

# A Polynomial Kernel for Funnel Arc Deletion Set

Marcelo Garlet Milani 

Technische Universität Berlin, Chair of Logic and Semantics, Germany

m.garletmillani@tu-berlin.de

---

## Abstract

In DIRECTED FEEDBACK ARC SET (DFAS) we search for a set of at most  $k$  arcs which intersect every cycle in the input digraph. It is a well-known open problem in parameterized complexity to decide if DFAS admits a kernel of polynomial size. We consider  $\mathcal{C}$ -ARC DELETION SET ( $\mathcal{C}$ -ADS), a variant of DFAS where we want to remove at most  $k$  arcs from the input digraph in order to turn it into a digraph of a class  $\mathcal{C}$ . In this work, we choose  $\mathcal{C}$  to be the class of *funnels*. FUNNEL-ADS is NP-hard even if the input is a DAG, but is fixed-parameter tractable with respect to  $k$ . So far no polynomial kernel for this problem was known. Our main result is a kernel for FUNNEL-ADS with  $\mathcal{O}(k^6)$  many vertices and  $\mathcal{O}(k^7)$  many arcs, computable in  $\mathcal{O}(nm)$  time, where  $n$  is the number of vertices and  $m$  the number of arcs of the input digraph.

**2012 ACM Subject Classification** Theory of computation → Fixed parameter tractability

**Keywords and phrases** graph editing, directed feedback arc set, parameterized algorithm, kernels, funnels

**Digital Object Identifier** 10.4230/LIPIcs.IPEC.2020.13

**Related Version** A full version of the paper is available at <https://arxiv.org/abs/1911.05520>.

**Acknowledgements** We thank the referees for their numerous helpful comments.

## 1 Introduction

In graph editing problems, we are given a (directed or undirected) graph  $G$  and a number  $k$ , and we search for a set of at most  $k$  vertices, edges or arcs whose removal or addition produces a graph with a desired property. There are several variants of these problems, and in this paper we consider the problem of removing arcs from a digraph in order to obtain a digraph in a given class  $\mathcal{C}$ . When  $\mathcal{C}$  is the class of all directed acyclic graphs (DAGs), the problem is called DIRECTED FEEDBACK ARC SET (DFAS). If we remove vertices instead of arcs, the problem is called DIRECTED FEEDBACK VERTEX SET (DFVS).

There are simple reductions between DFAS and DFVS. We can reduce DFAS to DFVS by taking the line digraph of the input. Removing a vertex from the reduced instance corresponds to removing an arc from the input instance and vice versa. For a reduction in the other direction, we split each vertex  $v$  into two vertices, say,  $v_o$  and  $v_i$ , connect them with an arc  $(v_i, v_o)$  and shift all outgoing arcs of  $v$  to  $v_o$  and all incoming arcs to  $v_i$ . In the context of *parameterized complexity*, such reductions are called *parameterized* as the parameter  $k$  is preserved. Hence, parameterized results are often stated for DFVS.

In a breakthrough paper it was proven that there is an algorithm for DFVS with running time  $4^k k! \cdot n^{\mathcal{O}(1)}$  [4], showing that the problem is *fixed-parameter tractable* (FPT) with respect to  $k$ . After obtaining an FPT result, it is natural to ask if the problem also admits a polynomial *kernel*, that is, if there is a polynomial-time algorithm which reduces the input instance to an instance of size at most  $\mathcal{O}(k^c)$  for some constant  $c$ . Such an algorithm is called a *kernelization* algorithm.



© Marcelo Garlet Milani;

licensed under Creative Commons License CC-BY

15th International Symposium on Parameterized and Exact Computation (IPEC 2020).

Editors: Yixin Cao and Marcin Pilipczuk; Article No. 13; pp. 13:1–13:13

Leibniz International Proceedings in Informatics



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

The existence of a polynomial kernel for DFVS is a fundamental open question in the field of parameterized complexity. One approach towards solving this question is to consider different parametrizations or restrictions of the input digraph. By considering progressively smaller parameters or more general digraph classes, one can hope to eventually close the gap between the restricted cases and the general case of DFVS.

On tournaments, DFVS admits a polynomial kernel [1]; this was extended to generalizations of tournaments as well [3]. When parameterized by solution size  $k$  and the size  $\ell$  of a treewidth  $\eta$ -modulator, DFVS admits a kernel of size  $(k \cdot \ell)^{\mathcal{O}(\eta^2)}$  [10].

One can also restrict the output instead, that is, we can consider  $\mathcal{C}$ -VERTEX DELETION SET ( $\mathcal{C}$ -VDS) or  $\mathcal{C}$ -ARC DELETION SET ( $\mathcal{C}$ -ADS), where, for a fixed digraph class  $\mathcal{C}$ , we search for a set of at most  $k$  vertices (arcs) whose removal turns the input into a digraph in  $\mathcal{C}$ . Unlike DFVS and DFAS,  $\mathcal{C}$ -VDS and  $\mathcal{C}$ -ADS can belong to different complexity classes depending on  $\mathcal{C}$ : While OUT-FOREST-ADS can be solved in polynomial time, OUT-FOREST-VDS is NP-hard [12]. Further, note that even if  $\mathcal{C}' \subseteq \mathcal{C}$ , a polynomial kernel for  $\mathcal{C}$ -ADS does not immediately imply a polynomial kernel for  $\mathcal{C}'$ -ADS, and the implication also does not work in the other direction. Indeed, while the problem is trivial when  $\mathcal{C}$  is the class of all independent sets or the class of all digraphs, it is NP-hard if  $\mathcal{C}$  is the class of DAGs, which contains all independent sets and is a subclass of all digraphs. In a sense, the complexity landscapes of  $\mathcal{C}$ -ADS and  $\mathcal{C}$ -VDS are much more fine-grained than the landscape of DFVS, and may allow for smaller steps towards more general results.

OUT-FOREST-ADS and PUMPKIN-ADS can be solved in polynomial time [12], while OUT-FOREST-VDS and PUMPKIN-VDS are NP-hard and admit polynomial kernels [2, 12] of size  $\mathcal{O}(k^2)$  and  $\mathcal{O}(k^3)$ , respectively [2].  $\mathcal{F}_\eta$ -VDS admits a polynomial kernel for constant  $\eta$ , where  $\mathcal{F}_\eta$  is the class of all digraphs with (undirected) treewidth at most  $\eta$  [10].

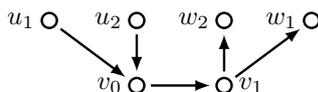
In this work we consider FUNNEL-ADS and provide a polynomial kernel with  $\mathcal{O}(k^6)$  vertices and  $\mathcal{O}(k^7)$  arcs. A digraph is a funnel if it is a DAG and every source to sink path has an arc which is not in any other source to sink path. FUNNEL-ADS is NP-hard even if the input is DAG, but it can be solved in  $\mathcal{O}(3^k \cdot (n + m))$  time [11], where  $k$  is the solution size. Out-forests and pumpkins are also funnels, but there are also dense funnels like complete bipartite digraphs (where all arcs go from the first partition to the second but not back).

Our results rely on characterizations for funnels based on forbidden subgraphs and on a “labeling” of the vertices [11]. We believe the techniques used here can be generalized to other digraph classes which are also similarly characterized, and hope they provide further insight about the classes  $\mathcal{C}$  for which  $\mathcal{C}$ -ADS admits a polynomial kernel.

## 2 Preliminaries

A (partial) function  $f : A \rightarrow B$  is a set of tuples  $(a, f(a)) \in A \times B$  where for every  $a \in A$  there is at most one  $b \in B$  with  $(a, b) \in f$  (that is,  $f(a) = b$ ). We write  $\text{Dom}(f)$  for the set of values  $a \in A$  for which  $f$  is defined. Hence,  $\emptyset$  is the undefined function, and  $f' \supseteq f$  if  $f'(x) = f(x)$  for every  $x \in \text{Dom}(f)$ . All our functions are *partial*, that is,  $\text{Dom}(f)$  is not necessarily  $A$ .

A *parameterized language*  $L$  is *fixed-parameter tractable* with respect to the parameter  $k$  if there is some algorithm with running time  $f(k) \cdot n^{\mathcal{O}(1)}$  deciding whether  $(x, k) \in L$ , where  $f$  is some computable function,  $n = |x|$  and  $k$  is the parameter (refer to [5, 6] for an introduction to parameterized complexity). We say that  $L$  *admits a problem kernel* if there is a polynomial-time algorithm which transforms an instance  $(x, k)$  into an instance  $(x', k')$  such that  $(x, k) \in L$  if and only if  $(x', k') \in L$ ,  $k' \leq k$  and  $|x'| \leq f(k)$  for some computable function  $f$ . If  $f$  is a polynomial, we say that  $L$  *admits a polynomial kernel* with respect to  $k$ .



■ **Figure 1**  $D_1$ , a forbidden subgraph for funnels.

When describing a kernelization algorithm, it is common to define *reduction rules*. These rules have a *condition* and an *effect*, and we say that a reduction rule is *applicable* if the condition is true. The effect of the reduction rule produces a new instance  $(x', k')$  of the problem, and a rule is said to be *safe* if  $(x', k') \in L$  if and only if the original instance is in  $L$ . We refer the reader to [8, 9] for surveys on kernelization and to [7] for a book on the topic.

We only consider directed graphs (digraphs) without loops or parallel arcs (but we allow arcs in opposite directions) in this paper. Let  $D$  be a digraph. The set of arcs of  $D$  is denoted by  $A(D)$ , and its set of vertices is  $V(D)$ . The set of outneighbors (inneighbors) in  $D$  of a vertex  $v \in V(D)$  is denoted by  $\text{out}_D(v)$  ( $\text{in}_D(v)$ ); the outdegree (indegree) of  $v$  is  $\text{outdeg}_D(v) = |\text{out}_D(v)|$  ( $\text{indeg}_D(v) = |\text{in}_D(v)|$ ). If the digraph  $D$  is clear from context, we omit it from the index. For a set  $U \subseteq V(D)$  we write  $\text{out}(U)$  for the set  $\{\text{out}(u) \mid u \in U\} \setminus U$  (and analogously for  $\text{in}(U)$ ). A vertex  $v$  is a *source* if  $\text{indeg}(v) = 0$ , and it is a *sink* if  $\text{outdeg}(v) = 0$ . We write  $H \subseteq D$  if  $H$  a *subgraph* of  $D$ ; the subgraph of  $D$  *induced* by  $U$  is given by  $D[U]$ . We write  $D - X$  for the operation of deleting a set of vertices or arcs  $X$  from  $D$ . Similarly, we add a set of arcs or vertices to  $D$  with  $D + X$ .

A *directed acyclic graph* (DAG) is a digraph which does not contain any directed cycle. A digraph  $D$  is a *funnel* if  $D$  is a DAG and for every path  $P$  from a source to a sink of  $D$  of length at least one there is some arc  $a \in A(P)$  such that for any different path  $Q$  from a (possibly different) source to a sink we have  $a \notin A(Q)$ . We repeat below several known characterizations for funnels, as they are particularly useful for our results.

► **Theorem 1** ([11], Theorem 1). *Let  $D$  be a DAG. The following statements are equivalent.*

- a.  $D$  is a *funnel*.
- b.  $V(D)$  can be partitioned into two sets  $F$  and  $M$  such that: (1)  $F$  induces an *out-forest*; (2)  $M$  induces an *in-forest*; and (3)  $(M \times F) \cap A(D) = \emptyset$ .
- c. No digraph in  $\mathcal{F} = \{D_i \mid i \in \{0, 1, \dots\}\}$  is contained in  $D$  as a (not necessarily induced) subgraph, where (see Figure 1 for an example)
  - $V(D_k) = \{u_1, u_2, w_1, w_2\} \cup \{v_i \mid 0 \leq i \leq k\}$ , and
  - $A(D_k) = \{(u_1, v_0), (u_2, v_0), (v_k, w_1), (v_k, w_2)\} \cup \{(v_i, v_{i+1}) \mid 1 \leq i \leq k-1\}$
- d.  $D$  does not contain  $D_0$  as a *butterfly minor*.

The digraphs in  $\mathcal{F}$  are called *forbidden subgraphs for funnels*. For a digraph  $D$  we define a *labeling* as a function  $\ell : V(D) \rightarrow \{F, M\}$ . We say that  $\ell$  is a *funnel labeling* for  $D$  if  $\text{Dom}(\ell) = V(D)$ , the set  $F = \{v \in V(D) \mid \ell(v) = F\}$  induces an out-forest in  $D$ , the set  $M = \{v \in V(D) \mid \ell(v) = M\}$  induces an in-forest in  $D$  and  $(M \times F) \cap A(D) = \emptyset$ . Due to Theorem 1(b), a digraph  $D$  is a funnel if and only if there exists a funnel labeling for  $D$ .

In the *feedback arc set* problem, we are given a digraph  $D$  and a  $k \in \mathbb{N}$  as an input, and we search for a set  $S \subseteq A(D)$  such that  $D - S$  is a DAG and  $|S| \leq k$ . We consider a variant of this problem where we want  $D - S$  to be a funnel instead, which is formally defined below.

#### FUNNEL ARC DELETION SET (FADS)

**Input** A digraph  $D$  and a number  $k \in \mathbb{N}$ .

**Question** Is there a set  $S \subseteq A(D)$  with  $|S| \leq k$  such that  $D - S$  is a funnel?

## 13:4 A Polynomial Kernel for Funnel Arc Deletion Set

To make better use of Theorem 1(b), we consider a more general problem in which some vertices might already be labeled with F or M, and the funnel we obtain in the end must respect this labeling. Formally, the problem is defined as follows.

**FUNNEL ARC DELETION LABELING (FADL)**

**Input** A digraph  $D$ , a labeling  $\ell : V(D) \rightarrow \{F, M\}$  and a number  $k \in \mathbb{N}$ .

**Question** Are there a set  $S \subseteq A(D)$  and a labeling  $\hat{\ell} \supseteq \ell$  such that  $\hat{\ell}$  is a funnel labeling for  $D - S$  and  $|S| \leq k$ ?

We say that  $(D, \ell, k)$  is the *input instance* and  $(S, \hat{\ell})$  is a solution for the input instance. This more general version of the problem allows us to decide which label a vertex will take and encode this in the instance itself. While technically not necessary, using FADL instead of FADS simplifies the kernelization algorithm and also the proofs. Due to space constraints, proofs marked with  $(\star)$  are deferred to the full version of the paper.

### 3 Basic reduction rules

We construct our kernelization algorithm by defining a series of reduction rules and then showing that, if no reduction rule is applicable, the input size is bounded in a polynomial of  $k$ . Our strategy is to partition the vertex set into labeled and unlabeled vertices, then bound the number of unlabeled vertices (Section 3.1) and use this to bound the number of labeled vertices (Section 3.2) as well. In this section we define some reduction rules which are useful both in Section 3.1 as well as in Section 3.2. For brevity, we assume that a reduction rule is no longer applicable to the input instance after it has been defined.

Let  $(D, \ell, k)$  be the input instance. From Theorem 1(c) we can see that a funnel has no vertex  $v$  with  $\text{indeg}(v) > 1$  and  $\text{outdeg}(v) > 1$ . Further,  $\text{indeg}(v) \leq 1$  if  $\ell(v) = F$ , and  $\text{outdeg}(v) \leq 1$  if  $\ell(v) = M$ . Hence, by simply counting the number of vertices disrespecting each case, we can obtain a lower bound for the number of arcs that need to be removed from  $D$  in order to obtain a funnel. As removing one arc changes the degree of two vertices, we obtain a bound of at most  $2k$  such vertices. The safety of the following reduction rule follows easily from Theorem 1.

► **Reduction Rule 1 (Lower Bound).** Let  $V_I \subseteq V(D)$  be the set of vertices with indegree greater than one, let  $V_O$  be the set of vertices with outdegree greater than one and let  $V_X = V_O \cap V_I$ . Output a trivial “no” instance if

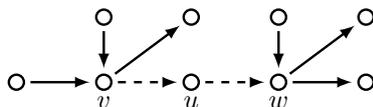
$$\sum_{u \in V_O, \ell(u)=M} (\text{outdeg}(u) - 1) + \sum_{u \in V_I, \ell(u)=F} (\text{indeg}(u) - 1) + \sum_{u \in V_X, u \notin \text{Dom}(\ell)} (\min\{\text{indeg}(u), \text{outdeg}(u)\} - 1) > 2k.$$

The following reduction rule is based on [11], with some modifications since the original reduction rule is applied as an intermediate step in an FPT algorithm and is not safe for kernelization. For certain vertices it is possible to optimally decide which label they should receive in an optimal solution. For example, vertices with outdegree greater than  $k + 1$  can always be labeled with F, as otherwise we would need to remove at least  $k + 1$  of its outgoing arcs, which is not possible.

► **Reduction Rule 2 (Set Label).** Let  $v \in V(D)$  be an unlabeled vertex.

Set  $\ell(v) := F$  if at least one of the following is true: (1)  $\text{indeg}(v) = 0$ ; (2)  $v$  has a single inneighbor  $u$  and  $\ell(u) = F$ ; (3) there are at least  $\text{indeg}(v) + 1$  vertices  $u \in \text{out}(v)$  with  $\ell(u) = M$  or  $\ell(u) = F \wedge \text{indeg}(u) = 1$ ; or (4)  $\text{outdeg}(v) > k + 1$ .

■ **Figure 2** A digraph which is not a funnel. Removing the arcs  $(v, u)$  and  $(u, w)$  results in a funnel.



Set  $\ell(v) := M$  if at least one of the following is true: (1)  $\text{outdeg}(v) = 0$ ; (2)  $v$  has a single outneighbor  $u$  and  $\ell(u) = M$ ; (3) there are at least  $\text{outdeg}(v) + 1$  vertices  $u \in \text{in}(v)$  with  $\ell(u) = F$  or  $\ell(u) = M \wedge \text{outdeg}(u) = 1$ ; or (4)  $\text{indeg}(v) > k + 1$ .

**Proof of safety of Set Label (RR 2).** Clearly, a solution for the reduced instance is also a solution for the original instance. For the other direction, we consider only the case where we set  $\ell(v) := F$ , as the other case is symmetric. Let  $\ell_r$  be the labeling obtained by the reduction rule. Let  $(S, \hat{\ell})$  be a solution for the original instance. We set  $\hat{\ell}_r := \hat{\ell}$  and  $\hat{\ell}_r(v) := F$ . If  $\hat{\ell}(v) = F$ , then clearly  $(S, \hat{\ell}_r)$  is a solution for the reduced instance. So assume  $\hat{\ell}(v) = M$ . This implies that  $\text{outdeg}(v) \leq k + 1$ , as otherwise  $|S| > k$ .

If  $\text{indeg}(v) = 0$ , or  $\text{indeg}(v) = 1$  and there is some  $u \in \text{in}(v)$  with  $\ell(u) = F$ , then  $\hat{\ell}_r$  is clearly also a funnel labeling for  $D - S$ .

Let  $U = \{u \in \text{out}(v) \mid \ell(u) = M \text{ or } \ell(u) = F \wedge \text{indeg}(u) = 1\}$ . If  $|U| \geq \text{indeg}(v) + 1$ , we construct an  $S_r$  from  $S$  as follows. We add all incoming arcs of  $v$  to  $S_r$  and remove from  $S_r$  all outgoing arcs  $(v, u)$  where  $u \in U$ . Since  $\hat{\ell}(v) = M$ , at least  $\text{outdeg}(v) - 1 \geq \text{indeg}(v)$  many outgoing arcs of  $v$  are in  $S$ . Hence, we remove at least  $\text{indeg}(v)$  arcs from  $S$  and add at most  $\text{indeg}(v)$ . Thus,  $|S_r| \leq |S|$ .

The digraph  $D - S_r$  does not contain cycles, as all incoming arcs of  $v$  were removed, so any cycle in  $D - S_r$  is also in  $D - S$ , which is a funnel. To see that  $\hat{\ell}_r$  is a funnel labeling of  $D - S_r$ , first note that we can always keep arcs  $(v, u)$  in  $D - S_r$  where  $\ell(u) = M$ . We can also keep arcs  $(v, u)$  in  $D - S_r$  where  $\ell(u) = F$  and  $\text{indeg}(u) = 1$ . As  $v$  has no incoming arcs in  $D - S_r$ , it lies in an out-forest. Hence,  $\hat{\ell}_r$  is a funnel labeling of  $D - S_r$ . ◀

Replacing an arc in a funnel by a directed path cannot create any cycles nor any forbidden subgraph for funnels. The next reduction rule reverses this operation: We can contract certain paths where all vertices have in- and outdegree one to a single arc. However, we cannot replace any such path: In the example in Figure 2, if we remove  $u$  and add the arc  $(v, w)$ , then the size of an optimal solution set decreases by one. Some cases where contracting an arc is safe are identified below.

► **Reduction Rule 3 (Dissolve Vertex) ( $\star$ ).** Let  $u, v, w$  be a path such that the following is true: (1)  $v, u \in \text{Dom}(\ell)$  implies  $\ell(v) = \ell(u)$ ; and (2)  $v, w \in \text{Dom}(\ell)$  implies  $\ell(v) = \ell(w)$ .

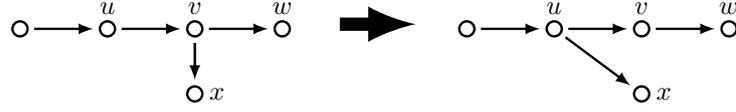
If  $\text{indeg}(v) = \text{outdeg}(v) = 1$  and  $(\text{indeg}(w) = 1 \vee \text{outdeg}(u) = 1)$ , delete the vertex  $v$  and add the arc  $(u, w)$ .

### 3.1 Bounding the number of unlabeled vertices

From Lower Bound (RR 1) we know there are few vertices with both in- and outdegree greater than one. In this section we bound the number of unlabeled vertices by considering the remaining unlabeled vertices, that is, vertices  $v$  with  $\text{indeg}(v) \leq 1$  or  $\text{outdeg}(v) \leq 1$ . Our strategy is to group such vertices into subgraphs of  $D$  with specific properties which we define later, and then develop reduction rules to both bound the maximum number of such subgraphs and also their size in any “yes” instance of FADL.

## 13:6 A Polynomial Kernel for Funnel Arc Deletion Set

■ **Figure 3** Example application of Shift Neighbors (RR 4).



Even if the previous reduction rules are not applicable, there can still exist some “large” subgraph  $H \subseteq D$  for which there is a “small” set  $S \subseteq A(D)$  such that the weakly-connected component of  $H$  is a funnel in  $D - S$ . Our goal is to bound the size of such subgraphs  $H$ .

We first define a specific type of subgraph of  $D$  which behaves like a funnel in the sense that the degrees of the vertices match Theorem 1(b). We call such subgraphs *local funnels* and formally define them below.

► **Definition 2.** An induced subgraph  $H \subseteq D$  is a local funnel in  $D$  if  $H$  is a funnel,  $H$  has only one source and its vertex set can be partitioned into  $F \uplus M = V(H)$  such that  $\text{indeg}_D(v) \leq 1$  for all  $v \in F$ ;  $\text{outdeg}_D(v) \leq 1$  for all  $v \in M$ ; and  $(M \times F) \cap A(H) = \emptyset$ .

Unlike *local funnels*, we might still have to remove many arcs from an *induced* funnel in  $D$ , as it can have, for example, several vertices  $v$  with  $\text{indeg}_D(v) > 1$  and  $\text{outdeg}_D(v) > 1$ . Our goal is to bound the size of each unlabeled local funnel (that is, each local funnel where none of the vertices have a label) and the number of unlabeled local funnels in  $D$ . We start by “pushing” as many vertices as we can to the neighborhood of the roots of the in- and out-forests of a local funnel. Consider for example a path  $u, v, w$  as in Figure 3, whose vertices have indegree one but can have higher outdegree. Intuitively, a cycle containing  $v$  and  $x$  must also contain  $u$ . To destroy this cycle, we can remove the unique incoming arc of  $u$ , as this will potentially destroy further cycles that contain  $u$  but not  $v$ . Hence, replacing the arc  $(v, x)$  with  $(u, x)$  in this case does not change the size of the solution.

By moving vertices in an out-tree towards its root  $s$ , we increase the outdegree of  $s$ . If the outdegree of  $s$  increases beyond  $k + 1$ , we can apply Set Label (RR 2) to  $s$ , giving it a label. By further applying Set Label (RR 2) to the neighbors of  $s$  which are in its out-tree, we can label the entire tree. As we are only considering unlabeled local funnels in this section, we can use the idea above to limit the branching of any in- or out-tree of an unlabeled local funnel.

We provide here a somewhat more general reduction rule which can also be applied if some vertices are labeled. Later, this reduction rule will again be useful to bound the number of labeled vertices. However, we need to carefully consider the possible labels of the vertices, as in some cases the rule would not be safe.

- **Reduction Rule 4 (Shift Neighbors).** Let  $u, v, w$  be a path.
- If  $\text{indeg}(u) = \text{indeg}(v) = \text{indeg}(w) = 1$ ,  $(u, M) \notin \ell$ ,  $(v, M) \notin \ell$  and there is an  $x \in \text{out}(v) \setminus \text{out}(u)$  with  $w \neq x$ , then remove the arc  $(v, x)$  and add the arc  $(u, x)$ .
  - If  $\text{outdeg}(u) = \text{outdeg}(v) = \text{outdeg}(w) = 1$ ,  $(v, F) \notin \ell$ ,  $(w, F) \notin \ell$  and there is an  $x \in \text{in}(v) \setminus \text{in}(w)$  with  $u \neq x$ , then remove the arc  $(x, v)$  and add the arc  $(x, w)$ .

Before proving that Shift Neighbors (RR 4) is safe, we need two simple observations about certain cases where we can safely exchange two arcs or add an arc.

► **Observation 3 (★).** Let  $H$  be a funnel with funnel labeling  $\ell$  and let  $x, u, v \in V(H)$  such that  $(v, x) \in A(H)$ ,  $(u, x) \notin A(H)$  and at least one of the following is true: (1)  $\ell(u) = F$ ; or (2)  $\ell(u) = M = \ell(v)$  and  $\text{outdeg}_H(u) = 0$ . Let  $H' = H - (v, x) + (u, x)$ . Then  $\ell$  is also a funnel labeling for  $H'$  if  $H'$  is a DAG.

► **Observation 4** ( $\star$ ). Let  $H$  be a DAG and  $x, u, v \in V(H)$  such that  $\{u\} = \text{in}(v)$ . Then  $H + (u, x)$  contains a cycle if and only if  $H + (v, x)$  contains a cycle.

**Proof of safety of Shift Neighbors (RR 4).** Consider the case where  $\text{indeg}(u) = \text{indeg}(v) = \text{indeg}(w) = 1$ ,  $(u, M) \notin \ell$ ,  $(v, M) \notin \ell$  and there is an  $x \in \text{out}(v) \setminus \text{out}(u)$  with  $w \neq x$ . The other case follows analogously. Let  $(D', \ell, k)$  be the reduced instance and  $(S_r, \hat{\ell}_r)$  be a solution for it. We construct a solution  $(S, \hat{\ell})$  for the input instance  $(D, \ell, k)$ .

First observe that, if  $(u, x) \in S_r$ , we can replace it with  $(v, x)$  in  $S$ , which means that  $D' - S_r$  and  $D - S$  are isomorphic. By setting  $\hat{\ell} := \hat{\ell}_r$ , we obtain the desired solution. If  $(u, x) \notin S_r$ , we consider the following cases.

*Case 1:*  $\hat{\ell}_r(v) = \text{F}$ . We set  $\hat{\ell} := \hat{\ell}_r$  and  $S := S_r$ . Let  $D^* = D - S$ . Clearly,  $D^* = D' - S_r - (u, x) + (v, x)$ . As  $u$  is the only inneighbor of  $v$ , from Observation 4 we know  $D^*$  is a DAG. From Observation 3, we know that  $\hat{\ell} = \hat{\ell}_r$  is a funnel labeling for  $D^*$ .

*Case 2:*  $\hat{\ell}_r(v) = \text{M} = \hat{\ell}_r(u)$ . If  $D' - S_r + (u, v)$  is a DAG, we can assume that  $(u, v) \notin S_r$ , implying  $(u, x) \in S_r$  (which was already considered).

If  $D' - S_r + (u, v)$  is not a DAG, then it contains a cycle with  $v$  and  $u$ , implying  $(u, v) \in S_r$ . In particular,  $\text{indeg}_{D' - S_r}(v) = 0$ . We set  $\hat{\ell} := \hat{\ell}_r$  and  $\hat{\ell}(v) := \text{F}$ . Clearly,  $\hat{\ell}$  is a funnel labeling for  $D' - S_r$ . From Observation 3 we have that  $\hat{\ell}$  is a funnel labeling for  $D - S_r$  as well.

*Case 3:*  $\hat{\ell}_r(v) = \text{M}$  and  $\hat{\ell}_r(u) = \text{F}$ . We set  $\hat{\ell} := \hat{\ell}_r$ ,  $\hat{\ell}(v) := \text{F}$  and  $S := S_r$ . As  $\{u\} = \text{in}_D(v)$  and  $\hat{\ell}_r(u) = \text{F}$ ,  $\hat{\ell}$  is a funnel labeling for  $D' - S_r$ . Let  $D^* = D - S$ .

From Observation 4 we know  $D^* = D' - S_r - (u, x) + (v, x)$  is a DAG since  $D' - S_r$  is a DAG. Hence, from Observation 3 we obtain that  $(S_r, \hat{\ell})$  is a solution for the input instance. In all cases a solution for the reduced instance implies a solution for the original instance.

Now assume there is a solution  $(S, \hat{\ell})$  for the original instance. We show that there is solution  $(S_r, \hat{\ell}_r)$  for the reduced instance. As in the previous direction, if  $(v, x) \in S$ , we can replace it with  $(u, x)$  and obtain the desired solution. So assume  $(v, x) \notin S$ .

If  $(u, v) \in S$ , let  $S_1 = S \cup \{(y, u)\}$ , where  $\{y\} = \text{in}(u)$ . Clearly,  $\hat{\ell}$  is a funnel labeling for  $D - S_1$ . We set  $\hat{\ell}_r := \hat{\ell}$  and  $\hat{\ell}_r(u) := \text{F}$ . As  $\text{indeg}_{D - S_1}(u) = 0$ ,  $\hat{\ell}_r$  is also a funnel labeling for  $D - S_1$ . From Observation 3 we know that  $\hat{\ell}_r$  is a funnel labeling for  $D_1 = D' - S_1$ . Since  $\text{indeg}_{D_1}(v) = 0 = \text{indeg}_{D_1}(u)$  and  $\hat{\ell}_r(u) = \text{F}$ , we have that  $\hat{\ell}_r$  is a funnel labeling for  $D_1 + (u, v)$ . Hence,  $(S \setminus \{(u, v)\}, \hat{\ell}_r)$  is a solution for the reduced instance.

In the following we consider the remaining cases where  $\{(u, v), (v, x)\} \cap S = \emptyset$ . Note that the case  $\hat{\ell}(u) = \text{M}$  and  $\hat{\ell}(v) = \text{F}$  does not happen under this assumption.

*Case 1:*  $\hat{\ell}(v) = \text{F} = \hat{\ell}(u)$ . We set  $\hat{\ell}_r := \hat{\ell}$  and  $S_r := S$ . Clearly,  $D' - S_r = D - S - (v, x) + (u, x)$ . From Observation 4 there is no cycle in  $D - S + (u, x)$  and, hence,  $D' - S_r$  is a DAG. Thus, from Observation 3 we have that  $\hat{\ell}_r$  is a funnel labeling for  $D' - S_r$ .

*Case 2:*  $\hat{\ell}(v) = \text{M} = \hat{\ell}(u)$ . Since  $(v, x) \notin S$ , we have  $(v, w) \in S$  and  $\hat{\ell}(x) = \text{M}$ . Further, we know that  $D' - S$  is a DAG due to Observation 4. Let  $S_1 = S \cup \{(u, v)\}$ . Clearly,  $\hat{\ell}$  is a funnel labeling for  $D - S_1$ , and  $D' - S_1$  is also a DAG. From Observation 3 we have that  $\hat{\ell}$  is a funnel labeling for  $D' - S_1$ .

We set  $\hat{\ell}_r := \hat{\ell}$  and  $\hat{\ell}_r(v) := \text{F}$ . Since  $\text{indeg}_{D' - S_1}(w) = 0 = \text{indeg}_{D' - S_1}(v)$ , we have that  $\hat{\ell}_r$  is a funnel labeling for  $D' - S_1 + (v, w)$ , regardless of the label of  $w$ . By setting  $S_r := (S \setminus \{(v, w)\}) \cup \{(u, v)\}$ , we get that  $\hat{\ell}_r$  is a funnel labeling for  $D' - S_r$  and  $|S_r| \leq |S|$ .

*Case 3:*  $\hat{\ell}(v) = \text{M}$  and  $\hat{\ell}(u) = \text{F}$ . Let  $S_r = S$  and  $\hat{\ell}_r = \hat{\ell}$ . Since  $(u, v) \notin S_r$ , from Observation 4 we know that  $D - S_r - (v, x) + (u, x)$  is a DAG. From Observation 3 we have that  $\hat{\ell}_r$  is a funnel labeling for  $D' - S_r$ .

In all cases we found a solution  $(S_r, \hat{\ell}_r)$  for the reduced instance, concluding the proof. ◀

## 13:8 A Polynomial Kernel for Funnel Arc Deletion Set

It is not always possible to exhaustively apply Shift Neighbors (RR 4): If  $u, v, w$  forms a cycle, we would shift  $x$  indefinitely through this cycle. To prevent this from happening, we need the following reduction rule:

► **Reduction Rule 5 (Break Cycle).** *Let  $C$  be a cycle in  $D$ . If every vertex in  $C$  has indegree (outdegree) one and either every vertex in  $C$  is unlabeled or every vertex in  $C$  is labeled with  $F$  ( $M$ ), then delete one arc of  $C$  and decrease  $k$  by one.*

**Proof of safety of Break Cycle (RR 5).** Let  $(v, u)$  be the arc removed by the reduction rule. Clearly, a solution for the reduced instance together with the arc  $(v, u)$  is a solution for the original instance. Let  $(S, \hat{\ell})$  be a solution for the original instance, and assume that  $(v, u) \notin S$ . Let  $(w, x)$  be an arc of  $C$  contained in  $S$ . Without loss of generality, we assume that  $(w, x)$  is the only incoming arc of  $x$ . The case where it is the only outgoing arc of  $w$  follows analogously.

We can assume that  $\hat{\ell}(v) = F$  for all  $v \in V(C)$ : If they were not labeled by  $\ell$  when the rule was applied, then by repeatedly applying Set Label (RR 2) (starting with  $x$ ) we can label them with  $F$ . Because  $\text{indeg}_D(v) = 1$  for every  $v \in C$ , it follows that  $C$  is the only cycle in  $D$  using the arc  $(w, x)$ . Hence,  $D' = D - S + (w, x) - (v, u)$  is a DAG. Further, as  $\hat{\ell}(w) = \hat{\ell}(x) = F$ , it is easy to see that  $\hat{\ell}$  is a funnel labeling for  $D'$ . ◀

If Shift Neighbors (RR 4) is not applicable, then many vertices in a long path  $P$  in a local funnel must share a common out- or inneighbor  $w$ . However, from Set Label (RR 2) we know that  $w$  receives a label if it has too many neighbors. The next and final reduction rule needed for bounding the number of unlabeled vertices exploits this property and allows us to label some vertex  $u$  in  $P$  if its predecessor  $v$  in  $P$  is adjacent to a labeled vertex  $w$ .

► **Reduction Rule 6 (Labeled Neighbor).** *Let  $(v, u)$  be an arc between unlabeled vertices. Set  $\ell(u) := F$  if  $\text{indeg}(u) = \text{indeg}(v) = 1$  and  $\exists w \in \text{out}(v) : \ell(w) = M$ . Set  $\ell(v) := M$  if  $\text{outdeg}(u) = \text{outdeg}(v) = 1$  and  $\exists w \in \text{in}(u) : \ell(w) = F$ .*

**Proof of safety of Labeled Neighbor (RR 6).** Assume, without loss of generality, that the first case of the rule was applied. The proof for the second case follows analogously (note that it is not possible for both cases to be applied simultaneously). Let  $(D, \ell_r, k)$  be the reduced instance. First note that  $\ell_r \supseteq \ell$ , which means that a solution for the reduced instance is already a solution for the original instance. Hence, it suffices to show that a solution  $(S, \hat{\ell})$  for the original instance implies a solution  $(S_r, \hat{\ell}_r)$  for the reduced instance.

If  $\hat{\ell}(u) = F$ , we set  $\hat{\ell}_r := \hat{\ell}$  and  $S_r := S$  and we are done. So assume that  $\hat{\ell}(u) = M$ .

*Case 1:*  $(v, u) \in S$ . We set  $S_r := S$ ,  $\hat{\ell}_r := \hat{\ell}$  and  $\hat{\ell}_r(u) := F$ . As  $\text{indeg}_{D-S}(u) = 0$ , we know that  $\hat{\ell}_r$  is also a funnel labeling for  $D - S$ .

*Case 2:*  $(v, u) \notin S$  and  $\hat{\ell}(v) = F$ . We set  $S_r := S$ ,  $\hat{\ell}_r := \hat{\ell}$  and  $\hat{\ell}_r(u) := F$ . As  $\hat{\ell}_r(v) = F = \hat{\ell}_r(u)$ , we may keep the arc  $(v, u)$  and  $\hat{\ell}_r$  is a funnel labeling for  $D - S_r$ .

*Case 3:*  $(v, u) \notin S$  and  $\hat{\ell}(v) = M$ . Then  $(v, w) \in S$ . We set  $\hat{\ell}_r := \hat{\ell}$ ,  $\hat{\ell}_r(u) := F$ ,  $\hat{\ell}_r(v) := F$ ,  $S_r := (S \setminus \{(v, w)\}) \cup \{(y, v)\}$ , where  $y$  is the unique inneighbor of  $v$ .

The digraph  $D - S_r$  is a DAG: if it has a cycle, the cycle would have to use the arc  $(v, w)$ , yet  $\text{indeg}_{D-S_r}(v) = 0$ , a contradiction. We now argue that  $\hat{\ell}_r$  is a funnel labeling for  $D - S_r$ . Since  $\text{indeg}_{D-S_r}(v) = 0$ ,  $\text{indeg}_{D-S_r}(u) = 1$  and  $\hat{\ell}_r(u) = F$ , the vertex  $v$  is the unique inneighbor of  $u$  in the out-forest of the funnel  $D - S_r$ . Finally, as  $\hat{\ell}_r(w) = M$ , the arc  $(v, w)$  is allowed in the funnel. Hence,  $\hat{\ell}_r$  is a funnel labeling for  $D - S_r$ . In all cases we find a solution  $(\hat{\ell}_r, S_r)$  for the reduced instance, concluding the proof. ◀

► **Lemma 5.** *Let  $s$  be some source (sink) of some unlabeled local funnel  $H$  in the reduced digraph  $D$ . Let  $P_1, P_2, \dots, P_a$  be a sequence of paths in  $H$  starting (ending) at  $s$  such that  $\text{indeg}(u) \leq 1$  ( $\text{outdeg}(u) \leq 1$ ) for each  $u$  in each  $P_i$ , and  $V(P_j) \not\subseteq V(P_i)$  for all  $1 \leq i, j \leq a$  where  $i \neq j$ . Let  $E$  be the set of end (start) points of all  $P_i$ . Then all of the following hold.*

(1)  $\text{outdeg}(u) > 1$  ( $\text{indeg}(u) > 1$ ) for any inner vertex  $u$  of any  $P_i$ .

(2)  $\text{out}(\bigcup_{i=1}^a V(P_i) \setminus E) \subseteq \text{out}(s)$  ( $\text{in}(\bigcup_{i=1}^a V(P_i) \setminus E) \subseteq \text{in}(s)$ ),

(3)  $V(P_i) \cap V(P_j) = \{s\}$  for each  $1 \leq i, j \leq a$  where  $i \neq j$ , and

(4)  $a \leq k + 1$  and  $|V(P_i)| \leq k + 2$  for each  $1 \leq i \leq a$ .

**Proof.** We consider the case where  $s$  is a source of  $H$ . The other case follows analogously.

Let  $u$  be some inner vertex of some  $P_i$  and  $w$  the unique outneighbor of  $u$  in  $P_i$ . By assumption on  $P_i$ , we have  $\text{indeg}_D(w) = 1$ . As Dissolve Vertex (RR 3) is not applicable, we have that  $\text{outdeg}(u) > 1$  (proving (1)). In particular,  $u$  has some outneighbor  $x$  not in  $P_i$ .

Let  $v$  be the inneighbor of  $u$  in  $P_i$ . Since  $\text{indeg}_D(v) = \text{indeg}_D(u) = \text{indeg}_D(w) = 1$  and Shift Neighbors (RR 4) is not applicable, we have  $x \in \text{out}_D(v)$ . By repeating this argument to the predecessors of  $u$  in  $P_i$ , we prove (2) (and also that  $a \leq k + 1$ , as  $\text{outdeg}_D(s) \leq k + 1$  due to Set Label (RR 2)).

Assume there are two paths  $P_i$  and  $P_j$  intersecting at more than one vertex. Let  $u$  be the last vertex of the intersection. Note that, if  $u$  is the last vertex of  $P_i$  or  $P_j$ , then one path has to contain the other. Hence,  $u$  has two outneighbors  $w_i$  and  $w_j$  lying on  $P_i$  and  $P_j$ , respectively, and  $w_i \neq w_j$ . But due to (2), we have  $w_i, w_j \in \text{out}_D(s)$ , implying  $\text{indeg}_D(w_i) > 1$  and  $\text{indeg}_D(w_j) > 1$ , a contradiction to our assumptions on  $P_i$  and  $P_j$  (proving (3)).

Let  $v_1, v_2, \dots, v_m$  be the sequence of vertices of a path  $P_i$ . From (1) we know that there is some  $w \in \text{out}_D(v_{m-1})$  outside of  $P_i$ . We also have  $w \in \text{out}_D(v_j)$  for all  $1 \leq j \leq m - 1$ , implying  $\text{indeg}_D(w) \geq m - 1$ . If  $m - 1 > k + 1$ , then  $\ell(w) = M$ , as Set Label (RR 2) is not applicable. However, as  $\text{indeg}_D(v_{m-1}) = 1 = \text{indeg}_D(v_{m-2})$ ,  $w \in \text{out}_D(v_{m-2})$  and Labeled Neighbor (RR 6) is not applicable, we have  $\ell(v_{m-1}) = F$ , a contradiction to the assumption that  $H$  is unlabeled. Hence,  $m - 1 \leq k + 1$ , implying  $|V(P_i)| \leq k + 2$  (proving (4)). ◀

► **Lemma 6** ( $\star$ ). *Let  $H$  be an unlabeled local funnel in  $D$ . Then  $|V(H)| \in \mathcal{O}(k^3)$ .*

We conclude by bounding the number of maximal vertex-disjoint unlabeled local funnels in  $D$ . Since we can always partition unlabeled vertices with in- or outdegree at most one into local funnels, by bounding the number of local funnels in such a partitioning, together with the bound on the size of each local funnel, we obtain a bound for the number of unlabeled vertices with in- or outdegree at most one.

Let  $\mathcal{H} = \{H_1, H_2, \dots, H_a\}$  be a set of maximal vertex-disjoint unlabeled local funnels in  $D$  (in this context, maximal means that  $H_i \cup H_j$  is not a local funnel for any two distinct  $H_i, H_j \in \mathcal{H}$ ). Let  $s_i$  be the unique source of  $H_i$  for each  $i$ . We now show that, if there is a solution removing at most  $k$  arcs, then  $|\mathcal{H}|$  is “small”. By contraposition this means that, if  $|\mathcal{H}|$  is “large”, then we have a “no” instance and can stop the kernelization process.

We start with the simple observation that cycles intersecting inside a local funnel must also intersect outside it.

► **Observation 7** ( $\star$ ). Let  $C_i$  and  $C_j$  be two distinct cycles in  $D$  such that  $V(C_i) \cap V(C_j) \subseteq H_\ell$  for some  $H_\ell \in \mathcal{H}$ . Then  $V(C_i) \cap V(C_j) = \emptyset$ .

We partition the set of maximal unlabeled local funnels  $\mathcal{H}$  into three sets (1)  $\mathcal{F} = \{H_i \in \mathcal{H} \mid \text{there is some } v \in V(H_i) \text{ with } \text{outdeg}_D(v) > 1\}$ ; (2)  $\mathcal{M} = \{H_i \in \mathcal{H} \mid \text{indeg}_D(s_i) > 1\}$ ; and (3)  $\mathcal{X} = \{H_i \in \mathcal{H} \mid \text{indeg}_D(s_i) = 1 \text{ and } \forall v \in V(H_i) : \text{outdeg}_D(v) = 1\}$ .

## 13:10 A Polynomial Kernel for Funnel Arc Deletion Set

► **Lemma 8.** *If there is a solution  $(S, \hat{\ell})$  for  $(D, \ell, k)$ , then  $|\mathcal{X}| \leq 2k^2$ .*

**Proof.** Let  $H_i \in \mathcal{X}$  and  $u$  be the unique inneighbor of  $s_i$ . Note that  $\text{outdeg}_D(s_i) = 1$ . As Dissolve Vertex (RR 3) is not applicable, we have that  $\text{outdeg}_D(u) > 1$  and  $\text{indeg}_D(w) > 1$ , where  $w$  is the unique outneighbor of  $s_i$ .

*Case 1:*  $u \in \text{Dom}(\ell)$ . Then  $\ell(u) = M$  since Set Label (RR 2) is not applicable. As  $\text{outdeg}(u) > 1$ , each  $H_j \in \mathcal{X}$  with  $s_j \in \text{out}_D(u)$  requires one more arc of  $u$  to be in  $S$ .

*Case 2:*  $u \notin \text{Dom}(\ell)$ . If  $\text{indeg}_D(u) = 1$ , then there is some  $v_i \in V(H_i)$  such that  $(v_i, u) \in A(D)$ , otherwise  $H_i$  would not be maximal. Hence, there is a cycle  $C_i$  containing  $u, s_i$  and  $v_i$ . If there is any other  $H_j \in \mathcal{X}$  with  $s_j \in \text{out}_D(u)$  and with some  $v_j \in V(H_j)$  such that  $(v_j, u)$ , then the cycle  $C_j$  containing  $u, s_j$  and  $v_j$  is arc-disjoint to the cycle  $C_i$  due to Observation 7. Thus,  $S$  must contain at least one arc of each such  $C_j$ , implying there are at most  $k$  local funnels  $H_j$  that fall into this case.

If  $\text{indeg}_D(u) > 1$ , one arc of  $u$  is in  $S$  as  $\text{outdeg}_D(u) > 1$ . Further,  $\text{outdeg}_D(u) \leq k$ . This means that there are at most  $k$  local funnels  $H_j \in \mathcal{X}$  with  $s_j \in \text{out}_D(u)$ . As there can be at most  $2k$  such vertices  $u$ , we have that there are at most  $2k^2$  local funnels  $H_j \in \mathcal{X}$  which fall into this case. In the worst case, we have  $|\mathcal{X}| \leq \max\{k + 1, 2k^2\} \leq 2k^2$ . ◀

► **Lemma 9** ( $\star$ ). *If there is a solution  $(S, \hat{\ell})$  for  $(D, \ell, k)$ , then  $|\mathcal{F}| \leq 2k^2 + 3k$ .*

► **Lemma 10** ( $\star$ ). *If there is a solution  $(S, \hat{\ell})$  for  $(D, \ell, k)$ , then  $|\mathcal{M}| \leq k^2 + 2k$ .*

From Lemmas 8 to 10, we easily obtain a bound for the number of vertices in unlabeled local funnels. Together with the fact that Lower Bound (RR 1) is not applicable, we obtain a bound for the number of unlabeled vertices in  $D$ .

► **Lemma 11** ( $\star$ ). *Let  $D$  be a reduced digraph. Then there are  $\mathcal{O}(k^5)$  vertices  $v \in V(D)$  with  $v \notin \text{Dom}(\ell)$  and  $\text{indeg}(v) = 1 \vee \text{outdeg}(v) = 1$ .*

### 3.2 Bounding the number of labeled vertices

In Section 3.1 we exploited the property that unlabeled vertices have bounded degree, and that we can label them if their neighborhood has some special structure captured by the reduction rules. For the labeled vertices, however, we can apply neither of those strategies. Instead, we first exploit the fact that we know the label of a vertex and use this to decide if an arc is never in an optimal solution or if it is always in an optimal solution.

Arcs from M to F vertices clearly need to be removed. We show that we can also ignore arcs from F to M vertices, that is, we can remove them without changing  $k$ .

► **Reduction Rule 7 (Remove Arcs)** ( $\star$ ). *Let  $(v, u) \in A(D)$ . If  $\ell(v) = F$  and  $\ell(u) = M$ , remove  $(v, u)$ . If  $\ell(v) = M$  and  $\ell(u) = F$ , remove  $(v, u)$  and decrease  $k$  by 1.*

We now identify certain vertices that can be removed safely. Clearly, sources and sinks cannot be in any cycle in  $D$ . By carefully considering the neighborhood of a source or sink  $v$ , we can also prove that  $v$  is not “relevant” for any forbidden subgraph for funnels in  $D$ .

► **Reduction Rule 8 (Sources and Sinks)** ( $\star$ ). *Let  $v \in V(D)$  be a labeled vertex where  $\text{out}(v) \cup \text{in}(v) \subseteq \text{Dom}(\ell)$ . Remove  $v$  if one of the following holds.*

1.  $\text{indeg}(v) = 0$  and no  $u \in \text{out}(v)$  exists with  $\ell(u) = F$  and  $\text{indeg}(u) > 1$ , or
2.  $\text{outdeg}(v) = 0$  and no  $u \in \text{in}(v)$  exists with  $\ell(u) = M$  and  $\text{outdeg}(u) > 1$ .

Having exhaustively applied Reduction Rules 7 and 8, we can bound the number of labeled vertices in  $D$ . Since Lower Bound (RR 1) is not applicable, we already have a bound for the number of vertices  $v$  with  $\ell(v) = F \wedge \text{indeg}(v) > 1$  or  $\ell(v) = M \wedge \text{outdeg}(v) > 1$ . Hence, we only need to consider vertices in the set  $L = \{v \in \text{Dom}(\ell) \mid \ell(v) = F \wedge \text{indeg}(v) \leq 1 \text{ or } \ell(v) = M \wedge \text{outdeg}(v) \leq 1\}$ .

To bound  $|L|$ , we exploit the bound on the number of unlabeled vertices from Lemma 11 and also the fact that such vertices have small degree as Set Label (RR 2) is not applicable. We first partition  $L$  into two subsets  $L_1 = \{v \in L \mid \text{in}(v) \cup \text{out}(v) \not\subseteq \text{Dom}(\ell)\}$  and  $L_2 = L \setminus L_1$ .

► **Lemma 12** ( $\star$ ).  $|L_1| \in \mathcal{O}(k^6)$ .

► **Lemma 13**.  $|L_2| \in \mathcal{O}(k)$ .

**Proof.** Let  $V_F = \{v \mid \ell(v) = F\}$  and  $L_F = V_F \cap L_2$ . The case for the vertices labeled with M follows analogously.

Since Remove Arcs (RR 7) is not applicable, we have  $\ell(u) = F$  for all  $u \in \text{out}(L_F) \cup \text{in}(L_F)$ . Let

$R_1 = \{u \in V_F \mid \text{indeg}(u) > 1\}$ ,  $R_2 = \{u \in L_F \mid \text{indeg}(u) \leq 1, \text{out}(u) \cap R_1 \neq \emptyset\}$  and  $R_3 = \{u \in L_F \mid \text{indeg}(u) \leq 1, \text{out}(u) \cap R_1 = \emptyset\}$ .

Note that  $L_2 = R_2 \cup R_3$  and  $R_1 \cap L_2 = \emptyset$ .

A solution set  $S \subseteq A(D)$  must contain at least  $\text{indeg}(v) - 1$  many incoming arcs of  $v$  for every  $v \in R_1$ . As each  $u \in R_2$  has some  $v \in R_1$  as outneighbor, we have  $|R_2| \leq 2k$ .

Let  $v \in R_3$ . We claim that  $v$  can reach some vertex of  $R_2$ . Since Sources and Sinks (RR 8) is not applicable and  $\text{out}(v) \cap R_1 = \emptyset$ , we have  $\text{indeg}(v) = 1$  and  $\text{outdeg}(v) \geq 1$ . This means that, if we successively follow the outneighbors of  $v$ , we reach a vertex of  $R_2$  or find a cycle  $C$  using only vertices of  $R_3$ . However, as Break Cycle (RR 5) is not applicable, such a cycle  $C$  cannot exist: every vertex  $v \in R_3$  has  $\text{indeg}(v) = 1$  and  $\ell(v) = F$ , implying we could apply Break Cycle (RR 5) to  $C$ . Hence, every vertex of  $R_3$  can reach some  $u \in R_2$ .

We greedily construct vertex-disjoint paths  $P_1, P_2, \dots, P_a$  ending in  $R_2$  whose inner vertices lie in  $R_3$ . For a vertex  $v \in R_3$  take an arbitrary  $u \in R_2$  such that  $v$  can reach  $u$ . Consider a path  $P$  from  $v$  to  $u$ . If none of its vertices lie in any already constructed  $P_i$ , we just take the path  $P$  into our set of paths. Otherwise, assume that  $P$  intersects some  $P_i$  at  $w$  and let  $w$  be the first such vertex in  $P$ . Since the indegree of any vertex in  $R_3 \cup R_2$  is at most one, we know that  $w$  is the starting point of  $P_i$ . Hence, we can obtain a path  $P_j$  by taking the path from  $v$  to  $w$  in  $P$  and then concatenating  $P_i$ . As  $w$  is the first vertex of  $P$  intersecting any other path, we get that  $P_j$  only intersects  $P_i$ . By replacing  $P_i$  with  $P_j$ , we obtain a path that also contains  $v$ . We repeat this process until we covered all  $v \in R_3$ .

Since  $|R_2| \leq 2k$ , we have  $a \leq 2k$ . We now prove that  $|V(P_i)| \leq 4$  for any  $P_i$  in our set of vertex-disjoint paths. Note that  $\text{indeg}(u) \leq 1$  for any vertex  $u \in V(P_i)$ .

Since Dissolve Vertex (RR 3) is not applicable, any inner vertex  $u$  of  $P_i$  has  $\text{outdeg}(u) > 1$ . Let  $w$  be the successor of  $u$  in  $P_i$ . As Shift Neighbors (RR 4) is not applicable, we have that  $\text{indeg}(w) > 1$  or  $v \in \text{out}(u)$  where  $v$  is the unique inneighbor of  $u$ . If  $\text{indeg}(w) > 1$ , then  $u \in R_2$  and is the endpoint of  $P_i$ , a contradiction to the assumption that  $u$  is an inner vertex of  $P_i$ . Otherwise, we know that  $u \notin \text{out}(w)$  as  $\text{indeg}(u) = 1$ . If  $u$  is the only inner vertex of  $P_i$ , then  $|V(P_i)| \leq 3$ . Otherwise, its successor  $w$  in  $P_i$  is an inner vertex of  $P_i$  (since  $v$  is the starting point of  $P_i$ , and so  $v \notin \text{out}(u)$ ). Hence, we can apply the same argumentation to  $w$  and conclude that it has some outneighbor  $x$  with  $\text{indeg}(x) > 1$ , implying  $x \in R_2$  and  $|V(P_i)| \leq 4$ .

Since  $|V(P_i)| \leq 4$  and  $a \leq 2k$ , we have that  $|L_2| \leq 6k$ . Because  $L_2 = R_2 \cup R_3$ , we have that  $|L_2| \leq 8k \in \mathcal{O}(k)$ , as desired. ◀

► **Lemma 14**. Let  $(D, \ell, k)$  be an FADL instance where Reduction Rules 1 to 8 are not applicable. Then  $|V(D)| \in \mathcal{O}(k^6)$  and  $|A(D)| \in \mathcal{O}(k^6)$ .

## 13:12 A Polynomial Kernel for Funnel Arc Deletion Set

**Proof.** As Lower Bound (RR 1) is not applicable, there are at most  $2k$  vertices  $v$  with  $\text{indeg}(v) > 1$  and  $\text{outdeg}(v) > 1$ , and also at most  $2k$  many vertices  $v$  with  $\ell(v) = F \wedge \text{indeg}(v) > 1$  or  $\ell(v) = M \wedge \text{outdeg}(v) > 1$ . From Lemma 11 we know there are  $\mathcal{O}(k^5)$  many unlabeled vertices  $v \in V(D)$  with  $\text{indeg}(v) \leq 1$  or  $\text{outdeg}(v) \leq 1$ . Finally, due to Lemmas 12 and 13 there are  $\mathcal{O}(k^6)$  vertices  $v$  with  $\ell(v) = F \wedge \text{indeg}(v) \leq 1$  or  $\ell(v) = M \wedge \text{outdeg}(v) \leq 1$ . As any vertex in  $D$  falls into one of these groups, we have  $|V(D)| \in \mathcal{O}(k^6)$ .

As Remove Arcs (RR 7) is not applicable, there is no arc  $(v, u)$  where  $v, u \in \text{Dom}(\ell)$  and  $\ell(v) \neq \ell(u)$ . Since there are  $\mathcal{O}(k^5)$  many unlabeled vertices and every unlabeled vertex has in- and outdegree at most  $k + 1$ , there are  $\mathcal{O}(k^6)$  arcs  $(v, u)$  where  $v \notin \text{Dom}(\ell)$  or  $u \notin \text{Dom}(\ell)$ .

Now let  $(v, u)$  be some arc where  $v, u \in \text{Dom}(\ell)$ . Note that  $\ell(v) = \ell(u)$ .

*Case 1:*  $v, u \in L$ . Then  $\text{outdeg}(v) = 1$  (if  $\ell(v) = M$ ) or  $\text{indeg}(u) = 1$  (if  $\ell(u) = F$ ). Thus, there can be at most  $|L| \in \mathcal{O}(k^6)$  many arcs  $(v, u)$  where  $v, u \in L$ .

*Case 2:*  $v, u \notin L$ . As Lower Bound (RR 1) is not applicable, there can be at most  $2k$  such vertices. Thus, there are at most  $4k^2$  arcs between labeled vertices not in  $L$ .

*Case 3:* Exactly one of  $v, u$  is in  $L$ .

*Case 3.1:*  $v \notin L \wedge \ell(v) = F$  or  $u \notin L \wedge \ell(u) = M$ . Then  $\text{indeg}(u) = 1$  or  $\text{outdeg}(v) = 1$ . Hence, there can be at most  $|L| \in \mathcal{O}(k^6)$  such arcs.

*Case 3.2:*  $v \notin L \wedge \ell(v) = M$  or  $u \notin L \wedge \ell(u) = F$ . If  $v \notin L$ , then at least half of its outgoing arcs need to be in a solution set. Similarly, if  $u \notin L$ , at least half of its incoming arcs need to be in a solution set. Hence, there can be at most  $2k$  many arcs falling into this case. By adding all cases together, we obtain that  $|A(D)| \in \mathcal{O}(k^6)$ , concluding the proof.  $\blacktriangleleft$

### 4 Computing the Kernel

In Sections 3.1 and 3.2 we defined the reduction rules for the kernelization process and showed that, if none of the reduction rules are applicable to a digraph  $D$ , then the size of  $D$  is polynomially bounded on  $k$ . To conclude the proof that FADS admits a polynomial problem kernel, we show that it is possible to apply all reduction rules in  $\mathcal{O}(nm)$  time and also reduce the FADL instance back into an FADS instance.

► **Lemma 15** ( $\star$ ). *We can exhaustively apply Reduction Rules 1 to 8 in  $\mathcal{O}(nm)$  time to an FADL instance  $(D, \ell, k)$ , where  $n = |V(D)|$  and  $m = |A(D)|$ .*

► **Theorem 16.** *FADS admits a kernel with  $\mathcal{O}(k^6)$  vertices and  $\mathcal{O}(k^7)$  arcs which can be computed in  $\mathcal{O}(nm)$  time, where  $n = |V(D)|$ ,  $m = |A(D)|$  and  $D$  is the input digraph.*

**Proof.** We start by reducing the FADS instance into an FADL instance  $(D, \ell, k)$  by adding an empty labeling  $\ell$ . Using Lemma 15, we can exhaustively apply all reduction rules to  $(D, \ell, k)$  in  $\mathcal{O}(nm)$  time.

From Lemma 14 we know  $|V(D)| \in \mathcal{O}(k^6)$  and  $|A(D)| \in \mathcal{O}(k^6)$ . We now reduce the FADL instance back into an FADS instance  $(D', k)$  in order to obtain a kernel for the original problem. We first set  $D' := D$  and add  $k + 2$  vertices  $f_1, f_2, \dots, f_{k+2}$  and  $k + 2$  vertices  $m_1, m_2, \dots, m_{k+2}$  to  $D'$ . Let  $v \in \text{Dom}(\ell)$ . If  $\ell(v) = F$ , we add the arc  $(v, f_i)$  for each  $1 \leq i \leq k + 2$ . If  $\ell(v) = M$ , we add the arc  $(m_i, v)$  for each  $1 \leq i \leq k + 2$ .

Trivially, a solution for the FADL instance is also a solution for the FADS instance. It is also easy to see that, if there is some arc set  $S_r \subseteq A(D')$  and some funnel labeling  $\hat{\ell}_r$  for  $D' - S_r$  such that  $\ell(v) \neq \hat{\ell}_r(v)$  for some  $v \in \text{Dom}(\ell)$ , then  $|S_r| > k$ . Hence, a solution for  $(D', k)$  implies a solution for  $(D, \ell, k)$ .

We added  $2k + 4$  vertices and  $\mathcal{O}(k^7)$  many arcs to  $D'$ , and so  $|V(D')| \in \mathcal{O}(k^6)$  and  $|A(D')| \in \mathcal{O}(k^7)$ , thus concluding the proof.  $\blacktriangleleft$

## 5 Conclusion

The kernelization algorithm provided in this paper heavily relies on the characterizations of Theorem 1 for funnels. Both the characterization by forbidden subgraphs as well as the labeling characterization allowed us to derive reduction rules based only on “local” substructures as the degree or neighborhood of a vertex. In a sense, this “locality” property saved us from computing any set of vertex-disjoint local funnels, despite the fact that the results and reduction rules from Section 3.1 heavily rely on local funnels.

The polynomial kernels for OUT-Forest-VDS and PUMPKIN-VDS due to [12] also rely on “localized” forbidden substructures. We consider that generalizing these results to larger digraph classes of unbounded treewidth, but which are characterized by forbidden substructures, to be a very interesting direction for future research.

Further, it would also be interesting to decide if FUNNEL-VDS admits a polynomial kernel or not (it is in FPT with respect to the solution size [11]), especially since a kernel for this problem would require considerably different ideas from the ones presented in this paper, as it is no longer clear how to exploit the vertex labeling in the vertex-deletion setting.

---

## References

- 1 Faisal N Abu-Khzam. A kernelization algorithm for  $d$ -hitting set. *Journal of Computer and System Sciences*, 76(7):524–531, 2010.
- 2 Akanksha Agrawal, Saket Saurabh, Roohani Sharma, and Meirav Zehavi. Kernels for deletion to classes of acyclic digraphs. *Journal of Computer and System Sciences*, 92:9–21, 2018.
- 3 Jørgen Bang-Jensen, Alessandro Maddaloni, and Saket Saurabh. Algorithms and kernels for feedback set problems in generalizations of tournaments. *Algorithmica*, 76(2):320–343, 2016.
- 4 Jianer Chen, Yang Liu, Songjian Lu, Barry O’sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *Journal of the ACM (JACM)*, 55(5):21, 2008.
- 5 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 4. Springer, 2015.
- 6 Rodney G Downey and Michael R Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.
- 7 Fedor V Fomin, Daniel Lokshantov, Saket Saurabh, and Meirav Zehavi. *Kernelization: theory of parameterized preprocessing*. Cambridge University Press, 2019.
- 8 Stefan Kratsch. Recent developments in kernelization: A survey. *Bulletin of EATCS*, 2(113), 2014.
- 9 Daniel Lokshantov, Neeldhara Misra, and Saket Saurabh. Kernelization–preprocessing with a guarantee. In *The Multivariate Algorithmic Revolution and Beyond*, pages 129–161. Springer, 2012.
- 10 Daniel Lokshantov, MS Ramanujan, Saket Saurabh, Roohani Sharma, and Meirav Zehavi. Wannabe bounded treewidth graphs admit a polynomial kernel for dfvs. In *Workshop on Algorithms and Data Structures*, pages 523–537. Springer, 2019.
- 11 Marcelo Garlet Millani, Hendrik Molter, Rolf Niedermeier, and Manuel Sorge. Efficient algorithms for measuring the funnel-likeness of DAGs. In Jon Lee, Giovanni Rinaldi, and A. Ridha Mahjoub, editors, *Combinatorial Optimization*, pages 183–195, Cham, 2018. Springer International Publishing.
- 12 Matthias Mnich and Erik Jan van Leeuwen. Polynomial kernels for deletion to classes of acyclic digraphs. *Discrete Optimization*, 25:48–76, 2017.