Structural Parameterizations of Feedback Vertex Set

Diptapriyo Majumdar

The Institute of Mathematical Sciences, Chennai, HBNI, India
diptapriyom@imsc.res.in

Abstract

A feedback vertex set in an undirected graph is a subset of vertices whose removal results in an acyclic graph. It is well-known that the problem of finding a minimum sized (or k-sized in case of decision version of) feedback vertex set (FVS) is polynomial time solvable in (sub)-cubic graphs, in pseudo-forests (graphs where each component has at most one cycle) and mock-forests (graphs where each vertex is part of at most one cycle). In general graphs, it is known that the problem is NP-complete, and has an $\mathcal{O}^*((3.619)^k)$ fixed-parameter algorithm and an $\mathcal{O}(k^2)$ kernel where k, the solution size, is the parameter. We consider the parameterized and kernelization complexity of feedback vertex set where the parameter is the size of some structure in the input. In particular, we show that

- FVS is fixed-parameter tractable, but is unlikely to have polynomial sized kernel when parameterized by the number of vertices of the graph whose degree is at least 4. This answers a question asked in an earlier paper.
- When parameterized by k, the number of vertices, whose deletion results in a pseudo-forest, we give an $\mathcal{O}(k^6)$ vertices kernel improving from the previously known $\mathcal{O}(k^{10})$ bound.
- When parameterized by the number k of vertices, whose deletion results in a mock-d-forest, we give a kernel consisting of $\mathcal{O}(k^{3d+3})$ vertices and prove a lower bound of $\Omega(k^{d+2})$ vertices (under complexity theoretic assumptions). Mock-d-forest for a constant d is a mock-forest where each component has at most d cycles.

1998 ACM Subject Classification G.2.2 Graph Algorithms, I.1.2 Analysis of Algorithms

Keywords and phrases Parameterized Complexity, Kernelization, Feedback Vertex Set, Structural Parameterization

Digital Object Identifier 10.4230/LIPIcs.IPEC.2016.21

1 Introduction

In the early years of parameterized complexity and algorithms, problems were almost always parameterized by solution size. Recent research has focused on other parameterizations based on structural properties of the input [16, 9, 15, 8], above or below guaranteed optimum values [14]. Such 'non-standard' parameters are known to be small in practice. Also once a problem is shown to be fixed-parameter tractable (and/or having a polynomial kernel) with respect to a parameter, it is a natural question whether it has a fixed-parameter algorithm or polynomial kernel with respect to a smaller parameter. Similarly, when a problem is W-hard or has no polynomial kernel then it is interesting to ask whether it is fixed-parameter tractable or admits a polynomial kernel when it is parameterized by a *structurally larger* parameter. We study such ecology of parameterization for Feedback Vertex Set.

Feedback Vertex Set in an undirected graph G asks whether G has a subset S of at most k vertices such that $G \setminus S$ is a forest, for a given integer k. The set S is called a feedback vertex set of the graph. The problem is known to be NP-complete even on

11th International Symposium on Parameterized and Exact Computation (IPEC 2016).
Editors: Jiong Guo and Danny Hermelin; Article No. 21; pp. 21:1–21:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

bipartite graphs [13] and in graphs of degree at most 4 [20], but is polynomial time solvable in sub-cubic graphs [22, 6, 5], asteroidal triple free graphs [19] and chordal bipartite graphs [17]. The problem is easy (polynomial time) to solve in pseudo-forests (graphs in which each component has at most one cycle), in mock-forests (graphs where each vertex is part of at most one cycle), in cliques and disjoint union of cliques. This is also one of the well-studied problems in parameterized complexity and when parameterized by solution size, it has an algorithm with running time $\mathcal{O}^*(3.619^k)$ [18][1] and a kernel with $\mathcal{O}(k^2)$ vertices and edges [21].

Some parameterizations by the size of some structure in the input have already been explored. FEEDBACK VERTEX SET parameterized by the size of maximum induced matching (also maximum independent set and vertex clique cover) has been shown to be $W[1]$-Hard but contained in XP (See [16, 1]). Bodlaender et al. [2] proved that FEEDBACK VERTEX SET parameterized by deletion distance to a cluster graph (disjoint union of cliques) has no polynomial kernel unless NP $\subseteq$ coNP/poly. Jansen et al. [16] give a survey of results known for such structural parameterization of feedback vertex set and show that

- FEEDBACK VERTEX SET parameteirzed by DELETION DISTANCE TO CHORDAL GRAPH is fixed-parameter tractable;
- FEEDBACK VERTEX SET parameterized by DELETION DISTANCE TO PSEUDO-FOREST has an $\mathcal{O}(f^{10})$ kernel and a kernel lower bound of $\Omega(f^4)$ where f is the size of the deletion distance to pseudo-forest of the input graph.
- FEEDBACK VERTEX SET parameterized by DELETION DISTANCE TO MOCK-FOREST has no polynomial kernel unless NP $\subseteq$ coNP/poly.

Our Results: Jansen et al. suggested in [16], *"An interesting question in this direction is whether* FEEDBACK VERTEX SET *is XP or FPT when parameterized by the vertex deletion distance to sub-cubic graphs or alternatively, parameterized by the number of vertices of degree more than* 3". While the first question remains open, our first result is an answer to the latter question (FVS-HIGH-DEGREE defined below). We answer it positively by providing a fixed-parameter algorithm running in time $\mathcal{O}^*(2^k)$. We also prove that this problem has no polynomial kernel unless NP $\subseteq$ coNP/poly.

FVS-HIGH-DEGREE **Parameter:** k
Input: An undirected graph G such that $|\{u \in V(G)|deg_G(u) > 3\}| \leq k$ and $\ell \in \mathbb{N}$.
Question: Does G have a feedback vertex set of size at most ℓ?

Our next result is an improved kernel for the following problem for which an $\mathcal{O}(k^{10})$ vertex kernel and a conditional lower bound of $\Omega(k^4)$ were given by Jansen et al. [16].

FVS-PSEUDO-FOREST **Parameter:** k
Input: An undirected graph G, $S \subseteq V(G)$ of size at most k such that $G[V(G) \setminus S]$ is a graph in which every component has at most one cycle and an integer ℓ.
Question: Does G have a feedback vertex set of size at most ℓ?

We give a kernel on $\mathcal{O}(k^6)$ vertices, narrowing the gap between upper and lower bound for the size of the kernel. Note that every feedback vertex set is also a pseudo-forest deletion set, but not all pseudo-forest deletion sets are feedback vertex sets. So, in this problem, our parameter is smaller than the solution size.

[1] $\mathcal{O}^*$ notation suppresses polynomial factors.

Finally, we consider a variation of mock-forests (called mock-d-forest) where each component has at most d cycles, where d is a constant, and consider the kernelization complexity of FVS parameterized by the deletion distance to mock-d-forests. It is easy to see that FVS is fixed-parameter tractable when parameterized by the deletion distance to mock-d-forest (or any mock-forest) as any mock-forest has tree-width at most 2. Also, it is easy to see that any pseudo-forest deletion set is also a mock-d-forest deletion set. But not all mock-d-forest deletion sets are pseudo-forest deletion sets. So, our parameter for this problem is not just smaller than solution size, it is even smaller than the parameter for FVS-Pseudo-Forest problem. But, it is larger than the size of the mock-forest deletion set for which case there is no polynomial kernel.

> FVS-Mock-d-Forest where $d \geq 2$ and d is a constant **Parameter:** k
> **Input:** An undirected graph G, $S \subseteq V(G)$ of size at most k such that $G[V(G) \setminus S]$ is a mock-forest where every component has at most d cycles and an integer ℓ.
> **Question:** Does G have a feedback vertex set of size at most ℓ?

When d is not bounded, then we know that this problem has no polynomial kernel unless NP $\subseteq$ coNP/poly [16]. Here, we provide a $\mathcal{O}(k^{3d+3})$ vertex kernel for this problem when d is a constant. And we also prove that a kernel consisting of $O(k^{d+2-\epsilon})$ is unlikely for any $\epsilon > 0$ unless NP $\subseteq$ coNP/poly. We assume that for FVS-Pseudo-Forest, the deletion set to pseudo-forest is given with the input. But, this is not a serious assumption as there are constant factor approximation algorithms [10, 12] for computing a minimum vertex deletion set to a pseudo-forest of a graph.

We organise our paper as follows. In Section 2, we introduce the notations. In Section 3, we provide the FPT Algorithm for FVS-High-Degree and prove that it has no polynomial kernel unless NP $\subseteq$ coNP/poly. In Section 4, we provide the improved polynomial kernel for FVS-Pseudo-Forest. In Section 5, we provide the polynomial kernel and a conditional kernel lower bound of $\Omega(k^{d+2})$ for FVS-Mock-d-Forest.

2 Preliminaries and Notations

By $[r]$, we mean the set $\{1, 2, \ldots r\}$. Throughout the paper we denote the *feedback vertex set number* (the size of a minimum feedback vertex set) by $fvs(G)$. Let S be a set of vertices. By $\binom{S}{r}$, we denote the family of subsets of S containing *exactly* r vertices. By $\binom{S}{\leq r}$, we denote the family of subsets of S containing *at most* r vertices. We call a pair of vertices (u, v) a *double edge* if there are at least 2 edges between u and v. Otherwise we call (u, v) a *non-double-pair*. For an edge (u, v) the *multiplicity* of (u, v) is the number of edges present between u and v. Let $G = (V, E)$ be a tree or a pseudo-forest or a mock-forest. Then a set of vertices $V' \subseteq V(G)$ is a *degree-2-path* if V' induces an acyclic path and every vertex of the path has degree exactly 2 in G. A degree-2-path is maximal if no proper superset of V' is a degree-2-path. Let G be a graph where we contract an edge (u, v). Then we denote $G' = G/(u, v)$ as the graph created by contraction of edge (u, v). Let uv be the contracted vertex as a result of contraction. Then, $N_{G'}(uv) = N_G(u) \cup N_G(v)$. We denote $G[B]$ by the graph induced on the vertex set $B \subseteq V(G)$. We say $G[B]$ is a *double-clique* if there are at least 2 edges between every pair of vertices in B.

We give the definitions of fixed-parameter tractability, kernelization, polynomial parameter transformation and its related facts.

2.1 Definitions and Properties

▶ **Definition 1** (Fixed-Parameter Tractability). Let $L \subseteq \sum^* \times \mathbb{N}$ is a parameterized language. L is said to be fixed-parameter tractable (or FPT) if there exists an algorithm $\mathcal{B}$, a constant c and a computable function f such that $\forall x, \forall k$, $\mathcal{B}$ on input (x, k) runs in at most $f(k).|x|^c$ time and outputs $(x, k) \in L$ iff $\mathcal{B}([x, k]) = 1$. We call the algorithm $\mathcal{B}$ as fixed-parameter algorithm.

▶ **Definition 2** (Slice-Wise Polynomial (XP)). Let $L \subseteq \sum^* \times \mathbb{N}$ is a parameterized language. L is said to be Slice-Wise Polynomial (or in XP) if there exists an algorithm $\mathcal{B}$, a constant c and computable functions f, g such that $\forall x, \forall k$, $\mathcal{B}$ on input (x, k) runs in at most $f(k).|x|^{g(k)+c}$ time and outputs $(x, k) \in L$ iff $\mathcal{B}([x, k]) = 1$. We call the algorithm $\mathcal{B}$ as XP Algorithm.

▶ **Definition 3** (Kernelization). Let $L \subseteq \sum^* \times \mathbb{N}$ be a parameterized language. Kernelization is a procedure that replaces the input instance (I, k) by a reduced instance (I', k') such that

- $k' \leq f(k)$, $|I'| \leq g(k)$ for some function f, g depending only on k.
- $(I, k) \in L$ if and only if $(I', k') \in L$.

The reduction from (I, k) to (I', k') must be computable in $poly(|I| + k)$ time. If $g(k) = k^{\mathcal{O}(1)}$ then we say that L admits a *polynomial kernel*.

▶ **Definition 4** (Soundness/Safeness of Reduction Rule). A reduction rule that replaces an instance (I, k) of a parameterized language L by a reduced instance (I', k') is said to be sound or safe if $(I, k) \in L$ if and only if $(I', k') \in L$.

▶ **Definition 5** (Polynomial parameter transformation (PPT)). Let P_1 and P_2 be two parameterized languages. We say that P_1 is polynomial parameter reducible to P_2 if there exists a polynomial time computable function (or algorithm) $f : \sum^* \times \mathbb{N} \to \sum^* \times \mathbb{N}$, a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that $(x, k) \in P_1$ if and only if $f(x, k) \in P_2$ and $k' \leq p(k)$ where $f((x, k)) = (x', k')$. We call f to be a polynomial parameter transformation from P_1 to P_2.

The following proposition gives the use of the polynomial parameter transformation for obtaining kernels for one problem from another.

▶ **Proposition 6** ([3]). *Let $P, Q \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems and assume that there exists a PPT from P to Q. Furthermore, assume that classical version of P is NP-hard and Q is in NP. Then if Q has a polynomial kernel then P has a polynomial kernel.*

2.2 Initial Preprocessing Rules

For the algorithms in Section 3, 4, 5, we use the following well known reduction rules. See Chapter 3, 4 of [6] for safeness of these Reduction Rules. Here ℓ is the size of the solution (fvs) being sought.

▶ **Reduction Rule 7.** *If there exists $u \in V(G)$ such that u has a self loop, then $G' \leftarrow G \setminus \{u\}, \ell' \leftarrow \ell - 1$.*

▶ **Reduction Rule 8.** *If there exists a vertex v such that $deg_G(v) \leq 1$, then $G' \leftarrow G \setminus v, \ell' \leftarrow \ell$.*

▶ **Reduction Rule 9.** *If there exists a vertex v such that $N_G(v) = \{u, w\}$, then delete the vertex v and add an edge (u, w) into G.*

Note that Reduction Rule 9 can create parallel edges.

▶ **Reduction Rule 10.** *If there exists an edge (u, v) whose multiplicity is more than 2, then reduce its multiplicity to 2.*

3 Feedback Vertex Set Parameterized by number of vertices of degree more than 3

FVS-High-Degree **Parameter:** $|\{u \in V(G)|deg_G(u) > 3\}| \leq k$
Input: An undirected graph $G = (V, E)$ and an integer ℓ.
Question: Does G have a feedback vertex set of size at most ℓ?

Note that our input can be a multigraph. We prove that this problem is fixed-parameter tractable and has no polynomial kernel unless $NP \subseteq coNP/poly$. First, we provide an FPT Algorithm to answer a question asked in [16]. Then, we prove that this problem has no polynomial kernel. Let $S = \{u \in V(G)|deg_G(u) > 3\}$. Throughout this section and in Sections 4, and 5, we use F to denote $G \setminus S$.

3.1 Fixed-Parameter Algorithm

Now, we provide the FPT Algorithm for this problem. We first make the graph minimum degree three. Then, we run over all possible subsets of S and for every subset S' of S, we reduce that instance to a polynomial time solvable problem.

▶ **Theorem 11.** *There exists an algorithm that runs in $\mathcal{O}(2^k \cdot n^{\mathcal{O}(1)})$ time for* FVS-High-Degree *problem.*

Proof. First we apply Reduction Rules 7, 8, 9, 10 in sequence and keep updating ℓ appropriately. Then the algorithm works as follows (see Algorithm 1 for pseudo-code).

We guess a subset $S' \subseteq S$ that intersects S with an ℓ sized feedback vertex set we are seeking for. If $G[S \setminus S']$ has a cycle, then we move on to the next guess. Otherwise, let $S'' = S \setminus S'$ and $G[S'']$ is a forest. Now, let $F = G \setminus S$. $\ell' = \ell - |S'|$. Now, we have to find a *minimum feedback vertex set* D of $G \setminus S'$ such that $S'' \cap D = \emptyset$. Note that every vertex in F has degree at most three in G and also in $G \setminus S'$. Now, we subdivide every edge $(u, v) \in E(F)$, by adding a new vertex $e_{u,v}$ and we add $e_{u,v}$ to S''. We get $T' = S'' \cup \{e_{u,v}|(u, v) \in E(F)\}$. u and v are the only two neighbours of $e_{u,v}$. Hence we have that for every vertex $u \in V(F)$, u has no neighbour in F. Let the graph we have currently is G''. Let $R' = V(G'') \setminus T'$. Note that R' is an independent set. Our goal is to find a feedback vertex set of G'' of at most ℓ' vertices that is disjoint from T'. Now, we pre-process G'' using the following reduction rules (also available in [18]) so that every vertex in R' has exactly three neighbours in T' and all such neighbours appear in different components of $G''[T']$. We need to apply these rules in sequence. Safeness of first two of them are easy to see.

▶ **Reduction Rule 12.** *If there exists $u \in R'$ such that $deg_{G''}(u) \leq 1$, then delete u.*

▶ **Reduction Rule 13.** *If there exists $u \in R'$ such that $G''[T' \cup \{u\}]$ has a cycle, then delete u from G'' and reduce ℓ' by 1.*

Notice that Reduction Rule 13 is also applicable when a vertex $u \in R'$ has exactly two neighbours in T' that are in same component of $G[T']$.

▶ **Reduction Rule 14.** *If there exists a vertex $u \in R'$ such that u has exactly 2 neighbours in T' and those two neighbours in different components of $G''[T']$, then move u to T'.*

▶ **Lemma 15** ($\star$[2]). *Reduction Rules 12, 13 and 14 are safe and can be implemented in polynomial time.*

When Reduction Rules 12, 13, 14 are not applicable, then our goal is to solve the following problem.

SPECIAL DISJOINT FEEDBACK VERTEX SET **Input:** An undirected graph $G = (V, E)$, $S_1 \cup S_2 = V(G)$, $S_1 \cap S_2 = \emptyset$, $G[S_1]$ is a forest, S_2 is an independent set and every vertex of S_2 has exactly 3 neighbours and all are in different components of $G[S_1]$. **Goal:** Find a minimum feedback vertex set W of G such that $W \cap S_1 = \emptyset$.

The following Lemma is due to Kociumaka and Pilipczuk [18] which uses matroid techniques.

▶ **Lemma 16** ([18]). *Let (G, S_1, S_2) be an instance of* SPECIAL DISJOINT FEEDBACK VERTEX SET. *Then there exists a polynomial time algorithm that finds a minimum feedback vertex set W of G such that $W \subseteq S_2$.*

Now, if $|W| \leq \ell'$, then we output YES. Otherwise we repeat the above steps for another subset of S. There are at most $2^{|S|}$ many such subset S' of S and after guessing subset, the problem is polynomial time solvable. If for every subset of S, it is seen that $|W| > \ell'$, then we output NO. Therefore, we have an algorithm that runs in time $\mathcal{O}^*(2^k)$. ◀

3.2 Kernelization Lower Bound

Now, to justify that FVS-HIGH-DEGREE has no polynomial kernel unless NP $\subseteq$ coNP/poly, we use the following theorem and the construction that is used by Jansen et al. [16].

▶ **Theorem 17** ([11]). *Let ϕ be a boolean formula in CNF form with n variables and m clauses. CNF-SAT parameterized by n has no polynomial kernel unless* NP $\subseteq$ coNP/poly.

Jansen et al. [16] provided a polynomial parameter transformation from CNF-SAT parameterized by number of variables, n to FEEDBACK VERTEX SET parameterized by deletion distance to MOCK-FOREST. In that construction, the size of the deletion distance to MOCK-FOREST is at most $4n$. In the same construction, the number of vertices of the graph whose degree is at least 4 is $2n$. For details, see Section 6 in [16]. So, the same transformation is also a polynomial parameter transformation from CNF-SAT parameterized by number of variables to FVS-HIGH-DEGREE. Thus, we have the following corollary.

▶ **Corollary 18.** FVS-HIGH-DEGREE *has no polynomial kernel unless* NP $\subseteq$ coNP/poly.

4 Improved Polynomial Kernel for Parameterization by Deletion Distance to Pseudo-Forest

FVS-PSEUDO-FOREST **Parameter:** k **Input:** An undirected graph G, $S \subseteq V(G)$ of size at most k such that $G[V(G) \setminus S]$ is a graph in which every component has at most one cycle, and an integer ℓ. **Question:** Does G have a feedback vertex set of size at most ℓ?

[2] Due to lack of space, the proofs of Lemmas and Observations marked $\star$ will appear in the full version.

Algorithm 1: FVS-PARAM-HIGH-DEGREE-VERTICES

input : $G = (V, E)$ and $\ell \in \mathbb{N}$
output : YES if $\exists C \subseteq V(G), |C| \leq \ell$ such that $G \setminus C$ is a forest, NO otherwise

$S \leftarrow \{u \in V(G) | deg_G(u) \geq 4\}$;
$\ell' \leftarrow \ell$;
for *every* $S' \subseteq S$ **do**
  if $G[S \setminus S']$ *is a forest* **then**
    $S'' \leftarrow S \setminus S'$;
    $\ell' \leftarrow \ell - |S'|$;
    $F = G \setminus S$;
    $T = \emptyset$;
    for *each* $(u, v) \in E(F)$ **do**
      $T \leftarrow T \cup \{e_{u,v}\}$;
    $T' \leftarrow T \cup S''$;
    $E' = E(G[S'']) \cup \{(u, e_{u,v}) | (u, v) \in E(F)\}$;
    $G'' = (T', E')$;
    Apply Reduction Rules 12, 13, 14 in this sequence and keep updating ℓ' appropriately.;
    When Reduction Rules 12, 13, 14 are not applicable, run algorithm for Lemma 16 and get W;
    if $|W| \leq \ell'$ **then**
      Return YES
Return NO;

Throughout the section for input (G, S, ℓ), we use F to denote $G[V(G) \setminus S]$. An $\mathcal{O}(k^{10})$ vertex kernel is provided by Jansen et al. [16]. We provide here an improved kernel. We first apply the Reduction Rules 7, 8, 9, 10. It is easy to see that these reduction rules can be applied in polynomial time. When Reduction Rules 7, 8, 9, 10 are not applicable, then every vertex of the graph has degree at least three and there are at most two edges between every pair of vertices. In particular, every vertex in $V(F)$ has at least one neighbour in S. We partition the vertices of F into H_1, H_2, H_3. We also partition the components of F into F_1, F_2, F_3, F_4. Formal notations are given as follows.

- $H_1 = \{u \in V(F) | deg_F(u) \leq 1\}$.
- $H_2 = \{u \in V(F) | deg_F(u) = 2\}$.
- $H_3 = \{u \in V(F) | deg_F(u) \geq 3\}$.
- F_1 – set of connected component of F that is a tree.
- F_2 – set of connected component of F that contains a vertex from H_1 and also contains a cycle. Let c_2 be the number of such components.
- F_3 – set of connected component of F that are induced cycles consisting of two vertices. Let c_3 be the number of such components.
- F_4 – set of connected component of F that are induced cycles of length at least three. Let c_4 be the number of such components.

Let $\mathcal{P}$ be the collection of *maximal degree-2-paths* in $F_1 \cup F_2$. Let $\hat{M}$ be a maximum matching in $G[\mathcal{P} \cup F_4]$. Also let $\hat{c} = c_2 + c_4$. We will use these notations in the rest of the section.

4.1 General Reduction Rules

Our first step is to device some reduction rules to bound the number of vertices in H_1. By pseudo-forest property, the number of vertices in H_3 becomes bounded. Now, to bound the number of vertices in H_2, we need to bound the number of edges in M, the number of maximal degree-2-paths in $\mathcal{P}$ and c_4. By pseudo-forest property, the number of maximal degree-2-paths in $\mathcal{P}$ also becomes bounded once $|H_1|$ and $|H_3|$ are bounded. In order to define such reduction rules, we need to use the fact crucially that the minimum degree of G is at least 3. In particular, for every vertex $v \in H_1$, either there exists $x \in S$ such that (x, v) is double-edge or there exists $x, y \in N_G(v) \cap S$. For every vertex $v \in H_1$, there exists $x \in S$ such that $(x, v) \in E(G)$. The reduction rules described in this subsection help to bound H_1 and also $M \cap (E(F_1) \cup E(F_2))$. These reduction Rules also appear in [4] in different form.

▶ **Reduction Rule 19.** *Let $x \in S$. Then $G' \leftarrow G \setminus \{x\}, \ell' \leftarrow \ell - 1$ if any of the following happens.*

- *There are at least $|S| + 1$ vertices in H_1 that are connected to x by a double-edge.*
- *There are at least $|S| + \hat{c} + 1$ vertices in $F_1 \cup F_2 \cup F_4$ that are matched by $\hat{M}$ and are connected to x by a double-edge.*
- *$N_G(x)$ contains both end points of at least $|S| + \hat{c} + 1$ edges in $\hat{M}$.*

▶ **Reduction Rule 20.** *Let $x, y \in S$ such that (x, y) is not a double-edge. Then, make (x, y) into a double-edge if one of the following happens.*

- *$|N_G(x) \cap N_G(y) \cap H_1| \geq |S| + 2$.*
- *$N_G(x) \cup N_G(y)$ contains both end points of at least $|S| + \hat{c} + 2$ edges of $\hat{M}$.*

Even though Reduction Rule 20 does not reduce the size of the graph, it helps to capture some constraints and also helps to apply some other reduction rules (for example Reduction Rule 21).

▶ **Reduction Rule 21.**

- *If there exists a vertex $u \in F$ such that $deg_F(u) = 0$, and there is no double edge attached to u and if $N_G(u) \cap S$ forms a double clique, then $G' \leftarrow G \setminus \{u\}$.*
- *If there exists $u \in F$ such that $deg_F(u) = 1$, and there is no double edge attached to u and $N_G(u) \cap S$ forms a double clique, then $G' \leftarrow G/(u, v)$ where $\{v\} = N_G(u) \cap F$. However, the multiple edges created because of contraction should not be deleted.*
- *If there exists $(u, v) \in M$ such that $N_G(u) \cap N_G(v) \cap S = \emptyset$, and no double edge is attached to either of u or v and $G[(N_G(u) \cup N_G(v)) \cap S]$ forms a double clique, then $G' \leftarrow G/(u, v)$. However, multiple edges or self loops created because of contraction should not be deleted.*

See Figure 1 for illustration.

4.2 Bounding $|H_1 \cup H_3|$

Now, we proceed to bound the number of vertices in F that have degree at most one and at least three. We know that F is a pseudo-forest. We need to use some structural properties of a pseudo-forest and also in applicability of the Reduction Rules 7, 8, 9, 10 19, 20 21. Here is an observation about pseudo-forest.

▶ **Observation 22** (⋆). *Let $G = (V, E)$ be a pseudo-forest and let $V_1 = \{v \in V(G) | deg_G(v) \leq 1\}$ and $V_3 = \{v \in V(G) | deg_G(v) \geq 3\}$. Then, $|V_3| \leq |V_1|$.*

Using the Observation 22, we have the following lemma.

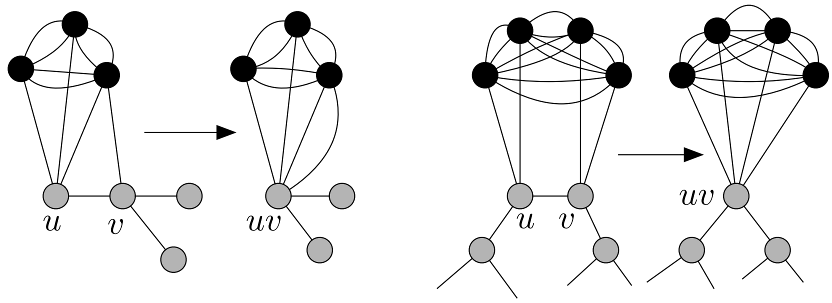

Figure 1 Illustration of Reduction Rule 21.

▶ **Lemma 23.** *When Reduction Rules 7, 8, 9, 10, 19, 20 and 21 are not applicable,* $|H_1 \cup H_3| = 2k^2 + 2(k+1)\binom{k}{2}$.

Proof. We know that $H_1 \cup H_3 \subseteq V(F_1 \cup F_2)$. Since Reduction Rules 7, 8, 9 are not applicable, every vertex in H_1 has at least two neighbours in S. As Reduction Rule 21 is not applicable, for every vertex $v \in H_1$, we associate either $z \in S$ when (x, z) is a double-edge. Otherwise we associate $(x, y) \in \binom{S}{2}$ for v, when $x, y \in N_G(v) \cap S$ and (x, y) is not a double-edge. As Reduction Rule 19 is not applicable, for every $z \in S$, there are at most $|S|$ vertices in H_1 that are connected by a double-edge. As Reduction Rule 20 is not applicable, for every $(x, y) \in \binom{S}{2}$ and (x, y) is not a double-edge, $N_G(x) \cap N_G(y)$ contain at most $|S| + 1$ vertices of H_1. Then $|H_1| \leq k^2 + (k+1)\binom{k}{2}$. By Observation 22, we know that $|H_3| \leq |H_1| \leq k^2 + (k+1)\binom{k}{2}$. So, $|H_1 \cup H_3| \leq 2k^2 + 2(k+1)\binom{k}{2}$. ◀

4.3 Bounding the number of components in F_3 and F_4

Now, what remains is to get an upper bound on $|H_2|$. Towards that, we need to bound $\hat{M}$ which requires to use an upper bound on the number of induced cycles, i.e. the number of components in F_4, i.e. c_4. We also need to bound c_3 to bound the number of vertices in H_2. In addition, we also need to use some facts from the earlier Subsection 4.1. By definition, for any component of F_3 and F_4, no vertex has exactly one neighbour in F. In particular the graph induced on the set of components of F_3 and F_4 is a two regular graph. To get an upper bound on number of such components, we need to do a little more work. We recall the following concept due to Jansen et al. [16].

▶ **Definition 24.** Let C be a connected component in $F_3 \cup F_4$ and let $X \subseteq N_G(C) \cap S$. We say that C *can be resolved with respect to* X if there exists $u \in C$ such that $C \setminus \{u\}$ is acyclic and for every connected component C' in $C \setminus \{u\}$, $|N_G(C') \cap X| \leq 1, |N_G(X) \cap C'| \leq 1$ and $G[(C \setminus \{u\}) \cup X]$ has no cycle.

The idea is that if a component C can be resolved with respect to its neighbourhood in S, then we can just delete that component and reduce the budget by 1. Every connected component of F_3 and F_4 are just induced cycles. When Reduction Rules 8, 9, 10 are not applicable, we show that the components in F_3 and F_4 have the following properties. A variation of the following lemma is provided in [16]. Here we provide an improved version of their lemma by using more facts that are useful for our purpose.

▶ **Lemma 25** ($\star$). *Let C be a connected component in $F_3 \cup F_4$ and Reduction Rules 8, 9 be not applicable. Then, if there exists $X \subseteq N_G(C) \cap S$ such that C can not be resolved with respect to X then the followings statements are true.*

- *If $C \in F_3$, then there exists $X' \subseteq X, |X'| \leq 4$ such that C can not be resolved with respect to X'.*
- *If $C \in F_4$, then there exists $X' \subseteq X, |X'| \leq 3$ such that C can not be resolved with respect to X'.*

Now, the idea behind the proof of Lemma 25 is that if for some $X \subseteq S, |X| \leq 3$ (or $|X| \leq 4$), there are a large number of components that can not be resolved with respect to X, then any minimum feedback vertex set must intersect X. Therefore, we have the following definition (also available in [16]).

▶ **Definition 26.** Let (G, S, ℓ) be an instance of FVS-Pseudo-Forest. We say that $X \subseteq S, |X| \leq 4$ (respectively $|X| \leq 3$), be such that at least t connected components in F_3 (respectively F_4) can not be resolved with respect to X, then we say that X *is saturated by t unresolvable components* in F_3 (respectively F_4).

▶ **Lemma 27** (⋆).
- *Let (G, S, ℓ) be an instance of* FVS-Pseudo-Forest *and $A \subseteq S, |A| \leq 3$ and A is saturated by $|S| + 4$ components in F_4, then any minimum feedback vertex set of G must intersect A.*
- *Let (G, S, ℓ) be an instance of* FVS-Pseudo-Forest *and $A \subseteq S, |A| \leq 4$ and A is saturated by $|S| + 7$ components in F_3, then any minimum feedback vertex set of G must intersect A.*

Now, we have the following Reduction Rule that follows from Lemma 27.

▶ **Reduction Rule 28.**
- *Let C be a connected component of F_3. If for each $A \subseteq \binom{S \cap N_G(C)}{\leq 4}$, component C can be resolved with respect to A or A is saturated by at least $|S| + 8$ non-resolvable components in F_3, then delete C and reduce ℓ by* 1
- *Let C be a connected component of F_4. If for each $A \subseteq \binom{S \cap N_G(C)}{\leq 3}$, component C can be resolved with respect to A or A is saturated by at least $|S| + 5$ non-resolvable components in F_4, then delete C and reduce ℓ by* 1.

▶ **Lemma 29** (⋆). *Reduction Rules 19, 20, 21 and 28 are safe and can be implemented in polynomial time.*

Now, we have the following lemma when the above reduction rules are not applicable.

▶ **Lemma 30.** *Recall that the number of components in F_3, F_4 are c_3, c_4 respectively. When Reduction Rule 7, 8, 9, 10 28 are not applicable, then*
- $c_3 \leq (k+7) \sum_{i=1}^{4} \binom{k}{i}$.
- $c_4 \leq (k+4) \sum_{i=1}^{3} \binom{k}{i}$.

Proof. Assume that the conditions hold. We prove the statement in the given order.
- Since Reduction Rule 28 is not applicable, for each component C in F_3, there is a set $A \in \binom{N_G(C) \cap S}{\leq 4} \subseteq \binom{|S|}{4}$ such that C can be resolved with respect to A and A is saturated by at most $|S| + 7$ components. Then, for each component C in F_3, we choose one such set A and charge C to A. Clearly we can charge to every set A at most $|S| + 7$ times, otherwise some set A would be saturated by $|S| + 8$ components. Hence the number of components in F_3 is at most $(|S| + 7) \sum_{i=1}^{4} \binom{|S|}{i} \leq (k+7) \sum_{i=1}^{4} \binom{k}{i}$.

- Similarly we can show that the number of components in F_4 is $(|S|+4)\sum_{i=1}^{3}\binom{|S|}{i} \leq (k+4)\sum_{i=1}^{3}\binom{k}{i}$. ◀

The following is an easy consequence of the Lemma 30.

▶ **Corollary 31.** *When Reduction Rule 7, 8, 9, 10 28 are not applicable, then, the number of vertices in F_3 is at most $2(k+7)\sum_{i=1}^{4}\binom{k}{i}$.*

4.4 Bounding $|H_2|$ and Putting Things together

Now, use the results in the earlier section and proceed to get an upper bound on the number of vertices in H_2. We need few more structural properties of pseudo-forests for that. As any component in F_2 has at least one vertex who has exactly one neighbour in F, number of components in F_2, i.e. $c_2 \leq |H_1|$. So, we have the following lemma.

▶ **Lemma 32.** *Recall that the number of components in F_2 is c_2. Then $c_2 \leq |H_1|$.*

Proof. Note that any component in F_2 must have at least one vertex from H_1. Therefore, the number of components in F_2 is at most $|H_1|$. ◀

Recall that in order to bound $|H_2|$, we also need an upper bound on the number of degree-2-paths in $\mathcal{P}$. The following is a structural property of pseudo-forest which helps to do so.

▶ **Observation 33** ($\star$)**.** *Let $G=(V,E)$ be a pseudo-forest where every component has at least one vertex of degree 1. Let $V_1 = \{v \in V(G) | deg_G(v) \leq 1\}$ and $V_3 = \{v \in V(G) | deg_G(v) \geq 3\}$ and $\mathcal{P}$ be the set of maximal degree-2-paths in G. Then, $|\mathcal{P}| \leq |V_3| + |V_1|$.*

▶ **Lemma 34.** *If Reduction Rule 7, 8, 9, 10, 19, 20, 21, 28 are not applicable, then the number of vertices in $|H_2| = \mathcal{O}(k^6)$.*

Proof. By Corollary 31, we have that $|V(F_3)| = \mathcal{O}(k^5)$. Recall that by Lemma 32, we have that $c_2 = \mathcal{O}(k^3)$. Also by Lemma 30, we have that $c_4 = \mathcal{O}(k^4)$. So, $\hat{c} = c_2 + c_4 = \mathcal{O}(k^4)$. Recall that $\hat{M}$ be the maximum matching in $\mathcal{P} \cup F_4$. As Reduction Rule 9 is not applicable, every vertex in H_2 has at least one neighbour in S. As Reduction Rule 21 is not applicable, for every $(u,v) \in \hat{M}$, we associate $z \in N_G(u) \cap N_G(v) \cap S$ when $N_G(u) \cap N_G(v) \cap S \neq \emptyset$ or (u,z) is a double-edge. Otherwise, we associate $x \in N_G(u) \cap S, y \in N_G(v) \cap S$ such that (x,y) is not a double-edge. For any $z \in S$, define $Matched(z) = \{(u,v) \in \hat{M} | (u,z)$ is a double-edge or $u,v \in N_G(z)\}$. As Reduction Rule 19 is not applicable, for every $z \in S$, $|Matched(z)| \leq 2|S| + 2\hat{c}$. Similarly as Reduction Rule 20 is not applicable, for every $x,y \in \binom{S}{2}$ and (x,y) is not a double-edge, $N_G(x) \cup N_G(y)$ contain both end points of at most $|S|+\hat{c}+1$ edges of $\hat{M}$. Since $\hat{c} = \mathcal{O}(k^4)$, we have that $|\hat{M}| \leq |S|(2|S|+2\hat{c}) + \binom{S}{2}(|S|+\hat{c}+1) \leq 2k^2 + 2k \cdot \mathcal{O}(k^4) + \binom{k}{2} \cdot \mathcal{O}(k^4) = \mathcal{O}(k^6)$. A *maximal degree-2-path* can also have only one vertex which is not matched by $\hat{M}$. Recall that $\mathcal{P}$ be the collection of all *maximal degree-2-paths* in $F_1 \cup F_2$. Using Observation 33, we get that $|\mathcal{P}| \leq |H_1| + |H_3| = \mathcal{O}(k^3)$. So, the number of vertices in H_2 that are not matched by $\hat{M}$ is at most $|\mathcal{P}| + |\hat{M}| = \mathcal{O}(k^6)$. So, $|H_2| = \mathcal{O}(k^6)$. ◀

The following is the main theorem of this section and this is an easy consequence of Lemma 23 and Lemma 34.

▶ **Theorem 35.** FVS-PSEUDO-FOREST *has a kernel consisting of $\mathcal{O}(k^6)$ vertices.*

5 Kernelization of Feedback Vertex Set Parameterized by Deletion distance to bounded Mock Forest

Now we consider the Feedback Vertex Set problem parameterized by the size of a deletion set whose deletion results in a *mock-d-forest*. Recall that a graph is called mock-d-forest when every vertex is contained in at most one cycle and every connected component has at most d cycles. Formal definition of the problem is given below.

> FVS-Mock-d-Forest for $d \geq 2$ and d is a constant **Parameter:** k
> **Input:** An undirected graph G, $S \subseteq V(G)$ of size at most k such that $G[V(G) \setminus S]$ is a graph of which every vertex participates in at most most one cycle, every component has at most d cycles for some constant d and an integer ℓ.
> **Question:** Does G have a feedback vertex set of size at most ℓ?

When d is not bounded, then there is no polynomial kernel unless NP $\subseteq$ coNP/poly. In this section, we first provide a polynomial kernel for this problem when d is a constant and $d \geq 2$. After that we provide a lower bound for this problem.

5.1 Polynomial Kernel for FVS-Mock-d-Forest

Our kernelization algorithm follows along the line of the kernel for FVS-Pseudo-Forest in the earlier section. Here, we need to use some special properties of mock-d-forest. We use $F = G \setminus S$ throughout the section. Let $\mathcal{P}_F$ be the collection of maximal acyclic degree-2-paths in F. Let M_F be a maximum matching in $\mathcal{P}_F$ and set of induced cycles in F. Let $\hat{c}$ be the total number of cycles in F. We partition $V(F)$ into three parts as follows.

- $F_1 = \{u \in V(F) | deg_F(u) \leq 1\}$.
- $F_2 = \{u \in V(F) | deg_F(u) = 2\}$.
- $F_3 = \{u \in V(F) | deg_F(u) \geq 3\}$.

Our first step is to bound the number of vertices in F_1. An upper bound on F_1 along with some properties of pseudo-forest, we get an upper bound on the number of vertices of F_3. Then, we have to bound the number of edges in M_F and the number of maximal acyclic degree-2-paths in $\mathcal{P}_F$. Now, we are ready to state the Reduction Rules. Our Reduction Rules in this section are generalisations of the Reduction Rules in Section 4. We apply the following two reduction rules that are more general variant of Reduction Rules 19, 20.

▶ **Reduction Rule 36.** *Let $x \in S$. Then $G' \leftarrow G \setminus \{x\}, \ell' \leftarrow \ell - 1$ if one of the following conditions is satisfied.*

- *There are at least $|S| + 1$ vertices in F_1 that are connected to x by a double-edge.*
- *There are at least $|S| + \hat{c} + 1$ vertices in F_2 that are matched by M_F and are connected to x by a double-edge.*
- *$N_G(x)$ contains both end points of at least $|S| + \hat{c} + 1$ edges in M_F.*

▶ **Reduction Rule 37.** *Let $(x, y) \in \binom{S}{2}$. Then make (x, y) into a double-edge if one of the following conditions is satisfied.*

- *$N_G(x) \cap N_G(y)$ contains at least $|S| + 1$ vertices from F_1.*
- *$N_G(x) \cap N_G(y)$ contains both end points of at least $|S| + \hat{c} + 2$ edges of M_F.*

We apply Reduction Rules 7, 8, 9, 10, 36, 37, 21 in the this order (Recall that we did similar in Section 4).

▶ **Lemma 38** ($\star$). *When Reduction Rules 7 8, 9, 10, 36, 37, 21 are not applicable, then $|F_1| = \mathcal{O}(k^3)$.*

We need to bound the number of vertices in F_2 and F_3. To have that, we need to bound the number of components in F. We need to do little more work for that. In particular, we need the following definition which is a generalisation of Definition 24.

▶ **Definition 39.** Let C be a connected component in F and let $X \subseteq N_G(C) \cap S$. We say that C *can be resolved with respect to* X if there exists $\{u_1, u_2, \ldots, u_d\} \subseteq C$ such that $C \setminus \{u_1, \ldots, u_d\}$ is acyclic and for every connected component C' in $C \setminus \{u_1, \ldots, u_d\}$, $|N_G(C') \cap X| \leq 1, |N_G(X) \cap C'| \leq 1$ and $G[(C \setminus \{u_1, \ldots, u_d\}) \cup X]$ has no cycle.

The following lemma is a generalisation of Lemma 25. It is useful to bound the number of components in F.

▶ **Lemma 40** (⋆). *Let C be a connected component of F having exactly d cycles and let $X \subseteq N_G(C) \cap S$ such that C can not be resolved with respect to X. Then, there exists $X' \subseteq X, |X'| \leq 3d$ such that C can not be resolved with respect to X'.*

The following definition is a generalisation of Definition 26 in Section 4.

▶ **Definition 41.** Let $A \subseteq S, |A| \leq 3d$ be such that there are t components in F that can not be resolved with respect to A. Then, we say that A *is saturated by t components* in F.

The following lemma is a property of mock-forest. We will need this to prove the safeness of the Reduction Rule 43.

▶ **Lemma 42** (⋆). *Let (G, S, ℓ) be an instance of* FVS-Mock-d-Forest] *and $A \subseteq S, |A| \leq 3d$ and A is saturated by $|S| + \binom{3d}{2} + 1$ components in F, then any minimum feedback vertex set of G must intersect A.*

Now, we have just one more reduction rule to get an upper bound on the number of components in F. And Lemma 45 is a consequence of inapplicability of Reduction Rule 43.

▶ **Reduction Rule 43.** *Let C be a connected component in F that contains some cycle. If for each $A \in \binom{N_G(C) \cap X}{\leq 3d}$, component C can be resolved with respect to A or A is saturated by $|S| + \binom{3d}{2} + 2$ components, then remove C and reduce ℓ by the number of cycles in C.*

▶ **Lemma 44** (⋆). *Reduction Rules 36, 37 and 43 are safe and can be implemented in polynomial time.*

▶ **Lemma 45.** *Let (G, S, ℓ) be an irreducible instance with respect to Reduction Rule 43, then number of components in F is at most $\mathcal{O}(|S|^{3d+1})$.*

Proof. Consider any component $C \in F$. Reduction Rule 43 is not applicable, therefore, there exists $A \subseteq S, |A| \leq 3d$ such that C can not be resolved with respect to A. Also, for the same reason, A can be saturated by at most $|S| + \binom{3d}{2} + 1$ components. Therefore, the number of components is at most $(|S| + \binom{3d}{2})\binom{|S|}{3d} \leq 9d^2.|S|^{3d+1} = \mathcal{O}(d^2.|S|^{3d+1})$. ◀

We have bounded the number of components in F. We already have bounded the number of vertices in F_1. We are left to bound $|F_3 \cup F_2|$. We need graph theoretic properties of mock-d-forest to get an upper bound on $|F_3|$. Recall that in Section 4, we used observations about pseudo-forest. Similarly, in this section, we use observations about mock-forest when there are at most d cycles in a mock-forest.

▶ **Observation 46** (⋆). *Let $G = (V, E)$ be a mock forest with c components where every component has at most d cycles. $V_1 = \{v \in V(G) | deg_G(v) \leq 1\}, V_2 = \{v \in V(G) | deg_G(v) = 2\}, V_3 = \{v \in V(G) | deg_G(v) \geq 3\}$. Then $|V_3| \leq |V_1| + 2cd - 2c$.*

Using Lemma 38 and Observation 46, we have the following Lemma.

▶ **Lemma 47.** *Let c be the number of components in F. Then, $|F_3| = \mathcal{O}(k^{3d+1})$.*

Proof. By Lemma 38, we know that $|F_1| = \mathcal{O}(k^3)$. Now, by Observation 46, we know that $|F_3| \leq |F_1| + 2c(d-1)$. Recall that $c = \mathcal{O}(k^{3d+1})$. Now, $c(d-1) \leq \hat{c} = \mathcal{O}(k^{3d+1})$. So, $|F_3| = \mathcal{O}(k^{3d+1})$. ◀

Now, what remains is to bound the number of vertices in F_2. For that, we need to bound M_F and also the number of maximal acyclic degree-2-paths in $\mathcal{P}_F$. Using structural properties of mock-d-forest, we have the following lemma that bounds the number of maximal acyclic degree-2-paths in F, i.e. $\mathcal{P}_F$.

▶ **Lemma 48 (⋆).** $|\mathcal{P}_F| = \mathcal{O}(k^{3d+1})$ *where c' is the number of components in F that have at least two cycles.*

Using the above observations and lemmas we have the following lemma.

▶ **Lemma 49 (⋆).** $|F_2| = \mathcal{O}(|S|^{3d+3})$.

Combining Lemma 38, 47, 49, we get the following theorem.

▶ **Theorem 50.** FVS-MOCK-d-FOREST *has a kernel consisting of $\mathcal{O}(k^{3d+3})$ vertices.*

5.2 Kernel Lower Bound for FVS-Mock-d-Forest

We provide a polynomial parameter transformation from $(d+2)$-CNF-SAT parameterized by the number of variables to FEEDBACK VERTEX SET parameterized by deletion distance to Mock-d-Forest where $d \geq 2$. A polynomial parameter transformation from CNF-SAT to FVS-MOCK-FOREST when every clause has exactly r literals where r is a power of 2 is already known [16]. We modify the construction for a polynomial parameter transformation from $(d+2)$-CNF-SAT to FVS-MOCK-d-FOREST where d is not necessarily a power of 2.

Let the clause C_i have $d_i \leq d+2$ literals. We provide a clause gadget of height j_i where $2^{j_i - 1} < d_i \leq 2^{j_i}$. We create d^2 many copies for this gadget. In this gadget, the terminal vertices are the corresponding vertices of literals (See figure 2). For clause C_q with its r'th copy, we name literals as $y_{q,r,1}, \ldots, y_{q,r,d_i}$. And we create a variable gadget for variable x_i as a cycle of 3 vertices. Let $\{t_i, f_i, e_i\}$ are those vertices. We define $S = \bigcup_{i=1}^{n} \{t_i, f_i, e_i\}$. Let $y_{q,r,j}$ be the j'th literal of clause C_q. Let the variable corresponding to that variable is x_i. Then, if the literal $y_{q,r,j}$ is $\bar{x_i}$, then we connect $y_{q,r,j}$ with f_i. Otherwise we connect $y_{q,r,j}$ with t_i. We do the same for every $r \in [d^2]$. We set $\ell = d^2 \sum_{i=1}^{m} (d_i - 2)$.

▶ **Lemma 51 (⋆).** *Let ϕ be a $(d+2)$-CNF formula. Let G_ϕ be the graph constructed from ϕ using the construction above. Then ϕ is satisfiable if and only if (G_ϕ, S, ℓ) is* YES-INSTANCE. *Thus there is a polynomial parameter transformation from $(d+2)$-CNF-SAT to* FVS-MOCK-d-FOREST.

▶ **Theorem 52.** *[[7]]* d-CNF-SAT *parameterized by n, the number of variables, has no kernel of size $\mathcal{O}(n^{d-\epsilon})$ for any $d \geq 3, \epsilon > 0$ unless* NP ⊆ coNP/poly.

Using Lemma 51 and Theorem 52, we have the following theorem.

▶ **Theorem 53.** FVS-d-MOCK-FOREST *has no kernel consisting of $\mathcal{O}(k^{d+2-\epsilon})$ vertices for every $d \geq 2, \epsilon > 0$ unless* NP ⊆ coNP/poly.

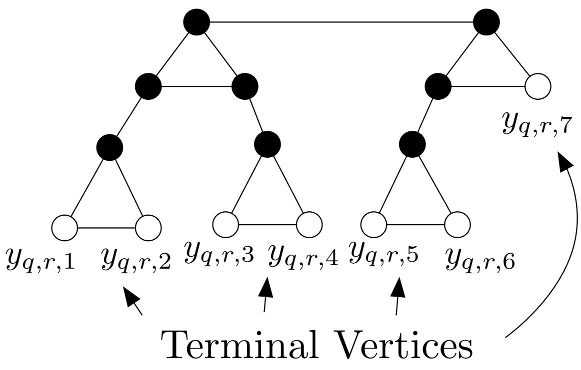

Figure 2 Illustration of Clause Gadget Construction for 7 literals.

6 Conclusion

We have given a kernel with $\mathcal{O}(k^6)$ vertices for FVS-PSEUDO-FOREST improving from an earlier $\mathcal{O}(k^{10})$ bound [16], and narrowing the gap with the $\Omega(k^4)$ conditional lower bound. Bridging the gap further is an interesting problem. We proved that FVS-HIGH-DEGREE is fixed-parameter tractable. Status of Feedback Vertex Set parameterized by deletion distance to a (sub)-cubic graph (a related problem) remains open, and we do not even know an XP algorithm for the problem. We considered FVS parameterized by deletion distance to mock-d-forest and proved an upper bound of $\mathcal{O}(k^{3d+3})$ and lower bound $\Omega(k^{d+2})$ under complexity theoretic assumptions. Narrowing this gap is another interesting future direction.

References

1 E. Balas and C. S. Yu. On graphs with polynomially solvable maximum-weight clique problem. *Networks*, 19(2):247–253, 1989.

2 H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014.

3 H. L. Bodlaender, S. Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011.

4 H. L. Bodlaender and T. C. van Dijk. A cubic kernel for feedback vertex set and loop cutset. *Theory Comput. Syst.*, 46(3):566–597, 2010.

5 Y. Cao, J. Chen, and Y. Liu. On feedback vertex set: New measure and new structures. *Algorithmica*, 73(1):63–86, 2015.

6 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

7 H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014.

8 R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

9 M. R. Fellows, B. M. P. Jansen, and F. A. Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *Eur. J. Comb.*, 34(3):541–566, 2013.

10 F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh. Planar F-Deletion: Approximation, Kernelization and Optimal FPT Algorithms. In *Proceedings of FOCS*, pages 470–479, 2012.

11 L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.

12 T. Fujito. A unified approximation algorithm for node-deletion problems. *Discrete Applied Mathematics*, 86(2-3):213–231, 1998.

13 M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

14 G. Gutin, E. J. Kim, M. Lampis, and V. Mitsou. Vertex cover problem parameterized above and below tight bounds. *Theory Comput. Syst.*, 48(2):402–410, 2011.

15 B. M. P. Jansen and S. Kratsch. Data reduction for graph coloring problems. *Inf. Comput.*, 231:70–88, 2013.

16 B. M. P. Jansen, V. Raman, and M. Vatshelle. Parameter ecology for feedback vertex set. *Tsinghua Science and Technology*, 19(4):387–409, 2014.

17 T. Kloks, C. Liu, and S. Poon. Feedback vertex set on chordal bipartite graphs. *CoRR*, abs/1104.3915v2, 2011.

18 T. Kociumaka and M. Pilipczuk. Faster deterministic feedback vertex set. *Inf. Process. Lett.*, 114(10):556–560, 2014.

19 D. Kratsch, H. Müller, and I. Todinca. Feedback vertex set on at-free graphs. *Discrete Applied Mathematics*, 156(10):1936–1947, 2008.

20 R. Rizzi. Minimum weakly fundamental cycle bases are hard to find. *Algorithmica*, 53(3):402–424, 2009.

21 S. Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Trans. Algorithms*, 6(2), 2010.

22 S. Ueno, Y. Kajitani, and S. Gotoh. On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. *Discrete Mathematics*, 72(1-3):355–360, 1988.