TREE DELETION SET HAS A POLYNOMIAL KERNEL (BUT NO OPT$^{\mathcal{O}(1)}$ APPROXIMATION)[*]

ARCHONTIA C. GIANNOPOULOU[†], DANIEL LOKSHTANOV[‡], SAKET SAURABH[§], AND ONDŘEJ SUCHÝ[¶]

Key words. Tree Deletion Set, Feedback Vertex Set, Kernelization, Linear Equations

AMS subject classifications. F.2.2, G.2.2

Abstract. In the TREE DELETION SET problem the input is a graph G together with an integer k. The objective is to determine whether there exists a set S of at most k vertices such that $G \setminus S$ is a tree. The problem is NP-complete and even NP-hard to approximate within any factor of OPT^c for any constant c. In this paper we give an $\mathcal{O}(k^5)$ size kernel for the TREE DELETION SET problem. An appealing feature of our kernelization algorithm is a new reduction rule, based on systems of linear equations, that we use to handle the instances on which TREE DELETION SET is hard to approximate.

1. Introduction.

In the TREE DELETION SET problem we are given as input an undirected graph G and integer k, and the task is to determine whether there exists a set $S \subseteq V(G)$ of size at most k such that $G \setminus S$ is a tree, that is, a connected acyclic graph. This problem was first mentioned by Yannakakis [27] and is related to the classical FEEDBACK VERTEX SET problem. Here input is a graph G and integer k and the goal is to decide whether there exists a set S on at most k vertices such that $G \setminus S$ is acyclic. The only difference between the two problems is that in TREE DELETION SET $G \setminus S$ is required to be connected, while in FEEDBACK VERTEX SET it is not. Both problems are known to be NP-complete [9, 27].

Despite the apparent similarity between the two problems their computational complexities differ quite dramatically. In particular, FEEDBACK VERTEX SET admits a 2-approximation algorithm, while TREE DELETION SET is known to not admit any approximation algorithm with ratio $\mathcal{O}(n^{1-\epsilon})$ for any $\epsilon > 0$, unless P = NP [1, 27]. With respect to parameterized algorithms, the two problems exhibit more similar behavior. Indeed, some of the techniques that yield fixed parameter tractable algorithms for FEEDBACK VERTEX SET [4, 5] can be adapted to also work for TREE DELETION SET [23].

It is also interesting to compare the behavior of the two problems with respect to polynomial time preprocessing procedures. Specifically, we consider the two problems in the realm of *kernelization*. We say that a parameterized graph problem admits a *kernel* of size $f(k)$ if there exists a polynomial time algorithm, called a *kernelization*

[*]A PRELIMINARY VERSION OF THIS PAPER HAS APPEARED IN THE PROCEEDINGS OF FSTTCS 2014 [10].

[†]Institute of Informatics, University of Warsaw, Warsaw, Poland. Email: archontia.giannopoulou@gmail.com. The research of this author leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 267959.

[‡]Department of Informatics, University of Bergen, Norway. Email: daniello@ii.uib.no. The research of this author was supported by the Bergen Research Foundation and the University of Bergen through project "BeHard".

[§]The Institute of Mathematical Sciences, Chennai, India. Email: saket@imsc.res.in. The research of this author was supported by the European Research Council through Starting Grant 306992 Parameterized Approximation.

[¶]Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, Czech Republic. Email: ondrej.suchy@fit.cvut.cz. The research of this author was partially supported by the grant 14-13017P of the Czech Science Foundation.

algorithm, that given as input an instance (G, k) to the problem outputs an equivalent instance (G', k') with $k' \leq f(k)$ and $|V(G')| + |E(G')| \leq f(k)$. If the function f is a polynomial, we say that the problem admits a *polynomial kernel*. We refer to the surveys [11, 20, 16] for an introduction to kernelization. For the FEEDBACK VERTEX SET problem, Burrage et al. [3] gave a kernel of size $\mathcal{O}(k^{11})$. Subsequently, Bodlaender [2] gave an improved kernel of size $\mathcal{O}(k^3)$ and finally Thomassé [24] gave a kernel of size $\mathcal{O}(k^2)$. On the other hand the existence of a polynomial kernel for TREE DELETION SET was open until this work. It seems difficult to directly adapt any of the known kernelization algorithms for FEEDBACK VERTEX SET to TREE DELETION SET. Indeed, Raman et al. [23] conjectured that TREE DELETION SET does not admit a polynomial kernel.

The main reason to conjecture that TREE DELETION SET does not admit a polynomial kernel stems from an apparent relation between kernelization and approximation algorithms (cf. [21, page 15]). Most problems that admit a polynomial kernel, also have approximation algorithms with approximation ratio polynomial in OPT (cf. [15, page 2]). Here OPT is the value of the optimum solution to the input instance. In fact many kernelization algorithms are already approximation algorithms with approximation ratio polynomial in OPT. This relation between approximation and kernelization led to a conjecture [22, 7] that VERTEX COVER does not admit a kernel with $(2-\epsilon)k$ vertices for $\epsilon > 0$, as this probably would yield a $(2-\epsilon)$-approximation for the problem thus violating the Unique Games Conjecture [14].

It is easy to show that an approximation algorithm for TREE DELETION SET with ratio $\mathrm{OPT}^{\mathcal{O}(1)}$ would yield an approximation algorithm for the problem with ratio $\mathcal{O}(n^{1-\epsilon})$ thereby proving $\mathtt{P} = \mathtt{NP}$. In particular, suppose TREE DELETION SET had an OPT^c algorithm for some constant c. Since the algorithm will never output a set of size more than n, the approximation ratio of the algorithm is upper bounded by $\min(\mathrm{OPT}^c, \frac{n}{\mathrm{OPT}}) \leq n^{1-\frac{1}{c+1}}$. This rules out approximation algorithms for TREE DELETION SET with ratio $\mathrm{OPT}^{\mathcal{O}(1)}$, and makes it very tempting to conjecture that TREE DELETION SET does not admit a polynomial kernel.

In this paper we show that TREE DELETION SET admits a kernel of size $\mathcal{O}(k^5)$. To the best of our knowledge this is among the few examples of problems that do admit a polynomial kernel, but do not admit any approximation algorithm with ratio $\mathrm{OPT}^{\mathcal{O}(1)}$ under plausible complexity assumptions. The only other example we are aware of is the problem MIN ONES 1-IN-3 SAT which is a special case of the CSP studied by Kratsch and Wahlström [17].

Our Methods. The starting point of our kernel are known reduction rules for FEEDBACK VERTEX SET adapted to our setting. We also adapt the strategy to model some "pendant parts" of the graph by weight on vertices during the kernelization process to simplify the structure of the graph. By applying these graph theoretical reduction rules we can show that there is a polynomial time algorithm that given an instance (G, k) of TREE DELETION SET outputs an equivalent instance (G', k') and a partition of $V(G')$ into sets B, T, and I such that

1. $|B| = \mathcal{O}(k^2)$,
2. $|T| = \mathcal{O}(k^4)$,
3. I is an independent set, and
4. for every $v \in I$, $N_{G'}(v) \subseteq B$, and $N_{G'}(v)$ is a double clique.

Here a "double clique" means that for every pair x, y of vertices in $N_{G'}(v)$, there are two edges between them. Thus we will allow G' to be a multigraph, and consider a double edge between two vertices as a cycle. In order to obtain a polynomial kernel for Tree Deletion Set it is sufficient to reduce the set I to size polynomial in k.

For every vertex $v \in I$ and tree deletion set S we know that $|N_{G'}(v) \setminus S| \leq 1$, since otherwise $G' \setminus S$ would contain a double edge. Further, if $v \notin S$ then v has to be connected to the rest of $G' \setminus S$ and hence $|N_{G'}(v) \setminus S| = 1$, implying that v is a leaf in $G' \setminus S$. Therefore $G' \setminus (S \cup I)$ must be a tree. We can now reformulate the problem as follows.

For each vertex u in $G' \setminus I$ we have a variable x_u which is set to 0 if $u \in S$ and $x_u = 1$ if $u \notin S$. For each vertex $v \in I$ we have a linear equation $\sum_{u \in N(v)} x_u = 1$. The task is to determine whether it is possible to set the variables to 0 or 1 such that (a) the subgraph of G' induced by the vertices with variables set to 1 is a tree and (b) the number of variables set to 0 plus the number of unsatisfied linear equations is at most k.

At this point it looks difficult to reduce I by graph theoretic means, as performing operations on these vertices correspond to making changes in a system of linear equations. In order to reduce I we prove that there exists an algorithm that given a set $\mathcal{S}$ of linear equations on n variables and an integer k in time $\mathcal{O}(|\mathcal{S}|n^{\omega-1}k)$ outputs a set $\mathcal{S}' \subseteq \mathcal{S}$ of at most $(n+1)(k+1)$ linear equations such that any assignment of the variables that violates at most k linear equations of $\mathcal{S}'$ satisfies all the linear equations of $\mathcal{S} \setminus \mathcal{S}'$. To reduce I we simply apply this result and keep only the vertices of I that correspond to linear equations in $\mathcal{S}'$. We believe that our reduction rule for linear equations will find more applications in the future and, while not as involved, adds a little to the toolbox of algebraic reduction rules for kernelization (see, for example, [6, 18, 19, 25]).

2. Basic Notions. For every positive integer n we denote $[n] = \{1, 2, \ldots, n\}$, and for every set S we denote by $\binom{S}{2}$ the 2-subsets of S. $\mathbb{N}$ denotes the set of positive integers, and $\mathbb{R}$ denotes the real numbers.

For a graph $G = (V, E)$, we use $V(G)$ to denote its vertex set V and $E(G)$ to denote its edge set E. If $S \subseteq V(G)$ we denote by $G \setminus S$ the graph obtained from G after removing the vertices of S. In the case where $S = \{u\}$, we abuse notation and write $G \setminus u$ instead of $G \setminus \{u\}$. For $S \subseteq V(G)$, the *neighborhood* of S in G, $N_G(S)$, is the set $\{u \in V(G) \setminus S \mid \exists v \in S : \{u, v\} \in E(G)\}$. Again, in the case where $S = \{v\}$ we abuse notation and write $N_G(v)$ instead of $N_G(\{v\})$. The degree of vertex v denoted $\deg(v)$ is the number of edges incident to it, loops being counted twice. A graph is connected if there is a path between any pair of its vertices. A connected component in a graph G is a set of vertices H such that $G[H]$ is connected and H is maximal with this property. We use $\mathcal{C}(G)$ to denote the set of the connected components of G. Given a graph G and a set $S \subseteq V(G)$, we say that S is a *feedback vertex set* of G if the graph $G \setminus S$ does not contain any cycles. In the case where $G \setminus S$ is also connected we call S *tree deletion set* of G. Moreover, given a set $S \subseteq V(G)$, we say that S is a double clique of G if every pair of vertices in S is joined by a double edge.

Given two vectors x and y we denote by $\mathbf{d}_H(x, y)$ the Hamming distance of x and y, that is, $\mathbf{d}_H(x, y)$ is equal to the number of positions where the vectors differ. For every $k \in \mathbb{N}$ we denote by $\mathbf{0}^k$ the k-component vector $(0, 0, \ldots, 0)$. When k is implied from the context we abuse notation and denote $\mathbf{0}^k$ as $\mathbf{0}$.

For a rooted tree T and vertex set M in $V(T)$ the least common ancestor-closure (*LCA-closure*) **LCA-closure**(M) is obtained by the following process. Initially, set $M' = M$. Then, as long as there are vertices x and y in M' whose least common ancestor w is not in M', add w to M'. Finally, output M' as the LCA-closure of M.

LEMMA 2.1 (Fomin et al. [8]). *Let T be a tree and $M \subseteq V(T)$. If $M' =$ **LCA-closure**(M) then $|M'| \leq 2|M|$ and for every connected component C of $T \setminus M'$, $|N_T(C)| \leq 2$.*

3. A polynomial kernel for TREE DELETION SET. In this section we prove a polynomial size kernel for a weighted variant of the TREE DELETION SET problem. More precisely the problem we will study is the following.

WEIGHTED TREE DELETION SET (wTDS)	
Instance:	A graph G, a function $w : V(G) \to \mathbb{N}$, and a non-negative integer k.
Parameter:	k.
Question:	Does there exist a set $S \subseteq V(G)$ such that $\sum_{v \in S} w(v) \leq k$ and $G \setminus S$ is a tree?

3.1. Known Reduction Rules for wTDS. In this subsection we state some already known reduction rules for wTDS that are going to be needed during our proofs.

REDUCTION RULE 1 (Raman et al. [23]). *If the input graph is disconnected, then delete all vertices in connected components of weight less than $(\sum_{v \in V} w(v)) - k$ and decrease k by the weight of the deleted vertices.*

OBSERVATION 1 (Raman et al. [23]). *If $\left(\sum_{v \in V} w(v)\right) > 2k$, then after the exhaustive application of Reduction Rule 1 the graph has at most one connected component.*

REDUCTION RULE 2 (Raman et al. [23]). *If v is of degree 1 and u is its only neighbor, then delete v and increase the weight of u by the weight of v.*

REDUCTION RULE 3 (Raman et al. [23]). *If $v_0, v_1, \ldots, v_l, v_{l+1}$ is a path in the input graph, such that $l \geq 3$ and $\deg(v_i) = 2$ for every $i \in [l]$, then replace the vertices $v_1, \ldots, v_l$ by two vertices u_1 and u_2 with edges $\{v_0, u_1\}$, $\{u_1, u_2\}$, and $\{u_2, v_{l+1}\}$ and with $w(u_1) = \min\{w(v_i) \mid i \in [l]\}$ and $w(u_2) = \left(\sum_{i=1}^{l} w(v_i)\right) - w(u_1)$.*

Given a vertex x of G, an *x-flower of order k* is a set of k cycles pairwise intersecting exactly in x. If G has an x-flower of order $k+1$, then x should be in every tree deletion set of weight at most k as otherwise we would need at least $k+1$ vertices to hit all cycles passing through x. Thus the following reduction rule is safe, that is, the instance obtained after the application of the reduction rule is equivalent to the original instance.

REDUCTION RULE 4. *Let (G, w, k) be an instance of* wTDS. *If G has an x-flower of order at least $k+1$, then remove x and decrease the parameter k by the weight of x. The resulting instance is $(G \setminus \{x\}, w|_{V(G)\setminus\{x\}}, k - w(x))$.*

The following theorem allows us to apply Reduction Rule 4 exhaustively in polynomial time. A version of the theorem appears also in [2], but the version given in [24] is significantly more powerful.

THEOREM 3.1 (Thomassé [24]). *Let G be a multigraph and x be a vertex of G without a self loop. Then in polynomial time we can find an x-flower of order $k+1$ or, if such an x-flower does not exist, a set of vertices $Z \subseteq V(G) \setminus \{x\}$ of size at most $2k$ intersecting every cycle containing x.*

REDUCTION RULE 5. *Let (G, w, k) be an instance of* wTDS. *If v is a vertex such that $w(v) > k+1$, then let $w(v) = k+1$.*

An instance (G, w, k) of wTDS is called *semi-reduced* if none of the Reduction Rules 1–5 can be applied. By Observation 1 such an instance is either connected or the total weight of all vertices is at most $2k$ and hence we have a kernel. Therefore, for the rest of the paper we assume that the instance is connected.

LEMMA 3.2. *If* (G, w, k) *is an instance of* wTDS *reduced with respect to Reduction Rule 5, then there is an equivalent instance* (G', k) *of* TREE DELETION SET *such that* $|V(G')| \leq (k+1)|V(G)|$ *and* $|E(G')| \leq |E(G)| + |V(G')|$.

Proof. The graph G' is obtained by introducing, for each $v \in V(G)$, $w(v) - 1$ vertices pending to v. The equivalence of the instances follows from Reduction Rule 2. The number of vertices follows from the fact that Reduction Rule 5 does not apply and, thus, $w(v) \leq k+1$ for every v. Finally, the number of edges follows from the construction. ∎

THEOREM 3.3 (Bafna et al. [1]). *There is an* $\mathcal{O}(\min\{|E(G)| \log |V(G)|, |V(G)|^2\})$ *time algorithm that given a graph* G *that admits a feedback vertex set of size at most* k *outputs a feedback vertex set of* G *of size at most* $2k$.

3.2. A structural decomposition. In this subsection we decompose an instance (G, w, k) of wTDS to an equivalent instance (G', w', k') where $V(G')$ is partitioned into three sets B, T, and I, such that the size of B and T is polynomial in k and I is an independent set. Notice then that in order to obtain a polynomial kernel for $wTDS$ what remains is to bound the size of the independent set I. This will be done in the next section. In this subsection we obtain the following result.

LEMMA 3.4. *There is a polynomial time algorithm that given a semi-reduced instance* (G, w, k) *of* wTDS *either correctly decides that* (G, w, k) *is a no-instance or outputs an equivalent instance* (G', w', k') *and a partition of* $V(G')$ *into sets* B, T, *and* I *such that*

(i) $|B| \leq 8k^2 + 2k$,
(ii) T *induces a forest and* $|T| \leq 240k^4 + 272k^3 + 65k^2 - 19k - 7$,
(iii) I *is an independent set, and*
(iv) for every $v \in I$, $N_{G'}(v) \subseteq B$, $|N_{G'}(v)| \leq 2k+1$, *and* $N_{G'}(v)$ *is a double clique.*

For an example of the structure of the graph G' obtained from Lemma 3.4, see Figure 1.

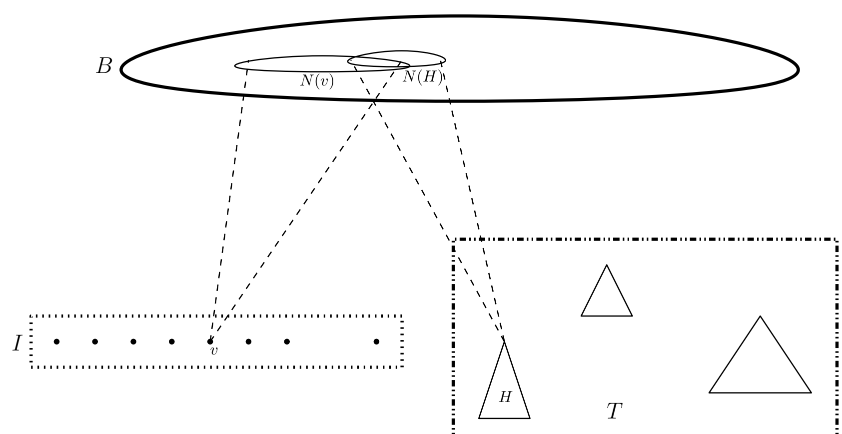

FIG. 1. *The vertex set of the graph* G' *is partitioned into a set* B, *a set* T *where every connected component* H *of* T *is a tree, and a set* I. *The set* I *induces an independent set and for every vertex* $v \in I$, $N_{G'}(v) \subseteq B$ *and* $N_{G'}(v)$ *induces a double clique.*

We split the proof of this lemma into several auxiliary lemmata. We start by identifying the set B (See also Figure 2).

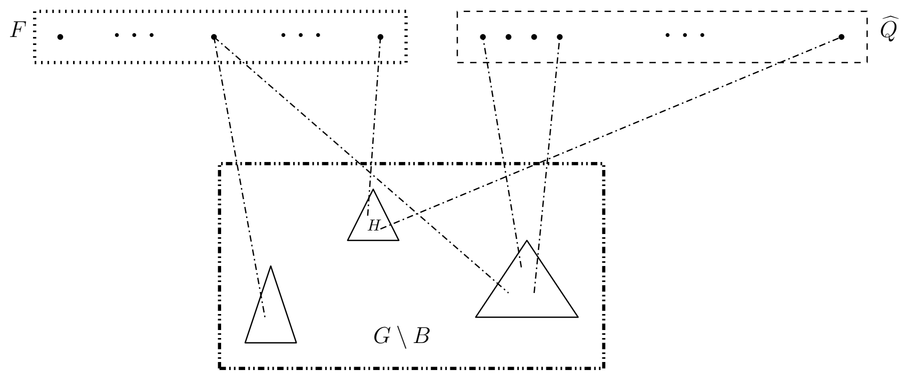

FIG. 2. *The vertex set of the graph G is partitioned into a set $B = F \cup \widehat{Q}$, and the connected components of $G \setminus B$.*

LEMMA 3.5. *There is a polynomial time algorithm that given a semi-reduced instance (G, w, k) of* wTDS *either correctly decides that (G, w, k) is a no-instance or finds two sets F and $\widehat{Q}$ such that, denoting $B = F \cup \widehat{Q}$, the following holds.*

(i) F is a feedback vertex set of G.
(ii) Each connected component of $G \setminus B$ has at most 2 neighbors in $\widehat{Q}$.
(iii) For every $H \in \mathcal{C}(G \setminus B)$ and $y \in B$, $|N_G(y) \cap H| \leq 1$, that is, every vertex y of F and every vertex y of $\widehat{Q}$ have at most one neighbor in every connected component H of $G \setminus B$.
(iv) $|B| \leq 8k^2 + 2k$.

Proof. First notice that every tree deletion set of G of weight at most k is also a feedback vertex set of G of size at most k in the underlying non-weighted graph. Thus, by applying Theorem 3.3 we may find in polynomial time a feedback vertex set F of G. If $|F| > 2k$, then output NO. Otherwise, $|F| \leq 2k$.

As the instance (G, w, k) is semi-reduced, Reduction Rule 4 is not applicable, and G does not contain an x-flower of order $k + 1$ for any $x \in F$. Therefore, from Theorem 3.1, we get that for every $x \in F$ we can find in polynomial time a set $Q^x \subseteq V(G) \setminus \{x\}$ intersecting every cycle that goes through x in G and such that $|Q^x| \leq 2k$. Let $Q = \bigcup_{x \in F} Q^x$.

Let $\mathcal{C}(G \setminus F) = \{H_1, H_2, \ldots, H_l\}$ and note that, as F is a feedback vertex set of G, each $G[H_i]$ is a tree. From now on, without loss of generality we will assume that each $G[H_i]$, $i \in [l]$, is rooted at some vertex $v_i \in H_i$.

Let $Q_i = H_i \cap Q$, $i \in [l]$. In other words, Q_i denotes the set of vertices of H_i that are also vertices of Q, $i \in [l]$. Let also $\widehat{Q}_i = \textbf{LCA-closure}(Q_i)$, that is, let $\widehat{Q}_i$ denote the least common ancestor-closure of the set Q_i in the tree $G[H_i]$. Finally, let $\widehat{Q} = \bigcup_{i \in [l]} \widehat{Q}_i$ and note that $\widehat{Q} \cap F = \emptyset$.

Let us now prove that F and $\widehat{Q}$ have the claimed properties. First of all, F is a feedback vertex set by construction, proving (i). Second, since for each x in F we have $|Q^x| \leq 2k$, we have $|Q| \leq 4k^2$, and from Lemma 2.1 we get that $|\widehat{Q}| = |\bigcup_{i \in [l]} \widehat{Q}_i| = \sum_{i \in [l]} |\widehat{Q}_i| \leq 2 \sum_{i \in l} |Q_i| \leq 2|Q| \leq 8k^2$. Together with $|F| \leq 2k$ this proves (iv). Third, from the construction of $\widehat{Q}$ and from Lemma 2.1 we get the property (ii).

Let us now prove (iii). Let $y \in B$ and $H \in \mathcal{C}(G \setminus B)$ and assume to the contrary that $|N_G(y) \cap H| \geq 2$. Then, as $G[H]$ is connected, the graph $G[H \cup \{y\}]$ contains a cycle that goes through y. If $y \in F$, we get a contradiction to the facts that $G[H \cup \{y\}]$ is a subgraph of $G \setminus Q^y$ and the set Q^y intersects every cycle that goes through y. If $y \in \widehat{Q}$, we get a contradiction, since $G[H \cup \{y\}]$ is a subgraph of $G \setminus F$ (recall that $\widehat{Q} \cap F = \emptyset$) and $G \setminus F$ is acyclic. ∎

The next lemma shows that if B is as in the previous lemma, then the size of connected components in the rest of the graph is bounded.

LEMMA 3.6. *If (G, w, k) and B are as in Lemma 3.5 and H is a connected component of $G \setminus B$, then $|H| \leq 12k + 7$.*

Proof. Let H be a connected component of $G \setminus B$. First recall that, from Lemma 3.5 every vertex of B has at most 1 neighbor in H and H has at most 2 neighbors in $\widehat{Q}$. This implies that there are at most $|F| + 2 \leq 2k + 2$ vertices in H that have a neighbor in $G \setminus H$, and in particular in B. We call this set of vertices N. Let H_1 be the set of vertices of degree 1 in $G[H]$, that is, the leaves of $G[H]$. From Reduction Rule 2 it follows that for every $v \in H_1$, $\deg_G(v) \geq 2$ and thus, as $H \in \mathcal{C}(G \setminus B)$, v has at least one neighbor in B. Therefore, $H_1 \subseteq N$ and $|H_1| \leq 2k + 2$.

Let now H_3 be the set of vertices of degree at least 3 in $G[H]$. For H_3 it is easy to see that, by standard combinatorial arguments on trees, $|H_3| \leq |H_1| - 1 \leq 2k + 1$.

Finally, let P be the set $N \cup H_3$ and $\mathcal{E}$ be the set of paths in $G[H]$ with endpoints in P. Again, as $|P| \leq 4k+3$, it holds that $|\mathcal{E}| \leq 4k+2$. Observe that by construction of $\mathcal{E}$ all the inner vertices of the paths in $\mathcal{E}$ have degree exactly 2. Therefore, from Reduction Rule 3 we get that every path in $\mathcal{E}$ contains at most 2 vertices. This implies that $|H \setminus P| \leq 8k+4$. To conclude, as $|H| = |H \setminus P| + |P|$, we get that $|H| \leq 12k+7$. ∎

Let x, y be two vertices of B. We say that the pair $\{x, y\}$ is in $\mathcal{P}^{\leq k+1}$ if there are at most $k + 1$ connected components H of $G \setminus B$ with $\{x, y\} \subseteq N_G(H)$ and that $\{x, y\}$ is in $\mathcal{P}^{\geq k+2}$ otherwise. Now we add to G a double edge between every pair in $\mathcal{P}^{\geq k+2}$ to obtain the graph $\widehat{G}$. The next lemma shows that the resulting instance is equivalent to the original one.

LEMMA 3.7. *The instance $(\widehat{G}, w, k)$, where $\widehat{G}$ is as defined above, is equivalent to (G, w, k).*

Proof. Let $\{x, y\} \in \mathcal{P}^{\geq k+2}$. Notice that each connected component H of $G \setminus B$ with $\{x, y\} \subseteq N_G(H)$ provides a separate path between x and y. Observe then that if neither x nor y belong to a tree deletion set D of G we need at least $k+1$ vertices to hit all the cycles, since otherwise there are at least two components $H_1, H_2 \in \mathcal{C}(G \setminus B)$ with $\{x, y\} \subseteq (N_G(H_1) \cap N_G(H_2))$ and $(H_1 \cup H_2) \cap D = \emptyset$ and thus the graph induced by $H_1 \cup H_2 \cup \{y, y'\}$ contains a cycle. This implies that (G, w, k) is a yes-instance if and only if at least one of the vertices x and y is contained in every tree deletion set of G of weight k. ∎

The following lemma shows that there are only few connected components of $G \setminus B$ having a neighborhood that is not a double clique in $\widehat{G}$.

LEMMA 3.8. *If (G, w, k) and B are as in Lemma 3.5 and $\widehat{G}$ as defined above, then there is a set $\mathcal{C}_T \subseteq \mathcal{C}(G \setminus B)$ such that*

(i) $|\mathcal{C}_T| \leq 20k^3 + 11k^2 - k - 1$,

(ii) for every H in $\mathcal{C}(G \setminus B) \setminus \mathcal{C}_T$, we have $N_G(H)$ is a double clique in $\widehat{G}$ and $|N_G(H) \cap \widehat{Q}| \leq 1$.

Proof. For $x, y \in B$ we denote $S(x, y) = \{H \in \mathcal{C}(G \setminus B) \mid \{x, y\} \subseteq N_G(H)\}$. Let us set $\mathcal{C}_T = \bigcup_{\{x,y\} \in \mathcal{P}^{\leq k+1}} S(x, y)$. Let us now assume that there is H in $\mathcal{C}(G \setminus B) \setminus \mathcal{C}_T$, and two vertices x and y in $N_G(H)$ that are not joined by a double edge. By construction of the graph $\widehat{G}$, this implies that $\{x, y\} \in \mathcal{P}^{\leq k+1}$. But this implies that H is in $\mathcal{C}_T$, a contradiction. Furthermore, for every $x, y \in \widehat{Q}$ we have $|S(x, y)| \leq 1$ as otherwise we would have a cycle in $G \setminus F$ and F is a feedback vertex set. Thus, if two vertices $x, y \in \widehat{Q}$ belong to $N_G(H)$ then, since $|S(x, y)| \leq 1$, $\{x, y\}$ is in $\mathcal{P}^{\leq k+1}$ and, therefore, H is in $|\mathcal{C}_T|$. Hence $\mathcal{C}_T$ satisfies (ii). It remains to prove (i).

Let us first mention that it is easy to see that $\mathcal{C}_T$ is of polynomial size. Indeed, we have $|\mathcal{C}_T| = |\bigcup_{\{x,y\} \in \mathcal{P}^{\leq k+1}} S(x, y)| \leq |B|^2(k + 1) = \mathcal{O}(k^5)$. For the purpose of the more precise size bound let us distinguish three subsets of $\mathcal{C}_T$:

$$\mathcal{T}^{FF} = \bigcup\nolimits_{\{x,y\} \subseteq F \wedge \{x,y\} \in \mathcal{P}^{\leq k+1}} S(x, y)$$
$$\mathcal{T}^{\widehat{Q}\widehat{Q}} = \bigcup\nolimits_{\{x,y\} \subseteq \widehat{Q} \wedge \{x,y\} \in \mathcal{P}^{\leq k+1}} S(x, y)$$
$$\mathcal{T}^{F\widehat{Q}} = \left(\bigcup\nolimits_{x \in F \wedge y \in \widehat{Q} \wedge \{x,y\} \in \mathcal{P}^{\leq k+1}} S(x, y) \right) \setminus \mathcal{T}^{\widehat{Q}\widehat{Q}}$$

Obviously, $\mathcal{C}_T \subseteq (\mathcal{T}^{FF} \cup \mathcal{T}^{\widehat{Q}\widehat{Q}} \cup \mathcal{T}^{F\widehat{Q}})$. Hence, to bound the size of $\mathcal{C}_T$ it is enough to bound the sizes of $\mathcal{T}^{FF}$, $\mathcal{T}^{\widehat{Q}\widehat{Q}}$, and $\mathcal{T}^{F\widehat{Q}}$. Note that for every $\{x, y\} \in \mathcal{P}^{\leq k+1}$ we have $|S(x, y)| \leq k + 1$. It follows that $|\mathcal{T}^{FF}| \leq \binom{|F|}{2}(k + 1) \leq \binom{2k}{2}(k + 1) = 2k^3 + k^2 - k$.

Next we claim that $|\mathcal{T}^{\widehat{Q}\widehat{Q}}| \leq |\widehat{Q}| - 1 \leq 8k^2 - 1$. For every $x, y \in \widehat{Q}$ we have $|S(x, y)| \leq 1$ as otherwise we would have a cycle in $G \setminus F$ and F is a feedback vertex set. Let A_Q be the graph with vertex set $\widehat{Q}$ where two vertices in $\widehat{Q}$ are connected by an edge if and only if they are the neighbors of a component $H \in \mathcal{T}^{\widehat{Q}\widehat{Q}}$. Hence, the number of edges of A_Q equals $|\mathcal{T}^{\widehat{Q}\widehat{Q}}|$. We now work towards showing that A_Q is a forest. Indeed, assume to the contrary that there exists a cycle in A_Q. Then it is easy to see that we may find a cycle in the graph $\widehat{H}$ induced by the components in $\mathcal{T}^{\widehat{Q}\widehat{Q}}$ which correspond to the edges of the cycle in A_Q and their neighborhood in $\widehat{Q}$. Recall that $\widehat{Q} \cap F = \emptyset$ and therefore $\widehat{H}$ is a subgraph of $G \setminus F$. This contradicts the fact that F is a feedback vertex set of G. Hence, A_Q is a forest and the claim follows.

For the upper bound on $\mathcal{T}^{F\widehat{Q}}$, for every $x \in F$ we partition the set $\widehat{Q}$ into two sets $R_x^{\leq 1}$ and $R_x^{\geq 2}$ in the following way.

$$R_x^{\leq 1} = \{y \in \widehat{Q} \mid \text{there is at most 1 component } H \in \mathcal{T}^{F\widehat{Q}} \text{ such that } \{x, y\} \subseteq N_G(H)\}$$
$$R_x^{\geq 2} = \{y \in \widehat{Q} \mid \{x, y\} \in \mathcal{P}^{\leq k+1} \text{ and there exist at least two distinct components}$$
$$H_1, H_2 \in \mathcal{T}^{F\widehat{Q}} \text{ such that } \{x, y\} \subseteq N_G(H_1) \cap N_G(H_2)\}.$$

Observe that $|\mathcal{T}^{F\widehat{Q}}| \leq \sum_{x \in F} \left(|R_x^{\leq 1}| + |R_x^{\geq 2}|(k + 1)\right)$ and for every $x \in F$, it trivially holds that $|R_x^{\leq 1}| \leq |\widehat{Q}| \leq 8k^2$.

Moreover, we claim that for every $x \in F$, $|R_x^{\geq 2}| \leq k$. Indeed, assume to the contrary that $|R_x^{\geq 2}| \geq k + 1$ for some $x \in F$. Then there exist $k + 1$ vertices $y_i \in \widehat{Q}$, $i \in [k + 1]$, such that for every i there exist two connected components H_1^i and H_2^i in $\mathcal{T}^{F\widehat{Q}} \subseteq \mathcal{C}(G \setminus B) \setminus \mathcal{T}^{\widehat{Q}\widehat{Q}}$ such that $\{x, y\} \subseteq N_G(H_1^i) \cap N_G(H_2^i)$. This implies that the graph induced by the vertex x, the vertices y_i, $i \in [k + 1]$, and the components H_1^i and H_2^i, $i \in [k + 1]$, contains an x-flower of order $k + 1$ (notice that, as none of the graphs belong to $\mathcal{T}^{\widehat{Q}\widehat{Q}}$, they are pairwise disjoint). This is a contradiction to the fact that G is semi-reduced. Therefore, for every $x \in F$ we have $|R_x^{\geq 2}| \leq k$.

Alltogether, we have

$$|\mathcal{T}^{F\widehat{Q}}| \leq \sum_{x \in F} \left(8k^2 + k(k+1)\right) \leq 18k^3 + 2k^2$$

and

$$\begin{aligned} |\mathcal{C}_T| &\leq |\mathcal{T}^{FF}| + |\mathcal{T}^{\widehat{Q}\widehat{Q}}| + |\mathcal{T}^{F\widehat{Q}}| \\ &\leq (2k^3 + k^2 - k) + (8k^2 - 1) + (18k^3 + 2k^2) \\ &= 20k^3 + 11k^2 - k - 1 \end{aligned}$$

proving (i). ∎

Let us denote $T = \bigcup_{H \in \mathcal{C}_T} H$. Note that by the properties of the set $\mathcal{C}_T$ we have $\mathcal{C}(\widehat{G} \setminus (B \cup T)) = \mathcal{C}(G \setminus B) \setminus \mathcal{C}_T$. Further, by Lemma 3.6 we have $|T| \leq |\mathcal{C}_T|(12k+7)$ and, hence, by Lemma 3.8,

$$|T| \leq (20k^3 + 11k^2 - k - 1)(12k + 7) = 240k^4 + 272k^3 + 65k^2 - 19k - 7.$$

We now prove that the components of $\mathcal{C}(G \setminus B)$ that are not in $\mathcal{C}_T$ behave as single vertices with respect to tree deletion sets.

Lemma 3.9. *If there exists a tree deletion set S of $\widehat{G}$ of weight at most k then there exists a tree deletion set $\widehat{S}$ of $\widehat{G}$ of weight at most k such that for every connected component $H \in \mathcal{C}(\widehat{G} \setminus (B \cup T))$, either $H \subseteq \widehat{S}$ or $H \cap \widehat{S} = \emptyset$.*

Proof. Recall that for every $H \in \mathcal{C}(\widehat{G} \setminus B) \setminus \mathcal{C}_T$ it holds that $N_{\widehat{G}}(H)$ is a double clique by Lemma 3.8 (ii). Therefore, either $N_{\widehat{G}}(H) \subseteq S$ or there exists a unique vertex of $N_{\widehat{G}}(H)$ that does not belong to S. Notice that in the case where $N_{\widehat{G}}(H) \subseteq S$, as $N_{\widehat{G}}(H)$ is a separator of $\widehat{G}$, it trivially follows that either $H \subseteq S$ or $V(\widehat{G}) \setminus H \subseteq S$ and we can assume that $S \cap H = \emptyset$ as $G[H]$ is a tree. Let us now assume that there exists a unique vertex w of $N_{\widehat{G}}(H)$ that does not belong to S and that $H \cap S \neq \emptyset$. As, from Lemma 3.5 (iii) every vertex of $N_{\widehat{G}}(H)$ has exactly one neighbor in H it follows that the graph $\widehat{G}[H \cup \{w\}]$ does not contain a cycle. Moreover, w is a cut vertex of $\widehat{G} \setminus S$ and therefore the graph $\widehat{G}[(V(\widehat{G}) \setminus S) \cup H \cup \{w\}]$ is a tree. Thus in the case where $H \cap S \neq \emptyset$ we can remove the vertices of H from S without introducing any cycles to the graph $\widehat{G}[(V(\widehat{G}) \setminus S) \cup H \cup \{w\}]$. Therefore $S \setminus H \subseteq S$ is also a tree deletion set of $\widehat{G}$ and this concludes the proof. ∎

Now, let G' be the graph obtained from $\widehat{G}$ after contracting every connected component H of $\widehat{G} \setminus (B \cup T)$ into a single vertex v_H and setting $w'(v_H) = \sum_{v \in H} w(v)$ and $w'(v) = w(v)$ for every $v \in (B \cup T)$. We also define I to be the set $V(G') \setminus (B \cup T)$. We now prove that such a contraction does not affect the instance.

Lemma 3.10. *If $\widehat{G}$, G', and w' are as defined above, then the instances $(\widehat{G}, w, k)$ and (G', w', k) are equivalent.*

Proof. Notice that if (G', w', k) is a yes-instance then $(\widehat{G}, w, k)$ is also a yes-instance; for every vertex v in the tree deletion set of weight at most k of G' we consider the vertex v in the tree deletion set of $\widehat{G}$ whenever $v \in B \cup T$ and the vertices of the connected component that was contracted to v whenever $v \in I$.

Let now $\widehat{S}$ be a tree deletion set of $\widehat{G}$ of weight at most k. From Lemma 3.9 we may assume that for every connected component H of $\widehat{G} \setminus (B \cup T)$ either $H \subseteq \widehat{S}$ of $H \cap \widehat{S} = \emptyset$. Then it is straightforward to see that the vertex set S consisting of the vertices $(B \cup T) \cap \widehat{S}$ and the vertices of I that correspond to the connected components of $\widehat{G} \setminus (B \cup T)$ whose vertices belong to $\widehat{S}$ is a tree deletion set of G' of weight equal to the weight of $\widehat{S}$. □

Lemma 3.4 now follows directly from Lemmata 3.5–3.10.

Remark 1. *Notice that for every pair of vertices in $\mathcal{P}^{\geq k+2}$ at least one of them has to be contained in a tree deletion set. Clearly, some of the common neighbors of the pair remain untouched and recall that in the final graph we would like those neighbors to be connected. It might then be tempting to say that among a pair of vertices in $\mathcal{P}^{\geq k+2}$ a solution must remove exactly one of them. This, however, is not the case as their remaining neighbors might be connected to the rest of the graph through other vertices of B. Hence it might be the case that both vertices of the pair are removed.*

3.3. Results on Linear Equations. Here we prove some results on linear equations that are crucial for our kernel. Our purpose is to assign a linear equation to each one of the vertices in I and then use these results to reduce the size of I. In particular, let $I = \{v_i \mid i \in [|I|]\}$ and $B = \{u_j \mid j \in [|B|]\}$. We assign an $\mathbb{F}$-variable x_j to u_j, $j \in [|B|]$, and a linear equation l_i over $\mathbb{F}$ to v_i, $i \in [|I|]$, where l_i is the equation $\sum_{j \in [|B|]} \alpha_{ij} x_j - 1 = 0$ and $\alpha_{ij} = 1$ if $u_j \in N_G(v_i)$ and 0 otherwise. The next lemmas will later on indicate which vertices of I we may safely remove from the instance.

Lemma 3.11. *Let $\mathbb{F}$ be a field. For every matrix $M \in \mathbb{F}^{m \times n}$ and positive integer k, there exists a submatrix $M' \in \mathbb{F}^{m' \times n}$ of M, where $m' \leq n(k+1)$, such that for every $x \in \mathbb{F}^n$ with $\mathbf{d}_H(M' \cdot x^T, \mathbf{0}^{m'}) \leq k$, $\mathbf{d}_H(M \cdot x^T, \mathbf{0}^{m}) = \mathbf{d}_H(M' \cdot x^T, \mathbf{0}^{m'})$. Furthermore, the matrix M' can be computed in time $\mathcal{O}(m \cdot n^{\omega-1} k)$, where ω is the matrix multiplication exponent ($\omega < 2.373$ [26]), assuming that the field operations take a constant time.*

Proof. In order to identify M' we identify $j_0 + 1 \leq k+1$ (non-empty) submatrices $B_0, B_1, \ldots, B_{j_0}$ of M, each having at most n rows, in the following way: First, let B_0 be a minimal submatrix of M whose rows span all the rows of M, that is, let B_0 be a base of the vector space generated by the rows of M, and let also M_0 be the submatrix obtained from M after removing the rows of B_0. We identify the rest of the matrices inductively as follows: For every $i \in [k]$, if M_{i-1} is not the empty matrix we let B_i be a minimal submatrix of M_{i-1} whose rows span all the rows of M_{i-1} and finally we let M_i be the matrix occurring from M_{i-1} after removing the rows of B_i.

We now define the submatrix M' of M. Let $j_0 \leq k$ be the greatest integer for which M_{j_0-1} is not the empty matrix. Let M' be the matrix consisting of the union of the rows of the (non-empty) matrices B_0 and B_i, $i \in [j_0]$. As the rank of the matrices M, M_i, $i \in [j_0]$, is upper bounded by n, the matrices B_0, B_i, $i \in [j_0]$, have at most n rows each, and therefore M' has at most $n(j_0+1) \leq n(k+1)$ rows. Observe that if $j_0 < k$ then the union of the rows of the non-empty matrices B_0, B_i, $i \in [j_0]$, contains all the rows of M and thus we may assume that $M' = M$ and the lemma trivially holds. Hence, it remains to prove the lemma for the case where $j_0 = k$, and therefore M' consists of the union of the matrices B_0, B_i, $i \in [k]$. As it always holds that $\mathbf{d}_H(M \cdot x^T, \mathbf{0}) \geq \mathbf{d}_H(M' \cdot x^T, \mathbf{0})$ it is enough to prove that for every $x \in \mathbb{F}^n$ for which $\mathbf{d}_H(M' \cdot x^T, \mathbf{0}) \leq k$, $\mathbf{d}_H(M \cdot x^T, \mathbf{0}) \leq \mathbf{d}_H(M' \cdot x^T, \mathbf{0})$. Thus, it is enough to prove that for every row r of the matrix M'' obtained from M after removing the rows of M', it holds that $\mathbf{d}_H(r \cdot x^T, \mathbf{0}) = 0$. Towards this goal let $x \in \mathbb{F}^n$ be a vector such that $\mathbf{d}_H(M' \cdot x^T, \mathbf{0}) \leq k$. From the Pigeonhole Principle there exists an i_0 such that $\mathbf{d}_H(B_{i_0} \cdot x^T, \mathbf{0}) = 0$, that is, if $r_1, r_2, \ldots, r_{|B_{i_0}|}$ are the rows of B_{i_0} then $r_j \cdot x^T = 0$, for every $j \in [|B_{i_0}|]$. Recall however that the row r of M'' is spanned by the rows $r_1, r_2, \ldots, r_{|B_{i_0}|}$ of B_{i_0}. Therefore, there exist $\lambda_j \in \mathbb{F}$, $j \in [|B_{i_0}|]$, such that

$r = \sum_{j \in [|B_{i_0}|]} \lambda_j r_j$. It follows that $r \cdot x^T = \sum_{j \in [|B_{i_0}|]} \lambda_j (r_j \cdot x^T) = 0$ and therefore $\mathbf{d}_H(r \cdot x^T, \mathbf{0}) = 0$. This implies that $\mathbf{d}_H(M \cdot x^T, \mathbf{0}) \leq \mathbf{d}_H(M' \cdot x^T, \mathbf{0})$. Finally, for a rectangular matrix of size $d \times s$, $d \leq s$, Ibarra et al. [12] give an algorithm that computes a maximal independent set of rows (a row basis) in $\mathcal{O}(d^{\omega-1}s)$ time. By running this algorithm $k+1$ times we can find the matrix M' in $\mathcal{O}(mn^{\omega-1}k)$ time and this completes the proof of the lemma. ☐

LEMMA 3.12. *Let $\mathbb{F}$ be a field. There exists an algorithm that given a set $\mathcal{S}$ of linear equations over $\mathbb{F}$ on n variables and an integer k outputs a set $\mathcal{S}' \subseteq \mathcal{S}$ of at most $(n+1)(k+1)$ linear equations over $\mathbb{F}$ such that any assignment of the variables that violates at most k linear equations of $\mathcal{S}'$ satisfies all the linear equations of $\mathcal{S} \setminus \mathcal{S}'$. Moreover, the running time of the algorithm is $\mathcal{O}(|\mathcal{S}|n^{\omega-1}k)$, assuming that the field operations take a constant time.*

Proof. Let $x_1, x_2, \ldots, x_n$ denote the n variables and α_{ij} denote the coefficient of x_j in the i-th linear equation of S, $i \in [|\mathcal{S}|]$, $j \in [n]$. Let also $\alpha_{i(n+1)}$ denote the constant term of the i-th linear equation of $\mathcal{S}$. In other words, the i-th equation of $\mathcal{S}$ is denoted as $\alpha_{i1}x_1 + \alpha_{i2}x_2 + \cdots + \alpha_{in}x_n + \alpha_{i(n+1)} = 0$. Finally, let M be the matrix where the j-element of the i-th row is α_{ij}, $i \in [|\mathcal{S}|]$, $j \in [n+1]$. From Lemma 3.11, it follows that for every positive integer k there exists a submatrix M' of M with at most $(n+1)(k+1)$ rows and $n+1$ columns such that for every $x \in \mathbb{F}^{n+1}$ for which $\mathbf{d}_H(M' \cdot x^T, \mathbf{0}) \leq k$, $\mathbf{d}_H(M \cdot x^T, \mathbf{0}) = \mathbf{d}_H(M' \cdot x^T, \mathbf{0})$ and M' can be computed in time $\mathcal{O}(|\mathcal{S}|n^{\omega-1}k)$. Let $\mathcal{S}'$ be the set of linear equations that correspond to the rows of M'. Let then $x_i = \beta_i$, $\beta_i \in \mathbb{F}$, $i \in [n]$, be an assignment that does not satisfy at most k of the equations of $\mathcal{S}'$. This implies that $\mathbf{d}_H(M' \cdot z, \mathbf{0}) \leq k$, where $z = (\beta_1, \beta_2, \ldots, \beta_n, 1)^T$. Again, from Lemma 3.11, we get that $\mathbf{d}_H(M \cdot z, \mathbf{0}) = \mathbf{d}_H(M' \cdot z, \mathbf{0})$. Thus, the above assignment satisfies all the linear equations of $\mathcal{S} \setminus \mathcal{S}'$. ☐

The requirement of constant time operations in field $\mathbb{F}$ in the above two lemmata might seem restrictive with respect to the standard field of reals (or rationals). This is necessary due to use of the algorithm of Ibarra et al. [12], which given a rectangular matrix of size $d \times s$, $d \leq s$, computes a maximal independent set of rows in $\mathcal{O}(d^{\omega-1}s)$ time. While this is not needed for our results, for reals we can replace the algorithm of Ibarra et al. by the following slower, yet still polynomial time procedure.

Start with an empty set B' of rows and taking the rows of the input matrix one by one, add them to the set B' if they are linearly independent of the rows currently in the set B'. Obviously, this way we obtain a maximal idenpendent set of rows. To check the linear independence of row r from a set B' formed by rows $r_1, \ldots, r_{|B'|}$, one has to decide, whether there are $\lambda_1, \ldots, \lambda_{|B'|} \in \mathbb{R}$ such that $\sum_{i=1}^{|B'|} \lambda_i r_i = r$. However, this can be formulated as an instance of linear programming, namely whether there is λ such that $B'^T \lambda^T = r$, i.e.,

$$\begin{pmatrix} B'^T \\ -B'^T \end{pmatrix} \lambda^T \leq \begin{pmatrix} r \\ -r \end{pmatrix}.$$

Here B'^T is a matrix having columns $r_1^T, r_2^T, \cdots, r_{|B'|}^T$. Since the feasibility of linear programming can be determined in polynomial time in the number of variables and length of the description [13], this gives polynomial time variants of the above lemmata for the field of reals. We would further like to remark here that the reason we choose to work over a finite field instead of, for example, the rationals is for purely technical reasons as it permits us to avoid potentially running into numbers that need exponentially many bits to represent.

3.4. The Main Theorem. In this subsection by combining the structural decomposition of Subsection 3.2 and Lemma 3.12 from Subsection 3.3 we obtain a kernel for wTDS of size $\mathcal{O}(k^4)$.

THEOREM 3.13. wTDS *admits a kernel with* $\mathcal{O}(k^4)$ *vertices and edges and encoding-size* $\mathcal{O}(k^4 \log k)$ *bits.*

Proof. Let (G, w, k) be an instance of wTDS. Without loss of generality we may assume that it is semi-reduced, G is connected, and that, from Lemma 3.4, $V(G)$ can be partitioned into three sets B, T, and I satisfying the conditions of Lemma 3.4. Note that, as G is connected, every vertex of I has at least one neighbor in B. We construct an instance (G', w', k) of wTDS in the following way. Let p be a prime number such that $|B| < p < 2|B|$. Such a prime number exists by a Bertrand's postulate (proved by Chebyshev in 1850). Let $\mathbb{F} = \mathbb{GF}(p)$, that is, the Galois field of order p. It takes at most $O(|B|^2) = O(k^4)$ time to find p and the multiplicative inverses in $\mathbb{F}$.

Let $I = \{v_i \mid i \in [|I|]\}$ and $B = \{u_j \mid j \in [|B|]\}$. We assign an $\mathbb{F}$-variable x_j to u_j, $j \in [|B|]$, and a linear equation l_i over $\mathbb{F}$ to v_i, $i \in [|I|]$, where l_i is the equation $\sum_{j\in[|B|]} \alpha_{ij} x_j - 1 = 0$ and $\alpha_{ij} = 1$ if $u_j \in N_G(v_i)$ and 0 otherwise. Let $\mathcal{L} = \{l_i \mid i \in [|I|]\}$ and $\mathcal{L}'$ be the subset of $\mathcal{L}$ obtained from Lemma 3.12. Let also $I' = \{v_p \in I \mid l_p \in \mathcal{L}'\}$ and $G' = G[B \cup T \cup I']$. Finally, let $w' = w|_{B\cup T\cup I'}$. We now prove that (G', w', k) is equivalent to (G, w, k).

We first prove that if (G, w, k) is a yes-instance then so is (G', w', k). Let S be a tree deletion set of G of weight at most k. Then $G \setminus S$ is a tree and, as for every vertex $v \in I \setminus S$, $N_G(v)$ is a double clique, v has degree exactly 1 in $G \setminus S$. Therefore, the graph obtained from $G \setminus S$ after removing $(I \setminus I')$ is still a tree. This implies that $S \setminus (I \setminus I')$ is a tree deletion set of G' of weight at most k and (G', w', k) is a yes-instance.

Let now (G', w', k) be a yes-instance and S be a tree deletion set of G' of weight at most k. We claim that there exist at most k vertices in I' whose neighborhood lies entirely in S. Indeed, assume to the contrary that there exist at least $k+1$ vertices of I' whose neighborhood lies entirely in S. Let J be the set of those vertices. Notice that for every vertex $v \in I'$, if $N_{G'}(v) \subseteq S$, then either $v \in S$ or $I' \setminus \{v\} \subseteq S$. Notice that if $J \subseteq S$, then S has weight at least $k+1$, a contradiction. Therefore, there exists a vertex $u \in J$ that is not contained in S. Then $I' \setminus \{u\} \subseteq S$. Moreover, recall that u has at least one neighbor z in B and from the hypothesis z is contained in S. Therefore $(I' \setminus \{u\}) \cup \{z\} \subseteq S$. As $|I'| \geq |J| = k+1$, it follows that $|I' \setminus \{u\}| \geq k$. Furthermore, recall that $B \cap I' = \emptyset$. Thus, $|S| \geq k+1$, a contradiction to the fact that S has weight at most k. Therefore, there exist at most k vertices of I' whose neighborhood is contained entirely in S. For every $j \in [|B|]$, let $x_j = \beta_j$, where $\beta_j = 0$ if $u_j \in S$ and 1 otherwise. Then there exist at most k linear equations in $\mathcal{L}'$ which are not satisfied by the above assignment. However, from the choice of $\mathcal{L}'$ all the linear equations in $\mathcal{L} \setminus \mathcal{L}'$ are satisfied and therefore, for every vertex u in $I \setminus I'$ we have $|N_G(u) \setminus S| \equiv 1 \pmod{p}$. Since $p > |B|$ this implies that u has exactly one neighbor in $G \setminus S$. Thus $G \setminus S$ is a tree and hence, S is a tree deletion set of G as well.

Notice that $V(G') = B \cup T \cup I'$, where $|I'| \leq 8k^3 + 10k^2 + 3k + 1$ (Lemma 3.12) and therefore $|V(G')| = \mathcal{O}(k^4)$. It is also easy to see that $|E(G')| = \mathcal{O}(k^4)$. Indeed, notice first that as the set I' is an independent set there are no edges between its vertices. Moreover, from Lemma 3.4 there are no edges between the vertices of the set I' and the set T. Observe that, from the construction of I and subsequently of I', Lemma 3.4 implies that every vertex of I' has at most $2k+2$ neighbors in B. As

$|I'| \leq 8k^3 + 10k^2 + 3k + 1$ there exist $\mathcal{O}(k^4)$ edges between the vertices of I' and the vertices of B. Notice that from (ii) of Lemma 3.4, T induces a forest and thus there exist at most $\mathcal{O}(k^4)$ edges between its vertices. Moreover, from (i) of Lemma 3.4, again there exist $\mathcal{O}(k^4)$ edges between the vertices of B. It remains to show that there exist $\mathcal{O}(k^4)$ edges with one endpoint in B and one endpoint in T. Recall first that every connected component has at most 2 neighbors in $\widehat{Q}$. Therefore, there exist at most $2k+2$ edges between every connected component of $\mathcal{C}_T$ and B. Moreover, from Lemma 3.8 we obtain that $\mathcal{C}_T$ contains $\mathcal{O}(k^3)$ connected components. Therefore, there exist $\mathcal{O}(k^4)$ edges with one endpoint in B and one endpoint in T. Thus, wTDS has a kernel of $\mathcal{O}(k^4)$ vertices and edges. Finally, from Reduction Rule 5, the weight of every vertex is upper bounded by $k+1$ and thus, it can be encoded using $\log(k+1)$ bits resulting to a kernel of wTDS with $\mathcal{O}(k^4 \log k)$ bits. $\square$

From Lemma 3.2 we immediately get the following.

COROLLARY 3.14. TREE DELETION SET *has a kernel with $\mathcal{O}(k^5)$ vertices and edges.*

4. Conclusions. As mentioned in the Introduction, TREE DELETION SET is not expected to admit an approximation algorithm with ratio $\mathrm{OPT}^{\mathcal{O}(1)}$ and as a result it had been tempting to argue that it also does not admit a polynomial kernel. The reason for this, it that for many `FPT` graph problems that admit a polynomial kernel it is easy to also obtain an approximation algorithm for them as the reduction rules designed for their kernels are also *approximation-preserving*. However, even though most of our reduction rules for Tree Deletion Set are also approximation-preserving, this is not the case for the last reduction that takes place on Theorem 3.13 and is based on Lemma 3.12, as it disregards an arbitrary number of connected components of the graph whose vertices should be included in an approximate solution. Thus, Lemma 3.12, and the transformation of part of our instance to a system of linear equations is what drives us away from an approximation algorithm for TREE DELETION SET yet allows us to derive a polynomial kernel for it.

Acknowledgements. We would like to thank the anonymous referees of this and of previous versions whose comments improved the overall presentation of our result.

REFERENCES

[1] VINEET BAFNA, PIOTR BERMAN, AND TOSHIHIRO FUJITO, *A 2-approximation algorithm for the undirected feedback vertex set problem*, SIAM J. Discrete Math., 12 (1999), pp. 289–297.

[2] HANS L. BODLAENDER AND THOMAS C. VAN DIJK, *A cubic kernel for feedback vertex set and loop cutset*, Theory Comput. Syst., 46 (2010), pp. 566–597.

[3] KEVIN BURRAGE, VLADIMIR ESTIVILL-CASTRO, MICHAEL R. FELLOWS, MICHAEL A. LANGSTON, SHEV MAC, AND FRANCES A. ROSAMOND, *The undirected feedback vertex set problem has a poly(k) kernel*, in Parameterized and Exact Computation - IWPEC, vol. 4169 of LNCS, 2006, pp. 192–202.

[4] YIXIN CAO, JIANER CHEN, AND YANG LIU, *On feedback vertex set: New measure and new structures*, Algorithmica, 73 (2015), pp. 63–86.

[5] JIANER CHEN, FEDOR V. FOMIN, YANG LIU, SONGJIAN LU, AND YNGVE VILLANGER, *Improved algorithms for feedback vertex set problems*, J. Comput. Syst. Sci., 74 (2008), pp. 1188–1198.

[6] R. CROWSTON, M. FELLOWS, G. GUTIN, M. JONES, E.J. KIM, F. ROSAMOND, I.Z. RUZSA, S. THOMASS, AND A. YEO, *Satisfying more than half of a system of linear equations over GF(2): A multivariate approach*, Journal of Computer and System Sciences, 80 (2014), pp. 687–696.

[7] Rodney G. Downey and Michael R. Fellows, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013.

[8] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh, *Planar F-deletion: Approximation, kernelization and optimal FPT algorithms*, in Foundations of Computer Science - FOCS 2012, 2012, pp. 470–479.

[9] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, 1979.

[10] Archontia C. Giannopoulou, Daniel Lokshtanov, Saket Saurabh, and Ondrej Suchý, *Tree deletion set has a polynomial kernel (but no opt^o(1) approximation)*, in 34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India, 2014, pp. 85–96.

[11] Jiong Guo and Rolf Niedermeier, *Invitation to data reduction and problem kernelization*, SIGACT News, 38 (2007), pp. 31–45.

[12] Oscar H. Ibarra, Shlomo Moran, and Roger Hui, *A generalization of the fast LUP matrix decomposition algorithm and applications*, J. Algorithms, 3 (1982), pp. 45–56.

[13] L.G. Khachiyan, *Polynomial algorithms in linear programming*, USSR Computational Mathematics and Mathematical Physics, 20 (1980), pp. 53–72.

[14] Subhash Khot and Oded Regev, *Vertex cover might be hard to approximate to within* $2-\varepsilon$, J. Comput. Syst. Sci., 74 (2008), pp. 335–349.

[15] Stefan Kratsch, *Polynomial kernelizations for MIN* $F^+\Pi$ *and MAX NP*, Algorithmica, 63 (2012), pp. 532–550.

[16] Stefan Kratsch, *Recent developments in kernelization: A survey*, Bulletin of the EATCS, 113 (2014).

[17] Stefan Kratsch and Magnus Wahlström, *Preprocessing of Min Ones problems: A dichotomy*, in Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I, vol. 6198 of Lecture Notes in Computer Science, Springer, 2010, pp. 653–665.

[18] Stefan Kratsch and Magnus Wahlström, *Representative sets and irrelevant vertices: New tools for kernelization*, in Foundations of Computer Science - FOCS 2012, 2012, pp. 450–459.

[19] Stefan Kratsch and Magnus Wahlström, *Compression via matroids: A randomized polynomial kernel for odd cycle transversal*, ACM Trans. Algorithms, 10 (2014), pp. 20:1–20:15.

[20] Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh, *Kernelization - preprocessing with a guarantee*, in The Multivariate Algorithmic Revolution and Beyond, vol. 7370 of LNCS, 2012, pp. 129–161.

[21] Dániel Marx, *Parameterized complexity and approximation algorithms*, The Computer Journal, 51 (2008), pp. 60–78.

[22] Rolf Niedermeier, *Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications)*, Oxford University Press, USA, March 2006.

[23] Venkatesh Raman, Saket Saurabh, and Ondřej Suchý, *An FPT algorithm for tree deletion set*, Journal of Graph Algorithms and Applications, 17 (2013), pp. 615–628.

[24] Stéphan Thomassé, *A* $4k^2$ *kernel for feedback vertex set*, ACM Transactions on Algorithms, 6 (2010).

[25] Magnus Wahlström, *Abusing the Tutte matrix: An algebraic instance compression for the K-set-cycle problem*, in Symposium on Theoretical Aspects of Computer Science - STACS 2013, vol. 20 of LIPIcs, 2013, pp. 341–352.

[26] Virginia Vassilevska Williams, *Multiplying matrices faster than Coppersmith-Winograd*, in Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012, ACM, 2012, pp. 887–898.

[27] Mihalis Yannakakis, *The effect of a connectivity requirement on the complexity of maximum subgraph problems*, J. ACM, 26 (1979), pp. 618–630.