraphs, we define the notion of contraction in an analogous way. For any pair of vertices $h, h' \in V(H)$, $hh' \in A(H)$ if and only if there is an arc directed from a vertex in $W(h)$ to a vertex in $W(h')$ in D. Let $\mathcal{W} = \{W(h) \mid h \in V(H)\}$. Observe that $\mathcal{W}$ defines a partition of vertices in G. We call $\mathcal{W}$ an *H-witness structure* of G. The sets in $\mathcal{W}$ are called *witness sets*. If a *witness set* contains more than one vertex of G then it is a *big* witness-set, otherwise it is a *small* witness set. A graph G is said to be *k-contractible* to a graph H if there exists $F \subseteq E(G)$ such that G/F is isomorphic to H and $|F| \leq k$. We will use the following observation in designing our kernels.

Observation 1. *Let G be a graph (or diagraph) which is k-contractible to a graph (or diagraph) H and $\mathcal{W}$ be an H-witness structure of G. Then,*

- *$|V(G)| \leq |V(H)| + k$;*
- *No witness set in $\mathcal{W}$ contains more than $k+1$ vertices;*
- *$\mathcal{W}$ has at most k big witness sets;*
- *The union of big witness sets in $\mathcal{W}$ contains at most $2k$ vertices.*

Parameterized Complexity.. We say that two instances, (I, k) and (I', k'), of a parameterized problem Q are *equivalent* if $(I, k) \in Q$ if and only if $(I', k') \in$ Q. A *reduction rule*, for a parameterized problem Q is an algorithm that takes an instance (I, k) of Q as input and outputs an instance (I', k') of Q in time polynomial in $|I|$ and k. If (I, k) and (I', k') are equivalent instances then we call the reduction rule is *safe*. A parameterized problem Q admits a kernel of size $g(k)$ (or $g(k)$-kernel) if there is a polynomial time algorithm (called *kernelization algorithm*) which takes as an input (I, k), and in time $\mathcal{O}(|I|^{\mathcal{O}(1)})$ returns an equivalent instance (I', k') of Q such that $|I'| + k' \leq g(k)$. Here, $g(\cdot)$ is a computable function whose value depends only on k. We mention definition of *polynomial compression* which we use while proving lower bounds on kernels.

Definition 2.1. *A polynomial compression of a parameterized language $Q \subseteq \Sigma^* \times \mathbb{N}$ into a language $\Pi \subseteq \Sigma^*$ is an algorithm that takes as input an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, and in time polynomial in $|x| + k$ returns a string y such that:*

- $|y| \leq p(k)$ *for some polynomial* $p(\cdot)$*, and*
- $y \in \Pi$ *if and only if* $(x, k) \in Q$.

For more details on parameterized complexity, we refer the reader to the books of Downey and Fellows [12], Flum and Grohe [16], Niedermeier [33], Cygan et al. [10] and the more recent book by Fomin et al. [19].

3. Kernel for Bounded Tree Contraction

In this section we design a kernelization algorithm for Bounded Tree Contraction (Bounded TC). Our algorithm is inspired by the kernelization algorithm for Path Contraction presented in [24]. Let (G, k, ℓ) be an instance of Bounded TC. It is safe to assume that the input graph G is connected otherwise it is a trivial No instance.

We first present some preliminary results. For every integer $\ell \geq 2$, consider a set of trees which has at most ℓ leaves. For $\ell = 2$, this set is a collection of all paths. The following observation states that this set of graphs is closed under edge contraction.

Observation 2. *Let* T *be a tree and* T' *be the graph obtained from* T *by contracting an edge* v_1v_2 *in* $E(T)$*. If* T *has at most* ℓ *leaves then* T' *is a tree with at most* ℓ *leaves.*

This set is also closed under an operation of *uncontracting* an edge with some additional conditions. We first formally define such operation. Consider a tree T and one of its internal vertex, say v. Let L, R be a partition of $N(v)$ such that none of them is an empty set. We define operation Split(T, v, L, R) as follows. See Figure 2 for illustration.

Split(T, v, L, R)**:** Remove vertex v and add two vertices v_1 and v_2. Make v_1 adjacent with every vertex in L and v_2 adjacent with every vertex in R. Add edge v_1v_2. If T' is the graph obtained from T by this operation then $V(T') = (V(T) \setminus \{v\}) \cup \{v_1, v_2\}$ and $E(T') = (E(T) \setminus (\{vu \mid u \in N(v)\})) \cup \{v_1u \mid u \in L\} \cup \{v_2u \mid u \in R\} \cup \{v_1v_2\}$.

The following lemma proves that this operation on a tree results in another tree with the same number of leaves.

Lemma 3.1. *Let* T *be a tree,* v *be an internal vertex of* T *and* $N(v)$ *is partitioned into two non-empty sets* L *and* R*. Let* T' *is the graph obtained*

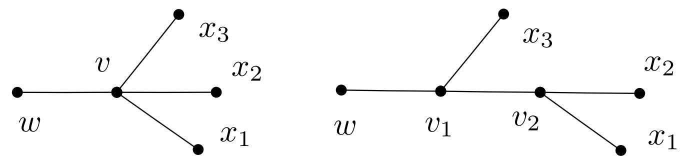

Figure 2: Operation $\textsc{Split}(T, v, L, R)$ with $L = \{x_3\}$ and $R = \{x_1, x_2\}$.

from T after applying $\textsc{Split}(T, v, L, R)$. *If T has at most ℓ leaves then T' is a tree with at most ℓ leaves.*

Proof. First, we prove that T' is a tree. Suppose not, then there exists a cycle in T'. Let C' be an induced cycle in T'. If C' contains at most one of v_1, v_2, then we can obtain a cycle C in T by replacing v_1 or v_2 by v. Otherwise, C contain both v_1 and v_2. Since, C' is an induced cycle and $v_1v_2 \in E(T')$, vertices v_1, v_2 appear consecutively in C'. Again, by replacing v_1v_2 by vertex v, we obtain a cycle in T which is a contradiction. Hence, T' is acyclic. Note that v_1v_2 is an edge in T' with $N_{T'}(v_1) \setminus \{v_2\} = L \neq \emptyset$ and $N_{T'}(v_2) \setminus \{v_1\} = R \neq \emptyset$, therefore v_1, v_2 are not leaves in T'. All leaves in T' remains as leaf vertices in T'. This implies that number of leaves in T' is no more than the number of leaves in T. $\square$

We now start describing a kernelization algorithm. It has only one reduction rule which finds and contracts a *irrelevant* edge. We argue that a cut edge whose removal results in two *large* connected components is an irrelevant edge.

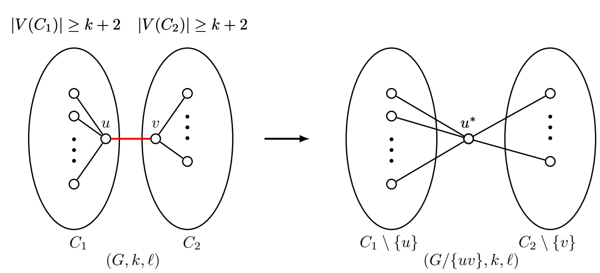

Figure 3: An illustration of Reduction Rule 3.1.

Reduction Rule 3.1. *Let uv be a cut-edge in G and C_1, C_2 be the connected components in $G - \{uv\}$. If $|V(C_1)|, |V(C_2)| \geq k + 2$ then contract uv. The resulting instance is (G', k, ℓ), where $G' = G/\{uv\}$.*

Informally speaking, since edge uv is a cut-edge, it is not a part of any cycle. We do not need to contract it to destroy any cycle. The only reason we might include it in a solution is to reduce the number of leaves in the resultant tree. As the sizes of both connected components of $G - \{uv\}$ is at least $k + 2$, contracting at most k edges can not destroy either of the connected components. Hence no endpoints of uv can be part of a leaf in the resulting graph. In other words, uv is irrelevant with respect to any solution of size at most k and can safely be contracted.

Lemma 3.2. *Reduction rule 3.1 is safe.*

Proof. We argue that (G, k, ℓ) is a Yes instance of Bounded TC if and only if (G', k, ℓ) is a Yes instance of Bounded TC.

To prove forward direction, let (G, k, ℓ) be a Yes instance of Bounded TC. Let F be a set of at most k edges such that G/F be a tree with at most ℓ leaves. By Observation 2, graph $G/(F \cup \{uv\})$ is also a tree with at most ℓ leaves. Note that $G/(F \cup \{uv\}) = (G/\{uv\})/(F \setminus \{uv\}) = G'/(F \setminus \{uv\})$. Hence $G'/(F \setminus \{uv\})$ is a tree with at most ℓ leaves. Since $|F \setminus \{uv\}| \leq |F| \leq k$, we can conclude that (G', k, ℓ) is a Yes instance of Bounded TC.

To prove reverse direction, let (G', k, ℓ) be a Yes instance of Bounded TC. Let F' be a set of at most k edges such that $G'/F' = T'$ is a tree with at most ℓ leaves. We first argue that G is $(|F'| + 1)$-contractible to a tree, say T_1, which has at most ℓ leaves. Using Split operation on T_1 we argue that G is actually $|F'|$-contractible to a tree with at most ℓ leaves.

Let $\mathcal{W}'$ be a T'-witness structure of G'. Let u^* be the vertex resulting while contracting edge uv in G to get G'. Consider vertex t^* in $V(T')$ such that u^* is in $W(t^*)$. Define set $W(t_1) := (W(t^*) \setminus \{u^*\}) \cup \{u, v\}$. Let $\mathcal{W}_1$ be the witness structure obtained from $\mathcal{W}'$ by removing $W(t^*)$ and adding $W(t_1)$. Note that $\mathcal{W}_1$ partitions $V(G)$ and for each W in $\mathcal{W}_1$, $G[W]$ is connected. Let T_1 be a graph obtained from G by contracting witness sets in $\mathcal{W}_1$. In other words, $\mathcal{W}$ is a T_1-witness structure of G. Note that T_1 can be obtained from G by contracting all edges in $F' \cup \{uv\}$. This implies T_1 can be obtained from G' by contracting all edges in F' and hence it is a tree with at most ℓ leaves. We conclude that G is $(|F'| + 1)$-contractible to a tree with at most ℓ leaves.

Since uv is a cut-edge in G, it is also a cut-edge in $G[W(t_1)]$. Let C_u and C_v be the connected components of $G[W(t_1)] - \{uv\}$ containing u and v, respectively. Further, let $W_u = V(C_u)$, $W_v = V(C_v)$. Consider a witness structure $\mathcal{W}$ of G obtained from $\mathcal{W}_1$ by removing $W(t_1)$ and adding W_u and W_v. Notice that $\mathcal{W}$ partitions $V(G)$ and for each W in $\mathcal{W}$, $G[W]$ is connected. Moreover, we need $|F'|$ many edges to contract all witness sets in $\mathcal{W}$. Let T be a graph obtained by contracting all witness sets in $\mathcal{W}$. In other words, $\mathcal{W}$ is a T-witness structure of G. Note that G is $|F'|$-contractible to T. The only thing which remains to prove is that T is a tree with at most ℓ leaves. We prove this by showing that T can be obtained from T_1 by SPLIT operation at vertex t_1.

We start by proving that t_1 is an internal vertex in T_1 by showing that it has at least two neighbors.

Claim. Vertex t_1 in T_1 has at least two neighbors.

Proof. Each witness set in $\mathcal{W}_1$ is of size at most $k+2$ and hence $|W(t_1)| \leq k+2$. If t_1 is the only vertex in T_1, then all the vertices in $(V(C_1) \cup V(C_2)) \setminus \{u, v\}$ are in $W(t_1)$. This implies that $|W(t_1)| \geq 2k+3$ which is a contradiction. If t_1 has unique neighbor, say $\hat{t}$, in $V(T_1)$, then $V(C_1) \cap W(\hat{t})$ and $V(C_2) \cap W(\hat{t})$ are both non empty as $|V(C_1)|$, $|V(C_2)| \geq k+2$ and $|W(t_1) \setminus \{u, v\}| \leq k$. Since uv is a cut-edge, any path connecting vertices in $V(C_1)$ and $V(C_2)$ must contain an edge uv. Both sets $V(C_1) \cap W(\hat{t})$ and $V(C_2) \cap W(\hat{t})$ are not empty but $W(\hat{t})$ does not contain u, v. This implies that $G'[W(\hat{t})]$ is not connected contradicting the fact that it is a witness set. Hence t_1 has at least two neighbors in T_1. $\diamond$

Consider a vertex t in T_1 which is adjacent with t_1. From above arguments, we know that exactly one of $V(C_1) \cap W(t)$ and $V(C_2) \cap W(t)$ is an empty set. Partition vertices in $N_{T'}(t_1)$ into two sets L and R depending on whether corresponding witness sets intersect C_1 or C_2. Formally, $L := \{t \mid t \in N_{T'}(t) \text{ and } W(t) \cap V(C_1) \neq \emptyset\}$ and $R := \{t \mid t \in N_{T'}(t) \text{ and } W(t) \cap V(C_2) \neq \emptyset\}$. Note that (L, R) is a partition of $N_{T_1}(t)$ and none of this set is empty. Let T be the graph obtained after operation SPLIT(T_1, t_1, L, R). By Lemma 3.1, T is a a tree with at most ℓ many leaves.

Hence, if there exist a set of edges F' in G' such that G/F' is a tree with at most ℓ leaves then G is $|F'|$-contractible to a tree with at most ℓ leaves. This concludes the proof of reverse direction. $\square$

We now argue that the exhaustive application of Reduction Rule 3.1 either returns a reduced instance of bounded size or we can conclude that

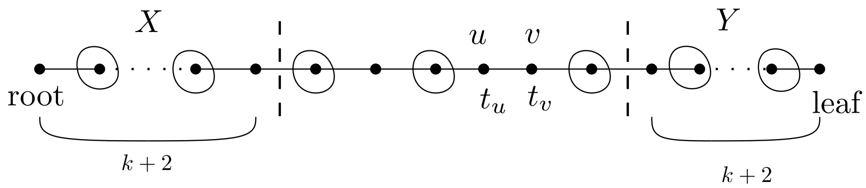

Figure 4: Parts of a longest path from root to a leaf. See Lemma 3.3.

the original instance is a NO instance.

Lemma 3.3. *Let (G, k, ℓ) be an instance of* BOUNDED TC *on which Reduction Rule 3.1 is not applicable. If (G, k, ℓ) is a* YES *instance of* BOUNDED TC*, then G has at most $\mathcal{O}(k\ell)$ vertices and $\mathcal{O}(k^2 + k\ell)$ edges.*

Proof. Let (G, k, ℓ) be a YES instance of BOUNDED TC and $F \subseteq E(G)$ be a solution such that $T = G/F$ is a tree with at most ℓ leaves. Fix an arbitrary vertex of the tree T as its root. Let $\mathcal{W}$ be a T-witness structure of G. As T is obtained using at most k edge contractions from G, $|V(G)| \leq |V(T)| + k$. Note that $|V(T)|$ is upper bounded by the number of different paths from the root to leaves times the maximum length of a path. Since the number of leaves in T is bounded by ℓ, the number of paths from the root to leaves is also bounded by ℓ.

Let $P = \{t_1, t_2, \ldots, t_q\}$ be a longest path from the root to a leaf in T. If $q \leq 2k + 5$ then $|V(T)| \leq \mathcal{O}(k\ell)$. Consider a case when $q > 2k + 5$. We argue that there does not exist i in $\{k + 2, \ldots, q - k - 2\}$ such that both $W(t_i)$ and $W(t_{i+1})$ are of cardinality one. Define two sets $X := \cup_{j \in \{1,2,\ldots,k+2\}} W(t_j)$ and $Y := \cup_{j \in \{q-(k+2),\ldots,q\}} W(t_j)$ of $V(G)$. See Figure 4. Notice that $|X|, |Y| \geq k + 2$. If there exists i in $\{k + 3, \ldots, q - k - 1\}$ such that $W(t_i) = \{u\}$ and $W(t_{i+1}) = \{v\}$ then uv is a cut-edge in G. Moreover, X, Y are in two different connected components of $G - \{uv\}$. Hence both the connected components of $G - \{uv\}$ are of size at least $k+2$. In this case, Reduction rule 3.1 is applicable. This contradicts the fact that (G, k, ℓ) is a reduced instance. Hence for i in $\{k + 2, \ldots, q - k - 2\}$, if $W(t_i)$ is a small witness set then $W(t_{i+1})$ is a big witness set. Since there are at most k big witness sets, the number of vertices in path P is at most $2k + 2(k + 2) = 4k + 4$. This implies $q \leq 4k + 4$ and $|V(T)| \leq \ell(4k + 4)$. Hence $|V(G)|$ is at most $\mathcal{O}(k\ell)$.

We now bound the number of edges in the graph G. Notice that the maximum degree of a vertex t in the tree T is bounded by ℓ. Since every edge contraction reduces the number of vertices by 1, the maximum degree of a vertex in G is at most $\ell + k$. If G/F is a tree then $G - V(F)$ is a forest. Since the size of the solution F is at most k, $|V(F)| \leq 2k$. As G is a simple graph, the number of edges of G with both of its end-points contained in $V(F)$ is at most $\mathcal{O}(k^2)$. Since $G - V(F)$ is a forest on at most $\mathcal{O}(k\ell)$ many vertices, the number of edges of G whose both endpoints are in $V(G) \setminus V(F)$ is bounded by $\mathcal{O}(k\ell)$. The number of edges which has exactly one endpoint in $V(F)$ is upper bounded by the maximum degree of G multiplied by the cardinality of $V(F)$ which is at most $\mathcal{O}(k^2 + k\ell)$. Hence the bound on the number of edges in G follows. $\square$

We are now ready to prove the main theorem of this section.

Theorem 3.1. Bounded Tree Contraction *has a kernel with* $\mathcal{O}(k\ell)$ *vertices and* $\mathcal{O}(k^2 + k\ell)$ *edges.*

Proof. Given an instance (G, k, ℓ), the algorithm applies Reduction Rule 3.1 as long as it is applicable. If the number of vertices and number of edges in the reduced instance is upper bounded by $\mathcal{O}(k\ell)$ and $\mathcal{O}(k^2 + k\ell)$, then algorithm returns reduced instance. If either of these upper bounds fails then the algorithm returns a trivial No instance.

We now argue the running time and correctness of this algorithm. To apply Reduction Rule 3.1, the algorithm needs to find a cut edge and check the number of vertices in connected components after removing that edge. This step can be performed in polynomial time. Each application of the reduction rule decreases the number of edges and thus it can be applied at most $|E(G)|$ many times. This implies that the kernelization algorithm terminates in polynomial time. Lemma 3.2 implies that Reduction Rule 3.1 is safe. Let (G', k, ℓ) be a reduced instance on which Reduction Rule 3.1 is not applicable. If G' does not have at most $\mathcal{O}(k\ell)$ vertices and $\mathcal{O}(k^2 + k\ell)$ edges, the algorithm correctly concludes that it is a No instance. The correctness of this step follows from Lemma 3.3. Otherwise, the algorithm returns a reduced instance as a kernel. $\square$

4. Kernel for Bounded Out-Tree Contraction

In this section, we design a polynomial kernel for Bounded Out-Tree Contraction. We start with some preliminary results regarding out-tree.

Digraph obtained by *subdividing* an arc of out-tree results in another out-tree. The operation of subdividing an arc uv in D is consists of deletion of the arc uv and addition of a new vertex w as an out-neighbor of u and an in-neighbor of v.

Observation 3. *Consider an out-tree T with at most ℓ leaves. Let T' be the out-tree obtained from T by one of the following operations.*

1. *subdividing an arc;*
2. *contracting an arc;*

Then, T' is an out-tree with at most ℓ leaves.

Proof. (1) Let t_1t_2 be an arc in T which is subdivided to obtain graph T'. Let t^* be newly added vertex while subdividing arc t_1t_2. Note that $d^-_T(t) = d^-_{T'}(t)$ and $d^+_T(t) = d^+_{T'}(t)$ for any vertex in t in $V(T') \setminus \{t^*\} = V(T)$. Also, $d^-_{T'}(t^*) = d^+_{T'}(t^*)$. This also implies that t^* is not a leaf in T'. Hence the number of leaves in T and T' is same. Every vertex in T' has in-degre at most one. If there exists a cycle in $G_{T'}$ which passes through t^* then the same cycle passes through t_1, t_2. This implies there exists a cycle in G_T which passes through t_1, t_2. This contradicts the fact that G_T is an underlying graph of an out-tree. Hence T' is an out-tree with at most ℓ leaves.
(2) Let t_1t_2 be an arc in T which is contracted to obtain graph T'. Let t^* be newly added vertex while contracting arc t_1t_2. Note that no vertex in T is has an arc to or from both t_1 and t_2. This implies $d^-_T(t) = d^-_{T'}(t)$ and $d^+_T(t) = d^+_{T'}(t)$ for any vertex in t in $V(T') \setminus \{t^*\} = V(T) \setminus \{t_1, t_2\}$. Moreover, by contruction, $d^-_T(t_1) = d^-_{T'}(t^*)$ and $d^+_T(t_2) = d^+_{T'}(t^*)$. Hence T' is an out-tree. Also, t^* is a leaf in T' if and only if t_2 is a leaf in T. This implies T' is an out-tree with at most ℓ leaves. □

In the following lemma, we argue that if D is k-contractible to an out-tree and there exists a long induced path then no minimal solution is incident on any vertex of this path.

Lemma 4.1. *Suppose D has a directed path $P = (v_0, v_1, \ldots, v_q, v_{q+1})$ with $q > k+1$ and $d^-(v) = d^+(v) = 1$ for each $i \in [q]$. Let F be a set of arcs of D such that $|F| \leq k$ and D/F is an out-tree with at most ℓ vertices. If F is minimal then it does not contain an edge incident on $V(P) \setminus \{v_0, v_{q+1}\}$.*

Proof. Assume that F contains at least one such arc. There are at least $k+1$ arcs with endpoints in $V(P) \setminus \{v_0, v_{q+1}\}$. Since $|F| \leq k$, there exists v_i in $\{v_0, v_1, \ldots, v_q, v_{q+1}\}$ such that $v_{i-1}v_i \in F$ and $v_iv_{i+1} \notin F$. Let $\mathcal{W}$ denote the corresponding T-witness structure of D where $T = D/F$. Now, let t and t' denote the vertices of T such that $\{v_{i-1}, v_i\} \subseteq W(t)$ and $v_{i+1} \in W(t')$. If $t = t'$ then $v_{i-1}, v_i, v_{i+1} \in W(t)$ and $v_i v_{i+1} \notin F$. As $G_D[W(t)]$ is connected, there must be a path connecting v_i, v_{i+1} in G_D which is entirely contained in $W(t)$. Any path between v_i, v_{i+1} which does not contain edges $v_i v_{i+1}$ must contain a path from v_i to v_0 and the path from v_{q+1} to v_{i+1}. It implies that $W(t)$ contains the vertices of the subpath $(v_{i+1}, \ldots, v_q, v_{q+1})$ and the vertices of the subpath $(v_0, v_1, \ldots, v_{i-1}, v_i)$. This implies $|W(t)| > k + 1$ which is a contradiction to the fact that T is obtained from D by contracting at most k edges. Hence $t \neq t'$. We now focus on $W(t)$ which, as argued above, does not contain v_{i+1}. Vertex v_i is not a cut vertex in $G_D[W(t)]$ as there is exactly one edge incident on it. This implies $G_D[W(t) \setminus v_i]$ is a connected graph. Define $\mathcal{W}' = (\mathcal{W} \setminus \{W(t)\}) \cup \{\{v_i\}\} \cup \{W(t) \setminus \{v_i\}\}$. Graph $D/(F \setminus \{v_{i-1}v_i\})$ is isomorphic to graph obtained by subdividing the arc tt' in the out-tree T. Thus, $\mathcal{W}'$ is an out-tree witness structure of D leading to the solution $F \setminus \{v_{i-1}v_i\}$ which contradicts the minimality of F. □

Note that in the above proof, we did not use the fact that T has at most ℓ leaves. Hence this claim is true for any out-tree. We mention the result explicitly in the following lemma.

Lemma 4.2. *Suppose D has a directed path $P = (v_0, v_1, \ldots, v_q, v_{q+1})$ with $q > k + 1$ and $d^-(v) = d^+(v) = 1$ for each $i \in [q]$. Let F be a set of arcs of D such that $|F| \leq k$ and D/F is an out-tree. If F is minimal then it does not contain an edge incident on $V(P) \setminus \{v_0, v_{q+1}\}$.*

We now present a kernelization algorithm. Let (D, k, ℓ) be an instance of Bounded OTC. Without loss of generality we assume that D is connected, else (D, k, ℓ) is a No instance. Recall that D is connected if its underlying undirected graph G_D is connected. The algorithm has only one reduction rule.

Reduction Rule 4.1. *Let $P = (v_0, v_1, \ldots, v_q, v_{q+1})$ be an indueced path in D with $q > k + 3$ and $d^-(v) = d^+(v) = 1$ for each $i \in [q]$. Then contract the arc $v_{q-1}v_q$ and let the resulting instance be (D', k, ℓ), where $D' = D/\{v_{q-1}v_q\}$.*

We note that unlike in the case of an undirected graph (Reduction Rule 3.1), it is not enough to find a cut arc whose remove results into two connected components of size at least $k + 1$. We might still have to contract this edge because of direction constraints. See Figure 5.

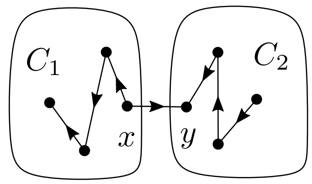

Figure 5: Different between reduction rules in case of directed and un-directed graphs.

Lemma 4.3. *Reduction rule 4.1 is safe and can be applied in polynomial time.*

Proof. We need to show that (D, k, ℓ) is a YES instance of BOUNDED OTC if and only if (D', k, ℓ) is a YES instance of BOUNDED OTC. Clearly, given D and P one can apply Reduction Rule 4.1 in polynomial time.

In the forward direction, let (D, k, ℓ) be a YES instance of BOUNDED OTC and let $F \subseteq A(D)$ such that $|F| \leq k$ and $T = D/F$ is an out-tree with at most ℓ leaves. By Observation 3, we know that $D/(F \cup \{v_{q-1}v_q\})$ is also an out-tree with at most ℓ leaves. However, $D/(F \cup \{v_{q-1}v_q\}) = (D/\{v_{q-1}v_q\})/(F \setminus \{v_{q-1}v_q\}) = D'/(F \setminus \{v_{q-1}v_q\})$. This implies that $D'/(F \setminus \{v_{q-1}v_q\})$ is an out-tree with at most ℓ leaves and $|F \setminus \{v_{q-1}v_q\}| \leq |F| \leq k$. Hence, it follows that (D', k, ℓ) is a YES instance of BOUNDED OTC.

In the reverse direction, let (D', k, ℓ) be a YES instance of BOUNDED OTC and let $F' \subseteq A(D')$ of size at most k such that $T' = D'/F'$ is an out-tree with at most ℓ leaves. Let $\mathcal{W}'$ be a T'-witness structure of D'. Let v^*_{q-1} be the vertex obtained after contracting the arc $v_{q-1}v_q$. Let P^* be the path from v_0 to v_{q+1} in graph D'. In other words, P^* is a path obtained from P by contracting edge $v_{q-1}v_q$. Since P^* is a path of size $k+2$, by Lemma 4.1, no edge in F' is incident on vertices in P^*. This implies that if $W(t^*)$ is the witness set in $\mathcal{W}'$ which contains v^*_{q-1} then $W(t^*)$ is a singleton witness set. Moreover, every vertex in $V(P) \setminus \{v_{q-1}, v_q\}$ is in singlton witness set in $\mathcal{W}'$. Let t_1, t_2 be two vertices in T' which are in-neighor and out-neighor, respectively, of t^*.

Consider a witness structure $\mathcal{W}$ obtained from $\mathcal{W}'$ by removing $\{v^*_{q-1}\}$ and adding two sets $\{v_{q-1}\}, \{v_q\}$. Formally, $\mathcal{W} = (\mathcal{W}' \setminus \{v^*_{q-1}\}) \cup \{\{v_{q-1}\}, \{v_q\}\}$. Note that $\mathcal{W}$ partitions $V(D)$ and for each $W \in \mathcal{W}$, $D[W]$ is connected. Let T be the digraph for which $\mathcal{W}$ is a T-witness structure of D. We argue that T is an out-tree with at most ℓ edges. Note that T can be obtained from T'

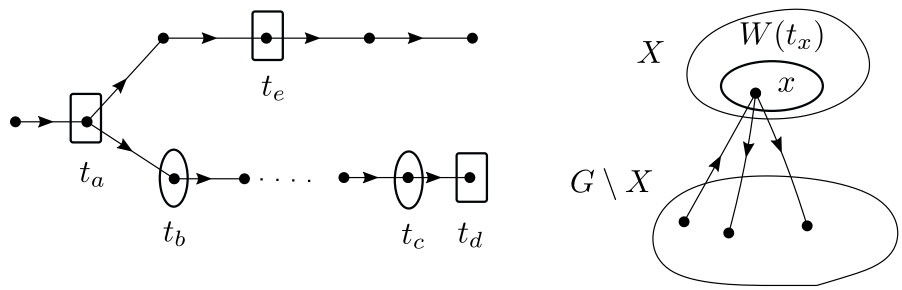

Figure 6: For left figure, please refer to Lemma 4.4. Vertices t_a, t_d are marked as they are part of $T_1 \cup T_3$. Vertices t_c, t_d are marked because they are end-points of a path. Vertex t_e marked as $W(t_e)$ is a big witness set. For figure on right, please refer to Lemma 4.5.

by subdividing edge $t^* t_2$. By Observation 3, T is an out-tree with at most ℓ leaves. This completes the proof of the lemma. $\square$

For simplicity, by (D, k, ℓ) we denote an instance of Bounded OTC on which the Reduction Rule 4.1 is not applicable.

Lemma 4.4. *Let (D, k, ℓ) be a* Yes *instance of* Bounded OTC *on which Reduction Rule 4.1 is not applicable. Then, D has at most $\mathcal{O}(k^2 + k\ell)$ vertices.*

Proof. Let (D, k, ℓ) be a Yes instance and $F \subseteq A(D)$ be a solution such that $T = D/F$ is an out-tree with at most ℓ leaves. Let $\mathcal{W}$ be a T-witness structure of a digraph D. For counting the number of vertices in D, we first count the vertices in T. Towards this, we employ a marking scheme. By M we denote the set of vertices in T that have been marked by our scheme. Let X be the set of vertices in T which corresponds to big witness sets in $\mathcal{W}$. We mark all the vertices in X. Let T_1, T_3 denote the set of vertices in T which have total degree exactly one and at least three, respectively in T. We mark all the vertices in T_1 and T_3. Note that $|T_1| \leq \ell + 1$. Here, we have $|T_1| \leq \ell + 1$, rather than $|T_1| \leq \ell$, to take into account the case when the root of T has total degree 1. Also, $|X| \leq k$ and $|T_3| \leq |T_1|$. Therefore, it follows that currently, the number of vertices in M is upper bounded by $k + 2\ell + 2$. See Figure 6.

Let $\mathcal{P}$ be the set of induced maximal (directed) paths in $T[V(T) \setminus M]$. Observe that, by viewing each path in $\mathcal{P}$ as an edge between vertices in M we get a tree on M. Thus, $|\mathcal{P}| \leq |M| - 1$. For each $P \in \mathcal{P}$, we additionally mark two of the endpoints in P. This increases the size of M by at most $2|P|$.

However, even now the size of $|M| = \mathcal{O}(k+\ell)$. Note that each of the unmarked vertices has in-degree and out-degree exactly one. Since Reduction Rule 4.1 is not applicable, therefore the length of each of the maximal paths comprising of unmarked vertices is bounded by $\mathcal{O}(k)$. But then, the number of vertices in T is bounded by $\mathcal{O}(k^2+k\ell)$. As T is obtained using at most k edge contractions from digraph D, it follows from Observation 1 that $|V(D)| \leq |V(T)| + k$. Since $|V(T)| = \mathcal{O}(k^2 + k\ell)$, this implies that $|V(D)| = \mathcal{O}(k^2 + k\ell)$. □

Lemma 4.5. *Let (D, k, ℓ) be a* YES *instance of* BOUNDED OTC *on which Reduction Rule 4.1 is not applicable. Then, D has at most $\mathcal{O}(k^2 + k\ell)$ arcs.*

Proof. Let (D, k, ℓ) be a YES instance and $F \subseteq A(D)$ be a set of edges such that $T = D/F$ is an out-tree with at most ℓ leaves. Let $\mathcal{W}$ be a T-witness structure of a digraph D. Let X be the set of vertices in D to which an edge in F is incident to. Notice that $|X| \leq 2k$. The number of arcs with both endpoints in X is bounded by $\mathcal{O}(k^2)$. Observe that the underlying undirected graph of $D - X$ is a forest with at most $\mathcal{O}(k^2 + k\ell)$ vertices. This implies the number of arcs in D that have both endpoints in $D - X$ is bounded by $\mathcal{O}(k^2 + k\ell)$. The only arcs that remain to be counted are those with one endpoint in $D-X$ and other in X. For a vertex $x \in X$, let t_x be the vertex in $V(T)$ such that $x \in W(t_x)$. Also let $\hat{N}$ be the neighbors of t_x in T. Observe that $|\hat{N}| \leq \ell + 1$. This together with Observation 1 implies that $|\cup_{t\in\hat{N}} W(t)|$ is bounded by $2k + \ell + 1$. Therefore, the maximum degree of a vertex in X is bounded by $\mathcal{O}(k + \ell)$. This implies that the number of arcs with one end point in X and other in $D - X$ is bounded by $\mathcal{O}(k^2 + k\ell)$. We have counted all types of arcs in D and hence, we conclude that the number of arcs in D is bounded by $\mathcal{O}(k^2 + k\ell)$. □

We are now ready to prove the main theorem of this section.

Theorem 4.1. BOUNDED OTC *admits a kernel of size $\mathcal{O}(k^2 + k\ell)$.*

Proof. Given an instance (D, k, ℓ), the algorithm repeatedly applies Reduction Rule 4.1, if applicable. By Lemma 4.3, we know that Reduction Rule 4.1 is safe and can be applied in polynomial time. Each application of reduction rule decreases the number of arcs and thus it can be applied only $|A(D)|$ times. If Reduction Rule 4.1 is not applicable then either the size of the instance is bounded by $\mathcal{O}(k^2 + k\ell)$, in which case we return a kernel of the desired size. Otherwise, the algorithm correctly concludes that the instance is a NO instance of BOUNDED OTC. The correctness of this step follows by Lemmas 4.4 and 4.5. □

5. Kernel for Bounded Cactus Contraction

In this section, we design a kernelization algorithm for Bounded Cactus Contraction. We start with some known properties of cactus.

Observation 4. *The following statements hold for a cactus T.*

1. $|E(T)| \leq 2|V(T)|$
2. *Every vertex of degree at least* 3 *is a cut-vertex.*

Proof. For a given cactus T, let $\mathcal{D}$ be its block decomposition.
(1) We prove this using the induction on the number of blocks in a cactus graph. Our induction hypothesis is: if number of blocks in T is strictly less than q then $|E(T)| \leq 2|V(T)|$. For the base case, consider when T has exactly one block. In this case, T is either an edge or a cycle. In either case, $|E(T)| \leq 2|V(T)|$.

Consider T which has q blocks. Let a block B corresponds to a leaf in $\mathcal{D}$. For this block, $|E(B)| \leq 2|V(B)| - 2$ as B is either an edge or a cycle. Let u be the unique cut vertex in B. Consider a cactus $T_1 = T - (V(B) \setminus \{u\})$. Since T_1 has $q-1$ blocks, by induction hypothesis, $|E(T_1)| \leq 2|V(T_1)|$.

Any edge in T is present in exactly one block. Hence $|E(T)| = |E(T_1)| + |E(B)|$. By construction, $|V(T)| = |V(T_1)| + |V(B)| - 1$ as u is counted in $V(T_1)$ and also in $V(B)$. Substituting upper bounds for $|E(T_1)|$ and $|E(B)|$, we get $|E(T)| \leq 2|V(T)|$.
(2) Consider a vertex u which has degree at least three. Since any block B is a cycle or an edge, any vertex u has at most 2 neighbors in B. Since u has a degree at least 3, u is present in at least two-block. This implies that u is a cut vertex. □

The operation of *subdividing* an edge uv results in the graph obtained by deleting uv and adding a new vertex w adjacent to both u and v.

Observation 5. *Consider a cactus T with at most ℓ leaves. Let T' be the graph obtained from T by one of the following operations.*

1. *subdividing an edge;*
2. *contracting an edge;*
3. *deleting a cut-edge uv and add two vertex disjoint path between u, v.*

Then, T' is a cactus with at most ℓ leaves.

Proof. Let $\mathcal{D}$ being the block decomposition of T with $\mathcal{B}$ being the set of block and K being the set of cut-vertices in T.

(1) Let T' be the graph obtained by subdividing an edge uv in T and w be the resulting vertex after subdivision. Since degree of w is 2 in T', any cycle which contains w must contain its neighbors u and v. Assume that T' is not a cactus then, there exists two distinct cycles C'_1, C'_2 in T' such that $E(C'_1) \cap E(C'_2) \neq \emptyset$. But then, by replacing w with the edge uv in C'_1, C'_2 (if present), we obtain cycles $\hat{C}_1$ and $\hat{C}_2$ in T with at least one common edge, contradicting that T is cactus.

Consider the case when the edge uv is a block, say B in T. In $\mathcal{D}$, by replacing B by B_1, B_2, each containing the edges uw, wv respectively, and adding w to K, we obtain a block decomposition of $\mathcal{D}'$ of T'. Since block decomposition of a connected graph is a tree, notice that $\mathcal{D}'$ can be obtained from $\mathcal{D}$ by sub-dividing an edge twice. In a tree, a sub-division of an edge does not increase the number of leaves. Hence follows that T' is a cactus with at most ℓ leaves. The remaining case is when the edge uv is not a block. Let B be a block containing the edge uv in $\mathcal{D}$. Then, by replacing B by $B \cup \{w\}$, we obtain a block decomposition of T' with the same number of leaves. This concludes the proof.

(2) Let T' be the graph obtained by contracting an edge uv in T and u^* be the resulting vertex. Suppose T' is not a cactus then there exists two distinct cycles C'_1, C'_2 in T' such that $E(C'_1) \cap E(C'_2) \neq \emptyset$. But then, by replacing w with the edge uv in C'_1, C'_2 (if present), we obtain cycles $\hat{C}_1$ and $\hat{C}_2$ in T with at least one common edge, contradicting that T is cactus.

Let B be the block containing the edge uv. Consider the case when B is just the edge uv. In this case, u, v must be in K. But then, by contracting the edges $uB, Bv \in E(\mathcal{D})$ we can obtain a block decomposition of T'. Notice that contracting an edge in a tree (block decomposition) cannot increase the number of leaves. Hence, it follows that T' is a cactus with at most ℓ leaves. The remaining case is when B contains some other vertex. Notice that if $u, v \notin K$, then by replacing B by $B' = (B \setminus \{u, v\}) \cup \{u^*\}$ in $\mathcal{D}$ we obtain a block decomposition of T', with exactly same number of leaves. If $u \in K$ and $v \notin K$, then by contracting the edge $uB \in E(\mathcal{D})$ we obtain a block decomposition of T' with the same number of leaves.

(3) Let T' be the graph obtained from T by deleting a cut-edge uv and replacing it by two vertex disjoint paths. Let C be the cycle obtained by adding these two vertex disjoint paths between u, v. Assume that T' is not a cactus then there exists two distinct cycles C'_1, C'_2 in T' such that $E(C'_1) \cap E(C'_2) \neq \emptyset$. Since u, v are cut-vertices in graph T', any cycle which is different from C, intersect with C in at most one vertex. Hence both C'_1, C'_2

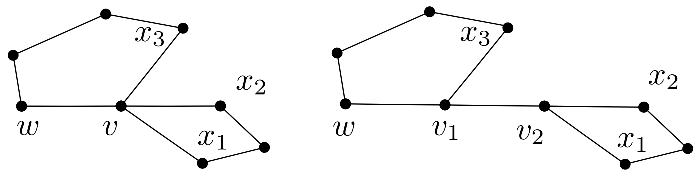

Figure 7: Operation $\textsc{Split}(T, v, L, R)$ with $L = \{w, x_3\}$ and $R = \{x_1, x_2\}$.

are distinct from C which implies C'_1 and C'_2 are two distinct cycles with at least one edge common in T which contradicts that it is cactus. Since uv is a cut-edge it is a block with u, v as cut-vertices. Let B be the block containing the edge uv, then we have $uB, Bv \in E(\mathcal{D})$. By replacing B with $V(C)$ we can obtain a block decomposition $\mathcal{D}'$ of T' with the same number of leaves. This concludes the proof. $\square$

We define operation Split on cactus in a similar way as we defined for trees with one additional condition. Consider a cactus T and one of its cut vertices, say v. Let L, R be a partition of $N(v)$ such that none of them is an empty set and there is no path between vertices of L and R in $G - \{v\}$.

Split(T, v, L, R)**:** Remove vertex v and add two vertices v_1 and v_2. Make v_1 adjacent with every vertex in L and v_2 adjacent with every vertex in R. Add edge v_1v_2. If T' is the graph obtained from T by this operation then $V(T') = (V(T) \setminus \{v\}) \cup \{v_1, v_2\}$ and $E(T') = (E(T) \setminus (\{vu \mid u \in N(v)\})) \cup \{v_1u \mid u \in L\} \cup \{v_2u \mid u \in R\} \cup \{v_1v_2\}$.

See Figure 7 for illustration. Second condition on (L, R) ensures that v_1v_2 is not a part of any cycle in new graph. The following observation, we prove that this operation on a cactus results in another cactus with the same number of leaves.

Lemma 5.1. *Let T be a cactus, v be a cut vertex of T and $N(v)$ is partitioned into two non-empty sets L and R such that there is no path between L and R in $T - v$. Let T' is the graph obtained from T after applying* $\textsc{Split}(T, v, L, R)$*. If T has at most ℓ leaves then T' is a cactus with at most ℓ leaves.*

Proof. For a cactus T and cut-vertex v, let $\mathcal{B}_v$ be set of blocks in T which contains a vertex v. Since v is a cut-vertex, there are at least two blocks in $\mathcal{B}_v$. Let L' and R' be the partition of $\mathcal{B}_v$ which vertices vertices from L and R respectively. Formally, $L' = \{B \mid x \in N_T(v) \cap B \text{ for some } x \in L\}$ and

$R' = \{B|\ y \in N_T(v) \cap B$ for some $y \in R\}$. As there is no path between vertices of L, R in $T - \{v\}$, if block B is in L' then it can not be in R'.

Let T' be the graph obtained from T by deleting a cut-vertex v and adding an edge v_1v_2 such that $N_{T'}(v_1) = L \cup \{v_2\}$ and $N_{T'}(v_2) = R \cup \{v_1\}$. Notice that v_1v_2 is an cut edge in T'. We can get a block decomposition $\mathcal{D}'$ of T' from the block decomposition $\mathcal{D}$ of T by following operations : (a) Delete v from K and adding v_1, v_2 to K. (b) Replace every B in L' by $(B \cup v_1) \setminus \{v\}$ and add edge v_1B in $E(\mathcal{D}')$. (c) Replace every B in R' by $(B \cup v_2) \setminus \{v\}$ and add edge v_2B in $E(\mathcal{D}')$ (d) Add new block $B = \{v_1, v_2\}$ and add edges v_1B and v_2B in $E(\mathcal{D}')$. It is easy to see that $\mathcal{D}'$ is a block decomposition of T'. Since every block is either an edge or a cycle, T' is a cactus. Moreover, the number of leaves in $\mathcal{D}'$ is equal to the number of leaves in $\mathcal{D}$ as newly added block is adjacent to two vertices in K. □

We make few observations regarding a cactus witness structure of a graph. Let T be a cactus obtained by contracting a set of edges in the graph G and $\mathcal{W}$ be a T-witness structure of G. Following lemma says that if the input graph contains a long induced path then we can find an edge that can be safely contracted.

Lemma 5.2. *Suppose graph G has a path $P = (u_0, u_1, \ldots, u_q)$ with $q \geq k+1$ such that all its internal vertices are of degree two. If $F \subseteq E(G)$ is a minimal set of edges of size at most k such that G/F is a cactus then F does not contain an edge in $E(P)$.*

Proof. Assume on the contrary that F contains an edge in $E(P)$. As there are at least $k+1$ edges in $E(P)$ and $|F| \leq k$, therefore there exists a vertex u_i in $V(P) \setminus \{u_0, u_q\}$ such that exactly one out of the two edges incident on it is contained in solution. Without loss of generality assume that $u_{i-1}u_i \in F$ and $u_iu_{i+1} \notin F$. Let $T = G/F$ and $\mathcal{W}$ be a T-witness structure of G. Let $t, t' \in V(T)$ such that $u_{i-1}, u_i \in W(t)$ and $u_{i+1} \in W(t')$. Consider the case when $t = t'$. F must contain all the edges in some spanning tree of $G[W(t)]$. Since $u_iu_{i+1} \notin F$, any spanning tree of $G[W(t)]$ not containing u_iu_{i+1} must contains all the edges in $E(P) \setminus \{u_iu_{i+1}\}$. But this implies $|W(t)| \geq k+2$ which is a contradiction to fact that each witness set is of size at most $k+1$. Therefore, we have that $t \neq t'$ which implies that $tt' \in E(T)$. Recall that u_i is a degree two vertex in G. This implies that u_i is not a cut-vertex in $G[W(t)]$ as there is exactly one edge incident to it in $G[W(t)]$. Therefore, $G[W(t) \setminus \{u_i\}]$ is connected. Let $\mathcal{W}' = (\mathcal{W} \setminus \{W(t)\}) \cup \{u_i\} \cup \{W(t) \setminus \{u_i\}\}$.

Observe that $\mathcal{W}'$ is a partition of $V(G)$ which is a G/F'-witness structure of G where $F' = F \setminus \{u_{i-1}u_i\}$. Notice that G/F' is the graph obtained by subdividing the edge tt' in the cactus T and by Observation 5(1) it follows that G/F' is a cactus. This contradicts the minimality of F. $\square$

We now present a kernelization algorithm. We assume that input graph is a connected otherwise we can return a trivial No instance. Exhaustive application of first reduction rule contracts an induced path of arbitrarily large length to a path of length $\mathcal{O}(k)$.

Reduction Rule 5.1. *If G has a path $P = (u_0, u_1, \ldots, u_{k+1}, u_{k+2})$ such that all of its internal vertex are of degree two, then contract $u_{k+1}u_{k+2}$. The resulting instance is (G', k, ℓ) where $G' = G/\{u_{k+1}u_{k+2}\}$.*

We prove that this reduction rule is safe using Lemma 5.2.

Lemma 5.3. *Reduction Rule 5.1 is safe.*

Proof. Let u^*_{k+1} be the resulting vertex after contraction of the edge $u_{k+1}u_{k+2}$. Given an instance (G, k, ℓ), one can find a path P which satisfies required property, in one exists, and apply reduction rule in polynomial time. We need to prove that (G, k, ℓ) is a Yes instance of Bounded CC if and only if (G', k, ℓ) is a Yes instance of Bounded CC.

Let (G, k, ℓ) be a Yes instance of Bounded CC and $F \subseteq E(G)$ such that $|F| \leq k$ and G/F is a cactus with at most ℓ leaves. From Observation 5 (2), we know that $G/(F \cup \{u_{k+1}u_{k+2}\})$ is also a cactus with at most ℓ leaves. This implies, $G/(F \cup \{u_{k+1}u_{k+2}\}) = (G/\{u_{k+1}u_{k+2}\})/(F \setminus \{u_{k+1}u_{k+2}\}) = G'/(F \setminus \{u_{k+1}u_{k+2}\})$ is a cactus with at most ℓ leaves. Also, $|F \setminus \{u_{k+1}u_{k+2}\}| \leq |F| \leq k$. Hence, it follows that (G', k, ℓ) is a Yes instance of Bounded CC.

Let (G', k, ℓ) be a Yes instance of Bounded CC and $F' \subseteq E(G')$ of size at most k be a minimal set such that $T' = G'/F'$ is a cactus with at most ℓ leaves. Let $\mathcal{W}'$ be a T'-witness structure of G'. Notice that in path $(u_0, u_1, \ldots, u_k, u^*_{k+1})$ every internal vertex is of degree exactly two. From Lemma 5.2, F' does not contain any edge incident to a vertex in $\{u_1, u_2, \ldots, u_k\}$, in particular to u_k. There exists $t'_k, t'_{k+1} \in T'$ such that $t'_k t'_{k+1} \in E(T)$ and $W(t'_k) = \{u_k\}$ and $u^*_{k+1} \in W(t'_{k+1})$. Let $\mathcal{W} = (\mathcal{W}' \setminus W(t'_{k+1})) \cup \{W(t_{k+1}), W(t_{k+2})\}$, where $W(t_{k+1}) = \{u_{k+1}\}$ and $W(t_{k+2}) = (W(t'_{k+1}) \cup \{u_{k+2}\}) \setminus \{u^*_{k+1}\}$. Since $N_{G'}(u^*_{k+1}) \setminus \{u_k\} = N_G(u_{k+2}) \setminus \{u_{k+1}\}$, $G[W(t_{k+2})]$ is connected. Let T be the graph obtained from G by contracting each witness set to a vertex. In other words, $\mathcal{W}$ is T-witness structure of graph G. Note that T can be obtained from T' by subdividing an edge $t'_k t'_{k+1}$. From Observation 5 (1) it follows that T is also a cactus with at most ℓ leaves. Since $F' \subseteq E(G)$ and it is also a spanning forest for $\mathcal{W}$, we can conclude that (G, k, ℓ) is also a Yes instance of Bounded CC. $\square$

Reduction Rule 5.1 can be applied in polynomial time. After exhaustive application of Reduction Rule 5.1 in the resulting graph G any induced path with internal vertices of degree 2 is of length at most $k+2$.

Suppose input graph G has a cut-edge uv. An optimal solution may contract one of the connected components of $G - \{uv\}$, along with an edge uv, to reduce the number of leaves in the resulting cactus. Consider the case when both connected components of $G - \{uv\}$ are *large* enough that neither of them is contained entirely in one witness set. In this case, no minimal solution contains the edge uv. The following reduction rule is based on this observation.

Reduction Rule 5.2. *If G has a cut-edge uv with C_1, C_2 being two connected components in $G - \{uv\}$ and $|V(C_1)|, |V(C_2)| \geq k+2$, then contract uv. The resulting instance is (G', k, ℓ) where $G' = G/\{uv\}$.*

Lemma 5.4. *Reduction Rule 5.2 is safe.*

Proof. Let u^* be the vertex obtained by contracting the edge uv. Given an instance (G, k, ℓ), one can find a cut-edge uv which satisfies the required property, if one exists and apply reduction rule in polynomial time. We need to prove that (G, k, ℓ) is a Yes instance of Bounded CC if and only if (G', k, ℓ) is a Yes instance of Bounded CC.

Let (G, k, ℓ) be a Yes instance of Bounded CC and $F \subseteq E(G)$ of size at most k such that G/F is a cactus T with at most ℓ leaves. As a consequence of Observation 5 (2), $G/(F \cup \{uv\})$ is also a cactus. Hence, $G/(F \cup \{uv\}) = (G/\{uv\})/(F \setminus \{uv\}) = G'/(F \setminus \{uv\})$ is a cactus with at most ℓ leaves. Also $|(F \setminus \{uv\}| \leq |F| \leq k$. This concludes that (G', k, ℓ) is a Yes instance of Bounded CC.

To prove reverse direction, let (G', k, ℓ) be a Yes instance of Bounded CC. Let F' be a set of at most k edges such that $G'/F' = T'$ is a cactus with at most ℓ leaves. We first argue that G is $(|F'|+1)$-contractible to a cactus, say T_1, which has at most ℓ leaves. Using Split operation on T_1 we argue that G is actually $|F'|$-contractible to a cactus with at most ℓ leaves.

Let $\mathcal{W}'$ be a T'-witness structure of G'. Let u^* be the vertex resulting while contracting edge uv in G to get G'. Consider vertex t^* in $V(T')$ such that u^* is in $W(t^*)$. Define set $W(t_1) := (W(t^*) \setminus \{u^*\}) \cup \{u, v\}$. Let $\mathcal{W}_1$ be the witness structure obtained from $\mathcal{W}'$ by removing $W(t^*)$ and adding $W(t_1)$. Note that $\mathcal{W}_1$ partitions $V(G)$ and for each W in $\mathcal{W}_1$, $G[W]$ is connected. Let T_1 be a graph obtained from G by contracting witness sets in $\mathcal{W}_1$. In other words, $\mathcal{W}$ is a T_1-witness structure of G. Note that T_1 can be obtained from G by contracting all edges in $F' \cup \{uv\}$. This implies T_1 can be obtained from G' by contracting all edges in F' and hence it is a cactus with at most ℓ leaves. We conclude that G is $(|F'| + 1)$-contractible to a cactus with at most ℓ leaves.

Since uv is a cut-edge in G, it is also a cut-edge in $G[W(t_1)]$. Let C_u and C_v be the connected components of $G[W(t_1)] - \{uv\}$ containing u and v, respectively. Further, let $W_u = V(C_u)$, $W_v = V(C_v)$. Consider a witness structure $\mathcal{W}$ of G obtained from $\mathcal{W}_1$ by removing $W(t_1)$ and adding W_u and W_v. Notice that $\mathcal{W}$ partitions $V(G)$ and for each W in $\mathcal{W}$, $G[W]$ is connected. Moreover, we need $|F'|$ many edges to contract all witness sets in $\mathcal{W}$. Let T be a graph obtained by contracting all witness sets in $\mathcal{W}$. In other words, $\mathcal{W}$ is a T-witness structure of G. Note that G is $|F'|$-contractible to T. The only thing which remains to prove is that T is a cactus with at most ℓ leaves. We prove this by showing that T can be obtained from T_1 by Split operation at vertex t_1. We start with the following claim.

Claim. Vertex t_1 is a cut vertex in T_1.

Proof. Each witness set in $\mathcal{W}_1$ is of size at most $k+2$ and hence $|W(t_1)| \leq k+2$. If t_1 is the only vertex in T_1, then all the vertices in $(V(C_1) \cup V(C_2)) \setminus \{u, v\}$ are in $W(t_1)$. This implies that $|W(t_1)| \geq 2k + 3$ which is a contradiction. If t_1 has unique neighbor, say $\hat{t}$, in $V(T_1)$, then $V(C_1) \cap W(\hat{t})$ and $V(C_2) \cap W(\hat{t})$ are both non empty as $|V(C_1)|$, $|V(C_2)| \geq k + 2$ and $|W(t_1) \setminus \{u, v\}| \leq k$. Since uv is a cut-edge in G, any path connecting vertices in $V(C_1)$ and $V(C_2)$ must contain an edge uv. Both sets $V(C_1) \cap W(\hat{t})$ and $V(C_2) \cap W(\hat{t})$ are not empty but $W(\hat{t})$ does not contain u, v. This implies that $G'[W(\hat{t})]$ is not connected contradicting the fact that it is a witness set. Hence, t_1 has at least two neighbors, say $\hat{t_1}, \hat{t_2}$ in T' such that $V(C_1) \cap W(\hat{t_1}) \neq \emptyset$ and $V(C_2) \cap W(\hat{t_2}) \neq \emptyset$. Assume that t_1 is not a cut vertex in T_1. There exist a path between $\hat{t}_1$ and $\hat{t}_2$ in $T_1 - \{t_1\}$. This implies there exists a path between $V(C_1)$ and $V(C_2)$ which does not contains an edge uv. This contradicts the fact that uv is an cut edge in G. Hence our assumption is wrong and t_1 is a cut vertex in T_1. $\diamond$

Consider a vertex t in T_1 which is adjacent with t_1. From above arguments, we know that exactly one of $V(C_1) \cap W(t)$ and $V(C_2) \cap W(t)$ is an empty

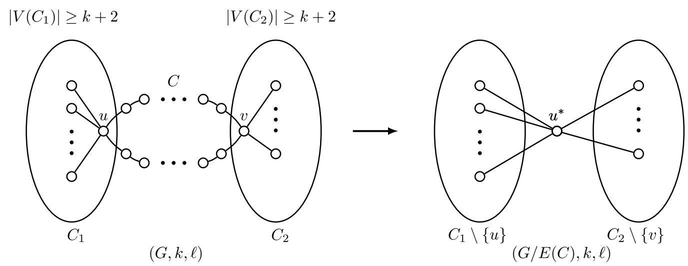

Figure 8: An illustration of Reduction Rule 5.3.

set. Partition vertices in $N_{T'}(t_1)$ into two sets L and R depending on whether corresponding witness sets intersect C_1 or C_2. Formally, $L := \{t \mid t \in N_{T'}(t)$ and $W(t) \cap V(C_1) \neq \emptyset\}$ and $R := \{t \mid t \in N_{T'}(t)$ and $W(t) \cap V(C_2) \neq \emptyset\}$. Note that (L, R) is a partition of $N_{T_1}(t)$ and none of this set is empty. Moreover, there is no path between vertices in L and R. Let T be the graph obtained after operation $\textsc{Split}(T_1, t_1, L, R)$. By Lemma 5.1, T is a cactus with at most ℓ many leaves.

Hence, if there exist a set of edges F' in G' such that G/F' is a tree with at most ℓ leaves then G is $|F'|$-contractible to a tree with at most ℓ leaves. This concludes the proof of reverse direction. □

We generalize notion of cut-edge to cycle whose removal disconnects the graph.

Definition 5.1 (Cut-Cycle)**.** *For a cycle C in graph G, C is a cut-cycle if in the block decomposition of G, there exists a block B such that $B = V(C)$ that contains exactly two cut-vertices.*

For example, in Figure 1, $\{v_3, v_4, v_5, v_6\}$ is a cut-cycle. Let C be a cut-cycle in G and u, v be the cut-vertices that it contains. Observe that $G - E(C)$ has exactly two non-trivial connected components (components with at least two vertices), one containing u and another containing v. Following reduction rule states that it is safe to contract certain cut-cycles.

Reduction Rule 5.3. *Let C be a cut-cycle in G containing cut-vertices u, v and C_1, C_2 be the non-trivial components of $G - E(C)$ such that $|V(C_1)|, |V(C_2)| \geq k+2$, then contract edges in $E(C)$. The resulting instance is (G', k, ℓ), where $G' = G/E(C)$.*

Lemma 5.5. *Reduction Rule 5.3 is safe.*

Proof. We prove the safeness of this reduction rule using an intermediate instance. Reduction Rule 5.3 can be applied in two steps. In first step, we delete all edges in $E(C)$ and add edge uv. In second step, we apply Reduction Rule 5.3 on cut edge uv. Let E_1 be set of edges in $E(C)$ which are not incident on u. Then, first step is equivalent to contracting all edges E_1 in G and renaming new vertex to v. Let $\tilde{G}$ be the graph obtained from G by contracting edges in E_1. To prove the lemma, we only need to argue that (G, k, ℓ) is an YES instance if and only if $(\tilde{G}, k, \ell)$ is an YES instance. The correctness of second step is implied by Lemma 5.4.

In the forward direction, let (G, k, ℓ) be a YES instance of BOUNDED CC and $F \subseteq E(G)$ of size at most k such that G/F is a cactus T, with at most ℓ leaves. As a consequence of Observation 5(2) it follows that $G/(F \cup E_1)$ is also a cactus with at most ℓ leaves. Hence $G/(F \cup E_1) = (G/E_1)/(F \setminus E_1) = \tilde{G}/(F \setminus E_1)$ is a cactus with at most ℓ leaves. Also, $|(F \setminus E_1)| \leq |F| \leq k$. This implies that $(\tilde{G}, k, \ell)$ is a YES instance of BOUNDED CC.

Let $(\tilde{G}, k, \ell)$ is a YES instance of BOUNDED CC. There exists $\tilde{F} \subseteq E(\tilde{G})$ such that $\tilde{G}/\tilde{F}$ is a cactus $\tilde{T}$ with at most ℓ leaves. Let $\tilde{\mathcal{W}}$ be $\tilde{T}$-witness structure of $\tilde{G}$ such that $u \in W(\tilde{t}_u)$ and $v \in W(\tilde{t}_v)$. Consider a witness structure $\mathcal{W}$ obtained from $\tilde{\mathcal{W}}$ by adding a singleton witness set for every vertex in $V(C) \setminus \{u, v\}$. Formally, $\mathcal{W} = \tilde{\mathcal{W}} \cup \{\{x\} \mid x \in V(C) \setminus \{u, v\}\}$. Notice that $\mathcal{W}$ partitions $V(G)$ and for each $W \in \mathcal{W}$, $G[W]$ is connected. Let T be the graph obtained from G by contracting witness sets in $\mathcal{W}$. In other words, $\mathcal{W}$ is T-witness structure of G. Notice that T is a graph obtained by replacing a cut-edge $\tilde{t}_u\tilde{t}_v$ in cactus $\tilde{T}$ by pair of vertex disjoint paths between vertices $\tilde{t}_u, \tilde{t}_v$. Hence, from Observation 5(3), T is a cactus with at most ℓ leaves. This concludes the proof of reverse direction.

Hence, if there exist a set of edges F' in G' such that G/F' is a tree with at most ℓ leaves then G is $|F'|$-contractible to a tree with at most ℓ leaves. $\square$

We say (G, k, ℓ) is a reduced instance of BOUNDED CC if none of the Reduction Rules 5.1, 5.2 and 5.3 are applicable.

Lemma 5.6. *Let (G, k, ℓ) be a reduced instance of* BOUNDED CC. *If (G, k, ℓ) is a* YES *instance of* BOUNDED CC, *then the number of vertices and edges in G is bounded by $\mathcal{O}(k^2 + k\ell)$.*

Proof. Suppose G is k-contractible to a cactus T with at most ℓ leaves. Let $\mathcal{W}$ be the T-witness structure of G and $\mathcal{D}$ be the block decomposition of T.

By definition of cactus, every block of T is either an edge or a cycle. We use the bound on the number of nodes in $\mathcal{D}$ and upper bound on the size of a block to bound the number of vertices in T. Let B be a block in T. If B is an edge in T, then it contains exactly two vertices. Otherwise, B contains at least 2 vertices. Let B_C, B_W are two subsets of B, defined as follows: B_C be the set of cut-vertices in T that belongs to B and B_W be the set of vertices $t \in B$ such that $|W(t)| > 1$. We bound the size of a block using the following claim.

Claim 1: $|B| \leq (k+3)|B_C \cup B_W|$.

Proof. Since the number of vertices in block B is more than 2, B induces a cycle in T. By Observation 4 and construction, for every vertex t in $B \setminus (B_C \cup B_W)$, $\mathsf{deg}_T(t) = 2$ and $|W(t)| = 1$. Consider a path $P = (t_x, t_1, t_2, \ldots, t_q, t_y)$ in T between two vertices $t_x, t_y \in B_C \cup B_W$ such that $\{t_1, t_2 \ldots, t_q\} \cap (B_C \cup B_W) = \emptyset$. Let $u_i \in W(t_i)$ for $i \in \{1, 2, \ldots, q\}$. Note that $|W(t_i)| = 1$, for all $i \in \{1, 2, \ldots, q\}$. Then, there exists a path $P' = (x, u_1, u_2, \ldots, u_q, y)$ in G such that $x \in W(t_x)$, $y \in W(t_y)$ and $\mathsf{deg}_G(u_i) = 2$ for all $i \in [q]$. Since Reduction Rule 5.1 is not applicable, therefore, $q \leq k$. Since B induces a cycle in T, there are at most $|B_C \cup B_W|$ such path and each path contains at most $k+3$ many vertices. Hence $|B| \leq (k+3)|B_C \cup B_W|$. $\diamond$

By the property of block decomposition of a graph, a node t_B corresponding to block B in $\mathcal{D}$ has degree equal to $|B_C|$. Let V_1, V_2, V_3 be the set of nodes of $\mathcal{D}$ which corresponds to a block in T and are of degree at most 1, degree 2 and degree at least 3 respectively. Since $\mathcal{D}$ has at most ℓ leaves, $|V_1| \leq \ell$ which in turn implies that $|V_3| \leq \ell$. From Proposition 1, it follows that the number of cut-vertices present in blocks with at least 3 cut-vertices is bounded by the following.

$$\sum_{t_B \in V_3} |B_C| \leq 3\ell \tag{1}$$

Note that the number of vertices in T corresponds to big witness set is at most k therefore we have the following inequality.

$$\sum_{t_B \in V_1 \cup V_2 \cup V_3} |B_W| \leq k \tag{2}$$

We fix an arbitrary vertex as the root of $\mathcal{D}$ (preferable vertex of degree at least 2). For counting purposes, we apply the following marking scheme to the nodes in $\mathcal{D}$. We start by marking all the leaves in $\mathcal{D}$. For a leaf t_B, keep marking the nodes on the path from the leaf to the root of that tree until the total number of vertices in T from the marked blocks is at least $k+2$. We say these marked vertices are *close* to the leaf t_B. Also, mark all the nodes t_B in $\mathcal{D}$ for which B_W is not empty. This completes the marking procedure. For leaf node t_B, let t_{B^*} be the last node marked by marking scheme to ensure that we have covered at least $k+2$ many vertices of T. Hence there are at most $k+1+|B^*|$ many vertices marked for the leaf t_B. Let $L' = \{t_{B^*} \mid t_B \in V_1\}$, *i.e.* the set of all the nodes which were the last marked node corresponding to some leaf. Notice that $|L'| \leq |V_1|$. Consider the subgraph $\mathcal{D}'$ of $\mathcal{D}$ induced on the vertices in $V_1 \cup L'$ and the cut-vertices their corresponding block contains. Note that in a block decomposition no two cut-vertices or two vertices corresponding to blocks are adjacent. This implies that the number number of vertices in $\mathcal{D}'$ is bounded by $\mathcal{O}(\ell)$. This helps us in establishing the following.

$$\sum_{t_{B^*} \in L'} |B^*_C| = \sum_{t_{B^*} \in L'} \mathsf{deg}_{\mathcal{T}}(t_{B^*}) \in \mathcal{O}(\ell)$$

Using the above relation, Claim 1 and Equation 2, we have the following.

$$\sum_{t_{B^*} \in L'} |B^*| \leq \sum_{t_B \in L'} (|B^*_C| + |B^*_W|)(k+2) \in \mathcal{O}(k^2 + k\ell)$$

Hence the total number of marked vertices which are close to leaf nodes are,

$$\sum_{t_B \in V_1} ((k+1) + |B^*|) \leq \sum_{t_B \in V_1} (k+1) + \sum_{t_{B^*} \in L'} |B^*| \in \mathcal{O}(k^2 + k\ell)$$

Let V_M be set of nodes t_B which are marked because B_W is not empty. By Equation 2, $|V_M| \leq k$. For $t_B \in V_M \cap (V_1 \cup V_2)$, $|B_C| \leq 2$ which implies $\sum_{t_B \in V_M \cap (V_1 \cup V_2)} |B_C| \leq 2k$.

$$\sum_{t_B \in V_M \cap (V_1 \cup V_2)} |B| \leq \sum_{t_B \in V_M \cap (V_1 \cup V_2)} (|B_C| + |B_W|)(k+2) \in \mathcal{O}(k^2)$$

For $t_B \in V_3 \cap V_M$, we use Equation 1 to obtain following bound.

$$\sum_{t_B \in V_3 \cap V_M} |B| \leq \sum_{t_B \in V_3 \cap V_M} (|B_C| + |B_W|)(k+2) \in \mathcal{O}(k^2 + k\ell)$$

We now count the number of vertices in blocks corresponding to unmarked nodes. We first argue that every unmarked node, associated block contains at least three cut-vertices. In other words, all the nodes in V_1, V_2 have been marked.

Claim 2: If t_B is not marked by above marking scheme, then $t_B \in V_3$.
Proof. We prove this by contradiction. Since all the nodes in V_1 are marked, assume that there exists unmarked node t_B in V_2 such that $|B_W| = 0$. Since B contains exactly two cut-vertices, $T - E(B)$ has exactly two non-trivial connected components, say T_1, T_2. Notice that each T_1, T_2 contains marked vertices corresponding to at least one leaf node and hence $|V(T_1)|, |V(T_2)| \geq k+2$. Since B does not contain any vertex t such that $|W(t)| > 1$, vertex set $X = \bigcup_{t \in B} W(t)$ is either a cut-edge or a cut-cycle in graph G. Moreover, $G - E(X)$ has two non-trivial connected components C_1, C_2 such that $V(C_1) = \bigcup_{t \in V(T_1)} W(t)$ and $V(C_2) = \bigcup_{t \in V(T_2)} W(t)$ which implies $|V(C_1)|, |V(C_2)| \geq k+2$. But in this case, Reduction Rule 5.2 or 5.3 is applicable on the instance. This contradicts that (G, k, ℓ) is a reduced instance. $\diamond$

Let U be the set of nodes which are unmarked. By Claim 2, $U \subseteq V_3$. By Equation 1 and using the fact that $|B_W| = 0$ for $t_B \in U$,

$$\sum_{t_B \in U} |B| = \sum_{t_B \in U} (k+3)|B_C| = (k+3) \cdot \sum_{t_B \in U} |B_C| \in \mathcal{O}(k\ell)$$

Combining all these upper bounds, we get $|V(T)| \leq \mathcal{O}(k^2 + k\ell)$. Since T is obtained from G with at most k edge contractions, it follows that $|V(G)| \leq |V(T)| + k$. This implies the desired bound on the vertices of the input graph. We now bound the number of edges in G. Notice that the maximum degree of a node in $\mathcal{D}$ is at most ℓ as the number of leaves in $\mathcal{D}$ is at most ℓ. This implies that any cut-vertex in T can be part of at most ℓ blocks. Since every vertex can be adjacent to at most 2 vertices in a block, the maximum degree of a vertex t in cactus T is at most 2ℓ. Every edge contraction can reduce the number of vertices by 1 hence the maximum degree of a vertex in G is at most $2\ell + k$. If G/F is a cactus then each component in $G - V(F)$ is also a cactus. Since the size of solution F is at most k, $|V(F)| \leq 2k$. As G is a simple graph, the number of edges of G with both of its endpoints in $V(F)$ is at most $\mathcal{O}(k^2)$. $G - V(F)$ is cactus on at most $\mathcal{O}(k^2 + k\ell)$ many vertices and hence by Observation 4, the number of edges of G whose both endpoints are in $V(G) \setminus V(F)$ is at most $\mathcal{O}(k^2 + k\ell)$. The number of edges which has exactly one endpoint in $V(F)$ is upper bounded by the maximum degree of G multiplied by the cardinality of F which is at most $\mathcal{O}(k^2 + k\ell)$. Hence the bound on the number of edges in G follows. $\square$

We are now ready to prove the main theorem of this section.

Theorem 5.1. Bounded Cactus Contraction *admits a kernel of size* $\mathcal{O}(k^2 + k\ell)$.

Proof. Given an instance (G, k, ℓ) of Bounded CC the kernelization algorithm exhaustively applies Reduction Rules 5.1, 5.2 or 5.2. If the number of vertices and edges in reduced graph is not upper bounded by $\mathcal{O}(k^2 + k\ell)$ then it returns a trivial no instance.

By Lemma 5.3; 5.4; and 5.5, these reduction rules are safe and can be applied in polynomial time. Each application of reduction rule decreases the number of edges thus it can be applied only $|E(G)|$ times. If none of the reduction rules are applicable then either the size of the instance is bounded by $\mathcal{O}(k^2 + k\ell)$, in which case we return a kernel of the desired size. Otherwise, the algorithm correctly concludes that the instance is a No instance of Bounded CC. Lemma 5.6 proves the correctness of this step of the algorithm. □

6. Kernel Lower Bounds

In this section we show that the kernelization algorithm presented in Sections 3, 4, and 5 are optimal assuming $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$. We mention one problem for which compression lower bound is known to be optimal under standard complexity assumptions. The problem Dominating Set takes as an input a graph and an integer k, and the goal is to decide whether the input graph contains a dominating set of size at most k. Any instance can be encoded with $\mathcal{O}(n^2)$ bits where n is the number of vertices in the input graph. Jansen and Pieterse proved that Dominating Set does not admit a compression of bit-size $\mathcal{O}(n^{2-\epsilon})$, for any $\epsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ [28]. We use this result to obtain compression lower bound for another problem which is more useful in reduction. The input instance for Red-Blue Dominating Set (RBDS) is a bipartite graph G with bi-partition (R, B) and an integer t. The question is whether R has a subset of at most t vertices that dominates B.

Proposition 2. Red-Blue Dominating Set *does not admit a polynomial compression of bit size* $\mathcal{O}(n^{2-\epsilon})$*, for any* $\epsilon > 0$ *unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$*. Here,* n *is the number of vertices in the input graph.*

Proof. Assuming a contradiction, suppose RBDS admits a compression into $L \subseteq \Sigma^*$ with bit-size in $\mathcal{O}(n^{2-\epsilon})$ for some $\epsilon > 0$, where n is the number

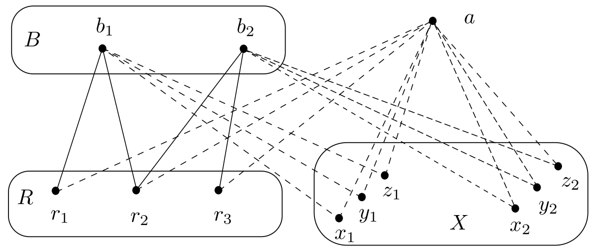

Figure 9: Kernel lower bound for Bounded TC.

of vertices in the input graph for RBDS. This implies that there exists an algorithm $\mathcal{A}$ which takes an instance $I = (G, R, B, k)$ of RBDS and in time $n^{\mathcal{O}(1)}$ returns an equivalent instance I' of L with $|I'| \in \mathcal{O}(n^{2-\epsilon})$.

Let (G, k) be an instance of Dominating Set and $n = |V(G)|$. We construct as instance (G', R, B, k') of RBDS as the following. For each $v \in V(G)$, we add vertices v_R and v_B to R and B, respectively. Further, for each $v_R \in R$ we make it adjacent to the corresponding copies in B of vertices in $N_G[v]$. Finally, we set $k' = k$. It is easy to see that (G, k) is a Yes instance of Dominating Set if and only if (G', R, B, k') is a Yes instance of RBDS. Furthermore, the reduction takes polynomial time and $|V(G')| \in \mathcal{O}(n)$. But then Dominating Set admits a compression into Π with bit-size $\mathcal{O}(n^{2-\epsilon})$, a contradiction. □

6.1. *Kernel Lower Bound for* Bounded Tree Contraction

To prove this, we present a parameter preserving reduction which given an instance (G, R, B, k) of Red Blue Dominating Set (RBDS), creates an instance (G', k', ℓ') of Bounded TC.

Reduction. Let (G, R, B, k) be an instance of RBDS. We construct graph G' in the following way. See Figure 9. Initialize $V(G') = V(G)$ and $E(G') = \{br \mid b \in B, r \in R \text{ and } br \in E(G)\}$. Add a vertex a in $V(G')$ and for every vertex r in R, add an edge ar to $E(G')$. For every vertex b_i in B, add three new vertices x_i, y_i, z_i to $V(G')$ and edges b_ix_i, b_iy_i, b_iz_i to $E(G')$ [1]. Define set

[1] It is sufficient to add two vertices for each b_i in B. We add three vertices so that this proof can be re-used to prove similar results in case of Bounded Cactus Contraction problem

$X := \{x_i, y_i, z_i \mid b_i \in B\}$. For every vertex x in X, add an edge ax to $E(G')$. Set $k' = |B| + k$ and $\ell' = |R| + 3|B| - k$.

In the following lemma, we prove some structural properties of a solution for (G', k', ℓ').

Lemma 6.1. *Let (G', k', ℓ') be a* YES *instance of* BOUNDED TC. *There exists a solution $F^* \subseteq E(G')$ of size at most k' such that for each b_i in B one of the following holds.*

1. *b_i is in $W(t_a)$ or*
2. *x_i, y_i, z_i are in $W(t_a)$.*

Here, $W(t_a)$ is the witness set containing a in (G'/F^*)-witness structure of G'.

Proof. Let F be a set of edges of size at most k in G' such that G'/F is a tree with at most ℓ leaves. Let $\mathcal{W}$ be a T-witness structure of G' where $T = G'/F$. Let t_a be the vertex in $V(T)$ such that $W(t_a)$ contains vertex a. For a vertex b_i in B, if b_i is in $W(t_a)$ then the lemma holds. Consider a case when b_i is not in $W(t_a)$. There exists a vertex t_b, different from t_a, such that b_i is contained in $W(t_b)$. Similarly, consider vertices t_x, t_y and t_z such that x_i, y_i and z_i are contained in $W(t_x), W(t_y)$ and $W(t_z)$, respectively.

If neither of t_a or t_b is contained in set $\{t_x, t_y, t_z\}$, then no two vertices in $\{t_x, t_y, t_z\}$ can be same as only neighbors of x_i, y_i, z_i are a and b_i, and a witness set needs to be connected. But then, by construction, $T[\{t_a, t_x, t_y, t_z, t_b\}]$ is a cycle, contradicting the fact that T is a tree. Therefore, at least one of $\{t_x, t_y, t_z\}$ is same as t_a or t_b. Without loss of generality, let $t_x \in \{t_a, t_b\}$. This implies there is an edge $t_a t_b$ is in T. If t_y or t_z is not equal to t_a or t_b then there exist a cycle contradicting that T is a tree. Suppose, all t_x, t_y, t_z are same as t_a, then the second condition of the lemma is satisfied. Consider a case when at least one of t_x, t_y, t_z, say t_x, is not same as t_a, that is $t_x = t_b$. The only edges incident to x_i in G' are ax_i and bx_i. This implies that $bx_i \in F$ and $W(t'_b) = W(t_b) \setminus \{x_i\}$ is connected. Since $ax_i \in E(G')$, set $W(t'_a) = W(t_a) \cup \{x_i\}$ is connected. Thus, replacing $W(t_b)$ by $W(t'_b)$ and $W(t_a)$ by $W(t'_a)$ in $\mathcal{W}$ yields another T-witness structure of G'. Furthermore, the spanning forest of the new witness structure, $F' = (F \setminus \{bx_i\}) \cup \{ax_i\}$ which has same cardinality as that of F. A similar swap can be carried out if $t_y = t_b$ or $t_z = t_b$. This concludes the proof. $\square$

In the following lemma, we argue that the reduction is safe.

Lemma 6.2. *(G, R, B, k) is a* Yes *instance of* RBDS *if and only if (G', k', ℓ') is a* Yes *instance of* Bounded TC*.*

Proof. Let (G, R, B, k) be a Yes instance of RBDS and S be a subset of R of size k such that S dominates every vertex in B. If S contains less than k vertices, then we take any of its superset of size exactly k. For each vertex b in B, we fix a vertex r in S such that b is neighbor of r in G. If there are multiple options for selecting r then we arbitrarily choose one of them. Let $F = \{br \mid b \in B \text{ and } br \in E(G)\} \cup \{ar \mid r \in S\}$. Note that $|F| = |B| + k = k'$ and $G'[V(F)]$ is connected. Let T be the graph obtained from G' by contracting F. Let $\mathcal{W}$ be a T-witness structure of G'. Consider a vertex t_a such that a is in $W(t_a)$. Since all the edges in F are contracted to one vertex, set $S \cup B$ is also contained in $W(t_a)$. By construction, $R \cup X$ is an independent set in G'. No vertex in $(R \cup X) \setminus S$ is incident on edge which has been contracted. In other words, these vertices form singleton witness sets in $\mathcal{W}$. Since $R \cup X$ is an independent set in G', it follows that set $T_{RX} = \{t_v \mid v \in (R \cup X) \setminus S\}$ is an independent set in T of size $|R| + 3|B| - k = \ell'$. Moreover, for all v in X', $av \in E(T)$. Therefore, T is a star (which is a tree) with ℓ' leaves. This implies that F is a solution to (G', k', ℓ').

In the reverse direction, let (G', k', ℓ') be a Yes instance of Bounded TC. By Lemma 6.1, there exists a solution F^* of size at most k' such that for every b_i in B, either b_i is in $W(t_a)$ or all of x_i, y_i, z_i are in $W(t_a)$. Here, $\mathcal{W}$ is the G'/F^*-witness structure of G' and t_a in $V(G'/F^*)$ such that vertex a is contained in witness set $W(t_a)$ in $\mathcal{W}$.

We partition vertices of B into two parts depending on whether they belong to $W(t_a)$ or not. Define $B_g = \{b_i \in B \mid b_i \in W(t_a)\}$. Let $R_a = R \cap W(t_a)$. Partition B_g into B_1 and B_2, depending on whether or not they have a neighbor in R_a. Formally, $B_1 = \{b_i \in B_g \mid N(b_i) \cap R_a \neq \emptyset\}$ and $B_2 = B_g \setminus B_1$. For a vertex b_i in B_2 at least one of x_i, y_i, z_i is present in $W(t_a)$ as there is no edge between b_i and a. Note that, by construction, x_i, y_i, z_i is not adjacent with b_j for $i \neq j$. This implies there exists a separate vertex for each b_i in B_2 which provides connectivity between a and b_i. Let X_2^B be set of vertices in $X \cap W(t_a)$ which provides adjacency between a and b_i for some b_i in B_2. For every b_i which is in $B \setminus B_g$, by Lemma 6.1, x_i, y_i, z_i are present in $W(t_a)$.

We can partition $W(t_a) \setminus \{a\}$ into following four parts: vertices in B (captured by B_g); vertices in R (captured by R_a); vertices in X which are present because corresponding b_i is not present (captured by $B \setminus B_g$); and vertices in X which are present because they are needed to provide connectivity between b_i and a (captured by X_2^B). This implies $|B_g| + 3|B \setminus B_g| + |R_a| + |X_2^B| + |\{a\}| \leq |W(t_a)|$.

We construct a solution S for RBDS by taking vertices in R_a and two more sets S_g and S_w. Informally, S_g dominates vertices in B_2 and S_w dominates vertices in $B \setminus B_g$. We construct S_g in following way. For every vertex b_i in B_2, arbitrary pick one of its neighbor in R and add it to S_g. Note that $|S_g| \leq |X_2^B|$. We create another set S_w in the following way. Initialize S_w to an empty set. For each b in $B \setminus B_g$, we add an arbitrary neighbor of b in R to S_w. This implies $|S_w| \leq |B \setminus B_g|$. As cardinality of F^* is at most $k + |B|$, size of $W(t_a)$ is at most $|W(t_a)| \leq k + |B| + 1$.

Putting all inequalities together, we get $|R_a| + |S_g| + |S_w| \leq k$ and every vertex in B is dominated some vertex in $R_a \cup S_g \cup S_w$. This concludes the proof. $\square$

We are now in a position to present a kernel lower bound for Bounded Tree Contraction.

Theorem 6.1. Bounded Tree Contraction *does not admit a compression of size $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$, for any $\epsilon > 0$.*

Proof. Assuming a contradiction, suppose Bounded TC admits a compression into $\Pi \subseteq \Sigma^*$ with bitsize in $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$, for some $\epsilon > 0$. This implies that there exists an algorithm $\mathcal{A}$ which takes an instance $I = (G, k, \ell)$ of Bounded TC and in polynomial time returns an equivalent instance I' of Π with $|I'| \in \mathcal{O}((k^2 + k\ell)^{1-\epsilon})$.

Let (G, R, B, k) be an instance of RBDS, where G is a graph on n vertices. Using the reduction described, we create an instance (G, k', ℓ') of Bounded TC with $|V(G'_D)| \in \mathcal{O}(n)$, $|E(G'_D)| \in \mathcal{O}(n^2)$, $k' = k \leq |R| \in \mathcal{O}(n)$ and $\ell' = |B| + k \in \mathcal{O}(n)$. By Lemma 6.2, (G, R, B, k) is a Yes instance of RBDS if and only if (G', k', ℓ') is a Yes instance of Bounded TC. On the instance (G, k', ℓ') we run the algorithm $\mathcal{A}$ to obtain an instance I of Π such that $|I| \in \mathcal{O}((k'^2 + k'\ell')^{1-\epsilon})$. But then we have obtained a compression of size $\mathcal{O}(n^{2-\epsilon})$ for RBDS, contradicting Proposition 2. $\square$

Corollary 6.1. Bounded Tree Contraction *does not admit a kernel of size $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$, for any $\epsilon > 0$.*

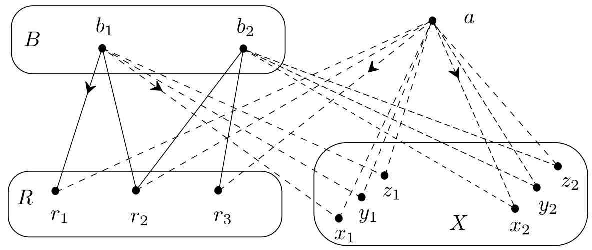

Figure 10: Kernel lower bound for Bounded OTC. For the sake of clarity, figure does not show directions for all arcs.

6.2. *Kernel Lower Bound for* Bounded Out-Tree Contraction

In this sub-section we present a parameter preserving reduction from given an instance (G, R, B, k) of RBDS to an instance (D', k', ℓ') of Bounded Out-Tree Contraction. This reduction is the same as the one presented in the previous sub-section with directions added to edges. For the sake of completeness, we present the entire proofs.

Reduction. Let (G, R, B, k) be an instance of RBDS. We construct graph G' in the following way. See Figure 10. Initialize $V(G') = V(G)$ and $E(G') = \{br \mid b \in B, r \in R \text{ and } br \in E(G)\}$. Add a vertex a in $V(G')$ and for every vertex r in R, add an edge ar to $E(G')$. For every vertex b_i in B, add three new vertices x_i, y_i, z_i to $V(G')$ and arcs b_ix_i, b_iy_i, b_iz_i to $E(G')$. Define set $X := \{x_i, y_i, z_i \mid b_i \in B\}$. We construct diagraph D' from G' by adding directions to edegs. For every vertex x in X, add an edge ax to $E(G')$. For every edge incident on a, add direction from a to other end point. Similarly, for any end incident on vertices in B, add direction from vertex in B to other end point. Set $k' = |B| + k$ and $\ell' = |R| + 3|B| - k$.

In the following lemma, we prove some structural properties of a solution to instance (D', k', ℓ').

Lemma 6.3. *Let (D', k', ℓ') be a* Yes *instance of* Bounded Out-Tree Contraction*. There exists a solution $F^* \subseteq E(D')$ of size at most k' such that for each b_i in B one of the following holds.*

1. *b_i is in $W(t_a)$ or*
2. *x_i, y_i, z_i are in $W(t_a)$.*

Here, $W(t_a)$ is the witness set containing a in (D'/F^*)-witness structure of D'.

Proof. Let F be a set of arcs of size at most k in D' such that D'/F is an out-tree with at most ℓ leaves. Let $\mathcal{W}$ be a T-witness structure of D' where $T = D'/F$. Recall that T_G denotes the underlying undirected graph of T. Since T is an out-tree, T_G is a tree. Let t_a be the vertex in $V(T)$ such that $W(t_a)$ contains a. For a vertex b_i in B, if b_i is in $W(t_a)$ then the lemma holds. Consider a case when b_i is not in $W(t_a)$. There exists a vertex t_b, different from t_a, such that b_i is contained in $W(t_b)$. Similarly, consider vertices t_x, t_y and t_z such that x_i, y_i and z_i are contained in $W(t_x), W(t_y)$ and $W(t_z)$, respectively.

If neither of t_a or t_b is contained in set $\{t_x, t_y, t_z\}$, then no two of $\{t_x, t_y, t_z\}$ can be same as only neighbors of x_i, y_i, z_i are a and b_i, and by definition, a witness set needs to be connected. But then, by construction, $T_G[\{t_a, t_x, t_y, t_z, t_b\}]$ is a cycle, contradicting the fact that T_G is a tree. Therefore, at least one of $\{t_x, t_y, t_z\}$ is same as t_a or t_b. Without loss of generality, let $t_s \in \{t_a, t_b\}$. This implies there is an edge $t_a t_b$ in T_G. If t_y or t_z is not equal to t_a or t_b then there exist a cycle contradicting that T_G is a tree. Suppose, all t_x, t_y, t_z are same as t_a, then the second condition of the lemma is satisfied. Consider a case when at least one of t_x, t_y, t_z, say t_x, is not same as t_a, which implies $t_x = t_b$. By construction, the only arcs incident to x_i in D' are ax_i and bx_i. This implies that $bx_i \in F$ and $W(t'_b) = W(t_b) \setminus \{x_i\}$ is connected. Since $ax_i \in A(D')$, set $W(t'_a) = W(t_a) \cup \{x_i\}$ is connected. Thus, replacing $W(t_b)$ by $W(t'_b)$ and $W(t_a)$ by $W(t'_a)$ in $\mathcal{W}$ yields another T-witness structure of D'. Furthermore, the spanning forest of the new witness structure, $F' = (F \setminus \{bx_i\}) \cup \{ax_i\}$ has same cardinality as that of F. A similar swap can be carried out if $t_y = t_b$ or $t_z = t_b$. This concludes the proof. □

In the following lemma, we argue that the reduction is safe.

Lemma 6.4. *(G, R, B, k) is a* Yes *instance of* RBDS *if and only if (D', k', ℓ') is a* Yes *instance of* Bounded OTC*.*

Proof. Let (G, R, B, k) be a Yes instance of RBDS and S be a subset of R of size k such that S dominates every vertex in B. If S contains less than k vertices, then we take any of its superset of size exactly k. For each vertex b in B, we fix a vertex r in S such that b is neighbor of r in G. If there are multiple options for selecting r then we arbitrarily choose one of them. Let $F = \{br \mid b \in B \text{ and } br \in E(G)\} \cup \{ar \mid r \in S\}$. Note that $|F| = |B| + k = k'$ and $D'[V(F)]$ is connected. Let T be the digraph obtained from D' by contracting edges in F. Let $\mathcal{W}$ be a T-witness structure of D'.

Consider a vertex t_a such that a is in $W(t_a)$. Since all the edges in F are contracted to one vertex, set $S \cup B$ is also contained in $W(t_a)$. Recall that $R \cup X$ is an independent set in $G_{D'}$. No vertex in $(R \cup X) \setminus S$ is incident on edge which has been contracted. In other words, these vertices form singleton witness sets in $\mathcal{W}$. Since $R \cup X$ is an independent set in $G_{D'}$, it follows that set $T_{RS} = \{t_v \mid v \in (R \cup X) \setminus S\}$ is an independent set in G_T of size $|R| + 3|B| - k = \ell'$. Moreover, for all v in X', arc av is present in $A(T)$. Therefore, T is a out-tree with ℓ' leaves. This implies that F is a solution to (D', k', ℓ').

In the reverse direction, let (D', k', ℓ') be a Yes instance of Bounded Out-Tree Contraction. By Lemma 6.1, there exists a solution F^* of size at most k' such that for every b_i in B, either b_i is in $W(t_a)$ or all of x_i, y_i, z_i are in $W(t_a)$. Here, $\mathcal{W}$ is the D'/F^*-witness structure of D' and t_a in $V(D'/F^*)$ such that vertex a is contained in witness set $W(t_a)$ in $\mathcal{W}$.

We partition vertices of B into two parts depending on whether they belong to $W(t_a)$ or not. Define set $B_g = \{b_i \in B \mid b_i \in W(t_a)\}$. Let $R_a = R \cap W(t_a)$. Partition B_g into B_1 and B_2, depending on whether or not they have a neighbor in R_a. Formally, $B_1 = \{b_i \in B_g \mid N(b_i) \cap R_a \neq \emptyset\}$ and $B_2 = B_g \setminus B_1$. For a vertex b_i in B_2 at least one of x_i, y_i, z_i is present in $W(t_a)$ as there is no arc between b_i and a. Note that, by construction, x_i, y_i, z_i is not adjacent with b_j for $i \neq j$. This implies there exists a separate vertex for each b_i in B_2 which provides connectivity between a and b_i. Let X_2^B be set of vertices in $X \cap W(t_a)$ which provides adjacency between a and b_i for some b_i in B_2. For every b_i which is in $B \setminus B_g$, by Lemma 6.3, x_i, y_i, z_i are present in $W(t_a)$.

We can partition $W(t_a) \setminus \{a\}$ into following four parts: vertices in B (captured by B_g); vertices in R (captured by R_a); vertices in X which are present because corresponding b_i is not present (captured by $B \setminus B_g$); and vertices in X which are present because they are needed to provide connectivity between b_i and a (captured by X_2^B). This implies $|B_g| + 3|B \setminus B_g| + |R_a| + |X_2^B| + |\{a\}| \leq |W(t_a)|$.

We construct a solution S for RBDS by taking vertices in R_a and two more sets S_g and S_w. Informally, S_g dominates vertices in B_2 and S_w dominates vertices in $B \setminus B_g$. We construct S_g in following way. For every vertex b_i in B_2, arbitrary pick one of its neighbor in R and add it to S_g. Note that $|S_g| \leq |X_2^B|$. We create another set S_w in the following way. Initialize S_w to an empty set. For each b in $B \setminus B_g$, we add an arbitrary neighbor of b in R to S_w. This implies $|S_w| \leq |B \setminus B_g|$. As cardinality of F^* is at most $k + |B|$, size of $W(t_a)$ is at most $|W(t_a)| \leq k + |B| + 1$.

Putting all inequalities together, we get $|R_a| + |S_g| + |S_w| \leq k$ and every vertex in B is dominated some vertex in $R_a \cup S_g \cup S_w$. This concludes the proof. □

We now argue that the kernel presented for Bounded OTC is optimal.

Theorem 6.2. Bounded Out-Tree Contraction *does not admit a compression of size* $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$*, for any* $\epsilon > 0$.

Proof. Assuming a contradiction, suppose Bounded Out-Tree Contraction admits a compression into $\Pi \subseteq \Sigma^*$ with bitsize in $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$, for some $\epsilon > 0$. This implies that there exists an algorithm $\mathcal{A}$ which takes an instance $I = (G, k, \ell)$ of Bounded Out-Tree Contraction and in polynomial time returns an equivalent instance I' of Π with $|I'| \in \mathcal{O}((k^2 + k\ell)^{1-\epsilon})$.

Let (G, R, B, k) be an instance of RBDS, where G is a graph on n vertices. Using the reduction described, we create an instance (G, k', ℓ') of Bounded Out-Tree Contraction with $|V(G'_D)| \in \mathcal{O}(n)$, $|E(G'_D)| \in \mathcal{O}(n^2)$, $k' = k \leq |R| \in \mathcal{O}(n)$ and $\ell' = |B| + k \in \mathcal{O}(n)$. By Lemma 6.4, (G, R, B, k) is a Yes instance of RBDS if and only if (D', k', ℓ') is a Yes instance of Bounded OTC. On the instance (G, k', ℓ') we run the algorithm $\mathcal{A}$ to obtain an instance I of Π such that $|I| \in \mathcal{O}((k'^2 + k'\ell')^{1-\epsilon})$. But then we have obtained a compression of size $\mathcal{O}(n^{2-\epsilon})$ for RBDS, contradicting Proposition 2. □

Corollary 6.2. Bounded Out-Tree Contraction *does not admit a kernel of size* $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$*, for any* $\epsilon > 0$.

6.3. Kernel Lower Bound for Bounded Cactus Contraction

In this section, we present a parameter preserving reduction from a given instance (G, R, B, k) of RBDS to an instance (G', k', ℓ') of Bounded Cactus Contraction.

Reduction. Let (G, R, B, k) be an instance of RBDS. We construct graph G' in the following way. See Figure 9. Initialize $V(G') = V(G)$ and $E(G') = \{br \mid b \in B, r \in R \text{ and } br \in E(G)\}$. Add a vertex a in $V(G')$ and for every vertex r in R, add an edge ar to $E(G')$. For every vertex b_i in B, add three new vertices x_i, y_i, z_i to $V(G')$ and edges b_ix_i, b_iy_i, b_iz_i to $E(G')$. Define set

$X := \{x_i, y_i, z_i \mid b_i \in B\}$. For every vertex x in X, add an edge ax to $E(G')$. Set $k' = |B| + k$ and $\ell' = |R| + 3|B| - k$.

Following the same spirit of proof as described in Section 3, we prove the following lemmas. Note that lemma implies if b_i is not present in $W(t_a)$ then at least two vertices in $\{x_i, y_i, z_i\}$ are present in $W(t_a)$ unlike in case of Bounded TC where all three were present.

Lemma 6.5. *Let (G', k', ℓ') be a* Yes *instance of* Bounded CC. *There exists a solution $F^* \subseteq E(G')$ of size at most k' such that for each $b_i \in B$ one of the following holds.*

- *b_i is in $W(t_a)$ or*
- *at least two of $\{x_i, y_i, z_i\}$ are in $W(t_a)$.*

Here, $W(t_a)$ is the witness set containing a in (G'/F^*)-witness structure of G'.

Proof. Let F be a set of edges of size at most k in G' such that G'/F is a tree with at most ℓ leaves. Let $\mathcal{W}$ be a T-witness structure of G' where $T = G'/F$. Let t_a be the vertex in $V(T)$ such that $W(t_a)$ contains a. For a vertex b_i in B, if b_i is in $W(t_a)$ then the lemma holds. Consider a case when b_i is not in $W(t_a)$. There exists a vertex t_b, different from t_a, such that b_i is in $W(t_b)$. Similarly, consider vertices t_x, t_y and t_z such that x_i, y_i and z_i are contained in $W(t_x), W(t_y)$ and $W(t_z)$, respectively.

If neither of t_a or t_b is contained in set $\{t_x, t_y, t_z\}$, then no two vertices in $\{t_x, t_y, t_z\}$ can be same as only neighbors of x_i, y_i, z_i are a and b_i, and a witness set needs to be connected. But then, by construction, $T[\{t_a, t_x, t_y, t_z, t_b\}]$ has at least two cycles which share an edge, contradicting that F is a solution. Without loss of generality, let $t_x \in \{t_a, t_b\}$. This implies there is an edge $t_a t_b$ is in T. If t_a and t_b not equal to t_y or t_z then, $T[\{t_a, t_y, t_z, t_b\}]$ has at least two cycles which share $t_a t_b$, contradicting that F is a solution. Therefore, at most one of t_x, t_y, t_z can be different from t_a or t_b. Without loss of generality, assume that $\{t_x, t_y\}$ is a subset of $\{t_a, t_b\}$. If both t_x, t_y are same as t_a, then the second condition of the lemma is satisfied. Therefore, we assume that at least one of t_x, t_y, say t_x, is not same as t_a which implies $t_x = t_b$. By construction, the only edges incident to x_i in G are ax_i and bx_i. This implies that $bx_i \in F$ and $W(t'_b) = W(t_b) \setminus \{x_i\}$ is connected. Since $ax_i \in E(G)$, $W(t'_a) = W(t_a) \cup \{x_i\}$ is connected. Thus, replacing $W(t_b)$ by $W(t'_b)$ and $W(t_a)$ by $W(t'_a)$ in $\mathcal{W}$ yields another T-witness structure of G. Furthermore, the spanning forest of the new witness structure, $F' = (F \setminus \{bx_i\}) \cup \{ax_i\}$

which has same cardinality as that of F. A similar swap can be carried out if $t_y = t_b$. Hence there a witness structure such that for each $b_i \in B$ if b_i is not in $W(t_a)$ then at least two of $\{x_i, y_i, z_i\}$ are in $W(t_a)$. $\square$

In the following lemma, we argue that the reduction is safe.

Lemma 6.6. *(G, R, B, k) is a* YES *instance of* RBDS *if and only if (G, k', ℓ') is a* YES *instance of* BOUNDED CC*.*

Proof. Let (G, R, B, k) be a YES instance of RBDS and S be a subset of R of size k such that S dominates every vertex in B. If S contains less than k vertices, then we take any of its superset of size exactly k. For each vertex b in B, we fix a vertex r_b in S such that b is neighbor of r_b in G. If there are multiple options for selecting r_b then we arbitrarily choose one of them. Let $F = \{br_b \mid b \in B\} \cup \{ar \mid r \in S\}$. Note that $|F| = |B| + k = k'$ and $G'[V(F)]$ is connected. Let T be the graph obtained from G' by contracting F. Let $\mathcal{W}$ be a T-witness structure of G'. Consider a vertex t_a such that a is in $W(t_a)$. Since all the edges in F are contracted to one vertex, set $S \cup B$ is also contained in $W(t_a)$. By construction, $R \cup X$ is an independent set in G'. No vertex in $(R \cup X) \setminus S$ is incident on edge which has been contracted. In other words, these vertices form singleton witness sets in $\mathcal{W}$. Since $R \cup X$ is an independent set in G', it follows that set $T_{RX} = \{t_v \mid v \in (R \cup X) \setminus S\}$ is an independent set in T of size $|R| + 3|B| - k = \ell'$. Moreover, for all v in X', $av \in E(T)$. Therefore, T is a star (which is a cactus) with ℓ' leaves. This implies that F is a solution to (G', k', ℓ').

In the reverse direction, let (G, k', ℓ') be a YES instance of BOUNDED CC and $F \subseteq E(G)$ be one of its solution. Then by Lemma 6.5, there exists a solution F^* of size at most k' such that for all $b_i \in B$, either $b_i \in W(t_a)$ or at least two of x_i, y_i, z_i are in $W(t_a)$. Here, $\mathcal{W}$ is a G/F^*-witness structure of G and $t_a \in V(G/F^*)$ such that $a \in W(t_a)$.

We partition vertices of B into two parts depending on whether they belong to $W(t_a)$ or not. Define $B_g = \{b_i \in B \mid b_i \in W(t_a)\}$. Let $R_a = R \cap W(t_a)$. Partition B_g into B_1 and B_2, depending on whether or not they have a neighbor in R_a. Formally, $B_1 = \{b_i \in B_g \mid N(b_i) \cap R_a \neq \emptyset\}$ and $B_2 = B_g \setminus B_1$. For a vertex b_i in B_2 at least one of x_i, y_i, z_i is present in $W(t_a)$ as there is no edge between b_i and a. Note that, by construction, x_i, y_i, z_i are not adjacent with b_j for $i \neq j$. This implies there exists a separate vertex for each b_i in B_2 which provides connectivity between a and b_i. Let X_2^B be set of vertices in $X \cap W(t_a)$ which provides adjacency between a and b_i for some b_i in B_2.

For every b_i which is in $B \setminus B_g$, by Lemma 6.5, at least two of vertices in $\{x_i, y_i, z_i\}$ are present in $W(t_a)$.

We can partition $W(t_a) \setminus \{a\}$ into following four parts: vertices in B (captured by B_g); vertices in R (captured by R_a); vertices in X which are present because corresponding b_i is not present (captured by $B \setminus B_g$); and vertices in X which are present because they are needed to provide connectivity between b_i and a (captured by X_2^B). This implies $|B_g| + 2|B \setminus B_g| + |R_a| + |X_2^B| + |\{a\}| \leq |W(t_a)|$.

We construct a solution S for RBDS by taking vertices in R_a and two more sets S_g and S_w. Informally, S_g dominates vertices in B_2 and S_w dominates vertices in $B \setminus B_g$. We construct S_g in following way. For every vertex b_i in B_2, arbitrary pick one of its neighbor in R and add it to S_g. Note that $|S_g| \leq |X_2^B|$. We create another set S_w in the following way. Initialize S_w to an empty set. For each b in $B \setminus B_g$, we add an arbitrary neighbor of b in R to S_w. This implies $|S_w| \leq |B \setminus B_g|$.

As cardinality of F^* is at most $k + |B|$, size of $W(t_a)$ is at most $|W(t_a)| \leq k + |B| + 1$. Putting all inequalities together, we get $|R_a| + |S_g| + |S_w| \leq k$ and every vertex in B is dominated some vertex in $R_a \cup S_g \cup S_w$. This concludes the proof. $\square$

We are now in the position to present a kernel lower bound for Bounded Cactus Contraction.

Theorem 6.3. Bounded Cactus Contraction *does not admit a compression of size $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$, for any $\epsilon > 0$.*

Proof. Assuming a contradiction, suppose Bounded CC admits a compression into $\Pi \subseteq \Sigma^*$ with bitsize in $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$, for some $\epsilon > 0$. This implies that there exists an algorithm $\mathcal{A}$ which takes an instance $I = (G, k, \ell)$ of Bounded CC and in polynomial time returns an equivalent instance I' of Π with $|I'| \in \mathcal{O}((k^2 + k\ell)^{1-\epsilon})$.

Let (G, R, B, k) be an instance of RBDS, where G is a graph on n vertices. Using the reduction described, we create an instance (G, k', ℓ') of Bounded CC with $|V(G'_D)| \in \mathcal{O}(n)$, $|E(G'_D)| \in \mathcal{O}(n^2)$, $k' = k \leq |R| \in \mathcal{O}(n)$ and $\ell' = |B| + k \in \mathcal{O}(n)$. By Lemma 6.6, (G, R, B, k) is a Yes instance of RBDS if and only if (G, k', ℓ') is a Yes instance of Bounded CC. On the instance (G, k', ℓ') we run the algorithm $\mathcal{A}$ to obtain an instance I of Π such that $|I| \in \mathcal{O}((k'^2 + k'\ell')^{1-\epsilon})$. But then we have obtained a compression of size $\mathcal{O}(n^{2-\epsilon})$ for RBDS, contradicting Proposition 2. $\square$

Corollary 6.3. Bounded Cactus Contraction *does not admit a kernel of size* $\mathcal{O}((k^2 + k\ell)^{1-\epsilon})$*, for any* $\epsilon > 0$.

7. Conclusion

In this article, we analyze the structure of the family of paths that allows Path Contraction to admit a polynomial kernel but forbids Tree Contraction. Apart from solution size k, we make the number of leaves, ℓ, as an additional parameter to bridge the gap between kernels of these two problems. We call this problem as Bounded Tree ContractionẆe present a polynomial kernel for this problem. We also prove that this kernel is optimal under a certain complexity assumption. We prove similar results for Out-Tree Contraction and Cactus Contraction problems.
Acknowledgement : The authors would like to thank Prof. Daniel Lokshtanov for helpful discussions.

[1] Takao Asano and Tomio Hirata. Edge-Contraction Problems. *Journal of Computer and System Sciences*, 26(2):197–208, 1983.

[2] Ivan Bliznets, Fedor V Fomin, Marcin Pilipczuk, and Michał Pilipczuk. A subexponential parameterized algorithm for proper interval completion. *SIAM Journal on Discrete Mathematics*, 29(4):1961–1987, 2015.

[3] Ivan Bliznets, Fedor V Fomin, Marcin Pilipczuk, and Michał Pilipczuk. Subexponential parameterized algorithm for interval completion. *ACM Transactions on Algorithms (TALG)*, 14(3):35, 2018.

[4] Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.

[5] Leizhen Cai and Chengwei Guo. Contracting few edges to remove forbidden induced subgraphs. In *International Symposium on Parameterized and Exact Computation (IPEC)*, pages 97–109, 2013.

[6] Yixin Cao. Linear recognition of almost interval graphs. In *ACM-SIAM Symposium On Discrete algorithms (SODA)*, pages 1096–1115, 2016.

[7] Yixin Cao. Unit interval editing is fixed-parameter tractable. *Information and Computation*, 253:109–126, 2017.

[8] Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. *ACM Transactions on Algorithms (TALG)*, 11(3):21, 2015.

[9] Yixin Cao and Dániel Marx. Chordal editing is fixed-parameter tractable. *Algorithmica*, 75(1):118–137, 2016.

[10] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*, volume 4. Springer, 2015.

[11] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[12] Rodney G Downey and Michael Ralph Fellows. *Parameterized Complexity*. Springer Science & Business Media, 2012.

[13] Pål Grønås Drange, Markus Sortland Dregi, Daniel Lokshtanov, and Blair D Sullivan. On the threshold of intractability. In *Algorithms-ESA 2015*, pages 411–423. Springer, 2015.

[14] Pål Grønås Drange, Fedor V Fomin, Michał Pilipczuk, and Yngve Villanger. Exploring subexponential parameterized complexity of completion problems. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 288–299, 2014.

[15] Pål Grønås Drange and Michał Pilipczuk. A polynomial kernel for trivially perfect editing. *Algorithmica*, 80(12):3481–3524, 2018.

[16] Jörg Flum and Martin Grohe. Parameterized Complexity Theory, 2006.

[17] Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, and Yngve Villanger. Tight bounds for parameterized complexity of cluster editing with a small number of clusters. *Journal of Computer and System Sciences*, 80(7):1430–1447, 2014.

[18] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar F-deletion: Approximation, kernelization and optimal FPT algorithms. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 470–479, 2012.

[19] Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019.

[20] Fedor V. Fomin and Yngve Villanger. Subexponential parameterized algorithm for minimum fill-in. *SIAM Journal on Computing*, 42(6):2197–2216, 2013.

[21] Esha Ghosh, Sudeshna Kolay, Mrinal Kumar, Pranabendu Misra, Fahad Panolan, Ashutosh Rai, and M. S. Ramanujan. Faster parameterized algorithms for deletion to split graphs. *Algorithmica*, 71(4):989–1006, 2015.

[22] Petr A. Golovach, Pim van 't Hof, and Daniel Paulusma. Obtaining planarity by contracting few edges. *Theoretical Computer Science*, 476:38–46, 2013.

[23] Sylvain Guillemot and Dániel Marx. A faster FPT algorithm for bipartite contraction. *Information Processing Letters*, 113(22-24):906–912, 2013.

[24] P. Heggernes, P. van't Hof, B. Lévêque, D. Lokshtanov, and C. Paul. Contracting Graphs to Paths and Trees. *Algorithmica*, 68(1):109–132, 2014.

[25] Pinar Heggernes, Pim van 't Hof, Bart M. P. Jansen, Stefan Kratsch, and Yngve Villanger. Parameterized complexity of vertex deletion into perfect graph classes. In *International Symposium on Fundamentals of Computation Theory (FCT)*, pages 240–251, 2011.

[26] Pinar Heggernes, Pim van 't Hof, Daniel Lokshtanov, and Christophe Paul. Obtaining a bipartite graph by contracting few edges. *SIAM Journal on Discrete Mathematics*, 27(4):2143–2156, 2013.

[27] Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A near-optimal planarization algorithm. In *ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, pages 1802–1811, 2014.

[28] Bart M. P. Jansen and Astrid Pieterse. Sparsification upper and lower bounds for graphs problems and not-all-equal SAT. In *International Symposium on Parameterized and Exact Computation (IPEC)*, pages 163–174, 2015.

[29] Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. In *Automata, Languages, and Programming - 40^{th} International Colloquium, (ICALP)*, pages 613–624, 2013.

[30] Sudeshna Kolay, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Quick but Odd Growth of Cacti. In *International Symposium on Parameterized and Exact Computation (IPEC)*, pages 258–269, 2015.

[31] Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. On the hardness of eliminating small induced subgraphs by contracting edges. In *International Symposium on Parameterized and Exact Computation (IPEC)*, pages 243–254, 2013.

[32] Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010.

[33] Rolf Niedermeier. *Invitation to Fixed Parameter Algorithms.* Oxford University Press, 2006.

[34] Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

[35] T Watanabe, T Ae, and A Nakamura. On the removal of forbidden graphs by edge-deletion or by edge-contraction. *Discrete Applied Mathematics*, 3(2):151–153, 1981.

[36] T Watanabe, T Ae, and A Nakamura. On the NP-hardness of edge-deletion and-contraction problems. *Discrete Applied Mathematics*, 6(1):63–78, 1983.