A 13k-kernel for Planar Feedback Vertex Set via Region Decomposition[*]

Marthe Bonamy[†]   Łukasz Kowalik[‡]

Abstract

We show a kernel of at most 13k vertices for the Feedback Vertex Set problem restricted to planar graphs, i.e., a polynomial-time algorithm that transforms an input instance (G, k) to an equivalent instance with at most 13k vertices. To this end we introduce a few new reduction rules. However, our main contribution is an application of the region decomposition technique in the analysis of the kernel size. We show that our analysis is tight, up to a constant additive term.

1 Introduction

A *feedback vertex set* in a graph $G = (V, E)$ is a set of vertices $S \subseteq V$ such that $G - S$ is a forest. In the Feedback Vertex Set problem, given a graph G and integer k one has to decide whether G has a feedback vertex set of size k. This is one of the fundamental NP-complete problems, in particular it is among the 21 problems considered by Karp [11]. It has applications e.g. in operating systems (see [15]), VLSI design, synchronous systems and artificial intelligence (see [8]).

In this paper we study kernelization algorithms, i.e., polynomial-time algorithms which, for an input instance (G, k) either conclude that G has no feedback vertex set of size k or return an equivalent instance (G', k'), called *kernel*. In this paper, by the size of the kernel we mean the number of vertices of G'. Burrage et al. [5] showed that Feedback Vertex Set has a kernel of size $O(k^{11})$, which was next improved to $O(k^3)$ by Bodlaender [3] and to $4k^2$ by Thomassé [16]. Actually, as argued by Dell and van Melkebeek [7] the kernel of Thomassé can be easily tuned to have the number of *edges* bounded by $O(k^2)$. This cannot be improved to $O(k^{2-\epsilon})$ for any $\epsilon > 0$, unless coNP $\subseteq$ NP$/$poly [7].

In this paper we study Planar Feedback Vertex Set problem, i.e., Feedback Vertex Set restricted to planar graphs. Planar versions of NP-complete graph problems often enjoy kernels with $O(k)$ vertices. Since an n-vertex planar graph has $O(n)$ edges, this implies they have $O(k)$ edges, and hence are called *linear kernels*. The first nontrivial result of that kind was presented in the seminal work of Alber, Fellows and Niedermeier [2] who showed a kernel of size 335k for Planar Dominating Set. One of the key concepts of their paper was the region decomposition technique in the analysis of the kernel size. Roughly, in this

---

[†]LIRMM, France

[‡]University of Warsaw, Poland

method the reduced plane instance is decomposed into $O(k)$ *regions* (i.e. subsets of the plane) such that every region contains $O(1)$ vertices of the graph. It was next applied by Guo and Niedermeier to a few more graph problems [10]. In fact it turns out that for a number of problems on planar graphs, including PLANAR DOMINATING SET and PLANAR FEEDBACK VERTEX SET, one can get a kernel of size $O(k)$ by general method of protrusion decomposition [9]. However, in this general algorithm the constants hidden in the O notation are very large, and researchers keep working on problem-specific linear kernels with the constants as small as possible [6, 14, 17, 13, 12].

In the case of PLANAR FEEDBACK VERTEX SET, Bodlaender and Penninkx [4] gave an algorithm which outputs a kernel of size at most $112k$. This was next improved by Abu-Khzam and Khuzam [1] to $97k$. Very recently, and independently of our work, Xiao [18] has presented an improved kernel of $29k$ vertices. However, neither of these papers uses the region decomposition. Indeed, it seems non-obvious how the regions of the region decomposition can be defined for PLANAR FEEDBACK VERTEX SET. Instead, the authors of the previous works cleverly apply simple bounds on the number of edges in general and bipartite planar graphs. Moreover, for certain problems these methods turned out to give better results and simpler proofs than those based on region decomposition, see e.g., the work of Wang, Yang, Guo and Chen [17] on CONNECTED VERTEX COVER, EDGE DOMINATING SET, and TRIANGLE PACKING in planar graphs improving previous results of Guo and Niedermeier [10].

Somewhat surprisingly, in this work we show that region decomposition can be successfully applied to PLANAR FEEDBACK VERTEX SET, and moreover it gives much tighter bounds than the previous methods. Furthermore, we add a few new reduction rules to improve the bound even further, to $13k$. More precisely, we show the following result.

Theorem 1. *There is an algorithm that, given an instance (G, k) of* PLANAR FEEDBACK VERTEX SET*, either reports that G has no feedback vertex set of size k or produces an equivalent instance with at most $13k - 24$ vertices. The algorithm runs in expected $O(n)$ time, where n is the number of vertices of G.*

We use the region decomposition approach in a slightly relaxed way: the regions are the faces of a k-vertex plane graph and the number of vertices of the reduced graph in each region is linear in the length of the corresponding face. We show that this gives a *tight bound*, i.e., we present a family of graphs which can be returned by our algorithm and have $13k - O(1)$ vertices.

Organization of the paper. In Section 2 we present a kernelization algorithm which is obtained from the algorithms in [4, 1] by generalizing a few reduction rules, and adding some completely new rules. In Section 3 we present an analysis of the size of the kernel obtained by our algorithm. In the analysis we assume that in the reduced graph, for every induced path with ℓ internal vertices, the internal vertices have at least three neighbors outside the path. Based on this, we get the bound of $(2\ell + 3)k - (4\ell + 4)$ for the number of vertices in the kernel. In Section 2 we present reduction rules which guarantee that in the kernel $\ell \leq 6$, resulting in the kernel size bound of $15k - 28$. To get the claimed bound of $13k - 24$ vertices in Section 4 we present a complex set of reduction rules, which allow us to conclude that $\ell \leq 5$. In Section 5 we discuss the running time of the algorithm. Finally, in Section 6 we discuss possibilities of further research.

Notation. In this paper we deal with multigraphs, though for simplicity we refer to them as graphs. (Even if the input graph is simple, our algorithm may introduce multiple edges.)
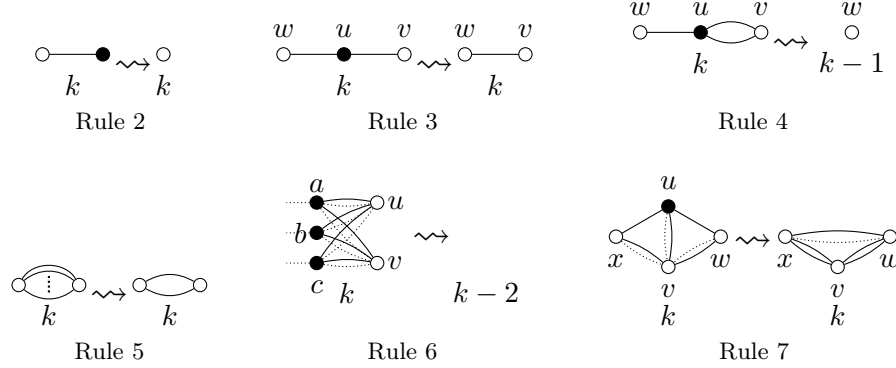
Figure 1: Reduction rules 1–7. Dashed edges are optional. We draw in black the vertices whose incident edges are all already drawn (as solid or dashed edges), in white the vertices which might be incident to other edges. Regardless of their color, vertices in the figures may not coincide.

By the *degree* of a vertex x in a multigraph G, denoted by $\deg_G(x)$, we mean the number of edges incident to x in G. By $N_G(x)$, or shortly $N(x)$, we denote the set of neighbors of x, while $N[x] = N(x) \cup \{x\}$ is the closed neighborhood of x. Note that in a multigraph $|N_G(x)| \leq \deg_G(x)$, but the equality does not need to hold. The neighborhood of a set of vertices S is defined as $N(S) = (\bigcup_{v \in S} N(v)) \setminus S$, while the closed neighborhood of S is $N[S] = (\bigcup_{v \in S} N(v)) \cup S$. For a face f in a plane graph, a *facial walk* of f is the shortest closed walk induced by all edges incident with f. The *length* of f, denoted by $d(f)$ is the length of its facial walk.

2   Our kernelization algorithm

In this section we describe our algorithm which outputs a kernel for PLANAR FEEDBACK VERTEX SET. The algorithm exhaustively applies reduction rules. Each reduction rule is a subroutine which finds in polynomial time a certain structure in the graph and replaces it by another structure, so that the resulting instance is equivalent to the original one. More precisely, we say that a reduction rule for parameterized graph problem P is *correct* when for every instance (G, k) of P it returns an instance (G', k') such that:

a) (G', k') is an instance of P,

b) (G, k) is a yes-instance of P iff (G', k') is a yes-instance of P, and

c) $k' \leq k$.

Below we state the rules we use. The rules are applied in the given order, i.e., in each rule we assume that the earlier rules do not apply. We begin with some rules used in the previous works [1, 4].

Rule 1 If there is a loop at a vertex v, remove v and decrease k by one.

Rule 2 Delete vertices of degree at most one.

Rule 3 If a vertex u is of degree two, with incident edges uv and uw, then delete u and add the edge vw. (Note that if $v = w$ then a loop is added.)

Rule 4 If a vertex u has exactly two neighbors v and w, edge uv is double, and edge uw is simple, then delete v and u and decrease k by one.

Rule 5 If there are at least three edges between a pair of vertices, remove all but two of the edges.

Rule 6 Assume that there are five vertices a, b, c, v, w such that 1) both v and w are neighbors of each of a, b, c and 2) each vertex $x \in \{a, b, c\}$ is incident with at most one edge xy such that $y \notin \{v, w\}$. Then remove all the five vertices and decrease k by two.

The correctness of the above reduction rules was proven in [1]. (In [1], Rule 6 is formulated in a slightly less general way which forbids multiplicity of some edges, but the correctness proof stays the same.) Now we introduce a few new rules.

Rule 7 If a vertex u has exactly three neighbors v, w and x, v is also adjacent to w and x, and both edges uw and ux are simple, then contract uv and add an edge wx (increasing its multiplicity if it already exists). If edge uv was not simple, add a loop at v.

Lemma 2. *Rule 7 is correct.*

Proof. Let G' be the graph obtained from a graph G by a single application of Rule 7. Let S be a feedback vertex set of size k in G'. We claim S is a feedback vertex set in G too. Assume for a contradiction that there is a cycle C in $G - S$. Then $u \in V(C)$, for otherwise $C \subseteq G'$. If $v \in S$ then $\{wu, ux\} \subseteq C$ and $C - \{wu, ux\} + \{wx\}$ is a cycle in G', a contradiction. If $v \notin S$, then $w, x \in S$ and hence v is the only neighbor of u in $G - S$, so C is the 2-cycle uvu. But then $G' - S$ contains a loop at v, a contradiction.

Let S be a feedback vertex set of size k in G. If $|\{u, v\} \cap S| = 2$, then $S \setminus \{u\} \cup \{w\}$ is a feedback vertex set of size k in G'. Assume $|\{u, v\} \cap S| = 1$. Then we can assume $v \in S$ for otherwise we replace S by $S \setminus \{u\} \cup \{v\}$, which is also a feedback vertex set in G. If there is a cycle C in $G' - S$, then $wx \in E(C)$, for otherwise $C \subseteq G - S$. But then $C - \{wx\} + \{wu, ux\}$ is a cycle in G, a contradiction. Finally, if $|\{u, v\} \cap S| = 0$ then both w and x are in S, so S is also a feedback vertex set in G'. $\qquad\square$

The graph modification in Rule 7 is an example of a *gadget replacement*, i.e., a subgraph of G is replaced by another subgraph in such a way that the answer to the FEEDBACK VERTEX SET problem does not change. We will use many rules of this kind, and their correctness proofs all use similar arguments. In order to make our proofs more compact, we define gadget replacement formally below, and prove a technical lemma (Lemma 3 below) which will be used in many rule correctness proofs.

Gadget replacement in graph A is a triple (X, Y, E_I), where $X \subseteq V(A)$, Y is a set of vertices disjoint with $V(A)$, and E_I is a set of edges with both endpoints in $N_A(X) \cup Y$. The *result* of gadget replacement is a new graph B, obtained from A by deleting X and $E(G[N[X]])$ and inserting Y and E_I.

For an example, in Rule 7, G' is a result of gadget replacement $(\{u\}, \emptyset, \{xv, wv, xw\})$. Note that if (X, Y, E_I) is a gadget replacement in A that results in B then $(Y, X, \{uv \in E(A) : u, v \in N_A[X]\})$ is a gadget replacement in B and its result is A.

Lemma 3. *Let (X, Y, E_I) be a gadget replacement in graph A, and let B be its result. Let $Q_A = A[N_A[X]]$ and $Q_B = B[Y \cup N_A(X)]$. Let S_A be a feedback vertex set in A. Let S_B be a subset of vertices of B such that $S_A \setminus V(Q_A) = S_B \setminus V(Q_B)$ and $Q_B - S_B$ is a forest. Finally, assume that for every pair $u, v \in N_A(X)$ if there is a (u, v)-path in $Q_B - S_B$ then there is a (u, v)-path in $Q_A - S_A$. Then S_B is a feedback vertex set of B.*
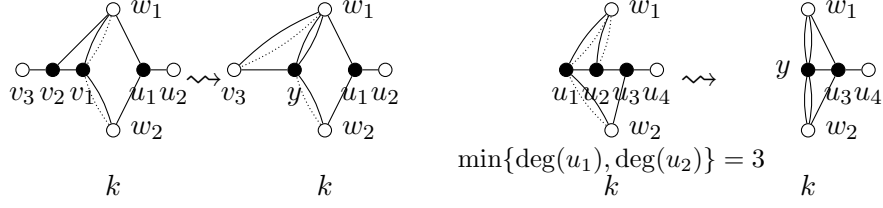
4

Figure 2: Reduction rules 8 and 9.

Proof. Assume for a contradiction that there is a cycle C in $B - S_B$. Since $Q_B - S_B$ is a forest, C has at least one vertex outside Q_B. Assume C has all vertices outside Q_B. But then $C \subseteq A$ and since $S_A \setminus V(Q_A) = S_B \setminus V(Q_B)$ we also have $C \subseteq A - S_A$, so S_A is not a feedback vertex set of A, a contradiction.

Hence we know that C has vertices both inside and outside Q_B. It follows that C can be divided into subpaths of two kinds: subpaths in $Q_B - S_B$ and in $(B - E(Q_B)) - S_B$. Every such subpath P_B in $Q_B - S_B$ is of the form $v_1, \ldots, v_t$, where $v_1, v_t \in N_A(X)$ and $v_2, \ldots, v_{t-1} \in V(Q_B) - S_B$. Hence, by the assumption of the lemma, there is a (v_1, v_t)-path P_A in $Q_A - S_A$. In particular, $v_1, v_t \notin S_A$ (we will use this observation later). Let C' be the closed walk obtained from C by replacing every maximal subpath in $Q_B - S_B$ by a path in $Q_A - S_A$. Now consider a maximal subpath P'_B of C in $(B - E(Q_B)) - S_B$. It is of the form $v_1, \ldots, v_t$, where $v_1, v_t \in N_A(X)$ and $v_2, \ldots, v_{t-1} \in (V(B) \setminus V(Q_B)) \setminus S_B$. Since v_1 and v_t are also endpoints of paths in $Q_B - S_B$, it holds that $v_1, v_t \notin S_A$ as argued above. Since $S_A \setminus V(Q_A) = S_B \setminus V(Q_B)$, it holds that $v_2, \ldots, v_{t-1} \in V(A) \setminus S_A$. Hence $P'_B \subseteq A - S_A$. It follows that C' is a closed walk in $A - S_A$, hence there is a cycle in $A - S_A$, a contradiction. $\square$

When applying Lemma 3, we will examine reachability relations in $Q_A - S_A$ and in $Q_B - S_B$, in order to check whether the last assumption holds. It will be convenient to introduce the following notation. For a graph H and set of vertices S, let $R_{H,S}$ be the reachability relation in H truncated to S, i.e., $(a, b) \in R_{H,S}$ iff $a, b \in S$ and there is an (a, b)-path in H. The set S does not need to be a subset of $V(H)$; for every vertex $a \in S \setminus V(H)$, $\{a\}$ forms an equivalence class of $R_{H,S}$.

Rule 8 Assume there are six vertices $v_1, v_2, v_3, u_1, u_2, w_1, w_2$, such that $N(u_1) = \{w_1, w_2, u_2\}$, $N(v_1) = \{w_1, w_2, v_2\}$, $N(v_2) = \{w_1, v_1, v_3\}$, and $\deg(v_2) = \deg(u_1) = 3$. Then contract the edge v_1v_2 to a new vertex y and add an edge w_1v_3, as presented in Figure 2 (left).

Lemma 4. *Rule 8 is correct.*

Proof. Let G' be the graph obtained from a graph G by a single application of Rule 8. Note that G' is a result of a gadget replacement (X, Y, E_I) with $X = \{v_1, v_2, u_1\}$ and $Y = \{y, u_1\}$

Let S be a feedback vertex set of size k in G. We claim that there is a feedback vertex set S' in G' of size at most k. If $|S \cap \{w_1, w_2, v_1, v_2, u_1\}| \geq 2$, then by Lemma 3 we see that $S' = (S \setminus \{w_1, w_2, v_1, v_2, u_1\}) \cup \{w_1, w_2\}$ works. Hence we can assume $|S \cap \{w_1, w_2, v_1, v_2, u_1\}| \leq 1$. Then $S' \cap \{w_1, w_2, v_1, v_2, u_1\} = \{w_1\}$ or $S' \cap \{w_1, w_2, v_1, v_2, u_1\} = \{v_1\}$ to hit the triangle $v_1v_2w_1$ and the quadrangle $v_1w_1u_1w_2$. In the prior case we pick $S = S'$. Then $R_{G[N_G[X]] - S, \{w_1, w_2, u_2, v_3\}}$ and $R_{G'[N_{G'}[Y]] - S', \{w_1, w_2, u_2, v_3\}}$ are the same relations since the $v_3v_2v_1$ path in G corresponds to the v_3y edge in G', so Lemma 3 applies. We are left with the case $S' \cap \{w_1, w_2, y_1, u_1\} = \{v_1\}$. Then we pick $S = (S' \setminus \{v_1\}) \cup \{y\}$. Again, $R_{G[N_G[X]] - S, \{w_1, w_2, u_2, v_3\}} = R_{G'[N_{G'}[Y]] - S', \{w_1, w_2, u_2, v_3\}}$ since the $v_3v_2w_1$ path in G corresponds to the v_3w_1 edge in G', so Lemma 3 applies.

5

Let S' be a feedback vertex set of size k in G'. We claim that there is a feedback vertex set S in G of size at most k. If $|S' \cap \{w_1, w_2, y_1, u_1\}| \geq 2$, then by Lemma 3 we see that $S = (S' \setminus \{y_1, y_2\}) \cup \{w_1, w_2\}$ works. Hence we can assume $|S' \cap \{w_1, w_2, y_1, u_1\}| \leq 1$. Then $S' \cap \{w_1, w_2, y_1, u_1\} = \{w_1\}$ or $S' \cap \{w_1, w_2, y_1, u_1\} = \{y\}$ to hit the digon $w_1 y$. In the prior case we pick $S = S'$. Then $R_{G[N_G[X]]-S, \{w_1, w_2, u_2, v_3\}} = R_{G'[N_{G'}[Y]]-S', \{w_1, w_2, u_2, v_3\}}$ since the $v_3 v_2 v_1$ path in G corresponds to the $v_3 y$ edge in G', so Lemma 3 applies. We are left with the case $S' \cap \{w_1, w_2, y_1, u_1\} = \{y\}$. Then we pick $S = S' \setminus \{y\} \cup \{v_1\}$. Again, $R_{G[N_G[X]]-S, \{w_1, w_2, u_2, v_3\}} = R_{G'[N_{G'}[Y]]-S', \{w_1, w_2, u_2, v_3\}}$ since the $v_3 v_2 w_1$ path in G corresponds to the $v_3 w_1$ edge in G', so Lemma 3 applies. $\square$

Rule 9 Assume $u_1 u_2 u_3 u_4$ is an induced path such that for two vertices w_1, w_2 outside the path, $N(u_1) = \{u_2, w_1, w_2\}$, $N(u_2) = \{u_1, u_3, w_1\}$ and $N(u_3) = \{u_2, u_4, w_2\}$, $\deg(u_3) = 3$, and $\min\{\deg(u_1), \deg(u_2)\} = 3$. Then replace $G[\{u_1, u_2, u_3, w_1, w_2\}]$ with the gadget presented in Figure 2 (right), i.e., remove u_1 and u_2 and add a vertex y, edges $y u_3$ and $u_3 w_1$, and double edges $y w_1$ and $y w_2$.

Lemma 5. *Rule 9 is correct.*

Proof. Let G' be the graph obtained from a graph G by a single application of Rule 9. Note that G' is a result of a gadget replacement (X, Y, E_I) with $X = \{u_1, u_2\}$ and $Y = \{y\}$

Let S be a solution of (G, k). If $|S \cap \{w_1, w_2, u_1, u_2, u_3\}| \geq 2$ or $S \cap \{w_1, w_2, u_1, u_2, u_3\} = \{w_1\}$ we proceed as in the proof of Lemma 4. Otherwise, to hit the triangle $w_1 u_1 u_2$, $S \cap \{w_1, w_2, u_1, u_2, u_3\}$ equals either $\{u_1\}$ or $\{u_2\}$. In both cases, $R_{G[N_G[X]]-S, \{w_1, w_2, u_3\}}$ has exactly one equivalence class $\{w_1, w_2, u_3\}$. We observe that for $S' = S \setminus \{u_1, u_2\} \cup \{y\}$ the relation $R_{G'[N_{G'}[Y]]-S', \{w_1, w_2, u_3\}}$ has also one equivalence class, so by Lemma 3, S' is a solution of (G', k).

Let S' be a feedback vertex set of size k in G'. If $|S' \cap \{w_1, w_2, y, u_3\}| \geq 2$ we proceed as in the proof of Lemma 4. Otherwise, $S' \cap \{w_1, w_2, u_1, u_2, u_3\} = \{y\}$. If $\deg_G(u_1) = 3$ then we put $S = S' \setminus \{y\} \cup \{u_2\}$, and otherwise $S = S' \setminus \{y\} \cup \{u_1\}$. Note that $G[\{w_1, w_2, u_1, u_2, u_3\}] - S$ is a forest, since $\min\{\deg(u_1), \deg(u_2)\} = 3$. Moreover, $R_{G[N_G[X]]-S, \{w_1, w_2, u_3\}}$ and $R_{G'[N_{G'}[Y]]-S', \{w_1, w_2, u_3\}}$ are the same (total) relation, so Lemma 3 applies and (S, k) is a solution of (G, k). $\square$

Rule 10 Let $A \subseteq V(G)$ and let w_1 and w_2 be two vertices in G, $w_1, w_2 \notin A$. If (i) no cycle in $G \setminus \{w_1, w_2\}$ intersects A, and (ii) there is a subgraph $Q \subseteq G[A \cup \{w_1, w_2\}]$ such for every vertex $x \in V(Q) \setminus \{w_1\}$, we have $\deg_Q(x) \leq |E(Q)| - |A| - 1$, then remove w_1 and decrease k by 1.

Lemma 6. *Rule 10 is correct.*

Proof. Let G' be the graph obtained from a graph G by a single application of Rule 10, i.e., $G' = G - w_1$. Let S be a feedback vertex set of size $k - 1$ in G'. Then every cycle in $G - S$ contains w_1, so $S \cup \{w_1\}$ is a feedback vertex set of size k in G.

Let S be a feedback vertex set of size k in G. If $w_1 \in S$, then clearly $S \setminus \{w_1\}$ is a solution of the instance $(G', k-1)$. Hence assume $w_1 \notin S$. We claim that $|S \cap V(Q)| \geq 2$. Assume the contrary, i.e., $|S \cap V(Q)| \leq 1$. Since $Q - S$ is a forest,

$$|E(Q-S)| \leq |V(Q-S)| - 1 = |V(Q)| - |S \cap V(Q)| - 1 = |A| + 1 - |S \cap V(Q)|. \quad (1)$$
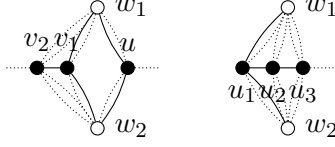
6

Figure 3: Configurations in lemmas 7 and 8.

On the other hand, by the degree bound, and because $w_1 \notin S$ and $|S \cap V(Q)| \leq 1$,

$$|E(Q - S)| \geq |E(Q)| - (|E(Q)| - |A| - 1)|S \cap V(Q)|. \tag{2}$$

By (1) and (2), $|A| + 1 \geq |E(Q)| - (|E(Q)| - |A| - 2)|S \cap V(Q)|$. Since $|S \cap V(Q)| \leq 1$ this implies $|A| + 1 \geq |E(Q)| - (|E(Q)| - |A| - 2) = |A| + 2$, a contradiction. It follows that $|S \cap V(Q)| \geq 2$. Then $S' = S \setminus \{u, v_1, v_2, v\} \cup \{w_1, w_2\}$ is of size at most k. Moreover, S' is a feedback vertex set in G, since S is a feedback vertex set and by (i). Again, this implies that $S' \setminus \{w_1\}$ is a solution of the instance $(G', k - 1)$, as required.

$\square$

Rule 10 is not used directly in our algorithm, because it seems impossible to detect it in $O(n)$ time. However, to get the claimed kernel size we need just two special cases of Rule 10, which are stated in lemmas 7 and 8 below.

Lemma 7. *Assume there are five vertices v_1, v_2, u, w_1, w_2 such that $N(v_1) = \{v_2, w_1, w_2\}$, $\{w_1, w_2\} \subseteq N(u)$, there is at most one edge incident to v_2 and a vertex outside $\{w_1, w_2, v_1\}$, and there is at most one edge incident to u and a vertex outside $\{w_1, w_2\}$. Then Rule 10 applies.*

Proof. It is easy to see that condition (i) of Rule 10 is satisfied. We proceed to condition (ii). Since Rule 3 does not apply, v_2 is adjacent to w_1 or w_2; by symmetry assume the former. Let $A = \{u, v_1, v_2\}$. We build $E(Q)$ as follows. We start with $E(Q) = \{v_2w_2, v_2v_1, v_1w_1, v_1w_2, uw_1, uw_2\}$. Since Rule 8 does not apply, $v_2w_2 \in E$ or one of $\{v_2w_1, uw_1, uw_2\}$ is a double edge. Hence, we add v_2w_2 or another copy of one of $\{v_2w_1, uw_1, uw_2\}$ to $E(Q)$, respectively. Note that for every $x \in V(Q) \setminus \{w_1\}$ we have $\deg_Q(x) \leq 3 = |E(Q)| - |A| - 1$, as required.

$\square$

Lemma 8. *Assume there are five vertices u_1, u_2, u_3, w_1, w_2 such that $N(u_1) = \{w_1, w_2, u_2\}$, $\{u_1, u_3\} \subseteq N(u_2) \subseteq \{w_1, w_2, u_1, u_3\}$, and there is at most one edge incident to u_3 and a vertex outside $\{w_1, w_2, u_2\}$. Moreover, the edges v_1v_2 and v_2v_3 are simple. Then Rule 10 applies.*

Proof. It is easy to see that condition (i) of Rule 10 is satisfied. We proceed to condition (ii). Let $A = \{u_1, u_2, u_3\}$. We build $E(Q)$ as follows. We start with $E(Q) = \{u_1w_1, u_1w_2, u_1u_2, u_2u_3\}$. There are some cases to consider. Since Rule 3 does not apply, $u_2w_1 \in E$ or $u_2w_2 \in E$. **CASE 1:** $u_2w_1, u_2w_2 \in E$. Then, since Rule 7 does not apply to the $w_1u_1w_2u_2$ cycle, u_1w_1 or u_1w_2 is a double edge. Moreover, since Rule 3 does not apply, $u_3w_1 \in E$ or $u_3w_2 \in E$. We add to $E(Q)$ edges u_2w_1, u_2w_2, either u_3w_1 or u_3w_2 (but not both), and the second copy of either u_1w_1 or u_1w_2 (but not both). Then $|E(Q)| = 8$ and $\max_{x \in V(Q) \setminus \{w_1\}} \deg_Q(x) = 4 = |E(Q)| - |A| - 1$, so (ii) holds.

CASE 2: Exactly one of u_2w_1 and u_2w_2 is an edge; by symmetry assume $u_2w_1 \in E$ and $u_2w_2 \notin E$. Since Rule 7 does not apply, $u_3w_1 \notin E$. And then since Rule 3 does not apply, $u_3w_2 \in E$.

CASE 2.1: u_3w_2 is a double edge. We add to $E(Q)$ edge u_2w_1, and both copies of u_3w_2. Then $|E(Q)| = 7$ and $\max_{x \in V(Q)} \deg_Q(x) = 3 = |E(Q)| - |A| - 1$, so (ii) holds.

CASE 2.2: u_3w_2 is a simple edge. Since Rule 9 does not apply, $\deg_G(u_1) \geq 4$ and $\deg_G(u_2) \geq 4$. We add to $E(Q)$ edges u_2w_1 and u_3w_2, exactly one edge incident to u_1 which is not yet in $E(Q)$ and exactly one edge incident to u_2 which is not yet in $E(Q)$. Then $|E(Q)| = 8$ and $\max_{x \in V(Q) \setminus \{w_1\}} \deg_Q(x) = 4 = |E(Q)| - |A| - 1$, so (ii) holds. $\square$

The following rule was shown to be correct by Abu-Khzam and Khuzam in [1].

Rule 11 Assume there is an induced path with endpoints u and v and with six internal vertices $v_1, \ldots, v_6$ such that for some vertices w_1, w_2 outside the path $N(\{v_1, \ldots, v_6\}) \setminus \{u, v\} = \{w_1, w_2\}$. If $|N(w_1) \cap \{v_1, \ldots, v_6\}| \geq |N(w_2) \cap \{v_1, \ldots, v_6\}|$, then remove w_1 and decrease k by one.

In [1] it was assumed that when Rule 11 described above is applied, G does not contain an induced path $v_1, \ldots, v_5$ such that for some vertex w, we have $N(v_2, v_3, v_4) \setminus \{v_1, v_5\} = \{w\}$. In our algorithm this is guaranteed by Rule 7 (slightly more general than their Rule 6). We are able to extend Rule 11 as follows.

Lemma 9. *Assume there is an induced path with endpoints u and v and with five internal vertices $v_1, \ldots, v_5$ such that for some vertices w_1, w_2 outside the path $N(\{v_1, \ldots, v_5\}) \setminus \{u, v\} = \{w_1, w_2\}$. Then there is an instance (G', k') with $|V(G')| < |V(G)|$ such that (G, k) is a yes-instance iff (G', k') is a yes-instance and $k' \leq k$.*

The proof of Lemma 9 involves five more rules and is quite technical; we defer it to Section 4. We stress here that Lemma 9 is not crucial for getting a substantial improvement of the kernel size. Indeed, if one uses Rule 11 instead of Lemma 9, the resulting kernel is of size at most $15k - 28$ (see Section 3). Let us also remark that by the analysis in Section 3, if someone manages to exclude paths described in Lemma 9 with only *four* internal vertices, the kernel size decreases further to $11k - 20$.

To complete the algorithm we need a final rejecting rule which is applied when the resulting graph is too big. In Section 3 we prove that Rule 12 is correct.

Rule 12 If the graph has more than $13k - 24$ vertices, return a trivial no-instance (conclude that there is no feedback vertex set of size k in G).

3   The size bound

In this section we prove the following theorem.

Theorem 10. *Let G be a planar graph such that rules 1–9 do not apply and G does not contain the configurations described in lemmas 7 and 8. Assume also that for every induced path P with endpoints u and v and with ℓ internal vertices v_1, $\ldots$, v_ℓ the internal vertices have at least three neighbors outside the path, i.e., $|N(\{v_1, \ldots, v_\ell\}) \setminus \{u, v\}| \geq 3$. If there is a feedback vertex set of size k in G, then $|V(G)| \leq (2\ell + 4)k - (4\ell + 6)$.*

Let S be a feedback vertex set of size k in G (i.e., a "solution"), and let F be the forest induced by $V(G) \setminus S$. Denote the set of vertices of F by $V_F = V(G) \setminus S$. We call the vertices in S *solution vertices* and the vertices in V_F *forest vertices*.

A partition of V_F. Now we define some subsets of V_F. Let $I_2, I_{3+} \subseteq V_F$ denote the vertices whose degree in F is two or at least three, respectively. The leaves of F are further partitioned into two subsets. Let L_2 and L_{3+} be the leaves of F that have two or at least three solution neighbors, respectively. By rules 2 and 3 all the vertices in G have degree at least 3. Hence, if a leaf of F has fewer than two solution neighbors, Rule 4 or Rule 5 applies. It follows that every leaf of F belongs to $L_2 \cup L_{3+}$. This proves claim (i) of Lemma 11 below.

Lemma 11. *Graph G satisfies the following properties.*

(i) *The sets I_2, I_{3+}, L_2, L_{3+} form a partition of V_F.*

(ii) *For every pair u, v of solution vertices there are at most two vertices $x, y \in L_2$ such that $N(x) \cap S = N(y) \cap S = \{u, v\}$.*

(iii) *Every vertex of G is of degree at least three.*

(iv) *Every face of G is of length at least two.*

Claim (ii) follows from the fact that Rule 6 does not apply to G. Claim (iii) follows because rules 2 and 3 do not apply to G and Claim (iv) by Rule 1.

The inner forest. Let F_I be the forest on the vertex set $I_{3+} \cup L_{3+}$ such that $uv \in E(F_I)$ iff for some integer $i \geq 0$, there is a path $ux_1 \cdots x_i v$ in forest F such that $u, v \in I_{3+} \cup L_{3+}$ and for every $j = 1, \ldots, i$, vertex x_i belongs to I_2.

Three sets of short chains. A path in F consisting of vertices from $I_2 \cup L_2$ will be called a *chain*. A chain is maximal if it is not contained in a bigger chain. In what follows we introduce three sets of (not necessarily maximal) chains, denoted by CL_2, C_{2-} and C_{3+}. We will do it so that each vertex in I_2 belongs to *at least one* chain from these sets of chains.

For every vertex $x \in L_2$, we consider the maximal chain $(y_1, \ldots, y_p)$ of degree 2 vertices in F such that y_1 is adjacent to x and no y_i has a solution neighbor outside $N_G(x) \cap S$. Then the chain $(x, y_1, \ldots, y_p)$ is an element of CL_2. Note that $L_2 \subseteq V(CL_2)$.

Chains of C_{2-} and C_{3+} are defined using the following algorithm. We consider maximal chains in F, one by one (note that all maximal chains are vertex-disjoint). Let $c = (x_1, x_2, \ldots, x_p)$ be a maximal chain. The vertices of c are ordered so that if $\{x_1, x_p\} \cap L_2 \neq \emptyset$, then $x_p \in L_2$. Using vertices of c we form disjoint bounded length chains and put them in the sets C_{2-} and C_{3+} as follows. Assume that for some $i < p$ the vertices of a prefix $(x_1, x_2, \ldots, x_i)$ have been already partitioned into such chains (in particular $i = 0$ if we begin to process c). There are three cases to consider.

Consider a shortest chain $c_i = (x_{i+1}, \ldots, x_j)$ such that the vertices of c_i have at least three solution neighbors, i.e., $|S \cap N(\{x_{i+1}, \ldots, x_j\})| \geq 3$. If the chain c_i exists, we put it in C_{3+}, and we proceed to the next vertices of c. Otherwise we consider the chain $c_i' = (x_{i+1}, \ldots, x_p)$. Note that vertices of c_i' have at most two solution neighbors.

If $x_p \in I_2$, then we add the chain c_i' to C_{2-} and we finish processing c. Note that then x_p is adjacent to a vertex $u \in L_{3+} \cup I_{3+}$ (otherwise c is not maximal, as we can extend it by a vertex in L_2). Moreover, because of the order of the vertices in c, we know that $x_1 \notin L_2$.

9

It follows that x_1 is also adjacent to a vertex $v \in L_{3+} \cup I_{3+}$. Hence, $uv \in E(F_I)$. We assign chain c_i' to edge uv.

If $x_p \in L_2$, then we do not form a new chain and we finish processing c. Note, however, that the vertices $\{x_{i+1}, \ldots, x_p\} \cap I_2$ belong to a chain in CL_2.

Note also that some vertices of the first chain c_0 can belong to two chains, one in C_{3+} and one in CL_2.

Let us summarize the main properties of the construction.

Lemma 12. *The following properties hold:*

(i) *Every vertex from I_2 belongs to a chain in CL_2, C_{2-} or C_{3+}.*

(ii) *Every chain in $CL_2 \cup C_{2-}$ has at most two solution neighbors.*

(iii) *Every chain in C_{3+} has at least three solution neighbors.*

(iv) *Every chain in C_{2-} is assigned to a different edge of inner forest F_I.*

(v) *Every chain in $C_{2-} \cup CL_2$ has at most $\ell - 1$ vertices.*

(vi) *Every chain in C_{3+} has at most ℓ vertices.*

A solution graph H_S. Let us introduce a new plane multigraph $H_S = (S, E_S)$. Since the vertices of H_S are the solution vertices we call it a *solution graph*. From now on, we fix a plane embedding of G. The vertices of H_S are embedded in the plane exactly in the same points as in G. The edge multiset E_S is defined as follows. For every triple (u, x, v) such that $u, v \in S$, $x \in L_2$ and there is a path uxv in G, we put an edge uv in E_S. Moreover, the edge uv is embedded in the plane exactly as one of the corresponding paths uxv (note that there can be up to four such paths if some edges are double). Note that by Lemma 11(ii), every edge of H_S has multiplicity at most two.

The set of faces of H_S is denoted by F_S. By $F_{S,2}$ we denote its subset with the faces of length two, while $F_{S,3+}$ are the remaining faces. Note that there are no faces of length 1 in H_S.

Lemma 13. *We have $|V(CL_2)| \leq 2(|E_S| - |F_{S,2}|)$.*

Proof. By the definition, for every vertex $x \in L_2$ there is a corresponding edge $uv \in E_S$, where $N_G(x) \cap S = \{u, v\}$. Also, for every chain c in CL_2 there is a corresponding vertex $x \in L_2$, and thus a corresponding edge $uv \in E_S$. We *assign* x, c and the vertices of c to the pair $\{u, v\}$.

Consider an arbitrary pair u, v such that $uv \in E_S$. Note that there are exactly $|E_S| - |F_{S,2}|$ such pairs. We claim that there are at most two elements in $V(CL_2)$ assigned to the pair $\{u, v\}$. Indeed, by Lemma 11(ii), there are at most two vertices in L_2 assigned to $\{u, v\}$. If there are no such vertices, no chain in CL_2 is assigned to $\{u, v\}$, so the claim holds. If there is exactly one vertex $x \in L_2$ assigned, there is exactly one chain $c \in CL_2$ assigned. By Lemma 5, chain c has at most two vertices, so the claim holds. Finally, if there are exactly two vertices $x, y \in L_2$ assigned, there are exactly two chains c_x and c_y assigned. By Lemma 4 we have $|V(c_x)| = |V(c_y)| = 1$. This concludes the proof. $\qquad\square$

Maximality. In what follows we assume that graph G is *maximal*, meaning that one can add neither an edge to $E(G)$ nor a vertex to L_2 obtaining a graph G' such that S is still a feedback

vertex set of G' and all the claims of lemmas 11, 12 and 13 hold. Note that the number of L_2-vertices which can be added to G is bounded, since each such vertex corresponds to an edge in H_S, and H_S has at most $6|S|$ edges as a plane multigraph with edge multiplicity at most two. Similarly, once the set of L_2-vertices is maximal, and hence the vertex set of G is fixed, the number of edges which can be added to G is bounded by $6|V(G)|$. It follows that such a maximal supergraph of G exists. Clearly, it is sufficient to prove Theorem 10 only in the case when G is maximal.

Lemma 14. *The planar graph H_S is connected.*

Proof. Assume now for contradiction that there is a partition $S = S_1 \cup S_2$ such that there is no edge in H_S between a vertex of S_1 and a vertex of S_2.

Every face of G is incident to at least one vertex of S, for otherwise the boundary of the face does not contain a cycle, a contradiction. Assume that a face f of G contains a solution vertex u_1 in S_1 and a solution vertex u_2 in S_2. Then we can add a vertex x, two edges xu_1 and two edges xu_2. Note that S is still a feedback vertex set in the new graph; in particular now $x \in L_2$. In the new graph there are no more vertices in L_2 adjacent to both u_1 and u_2 because of our assumption that S_1 and S_2 are not connected by an edge in H_S, so Lemma 11(ii) holds. Moreover, $|V(CL_2)|$ was increased by one and $|E_S| - |F_{S,2}|$ was also increased by one, so Lemma 13 holds. The other claims of lemmas 11 and 12 trivially hold, so F is not maximal, a contradiction.

Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be the collections of faces of G containing a vertex in S_1, or in S_2, respectively. We have shown above that $\mathcal{F}_1 \cup \mathcal{F}_2$ is a partition of the set of all the faces of G. Let V_1 and V_2 denote the sets of vertices incident to a face in $\mathcal{F}_1$, or in $\mathcal{F}_2$, respectively. Note that $V_1 \cap V_2 \neq \emptyset$, since there must be two neighboring faces, one in $\mathcal{F}_1$ and the other in $\mathcal{F}_2$. Let $x \in V_1 \cap V_2$. Since faces of G are of length at least two, x has in G at least two neighbors in $V_1 \cap V_2$. It follows that $G[V_1 \cap V_2]$ has minimum degree two, so $G[V_1 \cap V_2]$ contains a cycle. However, $(V_1 \cap V_2) \cap S = \emptyset$, since $\mathcal{F}_1$ and $\mathcal{F}_2$ are disjoint. Hence $V_1 \cap V_2 \subseteq F$, a contradiction. $\qquad \square$

Bounding the number of forest vertices in a face of H_S. For a face f of H_S and a set of vertices $A \subseteq V(G)$ we define A^f as the subset of A of vertices which are embedded in f or belong to the boundary of f. Note that all vertices of every chain belong to the same face f of H_S. When C is a set of chains, by C^f we denote the subset of chains of C which lie in f, i.e., $C^f = \{c \in C \; : \; V(c) \subseteq V(G)^f\}$.

Lemma 15. *For every face f of H_S, it holds that $|L_{3+}^f| + |I_{3+}^f| + |C_{3+}^f| \leq d(f) - 2$.*

Proof. First we note that the forest F^f is in fact a tree. Indeed, if F^f has more than one component, we can add an edge between two solution vertices on the boundary of f preserving planarity, what contradicts the assumed maximality.

Consider a plane subgraph A of G induced by $V(G)^f$, i.e., we take the plane embedding of G and we remove the vertices outside $V(G)^f$. Then we can define graph A_S, analogously to H_S. We treat f as a face of A_S. Let $u_1 u_2 \cdots u_{d(f)} u_1$ be the facial walk of f.

Consider an arbitrary vertex x of I_{3+}^f. Let $T_1, \ldots, T_r$ be the r trees obtained from the tree T in F containing x after removing r from T. Then $r \geq 3$ since x has at least three neighbors in T. By planarity, there are $2r$ indices $b_1, e_1, b_2, e_2, \ldots, b_r, e_r$ such that for every $i = 1, \ldots, r$

$$\{u_{b_i}, u_{e_i}\} \subseteq N(V(T_t)) \cap \{u_1, \ldots, u_{d(f)}\} \subseteq \{u_{b_i}, u_{b_i+1}, \ldots, u_{e_i}\}.$$

11

Then, for every $j \in \{b_1, b_2, \ldots, b_r\}$ there is an edge xu_j, for otherwise we can add it in the current plane embedding, contradicting the maximality of G. This means that every vertex in I_{3+}^f has at least three neighbors in $\{u_1, u_2, \ldots, u_{d(f)}\}$.

We further define B as the plane graph obtained from A by (1) replacing every triple (u, x, v) where $x \in L_2$, $u, v \in S$ and uxv forms a path by a single edge, (2) removing vertices of $V(CL_2)$, (3) contracting every chain from C_{3+} into a single vertex, and (4) contracting every chain from C_{2-} into a single edge. By (4) we mean that every maximal chain $d = x_1, \ldots, x_i$ of I_2 vertices which is contained in a chain from C_{2-}, is replaced by the edge yz where y and z are the forest neighbors (in $L_{3+} \cup I_{3+}$) of x_1 and x_i outside the chain d. Let us call the vertices of B that are not on the boundary of f as *inner vertices*.

Note that the set of inner vertices is in a bijection with $L_{3+}^f \cup I_{3+}^f \cup C_{3+}^f$. Moreover, I forms a tree, since F^f is a tree. Also, each inner vertex has at least three neighbors in $\{u_1, u_2, \ldots, u_{d(f)}\}$. We show that $|I| \leq d(f) - 2$ by the induction on $d(f)$. When $d(f) = 2$ the claim follows since each inner vertex has at least three neighbors on the boundary of f. Now assume $d(f) > 2$. Let x be leaf in the tree I. Then the edges from x to the boundary of face f split F into at least three different faces. The subtree $I - x$ lies in one of these faces, say face bounded by the cycle $xu_iu_{i+1}\cdots u_jx$. We remove x and vertices $u_{j+1}, \ldots, u_{i-1}$ (there is at least one of them) and we add edge u_iu_j. The outer face of the resulting graph is of length at most $d(f) - 1$, so we can apply induction and the claim follows.

$\square$

Lemma 16. *For every face f in H_S of length at least three,*

$$|V_F^f \setminus V(CL_2^f)| \leq \ell \cdot (d(f) - 2) - (\ell - 1).$$

Proof. We have

$$|V_F^f \setminus V(CL_2^f)| \leq |L_{3+}^f| + |I_{3+}^f| + |V(C_{3+}^f)| + |V(C_{2-}^f)|.$$

By Lemma 12(v) we get

$$|V_F^f \setminus V(CL_2^f)| \leq |L_{3+}^f| + |I_{3+}^f| + \ell|C_{3+}^f| + (\ell - 1)|C_{2-}^f|. \tag{3}$$

By Lemma 12(iv), $|C_{2-}^f|$ is bounded by the number of edges of the inner forest F_I. Hence, $|C_{2-}^f| \leq |L_{3+}^f| + |I_{3+}^f| - 1$ when $|L_{3+}^f| + |I_{3+}^f| > 0$ and $|C_{2-}^f| = 0$ otherwise. In the prior case, by (3) we get that

$$|V_F^f \setminus V(CL_2^f)| \leq \ell(|L_{3+}^f| + |I_{3+}^f| + |C_{3+}^f|) - (\ell - 1),$$

and the result then follows from Lemma 15. Hence it suffices to prove the claim when $|L_{3+}^f| = |I_{3+}^f| = |C_{2-}^f| = 0$. Then the forest F^f is a non-empty collection of paths, each with both endpoints in L_2. Let c be such a path on p vertices $x_1, \ldots, x_p$. Then $x_1 \in L_2$ and x_1 has exactly two neighbors u, v in S. Let i be the largest such that $N(\{x_1, \ldots, x_i\}) \cap S = \{u, v\}$. By definition, $(x_1, \ldots, x_i)$ is a chain in CL_2^f. We infer that if $i = p$ for every such path, then $|V_F^f \setminus V(CL_2^f)| = 0$ and the claim follows. Hence we can assume that $i < p$, i.e., x_{i+1} has a neighbor in $S \setminus \{u, v\}$. Then, by definition, $(x_1, \ldots, x_{i+1})$ is a chain in C_{3+}^f. Since $(x_1, \ldots, x_i) \in CL_2^f$, we get $|\{x_1, \ldots, x_{i+1}\} \setminus V(CL_2^f)| = 1$. Hence,

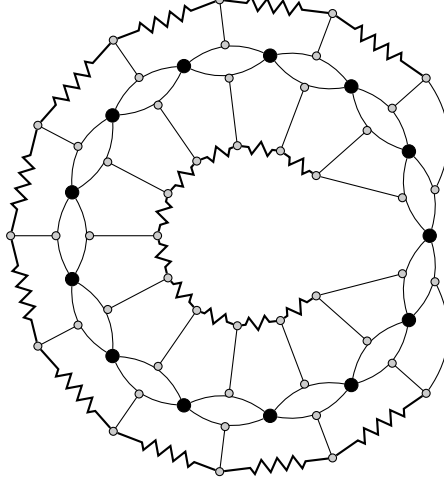$$|V_F^f \setminus V(CL_2^f)| \leq 1 + \ell(|C_{3+}^f| - 1),$$

Figure 4: A tight example. The big black vertices are solution vertices, the small gray ones are forest vertices. The zigzag edges represent paths of $\ell - 1$ forest vertices, each adjacent to the two available solution vertices. Asymptotically for larger cycles, we have $2\ell + 3$ forest vertices for each solution vertex.

what, by Lemma 15 , is bounded by $1 + \ell \cdot (d(f) - 3) = \ell \cdot (d(f) - 2) - (\ell - 1)$, as required.

$\square$

Lemma 17. *For every face f in H_S of length two, $V_F^f \subseteq V(CL_2^f)$.*

Proof. Since the boundary of f has only two solution vertices, F^f contains no vertices of L_{3+}^f, $V(C_{3+})^f$ or I_{3+}^f. Then by Lemma 12(iv), C_{2-}^f is also empty. The claim follows. $\square$

Now we proceed to the bound of Theorem 10. By Lemmas 16 and 17 we have

$$|V_F| \leq |V(CL_2)| + \sum_{f \in F_{S,3+}} (\ell(d(f) - 2) - (\ell - 1))$$

By Lemma 13 we get

$$
\begin{aligned}
|V_F| &\leq 2(|E_S| - |F_{S,2}|) + \sum_{f \in F_{S,3+}} (\ell(d(f) - 2) - (\ell - 1)) \\
&= 2(|E_S| - |F_{S,2}|) + \sum_{f \in F_S} (\ell(d(f) - 2) - (\ell - 1)) + (\ell - 1)|F_{2,S}| \\
&= (2\ell + 2)|E_S| - (3\ell - 1)|F_S| + (\ell - 3)|F_{2,S}| \\
&= (2\ell + 2)|E_S| - (2\ell + 2)|F_S| - (\ell - 3)|F_S| + (\ell - 3)|F_{2,S}| \\
&\leq (2\ell + 2)(|E_S| - |F_S|).
\end{aligned}
$$

By Lemma 14 graph H_S is connected, so we can apply Euler's formula $|S| - |E_S| + |F_S| = 2$. Thus,

$$
\begin{aligned}
|V(G)| = |V_F| + |S| &\leq (2\ell + 2)(|S| - 2) + |S|, \\
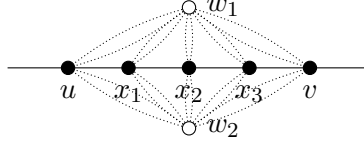&= (2\ell + 3)k - (4\ell + 4).
\end{aligned}
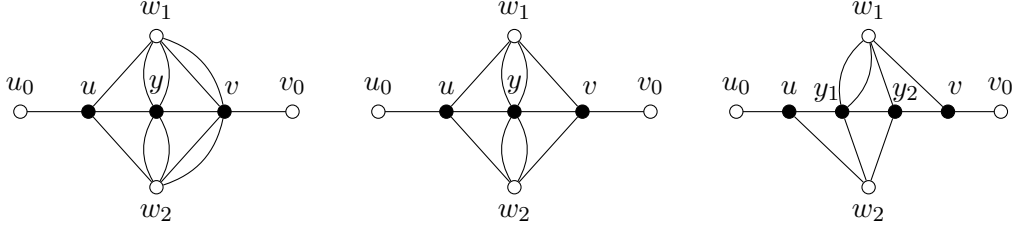$$

Figure 5: Configuration from Lemma 9.

Figure 6: Gadgets of rules 15, 16 and 17.

This concludes the proof of Theorem 10. By Lemma 9, we can put $\ell = 5$, which results in $|V(G)| \leq 13k - 24$. In Figure 4 we show an example of a graph, where our reduction rules do not apply and our analysis is tight (up to a constant additive term).

4   Reducing induced 5-paths with at most two neighbors

This section is devoted to a proof of Lemma 9. Let us recall its statement here.

Lemma 9 (restated)**.** *Assume there is an induced path $u_0ux_1x_2x_3vv_0$ such that for some vertices w_1, w_2 outside the path $N(\{u, x_1, x_2, x_3, v\}) \setminus \{u_0, v_0\} = \{w_1, w_2\}$. Then there is an instance (G', k') with $|V(G')| < |V(G)|$ such that (G, k) is a yes-instance iff (G', k') is a yes-instance and $k' \leq k$.*

Denote the path $ux_1x_2x_3v$ by P. Let $Q = G[V(P) \cup \{w_1, w_2\}]$. By symmetry we assume $|N(w_1) \cap V(P)| \geq |N(w_2) \cap V(P)|$. Then also $|N(w_1) \cap V(P)| \geq 3$, for otherwise Rule 3 applies. In our proof of Lemma 9 we do not apply a single rule, but one of four rules. The kernelization algorithm finds the family $\mathcal{S}$ of all feedback sets of Q. (Note that there is a bounded number of such sets.) Based on the structure of $\mathcal{S}$, one of the four rules is chosen and applied. Let us also define $\delta(Q) = N_G(X) = \{u, v, w_1, w_2\}$. Let us state the rules now.

Rule 13 Assume every set $S_Q \in \mathcal{S}$ satisfies at least one of the conditions below:

(1) $|S_Q| \geq 3$,

(2) $Q - S_Q$ contains a (u, v)-path,

(3) $w_1 \in S_Q$, or

(4) $w_2 \in S_Q$.

Then remove vertex w_1 and decrease k by one.

Rule 14 If $|N(w_2) \cap V(P)| \leq 2$ Then remove vertex w_1 and decrease k by one.

Rule 15 If every set $S_Q \in \mathcal{S}$ satisfies at least one of the conditions (1)-(4) or

14

(5) the sets $\{v\}$ and $\{u, w_1, w_2\}$ are the equivalence classes of $R_{Q-S_Q, \delta(Q)}$,

then replace Q by the gadget from Figure 6 (left), i.e., remove x_1, x_2, x_3 and edges in $G[Q]$, and add a vertex y and edges uy, yv, uw_1, uw_2 and double edges vw_1, vw_2, yw_1, yw_2.

Rule 16 If every set $S_Q \in \mathcal{S}$ satisfies at least one of the conditions (1)-(5) or

(6) the sets $\{u\}$ and $\{v, w_1, w_2\}$ are the equivalence classes of $R_{Q-S_Q, \delta(Q)}$,

then replace Q by the gadget from Figure 6 (middle), i.e., remove x_1, x_2, x_3 and edges in $G[Q]$, and add a vertex y and edges uy, yv, uw_1, uw_2, vw_1, vw_2 and double edges yw_1, yw_2.

Rule 17 If every set $S_Q \in \mathcal{S}$ satisfies at least one of the conditions (1)-(5) or

(7) the sets $\{u, w_2\}$ and $\{v, w_1\}$ are the equivalence classes of $R_{Q-S_Q, \delta(Q)}$,

then replace Q by the gadget from Figure 6 (right), i.e., remove x_1, x_2, x_3 and edges in $G[Q]$, and add vertices y_1, y_2 and edges uy_1, y_1y_2, y_2v, uw_2, y_1w_2, y_2w_2, y_2w_1, vw_1 and a double edge y_1w_1.

Note that the graph modifications in rules 15–17 can be phrased as gadget replacements (X, Y, E_I) in graph G, where $X = \{x_1, x_2, x_3\}$, $Y = \{y\}$ for rules 15 and 16, while $Y = \{y_1, y_2\}$ for Rule 17.

Lemma 18. *If there is an induced path described in Lemma 9, then one of rules 13–17 applies. Moreover, if Rule 17 is applied, then $\mathcal{S}$ contains both a set satisfying* (5) *and a set satisfying* (7).

Proof. Assume rules 13–16 do not apply. We will show that Rule 17 applies. Then there is a set $S_Q \in \mathcal{S}$ that satisfies none of (1)–(6). Since (1), (3) and (4) do not hold for S_Q, we infer that $|S_Q| \leq 2$ and $S_Q \subseteq \{u, x_1, x_2, x_3, v\}$. Recall that $|N(w_1) \cap V(P)| \geq 3$. Let p, q, r be three arbitrary vertices of $N(w_1) \cap V(P)$, sorted by increasing distance from u in the path $ux_1x_2x_3v$. Assume $|S_Q| \leq 1$. Then $q \in S_Q$, for otherwise S_Q is not a feedback vertex set of Q. But then there is a (u, v)-path in $Q - S_Q$: from u follow the path $ux_1x_2x_3v$ to p, then via w_1 to r and again follow the path $ux_1x_2x_3$ to v. Hence S_Q satisfies (2), a contradiction. In what follows we assume $|S_Q| = 2$.
CASE 1: In $Q - S_Q$ there is an edge $ab \in E(P)$. Then in particular $a, b \notin S_Q$. Since Rule 3 does not apply, both a and b have a neighbor in $\{w_1, w_2\}$. However, since $Q - S_Q$ does not contain cycles, the neighbor of a is different than the neighbor of b. Hence, $Q - S_Q$ contains path w_1abw_2 or path w_1baw_2. If $u \in \{a, b\}$, S_Q satisfies (5), a contradiction, and if $v \in \{a, b\}$, S_Q satisfies (6), a contradiction. Hence $\{u, v\} \cap \{a, b\} = \emptyset$. It follows that $\{a, b\} = \{x_1, x_2\}$ or $\{a, b\} = \{x_2, x_3\}$. In the former case $x_3 \in S_Q$ and in the latter case $x_1 \in S_Q$, for otherwise $Q - S_Q$ has a cycle because x_3 (resp. x_1) has a neighbor in $\{w_1, w_2\}$ by Rule 3. Since $|S_Q| = 2$, it follows that exactly one of u, v is not in S_Q. But by Rule 3 both of them have a neighbor in $\{w_1, w_2\}$, so (5) or (6) is satisfied, a contradiction.
CASE 2: $Q - S_Q$ contains no edge of $E(P)$. Then $S_Q = \{x_1, x_3\}$.

If $uw_2 \in E$, then $vw_2 \notin E$, for otherwise (2) holds. It follows that $vw_1 \in E$, for otherwise Rule 3 applies. Then $uw_1 \notin E$, for otherwise (2) holds.

By the same arguments, if $uw_2 \notin E$, then $uw_1 \in E$, $vw_1 \notin E$ and $vw_2 \in E$. We see that the two cases above are symmetric, so let us consider only the former one, i.e., $uw_2, vw_1 \in E$ and $uw_1, vw_2 \notin E$. Then also $w_1w_2 \notin E$, for otherwise (2) holds.

CASE 2.1: $x_2w_1 \notin E$. Since Rule 3 does not apply, $x_2w_2 \in E$. Since $|N(w_1) \cap V(P)| \geq 3$, $x_1, x_3 \in N(w_1)$. Hence $N_Q(w_1) = \{x_1, x_3, v\}$. Since $|N(w_2) \cap V(P)| \leq |N(w_1) \cap V(P)|$ and $|N(w_2) \cap V(P)| \geq 3$ as Rule 14 does not apply, we get $|N(w_2) \cap V(P)| = 3$. It follows that either $N_Q(w_2) = \{u, x_1, x_2\}$ or $N_Q(w_2) = \{u, x_2, x_3\}$. In both cases S_Q satisfies (7).

CASE 2.1.1: $N_Q(w_2) = \{u, x_1, x_2\}$. Consider an arbitrary $S'_Q \in \mathcal{S}$ that satisfies none of (1)–(4) or (7). Similarly as we argued for S_Q, $S'_Q \subseteq \{u, x_1, x_2, x_3, v\}$ and $|S'_Q| \leq 2$. Then $x_1 \in S'_Q$, for otherwise it is impossible to hit all three cycles w_2ux_1, $w_2x_1x_2$ and w_1x_3v. It follows that S'_Q contains x_3 or v, but in the former case (7) holds. Hence $S'_Q = \{x_1, v\}$, and (5) holds for S'_Q. This proves our claim.

CASE 2.1.2: $N_Q(w_2) = \{u, x_2, x_3\}$. Then we rename vertices of Q: swap names of w_1 and w_2, u and v, x_1 and x_3, obtaining $N_Q(w_1) = \{x_1, x_2, v\}$ and $N_Q(w_2) = \{u, x_1, x_3\}$. Notice that in this new setting $\{x_1, x_3\}$ is still a feedback vertex set of Q and satisfies (7). In the new setting, consider an arbitrary $S'_Q \in \mathcal{S}$ that satisfies none of (1)–(4) or (7). Similarly as we argued for S_Q, $S'_Q \subseteq \{u, x_1, x_2, x_3, v\}$ and $|S'_Q| \leq 2$. Then $x_1 \in S'_Q$, for otherwise it is impossible to hit all three cycles w_2ux_1, $w_1x_1x_2$ and $w_1x_1w_1x_3v$. It follows that S'_Q contains x_2, x_3 or v, but in the first case (2) holds and in the second case (7) holds. Hence $S'_Q = \{x_1, v\}$, and (5) holds for S'_Q. This proves our claim.

CASE 2.2: $x_2w_1 \in E$. Then $x_2w_2 \notin E$, for otherwise (2) holds. Since $|N(w_2) \cap V(P)| \geq 3$ by Rule 14, $x_1, x_3 \in N(w_2)$. Hence $N_Q(w_2) = \{u, x_1, x_3\}$. Since $|N(w_1) \cap V(P)| \geq 3$, we have $x_1w_1 \in E$ or $x_3w_1 \in E$. If both edges x_1w_1 and x_3w_1 are present, then Rule 7 applies, a contradiction. It follows that either $N_Q(w_2) = \{x_1, x_2, v\}$ or $N_Q(w_2) = \{x_2, x_3, v\}$. The former case was already considered in Case 2.1.2 (after renaming vertices). In the latter case, we rename vertices by swapping names of w_1 and w_2, u and v, x_1 and x_3. Thus we obtain the already considered Case 2.1.1. □

The lemma below will be very useful in proving that the rules above are correct in particular settings of subgraph Q.

Lemma 19. *In each of the situations below all rules 13–17 are correct.*

  (i) (G, k) is a no-instance.

  (ii) there is a solution S to the instance (G, k) such that $|V(Q) \cap S| \geq 2$ and

      (ii.1) $|V(Q) \cap S| \geq 3$, or

      (ii.2) in $Q - S$ there is an (x_1, x_5) path, or

      (ii.3) $w_1 \in S$, or

      (ii.4) $w_2 \in S$.

Proof. We begin with (i). We need to show that for each of the rules 13–17 the resulting instance (G', k') is a no-instance. Assume the contrary, i.e., let S' be a solution of (G', k'). We will show that there is a solution S of (G, k), contradicting our assumption. For rules 13 and 14 we see that $S = S' \cup \{w_1\}$ works. Now focus on the remaining rules. Let $Y = \{y\}$ for rules 15 and 16, and $Y = \{y_1, y_2\}$ for Rule 17. Let $Q' = G'[N[Y]]$. It is easy to verify that for each of the three rules $|V(Q') \cap S| \geq 2$. If $|V(Q') \cap S| \geq 3$ then we see that $S = S' \setminus V(Q') \cup \{w_1, w_2, x_1\}$ works. Hence we are left with the case $|V(Q') \cap S| = 2$.

Assume there is a (u, v)-path in $G' - S'$. Then we put $S = S' \setminus V(Q') \cup \{w_1, w_2\}$. Note that the equivalence classes of $R_{Q-S, \delta(Q)}$ are $\{u, v\}$, $\{w_1\}$ and $\{w_2\}$. Hence, by Lemma 3 (for $A = G'$ and $B = G$) S is a feedback vertex set of G, so (G, k) is a yes-instance, a contradiction.

Hence we can assume that there is no (u, v) path in $G' - S'$. Note that this implies that $V(Q') \cap S'$ is equal to $\{u, y\}$ for Rule 15, $\{u, y\}$ or $\{y, v\}$ for Rule 16, and $\{y_1, y_2\}$ or $\{y_1, v\}$ for Rule 17.

Consider Rule 15. Since Rule 13 does not apply, there is at least one feedback vertex set $S_Q \in \mathcal{S}$ of size at most two which satisfies (5). Since $V(Q') \cap S' = \{u, y\}$, the equivalence classes of $R_{Q'-S', \delta(Q)}$ are $\{v\}$ and $\{u, w_1, w_2\}$, hence by Lemma 3 (for $A = G'$ and $B = G$) $S = S' \setminus V(Q') \cup S_Q$ is a feedback vertex set G, so (G, k) is a yes-instance.

Now consider Rule 16. Note that it cannot happen that $\mathcal{S}$ contains only sets that satisfy (1)-(4), or (6), because then Rule 15 applies to Q with vertices renamed (swap the names of u and v, x_1 and x_3). Since rules 13 and 15 do not apply to Q, $\mathcal{S}$ contains both a feedback vertex set $S_Q^1 \in \mathcal{S}$ which satisfies (5) and $S_Q^2 \in \mathcal{S}$ which satisfies (6), and $|S_Q^{(5)}|, |S_Q^{(6)}| \leq 2$. Since $V(Q') \cap S' = \{u, y\}$ or $V(Q') \cap S' = \{y, v\}$ the equivalence classes of $R_{Q'-S', \delta(Q)}$ are $\{u\}$ and $\{v, w_1, w_2\}$, or $\{v\}$ and $\{u, w_1, w_2\}$, respectively. In the former case we put $S = S' \setminus V(Q') \cup S_Q^{(6)}$ and in the latter one $S = S' \setminus V(Q') \cup S_Q^{(5)}$. By Lemma 3 (for $A = G'$ and $B = G$) S is a feedback vertex set G, so (G, k) is a yes-instance.

Finally, consider Rule 17. By Lemma 18, $\mathcal{S}$ contains both a set $|S_Q^{(5)}|$ satisfying (5) and a set $|S_Q^{(7)}|$ satisfying (7). Since $V(Q') \cap S' = \{y_1, y_2\}$ or $V(Q') \cap S' = \{y_1, v\}$ the equivalence classes of $R_{Q'-S', \delta(Q)}$ are $\{u, w_2\}$ and $\{v, w_1\}$, or $\{v\}$ and $\{u, w_1, w_2\}$, respectively. In the former case we put $S = S' \setminus V(Q') \cup S_Q^{(7)}$ and in the latter one $S = S' \setminus V(Q') \cup S_Q^{(5)}$. By Lemma 3 (for $A = G'$ and $B = G$) S is a feedback vertex set G, so (G, k) is a yes-instance. This ends the proof of (i).

We proceed to (ii). We need to show that in each of the cases $(ii.1)$–$(ii.4)$, (G', k') is a yes-instance. To this end we will show a feedback vertex set S' of size at most k' in G'.

For $(ii.1)$ we pick $S' = (S \setminus V(Q)) \cup \{u, w_1, w_2\}$. Then the equivalence classes of $R_{Q'-S', \delta(Q)}$ are all singletons, so by Lemma 3 (for $A = G$, $B = G'$) S' is a feedback vertex set of G'.

For $(ii.2)$ we pick $S' = (S \setminus V(Q)) \cup \{w_1, w_2\}$. Then the equivalence classes of $R_{Q'-S', \delta(Q)}$ are $\{u, v\}$, $\{w_1\}$ and $\{w_2\}$, so by Lemma 3 (for $A = G$, $B = G'$) S' is a feedback vertex set of size at most k in G'.

For $(ii.3)$ we consider two cases. In case of rules 13 and 14 it is clear that $S' = S \setminus \{w_1\}$ works. Hence we can assume that Rule 14 does not apply. Let p, q, r be arbitrary three vertices of $N(w_2) \cap V(P)$, in the order of increasing distance from u in the path P. We can assume that $|V(Q) \cap S| \leq 2$, for otherwise we use $(ii.1)$. Then $r \in S$, for otherwise we need to include both p and q (at least) to hit all the cycles of Q. Hence $V(Q) \cap S = \{w_1, r\}$. But then there is a path going from u to p along P, then via w_2 to r and to v along P. Hence we apply $(ii.2)$.

For $(ii.4)$, we can assume that $|V(Q) \cap S| \leq 2$ and $w_1 \notin S$, for otherwise we apply $(ii.1)$ or $(ii.3)$. Since $|N(w_1) \cap V(P)| \geq 3$ we can pick three vertices p, q, r of $N(w_1) \cap V(P)$, in the order of increasing distance from u in the path P. Then $r \in S$, for otherwise we need to include both p and q (at least) to hit all cycles of Q. Hence $V(Q) \cap S = \{w_2, r\}$. But then there is a path going from u to p along P, then via w_1 to r and to v along P. Hence we apply $(ii.2)$. $\qquad\square$

Lemma 20. *Rules 13–17 are correct.*

Proof. By Lemma 19(i) it suffices to prove the correctness when there is a feedback vertex set S of size at most k.

Since $|N(w_1) \cap V(P)| \geq 3$, we see that Q contains cycles and hence $|V(Q) \cap S| > 0$, for otherwise S is not a feedback vertex set.

First we consider the case $|V(Q) \cap S| = 1$. Assume $|N(w_2) \cap V(P)| \geq 3$. By Rule 3, Q has a subgraph R consisting of path P, three edges, each joining a different vertex of P with w_1, and another three edges, each joining a different vertex of P with w_2. Note that R has 7 vertices, 10 edges and maximum degree 4. Then $R - S$ has 6 vertices and at least 6 edges, so S is not a feedback vertex set, a contradiction. Assume $|N(w_2) \cap V(P)| \leq 2$. Then Rule 13 or Rule 14 applies. We will show that $V(Q) \cap S = \{w_1\}$; then clearly $S' = S \setminus \{w_1\}$ is a solution of $(G', k - 1)$. Build a subgraph R of Q as follows. Start with $R = P$. For each $x \in N(w_2) \cap V(P)$, add a single edge xw_2 to R. Next, for each $x \in V(P) \setminus N(w_2)$, add a single edge xw_1 to R (which exists since Rule 3 does not apply). Note that R has 7 vertices, 9 edges and every vertex of R apart from w_1 has degree at most three in R. It follows that $V(Q) \cap S = \{w_1\}$, for otherwise $R - S$ has 6 vertices and at least 6 edges. This finishes the proof of the $|V(Q) \cap S| = 1$ case. From now on we assume $|V(Q) \cap S| \geq 2$.

By Lemma 19 we can assume that $|V(Q) \cap S| = 2$, there is no (u, v)-path in $Q - S$ and $V(Q) \cap S \subseteq V(P)$. In particular $V(Q) \cap S$ satisfies none of (1)–(4). For Rule 13 we are done. For the remaining rules we will show that G' has a feedback vertex S' set of size at most k'.

Consider Rule 14. If $|N(w_2) \cap V(P)| = 0$, then by Rule 3, $V(P) \subseteq N(w_1)$ and Rule 7 applies, a contradiction. If $|N(w_2) \cap V(P)| = 1$, then $S_1 = (S \setminus V(Q)) \cup \{w_1\} \cup (N(w_2) \cap V(P))$ is another feedback vertex set of size at most k in G and we can apply Lemma 19. Hence $|N(w_2) \cap V(P)| = 2$. We claim that in $Q - S$ vertex w_2 is reachable from u or v. We consider cases depending on the distance d in graph P between the two vertices of $N(w_2) \cap V(P)$.

If $d = 1$, then $N(w_2) \cap V(P) = \{x_1, x_2\}$ or $N(w_2) \cap V(P) = \{x_2, x_3\}$, since in other cases Rule 3 or Rule 7 applies. By symmetry we can assume $N(w_2) \cap V(P) = \{x_1, x_2\}$ and $\{u, x_3, v\} \subseteq N(w_1) \cap V(P)$. It follows that there are at most four possible feedback vertex sets of $Q - S$, namely $\{x_1, x_3\}$, $\{x_2, x_3\}$, $\{x_1, v\}$, and $\{x_2, v\}$. The prior two cases are excluded because then $Q - S$ contains a (u, v)-path, and in the latter two cases w_2 is reachable from u in $Q - S$ as required.

If $d = 2$, then $N(w_2) \cap V(P)$ equals $\{u, x_2\}$, $\{x_1, x_3\}$ or $\{x_2, v\}$, where the first and the last case are symmetric, so we skip the analysis of the first one. When $N(w_2) \cap V(P) = \{x_2, v\}$, then $\{u, x_1, x_3\} \subseteq N(w_1) \cap V(P)$ by Rule 3. It follows that there are at most five possible feedback vertex sets of $Q - S$, namely $\{u, x_2\}$, $\{u, x_3\}$, $\{x_1, x_2\}$, $\{x_1, x_3\}$ and $\{x_1, v\}$. In the first, second, third and fourth case w_2 is reachable from v in $Q - S$. In the last case w_2 is reachable from u in $Q - S$. When $N(w_2) \cap V(P) = \{x_1, x_3\}$, then $\{u, x_2, v\} \subseteq N(w_1) \cap V(P)$ by Rule 3. It follows that there are at most six possible feedback vertex sets of $Q - S$, namely $\{u, x_2\}$, $\{u, x_3\}$, $\{x_1, x_2\}$, $\{x_1, x_3\}$, $\{x_1, v\}$ and $\{x_2, v\}$. In the first and second case w_2 is reachable from v in $Q - S$. In the third and fourth case there is a (u, v)-path. In the fifth and sixth case w_2 is reachable from u in $Q - S$.

If $d = 3$, then $N(w_2) \cap V(P)$ equals $\{u, x_3\}$ or $\{x_1, v\}$. By symmetry assume the former. Then $\{x_1, x_2, v\} \subseteq N(w_1) \cap V(P)$ by Rule 3. It follows that there are at most six possible feedback vertex sets of $Q - S$, namely $\{u, x_2\}$, $\{x_1, x_2\}$, $\{x_1, x_3\}$, $\{x_1, v\}$, $\{x_2, x_3\}$ and $\{x_2, v\}$. In the first case w_2 is reachable from v in $Q - S$. In the second case there is a (u, v)-path. In the remaining cases w_2 is reachable from u in $Q - S$.

If $d = 4$ then Rule 3 or Rule 7 applies, a contradiction.

We have thus shown that in $Q - S$, the vertex w_2 is reachable from u or v. By symmetry assume the former. Let z be the vertex of $N(w_2) \cap V(P)$ which is closer to v on P. Note that neither w_2 nor u is reachable from v in $Q - \{w_1, z\}$. Hence, the equivalence classes of

$R_{Q-\{w_1,z\},\{w_1,w_2,u,v\}}$ are $\{v\}$, $\{w_1\}$ and a partition of $\{u,w_2\}$. Then, by Lemma 3 (applied to $A = B = G$), $S_1 = S \setminus V(Q) \cup \{w_1,z\}$ is another feedback vertex set of size at most k in G and we can apply Lemma 19. This finishes the proof of correctness of Rule 14.

Now consider Rule 15. Since $S \cap V(Q)$ satisfies none of (1)–(4), we infer that (5) applies to $S \cap V(Q)$. Then we pick $S' = S \setminus V(Q) \cup \{y,v\}$. Note that the equivalence classes of both $R_{Q'-S',\delta(Q)}$ and $R_{Q-S,\delta(Q)}$ are $\{v\}$, $\{u,w_1,w_2\}$. Then, by Lemma 3 S' is a feedback vertex set in G'.

Now consider Rule 16. Since $S \cap V(Q)$ satisfies none of (1)–(4), we infer that (5) or (6) applies to $S \cap V(Q)$. Then we pick $S' = S \setminus V(Q) \cup \{y,v\}$ or $S' = S \setminus V(Q) \cup \{u,y\}$, respectively. Note that the equivalence classes of both $R_{Q'-S',\delta(Q)}$ and $R_{Q-S,\delta(Q)}$ are either $\{v\}$, $\{u,w_1,w_2\}$ or $\{u\}$, $\{v,w_1,w_2\}$. Then, by Lemma 3 S' is a feedback vertex set in G'.

Finally consider Rule 17. Since $S \cap V(Q)$ satisfies none of (1)–(4), we infer that (5) or (7) applies to $S \cap V(Q)$. Then we pick $S' = S \setminus V(Q) \cup \{y_1,v\}$ or $S' = S \setminus V(Q) \cup \{y_1,y_2\}$, respectively. Note that the equivalence classes of both $R_{Q'-S',\delta(Q)}$ and $R_{Q-S,\delta(Q)}$ are either $\{v\}$, $\{u,w_1,w_2\}$ or $\{u,w_2\}$, $\{v,w_1\}$. Then, by Lemma 3, S' is a feedback vertex set in G'. □

This finishes the proof of Lemma 9.

5   Running time

It is easy to see that each of our reduction rules can be detected and performed in $O(n)$ time in such a way that loops and triple edges are not introduced. Since every rule except for Rule 1 and Rule 5 decreases the number of vertices, the total time needed for detecting and performing them is $O(n^2)$. In what follows we will show that it can be improved to $O(n)$ expected time.

We assume that the graph is stored using adjacency lists. Additionally, we use four data structures:

- A dictionary D_1 implemented as a hash table storing all pairs of adjacent vertices. The hash table stores the corresponding two adjacency list elements for each such pair.

- A dictionary D_2 implemented as a hash table storing all pairs of vertices (x,y) for which the set

$$S_{x,y} = \{z \in V \ : \ x,y \in N(z) \text{ and there is at most one edge } zu \text{ such that } u \notin \{x,y\}\}$$

is nonempty. The hash table stores the set $S_{x,y}$ for each such pair.

- A queue Q_{3+} storing all pairs (x,y) such that $|S_{x,y}| \geq 3$.

- A queue Q_s storing vertices with at most four neighbors (not necessarily all).

Once we have the dictionary D_1 answering adjacency queries in constant expected time, it is easy to detect and apply Rule 1 and Rule 5 immediately after a graph modification. The total expected time needed for that is bounded by total time of graph modifications. Hence, in what follows we exclude from our considerations Rule 1 and Rule 5 and we can assume that application of every rule decreases the number of vertices.

The major challenge in implementing the kernelization algorithm efficiently is *detecting* that a rule applies. In other words we have to find the particular subgraphs described in

the rules, which we call *configurations*, efficiently. If a configuration C appears in G then there is an injective homomorphism $h_C : V(C) \to V(G)$. Vertices of every configuration are partitioned into two categories: black and white vertices, defined in the figures. Let B_C and W_C denote the corresponding sets of vertices of C.

Lemma 21. *Assume Q_{3+} is empty. Then for every vertex v of G with at most four neighbors one can check in $O(1)$ time whether there is a configuration C in G such that $h_C^{-1}(v)$ is a black vertex.*

Proof. Since the number of configurations used in the algorithm is bounded, we can consider a fixed configuration C. We claim that there are only $O(1)$ candidate homomorphisms to check. Note that in each configuration, every white vertex has a black neighbor. It follows that it suffices to show that there are $O(1)$ mappings of all the black vertices of C to $V(G)$ that can extend to a homomorphism h_C, since the configurations are bounded and black vertices have bounded degree. This claim is immediate for configurations where black vertices induce a connected subgraph. The remaining configurations are those from Rule 6, Rule 8, and Lemma 7. Since Q_{3+} is empty, Rule 6 does not apply. In the configurations from the remaining two rules black vertices induce exactly two connected components. Moreover, for both these configurations, once we fix a mapping h of black vertices in the connected component Q of $C[B_C]$ such that $h^{-1}(v) \in V(Q)$, then there are vertices $a \in V(Q)$ and $b \in B_C \setminus V(Q)$ such that for some pair of white vertices x, y we have $x, y \in N_C(a)$ and $x, y \in N_C(b)$. Note also that in both cases there is at most one edge $h_C(b)u$ such that $u \notin \{h_C(x), h_C(y)\}$. But it means that the homomorphic image of b must belong to $S_{h_C(x),h_C(y)}$ and since $Q_{3+} = \emptyset$, there are only $O(1)$ candidates for it. Then there are only $O(1)$ candidates for other black vertices in the connected component of $C[B_C]$ containing b. The claim follows. $\square$

Now we can describe our algorithm.

Algorithm 1: KERNELIZE(G, k)

---

1 Initialize D_1, D_2, Q_{3+} according to their definitions;
2 Initialize Q_s with all vertices of G with at most four neighbors;
3 **while** $Q_{3+} \cup Q_s \neq \emptyset$ **do**
4   **if** $Q_{3+} \neq \emptyset$ **then**
5   Remove an element from Q_{3+} and apply Rule 6;
6   **else**
7   Remove a vertex from Q_s and apply Lemma 21;
8   If a configuration is found, apply the corresponding rule;

The correctness of Algorithm 1 follows from the following invariants.

Invariant 1 The information stored in D_1, D_2, Q_{3+} is up to date.

Invariant 2 If a configuration C appears in G, then there is a vertex $z \in Q_s$ such that for some black vertex $v \in V(C)$ we have $h_C(v) = z$.

Clearly, both invariants hold before before the while loop (Invariant 2 holds since all the black vertices have at most four neighbors). Moreover, after every modification of G resulting in an application of a rule we update the data structures so that both invariants hold, as follows.

- Whenever an adjacency list changes, we update D_1.

- If the set of incident edges of a vertex v with at most four neighbors changes, then we add v to Q_s, and for every pair (x, y) of its neighbors we add the pair to D_2 if needed; if $|S_{x,y}|$ grows to three, we add the pair to Q_{3+}.

- If an edge xy is added to G, we add all elements of $S_{x,y}$ to Q_s. Note that then $Q_{3+} = \emptyset$, since in Rule 6 no edges are added. Hence $|S_{x,y}| \leq 3$.

It is easy to check that the updates described above guarantee that both invariants are satisfied (note that the last item above is needed only to guarantee Invariant 1 for configurations of Rule 7, since this is the only configuration where two white vertices must be adjacent). We are left with the time complexity analysis. Observe that the expected time of the algorithm is bounded by a function which is linear in the total number of insertions to Q_{3+} and Q_s. (A lookup, insert or delete opertion in a hash table works in $O(1)$ expected time and this is the only source of randomness in the running time.) The number of insertions to Q_{3+} and Q_s is linear in the size of the input graph added to the number of applications of the rules. Since each application decreases the number of vertices, there are at most n of them. Hence the total expected time is bounded by $O(n)$. Note that we can turn it to $O(n \log n)$ deterministic time by replacing hash tables by balanced binary search trees.

6   Concluding remarks and further research

We have shown a kernel of $13k$ vertices for PLANAR FEEDBACK VERTEX SET. Our main contribution was applying the region decomposition technique in a new way. It would be interesting to see more applications of the region decomposition technique in problems in which it was not used before.

An obvious open problem is improving the kernel size even further. In particular, it would be nice to break the psychological barrier or *single digit kernel*, i.e., to get a $9k$-kernel. We suppose that if this is possible, it would require finding a number of new, very specialized reduction rules.

References

[1] F. N. Abu-Khzam and M. B. Khuzam. An improved kernel for the undirected planar feedback vertex set problem. In *Proc. IPEC'12*, volume 7535 of *Lecture Notes in Computer Science*, pages 264–273. Springer, 2012.

[2] J. Alber, M. R. Fellows, and R. Niedermeier. Polynomial-time data reduction for dominating set. *J. ACM*, 51(3):363–384, 2004.

[3] H. L. Bodlaender. A cubic kernel for feedback vertex set. In *Proc. STACS'08*, pages 320–331. Springer-Verlag, 2007.

[4] H. L. Bodlaender and E. Penninkx. A linear kernel for planar feedback vertex set. In *Proc. IWPEC'08*, volume 5018 of *LNCS*, pages 160–171. Springer, 2008.

[5] K. Burrage, V. Estivill-Castro, M. Fellows, M. Langston, S. Mac, and F. Rosamond. The undirected feedback vertex set problem has a poly(k) kernel. In *Proc. IWPEC'06*, pages 192–202. Springer-Verlag, 2006.

[6] J. Chen, H. Fernau, I. A. Kanj, and G. Xia. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. *SIAM J. Comput.*, 37(4):1077–1106, 2007.

[7] H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23, 2014.

[8] P. Festa, P. Pardalos, and M. Resende. Feedback set problems. In *Encyclopedia of optimization*, pages 1005–1016. Springer, 2009.

[9] F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Bidimensionality and kernels. In *SODA*, pages 503–510, 2010.

[10] J. Guo and R. Niedermeier. Linear problem kernels for np-hard problems on planar graphs. In *Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9-13, 2007, Proceedings*, volume 4596 of *Lecture Notes in Computer Science*, pages 375–386, 2007.

[11] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[12] L. Kowalik. Nonblocker in H-minor free graphs: Kernelization meets discharging. In *Parameterized and Exact Computation - 7th International Symposium, IPEC 2012, Ljubljana, Slovenia, September 12-14, 2012. Proceedings*, volume 7535 of *Lecture Notes in Computer Science*, pages 61–72, 2012.

[13] L. Kowalik, M. Pilipczuk, and K. Suchan. Towards optimal kernel for connected vertex cover in planar graphs. *Discrete Applied Mathematics*, 161(7-8):1154–1161, 2013.

[14] W. Luo, J. Wang, Q. Feng, J. Guo, and J. Chen. Improved linear problem kernel for planar connected dominating set. *Theor. Comput. Sci.*, 511:2–12, 2013.

[15] A. Silberschatz, P. B. Galvin, and G. Gagne. *Operating System Concepts*. Wiley Publishing, 8th edition, 2008.

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

[17] J. Wang, Y. Yang, J. Guo, and J. Chen. Planar graph vertex partition for linear problem kernels. *J. Comput. Syst. Sci.*, 79(5):609–621, 2013.

[18] M. Xiao. A new linear kernel for undirected planar feedback vertex set: Smaller and simpler. In *Proc. AAIM 2014*, volume 8546 of *Lecture Notes in Computer Science*, pages 288–298, 2014.