# Supplementary Material for "Variable-Deletion Backdoors to Planning" to appear at AAAI-2015

**Martin Kronegger, Sebastian Ordyniak, Andreas Pfandler**

In this document, we use the following notation: For $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, \ldots, n\}$.

## Additional Notions and Definitions

In this section we introduce some additional notions and definitions that are not required for the main text but only for some of the proofs in the appendix

Let $\mathbb{P}$ be a SAS$^+$ PLANNING instance and $S$ be a $c$-Extended Causal Backdoor of $\mathbb{P}$. Let $C_1$ and $C_2$ be two components in $\mathsf{C}(\mathbb{P}, S)$ that are equivalent using the isomorphism $\varphi$ from $\mathbb{P}(V(C_1) \cup S)$ to $\mathbb{P}(V(C_2) \cup S)$, $s_1 \in \mathsf{ST}(\mathbb{P}(V(C_1)))$, $s_2 \in \mathsf{ST}(\mathbb{P}(V(C_2)))$, and $a \in A$ with at least one precondition or effect on $C_1$, then we say an action $a' \in A$ corresponds to $a$ in $C_2$ if $a' = \varphi(a)$ and we write $s_1 = s_2$ if $s_1[v_1] = \varphi(s_2[\varphi(v_1)])$ for every $v_1 \in V(C_1)$.

For the proof of Theorem 5 we need to formally define the problem PARTITIONED CLIQUE. Given a $k$-partite graph $G = (V, E)$ with partition $\{P_1, \ldots, P_k\}$ of $V$ into sets of equal size, we want to find a $k$-clique, i.e., a set $V' \subseteq V$ of $k$ vertices such that $\forall u, v \in V'$, with $u \neq v$ there is an edge $\{u, v\} \in E$, and $\forall i \in [k]$ it holds that $|V' \cap P_i| = 1$. The problem is known to be $\mathsf{W}[1]$-complete when parameterized by $k$ (Pietrzak 2003).

A *polynomial parameter transformation* (PPT) from a parameterized problem $P$ to a parameterized problem $Q$ is a parameterized reduction from $P$ to $Q$ that maps instances $\langle \mathbb{I}, k \rangle$ of $P$ to instances $\langle \mathbb{I}', k' \rangle$ of $Q$ with the additional property that

1. $\langle \mathbb{I}', k' \rangle$ can be computed in time that is polynomial in $|\mathbb{I}| + k$, and
2. $k'$ is bounded by some polynomial $p$ of $k$.

**Proposition 17** ((Bäckström et al. 2013a, Proposition 4)). *Let $P$ and $Q$ be two parameterized problems such that there is a polynomial parameter reduction from $P$ to $Q$. Then, if $Q$ has a polynomial bi-kernel also $P$ has a polynomial bi-kernel.*

A *strong OR-composition algorithm* for a parameterized problem $P$ maps $t$ instances $\langle \mathbb{I}_1, k_1 \rangle, \ldots, \langle \mathbb{I}_t, k_t \rangle$ of $P$ to one instance $\langle \mathbb{I}, k \rangle$ of $P$ such that the algorithm runs in time polynomial in $\sum_{1 \leq i \leq t} |\mathbb{I}_i| + \max_{1 \leq i \leq t} k_i$, the parameter $k$ is bounded by a polynomial in $\max_{1 \leq i \leq t} k_i$, and $\langle \mathbb{I}, k \rangle \in P$ if and only if there is an $i$, where $1 \leq i \leq t$, such that $\langle \mathbb{I}_i, k_i \rangle \in P$.

**Proposition 18** ((Bäckström et al. 2013a, Proposition 4)). *If a parameterized problem $P$ has a strong OR-composition algorithm and its unparameterized version is* $\mathsf{NP}$*-hard, then it has no polynomial bi-kernel unless* $\mathsf{coNP} \subseteq \mathsf{NP}/\text{poly}$.

## Proof of Claim 1

*Claim* 1 (*). There is an equivalent instance $\langle \mathbb{P}'', k' \rangle$ with at most $k''$ variables *and* at most $(k'')^2$ actions.

*Proof.* To prove this claim, we first show that in order to decide $c$-DET it suffices to keep a *simple* digraph $H$ with at most $k''$ vertices. This immediately yields a polynomial bi-kernel. After showing this polynomial bi-kernel we will show that $H$ can in fact be translated back to an instance $\langle \mathbb{P}'', k' \rangle$ of $c$-DET whose size is polynomially bounded in $k'$, i.e., we obtain a polynomial kernel.

To start with, observe, that for the detection problem it suffices to only keep the extended causal graph. Furthermore, arcs differing only in their label have no influence on the size of the backdoor since all arcs with the same endpoints will be deleted whenever one of their endpoints is. Thus, $H$ can be constructed as follows: $H$ has the same vertices $N$ as $\mathcal{G}_{\mathrm{E}}(\mathbb{P}')$ and for each $x, y \in N$, $H$ contains an arc $(x, y)$ whenever $\mathcal{G}_{\mathrm{E}}(\mathbb{P}')$ contains an arc $(x, y)$ with label $l$. The simple digraph $H$ is then a polynomial bi-kernel for $\langle \mathbb{P}, k \rangle$.

In order to obtain a polynomial kernel, let us transform $H$ into $\mathbb{P}''$ as follows. For every vertex $v \in V(H)$ we introduce, by slight abuse of notation, a variable $v$ to the set of variables $V$. The domain $D$ is set to be $\{0, 1\}$. Then, whenever there is an arc $(v, w)$ in $H$ with $v, w \in V(H)$, we add a new action $a$ with $\mathsf{pre}(a)[v] = 0$ and $\mathsf{eff}(a)[w] = 0$ to the set of actions $A$. In the initial state $I$ all variables are set to 0 while the goal needs every variable to be 1. Clearly, the instance $\mathbb{P}'' = \langle V, D, A, I, G \rangle$ of SAS$^+$ PLANNING or BOUNDED SAS$^+$ PLANNING is a trivial NO-instance but it is easy to see that

(i) $\mathsf{IL}(\mathcal{G}_{\mathrm{E}}(\mathbb{P}'')) = H$ where $\mathsf{IL}(\cdot)$ removes the labels from arcs in a labeled digraph to obtain a digraph, and thus

(ii) the instance $\langle \mathbb{P}'', k' \rangle$ of $c$-DET is equivalent to $\langle \mathbb{P}, k \rangle$.

It is now easy to see that $\langle \mathbb{P}'', k' \rangle$ is a polynomial kernel. □

## Proof of Theorem 4

**Theorem 4** (*). 2-EVAL *is* paraNP*-hard.*

*Proof.* We reduce from the well-known NP-complete problem 3-SAT. Let $\varphi$ be a 3-CNF formula with variables $x_1, \dots, x_n$ and clauses $C_1, \dots, C_m$. We construct an instance $\langle \mathbb{P}, S \rangle$ of 2-EVAL as follows. $\mathbb{P}$ has two binary variables $s_i$ and $x_i$ for every $i$ with $1 \leq i \leq n$ and one variable $\varphi$ with domain $\{0, \dots, m\}$. Furthermore, for every $i$ with $1 \leq i \leq n$ and $b \in \{0, 1\}$, $\mathbb{P}$ contains an action $a_{x_i}(b)$ such that $\mathsf{pre}(a_{x_i}(b))[s_i] = 0$, $\mathsf{eff}(a_{x_i}(b))[s_i] = 1$, and $\mathsf{eff}(a_{x_i}(b))[x_i] = b$. Finally, for every $j$ with $1 \leq j \leq m$ and $l \in \{1, 2, 3\}$ such that $x_i$ or $\bar{x}_i$ is the $l$-th literal of $C_j$, $\mathbb{P}$ contains the action $a_j^l$ such that $\mathsf{pre}(a_j^l)[s_i] = 1$, $\mathsf{pre}(a_j^l)[\varphi] = j - 1$, $\mathsf{eff}(a_j^l)[\varphi] = j$ and either $\mathsf{pre}(a_j^l)[x_i] = 1$ if $x_i$ is the $l$-th literal of $C_j$ or $\mathsf{pre}(a_j^l)[x_i] = 0$ if $\bar{x}_i$ is the $l$-th literal of $C_j$. The initial state requires all variables to have value 0 and the goal state only requires the variable $\varphi$ to have value $m$. We now set $S$ to contain only the variable $\varphi$, which implies that $\mathsf{cc\text{-}size}(\mathcal{G}_{\mathrm{E}}(\mathbb{P} \setminus S)) \leq c = 2$, as required.

It is now straightforward to show that $\varphi$ is satisfiable if and only if $\mathbb{P}$ has a plan. $\square$

## Proof of Theorem 5

**Theorem 5** (*). $c$-BOUNDED EVAL *is* W*[1]-hard (even if additionally parameterized by the number of variables) and in* XP.

*Proof.* We present a reduction from PARTITIONED CLIQUE. Let $G' = (V, E)$ be a $k$-partite graph where $V$ is partitioned into $P_1, \dots, P_k$ with $|P_i| = |P_j|$ for $i, j \in [k]$ and let $V = \{v_1, \dots, v_n\}$. We create an instance $\langle \mathbb{P}, k'', S \rangle$ of $c$-BOUNDED EVAL as follows. The set of variables is given by $V' := P \cup C$, where $P = \{P_1, \dots, P_k\}$ contains (by slight abuse of notation) variables representing the partitions of the vertices of graph $G'$ and $C = \{c_{i,j} \mid 1 \leq i < j \leq k\}$ contains a set of pair-variables representing the edges between the partitions that have to be covered. The domain elements are used to represent the state of the pair variables as well as the vertices of the graph and hence we define $D := \{0, 1\} \cup V$. We create two sets of actions:

- An action $set_i^j$ is contained in set $A_{\text{sel}}$ for each $i \in [k]$ such that $j \in P_i$. We define the precondition as $\mathsf{pre}(set_i^j)[P_i] = 0$ and the effect as $\mathsf{eff}(set_i^j)[P_i] = j$.
- For each edge $\{v_i, v_j\} \in E$ such that $v_i \in P_{i'}, v_j \in P_{j'}$, for $1 \leq i < j \leq n$ and $i', j' \in [k]$ (with $i' \neq j'$) an action $check_{i,j}$ is contained in the set $A_{\text{check}}$. The preconditions and effects are given by $\mathsf{pre}(check_{i,j})[P_{i'}] = v_i$, $\mathsf{pre}(check_{i,j})[P_{j'}] = v_j$, and $\mathsf{eff}(check_{i,j})[c_{i',j'}] = 1$.

In the following, let $k' = \binom{k}{2}$. The set of actions is then given by $A := A_{\text{sel}} \cup A_{\text{check}}$, the initial state by $I := 0^{k+k'}$, and in the goal $G$ we set all variables in $V_{\text{check}}$ to 1 and all others to undefined ($\mathbf{u}$). Finally, the constructed instance is given by $\langle \mathbb{P}, k'' := k + k', S := V' \rangle$ with $\mathbb{P} := \langle V', D, A, I, G \rangle$.

In this construction the actions from $A_{\text{sel}}$ are used to assign the selected vertex to the respective partition. After that the actions from $A_{\text{check}}$ are used to set the variables in $C$ to 1 in order to reach the goal. As the bound on the plan length is $k + k'$ this is just sufficient to assign $k$ vertices to the $k$ partitions and to set all edges of the $k$-clique to 1. This is, however, only possible if the assigned vertices form a $k$-clique in $G'$.

Observe that $S$ is a trivial variable-deletion backdoor set and that the number of variables $|V'|$ is indeed bounded by $k + k'$. Furthermore, notice that the unbounded domain size is essential to represent the vertices to choose from. It is easy to verify that $\langle \mathbb{P}, k'', S \rangle$ is a YES-instance iff $\langle k, G', \{V_1, \dots, V_k\} \rangle$ is a YES-instance.

For the membership in XP notice that all possible plans of length $k''$ can be trivially bounded by $\mathcal{O}(n^{k''})$, which immediately yields the membership result. $\square$

## Proof of Lemma 6

**Lemma 6** (*). $|\mathrm{ET}(\mathbb{P}, S)|$ *is at most* $c \cdot ((|D| + 1)^{2(c+|S|)}) \cdot 2^{(|D|+1)^{2(c+|S|)}}$.

*Proof.* The claim of the lemma follows from the following observations for every component $C \in \mathsf{C}(\mathbb{P}, S)$:

(i) $C$ has at most $c$ variables,

(ii) there are at most $|D|^{c+|S|} \cdot (|D| + 1)^{c+|S|} \leq (|D| + 1)^{2(c+|S|)}$ possible configurations for the initial state and for the goal state on the variables in $V(C) \cup S$,

(iii) there are at most $(|D| + 1)^{2(c+|S|)}$ possible combinations of preconditions and effects for any action that involves only variables in $V(C) \cup S$,

(iv) there are at most $(|D| + 1)^{2(c+|S|)}$ possible distinct actions that involve only variables in $V(C) \cup S$,

(v) there are at most $2^{(|D|+1)^{2(c+|S|)}}$ distinct sets of actions that involve only variables in $V(C) \cup S$.

$\square$

## Proof of Lemma 7

**Lemma 7** (*). *Given a* SAS$^+$ PLANNING *instance* $\mathbb{P}$*, a c-Extended Causal Backdoor* $S$ *of* $\mathbb{P}$*, and* $l \in \mathbb{N}$*, we can solve* $c$-EVAL *and* $c$-BOUNDED EVAL *of* $\mathrm{E}(\mathbb{P}, S, l)$ *in time* $\mathcal{O}(|D|^{V(\mathrm{E}(\mathbb{P},S,l))} + |V| \cdot |\mathrm{ET}(\mathbb{P}, S)|)$.

*Proof.* Let $\mathbb{P}$, $S$, and $l \in \mathbb{N}$ be defined as in the statement of the lemma. Furthermore, let $\mathbb{P}' = \mathrm{E}(\mathbb{P}, S, l)$ and $k \in \mathbb{N}$. We need to solve the instances $\mathbb{I} = \langle \mathbb{P}', S \rangle$ of $c$-EVAL and $\mathbb{I}_B = \langle \mathbb{P}', S, k \rangle$ of $c$-BOUNDED EVAL. We do this by executing the following three steps:

1) Compute the equivalence class with respect to $\equiv$ of each component $C$ in $\mathrm{C}(\mathbb{P}, S)$.
2) Compute the reduced instance $\mathbb{P}'$ from $\mathbb{P}$ by deleting all but at most $l$ components for each equivalence class from $\mathbb{P}$.
3) Compute the state-transition graph of the reduced instance $\mathbb{P}'$ and solve $\mathbb{I}$ or $\mathbb{I}_B$, respectively. Return YES, if $\mathbb{I}$ or $\mathbb{I}_B$ is a YES-instance, respectively, and NO otherwise.

The runtime of the algorithm is obtained as follows. Step 1 needs time $\mathcal{O}(|V| \cdot |\mathrm{ET}(\mathbb{P}, S)|)$ (recall that $\mathrm{ET}(\mathbb{P}, S)$ is the number of equivalence classes with respect to $\equiv$), Step 2 can be executed in time $\mathcal{O}(|V|)$ (by using clever data structures), and Step 3 needs time at most $\mathcal{O}(|D|^{V(\mathrm{E}(\mathbb{P},S,l))})$. Hence, the total runtime of the algorithm is $\mathcal{O}(|D|^{V(\mathrm{E}(\mathbb{P},S,l))} + |V| \cdot |\mathrm{ET}(\mathbb{P}, S)|)$. ☐

## Proof of Theorem 8

**Theorem 8** (*). $c$-BOUNDED EVAL *is fpt for instances with bounded domain.*

*Proof.* Let $\mathbb{I} = \langle \mathbb{P}, k, S \rangle$ be an instance of $c$-BOUNDED EVAL. Because we only consider plans of length at most $k$ for $\mathbb{P}$ and every action of $\mathbb{P}$ effects at most one component, we obtain that every such plan of $\mathbb{P}$ changes variables of at most $k$ components of $\mathrm{C}(\mathbb{P}, S)$. Hence, the instance $\mathbb{P}' = \mathrm{E}(\mathbb{P}, S, k + 1)$ is equivalent to $\mathbb{P}$. It follows from Lemma 7 that $\mathbb{I}$ can be solved in time $\mathcal{O}(D^{V(\mathbb{P}')} + |V| \cdot |\mathrm{ET}(\mathbb{P}, S)|)$. Because $\mathbb{P}'$ has at most $|S| + c \cdot (k + 1) \cdot |\mathrm{ET}(\mathbb{P}, S)|$ many variables, the result follows from the upper bound for $\mathrm{ET}(\mathbb{P}, S)$ given in Lemma 6. ☐

## Proof of Lemma 9

**Lemma 9** (*). *Let* $\mathbb{P}$ *be a* SAS$^+$ PLANNING *instance and* $s_1, s_2$ *be two equivalent (total) states of the variables of* $\mathbb{P}$*. Then, for any plan from* $s_1$ *to* $G$ *in* $\mathbb{P}$*, there is a plan from* $s_2$ *to* $G$ *in* $\mathbb{P}$ *of the same length, and vice versa.*

*Proof.* Let $\omega_1$ be a plan from $s_1$ to $G$ in $\mathbb{P}$. We show how to construct a plan $\omega_2$ from $s_2$ to $G$ in $\mathbb{P}$ of the same length as $\omega_1$.

For any action $a \in A$, we define the action $a'$ as follows:

(i) if $a$ has no preconditions or effects on a component in $\mathrm{C}(\mathbb{P}, S)$, we set $a' = a$,

(ii) otherwise, let $C_1$ be the component in $\mathrm{C}(\mathbb{P}, S)$ such that $a$ has at least one precondition or effect on $C_1$ and let $C_2$ be a component in $\mathrm{C}(\mathbb{P}, S)$ equivalent to $C_1$ such that $s_1[C_1] = s_2[C_2]$, then we set $a'$ to be the action that corresponds to $a$ in $C_2$. Note that because $\mathrm{ESV}(\mathbb{P}, S, s_1) = \mathrm{ESV}(\mathbb{P}, S, s_2)$ such a component $C_2$ does always exist.

Then, $\omega_2$ is obtained from $\omega_1$ by replacing every action $a \in \omega_1$ with the action $a'$ as defined above. It is straightforward to show that $\omega_2$ is a plan from $s_2$ to $G$ in $\omega$. Because the reverse direction is symmetric this concludes the proof of the lemma. ☐

## Proof of Theorem 10

**Theorem 10** (*). $c$-EVAL *is in* XP *for planning instances with bounded domain.*

*Proof.* Let $\mathbb{I} = \langle \mathbb{P}, S \rangle$ be an instance of $c$-EVAL with maximum domain size $|D|$, and let $s$ be a (total) state of $\mathbb{P}$. Then, the vector $v = \mathrm{ESV}(\mathbb{P}, S, s)$ has at most $\sum_{i=1}^{i \leq |\mathrm{ET}(\mathbb{P},S)|} |\mathrm{ST}(\mathbb{P}(V(C_i)))| \leq |\mathrm{ET}(\mathbb{P}, S)| \cdot |D|^c$ entries each of them reaching from 0 to at most the number $n$ of components of $\mathrm{C}(\mathbb{P}, S)$. Similarly, the number of states of the variables of $S$ is at most $|D|^{|S|}$. It follows that the number of equivalent classes (of the total states of $\mathbb{P}$) is at most $(n + 1)^{|\mathrm{ET}(\mathbb{P},S)| \cdot |D|^c} \cdot |D|^{|S|}$. Using the upper bound on $\mathrm{ET}(\mathbb{P}, S)$ from Lemma 6, we obtain that for fixed $c$, $|D|$, and $|S|$, we can build the compressed state-transition graph, and hence solve the instance $\mathbb{I}$, in time that is polynomial in the input size, as required. ☐

## Proof of Theorem 11

**Theorem 11.** *Let $\mathbb{P}$ be a* SAS$^+$ PLANNING *instance with goal $G$, $S$ be a c-Extended Causal Backdoor of $\mathbb{P}$, $d = \text{ESV-D}(\mathbb{P}, S)+1$, and $\mathbb{V}(\mathbb{P}, S) = \langle Q, \Delta\rangle$ be the d-VASS obtained from $\mathbb{P}$ and $S$. Furthermore, let $s_I$ and $s_G$ be two $\mathbb{V}(\mathbb{P}, S)$-states (as defined above) and $g$ be the number of components $C$ in $\text{C}(\mathbb{P}, S)$ with $G[V(C)] \neq \mathbf{u}$. Then, $s_I$ and $s_G$ satisfy the following properties: (P1) $\mathbb{P}$ has a plan of length at most $k$ if and only if there is a covering of $s_G$ from $s_I$ in $\mathbb{V}(\mathbb{P}, S)$ of length at most $2k + g + 2$, (P2) $|Q| \leq |\text{ST}(\mathbb{P}(S))| \cdot (1 + |A(\mathbb{P}_\equiv)|) + 1$, (P3) $||\Delta|| \leq 1$, (P4) $||s_G|| = g$, and (P5) $\mathbb{V}(\mathbb{P}, S)$ can be constructed in time $\mathcal{O}(|\text{ET}(\mathbb{P}, S)| \cdot (|V| + (|D| + 1)^{2(|S|+c)}))$.*

*Proof.* We start by showing Property (P1). We need the following claim.

*Claim* 2 (*). There is a run of length $l$ from $s_I$ to some $\mathbb{V}(\mathbb{P}, S)$-state $s_e \in \{\, \langle q_s, v\rangle \mid s \in \text{ST}(\mathbb{P}(S))$ and $v \in \mathbb{N}^d \,\}$ if and only if there is a total state $s$ of $\mathbb{P}$ and a sequence $\omega$ of actions of length $\frac{l-1}{2}$ such that $s$ is the result of $\omega$ in $I$ and $s_e = \langle q_{s[S]}, \text{ESV}(\mathbb{P}, S, s)^*\rangle$.

The above claim follows by a repeated application of the following claim.

*Claim* 3. Let $s$ be a total state of $\mathbb{P}$. Then there is a run from $\langle q_{s[S]}, \text{ESV}(\mathbb{P}, S, s)^*\rangle$ to some $\mathbb{V}(\mathbb{P}, S)$-state $s_e \in \{\, \langle q_s, v\rangle \mid s \in \text{ST}(\mathbb{P}(S))$ and $v \in \mathbb{N}^d \,\}$ using (at most) 2 transitions if and only if there is a total state $s'$ and an action $a$ of $\mathbb{P}$ such that $s'$ is the result of $a$ in $s$ on $\mathbb{P}$ and $s_e = \langle q_{s'[S]}, \text{ESV}(\mathbb{P}, S, s')^*\rangle$.

Towards showing the forward direction of the claim, assume that there is a run $r$ from $\langle q_{s[S]}, \text{ESV}(\mathbb{P}, S, s)^*\rangle$ to some $s_e = \langle q_e, v_e\rangle$ that uses (at most) two transitions. Because of the construction of $\mathbb{V}(\mathbb{P}, S)$, $r$ must use two transitions of the form $t_1 = \langle q_{s[S]}, v_1, q_{s[S],a_\equiv}\rangle$ and $t_2 = \langle q_{s[S],a_\equiv}, v_2, q_e\rangle$ for some $a_\equiv \in A_\equiv$. W.l.o.g. we will assume that $a_\equiv$ has at least one precondition or effect on some component $C_\equiv \in \text{C}(\mathbb{P}_\equiv, S)$ (otherwise the proof becomes only easier). Because of S2 there is a unique nonzero entry of $v_1$ and this entry has value $-1$ and corresponds to a state of $C_\equiv$ that is compatible with $\mathsf{pre}(a_\equiv)[V(C_\equiv)]$. Because $r$ is a run, we obtain that the corresponding entry in $\text{ESV}(\mathbb{P}, S, s)^*$ is at least 1 and hence there is a component $C \in \text{C}(\mathbb{P}, S)$ that is equivalent to $C_\equiv$ such that $s[V(C)]$ is compatible with the preconditions of the (unique) action $a$ in $\mathbb{P}(S \cup V(C))$ that is equivalent to $a_\equiv$. Furthermore, again because of S2, the unique non-zero entry of $v_2$ is $+1$ and corresponds to the result of $a$ in $s[V(C)]$ on $\mathbb{P}(V(C))$. Let $s'$ be the result of $a$ in $s$ on $\mathbb{P}$. Then, $v_e = \text{ESV}(\mathbb{P}, S, s')^*$, as required. It remains to show that $e = s'[S]$. But this follows immediately from S2, because $s[S] \in Z(\mathbb{P}_\equiv, \mathsf{pre}(a_\equiv), S)$ and $e[S]$ is the result of $a_\equiv$ in $s[S]$ on $\mathbb{P}_\equiv(S)$.

For the reverse direction, assume that $s'$ is the result of $a$ in $s$ on $\mathbb{P}$. W.l.o.g., we can assume that $a$ has at least one precondition or effect on some component $C \in \text{C}(\mathbb{P}, S)$ as otherwise the proof would be merely simpler. Let $a_\equiv$ be the unique action that corresponds to $a$ in $\mathbb{P}_\equiv$ and let $C_\equiv$ be the unique component equivalent to $C$ in $\mathbb{P}_\equiv$. Because $s'$ is the result of $a$ in $s$, we obtain that (1) $s[S] \in Z(\mathbb{P}_\equiv, \mathsf{pre}(a_\equiv), S)$, (2) $s[V(C)] \in Z(\mathbb{P}_\equiv, \mathsf{pre}(a_\equiv), V(C_\equiv))$, and (3) the $\text{ESV-I}(\mathbb{P}_\equiv, S, C_\equiv, s[V(C)])$-th entry of $\text{ESV}(\mathbb{P}, S, s)$ is at least 1. (1) and (2) together with S2 (second subitem) imply that there is a control point $q_{s[S],a_\equiv}$ and two transitions $t_1 = \langle q_{s[S]}, v_1, q_{s[S],a_\equiv}\rangle$ and $t_2 = \langle q_{s[S],a_\equiv}, v_2, q_{s'[S]}\rangle$ , where $v_1 \in \mathbb{N}^d$ is $-1$ at the $\text{ESV-I}(\mathbb{P}_\equiv, S, C_\equiv, s[V(C)])$-th entry and 0 on all other entries and $v_2 \in \mathbb{N}^d$ is $+1$ at the $\text{ESV-I}(\mathbb{P}_\equiv, S, C, s'[S])$-th entry and 0 on all other entries. Hence, it follows from (3) that executing the transitions $t_1$ and $t_2$ from the state $\langle q_{s[S]}, \text{ESV}(\mathbb{P}, S, s)^*\rangle$ leads to the required run. This concludes the proof of the claim.

Using Claim 2, we are now ready to Property (P1). Assume that $\omega = \langle a_1, \dots, a_k\rangle$ is a plan for $\mathbb{P}$ that results in the state $s$. It follows from Claim 2 that there is a run $r$ from $s_I$ to $\langle q_{s[S]}, \text{ESV}(\mathbb{P}, S, s)^*\rangle$ of length $2k + 1$. Because $s$ is a goal state, we obtain from S3 that there is a transition $t = \langle q_{s[S]}, \mathbf{0}, q_G\rangle$. Furthermore, again because $s$ is a goal state, it follows that the sum of all entries of $\text{ESV}(\mathbb{P}, S, s)$ that correspond to a goal state of some component is equal to $g$. Hence, we obtain a covering of $s_G$ from $s_I$ of length $2k + 1 + 1 + g = 2k + g + 2$, by appending $t$ followed by all the $g$ applicable transitions of the form $\langle q_G, v, q_G\rangle$ (obtained from S4) to $r$.

For the reverse direction assume that $r$ is a covering of length $l$ of $s_G$ from $s_I$ in $\mathbb{V}(\mathbb{P}, S)$. Since all outgoing transitions from $q_G$ are self-loops, we obtain that $\mathbb{V}(\mathbb{P}, S)$-states involving $q_G$ only appear (together) at the end of $r$. Let $s_e$ be the last state reached by $r$ before the first occurrence of a state involving $q_G$ and let $l'$ be the length of the subsequence of $r$ from $s_I$ to $s_e$. Because the only transitions leading to a state involving $q_G$ are from states in $\{\, \langle q_s, v\rangle \mid s \in \text{ST}(\mathbb{P}(S))$ and $v \in \mathbb{N}^d \,\}$, we obtain that $s_e \in \{\, \langle q_s, v\rangle \mid s \in \text{ST}(\mathbb{P}(S))$ and $v \in \mathbb{N}^d \,\}$ (see S3). It follows from Claim 2 that there is a total state $s$ of $\mathbb{P}$ and a sequence of actions $\omega$ of length $\frac{l'-1}{2}$ such that $s$ is the result of $\omega$ in $I$ and $s_e = \langle q_{s[S]}, \text{ESV}(\mathbb{P}, S, s)^*\rangle$. We claim that $\omega$ is a plan of length $\frac{l-g-2}{2}$ for $\mathbb{P}$. Since the only transitions leading to $q_G$ are of the form $\langle q_{s'}, \mathbf{0}, q_G\rangle$ where $s' \in Z(\mathbb{P}_\equiv, G_\equiv, S)$ (see S3), we obtain that $s[S]$ is compatible with $G$. Let $\langle q_G, v\rangle$ be the last $\mathbb{V}(\mathbb{P}, S)$-state visited by $r$. Because $r$ covers $s_G$, we obtain that $v[d] \geq g$. Because all transitions that increase the value of the $d$-th dimension in $\mathbb{V}(\mathbb{P}, S)$ also decrease the value of a dimension that corresponds to some goal state of some component by the same amount (see S4), it follows that the number of components, whose state in $s$ is compatible with the goal state, is at least $g$. Because $g$ is equal to the total number of components in $\text{C}(\mathbb{P}, S)$ for which the goal state is defined (see S5), we obtain that all of these components have reached a goal state in $s$. Hence, $s$ is compatible with $G$. Because exactly $g + 1$ transitions are executed by $r$ after the state $s_e$, we obtain that $l' = l - 1 - g$ and hence the length of $\omega$ is $\frac{l'-1}{2} = \frac{l-g-2}{2}$, as required.

Properties P2–P4 now follow immediately from the construction.

Towards showing Property P5, first note that to construct $\mathbb{V}(\mathbb{P}, S)$ from $\mathbb{P}$ and $S$, we first need to construct $\mathbb{P}_{\equiv}$, which can be done in time $\mathcal{O}(|V| \cdot |\text{ET}(\mathbb{P}, S)|)$ (see also the proof of Lemma 7) and then we need to execute steps S1–S5 of the construction given in Section 3.2. Step S1 takes time $\mathcal{O}(|D|^{|S|})$, Step S2 takes time $\mathcal{O}(|A(\mathbb{P}_{\equiv})| \cdot |D|^{|S|+c}) = \mathcal{O}((|D|+1)^{2(|S|+c)} \cdot |\text{ET}(\mathbb{P}, S)| \cdot |D|^{|S|+c}) = \mathcal{O}((|D|+1)^{2(|S|+c)} \cdot |\text{ET}(\mathbb{P}, S)|)$, Step S3 takes time $\mathcal{O}(|D|^{|S|})$, Step S4 takes time $\mathcal{O}(|\text{ET}(\mathbb{P}, S)||D|^{c})$, and the time needed for Step S5 is superseded by the time to construct $\mathbb{P}_{\equiv}$. Hence, the total runtime required to construct $\mathbb{V}(\mathbb{P}, S)$ is $\mathcal{O}(|\text{ET}(\mathbb{P}, S)| \cdot (|V| + (|D|+1)^{2(|S|+c)}))$, as required.

□

## Proof of Lemma 14

**Lemma 14** (*). *Let $\mathbb{I} = \langle \mathbb{P}, S \rangle$ be an instance of c-*EVAL *that contains no mixed actions. Then $\mathbb{I}$ is equivalent to the c-*EVAL *instance $\mathbb{I}' = \langle \mathbb{P}', S \rangle$, where $\mathbb{P}' = \text{E}(\mathbb{P}, S, m)$ and $m = \max_{T \in \text{ET}(\mathbb{P},S)} |\text{ST}(\mathbb{P}(V(T)))|$.*

*Proof.* Let $\mathbb{I} = \langle \mathbb{P}, S \rangle$, $\mathbb{I}' = \langle \mathbb{P}', S \rangle$, and $m$ be defined as in the statement of the lemma. We will show that $\mathbb{P}$ has a plan if and only if $\mathbb{P}'$ has a plan. Suppose that $\mathbb{P}$ has a plan $\omega = \langle a_1, \dots, a_l \rangle$. We will first inductively define sequences $\omega'(i)$ of actions of $\mathbb{P}'$, for every $i$ with $0 \leq i \leq l$, that satisfy the Properties P1–P4 below. In the following let $s_i$ be the result of $\langle a_1, \dots, a_i \rangle$ in $I$ on $\mathbb{P}$ and let $s'_i$ be the result of $\omega'(i)$ in $I'$ on $\mathbb{P}'$.

P1 $\omega'(i)$ is a valid sequence of actions in $\mathbb{P}'$ starting from $I'$,

P2 $s_i[S] = s'_i[S]$.

P3 For every $T \in \text{ET}(\mathbb{P}, S)$ and every $z \in \text{ST}(\mathbb{P}(V(T))) \setminus \{I[V(T)]\}$, the following holds: there is a component $C' \in \text{C}(\mathbb{P}', S)$ equivalent to $T$ such that $z = s'_i[V(C')]$ if and only if there are $j_1$ and $j_2$ with $0 \leq j_1 \leq i$ and $i \leq j_2 \leq l$ and two components $C_1$ and $C_2$ equivalent to $T$ in $\text{C}(\mathbb{P}, S)$ such that $z = s_{j_1}[V(C_1)]$ and $z = s_{j_2}[V(C_2)]$.

P4 For every $T \in \text{ET}(\mathbb{P}, S)$, the number of components $C' \in \text{C}(\mathbb{P}', S)$ equivalent to $T$ such that $s'_i[V(C')] = I'[V(C')]$ is at least the number of components equivalent to $T$ in $\mathbb{P}'$ minus the number of states $z \in \text{ST}(T) \setminus \{I[V(T)]\}$ such that there is a $j$ with $0 \leq j \leq i$ and a component $C \in \text{C}(\mathbb{P}, S)$ such that $s_j[V(C)] = z$.

We start by setting $\omega'(0)$ to the empty sequence. Clearly, $\omega'(0)$ satisfies the properties P1–P4. So suppose that we have already constructed a sequence $\omega'(i-1)$ satisfying the properties P1–P4. We show next how to obtain a sequence $\omega'(i)$ from $\omega'(i-1)$ that also satisfies the properties P1–P4. Depending on the preconditions and effects of $a_i$, we distinguish the following cases:

- If $a_i$ has neither an effect nor a precondition on any component of $\text{C}(\mathbb{P}, S)$, we set $\omega'(i) = \omega'(i-1), \langle a_i \rangle$. Clearly, $\omega'(i)$ satisfies the properties P1–P4.
- If $a_i$ has either a precondition or an effect on a component $C$ in $\text{C}(\mathbb{P}, S)$ but $s_i[V(C)] = s_{i-1}[V(C)]$, we set $\omega'(i) = \omega'(i-1), \langle a'_i \rangle$, where $a'_i$ is the action that corresponds to $a_i$ in a component $C' \in \text{C}(\mathbb{P}', S)$ equivalent to $C$ such that $s'_{i-1}[V(C')] = s_{i-1}[V(C)]$. Note that such a component $C'$ must always exists due to the properties P3 and P4. Clearly, $\omega'(i)$ satisfies the properties P1–P4.
- Otherwise, i.e., if $a_i$ has either a precondition or an effect on a component $C$ in $\text{C}(\mathbb{P}, S)$ and $s_i[V(C)] \neq s_{i-1}[V(C)]$, we distinguish the following cases:
  - there is a $j$ with $0 \leq j < i$ and a component $C' \in \text{C}(\mathbb{P}, S)$ equivalent to $C$ such that $s_j[V(C')] = s_i[V(C)]$, we again distinguish the following cases:
    - if $s_{i-1}[V(C)] = I[V(C)]$, we set $\omega'(i) = \omega'(i-1)$. Clearly, $\omega'(i)$ satisfies the properties P1–P4.
    - if there is a $e$ with $i < e \leq l$ and a component $C'' \in \text{C}(\mathbb{P}, S)$ equivalent to $C$ with $s_{i-1}[V(C)] = s_e[V(C'')]$, we set $\omega'(i) = \omega'(i-1)$. Clearly, $\omega'(i)$ satisfies the properties P1–P4.
    - otherwise, let $C_1, \dots, C_t$ be all the components in $\text{C}(\mathbb{P}', S)$ equivalent to $C$ such that $s'_{i-1}[V(C_p)] = s_{i-1}[V(C)]$ for every $p$ with $1 \leq p \leq t$. Furthermore, for every $p$ with $1 \leq p \leq t$, let $a_i^p$ be the action in $C_p$ that corresponds to $a_i$. Then, we set $\omega'(i) = \omega'(i-1), \langle a_i^1, \dots, a_i^t \rangle$. Clearly, $\omega'(i)$ satisfies the properties P1–P4.
  - otherwise, we again distinguish the following cases:
    - if $s_{i-1}[V(C)] = I[V(C)]$, we set $\omega'(i) = \omega'(i-1), \langle a'_i \rangle$, where $a'_i$ is the action that corresponds to $a_i$ in a component $C' \in \text{C}(\mathbb{P}', S)$ equivalent to $C$ such that $s'_{i-1}[V(C')] = s_{i-1}[V(C)]$. Note that such a component $C'$ always exists due to Property P4. Clearly, $\omega'(i)$ satisfies the properties P1–P4.
    - if $s_{i-1}[V(C)] \neq I[V(C)]$ and for all $j$ with $i \leq j \leq l$ there is no component $C'' \in \text{C}(\mathbb{P}, S)$ equivalent to $C$ such that $s_j[V(C'')] = s_{i-1}[V(C)]$. Let $C_1, \dots, C_t$ be all components in $\text{C}(\mathbb{P}', S)$ equivalent to $C$ such that $s'_{i-1}[V(C_q)] = s_{i-1}[V(C)]$ and let $a_i^1, \dots, a_i^t$ be the actions corresponding to $a_i$ in $C_q$ for every $1 \leq q \leq t$. Because of Property P3, we have $t \geq 1$. If $t > 1$ then we set $\omega'(i) = \omega'(i-1), \langle a_i^1 \rangle$. Otherwise, let $C_I \in \text{C}(\mathbb{P}', S)$ be a component equivalent to $C$ such that $s'_{i-1}[V(C_I)] = I'[V(C_I)]$ and for every action $a$ in $\mathbb{P}'(C_1 \cup S)$ that has at least one effect in $C_1$ let $a_I$ be the corresponding action in $C_I$. Note that such a component $C_I$ always exists due to Property P4. We obtain $\omega'(i)$ from

$\omega'(i-1)$ by inserting the action $a_I$ after every occurrence of an action $a$ in $\omega'(i-1)$ that has at least one effect in $C_1$. Clearly, $\omega'(i)$ satisfies the properties P1–P4.

* otherwise, let $C_1, \dots, C_t$ be all the components in $\mathrm{C}(\mathbb{P}', S)$ equivalent to $C$ such that $s'_{i-1}[V(C_p)] = s_{i-1}[V(C)]$ for every $p$ with $1 \leq p \leq t$. Furthermore, for every $p$ with $1 \leq p \leq t$, let $a_i^p$ be the action in $C_p$ that corresponds to $a_i$. Then, we set $\omega'(i) = \omega'(i-1), \langle a_i^1, \dots, a_i^t \rangle$. Clearly, $\omega'(i)$ satisfies the properties P1–P4.

This shows that a sequence $\omega'(l)$ satisfying the properties P1–P4 can be constructed. Observe that already $\omega'(l)$ almost constitutes a plan for $\mathbb{P}'$. In particular, because $\omega$ is a plan for $\mathbb{P}$, we obtain from Property P2 that $s'_l[S]$ is compatible with $G'[S]$, and Property P3 implies that for every component $C$ in $\mathrm{C}(\mathbb{P}', S)$, with $s'_l[V(C)] \neq I'[V(C)]$, it holds that $s'_l[V(C)]$ is compatible with $G'[V(C)]$. Hence, it only remains to show how to extend $\omega'(l)$ to ensure that also the components that are still in the initial state after applying $\omega'(l)$ to $I'$ achieve a goal state. We achieve this by extending $\omega'(l)$ in the following way for every $T \in \mathrm{ET}(\mathbb{P}, S)$:

Let $C_1, \dots, C_t$ be the components equivalent to $T$ in $\mathrm{C}(\mathbb{P}', S)$ such that $s'_l[V(C_q)] = I'[V(C_q)]$ for every $1 \leq q \leq t$. Clearly, if $t = 0$ then all components in $\mathrm{C}(\mathbb{P}', S)$ that are equivalent to $T$ have reached a goal state after executing $\omega'(l)$ and there is no need to adapt $\omega'(l)$ for the components equivalent to $T$. Otherwise, i.e., if $t \geq 1$, let $C$ be a component in $\mathrm{C}(\mathbb{P}', S)$ equivalent to $T$ such that $s'_l[V(C)]$ is compatible with $G'[V(C)]$. Observe that such a component $C$ must exits because of Property P3 and our assumption that $\omega$ is a plan for $\mathbb{P}$. Furthermore, for every action $a$ of $\mathbb{P}'(C \cup S)$ with at least one effect in $C$ and for every $1 \leq q \leq t$, let $a^q$ be the action of $C_q$ that corresponds to $a$. In order to ensure that the components $C_1, \dots, C_t$ reach a goal state, we insert the actions $a^1, \dots, a^t$ after every occurrence of an action $a$ in $\omega(l)$ that has at least one effect on $C$. After applying the above procedure for every $T \in \mathrm{ET}(\mathbb{P}, S)$, we obtain a plan $\omega'$ for $\mathbb{P}'$. This shows the forward direction of our claim.

For the reverse direction suppose that $\mathbb{P}'$ has a plan $\omega'$. Because $\mathbb{P}'$ is a sub-instance of $\mathbb{P}$, we obtain that $\omega'$ is a valid sequence of actions for $\mathbb{P}$ that can be applied to the initial state. Let $s$ be the result of $\omega'$ in $I$ on $\mathbb{P}$. Because $\omega'$ is a plan for $\mathbb{P}'$, it holds that $s[V(\mathbb{P}')]$ is compatible with the goal state. It hence only remains to show how $\mathbb{P}'$ can be extended to a plan for the remaining variables (components) of $\mathbb{P}$. We will obtain a plan $\omega$ for $\mathbb{P}$ by applying the following procedure for every component $C \in \mathrm{C}(\mathbb{P}, S) \setminus \mathrm{C}(\mathbb{P}', S)$:

Because $\omega'$ does not contain any actions with an effect on $C$, it holds that $s[V(C)] = I[V(C)]$. Furthermore, there is a component $C'$ of $\mathbb{P}'$ (and hence also of $\mathbb{P}$) such that $C'$ is equivalent to $C$ and $s[V(C')]$ is compatible with $G[V(C')]$. For every actions $a'$ of $\mathbb{P}(C' \cup S)$ that has at least one effect on $C'$, let $a$ be the unique action of $\mathbb{P}(C \cup S)$ that corresponds to $a'$ in $C$. To ensure that $C$ reaches a goal state, we insert the action $a$ after every occurrence of an action $a'$ in $\omega'$ that has at least one effect on $C'$. After applying the above procedure for every $C \in \mathrm{C}(\mathbb{P}, S) \setminus \mathrm{C}(\mathbb{P}', S)$, we obtain a plan $\omega$ for $\mathbb{P}$. ☐

## Proof of Theorem 16

To obtain a kernelization lower-bound result for variable-deletion backdoors, we first show that SAS$^+$ planning with bounded domain does not admit a polynomial kernel when parameterized by the number of variables.

**Lemma 19.** *SAS$^+$ planning with bounded domain parameterized by the number of variables does not admit a polynomial kernel unless* coNP $\subseteq$ NP/poly

*Proof.* We show this result by presenting a strong OR-composition for SAS$^+$ planning under bounded domain parameterized by the number of variables. Due to Proposition 18 this implies that SAS$^+$ planning under bounded domain parameterized by the number of variables does not have a polynomial (bi-)kernel unless coNP $\subseteq$ NP/poly.

Let $(\mathbb{P}_1, |V_1|), \dots, (\mathbb{P}_t, |V_t|)$ be a sequence of SAS$^+$ instances where $\mathbb{P}_i = \langle V_i, D_i, A_i, I_i, G_i \rangle$, for $i \in [t]$. By $V_{\text{max}}$ we denote an arbitrary element of $\{V_1, \dots, V_t\}$ of maximum cardinality. Furthermore let $D = D_1 \cup \dots \cup D_t$ and $p = |V_{\text{max}}|$. W.l.o.g. we assume now that all instances $\mathbb{P}_1, \dots, \mathbb{P}_t$ range over the variables $V_{\text{max}}$ and over domain $D$. Notice that this can easily be ensured by renaming the variables of the instances.

We assume (w.l.o.g.) that $t = 2^x$ for some $x \in \mathbb{N}$ as this can be easily ensured via "padding" with no-instances. The proof proceeds now by distinguishing between two cases.

Case 1: $t > d^p$. Recall that for any fixed domain size $d$, SAS$^+$ planning can be solved in time $\mathcal{O}^*(d^p)$, since the size of the state transition graph is bounded by $\mathcal{O}(d^p)$. Let $n := \max_{i=1}^t \|\mathbb{P}_i\|$. Thus, solving all instances in the sequence can be done in time $\mathcal{O}(t \cdot d^p \cdot poly(n))$.

In case there is an instance $\mathbb{P}_i$ which is a yes-instance of SAS$^+$, we return this witness $\mathbb{P}_i$. Otherwise $\mathbb{P}_1$ is returned which is known to be a no-instance. Since $\mathcal{O}(t \cdot d^p \cdot poly(n)) \leq \mathcal{O}(t^2 \cdot poly(n))$, the time of the whole procedure is bounded by a polynomial in $\sum_{i=1}^t \|\mathbb{P}_i\|$. Hence, we have obtained a composition for this case (Case 1).

Case 2: $t \leq d^p$. In this case we need to actually create a new instance that has a solution if and only if at least one of the input instances has a solution. To this end, we will build a set of "selection actions" that allow to select one of the instances while blocking all other instances. W.l.o.g. let us assume that the domain $D$ contains at least two elements 0 and 1.

We create a selection gadget for the instances as follows. Let $V_{\text{sel}} = \{sel_1^0, sel_1^1, \dots, sel_{\log t}^0, sel_{\log t}^1\}$ and $V_{\text{helpers}} = \{i^*, g^*\}$ be sets of variables such that $V_{\text{sel}} \cap V = \emptyset$ and $V_{\text{helpers}} \cap V = \emptyset$. The set of actions $A_{\text{sel}}$ contains now for each $i \in [\log t]$ two actions

$select_i^0$ and $select_i^1$ for which $\mathsf{pre}(select_i^0)[sel_i^1] = 0$ and $\mathsf{pre}(select_i^1)[sel_i^0] = 0$. In addition, we set $\mathsf{eff}(select_i^0)[sel_i^0] = 1$ and $\mathsf{eff}(select_i^1)[sel_i^1] = 1$.

Let $b(i)$ be a bijection from $[t]$ into $\{0,1\}^{\log t}$. For $i \in [t]$ and $j \in [\log t]$, we denote by $b(i)[j]$ the projection to the j-th coordinate of $b(i)$. For example if $b(x) = 1001$, $b(x)[3] = 0$ and $b(x)[4] = 1$.

We can now create for $i \in [t]$ the set $A_i'$ as follows: $A_i' \coloneqq \{a' \mid a \in A_i\}$. For each $i \in [t]$, each $a' \in A_i'$, and each $j \in [\log t]$ we additionally require in the precondition that $\mathsf{pre}(a')[sel_j^{b(i)[j]}] = 1$ and that $\mathsf{pre}(a')[i^*] = 1$. This construction now allows to select an instance $\mathbb{P}_i$ (represented by the bit-vector $b(i)$) by executing the appropriate sequence of actions $A_{\text{sel}}$ of length $\log t$. With this construction we ensure that only the actions of the selected instance $A_i'$ will eventually become applicable since the actions of all other instances are blocked by the preconditions added above. Moreover, we ensure that the initial state was correctly initialized, which can be done as follows.

We introduce two sets of actions $A_{\text{init}} = \{init_1, \ldots, init_t\}$ and $A_{\text{goal}} = \{goal_1, \ldots, goal_t\}$ with the following properties. First, for each $i \in [t]$, each $j \in [\log t]$, and each $v \in V$ we set $\mathsf{pre}(init_i)[sel_j^{b(i)[j]}] = 1$, $\mathsf{eff}(init_i)[v] = I_i[v]$, $\mathsf{pre}(init_i)[i^*] = 0$, and $\mathsf{eff}(init_i)[i^*] = 1$. Second,for each $i \in [t]$, each $j \in [\log t]$, and each $v \in V$ we set $\mathsf{pre}(goal_i)[sel_j^{b(i)[j]}] = 1$, $\mathsf{pre}(goal_i)[v] = G_i[v]$, $\mathsf{pre}(goal_i)[i^*] = 1$, and $\mathsf{eff}(goal_i)[g^*] = 1$.

The composed planning instance is now given by $(\mathbb{P}, |V'|)$, where $\mathbb{P} = \langle V', D, A', I', G' \rangle$, where $V' = (V \cup V_{\text{sel}} \cup V_{\text{helpers}})$, $A' = (\bigcup_1^t A_i' \cup A_{\text{init}} \cup A_{\text{goal}})$, $I' = 0^{|V|+\log t+2}$, and $G'[g^*] = 1$. Notice that the parameter remains polynomial as $|V'| = |V| + \log t + 2$. Therefore, we have also obtained a composition algorithm for the second case (Case 2).

It follows from Proposition 18 that this problem does not admit a polynomial bi-kernel (and hence no polynomial kernel) unless $\mathsf{coNP} \subseteq \mathsf{NP}/\text{poly}$. ☐

**Theorem 16** (*). *Neither $c$-*Eval *nor $c$-*Bounded Eval *with bounded domain admit a polynomial kernel unless* $\mathsf{coNP} \subseteq \mathsf{NP}/\text{poly}$. *This even holds for $c$-*Eval *without goals on components and mixed actions.*

*Proof.* Observe that by putting all variables in the variable-deletion backdoor set, we can polynomially bound the backdoor size by the number of variables. Hence, as long as the domain size is bounded we can perform a simple PPT from $\text{SAS}^+$ Planning planning to $c$-Eval, which due to Lemma 19 and Proposition 17 implies that Eval for instance with bounded domain does not have a polynomial kernel unless $\mathsf{coNP} \subseteq \mathsf{NP}/\text{poly}$. In addition, notice that in the construction of the previous proof the plan length can be bounded by $k_{\text{max}} + \log t + 2$, where $k_{\text{max}}$ denotes the maximum bound on the plan length over all instances in the sequence $(\mathbb{P}_1, k_1, |V_1|), \ldots, (\mathbb{P}_t, k_t, |V_t|)$.

Since all variables are contained in the backdoor there are no components. Thus,

(i) the size of the goal is bounded by the parameter,
(ii) the goal is defined only over variables in the backdoor, and
(iii) there are no mixed actions.

For this reason, the result also holds for $c$-Eval that have neither goals on components nor mixed actions and thus in particular for $c$-Eval when additionally parameterized by the size of the goal and $c$-Eval on instances with no mixed actions. ☐