Tight Complexity Bounds for Counting Generalized Dominating Sets in Bounded-Treewidth Graphs

Abstract

We investigate how efficiently a well-studied family of domination-type problems can be solved on bounded-treewidth graphs. For sets σ, ρ of non-negative integers, a (σ,ρ)-set of a graph G is a set S of vertices such that $|N(u) \cap S| \in \sigma$ for every $u \in S$, and $|N(v) \cap S| \in \rho$ for every $v \notin S$. The problem of finding a (σ,ρ)-set (of a certain size) unifies standard problems such as Independent Set, Dominating Set, Independent Dominating Set, and many others.

For almost all pairs of finite or cofinite sets (σ,ρ), we determine (under standard complexity assumptions) the best possible value $c_{\sigma,\rho}$ such that there is an algorithm that counts (σ,ρ)-sets in time $c_{\sigma,\rho}^{\mathsf{tw}} \cdot n^{O(1)}$ (if a tree decomposition of width tw is given in the input). Let s_{top} denote the largest element of σ if σ is finite, or the largest missing integer $+1$ if σ is cofinite; r_{top} is defined analogously for ρ. Surprisingly, $c_{\sigma,\rho}$ is often significantly smaller than the natural bound $s_{\text{top}} + r_{\text{top}} + 2$ achieved by existing algorithms [van Rooij, 2020]. Toward defining $c_{\sigma,\rho}$, we say that (σ,ρ) is m-structured if there is a pair (α,β) such that every integer in σ equals α mod m, and every integer in ρ equals β mod m. Then, setting

- $c_{\sigma,\rho} = \max\{s_{\text{top}}, r_{\text{top}}\} + 1$ if (σ,ρ) is m-structured for some $\text{m} \geq 3$, or 2-structured with $s_{\text{top}} \neq r_{\text{top}}$, or 2-structured with $s_{\text{top}} = r_{\text{top}}$ being odd,
- $c_{\sigma,\rho} = \max\{s_{\text{top}}, r_{\text{top}}\} + 2$ if (σ,ρ) is 2-structured, but not m-structured for any $\text{m} \geq 3$, and $s_{\text{top}} = r_{\text{top}}$ is even, and
- $c_{\sigma,\rho} = s_{\text{top}} + r_{\text{top}} + 2$ if (σ,ρ) is not m-structured for any $\text{m} \geq 2$,

we provide algorithms counting (σ,ρ)-sets in time $c_{\sigma,\rho}^{\mathsf{tw}} \cdot n^{O(1)}$. For example, for the Exact Independent Dominating Set problem (also known as Perfect Code) corresponding to $\sigma = \{0\}$ and $\rho = \{1\}$, this improves the $3^{\mathsf{tw}} \cdot n^{O(1)}$ algorithm of van Rooij to $2^{\mathsf{tw}} \cdot n^{O(1)}$.

Despite the unusually delicate definition of $c_{\sigma,\rho}$, we show that our algorithms are most likely optimal, i.e., for any pair (σ,ρ) of finite or cofinite sets where the problem is non-trivial (except those having cofinite σ with $\rho = \mathbb{Z}_{\geq 0}$), and any $\varepsilon > 0$, a $(c_{\sigma,\rho} - \varepsilon)^{\mathsf{tw}} \cdot n^{O(1)}$-algorithm counting the number of (σ,ρ)-sets would violate the Counting Strong Exponential-Time Hypothesis (#SETH). For finite sets σ and ρ, our lower bounds also extend to the decision version, showing that our algorithms are optimal in this setting as well. In contrast, for many cofinite sets, we show that further significant improvements for the decision and optimization versions are possible using the technique of representative sets.

Contents

1 Introduction **1**

2 Technical Overview **4**
2.1 Faster Algorithms 4
2.2 Lower Bounds 9

3 Preliminaries **15**
3.1 Basics 15
3.2 Generalized Dominating Sets 17

I Faster Algorithms **20**

4 Faster Algorithms for Structured Pairs **20**
4.1 Structural Insights into the m-Structured Case 20
4.2 Exploiting Structure: Fast Join Operations 29
4.3 Faster Algorithms for Generalized Dominating Set Problems 37

5 Faster Algorithms via Representative Sets **41**

II Lower Bounds **48**

6 High-level Constructions for Proving Lower Bounds for GenDomSet **48**
6.1 Decision Problem 50
6.2 Counting Problem 51

7 Constructing Providers **52**
7.1 Providers Having Either σ-States or ρ-States 53
7.2 Providers Having σ-States Together with ρ-States 55

8 Constructing Managers **62**
8.1 Proof of Lemma 8.4: A Blueprint for Managers 65

9 Lower Bound for the Problem with Relations **69**
9.1 Decision Problem 69
9.2 Counting Problem 78

10 Realizing Relations **82**
10.1 Decision Problem 82
10.2 Counting Problem 89

1 Introduction

Since treewidth was defined independently in multiple equivalent ways in the 70s [7, 30, 44], algorithms on bounded-treewidth graphs have been investigated for decades from different viewpoints. It was observed already in the 80s that many of the basic NP-hard problems can be solved efficiently on bounded-treewidth graphs using a dynamic programming approach [3, 6, 9]. Courcelle's Theorem [16] formalized this observation for a large class of algorithmic problems definable in monadic second-order logic. Algorithms on bounded-treewidth graphs were studied not only for their own sake, but also because they served as useful tools in other algorithms, most notably for planar problems and problems in the parameterized setting [4, 12, 20–22, 34].

Over the years, the focus shifted to trying to make the algorithms as efficient as possible. For example, given a tree decomposition of width tw, the Dominating Set problem can be solved in time $3^{\mathsf{tw}} \cdot n^{O(1)}$ using dynamic programming and subset convolution, but this running time was achieved only after multiple rounds of improvements [1, 8, 46, 50]. The search for more efficient algorithms is complemented by conditional lower bounds showing that certain forms of running times are the best possible. For problems that can be solved in time $c^{\mathsf{tw}} \cdot n^{O(1)}$, a line of research started by Lokshtanov, Marx, and Saurabh [36] gives tight lower bounds on the best possible c appearing in the running time [11, 18, 19, 23, 24, 33, 37, 41, 42]. For example, Lokshtanov et al. [36] showed that $3^{\mathsf{tw}} \cdot n^{O(1)}$ is probably optimal for Dominating Set: assuming the Strong Exponential-Time Hypothesis (SETH), there is no algorithm for Dominating Set that solves the problem in time $(3-\varepsilon)^{\mathsf{tw}} \cdot n^{O(1)}$ for some $\varepsilon > 0$ if given a graph with a tree decomposition of width tw. The goal of this paper is to try to prove similar tight bounds for a class of generalized domination problems. Our findings show that, despite decades of intensive research, even very simple problems are poorly understood, and one can find surprises even when looking at the simplest of problems.

Telle [45] introduced the notion of (σ,ρ)-sets as a common generalization of independent sets and dominating sets. For sets σ, ρ of non-negative integers, a (σ,ρ)-set of a graph G is a set S of vertices such that $|N(u) \cap S| \in \sigma$ for every $u \in S$, and $|N(v) \cap S| \in \rho$ for every $v \notin S$. With different choices of σ and ρ, the problem of finding a (σ,ρ)-set (of a certain size) can express various well-studied algorithmic problems:

- $\sigma = \{0\}$, $\rho = \{0,1,\ldots\}$
  Independent Set: find a set S of k vertices that are pairwise non-adjacent.
- $\sigma = \{0\}$, $\rho = \{0,1\}$
  Strong Independent Set: find a set S of k vertices that are pairwise at distance at least 2.
- $\sigma = \{0,1,\ldots\}$, $\rho = \{1,2,\ldots\}$
  Dominating Set: find a set S of k vertices such that every remaining vertex has a neighbor in S.
- $\sigma = \{0\}$, $\rho = \{1,2,\ldots\}$
  Independent Dominating Set: find an independent set S of k vertices such that every remaining vertex has a neighbor in S.
- $\sigma = \{0\}$, $\rho = \{1\}$
  Exact Independent Dominating Set/Perfect Code: find an independent set S of k vertices such that every remaining vertex has *exactly one* neighbor in S.

- $\sigma = \{1, 2, \ldots\}$, $\rho = \{1, 2, \ldots\}$
  Total Dominating Set: find a set S of k vertices such that every vertex in the graph has *at least one* neighbor in S.

- $\sigma = \{0, 1, \ldots\}$, $\rho = \{1\}$
  Perfect Dominating Set: find a set S of k vertices such that every remaining vertex has *exactly one* neighbor in S.

- $\sigma = \{0, 1 \ldots, d\}$, $\rho = \{0, 1, \ldots\}$
  Induced Bounded-Degree Subgraph: find a set S of k vertices that have *at most* d neighbors in S.

- $\sigma = \{d\}$, $\rho = \{0, 1, \ldots\}$
  Induced d-Regular Subgraph: find a set S of k vertices that have *exactly* d neighbors in S.

Problems related to finding (σ, ρ)-sets received significant attention both from the complexity viewpoint and for demonstrating the robustness of algorithmic techniques [13–15, 25, 26, 29, 31, 32, 47–49]. Some authors call these types of problems *locally checkable vertex subset problems* (LC-VSP).

For the case when each of σ and ρ is finite or cofinite, van Rooij [48] presented a general technique for finding (σ, ρ)-sets on graphs of bounded treewidth. For a set σ of finite or cofinite integers, we write s_{top} to denote the maximum element of σ if σ is finite, and the maximum missing integer $+1$ if σ is cofinite; r_{top} is defined analogously based on ρ. When one tries to solve a problem in time $f(\mathsf{tw}) \cdot n^{O(1)}$ on a graph with a given tree decomposition of width tw, and the goal is to make the function $f(\mathsf{tw})$ as slowly growing as possible, then typically there are two main bottlenecks: the number of subproblems in the dynamic programming, and the efficient handling of join nodes. It was observed by van Rooij [48] that when we consider the problem of finding a (σ, ρ)-set, then each vertex in a partial solution has essentially $s_{\text{top}} + r_{\text{top}} + 2$ states. For example, if a vertex is unselected in a partial solution and ρ is finite, then we need to distinguish between having exactly $0, 1, \ldots, r_{\text{top}}$ neighbors in the partial solution, yielding $r_{\text{top}} + 1$ possibilities. If ρ is cofinite, then when need to distinguish between having exactly $0, 1, \ldots, r_{\text{top}} - 1$, or at least r_{top} neighbors (again $r_{\text{top}} + 1$ possibilities). In a similar way, a selected vertex has $s_{\text{top}} + 1$ different states, giving a total number of $s_{\text{top}} + r_{\text{top}} + 2$ states for each vertex. This suggests that we need to consider about $(s_{\text{top}} + r_{\text{top}} + 2)^{\mathsf{tw}}$ different subproblems at each node of the tree decomposition. Furthermore, van Rooij [48] showed that all these subproblems can be solved in time $(s_{\text{top}} + r_{\text{top}} + 2)^{\mathsf{tw}} \cdot n^{O(1)}$ by using a fast generalized convolution algorithm in each step. The algorithm can be extended to require a specific size for the set S, thus, allowing us to solve minimization/maximization problems or to count the number of solutions.

Theorem 1.1 (van Rooij [48]). *Let σ and ρ be two finite or cofinite sets. Given a graph G with a tree decomposition of width tw and an integer k, we can count the number of (σ, ρ)-sets of size exactly k in time $(s_{\text{top}} + r_{\text{top}} + 2)^{\mathsf{tw}} \cdot n^{O(1)}$.*

Is the upper bound in Theorem 1.1 optimal for every pair (σ, ρ)? In our first main result, we show that there are pairs (σ, ρ) for which $(s_{\text{top}} + r_{\text{top}} + 2)^{\mathsf{tw}}$ overstates the number of possible subproblems that we need to consider at each step of the dynamic programming algorithm. Together with efficient convolution techniques that we develop for this problem, it follows that there are pairs (σ, ρ) for which the $(s_{\text{top}} + r_{\text{top}} + 2)^{\mathsf{tw}} \cdot n^{O(1)}$ algorithm is not optimal and can be improved.

To be more specific, we say that (σ, ρ) is m-*structured* if there is a pair (α, β) such that every integer in σ is exactly α mod m, and every integer in ρ is exactly β mod m. For example, the pairs $(\{0,3\},\{3\})$ and $(\{0,3\},\{1,4\})$ are both 3-structured, but the pair $(\{0,3\},\{3,4\})$ is not m-structured for any $m \geq 2$. Notice that if a set is cofinite, then it cannot be m-structured for any $m \geq 2$. Furthermore, if $|\sigma| = |\rho| = 1$, then (σ, ρ) is m-structured for every m.

Definition 1.2. *Let σ and ρ be two finite or cofinite sets of non-negative integers. We define $c_{\sigma,\rho}$ by setting*

- $c_{\sigma,\rho} = \max\{s_{\text{top}}, r_{\text{top}}\} + 1$ *if (σ, ρ) is* m*-structured for some $m \geq 3$, or 2-structured with $s_{\text{top}} \neq r_{\text{top}}$, or 2-structured with $s_{\text{top}} = r_{\text{top}}$ being odd,*
- $c_{\sigma,\rho} = \max\{s_{\text{top}}, r_{\text{top}}\} + 2$ *if (σ, ρ) is 2-structured, but not* m*-structured for any $m \geq 3$, and $s_{\text{top}} = r_{\text{top}}$ is even, and*
- $c_{\sigma,\rho} = s_{\text{top}} + r_{\text{top}} + 2$ *if (σ, ρ) is not* m*-structured for any $m \geq 2$.*

For example, $c_{\{0,3\},\{3\}} = 4$, $c_{\{0,3\},\{1,4\}} = 5$, $c_{\{1,3\},\{4\}} = 5$, and $c_{\{2,4\},\{4\}} = 6$. Our main observation is that we need to consider only roughly $(c_{\sigma,\rho})^{\mathsf{tw}}$ subproblems at each step of the dynamic programming algorithm: if (σ, ρ) is m-structured, then parity/linear algebra type of arguments show that many of the subproblems cannot be solved.

Theorem 1.3. *Let σ and ρ denote two finite or cofinite sets. Given a graph G with a tree decomposition of width tw and an integer k, we can count the number of (σ, ρ)-sets of size exactly k in time $(c_{\sigma,\rho})^{\mathsf{tw}} \cdot n^{O(1)}$.*

In particular, for Exact Independent Dominating Set (that is, $\sigma = \{0\}$, $\rho = \{1\}$), we have $s_{\text{top}} + r_{\text{top}} + 2 = 3$, while $c_{\sigma,\rho} = 2$ as (σ, ρ) is 3-structured. Therefore, Theorem 1.1 gives a $3^{\mathsf{tw}} \cdot n^{O(1)}$ time algorithm, which we improve to $2^{\mathsf{tw}} \cdot n^{O(1)}$ by Theorem 1.3. This shows that despite the decades-long interest in algorithms for bounded-treewidth graphs, there were new algorithmic ideas to discover even for *literally the simplest* of the non-trivial (σ, ρ)-set problems.

The improvement from $s_{\text{top}} + r_{\text{top}} + 2$ to $c_{\sigma,\rho} = \max\{s_{\text{top}}, r_{\text{top}}\} + 1$ in the base of the exponent can be significant, but one can say that Theorem 1.3 improves upon the algorithm of Theorem 1.1 in this way only in exceptional cases (and there is even an exception to the exception where the improvement is only to $c_{\sigma,\rho} = \max\{s_{\text{top}}, r_{\text{top}}\} + 2$). One may suspect that further improvements are possible, possibly leading to improvements that can be described in a more uniform way. However, we show that the delicate nature of this improvement is not a shortcoming of our algorithmic techniques, but inherent to the problem: for the counting version, $c_{\sigma,\rho}$ *precisely* characterizes the best possible base of the exponent (except when σ is cofinite with $\rho = \mathbb{Z}_{\geq 0}$). For the following lower bound, we need to exclude pairs (σ, ρ) where the problem is trivially solvable: we say that (σ, ρ) is *non-trivial* if $\rho \neq \{0\}$, and $(\sigma, \rho) \neq (\{0, 1, \ldots\}, \{0, 1, \ldots\})$.

Theorem 1.4. *Let (σ, ρ) denote a non-trivial pair of finite or cofinite sets such that $\rho \neq \mathbb{Z}_{\geq 0}$ or σ is finite. If there is an $\varepsilon > 0$ and an algorithm that counts in time $(c_{\sigma,\rho} - \varepsilon)^{\mathsf{tw}} \cdot n^{O(1)}$ the number of (σ, ρ)-sets in a given graph with a given tree decomposition of width tw, then the Counting Strong-Exponential Time Hypothesis (#SETH) fails.*

The algorithm of Theorem 1.3 achieves its running time by considering roughly $(c_{\sigma,\rho})^{\mathsf{tw}}$ subproblems at each node of the tree decomposition. The lower bound of Theorem 1.4 can be interpreted as showing that at least that many subproblems really need to be considered by any algorithm that solves the counting problem. Does this remain true for the decision version as well? For finite σ and ρ this is indeed the case and we obtain a matching lower bound.

Theorem 1.5. *Let (σ, ρ) be a non-trivial pair of finite sets (i.e., non-empty and $0 \notin \rho$). If there is an $\varepsilon > 0$ and an algorithm that decides in time $(c_{\sigma,\rho} - \varepsilon)^{\mathsf{tw}} \cdot n^{O(1)}$ whether there is a (σ, ρ)-set in a given graph with a given tree decomposition of width tw, then the Strong-Exponential Time Hypothesis (SETH) fails.*

Intriguingly, if at least one of σ or ρ is cofinite, the technique of *representative sets* [10, 27, 28, 39, 40, 43] can be used to significantly reduce the number of subproblems that need to be considered at each node. The main idea is that we do not need to solve every subproblem, but rather, we only need a small representative set of partial solutions with the property that if there is a solution to the whole instance, then there is one that extends a partial solution in our representative set. This idea becomes relevant for example when σ or ρ is cofinite with only a few missing integers: then we do not need a collection with every possible type of partial solution, but rather, we only need a collection of partial solutions that can avoid a small list of forbidden degrees at every vertex. We formalize this idea by presenting an algorithm where the base of the exponent does not depend on the largest missing integer in the cofinite set, but depends only on the number of missing integers. We write $\varnothing \neq \tau \subseteq \mathbb{Z}_{\geq 0}$ for a finite or cofinite set. If τ is finite, then we define $\mathrm{cost}(\tau) := \max(\tau)$. Otherwise, τ is cofinite and we define $\mathrm{cost}(\tau) := |\mathbb{Z}_{\geq 0} \setminus \tau|$. Further, let ω denote the matrix multiplication exponent [2].

Theorem 1.6. *Suppose $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ are finite or cofinite. Also, set $t_{\mathrm{cost}} := \max\{\mathrm{cost}(\rho), \mathrm{cost}(\sigma)\}$. Then, there is an algorithm $\mathcal{A}$ that, given a graph G and a nice tree decomposition of G of width tw, decides whether G has a (σ, ρ)-DomSet in time*

$$2^{\mathsf{tw}} \cdot (t_{\mathrm{cost}} + 1)^{\mathsf{tw}(\omega+1)} \cdot (t_{\mathrm{cost}} + \mathsf{tw})^{O(1)} \cdot |V(G)|.$$

While the algorithm of Theorem 1.3 can be significantly more efficient than the algorithm of Theorem 1.6, it is unlikely to be tight in general, and it remains highly unclear what the best possible running time should be. For a tight result, one would need to overcome at least two major challenges: proving tight upper bounds on the size of representative sets, and understanding whether they can be handled without using matrix-multiplication based methods.

2 Technical Overview

2.1 Faster Algorithms

We start with an overview on the techniques used to prove our algorithmic results. First, let us turn to Theorem 1.3. With Theorem 1.1 in mind, it suffices to consider the case where (σ, ρ) is m-structured from some $\mathrm{m} \geq 2$.

2.1.1 Structured Sets

As hinted to earlier, our algorithms are based on the "dynamic programming on tree decompositions" paradigm. Hence, let us first briefly recall the definition of tree decompositions (see Section 3 for more details). A tree decomposition of a graph G consists of a rooted tree T and a bag X_t for every node t of T with the following properties:

1. every vertex v of G appears in at least one bag,
2. for every vertex v of G, the bags containing v correspond to a connected subtree of T, and
3. if two vertices of G are adjacent, then there is at least one bag containing both of them.

The width of a tree decomposition is the size of the largest bag minus one, and the treewidth of a graph G is the smallest possible width of a tree decomposition of G. Further, for our exposition, for a node t of T, we write V_t for the union of all bags $X_{t'}$ where t' is a descendant of t (including t itself).

Let us also recall the most common structure of (dynamic programming) algorithms on tree decompositions (of width tw). Typically, we define suitable subproblems for each node t of the decomposition, and then solve them in a bottom up way. In particular, we construct partial solutions that we aim to extend into full solutions while moving up the tree decomposition. In order to fastly identify which partial solutions can be extended to full solutions, we classify them into a (limited) number of types: if two partial solutions have the same type and one has an extension into a full solution, then the same extension would work for the other solution as well. Now, the subproblems at node t correspond to computing which types of partial solutions are possible. Finally, we need to argue that if we have solved every subproblem for every child t' of t, then the subproblems at t can be solved efficiently as well.

For a more detailed description of how we implement said general approach, first suppose for simplicity that both σ and ρ are finite (in fact, this is always the case when (σ, ρ) is m-structured from some $\mathrm{m} \geq 2$). Now, for us, a partial solution at a node t is a set $S \subseteq V_t$ such that $|N(u) \cap S| \in \sigma$ for every $u \in S \setminus X_t$, and $|N(v) \cap S| \in \rho$ for every $v \in V_t \setminus (S \cup X_t)$, that is, all vertices not in X_t satisfy the σ constraints and ρ constraints, but the vertices in X_t may not. In particular, it may happen that a vertex $v \in X_t \setminus S$ does not yet have a correct number of selected neighbors, that is, $|N(v) \cap S| \notin \rho$, since said vertex may receive additional selected neighbors that lie outside of V_t.

Now, two partial solutions $S_1, S_2 \subseteq V(G)$ have the same type if

1. $X_t \cap S_1 = X_t \cap S_2$, and
2. $|N(v) \cap S_1| = |N(v) \cap S_2|$ for all $v \in X_t$.

Indeed, in this situation, it is easy to verify that, for any $S \subseteq V(G) \setminus V_t$, we have that $S \cup S_1$ is a (σ, ρ)-set if and only if $S \cup S_2$ is a (σ, ρ)-set. In other words, the type of a solution S is determined by specifying, for each $v \in X_t$, whether $v \in S$ and how many selected neighbors v has. Hence, we can describe such a type by a string $y \in \mathbb{A}^{X_t}$, where $\mathbb{A} = \{\sigma_0, \ldots, \sigma_{s_{\mathrm{top}}}, \rho_0, \ldots, \rho_{r_{\mathrm{top}}}\}$, that associates with every $v \in X_t$ a *state* $y[v] \in \mathbb{A}$,

- where state σ_i means that $v \in S$ and v has i selected neighbors, and
- where state ρ_j means that $v \notin S$ and v has j selected neighbors.

Observe that we can immediately dismiss partial solutions that assign too many selected neighbors to a vertex (that is, for instance, a state σ_i for $i > s_{\mathrm{top}}$), since such partial solutions can never be extended to a full solution (assuming σ is finite; for cofinite σ all states σ_i, for $i \geq s_{\mathrm{top}}$, are equivalent). This gives a trivial upper bound of $(s_{\mathrm{top}} + r_{\mathrm{top}} + 2)^{\mathsf{tw}+1}$ for the number of types; and yields essentially the known algorithms.

However, it is highly unclear if said trivial upper bound is tight: is it really possible that for every string $y \in \mathbb{A}^{X_t}$ there is some partial solution S_y that corresponds to y? In general, this turns out to be indeed the case. Consider the classical Dominating Set problem (that is $\rho = \{1, 2, \ldots\}, \sigma = \{0, 1, 2, \ldots\}$) and the following example. Suppose that for some bag X_t, each of its vertices $v_i \in X_t$ has a single neighbor $v_i' \in V_t \setminus X_t$, and suppose that all vertices $v_\star'$ share a common vertex $w \in V_t \setminus X_t$. Consult Figure 2.1 for a visualization of this example.

Now, for any string $y \in \mathbb{A}^{X_t} = \{\rho_0, \rho_1, \sigma_1\}^{X_t}$, there is indeed a partial solution (select w, now any selection of $v_\star'$ or $v_\star$ is a valid partial solution), that is, each string $y \in \mathbb{A}^{X_t}$ is indeed *compatible*

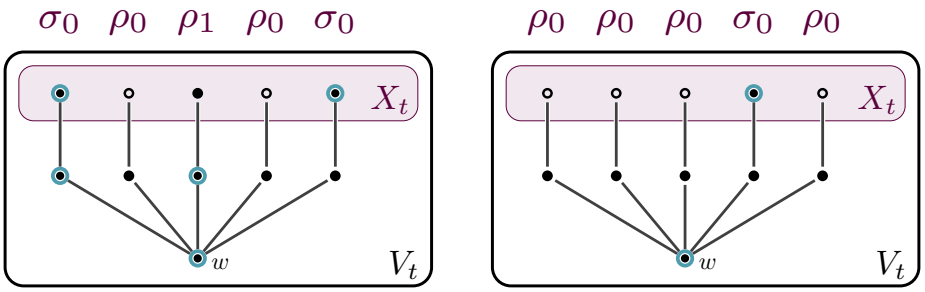


Figure 2.1: For Dominating Set (where $\rho = \{1, 2, \dots\}, \sigma = \{0, 1, 2, \dots\}$), it is easy to construct an example where any string $y \in \mathbb{A}^{X_t} = \{\rho_0, \rho_1, \sigma_1\}^{X_t}$ is compatible with t: we depict selected vertices as encircled blue and unselected vertices without a selected neighbor as a hollow black circle. Observe that after selecting w, any selection of the remaining vertices constitutes a valid partial solution. Hence, there are $3^{|X_t|} = (s_{\text{top}} + r_{\text{top}} + 2)^{|X_t|}$ compatible strings in this case.

with t. In particular, there are $(s_{\text{top}} + r_{\text{top}} + 2)^{|X_t|} = (0 + 1 + 2)^{|X_t|} = 3^{|X_t|}$ strings compatible with t; for $|X_t| = \mathsf{tw}$ the trivial upper bound on the number of types is thus indeed tight.

In stark contrast to this rather unsatisfactory situation, we show that for m-structured (σ, ρ) where $\mathrm{m} \geq 2$, indeed, not all $(s_{\text{top}} + r_{\text{top}} + 2)^{\mathsf{tw}+1}$ different strings (or types) can be compatible with t — we can then exploit this to obtain Theorem 1.3.

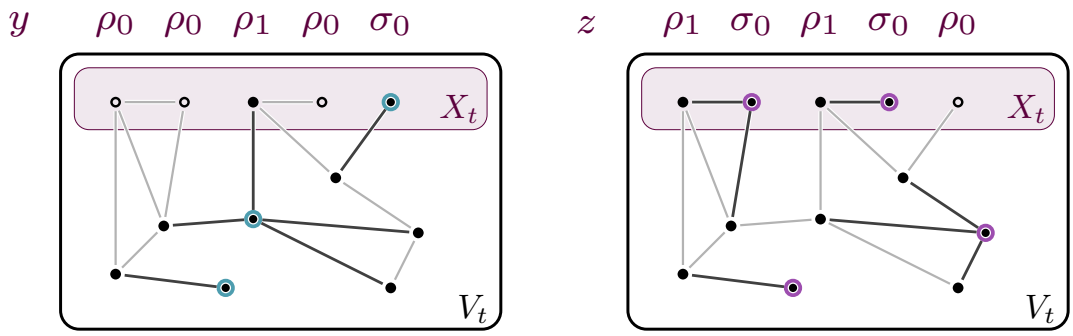
To upper-bound the number of compatible strings in said case, let us first decompose strings $y \in \mathbb{A}^{X_t}$ into a *σ-vector* $\vec{\sigma}(y) \in \{0, 1\}^{X_t}$, defined via $\vec{\sigma}(y)[v] = 1$ if $y[v] = \sigma_c$ for some c, and a *weight vector* $\vec{w}(y) \in \{0, \dots, \max(s_{\text{top}}, r_{\text{top}})\}^{X_t}$, defined via $\vec{w}(y)[v] = c$ if $y[v] \in \{\sigma_c, \rho_c\}$. Now, our main structural insight reads as follows.

Lemma 2.1. *Suppose (σ, ρ) is m-structured (for $\mathrm{m} \geq 2$). Let $y, z \in \mathbb{A}^{X_t}$ denote strings that are compatible with t with witnesses $S_y, S_z \subseteq V_t$ such that $|S_y \setminus X_t| \equiv_{\mathrm{m}} |S_z \setminus X_t|$. Then, $\vec{\sigma}(y) \cdot \vec{w}(z) \equiv_{\mathrm{m}} \vec{\sigma}(z) \cdot \vec{w}(y)$.*

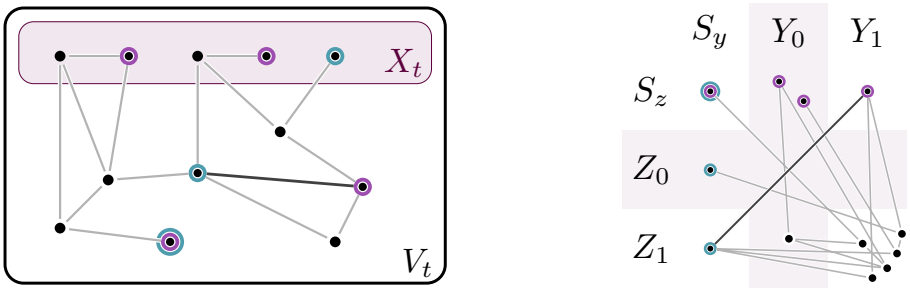
For an intuition for this lemma, let us move to the Exact Independent Dominating Set (or Perfect Code) problem as a specific example which corresponds to $\sigma = \{0\}$ and $\rho = \{1\}$. Observe that $(\{0\}, \{1\})$ is indeed 2-structured. Consider two partial solutions $S_y, S_z \subseteq V_t$. (Also consult Figure 2.2a for a visualization of an example.) Note that both S_y and S_z are independent sets in G since $\sigma = \{0\}$. Now, let us count the edges between S_y and S_z. To that end, let us define Y_0 as the set of vertices from $V_t \setminus S_y$ that have no selected neighbor, and Y_1 as the set of vertices from $V_t \setminus S_y$ that have one selected neighbor (observe that $Y_0 \subseteq X_t$). Observe that (S_y, Y_0, Y_1) forms a partition of V_t. We define Z_0 and Z_1 analogously for the partial solution S_z.

Now, every vertex in $S_z \cap Y_0$ has no neighbor in S_y, while every vertex in $S_z \cap Y_1$ has exactly one neighbor in S_y. Recalling that vertices from $S_z \cap S_y$ have no neighbor in S_y (since S_y is an independent set), the number of edges from S_z to S_y equals $|S_z \cap Y_1|$. Repeating the same argument with reversed roles, we get that the number of edges from S_y to S_z equals $|S_y \cap Z_1|$. So $|S_z \cap Y_1| = |S_y \cap Z_1|$. Assuming $X_t = V_t$, this condition is equivalent to $\vec{\sigma}(y) \cdot \vec{w}(z) = \vec{\sigma}(z) \cdot \vec{w}(y)$. To obtain the conclusion of the lemma also for $X_t \neq V_t$, we use that $|S_y \setminus X_t| \equiv_2 |S_z \setminus X_t|$ and $Y_0, Z_0 \subseteq X_t$. Consult Figure 2.2b for a visualization of said proof sketch for the example from Figure 2.2a.

Now, how can we use Lemma 2.1 to derive bounds on the number of compatible strings $y \in \mathbb{A}^{X_t}$? First, we partition the compatible strings into sets L_i, for $i \in [0 \,..\, m-1]$, where L_i contains all compatible strings y for which there is a partial solution S_y that satisfies $|S_y \setminus X_t| \equiv_m i$. Hence,



(a) A bag X_t and the union V_t of the bags that are not above X_t. For $\rho = \{1\}, \sigma = \{0\}$ (which are 2-structured) the strings $y = \rho_0\rho_0\rho_1\rho_0\sigma_0$ and $z = \rho_1\sigma_0\rho_1\sigma_0\rho_0$ are compatible with t; the vertices of the corresponding partial solutions S_y and S_z are encircled in blue and purple (respectively); we depict unselected vertices without selected neighbors as empty circles. We have $|S_y \setminus X_t| = |S_z \setminus X_t| = 2$ and $\vec{\sigma}(y) \cdot \vec{w}(z) = (0,0,0,0,1) \cdot (1,0,1,0,0) = 0 \equiv_2 0 = (0,1,0,1,0) \cdot (0,0,1,0,0) = \vec{\sigma}(z) \cdot \vec{w}(y)$.



(b) The partial solutions S_y and S_z depicted at once. Observe that edges between a vertex $v_y \in S_y$ and $v_z \in S_z$ are possible only if v_y and v_z have exactly one selected neighbor in the corresponding other partial solution.

Figure 2.2: An example for partial solutions and edges between them for Perfect Code.

Lemma 2.1 yields that $\vec{\sigma}(y) \cdot \vec{w}(z) \equiv_m \vec{\sigma}(z) \cdot \vec{w}(y)$ for all $y, z \in L_i$. We then show that $|L_i| \leq c_{\sigma,\rho}^{|X_t|}$ for all $i \in [0\,..\,m-1]$ by using arguments that have a linear-algebra flavor. For the case $\sigma = \{0\}$ and $\rho = \{1\}$, we for example obtain that $|L_i| \leq 2^{|X_t|}$. On a high level, our intuition here is that the condition $\vec{\sigma}(y) \cdot \vec{w}(z) \equiv_m \vec{\sigma}(z) \cdot \vec{w}(y)$ says that the set of σ-vectors is in some sense "orthogonal" to the set of weight-vectors. More precisely, it turns out that we can partition the set $X_t = A \uplus B$ such that

- A determines the σ-vectors of L_i, that is, fixing all positions $v \in A$ of a σ-vector $\vec{s}$ completely determines $\vec{s}$, and
- B determines the weight-vectors modulo m for every fixed σ-vector, that is, fixing all positions $v \in B$ of a weight-vector that appears together with a fixed σ-vector $\vec{s}$, determines all positions of the weight vector modulo m.

For example, for $\sigma = \{0\}$ and $\rho = \{1\}$, this implies that $|L_i| \leq 2^{|A|} \cdot 2^{|B|} = 2^{|X_t|}$. Indeed, there are $2^{|A|}$ many potential σ-vectors $\vec{s}$, and, for every fixed $\vec{s}$, we have $2^{|B|}$ options for choosing the weight vector modulo 2. Since $\max(s_{\text{top}}, r_{\text{top}}) \leq 1$, determining the weight vector modulo 2 actually completely determines the weight vector, and so the upper bound follows.

For $m \geq 3$, the largest number of types can generally be achieved when $A = \varnothing$, since, intuitively speaking, in comparison to the trivial upper bound, we roughly save a factor of $m^{|A|} \cdot 2^{|B|}$. In this

case, it is easy to see that $|L_i| \leq (\max\{s_{\text{top}}, r_{\text{top}}\}+1)^{|X_t|}$ since we have that many choices for weight vectors. It may be tempting to believe that the same holds for $\mathrm{m} = 2$ (since here, it does not seem to matter how we choose A). However, there is one notable exception when $s_{\text{top}} = r_{\text{top}}$ is even. In this case, the language

$$L^* \coloneqq \{y \in \mathbb{A}^{X_t} \mid \vec{w}(y)[\,v\,] \equiv_{\mathrm{m}} 0 \text{ for all } v \in X_t\}$$

has size $|L^*| = (\lfloor r_{\text{top}}/\mathrm{m} \rfloor + \lfloor s_{\text{top}}/\mathrm{m} \rfloor + 2)^{|X_t|} = (\max\{s_{\text{top}}, r_{\text{top}}\} + 2)^{|X_t|}$ which explains why this case stands out in Definition 1.2. Note that for L^* we need to choose $A = X_t$ and $B = \varnothing$.

Overall, we obtain that the number of compatible strings is bounded by $O(c_{\sigma,\rho}^{|X_t|})$. With our improved bound at hand, we can now obtain improved dynamic programming algorithms.

At this point, however, we face a second challenge. When performing dynamic programming along a tree decomposition, the most expensive step is to perform a join operation. In this situation, the current node t has exactly two children t_1 and t_2 and $X_t = X_{t_1} = X_{t_2}$. In [48], van Rooij provides various convolution-based methods to efficiently compute the set of compatible strings for t given the sets of compatible strings for t_1 and t_2. We wish to apply the same methods, but unfortunately this is not directly possible since the convolution-based methods are not designed to handle restrictions of the input space. To circumvent this issue, our solution is to design a specialized compression method that again exploits the structure described above. With the partition (A, B) of X_t at hand, this seems rather straightforward by simply omitting the redundant information. However, the crux is that we need to design the compression in such a way that it agrees with the join operation: two compatible strings y_1 and y_2 for t_1 and t_2 can be joined into a compatible string y for t if and only if the compressed strings y_1' and y_2' can be joined into the compressed string y'. This condition makes the compression surprisingly tricky, and here we need to add several "checksums" to the compressed strings to ensure the required equivalence.

2.1.2 Representative Sets

The algorithm described above determined, for every node t and string $y \in \mathbb{A}^{X_t}$, whether there is a partial solution $S \subseteq V_t$ that has type y. Our lower bounds show that this strategy is essentially optimal when we want to count the number of solutions. But, for the decision version, the idea of representative sets can give significant improvements in some cases.

As an illustrative example, let us consider the problem with $\sigma = \{0\}$, $\rho = \mathbb{Z}_{\geq 0} \setminus \{10\}$, and suppose that $X_t = \{v\}$. Then, the partial solutions $S \subseteq V_t$ have $r_{\text{top}} + s_{\text{top}} + 2 = 13$ different types: either $v \in S$ has 0 neighbors in S, or $v \notin S$ and v has $0, 1, \ldots, 10$, or ≥ 11 neighbors in S. However, we do not need to know exactly which of these types correspond to partial solutions. A smaller amount of information is already sufficient to decide if there is a partial solution that is compatible with some extension $S' \subseteq V(G) \setminus V_t$. For example, if we have partial solutions that correspond to, say, the σ_0, ρ_7, and ρ_8 states, then *every* extension S' that extends *some* partial solution $S \subseteq V_t$ extends one of these three partial solutions. Indeed, suppose that $v \notin S$ and extension S' gives i further neighbors to v, then S can be replaced by a partial solution corresponding to the ρ_7 state unless $i = 3$, in which case S can be replaced by the solution corresponding to ρ_8.

In general, we want to compute a *representative set* of all the partial solutions of V_t such that if there is one partial solution that is extended by some set $S' \subseteq V(G) \setminus V_t$, then there is at least one partial solution in the representative set that is extendable by S'. When $|X_t| > 1$, then it is far from trivial to obtain upper bounds on the size of the required representative sets and to compute them efficiently. Earlier work [38] showed a connection between these type of representative sets and representative sets in linear matroids, and hence, known algebraic techniques can be used [27, 28, 35]. The upper bounds depend on the number of missing integers from the cofinite sets,

but *do not* depend on the largest missing element. Thus, this technique is particularly efficient when a few large integers are missing from ρ or σ. However, the price we have to pay is that the algebraic techniques require matrix operations and the matrix multiplication exponent ω appears in the exponent of the running time. This makes it unlikely that we can get tight upper bounds similar to Theorem 1.4.

2.2 Lower Bounds

Next, we give an overview of the techniques used to prove our lower bound results.

2.2.1 Hardness for Relation Problems

For the lower bounds, we follow the ideas of previous lower bounds [18, 36, 37] given for problems parameterized by treewidth. We present a reduction from SAT to (σ, ρ)-DomSet by constructing a graph which has "small" treewidth. Instead of giving a direct reduction, we split it in two parts. The first part shows a lower bound for the intermediate problem (σ, ρ)-Dominating Set w. Relations ((σ, ρ)-DomSet$^{\text{Rel}}$) which extends the (σ, ρ)-DomSet problem by the possibility of having special (complex) vertices to which a relation is assigned. These vertices are never selected and instead of requiring the number of selected neighbors to be in ρ, they enforce a relation on their neighborhood. The second step then considers the problem of removing these relations. For now we just consider the variant with relations, i.e., (σ, ρ)-DomSet$^{\text{Rel}}$.

Naive and Improved Construction. We first recap the naive approach and its downsides. There the graph of the (σ, ρ)-DomSet$^{\text{Rel}}$ instance has a grid-like structure with one column for each clause of the SAT formula, and one row for each variable of the formula, where each edge in each row is subdivided by an *information vertex*. The crossing points of the grid are *relations* which check whether the corresponding clause is satisfied by the assignments which is encoded by the selection status of the information vertex. This information is then propagated to the relation in the next row. As the assignment is only encoded by two states of the information vertices (selected or not), this approach could only give a lower bound of $(2-\varepsilon)^{\mathsf{tw}}$.

To obtain better lower bounds of the form $(|\mathbb{A}|-\varepsilon)^{\mathsf{tw}}$, we need more states for the information vertices than only "selected" or "unselected", which we may write as σ_0 and ρ_0. Especially, we need to be able to add neighbors to these information vertices from the left, but also from the right side. Instead of using only the states σ_0 and ρ_0, we want to use the remaining states in $\mathbb{A}$ as well. We later explain why this is actually not always possible.

Before we describe it in more detail, see Figure 2.3 for an illustration of the improved construction. Assuming we can use all these states, we partition the variables into $n/\log(|\mathbb{A}|)$ groups of size $\log(|\mathbb{A}|)$ each.[1] By this choice, there is a one-to-one correspondence between assignments to each group and the states in $\mathbb{A}$. Similarly to the naive approach, the constructed graph has a grid-like structure with m columns, each corresponding to one clause of the SAT formula, but only $n/\log(|\mathbb{A}|)$ many rows, each corresponding to one group of variables. Moreover, at the crossing points of rows and columns, we have relations which check that the assignment satisfies the clause. The most notable difference is the neighborhood of the information vertices which we place between two crossing points in the grid. Let w_i^j be the information vertex with the crossing between the ith row and jth column to its left and the crossing between the ith row and $j+1$th column to its right. Each such information vertex gets $\max\{s_{\text{top}}, r_{\text{top}}\}$ neighbors on the left and the same number of

[1] For ease of presentation, we ignore rounding and parity issues here, which can be resolved by partitioning into a somewhat smaller number of somewhat larger groups.

neighbors to the right. The state of an information vertex is determined by its selection status and the number of selected neighbors to its left. As mentioned above, this state encodes the assignment for the corresponding group of variables. In each column, we connect two relations exactly when they are assigned to two neighboring crossing points, that is, they share an edge between them.

For the basic idea, it suffices to think of σ, ρ as being finite and just containing one element. For example, let $\sigma = \{4\}$ and $\rho = \{3\}$. Whenever the information vertex w_i^j is selected and has ℓ selected neighbors to the left, it must have $s_{\text{top}} - \ell = 4 - \ell$ selected neighbors on the right because of the constraints imposed by σ. The relation at crossing point $(i, j+1)$ is defined such that the next information vertex w_i^{j+1} gets ℓ selected neighbors from the left. By defining all relations which we assign to crossing points in this way, all information vertices of one row get the same number of neighbors from the left. Hence, they can encode the same assignment. The same arguments work for the case when w_i^j is not selected, but then w_i^j must have $r_{\text{top}} - \ell = 3 - \ell$ neighbors to the right if it has ℓ neighbors to the left. As the behavior of the relation depends on the information vertices, they are additionally connected to the information vertices and not only their neighbors toward the relation. To ensure that clauses that are initially unsatisfied are eventually satisfied, we add corresponding relations to the first and last row of the graph.

Observe that the information vertices of one column cut the graph in two halves. Hence, the treewidth of the graph can be bounded by the number of rows plus some constant which takes care of the relations and other vertices. Moreover, when we cut the graph at the information vertices of one column, we get a direct correspondence to the results from the upper bound. Indeed, the states the information vertices might get in a solution precisely correspond to the strings in the language compatible with the graph. Consequently, for each such set of information vertices, the number of states they can have is trivially upper-bounded by $|\mathbb{A}|^{\mathsf{tw}+1}$. However, as we have seen in Section 2.1, the number of states that can actually appear is often much smaller (i.e., when (σ, ρ) is m-structured). In such a case, we can only hand over states from a suitable subset $A \subseteq \mathbb{A}$ to the information vertices, and we get a weaker lower bound of only $(|A| - \varepsilon)^{\mathsf{tw}}$. The entire reduction is presented in Section 9 and stated formally in Lemma 6.5 for the decision version. Afterwards, we extend the results to the counting version of the problem, denoted by (σ, ρ)-#DomSet$^{\text{Rel}}$, in Lemma 6.8.

Difficulties in the Construction. Observe that the information vertices are always happy; if there are ℓ selected vertices on the left, then we can select $s_{\text{top}} - \ell$ or $r_{\text{top}} - \ell$ vertices on the right depending on whether the information vertex is selected or not, respectively. However, the previous construction ignores a very subtle but surprisingly crucial issue; the neighbors of the information vertices also have to be happy, that is, they must get a feasible number of selected neighbors to satisfy the σ- or ρ-constraints. It turns out that this is not always possible while simultaneously being able to give all states to the information vertices. Instead, depending on (σ, ρ), as we already indicated above, we have to restrict the states for the information vertices to be able to make their neighbors happy. Note that this property is crucially needed as otherwise no solution exists. When proving the lower bound, we define, for each case mentioned in Definition 1.2, a *manager gadget* by which we achieve the mentioned bound. The formal introduction of these gadgets and their construction is given in Section 8.

Handling General Cofinite Sets. The basic idea of the hardness result for (σ, ρ)-DomSet$^{\text{Rel}}$ is easiest to understand when each of σ and ρ contains only a single integer. Then, an information vertex really transfers information: its selection status and the number of neighbors on the left uniquely determine the number of neighbors on the right. We can interpret a selected vertex

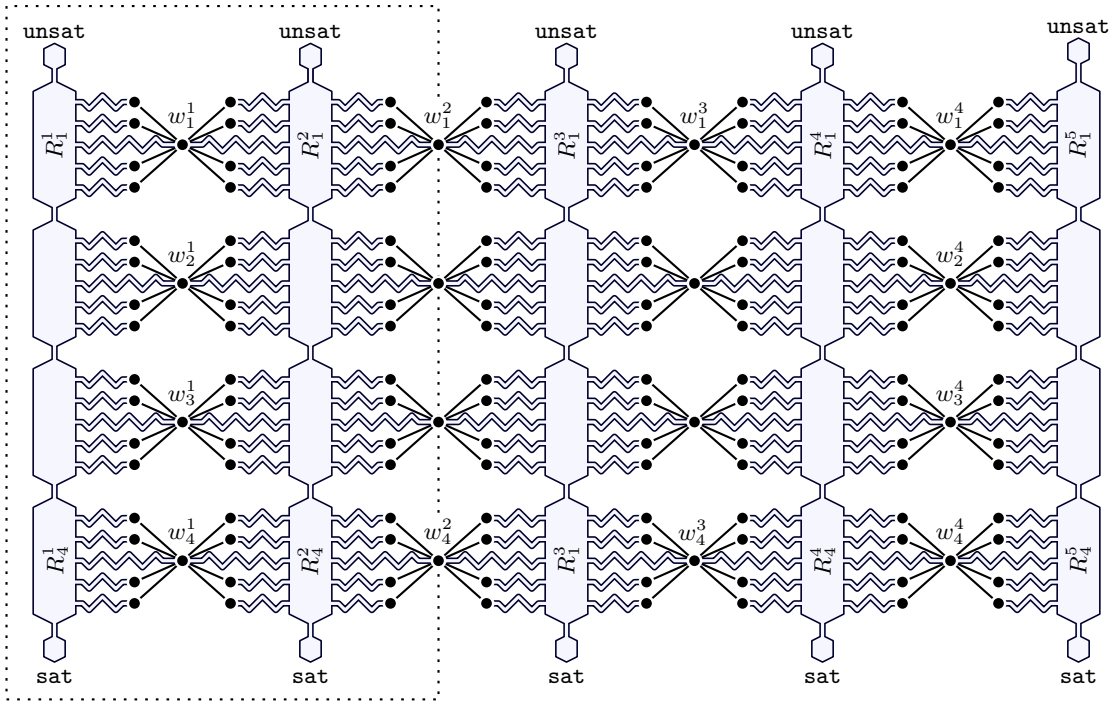


Figure 2.3: The figure illustrates the construction for the lower bound given a SAT formula with five clauses, depicted as columns, and variables which are grouped into four groups, depicted as rows. The relations R_i^j check if the assignment for the ith group satisfies the jth clause.

as a requirement "$= s_{\text{top}}$" on the number of neighbors. With some changes, the argument can be made to work with arbitrary finite σ: such a set enforces a requirement *stronger* than "$\leq s_{\text{top}}$". Similarly, if σ is simple cofinite, that is, $\sigma = \{s_{\text{top}}, \dots, \}$, then σ enforces *exactly* "$\geq s_{\text{top}}$". However, an arbitrary cofinite set enforces a requirement *weaker* than "$\geq s_{\text{top}}$", which is not useful for our purposes.

We handle general cofinite sets by presenting a reduction from simple cofinite sets. In particular, suppose that σ is cofinite; then we reduce from the case where $\sigma' = \{s_{\text{top}}, \dots\}$. As we shall see, the reduction relies heavily on the counting nature of the problem. Given a graph G, the reduction creates a graph G' by attaching to each vertex v a certain gadget with the following properties:

- If v is unselected, then the gadget has a unique extension, which does not give any additional neighbors to v.
- If v is selected, then every extension of the gadget provides at most s_{top} new neighbors to v.
- If v is selected, then the gadget has d_i extensions that provide i new neighbors to v.

Given the numbers $d_0, \dots, d_{s_{\text{top}}}$, it is not very difficult to use the relations to construct a gadget with exactly these types and number of extensions.

Let us consider $\sigma = \{1, 3, 4, 6, 7, \dots\}$ as an illustrative example, and suppose for simplicity that we attached such a gadget only to a single vertex v. Consider now partial solutions of the original graph G where every vertex satisfies the requirements, except potentially v. Suppose that v has 0 neighbors in a partial solution; it needs 1, 3, 4, or at least 6 further neighbors to satisfy the σ constraint. Therefore, the gadget has exactly $d_1 + d_3 + d_4 + d_6$ extensions where the degree of v becomes a member of σ. Similarly, if v already has one neighbor in the partial solution, then it

needs 0, 2, 3, or *at least* 5 neighbors and thus, the gadget has exactly $d_0+d_2+d_3+d_5+d_6$ extensions where v satisfies the degree condition, and so on.

We would like to build a gadget where the integers d_i are chosen in such a way that the gadget has exactly 1 (or some other constant) extension if v already has at least 6 neighbors, and 0 extensions if v has less than 6 neighbors. If we can build such a gadget, then we can effectively force that v has at least 6 neighbors. Based on the discussion above, the d_i's have to be chosen such that they satisfy the following system of equations:

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

One can observe that the coefficient matrix has a form that guarantees that it is non-singular: every element on or below the codiagonal is 1, while every element one row above the codiagonal is 0. Therefore, the system of equations has a solution, so we could construct the appropriate gadget. The catch is, however, that some of the d_i's could be negative, making it impossible to have a gadget with exactly that many extensions. We can solve this issue in the following way: if d_i is negative, then we design the gadget to have $2^x \cdot |d_i|$ extensions, where x is some parameter of the construction. If we attach such gadgets to every vertex, then it can be observed that the number of solutions is a polynomial in $y = 2^x$. We can recover the coefficients of this polynomial using interpolation techniques. Then, we evaluate the polynomial at $y = -1$, which then will simulate exactly d_i extensions.

2.2.2 Realizing Relations

The remaining step to complete the lower bound is to remove the relations.

Relations for the Decision Problem. For the decision version, in order to establish Theorem 1.5, we follow previous approaches [18, 37], and first realize very specific relations such as the "Hamming Weight One" relation and the "Equality" relation. In a second step, we use these relations to design graphs which behave exactly like arbitrarily complex relations. The final step for the lower bound in the decision version is to replace the relations by these graphs. As the realization of relations again depends on σ and ρ, we only give the lower bounds when both sets are finite.

Relations for the Counting Problem. To reduce (σ,ρ)-#DomSet$^{\text{Rel}}$ to (σ,ρ)-#DomSet, it will be convenient to introduce a natural generalization of the problem where not necessarily every vertex has the same constraint (σ,ρ). Instead, we have an additional set $\mathcal{P}$ of pairs and the input contains a mapping specifying for every vertex if it is constrained by (σ,ρ) or by some other pair in $\mathcal{P}$. For a fixed set $\mathcal{P}$ of pairs, we denote by (σ,ρ)-#DomSet$^{\mathcal{P}}$ this extension. As the first step of the proof, we show that if

- $(\varnothing,\{1\}) \in \mathcal{P}$,
- $(\varnothing,\{0,1\}) \in \mathcal{P}$, or
- $(\varnothing,\mathbb{Z}_{\geq 1}) \in \mathcal{P}$,

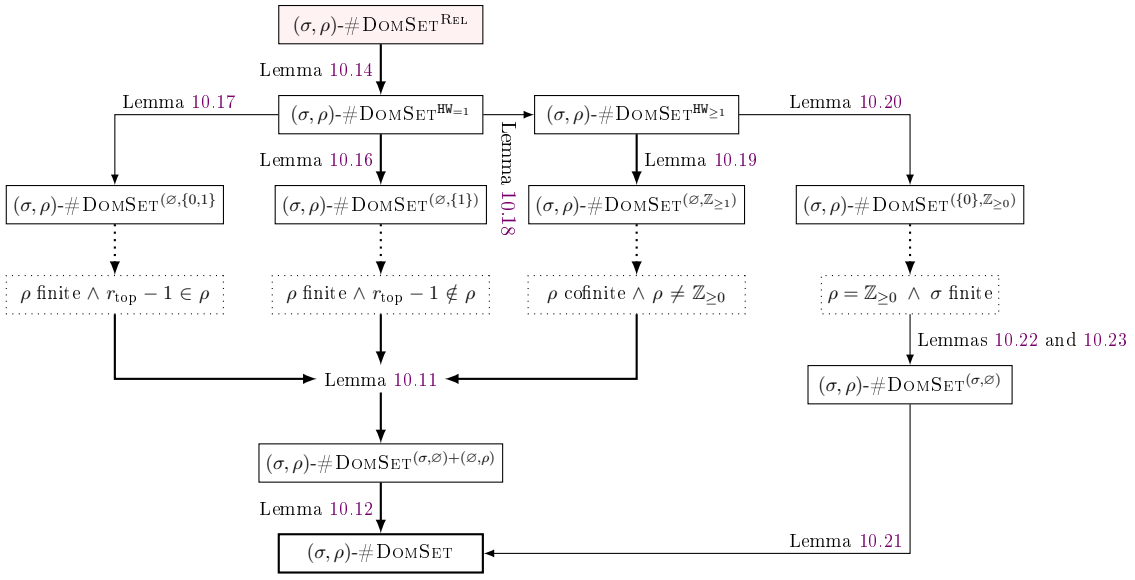


Figure 2.4: The reductions for the counting version to remove the relations. The inital starting problem is (σ,ρ)-#DomSet$^{\text{Rel}}$.

then (σ,ρ)-#DomSet$^{\text{Rel}}$ can be reduced to (σ,ρ)-#DomSet$^{\mathcal{P}}$ in a treewidth-preserving way. The proof utilizes the realization of the relations (as in the decision problem) using "Hamming Weight One" relations, and then simulating this relation using one of the three pairs listed above. Then, the main part of the proof is to show that, for every (σ,ρ) pair we consider, one of the three pairs can be simulated with the (σ,ρ)-#DomSet problem, see Figure 2.4.

Toward this goal, we simulate some number of other pairs. A key intermediate goal is to "force a vertex to be unselected", that is, to simulate an $(\varnothing,\rho)$ vertex. Similarly, another important goal is to "force a vertex to be selected" by simulating a $(\sigma,\varnothing)$ vertex. Now, let us do the following:

- introduce $r_{\text{top}}-1$ cliques, each consisting of $s_{\min}+1$ vertices of type $(\sigma,\varnothing)$
- introduce one vertex v of type $(\varnothing,\rho)$, adjacent to one vertex in each clique.

The vertex v can be considered as an always unselected vertex that already has $r_{\text{top}}-1$ extra neighbors, that is, ρ is "shifted down" by $r_{\text{top}}-1$. Then, depending on whether ρ is cofinite, finite with $r_{\text{top}}-1\in\rho$ or finite with $r_{\text{top}}-1\notin\rho$, the vertex v effectively becomes an $(\varnothing,\mathbb{Z}_{\geq 1})$, $(\varnothing,\{0,1\})$, or $(\varnothing,\{1\})$ vertex, respectively.

In the rest of this section, we highlight some of the technical ideas behind the simulation of $(\varnothing,\rho)$ and $(\sigma,\varnothing)$ vertices. The following notation will be convenient for the description. Let $\mathcal{G}$ be a gadget with a single portal vertex v. We denote by $\text{ext}_{\mathcal{G}}(\rho_i)$ the number of partial solutions S of $\mathcal{G}$ where every vertex except v satisfies the constraints, $v\notin S$, and v has exactly i neighbors in S. The definition of $\text{ext}_{\mathcal{G}}(\sigma_i)$ is analogous with $v\in S$. The proof proceeds by constructing gadgets where the numbers of extensions satisfy certain properties. Let us assume first that σ and ρ are finite.

Step 1. First, we construct $\mathcal{G}_1$ with $\text{ext}_{\mathcal{G}_1}(\rho_0),\text{ext}_{\mathcal{G}_1}(\rho_1)\geq 1$, but $\text{ext}_{\mathcal{G}_1}(\rho_i)=0$ for $i\geq 2$. The construction is easy: the graph consists of a portal vertex v, and two sets of vertices X,Y with a single edge between X and v. All we need to ensure is that both X and Y are solutions, which can be ensured by having degree s_{top} inside X and Y, and degree r_{top} between X and Y.

Step 2. Next, we want to construct a graph $\mathcal{G}_2$ where $\mathrm{ext}_{\mathcal{G}_2}(\rho_0)$ and $\mathrm{ext}_{\mathcal{G}_2}(\sigma_0)$ are both at least one, but different. In other words, the choice of selecting/not selecting the portal v influences the number of extensions in the gadget where the neighbors of the portal are not selected. Intuitively, this should be the case for most graphs, but to find a graph where this is provably so, we argue in the following way. Take x copies of the $\mathcal{G}_1$ gadget defined above, and identify their portal vertices into a single vertex w. The portal of $\mathcal{G}_2$ is a new vertex v adjacent to w. The number $\mathrm{ext}_{\mathcal{G}_2}(\rho_0)$ corresponds to partial solutions S where $v, w \notin S$. Then, each copy of $\mathcal{G}_1$ has $\mathrm{ext}_{\mathcal{G}_1}(\rho_1)$ extensions that add a new neighbor to w and $\mathrm{ext}_{\mathcal{G}_1}(\rho_0)$ extensions that do not. These extra neighbors should satisfy the constraint given by ρ on w. Therefore, we have

$$\mathrm{ext}_{\mathcal{G}_2}(\rho_0) = \sum_{i \in \rho} \binom{x}{i} \mathrm{ext}_{\mathcal{G}_1}(\rho_1)^i \, \mathrm{ext}_{\mathcal{G}_1}(\rho_0)^{x-i}.$$

The expression for $\mathrm{ext}_{\mathcal{G}_2}(\sigma_0)$ is similar, but counts partial solutions containing v, giving an extra selected neighbor to w. Thus,

$$\mathrm{ext}_{\mathcal{G}_2}(\sigma_0) = \sum_{i+1 \in \rho} \binom{x}{i} \mathrm{ext}_{\mathcal{G}_1}(\rho_1)^i \, \mathrm{ext}_{\mathcal{G}_1}(\rho_0)^{x-i}.$$

Observe that dividing both expressions by $\mathrm{ext}_{\mathcal{G}_1}(\rho_0)^x$ gives two polynomials in x, where the first one has degree r_{top}, and the second one $r_{\mathrm{top}} - 1$. Thus, there has to be an x where they are different.

Step 3. The next step is to implement a vertex $(\varnothing, \{0\})$, that is, a vertex that cannot be selected and should not get any neighbors. This can be used to force any number of other vertices to be unselected. We construct a gadget $\mathcal{G}_3$ by creating x copies of $\mathcal{G}_2$ and identifying their portal vertices into a single portal vertex. Suppose that we extend a graph G to a graph G' by identifying the portal of $\mathcal{G}_3$ with a vertex $v \in V(G)$. Let a_d be the number of partial solutions in G where v is unselected and has exactly d selected neighbors; b_d is defined similarly for v being selected. Now the number of solutions in G' is

$$\underbrace{\sum_{d=0}^{r_{\mathrm{top}}} a_d \sum_{i+d \in \rho} \binom{x}{i} \mathrm{ext}_{\mathcal{G}_2}(\rho_1)^i \, \mathrm{ext}_{\mathcal{G}_2}(\rho_0)^{x-i}}_{A:=} + \underbrace{\sum_{d=0}^{s_{\mathrm{top}}} b_d \sum_{i+d \in \rho} \binom{x}{i} \mathrm{ext}_{\mathcal{G}_2}(\sigma_1)^i \, \mathrm{ext}_{\mathcal{G}_2}(\sigma_0)^{x-i}}_{B:=}$$

The important observation here is that $\mathrm{ext}_{\mathcal{G}_2}(\rho_0) \neq \mathrm{ext}_{\mathcal{G}_2}(\sigma_0)$ implies that the two terms A and B above have very different orders of magnitude. Assume that $\mathrm{ext}_{\mathcal{G}_2}(\rho_0) < \mathrm{ext}_{\mathcal{G}_2}(\sigma_0)$ (the other case is similar). Then, A is roughly at most $2^{|V(G)|} \cdot \mathrm{ext}_{\mathcal{G}_2}(\rho_0)^x$, while B is divisible by $\mathrm{ext}_{\mathcal{G}_2}(\sigma_0)^{x-r_{\mathrm{top}}}$, which is much larger for sufficiently large values of x. Therefore, if we compute $A + B$ modulo $\mathrm{ext}_{\mathcal{G}_2}(\sigma_0)^{x-r_{\mathrm{top}}}$, then B vanishes and we can recover A.

Dividing A by $\mathrm{ext}_{\mathcal{G}_2}(\rho_0)^x \neq 0$ gives a polynomial in x of degree r_{top}, whose coefficients we can recover using interpolation. (For simplicity, we assume here that $\mathrm{ext}_{\mathcal{G}_2}(\rho_1) \neq 0$; otherwise, we treat it as a separate case). Observe that only $d = 0$ contributes a term $x^{r_{\mathrm{top}}}$, and thus, from the coefficient of $x^{r_{\mathrm{top}}}$ we can recover the value of a_0. This counts the number of partial solutions where v is not selected and has no neighbors, that is, it simulates an $(\varnothing, \{0\})$ vertex, as we wanted.

Step 4. Once we can express a single $(\varnothing, \{0\})$, attaching it to any set S of vertices forbids selecting any vertex of S without adding any neighbors, effectively making them $(\varnothing, \rho)$ vertices. A further interpolation argument of similar flavor allows us to simulate an arbitrary number of $(\sigma, \varnothing)$ vertices. We omit the details here.

Step 5. There are a number of difficulties with this approach if σ and/or ρ is cofinite. For example, suppose that ρ is cofinite and consider the term A above. As ρ is not finite, A cannot be written as the sum of a constant number of binomial terms. Instead, we can write A as all possible extensions to $\mathcal{G}_3$, minus those extensions that give the wrong number of neighbors to v. That is,

$$A := \sum_{d=0}^{r_{\text{top}}} a_i \left((\text{ext}_{\mathcal{G}_2}(\rho_0) + \text{ext}_{\mathcal{G}_2}(\rho_1))^x - \sum_{i+d \notin \rho} \binom{x}{i} \text{ext}_{\mathcal{G}_2}(\rho_1)^i \, \text{ext}_{\mathcal{G}_2}(\rho_0)^{x-i} \right).$$

Observe that this is a combination of an exponential term and some polynomials. In the cofinite cases, we have to deal with functions of the form $f(x) = c^x - p(x)$, where $p(x)$ is a polynomial. A key technique to handle such expressions is to compute $f(x+1) - c \cdot f(x) = p(x+1) - c \cdot p(x)$, which is a polynomial of the same degree as $p(x)$, and hence, much easier to work with.

Step 6. The case when $\rho = \mathbb{Z}_{\geq 0}$ has to be treated as a special case. First, we establish a reduction from (σ, ρ)-#DomSet$^{\text{Rel}}$ to (σ, ρ)-#DomSet$^{\mathcal{P}}$ when $(\{0\}, \mathbb{Z}_{\geq 0}) \in \mathcal{P}$. Then, we show that if σ is finite and $\rho = \mathbb{Z}_{\geq 0}$, then (σ, ρ)-#DomSet can simulate such vertices. However, when σ is cofinite this approach breaks down. In that case, a reduction from (σ, ρ)-#DomSet$^{\text{Rel}}$ to (σ, ρ)-#DomSet$^{\mathcal{P}}$ seems very challenging. Intuitively, $v \in S$ or $v \notin S$ can be distinguished only if v has a neighbor $u \in S$ (as the unselected neighbors of v are not affected by the status of v). Thus, in a sense, a gadget cannot detect the status of the vertex v without changing its number of neighbors, contrary to how relations are defined in (σ, ρ)-#DomSet$^{\text{Rel}}$. Problems with $\rho = \mathbb{Z}_{\geq 0}$ and σ cofinite seem to be of a very different type that need a separate treatment.

3 Preliminaries

3.1 Basics

Numbers, Sets, Strings, and Vectors

We use $a \equiv_{\text{m}} b$ as shorthand for $a \equiv b \pmod{\text{m}}$.

We write $\mathbb{Z}_{\geq 0} = \{0, 1, 2, 3, \dots\}$ to denote the set of non-negative integers and $\mathbb{Z}_{>0} = \{1, 2, 3, \dots\}$ to denote the positive integers. For integers i, j, we write $[\,i \,..\, j\,]$ for the set $\{i, \dots, j\}$, and $[\,i \,..\, j\,)$ for the set $\{i, \dots, j-1\}$. The sets $(\,i \,..\, j\,]$ and $(\,i \,..\, j\,)$ are defined similarly. A set $\tau \subseteq \mathbb{Z}_{\geq 0}$ is *cofinite* if $\mathbb{Z}_{\geq 0} \setminus \tau$ is finite. Also, we say that τ is *simple cofinite* if $\tau = \{n, n+1, n+2, \dots\}$ for some $n \in \mathbb{Z}_{\geq 0}$.

We write $s = s[\,1\,]s[\,2\,] \cdots s[\,n\,]$ for a *string* of length $|s| = n$ over an alphabet Σ. In some cases, we also write $s = (s[\,1\,], s[\,2\,], \cdots, s[\,n\,])$ to explicitly point to the individual characters of s. We write Σ^n for the set (or *language*) of all strings of length n, and $\Sigma^* := \bigcup_{n \geq 0} \Sigma^n$ for the set of all strings over Σ. We use ε to denote the empty string. For a string $s \in \Sigma^n$ and positions $i \leq j \in [\,1 \,..\, n\,]$, we write $s[\,i \,..\, j\,] := s[\,i\,] \cdots s[\,j\,]$; accordingly, we define $s(\,i \,..\, j\,]$, $s[\,i \,..\, j\,)$, and $s(\,i \,..\, j\,)$. Finally, for two strings s, t, we write st for their concatenation — in particular, we have $s = s[\,1 \,..\, i\,]s(\,i \,..\, n\,]$. Sometimes we are interested in the number of occurrences of an element $a \in \Sigma$ in a string s. To this end, we use the notation $\#_a(s)$ for the number of occurrences of a in s, that is, $\#_a(s) := |\{i \in |s| \mid s[\,i\,] = a\}|$. Also, for $A \subseteq \Sigma$, $\#_A(s)$ denotes the number of occurrences of elements from A in s, that is, $\#_A(s) := \sum_{a \in A} \#_a(s)$.

For a finite set X (for instance, a set of vertices of a graph), we write $\Sigma^X := \Sigma^{|X|}$ to emphasize that we index the strings in (subsets of) Σ^X with elements from X: for an $x_i \in X$, we write $s[\,x_i\,] := s[\,i\,]$.

Similarly, for an n-dimensional vector space $\mathbb{V}$, we view its elements as strings of length n and correspondingly write $v = v[1]v[2]\dots v[n] \in \mathbb{V}$. In addition to the notions defined for strings, for a set of positions $P \subseteq [1..n]$, we write $v[P] \coloneqq \bigcirc_{a\in P} x[a]$ for the $|P|$-dimensional vector that contains only the components of v whose indices are in P.

To improve readability, we sometimes use a "ranging star" $\star$ to range over unnamed objects. For example, if we wish to define a function $f\colon \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$, then we write $f(\star, 4) = 5$ to specify that $f(i, 4) = 5$ for all $i \in \mathbb{Z}_{\geq 0}$.

Graphs

We use standard notation for graphs. A *graph* is a pair $G = (V(G), E(G))$ with finite vertex set $V(G)$ and edge set $E(G) \subseteq \binom{V(G)}{2}$. Unless stated otherwise, all graphs considered in this paper are simple (that is, there are no loops or multiedges) and undirected. We use uv as a shorthand for edges $\{u, v\} \in E(G)$. We write $N_G(v)$ for the *(open) neighborhood* of a vertex $v \in V(G)$, that is, $N_G(v) \coloneqq \{w \in V(G) \mid vw \in E(G)\}$. The *degree* of v is the size of its (open) neighborhood, that is, $\deg_G(v) \coloneqq |N_G(v)|$. The *closed neighborhood* is $N_G[v] \coloneqq N_G(v) \cup \{v\}$. We usually omit the index G if it is clear from the context. For $X \subseteq V(G)$, we write $G[X]$ to denote the *induced subgraph* on the vertex set X, and $G - X \coloneqq G[V(G) \setminus X]$ denotes the induced subgraph on the complement of X.

Treewidth

Next, we define tree decompositions and recall some of their basic properties. For a more thorough introduction to tree decompositions and their many applications, we refer the reader to [20, Chapter 7].

Fix a graph G. A *tree decomposition* of G is a pair (T, β) that consists of a rooted tree T and a mapping $\beta\colon V(T) \to 2^{V(G)}$ such that

(T.1) $\bigcup_{t\in V(T)} \beta(t) = V(G)$,

(T.2) for every edge $vw \in E(G)$, there is some node $t \in V(T)$ such that $\{u, v\} \subseteq \beta(t)$, and

(T.3) for every $v \in V(G)$, the set $\{t \in V(T) \mid v \in \beta(t)\}$ induces a connected subtree of T.

The *width* of a tree decomposition (T, β) is defined as $\max_{t\in V(T)} |\beta(t)| - 1$. The *treewidth* of a graph G, denoted by $\mathsf{tw}(G)$, is the minimum width of a tree decomposition of G.

When designing algorithms on graphs of bounded treewidth, it is instructive to work with nice tree decompositions. Let (T, β) denote a tree decomposition and write $X_t \coloneqq \beta(t)$ for $t \in V(T)$. We say (T, β) is *nice* if $X_r = \varnothing$ where r denotes the root of T, $X_\ell = \varnothing$ for all leaves $\ell \in V(T)$, and every internal node $t \in V(T)$ has one of the following types:

Introduce: t has exactly one child t' and $X_t = X_{t'} \cup \{v\}$ for some $v \notin X_{t'}$; the vertex v *is introduced at* t,

Forget: t has exactly one child t' and $X_t = X_{t'} \setminus \{v\}$ for some $v \in X_{t'}$; the vertex v *is forgotten at* t, or

Join: t has exactly two children t_1, t_2 and $X_t = X_{t_1} = X_{t_2}$.

It is well-known that every tree decomposition (T, β) of G of width tw can be turned into a nice tree decomposition of the same width tw of size $O(\mathsf{tw} \cdot V(T))$ in time $O(\mathsf{tw}^2 \cdot \max(|V(G), V(T)|))$ (see, for instance, [20, Lemma 7.4]).

Treewidth-Preserving Reductions

To prove hardness results for computational problems on graphs of bounded treewidth (or variations thereof where treewidth is defined) we rely on reduction chains that, up to additive constants, preserve the treewidth of the input graphs.

Definition 3.1 (treewidth-preserving reduction)**.** *Let A and B denote two computational problems for which (some notion of) treewidth is defined.*

We write $A \leq_{\mathsf{tw}} B$ if there is a treewidth-preserving reduction *from A to B, that is, a polynomial-time Turing reduction from A to B that, if executed on an instance I of A given with a tree decomposition of width at most t, makes oracle calls to B only on instances that have size polynomial in $|I|$ and that are given with a tree decomposition of width at most $t + O(1)$.*

We say that A is the source *and B is the* target *of the reduction.*

Observation 3.2. *A sequence of treewidth-preserving reductions $A_1 \leq_{\mathsf{tw}} A_2 \leq_{\mathsf{tw}} \ldots \leq_{\mathsf{tw}} A_k$ gives a treewidth-preserving reduction from A_1 to A_k if $k \in O(1)$ (with respect to the input size of A_1).*

Note that we extend these definitions and observations in the obvious way to pathwidth-preserving reductions.

3.2 Generalized Dominating Sets

In the following, let $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ denote two sets that are finite or cofinite.

Basics

Fix a graph G. A set of vertices $S \subseteq V(G)$ is a (σ, ρ)-*set* if $|N(u) \cap S| \in \sigma$ for every $u \in S$, and $|N(v) \cap S| \in \rho$ for every $v \in V(G) \setminus S$. The (decision version of the) (σ, ρ)-DomSet problem takes as input a graph G, and asks whether G has a (σ, ρ)-set $S \subseteq V(G)$. We use (σ, ρ)-#DomSet to refer to the counting version, that is, the input to the problem is a graph G, and the task is to determine the number of (σ, ρ)-sets $S \subseteq V(G)$.

We say (σ, ρ) is *trivial* if $\rho = \{0\}$ or $(\sigma, \rho) = (\mathbb{Z}_{\geq 0}, \mathbb{Z}_{\geq 0})$.

Fact 3.3. *Suppose (σ, ρ) is trivial. Then, (σ, ρ)-#DomSet can be solved in polynomial time.*

Proof. For $(\sigma, \rho) = (\mathbb{Z}_{\geq 0}, \mathbb{Z}_{\geq 0})$, the number of (σ, ρ)-sets is $2^{|V(G)|}$. For $\rho = \{0\}$, the number of (σ, ρ)-sets is 2^c, where c denotes the number of those connected components of G where every vertex degree is contained in σ. □

In order to analyze the complexity of (σ, ρ)-DomSet (and (σ, ρ)-#DomSet) for non-trivial pairs (σ, ρ), we associate the following parameters with (σ, ρ). We define

$$r_{\text{top}} := \begin{cases} \max(\rho) & \text{if } \rho \text{ is finite,} \\ \max(\mathbb{Z} \setminus \rho) + 1 & \text{if } \rho \text{ is cofinite,} \end{cases} \quad \text{and} \quad s_{\text{top}} := \begin{cases} \max(\sigma) & \text{if } \sigma \text{ is finite,} \\ \max(\mathbb{Z} \setminus \sigma) + 1 & \text{if } \sigma \text{ is cofinite.} \end{cases} \tag{3.1}$$

Moreover, we set $t_{\text{top}} := \max(r_{\text{top}}, s_{\text{top}})$. For pairs (σ, ρ) that are structured in a certain way, we are able to obtain more efficient algorithms.

Definition 3.4 (m-structured sets)**.** *Fix an integer $\mathrm{m} \geq 1$. A set $\tau \subseteq \mathbb{Z}_{\geq 0}$ is* m-structured *if there is some number c^* such that*

$$c \equiv_{\mathrm{m}} c^*$$

for all $c \in \tau$.

We say that (σ, ρ) is m-*structured* if both ρ and σ are m-structured. Observe that (σ, ρ) is always 1-structured.

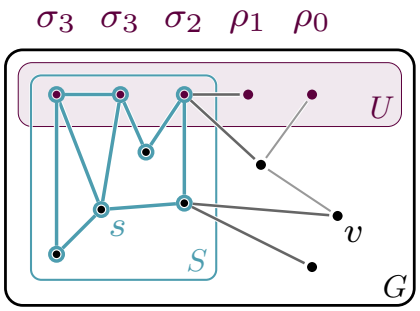


Figure 3.1: A graph G and subsets of vertices U and S. For $\rho = \{1\}, \sigma = \{2, 4\}$ the set S is a partial solution (with respect to U), as every blue vertex $s \in S \setminus U$ satisfies $|N(s) \cap S| \in \{2, 4\} = \sigma$ and as every black vertex $v \in V(G) \setminus (S \cup U)$ satisfies $|N(v) \cap S| \in \{1\} = \rho$. The depicted set S corresponds to the compatible string $\sigma_3\sigma_3\sigma_2\rho_1\rho_0$ (written above G). Note that S would not be a partial solution for $\sigma = \{4\}$, as all but one blue vertex have only 2 neighbors in S.

Partial Solutions and States

For both our algorithmic and our hardness results, a key ingredient is the description of *partial solutions*.

A *graph with portals* is a pair (G, U), where G is a graph and $U \subseteq V(G)$. If $U = \{u_1, \ldots, u_k\}$, then we also write $(G, u_1, \ldots, u_k)$ instead of (G, U).

Intuitively speaking, the idea of this notion is that G may be part of some larger graph that interacts with G only via vertices from U. In particular, in the context of the (σ, ρ)-DomSet problem, vertices in U do not necessarily need to satisfy the definition of a (σ, ρ)-set since they may receive further selected neighbors from outside of G.

Definition 3.5 (partial solution)**.** *Fix a graph with portals (G, U). A set $S \subseteq V(G)$ is a* partial solution *(with respect to U) if*

(PS1) for each $v \in V(G) \setminus (S \cup U)$, we have $|N(v) \cap S| \in \rho$, and

(PS2) for each $v \in S \setminus U$, we have $|N(v) \cap S| \in \sigma$.

To describe whether vertices from U are selected into partial solutions and how many selected neighbors they already have inside G, we associate a state with every vertex from U.

Formally, we write $\mathbb{S}_{\mathsf{full}} \coloneqq \{\sigma_i \mid i \in \mathbb{Z}_{\geq 0}\}$ for the set of potential ρ-states, and we write $\mathbb{R}_{\mathsf{full}} \coloneqq \{\rho_i \mid i \in \mathbb{Z}_{\geq 0}\}$ for the set of potential σ-states. We also write $\mathbb{A}_{\mathsf{full}} \coloneqq \mathbb{R}_{\mathsf{full}} \cup \mathbb{S}_{\mathsf{full}}$ for the set of *all* potential states.

Definition 3.6 (compatible strings)**.** *Fix a graph with portals (G, U). A string $x \in \mathbb{A}_{\mathsf{full}}^U$ is* compatible with (G, U) *if there is a partial solution $S_x \subseteq V(G)$ such that*

(X1) for each $v \in U \cap S_x$, we have $x[v] = \sigma_s$, where $s = |N(v) \cap S_x|$, and

(X2) for each $v \in U \setminus S_x$, we have $x[v] = \rho_r$, where $r = |N(v) \cap S_x|$.

We also refer to the vertices in S_x as being selected *and say that S_x is a* (partial) solution, selection, *or* witness *that* witnesses *x.*

Consult Figure 3.1 for a visualization of an example of a partial solution and its corresponding compatible string.

Observe that, despite $\mathbb{A}_{\mathsf{full}}$ being an infinite alphabet, for every graph with portals (G, U), only finitely many strings x can be realized. Indeed, if $|V(G)| = n$, then every compatible string can only have characters from $\mathbb{A}_n = \mathbb{S}_n \cup \mathbb{R}_n$, where $\mathbb{S}_n \coloneqq \{\sigma_i \mid i \in [0\,..\,n]\}$ and $\mathbb{R}_n \coloneqq \{\rho_i \mid i \in [0\,..\,n]\}$.

Definition 3.7 (realized language, L-provider)**.** *For a graph with portals (G, U), we define its realized language as*

$$L(G, U) \coloneqq \{x \in \mathbb{A}_{\mathsf{full}}^U \mid x \text{ is compatible with } (G, U)\}.$$

For a language $L \subseteq \mathbb{A}_{\mathsf{full}}^U$, we say that (G, U) is an L-realizer if $L = L(G, U)$.
For a language $L \subseteq \mathbb{A}_{\mathsf{full}}^U$, we say that (G, U) is an L-provider if $L \subseteq L(G, U)$.

Again, observe that $L(G, U) \subseteq \mathbb{A}_n^U$, where $n = |V(G)|$.

In fact, for most of our applications, it makes sense to restrict the alphabet even further. Recall the definition of s_{top} and r_{top} from Equation (3.1). Suppose that σ is finite. Then, we are usually not interested in partial solutions S where some vertex from U is selected and already has more than s_{top} selected neighbors (as it is impossible to extend this partial solution into a full solution). Also, if σ is infinite, it is usually irrelevant to us whether a selected vertex has exactly s_{top} selected neighbors, or more than s_{top} selected neighbors, since both options lead to the same outcome for all possible extensions of a partial solution. For this reason, we typically[2] restrict ourselves to the alphabets

$$\mathbb{R} \coloneqq \{\rho_0, \ldots, \rho_{r_{\text{top}}}\} \quad \text{and} \quad \mathbb{S} \coloneqq \{\sigma_0, \ldots, \sigma_{s_{\text{top}}}\}.$$

As before, we define $\mathbb{A} \coloneqq \mathbb{R} \cup \mathbb{S}$.

Definition 3.8 (inverse of a state)**.** *For a state $\sigma_s \in \mathbb{S}$, the* inverse of σ_s with respect to σ *is the state $\mathsf{inv}^\sigma(\sigma_s) = \sigma_{s_{\text{top}}-s}$. For a state $\rho_r \in \mathbb{R}$, the* inverse of ρ_r with respect to ρ *is the state $\mathsf{inv}^\rho(\rho_r) = \rho_{r_{\text{top}}-r}$.*

We set $\mathsf{inv}^\sigma(\rho_r) = \rho_r$ and $\mathsf{inv}^\rho(\sigma_s) = \sigma_s$. For all states $a \in \mathbb{A}$, the inverse of a with respect to σ, ρ is the state $\mathsf{inv}^{\sigma,\rho}(a) = \mathsf{inv}^\sigma(\mathsf{inv}^\rho(a))$.

We extend this in the natural way to strings by applying it coordinate-wise, and to sets by applying it to each element of the set.

[2] Sometimes, it turns out to be more convenient to work with the more general variants; we clearly mark said (rare) occurrences of $\mathbb{A}_{\mathsf{full}}$ and $\mathbb{A}_n$.

Part I
Faster Algorithms

4 Faster Algorithms for Structured Pairs

The goal of this section is to prove Theorem 1.3. With Theorem 1.1 in mind, we can restrict ourselves to the case where (σ, ρ) is m-structured for some $\mathrm{m} \geq 2$. In particular, both σ and ρ are finite in this case.

So for the remainder of this section, suppose that $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ are finite non-empty sets such that $\rho \neq \{0\}$. Recall that $r_{\mathrm{top}} \coloneqq \max \rho$, $s_{\mathrm{top}} \coloneqq \max \sigma$, and $t_{\mathrm{top}} \coloneqq \max(r_{\mathrm{top}}, s_{\mathrm{top}})$. Also suppose that (σ, ρ) is m-structured for some $\mathrm{m} \geq 2$, that is, there are integers $B, B' \in [0 \mathinner{..} \mathrm{m})$ such that $r \equiv_{\mathrm{m}} B$ for every $r \in \rho$ and $s \equiv_{\mathrm{m}} B'$ for every $s \in \sigma$. Without loss of generality, we may assume that $t_{\mathrm{top}} + 1 \geq \mathrm{m}$ (if $\mathrm{m} > t_{\mathrm{top}} + 1$, then $|\rho| = |\sigma| = 1$, which implies that (σ, ρ) is m-structured for every $\mathrm{m} \geq 2$).

For this case, we present faster dynamic-programming on tree-decomposition-based algorithms for (σ, ρ)-DomSet. In particular, we prove the following result.

Theorem 4.1. *Let (σ, ρ) denote finite m-structured sets for some $\mathrm{m} \geq 2$. Then, there is an algorithm $\mathcal{A}$ that, given a graph G and a nice tree decomposition of G of width tw, decides whether G has a (σ, ρ)-DomSet.*

If $\mathrm{m} \geq 3$ or t_{top} is odd or $\min(r_{\mathrm{top}}, s_{\mathrm{top}}) < t_{\mathrm{top}}$, then algorithm $\mathcal{A}$ runs in time

$$(t_{\mathrm{top}} + 1)^{\mathsf{tw}} \cdot (t_{\mathrm{top}} + \mathsf{tw})^{O(1)} \cdot |V(G)|.$$

If $\mathrm{m} = 2$, t_{top} is even, and $r_{\mathrm{top}} = s_{\mathrm{top}} = t_{\mathrm{top}}$, then algorithm $\mathcal{A}$ runs in time

$$(t_{\mathrm{top}} + 2)^{\mathsf{tw}} \cdot (t_{\mathrm{top}} + \mathsf{tw})^{O(1)} \cdot |V(G)|.$$

Observe that Theorem 4.1 is concerned with the decision version of the problem, and not the counting version. The reason is that we find it more convenient to first explain the algorithm for the decision version, and explain afterward how to modify the algorithm for the counting version. Also observe that, for the decision version, we obtain a linear bound on the running time in terms of the number of vertices, whereas for the counting version, we only have a polynomial bound.

We prove Theorem 4.1 in two steps. First, we obtain structural insights and an upper bound on the number of states that need to be maintained during the run of the dynamic programming algorithm. Second, we then show how to efficiently merge such states using a fast convolution-based algorithm.

4.1 Structural Insights into the m-Structured Case

In this section, we work with the alphabet $\mathbb{A} \coloneqq \{\sigma_0, \ldots, \sigma_{s_{\mathrm{top}}}, \rho_0, \ldots, \rho_{r_{\mathrm{top}}}\}$. Also, recall that $\mathbb{S} \coloneqq \{\sigma_0, \ldots, \sigma_{s_{\mathrm{top}}}\}$ and $\mathbb{R} \coloneqq \{\rho_0, \ldots, \rho_{r_{\mathrm{top}}}\}$. For a graph with portals (G, U), we aim to obtain a (tight) bound on the size of the realized language $L(G, U)$ in terms of s_{top} and r_{top}. Thereby, we also bound the number of states that are required in our dynamic-programming-based approach for computing a solution for a given graph G. To that end, we first define certain vectors associated with a string $x \in \mathbb{A}^n$, essentially decomposing a string into its σ/ρ component and its "weight"-component.

To able to reuse the definition in later sections, we state it for the full alphabet $\mathbb{A}_{\mathsf{full}}$.

Definition 4.2. *For a string $x \in \mathbb{A}^n_{\mathsf{full}}$, we define*

- *the* σ-vector of x *as* $\vec{\sigma}(x) \in \{0,1\}^n$ *with*

$$\vec{\sigma}(x)[\,i\,] \coloneqq \begin{cases} 1 & \text{if } x[\,i\,] \in \mathbb{S}, \\ 0 & \text{if } x[\,i\,] \in \mathbb{R}. \end{cases}$$

- *the* weight-vector of x *as* $\vec{w}(x) \in \mathbb{Z}^n_{\geq 0}$ *with*

$$\vec{w}(x)[\,i\,] \coloneqq c, \quad \text{where } x[\,i\,] \in \{\rho_c, \sigma_c\}.$$

- *the* m-weight-vector of x *as* $\vec{w}_{\mathrm{m}}(x) \in \mathbb{Z}^n_{\mathrm{m}}$ *with*

$$\vec{w}_{\mathrm{m}}(x)[\,i\,] \coloneqq \vec{w}(x)[\,i\,] \bmod \mathrm{m}.$$

For a language $L \subseteq \mathbb{A}^n_{\mathsf{full}}$, we write $\vec{\sigma}(L) \coloneqq \{\vec{\sigma}(x) \mid x \in L\}$ for the set of all σ-vectors of L, we write $\vec{w}(L) \coloneqq \{\vec{w}(x) \mid x \in L\}$ for the set of all weight-vectors of L, and we write $\vec{w}_{\mathrm{m}}(L) \coloneqq \{\vec{w}_{\mathrm{m}}(x) \mid x \in L\}$ for the set of all m*-weight-vectors of L.*

Finally, for a vector $\vec{s} \in \{0,1\}^n$, we define the capacity of $\vec{s}$ *as* $\mathrm{cap}_{\vec{s}} \in \{0, \ldots, t_{\mathrm{top}}\}^n$ *with*

$$\mathrm{cap}_{\vec{s}}[\,i\,] \coloneqq \begin{cases} s_{\mathrm{top}} & \text{if } \vec{s}[\,i\,] = 1, \\ r_{\mathrm{top}} & \text{if } \vec{s}[\,i\,] = 0. \end{cases}$$

To bound the size of realized languages, we proceed as follows. First, we compare two different partial solutions with respect to a fixed set U to obtain certain frequency properties of the characters of the corresponding string in $\mathbb{A}^U$ (expressed as σ-vectors and m-weight vectors). In a second step, we then show that there is only a moderate number of strings with said structure.

Fix a graph G, a subset of its vertices $U \subseteq V(G)$, and the realized language $L \coloneqq L(G,U) \subseteq \mathbb{A}^U$. For a string $x \in L$ and an integer $\mathrm{m} \geq 2$, we define the ordered partition $P_{\mathrm{m}}(x) \coloneqq (X_{\sigma,0}, \ldots, X_{\sigma,\mathrm{m}-1}, X_{\rho,0}, \ldots, X_{\rho,\mathrm{m}-1})$ of U with[3]

$$\begin{aligned} X_{\sigma,0} &\coloneqq \{v \in U \mid \vec{\sigma}(x)[\,v\,] = 1 \text{ and } \vec{w}_{\mathrm{m}}(x)[\,v\,] = 0\}, \\ &\ldots \\ X_{\sigma,\mathrm{m}-1} &\coloneqq \{v \in U \mid \vec{\sigma}(x)[\,v\,] = 1 \text{ and } \vec{w}_{\mathrm{m}}(x)[\,v\,] = \mathrm{m}-1\}, \\ X_{\rho,0} &\coloneqq \{v \in U \mid \vec{\sigma}(x)[\,v\,] = 0 \text{ and } \vec{w}_{\mathrm{m}}(x)[\,v\,] = 0\}, \\ &\ldots \\ X_{\rho,\mathrm{m}-1} &\coloneqq \{v \in U \mid \vec{\sigma}(x)[\,v\,] = 0 \text{ and } \vec{w}_{\mathrm{m}}(x)[\,v\,] = \mathrm{m}-1\}. \end{aligned}$$

Lemma 4.3. *Let (G,U) be a graph with portals and let $L \coloneqq L(G,U) \subseteq \mathbb{A}^U$ denote its realized language. Also let $x, y \in L$ denote strings with witnesses $S_x, S_y \subseteq V(G)$ such that $|S_x \setminus U| \equiv_{\mathrm{m}} |S_y \setminus U|$. Then, $\vec{\sigma}(x) \cdot \vec{w}_{\mathrm{m}}(y) \equiv_{\mathrm{m}} \vec{\sigma}(y) \cdot \vec{w}_{\mathrm{m}}(x)$.*

Proof. Consult Figure 4.1 for a visualization of an example.

In a first step, we count the number of edges between S_x and S_y in two different ways. The corresponding ordered partitions are

$$P_{\mathrm{m}}(x) = (X_{\sigma,0}, \ldots, X_{\sigma,\mathrm{m}-1}, X_{\rho,0}, \ldots, X_{\rho,\mathrm{m}-1}) \text{ and } P_{\mathrm{m}}(y) = (Y_{\sigma,0}, \ldots, Y_{\sigma,\mathrm{m}-1}, Y_{\rho,0}, \ldots, Y_{\rho,\mathrm{m}-1}).$$

[3] We chose this slightly convoluted-looking definition to simplify our exposition in Lemma 4.3.

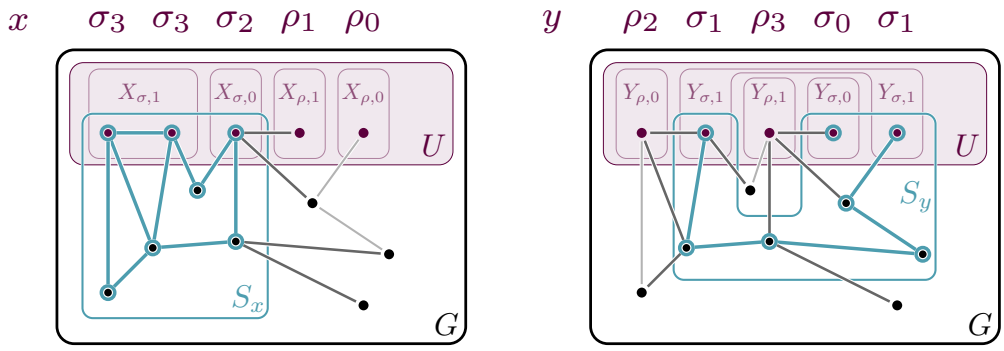


Figure 4.1: A graph G with portals U. For $\rho = \{1,3\}, \sigma = \{2,4\}$ (which are 2-structured) the strings $x = \sigma_3\sigma_3\sigma_2\rho_1\rho_0$ and $y = \rho_2\sigma_1\rho_3\sigma_0\sigma_1$ are compatible with (G,U); the corresponding partial solutions S_x and S_y, as well as the partitions of U are depicted above. We have $|S_x \setminus U| = |S_y \setminus U| = 4$ and $\vec{\sigma}(x) \cdot \vec{w}_{\mathrm{m}}(y) = (1,1,1,0,0) \cdot (0,1,1,0,1) = 2 \equiv_2 2 = (0,1,0,1,1) \cdot (1,1,0,1,0) = \vec{\sigma}(y) \cdot \vec{w}_{\mathrm{m}}(x)$.

Recall the integers $B, B' \in [0\,..\,\mathrm{m})$ with $r \equiv_{\mathrm{m}} B$ for every $r \in \rho$ and $s \equiv_{\mathrm{m}} B'$ for every $s \in \sigma$. Observe that by (PS1), every vertex in $V(G) \setminus (U \cup S_y)$ has $(B + \mathrm{m}\,\ell)$ neighbors in S_y (for some non-negative integer ℓ). In particular, this holds for all vertices in $S_x \setminus (U \cup S_y)$. Similarly, observe that by (PS2), every vertex in $S_x \cap S_y$ has $(B' + \mathrm{m}\,\ell')$ neighbors in S_y (for some non-negative integer ℓ'). Finally, a vertex v in $S_x \cap U = \bigcup_{j \in [0\,..\,\mathrm{m})} X_{\sigma,j}$ has exactly $i + \mathrm{m}\,\ell''$ neighbors in S_y (for some non-negative integer ℓ'') if and only if v is in one of $Y_{\rho,i}$ or $Y_{\sigma,i}$.

Writing $\vec{E}(X,Y) \coloneqq \{(v,w) \in E(G) \mid v \in X, w \in Y\}$, we obtain

$$\begin{aligned}
|\vec{E}(S_x,S_y)| &\equiv_{\mathrm{m}} B \cdot |S_x \setminus (U \cup S_y)| \\
&\quad + B' \cdot |S_x \cap S_y| \\
&\quad + \sum_{i \in [1\,..\,\mathrm{m})} \sum_{j \in [0\,..\,\mathrm{m})} i \cdot \big(|X_{\sigma,j} \cap Y_{\rho,i}| + |X_{\sigma,j} \cap Y_{\sigma,i}|\big) \\
&\equiv_{\mathrm{m}} B \cdot (|S_x \setminus U| - |S_x \cap S_y| + |S_x \cap S_y \cap U|) \\
&\quad + B' \cdot |S_x \cap S_y| \\
&\quad + \sum_{i \in [1\,..\,\mathrm{m})} \sum_{j \in [0\,..\,\mathrm{m})} i \cdot \big(|X_{\sigma,j} \cap Y_{\rho,i}| + |X_{\sigma,j} \cap Y_{\sigma,i}|\big).
\end{aligned}$$

In a symmetric fashion, we count the edges from S_y to S_x:

$$\begin{aligned}
|\vec{E}(S_y,S_x)| &\equiv_{\mathrm{m}} B \cdot (|S_y \setminus U| - |S_y \cap S_x| + |S_y \cap S_x \cap U|) \\
&\quad + B' \cdot |S_y \cap S_x| \\
&\quad + \sum_{i \in [1\,..\,\mathrm{m})} \sum_{j \in [0\,..\,\mathrm{m})} i \cdot \big(|Y_{\sigma,j} \cap X_{\rho,i}| + |Y_{\sigma,j} \cap X_{\sigma,i}|\big).
\end{aligned}$$

Now, we combine the previous equations and use the assumption that $|S_x \setminus U| \equiv_{\mathrm{m}} |S_y \setminus U|$ to obtain

$$\sum_{i \in [1\,..\,\mathrm{m})} \sum_{j \in [0\,..\,\mathrm{m})} i \cdot \big(|X_{\sigma,j} \cap Y_{\rho,i}| + |X_{\sigma,j} \cap Y_{\sigma,i}|\big) \equiv_{\mathrm{m}} \sum_{i \in [1\,..\,\mathrm{m})} \sum_{j \in [0\,..\,\mathrm{m})} i \cdot \big(|Y_{\sigma,j} \cap X_{\rho,i}| + |Y_{\sigma,j} \cap X_{\sigma,i}|\big).$$

Next, we unfold the definitions for $X_\star, Y_\star$ and observe that

$$\begin{aligned}
&\sum_{i\in[1\,..\,\mathrm{m})}\sum_{j\in[0\,..\,\mathrm{m})} i\cdot\big(|X_{\sigma,j}\cap Y_{\rho,i}|+|X_{\sigma,j}\cap Y_{\sigma,i}|\big)\\
&\quad\equiv_{\mathrm{m}}\sum_{i\in[1\,..\,\mathrm{m})}\sum_{j\in[0\,..\,\mathrm{m})} i\cdot|\{k\in[1\,..\,n]\mid \vec{\sigma}(x)[k]=1,\ \vec{w}_{\mathrm{m}}(x)[k]=j,\text{ and }\vec{w}_{\mathrm{m}}(y)[k]=i\}|\\
&\quad\equiv_{\mathrm{m}}\sum_{i\in[1\,..\,\mathrm{m})} i\cdot|\{k\in[1\,..\,n]\mid \vec{\sigma}(x)[k]=1\text{ and }\vec{w}_{\mathrm{m}}(y)[k]=i\}|\\
&\quad\equiv_{\mathrm{m}}\vec{\sigma}(x)\cdot\vec{w}_{\mathrm{m}}(y).
\end{aligned}$$

Hence, the claimed $\vec{\sigma}(x)\cdot\vec{w}_{\mathrm{m}}(y)\equiv_{\mathrm{m}}\vec{\sigma}(y)\cdot\vec{w}_{\mathrm{m}}(x)$ follows, completing the proof. □

With Lemma 4.3 in mind, in order to bound the size of a realized language, it suffices (up to a factor of m) to bound the size of a language $L\subseteq\mathbb{A}^n$ such that $L\times L\subseteq\mathcal{R}^n$, where

$$\mathcal{R}^n\coloneqq\{(x,y)\in\mathbb{A}^n\times\mathbb{A}^n\mid\vec{\sigma}(x)\cdot\vec{w}_{\mathrm{m}}(y)\equiv_{\mathrm{m}}\vec{\sigma}(y)\cdot\vec{w}_{\mathrm{m}}(x)\}. \tag{4.1}$$

Observe that the relation $\mathcal{R}^n$ is reflexive and symmetric.

We proceed to exploit the relation $\mathcal{R}^n$ to obtain size bounds for realized languages (in Section 4.1.1). Afterward, in Section 4.1.2, we investigate how said size bounds behave when combining realized languages.

4.1.1 Bounding the Size of a Single Realized Language

The goal of this section is to show the following result.

Theorem 4.4. *Let $L\subseteq\mathbb{A}^n$ denote a language with $L\times L\subseteq\mathcal{R}^n$.*
If $\mathrm{m}\geq 3$ or t_{top} is odd or $\min(r_{\mathrm{top}},s_{\mathrm{top}})<t_{\mathrm{top}}$, then $|L|\leq(t_{\mathrm{top}}+1)^n$.
If $\mathrm{m}=2$, t_{top} is even, and $r_{\mathrm{top}}=s_{\mathrm{top}}=t_{\mathrm{top}}$, then $|L|\leq(t_{\mathrm{top}}+2)^n$.

Note that the bounds of Theorem 4.4 are essentially optimal; consider the following example.

Example 4.5. *Consider the languages*

- $L_1\coloneqq\mathbb{R}^n=\{v\in\mathbb{A}^n\mid\vec{\sigma}(v)=0\}$,
- $L_2\coloneqq\{v\in\mathbb{S}^n\mid\sum_{\ell\in[1\,..\,n]}\vec{w}(v)[\ell]\equiv_{\mathrm{m}}0\}$, *and*
- $L_3\coloneqq\{v\in\mathbb{A}^n\mid\vec{w}(v)[\ell]\equiv_{\mathrm{m}}0$ *for all* $\ell\in[1\,..\,n]\}$.

It is straightforward to see that $L_i\times L_i\subseteq\mathcal{R}^n$ for all $i\in\{1,2,3\}$. We have that $|L_1|=(r_{\mathrm{top}}+1)^n$, $|L_2|\geq(s_{\mathrm{top}}+1)^{n-1}$ and

$$|L_3|=\left(\left\lceil\frac{r_{\mathrm{top}}+1}{\mathrm{m}}\right\rceil+\left\lceil\frac{s_{\mathrm{top}}+1}{\mathrm{m}}\right\rceil\right)^n.$$

Observe that $|L_3|=(t_{\mathrm{top}}+2)^n$ if $\mathrm{m}=2$, t_{top} is even, and $r_{\mathrm{top}}=s_{\mathrm{top}}=t_{\mathrm{top}}$. In all other cases, $|L_3|\leq(t_{\mathrm{top}}+1)^n$. In particular, the language L_3 indicates why the case $\mathrm{m}=2$ with even $r_{\mathrm{top}}=s_{\mathrm{top}}=t_{\mathrm{top}}$ stands out.

Toward proving Theorem 4.4, we start by showing that, for strings with the same σ-vector, the difference of their m-weight-vectors is "orthogonal" to the σ-vector of any other string.

Lemma 4.6. *Let $L \subseteq \mathbb{A}^n$ denote a language with $L \times L \subseteq \mathcal{R}^n$. For any three strings $v, w, z \in L$ with $\vec{\sigma}(v) = \vec{\sigma}(w)$, we have*

$$\big(\vec{w}_{\mathrm{m}}(v) - \vec{w}_{\mathrm{m}}(w)\big) \cdot \vec{\sigma}(z) \equiv_m 0.$$

Proof. Fix strings $v, w, z \in L$. By the definition of $\mathcal{R}^n$, we have

$$\vec{\sigma}(v) \cdot \vec{w}_{\mathrm{m}}(z) \equiv_{\mathrm{m}} \vec{\sigma}(z) \cdot \vec{w}_{\mathrm{m}}(v) \quad \text{and} \quad \vec{\sigma}(w) \cdot \vec{w}_{\mathrm{m}}(z) \equiv_{\mathrm{m}} \vec{\sigma}(z) \cdot \vec{w}_{\mathrm{m}}(w).$$

Using the assumption that $\vec{\sigma}(v) = \vec{\sigma}(w)$, we conclude that

$$\vec{\sigma}(z) \cdot \vec{w}_{\mathrm{m}}(v) \equiv_{\mathrm{m}} \vec{\sigma}(z) \cdot \vec{w}_{\mathrm{m}}(w),$$

which yields the claim after rearranging. □

Next, we explore the implications of Lemma 4.6. Intuitively, we show that, for a language L of strings of length n, each of the n positions contributes either to vectors from $\vec{\sigma}(L)$ or to vectors from $\vec{w}_{\mathrm{m}}(L)$. Formally, let us start with the notion of a σ-defining set.

Definition 4.7 (σ-defining set)**.** *Let $L \subseteq \mathbb{A}^n$. A set $S \subseteq [\,1\,..\,n\,]$ is* σ-defining *for $\vec{\sigma}(L)$ if S is an inclusion-minimal set of positions that uniquely characterize the σ-vectors of the strings in L, that is, for all $u, v \in L$, we have*

$$\vec{\sigma}(u)[\,S\,] = \vec{\sigma}(v)[\,S\,] \quad \Longrightarrow \quad \vec{\sigma}(u) = \vec{\sigma}(v). \tag{4.2}$$

Remark 4.8. *As a σ-defining S is (inclusion-)minimal, observe that, for each position $i \in S$, there are pairs of witness vectors $w_{1,i}, w_{0,i} \in \vec{\sigma}(L)$ that differ (on S) only at position i, with $w_{1,i}[\,i\,] = 1$, that is,*

- $w_{1,i}[\,S \setminus i\,] = w_{0,i}[\,S \setminus i\,]$,
- $w_{1,i}[\,i\,] = 1$, *and*
- $w_{0,i}[\,i\,] = 0$.

We write $\mathcal{W}_S \coloneqq \{w_{1,i}, w_{0,i} \mid i \in S\}$ for a set of witness vectors for $\vec{\sigma}(L)$. Note that, as S itself, the witness vectors $\mathcal{W}_S$ do not directly depend on strings in L, but only on the σ-vectors of L.

Lemma 4.9. *Let $L \subseteq \mathbb{A}^n$ denote a language with $L \times L \subseteq \mathcal{R}^n$ and let S denote a σ-defining set for L.*

Then, for any two strings $u, v \in L$ with $\vec{\sigma}(u) = \vec{\sigma}(v)$, the remaining positions $\bar{S} \coloneqq [\,1\,..\,n\,] \setminus S$ uniquely characterize the m*-weight vectors of u and v, that is, we have*

$$\vec{w}_{\mathrm{m}}(u)[\,\bar{S}\,] = \vec{w}_{\mathrm{m}}(v)[\,\bar{S}\,] \quad \Longrightarrow \quad \vec{w}_{\mathrm{m}}(u) = \vec{w}_{\mathrm{m}}(v). \tag{4.3}$$

Proof. Let $S \subseteq [\,1\,..\,n\,]$ denote a σ-defining set for $\vec{\sigma}(L)$ with witness vectors $\mathcal{W}_S$ (see Remark 4.8), and consider the set $\bar{S} \coloneqq [\,1\,..\,n\,] \setminus S$. We proceed to show that (4.3) is satisfied. To that end, let $u, v \in L$ denote strings with $\vec{\sigma}(u) = \vec{\sigma}(v)$ and $\vec{w}_{\mathrm{m}}(u)[\,\bar{S}\,] = \vec{w}_{\mathrm{m}}(v)[\,\bar{S}\,]$. We need to argue that $\vec{w}_{\mathrm{m}}(u) = \vec{w}_{\mathrm{m}}(v)$, and, in particular, that $\vec{w}_{\mathrm{m}}(u)[\,S\,] = \vec{w}_{\mathrm{m}}(v)[\,S\,]$. Hence, we proceed to show that, for every $i \in S$, we have $\vec{w}_{\mathrm{m}}(u)[\,i\,] = \vec{w}_{\mathrm{m}}(v)[\,i\,]$.

Now, fix a position $i \in S$ and corresponding witness vectors $w_{1,i}, w_{0,i} \in \mathcal{W}_S$. We proceed by showing two immediate equalities.

Claim 4.10. *The strings $u, v, w_{1,i}$, and $w_{0,i}$ satisfy*

$$\big(\vec{w}_{\mathrm{m}}(u) - \vec{w}_{\mathrm{m}}(v)\big) \cdot \big(w_{1,i} - w_{0,i}\big) \equiv_{\mathrm{m}} 0.$$

Proof of Claim. By Lemma 4.6, we have that

$$\begin{aligned}\big(\vec{w}_{\mathrm{m}}(u) - \vec{w}_{\mathrm{m}}(v)\big) \cdot \big(w_{1,i} - w_{0,i}\big) &\equiv_{\mathrm{m}} \big(\vec{w}_{\mathrm{m}}(u) - \vec{w}_{\mathrm{m}}(v)\big) \cdot w_{1,i} - \big(\vec{w}_{\mathrm{m}}(u) - \vec{w}_{\mathrm{m}}(v)\big) \cdot w_{0,i}\\ &\equiv_{\mathrm{m}} 0 - 0\\ &\equiv_{\mathrm{m}} 0,\end{aligned}$$

which completes the proof. ◁

Claim 4.11. *The strings $u, v, w_{1,i}$, and $w_{0,i}$ satisfy*

$$\big(\vec{w}_{\mathrm{m}}(u) - \vec{w}_{\mathrm{m}}(v)\big) \cdot \big(w_{1,i} - w_{0,i}\big) \equiv_{\mathrm{m}} \big(\vec{w}_{\mathrm{m}}(u)[\,i\,] - \vec{w}_{\mathrm{m}}(v)[\,i\,]\big) \cdot \big(w_{1,i}[\,i\,] - w_{0,i}[\,i\,]\big).$$

Proof of Claim. Observe that, by assumption, for every component $j \in \bar{S}$, we have $\vec{w}_{\mathrm{m}}(u)[\,j\,] = \vec{w}_{\mathrm{m}}(v)[\,j\,]$. Observe further that, by the definitions of i and S, for every component $j \in S \setminus i$, we have $w_{1,i}[\,j\,] = w_{0,i}[\,j\,]$, which yields the claim. ◁

Now, combining Claims 4.10 and 4.11 yields

$$\begin{aligned}0 &\equiv_{\mathrm{m}} \big(\vec{w}_{\mathrm{m}}(u) - \vec{w}_{\mathrm{m}}(v)\big) \cdot \big(w_{1,i} - w_{0,i}\big)\\ &\equiv_{\mathrm{m}} \big(\vec{w}_{\mathrm{m}}(u)[\,i\,] - \vec{w}_{\mathrm{m}}(v)[\,i\,]\big) \cdot \big(w_{1,i}[\,i\,] - w_{0,i}[\,i\,]\big)\\ &\equiv_{\mathrm{m}} \big(\vec{w}_{\mathrm{m}}(u)[\,i\,] - \vec{w}_{\mathrm{m}}(v)[\,i\,]\big) \cdot \big(1 - 0\big)\\ &\equiv_{\mathrm{m}} \vec{w}_{\mathrm{m}}(u)[\,i\,] - \vec{w}_{\mathrm{m}}(v)[\,i\,].\end{aligned}$$

In other words, $\vec{w}_{\mathrm{m}}(u)[\,i\,] = \vec{w}_{\mathrm{m}}(v)[\,i\,]$ for all $i \in S$, which yields (4.3), and hence, the claim. □

As a direct consequence of Lemma 4.9, we obtain a first upper bound on the size of languages $L \subseteq \mathbb{A}^n$ with $L \times L \subseteq \mathcal{R}^n$.

Corollary 4.12. *Let $L \subseteq \mathbb{A}^n$ denote a language with $L \times L \subseteq \mathcal{R}^n$, and let S denote a σ-defining set for $\vec{\sigma}(L)$. Then, we have*

$$|L| \leq (t_{\mathrm{top}} + 1)^{n-|S|} \cdot \sum_{k=0}^{|S|} \binom{|S|}{k} \left\lceil \frac{s_{\mathrm{top}} + 1}{\mathrm{m}} \right\rceil^{k} \left\lceil \frac{r_{\mathrm{top}} + 1}{\mathrm{m}} \right\rceil^{|S|-k}.$$

Proof. For a $k \in \{0, \ldots, |S|\}$, write L_k to denote the set of all strings $x \in L$ such that $\vec{\sigma}(x)[\,S\,]$ has a Hamming-weight of exactly k, that is, $\vec{\sigma}(x)[\,S\,]$ contains exactly k entries equal to 1.

As L decomposes into the different sets L_k, we obtain $|L| = \sum_{k=0}^{|S|} |L_k|$. Hence, it suffices to show that, for each $k \in \{0, \ldots, |S|\}$, we have

$$|L_k| \leq (t_{\mathrm{top}} + 1)^{n-|S|} \cdot \binom{|S|}{k} \cdot \left\lceil \frac{s_{\mathrm{top}} + 1}{\mathrm{m}} \right\rceil^{k} \cdot \left\lceil \frac{r_{\mathrm{top}} + 1}{\mathrm{m}} \right\rceil^{|S|-k}. \tag{4.4}$$

We proceed to argue that Inequality (4.4) does indeed hold. To that end, fix a $k \in \{0, \ldots, |S|\}$, and observe that a string $x \in L$ (and hence, $x \in L_k$) is uniquely determined by its σ-vector $\vec{\sigma}(x)$ and its weight vector $\vec{w}(x)$ (where elements are not taken modulo m). Note that as $L \times L \subseteq \mathcal{R}^n$, not all pairs of σ-vectors and weight-vectors correspond to a string in L_k. Hence, we write

$$|L_k| \leq \sum_{\vec{s} \in \vec{\sigma}(L_k)} |\{\vec{w}(x) \mid x \in L_k \text{ and } \vec{\sigma}(x) = \vec{s}\}|.$$

Now, as S is σ-defining for $\vec{\sigma}(L)$ (and hence, for $\vec{\sigma}(L_k) \subseteq \vec{\sigma}(L)$), for each σ-vector for L_k when restricted to the positions S, there is exactly one σ-vector for L_k (on all positions). In particular, the number of different σ-vectors of L_k is equal to the number of different σ-vectors on the positions S. Further, by construction of L_k, all σ-vectors on S have Hamming-weight exactly k. Hence, we obtain

$$|\vec{\sigma}(L_k)| = |\{\vec{\sigma}(x) \mid x \in L_k\}| = |\{\vec{\sigma}(x)[\,S\,] \mid x \in L_k\}| \leq \binom{|S|}{k}.$$

Now, fix a σ-vector $\vec{s} \in \vec{\sigma}(L_k)$, and write $L_{k,\vec{s}} \coloneqq \{x \in L_k \mid \vec{\sigma}(x) = \vec{s}\}$ for all strings in L_k with σ-vector $\vec{s}$. By Lemma 4.9, for each m-weight vector for $L_{k,\vec{s}}$ when restricted to the positions $\bar{S} \coloneqq [\,1\,..\,n\,] \setminus S$, there is exactly one m-weight vector for $L_{k,\vec{s}}$ (on all positions):

$$|\{\vec{w}_{\mathrm{m}}(x) \mid x \in L_{k,\vec{s}} \text{ and } \vec{w}_{\mathrm{m}}(x)[\,\bar{S}\,] = u\}| = 1. \tag{4.5}$$

Hence, it remains to count weight-vectors instead of m-weight vectors. To that end, for each possible weight-vector on $\bar{S}$, we count the possible extensions into a weight-vector on all positions. Writing u_{m} for the m-weight vector corresponding to a weight vector u, we obtain

$$\begin{aligned}
|\vec{w}(L_{k,\vec{s}})| &\leq \sum_{u \in \{\vec{w}(x)[\,\bar{S}\,] \mid x \in L_{k,\vec{s}}\}} |\{\vec{w}(x)[\,S\,] \mid x \in L_{k,\vec{s}} \text{ and } \vec{w}(x)[\,\bar{S}\,] = u\}| \\
&\leq \sum_{u \in \{\vec{w}(x)[\,\bar{S}\,] \mid x \in L_{k,\vec{s}}\}} |\{\vec{w}(x)[\,S\,] \mid x \in L_{k,\vec{s}} \text{ and } \vec{w}_{\mathrm{m}}(x)[\,\bar{S}\,] = u_{\mathrm{m}}\}|.
\end{aligned}$$

Finally, we bound $|\{\vec{w}(x)[\,S\,] \mid x \in L_{k,\vec{s}} \text{ and } \vec{w}_{\mathrm{m}}(x)[\,\bar{S}\,] = u_{\mathrm{m}}\}|$. To that end, observe that by Equation (4.5), the corresponding m-weight vector is unique. Hence, we need to bound only the number of different weight vectors (on S) that result in the same m-weight vector (on S). By construction, on the positions S, the string x contains exactly k characters $\sigma_\star$—for each such position, there are at most $\lceil (s_{\mathrm{top}}+1)/\mathrm{m} \rceil$ different characters having the same m-weight vector; for each of the remaining $|S|-k$ positions, there are at most $\lceil (r_{\mathrm{top}}+1)/\mathrm{m} \rceil$ different characters having the same m-weight vector. This yields

$$|\{\vec{w}(x)[\,S\,] \mid x \in L_{k,\vec{s}} \text{ and } \vec{w}_{\mathrm{m}}(x)[\,\bar{S}\,] = u_{\mathrm{m}}\}| \leq \left\lceil \frac{s_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{k} \left\lceil \frac{r_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{|S|-k}.$$

Combining the previous steps with the final observation that

$$|\{\vec{w}(x)[\,\bar{S}\,] \mid x \in L_{k,\vec{s}}\}| \leq (t_{\mathrm{top}}+1)^{|\bar{S}|} = (t_{\mathrm{top}}+1)^{n-|S|},$$

we obtain the claimed Inequality (4.4):

$$\begin{aligned}
|L_k| &\leq \sum_{\vec{s} \in \vec{\sigma}(L_k)} |\vec{w}(L_{k,\vec{s}})| \\
&\leq \binom{|S|}{k} \cdot \sum_{u \in \{\vec{w}(x)[\,\bar{S}\,] \mid x \in L_{k,\vec{s}}\}} |\{\vec{w}(x)[\,S\,] \mid x \in L_{k,\vec{s}} \text{ and } \vec{w}_{\mathrm{m}}(x)[\,\bar{S}\,] = u_{\mathrm{m}}\}| \\
&\leq (t_{\mathrm{top}}+1)^{n-|S|} \cdot \binom{|S|}{k} \cdot \left\lceil \frac{s_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{k} \cdot \left\lceil \frac{r_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{|S|-k}.
\end{aligned}$$

Overall, this yields the desired bound. □

In a final step before proving Theorem 4.4, we tidy up the unwieldy upper bound from Corollary 4.12.

Lemma 4.13. *For any non-negative integers n and $a \in [0\,..\,n]$, we have*

$$\begin{aligned}(t_{\mathrm{top}}+1)^{n-a} \cdot \sum_{k=0}^{a} \binom{a}{k} \left\lceil \frac{s_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{k} \left\lceil \frac{r_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{a-k} \\ \leq \begin{cases} (t_{\mathrm{top}}+2)^{n} & \text{if } \mathrm{m}=2 \text{ and } t_{\mathrm{top}}=s_{\mathrm{top}}=r_{\mathrm{top}} \text{ is even,} \\ (t_{\mathrm{top}}+1)^{n} & \text{otherwise.} \end{cases}\end{aligned}$$

Proof. In a first step, applying $t_{\mathrm{top}} = \max(r_{\mathrm{top}}, s_{\mathrm{top}})$ and the Binomial Theorem yields

$$\begin{aligned}&(t_{\mathrm{top}}+1)^{n-a} \cdot \sum_{k=0}^{a} \binom{a}{k} \left\lceil \frac{s_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{k} \left\lceil \frac{r_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{a-k} \\ &\leq (t_{\mathrm{top}}+1)^{n-a} \cdot \sum_{k=0}^{a} \binom{a}{k} \left\lceil \frac{t_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{k} \left\lceil \frac{t_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{a-k} \\ &= (t_{\mathrm{top}}+1)^{n-a} \cdot \left(2 \cdot \left\lceil \frac{t_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil\right)^{a}.\end{aligned}$$

In a next step, we investigate the term $\lceil (t_{\mathrm{top}}+1)/\mathrm{m} \rceil$.

First, if $\mathrm{m} \geq 3$ or if $\mathrm{m} = 2$ and t_{top} is odd, we have

$$2 \cdot \left\lceil \frac{t_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil \leq t_{\mathrm{top}}+1.$$

Hence, in these cases, we directly obtain

$$(t_{\mathrm{top}}+1)^{n-a} \cdot \left(2 \cdot \left\lceil \frac{t_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil\right)^{a} \leq (t_{\mathrm{top}}+1)^{n}.$$

Next, if $\mathrm{m} = 2$ and $t_{\mathrm{top}} = s_{\mathrm{top}} = r_{\mathrm{top}}$ is even, we have

$$2 \cdot \left\lceil \frac{t_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil = t_{\mathrm{top}}+2.$$

Hence, in this case, we directly obtain

$$(t_{\mathrm{top}}+1)^{n-a} \cdot \left(2 \cdot \left\lceil \frac{t_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil\right)^{a} \leq (t_{\mathrm{top}}+2)^{n}.$$

Finally, if $\mathrm{m} = 2$, t_{top} is even, and $\min(r_{\mathrm{top}}, s_{\mathrm{top}}) < t_{\mathrm{top}}$, we need a more careful analysis. Restarting from the initial term, we apply the Binomial Theorem and obtain

$$\begin{aligned}&(t_{\mathrm{top}}+1)^{n-a} \cdot \sum_{k=0}^{a} \binom{a}{k} \left\lceil \frac{s_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{k} \left\lceil \frac{r_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil^{a-k} \\ &= (t_{\mathrm{top}}+1)^{n-a} \cdot \left(\left\lceil \frac{s_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil + \left\lceil \frac{r_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil\right)^{a} \\ &\leq (t_{\mathrm{top}}+1)^{n-a} \cdot \left(\left\lceil \frac{t_{\mathrm{top}}}{\mathrm{m}} \right\rceil + \left\lceil \frac{t_{\mathrm{top}}+1}{\mathrm{m}} \right\rceil\right)^{a} \\ &= (t_{\mathrm{top}}+1)^{n-a} \cdot \left(\frac{t_{\mathrm{top}}}{2} + \frac{t_{\mathrm{top}}+2}{2}\right)^{a} \\ &\leq (t_{\mathrm{top}}+1)^{n}.\end{aligned}$$

This completes the proof. □

Finally, combining Corollary 4.12 and Lemma 4.13 directly yields Theorem 4.4, which we restate here for convenience.

Theorem 4.4. *Let $L \subseteq \mathbb{A}^n$ denote a language with $L \times L \subseteq \mathcal{R}^n$.*
If $\mathrm{m} \geq 3$ or t_{top} is odd or $\min(r_{\mathrm{top}}, s_{\mathrm{top}}) < t_{\mathrm{top}}$, then $|L| \leq (t_{\mathrm{top}} + 1)^n$.
If $\mathrm{m} = 2$, t_{top} is even, and $r_{\mathrm{top}} = s_{\mathrm{top}} = t_{\mathrm{top}}$, then $|L| \leq (t_{\mathrm{top}} + 2)^n$.

Proof. Let $L \subseteq \mathbb{A}^n$ denote a language with $L \times L \subseteq \mathcal{R}^n$, and let S denote a σ-defining set for L. By Corollary 4.12, we obtain

$$|L| \leq (t_{\mathrm{top}} + 1)^{n-|S|} \cdot \sum_{k=0}^{|S|} \binom{|S|}{k} \left\lceil \frac{s_{\mathrm{top}} + 1}{\mathrm{m}} \right\rceil^k \left\lceil \frac{r_{\mathrm{top}} + 1}{\mathrm{m}} \right\rceil^{|S|-k}.$$

Applying Lemma 4.13 yields the claim. □

4.1.2 Bounding the Size of Combinations of Realized Languages

Having understood a single realized language, we turn to *combinations* of realized languages next.

Definition 4.14. *For two strings $x, y \in \mathbb{A}^n$, we define their* combination *as the string $x \oplus y \in (\mathbb{A} \cup \{\bot\})^n$ obtained via*

$$(x \oplus y)[\ell] \coloneqq \begin{cases} \sigma_k & \text{if } x[\ell] = \sigma_i \text{ and } y[\ell] = \sigma_j \text{ and } i + j = k \leq s_{\mathrm{top}}, \\ \rho_k & \text{if } x[\ell] = \rho_i \text{ and } y[\ell] = \rho_j \text{ and } i + j = k \leq r_{\mathrm{top}}, \\ \bot & \text{otherwise.} \end{cases}$$

We say that x and y can be joined *if, for each position $\ell \in [1\,..\,n]$, we have $(x \oplus y)[\ell] \neq \bot$.*

For two languages $L_1, L_2 \subseteq \mathbb{A}^n$, we define their combination *as the set of all combinations of strings that can be joined:*

$$L_1 \oplus L_2 \coloneqq \{x \oplus y \mid x \in L_1 \text{ and } y \in L_2 \text{ such that } x, y \text{ can be joined}\}.$$

Observe that, for strings $x \in L_1$ and $y \in L_2$, their combination $x \oplus y$ is in $L_1 \oplus L_2$ if and only if x and y share a common σ-vector and the sum of their weight-vectors does not "overflow", that is, we have[4]

$$L_1 \oplus L_2 = \{x \oplus y \mid x \in L_1,\ y \in L_2,\ \vec{\sigma}(x) = \vec{\sigma}(y),\ \text{and } \vec{w}(x) + \vec{w}(y) \leq \mathrm{cap}_{\vec{\sigma}(x)}\}. \tag{4.6}$$

Finally, observe that, for strings $x \in L_1$ and $y \in L_2$ that can be joined, we have

$$\vec{w}(x) + \vec{w}(y) = \vec{w}(x \oplus y). \tag{4.7}$$

We use the remainder of this section to show that Corollary 4.12 and Theorem 4.4 easily lift to the combinations of realized languages. Thereby, we show that the number of partial solutions does not significantly increase by combining realized languages. We start with a small collection of useful properties of combinations of realized languages. First, we discuss how σ-vectors behave under combinations of languages.

Lemma 4.15. *Let $L_1, L_2 \subseteq \mathbb{A}^n$ denote languages with $L_1 \times L_1 \subseteq \mathcal{R}^n$ and $L_2 \times L_2 \subseteq \mathcal{R}^n$.*
Then, $(L_1 \oplus L_2)$ has the σ-vectors that appear for both L_1 and L_2, that is,

$$\vec{\sigma}(L_1 \oplus L_2) \subseteq \vec{\sigma}(L_1) \cap \vec{\sigma}(L_2).$$

[4] For ease of notation, we use "$\leq$" component-wise on vectors.

Proof. The proof follows immediately from Definition 4.14. Indeed, no string of the language $L_1 \oplus L_2$ has a $\perp$ character, as $(L_1 \oplus L_2)$ contains only combinations of strings with the same σ-vectors. Hence, the strings in $L_1 \oplus L_2$ may differ from strings in L_1 or L_2 only in their weight vectors, and $\vec{\sigma}(L_1 \oplus L_2)$ has only those σ-vectors that appear in both $\vec{\sigma}(L_1)$ and $\vec{\sigma}(L_2)$. Furthermore, note that due to "overflows" in the weight-vectors, a σ-vector in $\vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$ might not be in $\vec{\sigma}(L_1 \oplus L_2)$. □

Next, we show that having the relation $\mathcal{R}^n$ transfers as well.

Lemma 4.16. *Let $L_1, L_2 \subseteq \mathbb{A}^n$ denote languages with $L_1 \times L_1 \subseteq \mathcal{R}^n$ and $L_2 \times L_2 \subseteq \mathcal{R}^n$. Then, we have $(L_1 \oplus L_2) \times (L_1 \oplus L_2) \subseteq \mathcal{R}^n$.*

Proof. Fix strings $x, y \in L_1 \oplus L_2$. By Definition 4.14, this means that there are strings $x_1, y_1 \in L_1$ and $x_2, y_2 \in L_2$ such that $x = x_1 \oplus x_2$ and $y = y_1 \oplus y_2$. Expanding the definition of $\oplus$ yields

$$\vec{\sigma}(x) \cdot \vec{w}_{\mathrm{m}}(y) = \vec{\sigma}(x) \cdot (\vec{w}_{\mathrm{m}}(y_1) + \vec{w}_{\mathrm{m}}(y_2)) = \vec{\sigma}(x) \cdot \vec{w}_{\mathrm{m}}(y_1) + \vec{\sigma}(x) \cdot \vec{w}_{\mathrm{m}}(y_2).$$

As $\oplus$ does not change σ-vectors for strings that can be joined, we obtain

$$\vec{\sigma}(x) \cdot \vec{w}_{\mathrm{m}}(y) = \vec{\sigma}(x_1) \cdot \vec{w}_{\mathrm{m}}(y_1) + \vec{\sigma}(x_2) \cdot \vec{w}_{\mathrm{m}}(y_2)$$

Next, we use $(x_1, y_1) \in \mathcal{R}^n$ and $(x_2, y_2) \in \mathcal{R}^n$ to obtain

$$\vec{\sigma}(x) \cdot \vec{w}_{\mathrm{m}}(y) \equiv_{\mathrm{m}} \vec{\sigma}(y_1) \cdot \vec{w}_{\mathrm{m}}(x_1) + \vec{\sigma}(y_2) \cdot \vec{w}_{\mathrm{m}}(x_2)$$

Again, as $\oplus$ does not change σ-vectors for strings that can be joined, we obtain

$$\begin{aligned} \vec{\sigma}(x) \cdot \vec{w}_{\mathrm{m}}(y) &\equiv_{\mathrm{m}} \vec{\sigma}(y) \cdot \vec{w}_{\mathrm{m}}(x_1) + \vec{\sigma}(y) \cdot \vec{w}_{\mathrm{m}}(x_2) \\ &= \vec{\sigma}(y) \cdot (\vec{w}_{\mathrm{m}}(x_1) + \vec{w}_{\mathrm{m}}(x_2)) \\ &= \vec{\sigma}(y) \cdot \vec{w}_{\mathrm{m}}(x), \end{aligned}$$

which completes the proof that $(x, y) \in \mathcal{R}^n$. □

Now, we directly obtain that Corollary 4.12 lifts:

Corollary 4.17. *Let $L_1, L_2 \subseteq \mathbb{A}^n$ denote languages with $L_1 \times L_1 \subseteq \mathcal{R}^n$ and $L_2 \times L_2 \subseteq \mathcal{R}^n$. Further, let S denote a σ-defining set for $\vec{\sigma}(L_1 \oplus L_2)$. Then, we have*

$$|L_1 \oplus L_2| \leq (t_{\mathrm{top}} + 1)^{n-|S|} \cdot \sum_{k=0}^{|S|} \binom{|S|}{k} \left\lceil \frac{s_{\mathrm{top}} + 1}{\mathrm{m}} \right\rceil^k \left\lceil \frac{r_{\mathrm{top}} + 1}{\mathrm{m}} \right\rceil^{|S|-k}.$$

Proof. By Lemma 4.16, we can use Corollary 4.12 on the language $L_1 \oplus L_2$. □

4.2 Exploiting Structure: Fast Join Operations

Recall that the bound in Theorem 4.4 yields an upper bound on the number of partial solutions for a graph G and a subset U of its vertices. Recall further that, in the end, we intend to use an algorithm based on the dynamic programming on a tree decomposition paradigm. Hence, we need to be able to efficiently compute possible partial solutions for a graph given the already computed partial solutions for some of its subgraphs. We tackle this task next. In particular, we show how to generalize known convolution techniques to compute the combination of realized languages:

Theorem 4.18. *Let $L_1, L_2 \subseteq \mathbb{A}^n$ denote languages with $L_1 \times L_1 \subseteq \mathcal{R}^n$ and $L_2 \times L_2 \subseteq \mathcal{R}^n$, and let S denote a σ-defining set for $\vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$. Then, we can compute the language $L_1 \oplus L_2$ in time*

$$(n + t_{\text{top}})^{O(1)} \cdot (t_{\text{top}} + 1)^{n-|S|} \cdot \sum_{k=0}^{|S|} \binom{|S|}{k} \left\lceil \frac{s_{\text{top}} + 1}{\mathrm{m}} \right\rceil^{k} \left\lceil \frac{r_{\text{top}} + 1}{\mathrm{m}} \right\rceil^{|S|-k}$$
$$\leq \begin{cases} (n + t_{\text{top}})^{O(1)} \cdot (t_{\text{top}} + 2)^n & \text{if } \mathrm{m} = 2 \text{ and } t_{\text{top}} = s_{\text{top}} = r_{\text{top}} \text{ is even,} \\ (n + t_{\text{top}})^{O(1)} \cdot (t_{\text{top}} + 1)^n & \text{otherwise.} \end{cases}$$

Toward proving Theorem 4.18, first recall that strings $x_1 \in L_1$ and $x_2 \in L_2$ decompose into a σ-vector and a weight-vector each. Further, recall from Equation (4.6) that $x_1 \oplus x_2$ is in $L_1 \oplus L_2$ if and only if x_1 and x_2 share a common σ-vector and the sum of their weight-vectors does not "overflow". This observation yields the following proof strategy. For each different σ-vector $\vec{s} \in \vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$, we compute all possible sums of the weight-vectors for strings with σ-vector $\vec{s}$. Afterward, we filter out resulting vectors where an overflow occurred. To implement this strategy, we intend to make use of the tools developed by van Rooij [48]; in particular, the following result.

Fact 4.19 ([48, Lemma 3])**.** *For integers $d_1, \ldots, d_n$ and $D \coloneqq \prod_{i=1}^{n} d_i$, let p denote a prime such that in the field $\mathbb{F}_p$, the d_i-th root of unity exists for each $i \in [\![1..n]\!]$. Further, for two functions $f, g \colon \mathbb{Z}_{d_1} \times \cdots \times \mathbb{Z}_{d_n} \to \mathbb{F}_p$, let $h \colon \mathbb{Z}_{d_1} \times \cdots \times \mathbb{Z}_{d_n} \to \mathbb{F}_p$ denote the convolution*

$$h(a) \coloneqq \sum_{a_1 + a_2 = a} f(a_1) \cdot g(a_2).$$

Then, we can compute the function h in $O(D \log D)$ many arithmetic operations (assuming a d_i-th root of unity ω_i is given for all $i \in [\![1..n]\!]$).

Before we continue, let us briefly comment on how to find an appropriate prime p, as well as the roots of unity ω_i.

Remark 4.20. *Suppose M is a sufficiently large integer such that all images of the functions f, g, h are in the range $[\![0..M]\!]$. In particular, suppose that $M \geq D$. Suppose $d'_1, \ldots, d'_\ell$ is the list of integers obtained from $d_1, \ldots, d_n$ by removing duplicates (in all our applications, $\ell \leq 4$). Let $D' \coloneqq \prod_{i=1}^{\ell} d'_i$. We consider candidate numbers $m_j \coloneqq 1 + D'j$ for all $j \geq 1$. By the Prime Number Theorem for Arithmetic Progressions (see, for instance, [5]) there is a prime p such that*

1. *$p = m_j$ for some $j \geq 1$,*
2. *$p > M$, and*
3. *$p = O(\varphi(D')M)$,*

where φ denotes Euler's totient function. Such a number can be found in time

$$O\Big(p\big(\log p\big)^c\Big) = O\Big(\varphi(D')M\big(\log(\varphi(D')M)\big)^c\Big)$$

for some absolute constant c exploiting that prime testing can be done in polynomial time.

Now, fix $i \in [\![1..n]\!]$ and fix $k_i \coloneqq D'/d_i$. For every $x \in \mathbb{F}_p$, we have that $x^{p-1} = 1$, and hence, x^{k_i} is a d_i-th root of unity if and only $(x^{k_i})^i \neq 1$ for all $i < d_i$. Hence, given an element $x \in \mathbb{F}_p$, it can checked in time

$$O\Big(d_i \cdot (\log p)^c\Big)$$

whether x^{k_i} is a d_i-th root of unity. Due to our choice of p, this test succeeds for at least one $x \in \mathbb{F}_p$. Thus, a d_i-th root of unity ω_i for every $i \in [1..n]$ can be found in time

$$O\Big(n \cdot p \cdot \max_{i \in [1..n]} d_i \cdot (\log p)^c\Big).$$

Now, let us return to the problem at hand. The most naive approach, applying Fact 4.19 to the weight-vectors directly, is not fast enough for our purposes: A single convolution already takes time $\widetilde{O}((t_{\text{top}}+1)^n)$, and so, using such a convolution for each of the up to $2^{|S|}$ different σ-vectors is hence far too slow. Instead of convolving weight-vectors directly, we hence turn to Lemma 4.9: for a fixed σ-vector $\vec{s}$, there are far less (depending on the size of S) than $(t_{\text{top}}+1)^n$ different weight vectors. We can exploit this by *compressing* the weight-vectors to a smaller representation, and then convolving the resulting compressed vectors. Formally, we first make our intuition of "exploiting" Lemma 4.9 more formal by defining a useful auxiliary vector.

Definition 4.21. *Let $L \subseteq \mathbb{A}^n$ denote a (non-empty) language with $L \times L \subseteq \mathcal{R}^n$, let S denote a σ-defining set for $\vec{\sigma}(L)$, let $\mathcal{W}_S \subseteq \vec{\sigma}(L)$ denote a corresponding set of witness vectors, and set $\bar{S} \coloneqq [1..n] \setminus S$.*

For two vectors $u, o \in [0..t_{\text{top}}]^n$ and a position $\ell \in S$, we define the remainder $\text{rem}_{\mathcal{W}_S}(u,o)$ *at ℓ as*

$$\text{rem}_{\mathcal{W}_S}(u,o)[\ell] \coloneqq \sum_{i \in \bar{S}} \big(u[i] - o[i]\big) \cdot \big(w_{1,\ell}[i] - w_{0,\ell}[i]\big).$$

Remark 4.22. *Observe that, if we restrict u and o to be the weight-vectors of strings in L with a common σ-vector $\vec{s} \in \vec{\sigma}(L)$, that is, $u, o \in \{\vec{w}(x) \mid x \in L \text{ and } \vec{\sigma}(x) = \vec{s}\}$, then, for any $\ell \in S$, Lemma 4.6 yields*

$$\begin{aligned} u[\ell] - o[\ell] + \text{rem}_{\mathcal{W}_S}(u,o)[\ell] &= u[\ell] - o[\ell] + \sum_{i \in [1..n] \setminus S} \big(u[i] - o[i]\big) \cdot \big(w_{1,\ell}[i] - w_{0,\ell}[i]\big) \\ &= \big(u - o\big) \cdot \big(w_{1,\ell} - w_{0,\ell}\big) = \big(u - o\big) \cdot w_{1,\ell} - \big(u - o\big) \cdot w_{0,\ell} \\ &\equiv_{\text{m}} 0 - 0 \equiv_{\text{m}} 0. \end{aligned}$$

Note that Remark 4.22 closely mirrors Claim 4.10. Further, if we pick an arbitrary vector $o \in \{\vec{w}(x) \mid x \in L \text{ and } \vec{\sigma}(x) = \vec{s}\}$ to act as an "origin", then we can shift all vectors in $\{\vec{w}(x) \mid x \in L \text{ and } \vec{\sigma}(x) = \vec{s}\}$ so that their coordinates on S become divisible by m. We can then exploit this to compress the coordinates on S. In other words, the reduced flexibility due to fixing the σ-vector $\vec{s}$ translates to a reduced flexibility in the choice of coordinates for the positions that define the possible σ-vectors.

Finally, as we intend to add (the components of) compressed vectors modulo some number d_i (see Fact 4.19), we need to add "checksums" to the compressed vectors to be able to detect "overflows". This leads to the following definition.

Definition 4.23. *Let $L \subseteq \mathbb{A}^n$ denote a (non-empty) language with $L \times L \subseteq \mathcal{R}^n$, let S denote a σ-defining set for $\vec{\sigma}(L)$, let $\mathcal{W}_S \subseteq \vec{\sigma}(L)$ denote a corresponding set of witness vectors, and set $\bar{S} \coloneqq [1..n] \setminus S$.*

Further, fix a vector $\vec{s} \in \vec{\sigma}(L)$ and an origin vector $o \in [0..t_{\text{top}}]^n$ such that, for any $u \in \{\vec{w}(x) \mid x \in L \text{ and } \vec{\sigma}(x) = \vec{s}\}$, we have

$$u[\ell] - o[\ell] + \text{rem}_{\mathcal{W}_S}(u,o)[\ell] \equiv_{\text{m}} 0.$$

For a (weight-)vector $z \in \{\vec{w}(x) \mid x \in L \text{ and } \vec{\sigma}(x) = \vec{s}\}$, we define the σ-compression with origin o and type $\vec{s}$ *as the following $(n+2)$-dimensional vector $z{\downarrow_o}$:*

$$\begin{aligned}
z{\downarrow_o}[\,\ell\,] &\coloneqq z[\,\ell\,] && \bmod t_{\text{top}}+1, && \ell \in \bar{S},\\
z{\downarrow_o}[\,\ell\,] &\coloneqq \frac{z[\,\ell\,] - o[\,\ell\,] + \operatorname{rem}_{\mathcal{W}_S}(z,o)[\,\ell\,]}{\mathrm{m}} && \bmod \left\lceil \frac{\operatorname{cap}_{\vec{s}}[\,\ell\,]+1}{\mathrm{m}} \right\rceil, && \ell \in S,\\
z{\downarrow_o}[\,n+1\,] &\coloneqq \sum_{i \in \bar{S}} z[\,i\,] && \bmod 2n(t_{\text{top}}+1),\\
z{\downarrow_o}[\,n+2\,] &\coloneqq \sum_{i \in S} z[\,i\,] && \bmod 2n(t_{\text{top}}+1).
\end{aligned}$$

Further, we write $\mathcal{Z}_{S,\vec{s}}$ for the $(n+2)$-dimensional space of all σ-compressed vectors for S and $\vec{s}$ (and potentially different o).

Remark 4.24. *With Fact 4.19 in mind and writing $S_{\vec{s},c} \coloneqq \{\ell \in S \mid \vec{s}[\,\ell\,] = c\}$, we observe that*

$$|\mathcal{Z}_{S,\vec{s}}| = \left\lceil \frac{s_{\text{top}}+1}{\mathrm{m}} \right\rceil^{|S_{\vec{s},1}|} \cdot \left\lceil \frac{r_{\text{top}}+1}{\mathrm{m}} \right\rceil^{|S_{\vec{s},0}|} \cdot (t_{\text{top}}+1)^{n-|S|} \cdot 4n^2(t_{\text{top}}+1)^2.$$

In particular, using Fact 4.19 on the σ-compressed vectors yields a significant speed-up over the direct application to the weight vectors (whose domain has a size of $(t_{\text{top}}+1)^n$). □

Remark 4.25. *Observe that, for a fixed origin vector o, the mapping $\star{\downarrow_o}$ is injective and we can easily recover the original weight-vector z from its σ-compression $z{\downarrow_o}$:*[5]

$$\begin{aligned}
z[\,\ell\,] &\coloneqq z{\downarrow_o}[\,\ell\,], && \ell \in \bar{S},\\
z[\,\ell\,] &\coloneqq \big(\mathrm{m} \cdot z{\downarrow_o}[\,\ell\,] + o[\,\ell\,] - \operatorname{rem}_{\mathcal{W}_S}(z{\downarrow_o}, o)[\,\ell\,]\big) \bmod \mathrm{m} \left\lceil \frac{\operatorname{cap}_{\vec{s}}[\,\ell\,]+1}{\mathrm{m}} \right\rceil\\
&= \Big(\mathrm{m} \cdot z{\downarrow_o}[\,\ell\,] + o[\,\ell\,]\\
&\qquad - \sum_{i \in \bar{S}} \big(z{\downarrow_o}[\,i\,] - o[\,i\,]\big) \cdot \big(w_{1,\ell}[\,i\,] - w_{0,\ell}[\,i\,]\big)\Big) \bmod \mathrm{m} \left\lceil \frac{\operatorname{cap}_{\vec{s}}[\,\ell\,]+1}{\mathrm{m}} \right\rceil, && \ell \in S.
\end{aligned}$$

Further, for elements $x \in \mathcal{Z}_{S,\vec{s}}$ that cannot be obtained from a σ-compression, we have that either

$$\sum_{i \in \bar{S}} x[\,i\,] \neq x[\,n+1\,] \quad \text{or} \quad \sum_{i \in S} x[\,i\,] \neq x[\,n+2\,] \quad \text{or} \quad x[\,\ell\,] > \operatorname{cap}_{\vec{s}}[\,\ell\,] \quad \text{for an } \ell \in [\,1\,..\,n\,].$$

Hence, given a subset of $\mathcal{Z}_{S,\vec{s}}$, we can quickly identify which vectors are indeed σ-compressed weight vectors. □

In a next step, we discuss how addition and σ-compression interact with each other.

Lemma 4.26. *Let $L_1, L_2 \subseteq \mathbb{A}^n$ denote (non-empty) languages with $L_1 \times L_1 \subseteq \mathcal{R}^n$ and $L_2 \times L_2 \subseteq \mathcal{R}^n$, let S denote a σ-defining set for $\vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$, let $\mathcal{W}_S \subseteq \vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$ denote a corresponding set of witness vectors, and set $\bar{S} \coloneqq [\,1\,..\,n\,] \setminus S$.*

Further, fix a σ-vector $\vec{s} \in \vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$, as well as weight-vectors

$$o \in \{\vec{w}(x) \mid x \in L_1 \text{ and } \vec{\sigma}(x) = \vec{s}\} \quad \text{and} \quad p \in \{\vec{w}(x) \mid x \in L_2 \text{ and } \vec{\sigma}(x) = \vec{s}\}.$$

[5] Observe that we exploit $\operatorname{rem}_{\mathcal{W}_S}(z,o)[\,\ell\,] = \operatorname{rem}_{\mathcal{W}_S}(z{\downarrow_o}, o)[\,\ell\,]$.

For any two strings $u \in L_1, v \in L_2$ with $\vec{\sigma}(u) = \vec{\sigma}(v) = \vec{s}$, we have that u and v can be joined if and only if there is a string $z \in L_1 \oplus L_2$ with

$$\vec{w}(u)\!\downarrow_o + \vec{w}(v)\!\downarrow_p = \vec{w}(z)\!\downarrow_{o+p}. \tag{4.8}$$

If z exists, we have $z = u \oplus v$.

Proof. Fix two strings $u \in L_1, v \in L_2$ with $\vec{\sigma}(u) = \vec{\sigma}(v) = \vec{s}$, and suppose that they can be joined. Equation (4.7) now yields $\vec{w}(u) + \vec{w}(v) = \vec{w}(u \oplus v)$ and, in particular, for each position $\ell \in [\![1\,..\,n]\!]$, that

$$\vec{w}(u)[\![\,\ell\,]\!] + \vec{w}(v)[\![\,\ell\,]\!] \leq \mathrm{cap}_{\vec{s}}[\![\,\ell\,]\!] \leq t_{\mathrm{top}}.$$

Now, for Equation (4.8), only positions $\ell \in S$ warrant a short justification; for all other positions the result is immediate from Definition 4.23. Hence, for a position $\ell \in S$, first observe that we have

$$\begin{aligned}\mathrm{rem}_{\mathcal{W}_S}(\vec{w}(u), o)[\![\,\ell\,]\!] + \mathrm{rem}_{\mathcal{W}_S}(\vec{w}(v), p)[\![\,\ell\,]\!] &= \mathrm{rem}_{\mathcal{W}_S}(\vec{w}(u) + \vec{w}(v), o+p)[\![\,\ell\,]\!] \\ &= \mathrm{rem}_{\mathcal{W}_S}(\vec{w}(u \oplus v), o+p)[\![\,\ell\,]\!].\end{aligned}$$

Now, we obtain

$$\begin{aligned}0 &\equiv_{\mathrm{m}} \big(\vec{w}(u)[\![\,\ell\,]\!] - o[\![\,\ell\,]\!] + \mathrm{rem}_{\mathcal{W}_S}(\vec{w}(u), o)[\![\,\ell\,]\!]\big) + \big(\vec{w}(v)[\![\,\ell\,]\!] - p[\![\,\ell\,]\!] + \mathrm{rem}_{\mathcal{W}_S}(\vec{w}(v), p)[\![\,\ell\,]\!]\big) \\ &= \vec{w}(u \oplus v)[\![\,\ell\,]\!] - (o+p)[\![\,\ell\,]\!] + \mathrm{rem}_{\mathcal{W}_S}(\vec{w}(u \oplus v), o+p)[\![\,\ell\,]\!],\end{aligned}$$

which yields the claim.

For the other direction, fix two strings $u \in L_1, v \in L_2$ with $\vec{\sigma}(u) = \vec{\sigma}(v) = \vec{s}$, and suppose that there is a string $z \in L_1 \oplus L_2$ with $\vec{w}(u)\!\downarrow_o + \vec{w}(v)\!\downarrow_p = \vec{w}(z)\!\downarrow_{o+p}$. We proceed to show that then, indeed, u and v can be joined and $z = u \oplus v$.

First, consider the positions in the set $\bar{S}$, and in particular, fix an $\ell \in \bar{S}$. Now, we have

$$\vec{w}(z)[\![\,\ell\,]\!] = \vec{w}(z)\!\downarrow_{o+p}[\![\,\ell\,]\!] \equiv_{t_{\mathrm{top}}+1} \vec{w}(u)\!\downarrow_o[\![\,\ell\,]\!] + \vec{w}(v)\!\downarrow_p[\![\,\ell\,]\!] = \vec{w}(u)[\![\,\ell\,]\!] + \vec{w}(v)[\![\,\ell\,]\!].$$

In combination with $0 \leq \vec{w}(z)[\![\,\ell\,]\!] \leq \mathrm{cap}_{\vec{s}}[\![\,\ell\,]\!] \leq t_{\mathrm{top}} + 1$ and $0 \leq \vec{w}(u)[\![\,\ell\,]\!] + \vec{w}(v)[\![\,\ell\,]\!]$, we obtain

$$\vec{w}(z)[\![\,\ell\,]\!] \leq \vec{w}(u)[\![\,\ell\,]\!] + \vec{w}(v)[\![\,\ell\,]\!]. \tag{4.9}$$

Now, exploiting the "checksums", we obtain

$$\begin{aligned}\sum_{i \in \bar{S}} \vec{w}(z)[\![\,i\,]\!] &= \vec{w}(z)\!\downarrow_{o+p}[\![\,n+1\,]\!] \\ &\equiv_{2n(t_{\mathrm{top}}+1)} \vec{w}(u)\!\downarrow_o[\![\,n+1\,]\!] + \vec{w}(v)\!\downarrow_p[\![\,n+1\,]\!] = \sum_{i \in \bar{S}} \vec{w}(u)[\![\,i\,]\!] + \sum_{i \in \bar{S}} \vec{w}(v)[\![\,i\,]\!].\end{aligned} \tag{4.10}$$

Finally, as $u \in L_1$, $v \in L_2$, and $z \in L_1 \oplus L_2$, we have that, for all positions $i \in [\![1\,..\,n]\!]$,

$$0 \leq \vec{w}(u)[\![\,i\,]\!] \leq \mathrm{cap}_{\vec{s}}[\![\,\ell\,]\!] \leq t_{\mathrm{top}}, \quad 0 \leq \vec{w}(v)[\![\,i\,]\!] \leq \mathrm{cap}_{\vec{s}}[\![\,\ell\,]\!] \leq t_{\mathrm{top}}, \quad \text{and} \quad 0 \leq \vec{w}(z)[\![\,i\,]\!] \leq \mathrm{cap}_{\vec{s}}[\![\,\ell\,]\!] \leq t_{\mathrm{top}},$$

and hence,

$$0 \leq \sum_{i \in \bar{S}} \vec{w}(u)[\![\,i\,]\!] + \sum_{i \in \bar{S}} \vec{w}(v)[\![\,i\,]\!] < 2n(t_{\mathrm{top}} + 1) \text{ and } 0 \leq \sum_{i \in \bar{S}} \vec{w}(z)[\![\,i\,]\!] < 2n(t_{\mathrm{top}} + 1).$$

We conclude that Equation (4.10) is in fact an equality (over $\mathbb{Z}$). Hence, in combination with Equation (4.9), we obtain the desired

$$\vec{w}(u)[\,\ell\,] + \vec{w}(v)[\,\ell\,] = \vec{w}(z)[\,\ell\,] \leq \mathrm{cap}_{\vec{s}}[\,\ell\,].$$

Therefore, indeed $\vec{w}(u \oplus v)[\,\ell\,] = \vec{w}(u)[\,\ell\,] + \vec{w}(v)[\,\ell\,] = \vec{w}(z)[\,\ell\,]$.

Next, consider the positions in the set S, and in particular, fix an $\ell \in S$. The overall proof strategy is the same as before. We only need to adapt to the slightly more complicated definition of $\star{\downarrow_\star}[\,\ell\,]$. Writing $\mathrm{m}' \coloneqq \lceil (\mathrm{cap}_{\vec{s}}[\,\ell\,] + 1)/\mathrm{m} \rceil$ and applying Remark 4.25, we obtain

$$\begin{aligned}
\vec{w}(z)[\,\ell\,] &\equiv_{\mathrm{m}\cdot\mathrm{m}'} \mathrm{m} \cdot \vec{w}(z){\downarrow_{o+p}}[\,\ell\,] + (o+p)[\,\ell\,] - \mathrm{rem}_{\mathcal{W}_S}(\vec{w}(u) + \vec{w}(v), o+p)[\,\ell\,] \\
&\equiv_{\mathrm{m}\cdot\mathrm{m}'} \mathrm{m} \cdot (\vec{w}(u){\downarrow_o}[\,\ell\,] + \vec{w}(v){\downarrow_p}[\,\ell\,]) + (o[\,\ell\,] + p[\,\ell\,]) - \mathrm{rem}_{\mathcal{W}_S}(\vec{w}(z){\downarrow_{o+p}}, o+p)[\,\ell\,] \\
&\equiv_{\mathrm{m}\cdot\mathrm{m}'} \big(\mathrm{m} \cdot \vec{w}(u){\downarrow_o}[\,\ell\,] + o[\,\ell\,] - \mathrm{rem}_{\mathcal{W}_S}(\vec{w}(u){\downarrow_o}, o)[\,\ell\,]\big) \\
&\qquad + \big(\mathrm{m} \cdot \vec{w}(v){\downarrow_p}[\,\ell\,] + p[\,\ell\,] - \mathrm{rem}_{\mathcal{W}_S}(\vec{w}(v){\downarrow_p}, p)[\,\ell\,]\big) \\
&\equiv_{\mathrm{m}\cdot\mathrm{m}'} \vec{w}(u)[\,\ell\,] + \vec{w}(v)[\,\ell\,].
\end{aligned}$$

In combination with $0 \leq \vec{w}(z)[\,\ell\,] \leq \mathrm{cap}_{\vec{s}}[\,\ell\,] \leq \mathrm{m} \cdot \mathrm{m}'$ and $0 \leq \vec{w}(u)[\,\ell\,] + \vec{w}(v)[\,\ell\,]$, we obtain

$$\vec{w}(z)[\,\ell\,] \leq \vec{w}(u)[\,\ell\,] + \vec{w}(v)[\,\ell\,]. \tag{4.11}$$

Again, exploiting the "checksums", we obtain

$$\begin{aligned}
\sum_{i \in S} \vec{w}(z)[\,i\,] &= \vec{w}(z){\downarrow_{o+p}}[\,n+1\,] \\
&\equiv_{2n(t_{\mathrm{top}}+1)} \vec{w}(u){\downarrow_o}[\,n+1\,] + \vec{w}(v){\downarrow_p}[\,n+1\,] = \sum_{i \in S} \vec{w}(u)[\,i\,] + \sum_{i \in S} \vec{w}(v)[\,i\,].
\end{aligned} \tag{4.12}$$

Finally, as $u \in L_1$, $v \in L_2$, and $z \in L_1 \oplus L_2$, we have that, for all positions $i \in [\,1\,..\,n\,]$,

$$0 \leq \vec{w}(u)[\,i\,] \leq \mathrm{cap}_{\vec{s}}[\,\ell\,] \leq t_{\mathrm{top}}, \quad 0 \leq \vec{w}(v)[\,i\,] \leq \mathrm{cap}_{\vec{s}}[\,\ell\,] \leq t_{\mathrm{top}}, \quad \text{and} \quad 0 \leq \vec{w}(z)[\,i\,] \leq \mathrm{cap}_{\vec{s}}[\,\ell\,] \leq t_{\mathrm{top}},$$

and hence,

$$0 \leq \sum_{i \in S} \vec{w}(u)[\,i\,] + \sum_{i \in S} \vec{w}(v)[\,i\,] < 2n(t_{\mathrm{top}}+1) \text{ and } 0 \leq \sum_{i \in S} \vec{w}(z)[\,i\,] < 2n(t_{\mathrm{top}}+1).$$

We conclude that Equation (4.12) is in fact an equality (over $\mathbb{Z}$). Hence, in combination with Equation (4.11), we obtain the desired

$$\vec{w}(u)[\,\ell\,] + \vec{w}(v)[\,\ell\,] = \vec{w}(z)[\,\ell\,] \leq \mathrm{cap}_{\vec{s}}[\,\ell\,].$$

Therefore, indeed $\vec{w}(u \oplus v)[\,\ell\,] = \vec{w}(u)[\,\ell\,] + \vec{w}(v)[\,\ell\,] = \vec{w}(z)[\,\ell\,]$.

Overall, we obtain that u and v can be joined and that $z = u \oplus v$, which completes the proof. □

Finally, we are ready to give algorithms. First, we discuss how to compute a σ-defining set S (as well as witness strings that certify that S is indeed a minimal set).

Lemma 4.27. *Given a language $L \subseteq \mathbb{A}^n$, we can compute a σ-defining set S for $\vec{\sigma}(L)$, as well as a set of witness vectors $\mathcal{W}_S$ for S, in time $O(|L| \cdot n^4)$.*

Proof. Given a language $L \subseteq \mathbb{A}^n$, we first compute the set $\vec{\sigma}(L)$ of all σ-vectors. Then, starting with $S \coloneqq [\![1..n]\!]$, we repeatedly iterate over the positions 1 to n; for each position i, we check if i can be removed from S, that is, whether removing the i-th position from each vector in $\vec{\sigma}(L)$ does not decrease the size of $\vec{\sigma}(L)$. If removing the i-th position would decrease the size of $\vec{\sigma}(L)$, we also store two witness vectors that differ only at position i, but not at the remaining positions in S. We stop when no further positions can be removed; we return the resulting set S, as well as the corresponding pairs of witness vectors.

We can check if a position $i \in S$ can be removed by checking if the (multi-)set

$$\vec{\sigma}(L)_{S,i} \coloneqq \{v[\![S \setminus \{i\}]\!] \mid v \in \vec{\sigma}(L)\}$$

(where position i is removed) contains a duplicate element. This we can do by a linear scan over $\vec{\sigma}(L)_{S,i}$ to construct $\vec{\sigma}(L)_{S,i}$, sorting $\vec{\sigma}(L)_i$, and then another linear scan over $\vec{\sigma}(L)_{S,i}$. Observe that if we indeed detect a duplicate, we have also found the required witness.

For the correctness, observe that during the algorithm we maintain that the positions in S uniquely identify the vectors in $\vec{\sigma}(L)$; as we maintain the size of $\vec{\sigma}(L)$, the resulting set does indeed also uniquely identify the vectors in the initial set $\vec{\sigma}(L)$. Further, our algorithm trivially ensures that S is minimal, and hence, the returned set S is indeed σ-defining for $\vec{\sigma}(L)$.

For the running time, we can compute $\vec{\sigma}(L)$ in time $O(|L| \log |\vec{\sigma}(L)| \cdot n) = O(|L| \cdot n^2)$ by iterating over L and computing each σ-vector separately; filtering out duplicates by using an appropriate data structure. Next, observe that we iterate over all positions in S at most n times; in each iteration, we check for at most n positions whether they can be removed from S. The check if we can remove a position from S runs in the time it takes to sort $\vec{\sigma}(L)$, which we can bound by $O(|\vec{\sigma}(L)| \log |\vec{\sigma}(L)| \cdot n) = O(|L| \cdot n^2)$. Hence, in total the algorithm runs in the claimed running time of $O(|L| \cdot n^4)$, which completes the proof. □

Lastly, we prove the promised main result, which we restate here for convenience.

Theorem 4.18. *Let $L_1, L_2 \subseteq \mathbb{A}^n$ denote languages with $L_1 \times L_1 \subseteq \mathcal{R}^n$ and $L_2 \times L_2 \subseteq \mathcal{R}^n$, and let S denote a σ-defining set for $\vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$. Then, we can compute the language $L_1 \oplus L_2$ in time*

$$\begin{aligned}
&(n + t_{\text{top}})^{O(1)} \cdot (t_{\text{top}} + 1)^{n-|S|} \cdot \sum_{k=0}^{|S|} \binom{|S|}{k} \left\lceil \frac{s_{\text{top}} + 1}{\mathrm{m}} \right\rceil^k \left\lceil \frac{r_{\text{top}} + 1}{\mathrm{m}} \right\rceil^{|S|-k} \\
&\quad \leq \begin{cases} (n + t_{\text{top}})^{O(1)} \cdot (t_{\text{top}} + 2)^n & \text{if } \mathrm{m} = 2 \text{ and } t_{\text{top}} = s_{\text{top}} = r_{\text{top}} \text{ is even,} \\ (n + t_{\text{top}})^{O(1)} \cdot (t_{\text{top}} + 1)^n & \text{otherwise.} \end{cases}
\end{aligned}$$

Proof. Given the languages L_1 and L_2, we first compute $\vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$ and drop any strings from L_1 and L_2 whose σ-vectors are not in $\vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$. Next, we use Lemma 4.27 to compute a σ-defining set S for $\vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$ as well as a set of witness vectors $\mathcal{W}_S$. Now, for each σ-vector $\vec{s} \in \vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$, we compute the sets

$$\vec{w}(L_{1,\vec{s}}) \coloneqq \{\vec{w}(u) \mid u \in L_1 \text{ and } \vec{\sigma}(u) = \vec{s}\} \quad \text{and} \quad \vec{w}(L_{2,\vec{s}}) \coloneqq \{\vec{w}(v) \mid v \in L_2 \text{ and } \vec{\sigma}(v) = \vec{s}\}.$$

Next, we pick arbitrary vectors $o \in L_{1,\vec{s}}$ and $p \in L_{2,\vec{s}}$, and compute the functions $f_1, f_2 : \mathcal{Z}_{S,\vec{s}} \to \mathbb{Z}$

$$f_1(x) \coloneqq \begin{cases} 1 & \text{if } x = u{\downarrow_o} \text{ for some } u \in L_{1,\vec{s}}, \\ 0 & \text{otherwise;} \end{cases} \quad \text{and} \quad f_2(y) \coloneqq \begin{cases} 1 & \text{if } y = v{\downarrow_p} \text{ for some } v \in L_{2,\vec{s}}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.13}$$

Using Fact 4.19, we compute the function $h : \mathcal{Z}_{S,\vec{s}} \to \mathbb{Z}$,

$$h(a) \coloneqq \sum_{x+y=a} f_1(x) \cdot f_2(y).$$

Using Remark 4.25, we compute the set $\vec{w}(L_{1,2,\vec{s}})$ of all weight-vectors whose compression has a positive value for h:

$$\vec{w}(L_{1,2,\vec{s}}) \coloneqq \{z \mid \exists a \in \mathcal{Z}_{S,\vec{s}} \colon h(a) > 0 \text{ and } z{\downarrow_{o+p}} = a\}.$$

Iterating over $\vec{w}(L_{1,2,\vec{s}})$, we uniquely reconstruct a string from the weight vector and $\vec{s}$ in the straightforward way to obtain

$$L_{1,2,\vec{s}} \coloneqq \{z \in \mathbb{A}^n \mid \vec{\sigma}(z) = \vec{s} \text{ and } \vec{w}(z) \in \vec{w}(L_{1,2,\vec{s}})\}.$$

Finally, we return the union $L_{1,2}$ of all sets $L_{1,2,\vec{s}}$ computed,

$$L_{1,2} \coloneqq \bigcup_{\vec{s} \in \vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)} L_{1,2,\vec{s}}.$$

For the correctness, first observe that by Lemma 4.15, any string $z \in L_1 \oplus L_2$ has the same σ-vector as a string in L_1 and a string in L_2. Hence, we can indeed compute the strings in $L_1 \oplus L_2$ for each σ-vector separately. Next, by Lemma 4.26 and Remark 4.25, the set $\vec{w}(L_{1,2,\vec{s}})$ is indeed the set of all weight-vectors of strings in $L_1 \oplus L_2$ with σ-vector $\vec{s}$:

$$\vec{w}(L_{1,2,\vec{s}}) = \vec{w}(\{z \in L_1 \oplus L_2 \mid \vec{\sigma}(z) = \vec{s}\}).$$

Hence, in total, the algorithm does indeed compute $L_1 \oplus L_2 = L_{1,2}$.

For the running time, we can compute the sets $\vec{\sigma}(L_1)$ and $\vec{\sigma}(L_2)$ in total time

$$O\big((|L_1| + |L_2|) \log(|\vec{\sigma}(L_1)| + |\vec{\sigma}(L_2)|) \cdot n\big) = O(\max\{|L_1|, |L_2|\} \cdot n^2)$$

by iterating over L_1 (or L_2) and computing each σ-vector separately; filtering out duplicates by using an appropriate data structure (note that $|\vec{\sigma}(L_1)|, |\vec{\sigma}(L_2)| \leq 2^n$). Afterward, we can compute $\vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$ in the same running time by using standard algorithms for merging sets.

Using the algorithm from Lemma 4.27, we can compute the σ-defining set S (as well as the corresponding witness vectors) in time $O(\max\{|L_1|, |L_2|\} \cdot n^4)$. As S is σ-defining for $\vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$, we have $|\vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)| \leq 2^{|S|}$. Now, for a fixed σ-vector $\vec{s} \in \vec{\sigma}(L_1) \cap \vec{\sigma}(L_2)$, write $k(\vec{s}) \coloneqq |\{i \in S \mid \vec{s}[\,i\,] = 1\}|$ for the number of entries 1 of the vector $\vec{s}$ on positions from S. Recalling Remark 4.24, we see that the application of Fact 4.19 takes time

$$(n + t_{\text{top}})^{O(1)} \cdot (t_{\text{top}} + 1)^{n-|S|} \left\lceil \frac{s_{\text{top}} + 1}{\mathrm{m}} \right\rceil^{k(\vec{s})} \left\lceil \frac{r_{\text{top}} + 1}{\mathrm{m}} \right\rceil^{|S|-k(\vec{s})}.$$

Here, using the notation of Remark 4.20, we set $M \coloneqq |\mathcal{Z}_{S,\vec{s}}|$ and exploit that $D' = (n + t_{\text{top}})^{O(1)}$ to compute the prime p and the required roots of unity in the desired time. Finally, recovering $L_{1,2,\vec{s}}$ can be done with a linear pass over $\mathcal{Z}_{S,\vec{s}}$ in the same running time; combining the recovered (disjoint) sets can then be done in linear time of the returned result $L_1 \oplus L_2$.

In total, the algorithm thus runs in time

$$\begin{aligned}&(n + t_{\text{top}})^{O(1)} \cdot \max\{|L_1|, |L_2|, |L_1 \oplus L_2|\} \\ &\quad + (n + t_{\text{top}})^{O(1)} \cdot (t_{\text{top}} + 1)^{n-|S|} \cdot \sum_{k=0}^{|S|} \binom{|S|}{k} \left\lceil \frac{s_{\text{top}} + 1}{\mathrm{m}} \right\rceil^{k} \left\lceil \frac{r_{\text{top}} + 1}{\mathrm{m}} \right\rceil^{|S|-k}.\end{aligned}$$

Using Corollaries 4.12 and 4.17, the running time simplifies to

$$(n + t_{\text{top}})^{O(1)} \cdot (t_{\text{top}} + 1)^{n-|S|} \cdot \sum_{k=0}^{|S|} \binom{|S|}{k} \left\lceil \frac{s_{\text{top}} + 1}{\mathrm{m}} \right\rceil^{k} \left\lceil \frac{r_{\text{top}} + 1}{\mathrm{m}} \right\rceil^{|S|-k}.$$

Finally, Lemma 4.13 yields the tidier upper bound, which completes the proof. □

4.3 Faster Algorithms for Generalized Dominating Set Problems

Finally, we use Theorem 4.18 to obtain faster algorithms for (σ, ρ)-DomSet; that is, we prove Theorem 4.1, which we restate here for convenience.

Theorem 4.1. *Let (σ, ρ) denote finite m-structured sets for some $\mathrm{m} \geq 2$. Then, there is an algorithm $\mathcal{A}$ that, given a graph G and a nice tree decomposition of G of width tw, decides whether G has a (σ, ρ)-DomSet.*

If $\mathrm{m} \geq 3$ or t_{top} is odd or $\min(r_{\text{top}}, s_{\text{top}}) < t_{\text{top}}$, then algorithm $\mathcal{A}$ runs in time

$$(t_{\text{top}} + 1)^{\mathsf{tw}} \cdot (t_{\text{top}} + \mathsf{tw})^{O(1)} \cdot |V(G)|.$$

If $\mathrm{m} = 2$, t_{top} is even, and $r_{\text{top}} = s_{\text{top}} = t_{\text{top}}$, then algorithm $\mathcal{A}$ runs in time

$$(t_{\text{top}} + 2)^{\mathsf{tw}} \cdot (t_{\text{top}} + \mathsf{tw})^{O(1)} \cdot |V(G)|.$$

Proof. For ease of notation, let us define $\tau := t_{\text{top}} + 1$ if $\mathrm{m} \geq 3$ or t_{top} is odd or $\min(r_{\text{top}}, s_{\text{top}}) < t_{\text{top}}$, and $\tau := t_{\text{top}} + 2$ if $\mathrm{m} = 2$, t_{top} is even, and $r_{\text{top}} = s_{\text{top}} = t_{\text{top}}$.

Let (T, β) denote the nice tree decomposition of G. For $t \in V(T)$, we set $X_t := \beta(t)$ and write V_t for the set of vertices contained in bags below t (including t itself).

For each node $t \in V(T)$ and each $i \in [0 \,..\, \mathrm{m})$, we compute the language $L_{t,i} \subseteq \mathbb{A}^{X_t}$ of all strings $x \in \mathbb{A}^{X_t}$ that are compatible with $(G[V_t], X_t)$[6] via a witnessing solution set S_x such that $|S_x \setminus X_t| \equiv_{\mathrm{m}} i$. We have that $L_{t,i} \times L_{t,i} \subseteq \mathcal{R}^{|X_t|}$ by Lemma 4.3, and hence, Theorem 4.4 yields

$$|L_{t,i}| \leq \tau^{|X_t|} \leq \tau^{\mathsf{tw}+1}. \tag{4.14}$$

We compute the sets $L_{t,i}$ for nodes $t \in V(T)$ in a bottom-up fashion starting at the leaves of T.

For a leaf t of T, we have $X_t = V_t = \varnothing$ and $L_{t,i} = \{\varepsilon\}$ for every $i \in [0 \,..\, \mathrm{m})$ (where ε denotes the empty string).

For an internal node t, suppose we already computed all sets $L_{t',i}$ for all children t' of t. We proceed depending on the type of t.

Forget: First, suppose t is a forget-node, and let t' denote the unique child of t. Also, assume that $X_{t'} = X_t \cup \{v\}$, that is, $v \in V(G)$ is the vertex forgotten at t. We say that a string $x \in \mathbb{A}^{X_{t'}}$ *is ρ-happy at v* if $x[v] = \rho_c$ and $c \in \rho$. Also, we say that a string $x \in \mathbb{A}^{X_{t'}}$ *is σ-happy at v* if $x[v] = \sigma_d$ and $d \in \sigma$. It is easy to see that

$$\begin{aligned} L_{t,i} = \; & \{x[X_t] \mid x \in L_{t',i} \text{ such that } x \text{ is } \rho\text{-happy at } v\} \\ & \cup \{x[X_t] \mid x \in L_{t',i-1} \text{ such that } x \text{ is } \sigma\text{-happy at } v\}, \end{aligned}$$

where the index $i - 1$ is taken modulo m. Hence, using Equation (4.14), for each $i \in [0 \,..\, \mathrm{m})$, we can compute the set $L_{t,i}$ in time $\tau^{\mathsf{tw}} \cdot (t_{\text{top}} + \mathsf{tw})^{O(1)}$.

[6] To follow standard notation for dynamic programming algorithms on tree decompositions, we use X_t here to denote the set of portal vertices.

Introduce: Next, consider the case that t is an introduce-node, and let t' denote the unique child of t. Suppose that $X_t = X_{t'} \cup \{v\}$, that is, $v \in V(G)$ is the vertex introduced at t. Note that we have $N_G(v) \cap V_t \subseteq X_{t'}$. We say a string $x \in \mathbb{A}^{X_t}$ *ρ-extends* a string $y \in \mathbb{A}^{X_{t'}}$ if

1. $x[\,v\,] = \rho_c$ for some $c \in \{0, \ldots, r_{\text{top}}\}$,
2. $c = |\{w \in N_G(v) \mid y[\,w\,] \in \mathbb{S}\}|$, and
3. $x[\,X_{t'}\,] = y[\,X_{t'}\,]$.

A string $x \in \mathbb{A}^{X_t}$ *σ-extends* a string $y \in \mathbb{A}^{X_{t'}}$ if

1. $x[\,v\,] = \sigma_d$ for some $d \in \{0, \ldots, s_{\text{top}}\}$,
2. $d = |\{w \in N_G(v) \mid y[\,w\,] \in \mathbb{S}\}|$,
3. $\vec{\sigma}(x)[\,w\,] = \vec{\sigma}(y)[\,w\,]$ for all $w \in X_{t'}$,
4. $\vec{w}(x)[\,w\,] = \vec{w}(y)[\,w\,]$ for all $w \in X_{t'} \setminus N_G(v)$, and
5. $\vec{w}(x)[\,w\,] = \vec{w}(y)[\,w\,] + 1$ for all $w \in X_{t'} \cap N_G(v)$.

Finally, a string $x \in \mathbb{A}^{X_t}$ *extends* a string $y \in \mathbb{A}^{X_{t'}}$ if it ρ-extends y or it σ-extends y. We have that

$$L_{t,i} = \{x \in \mathbb{A}^{X_t} \mid x \text{ extends some } y \in L_{t',i}\}.$$

Since, for every string $y \in \mathbb{A}^{X_{t'}}$, there are at most two strings $x \in \mathbb{A}^{X_t}$ such that x extends y, all of the languages $L_{t,i}$ can be computed by iterating over all sets $L_{t',i}$ once. This takes time $\tau^{\mathsf{tw}} \cdot (t_{\text{top}} + \mathsf{tw})^{O(1)}$ using Equation (4.14).

Join: Finally, suppose that t is a join-node, and let t_1, t_2 denote the two children of t. Observe that $X_t = X_{t_1} = X_{t_2}$.

To compute the sets $L_{t,i}$, we intend to rely on the algorithm from Theorem 4.18. However, this is not directly possible since the weight-vectors of the resulting strings would not be correct. Indeed, suppose that $x_1, x_2 \in \mathbb{A}^{X_t}$ are strings that are compatible with $(G[\,V_{t_1}\,], X_{t_1})$ and $(G[\,V_{t_2}\,], X_{t_2})$, respectively. Also, suppose that x_1 and x_2 can be joined. Then, $x_1 \oplus x_2$ is not (necessarily) compatible with $(G[\,V_t\,], X_t)$ since, for every $v \in X_t$, the vertices from $N_G(v) \cap X_t$ that are contained in the solution set are counted twice. For this reason, we first modify all the strings from the languages $L_{t_1,i}$ such that indices do not take solution vertices from the set X_t into account.

For each $i \in [\,0\,..\,\mathrm{m}\,)$ and each $x \in L_{t_1,i}$, we perform the following steps. We define the string $\widehat{x} \in \mathbb{A}^{X_t}$ as

$$\widehat{x}[\,v\,] \coloneqq \begin{cases} \rho_c & \text{if } x[\,v\,] = \rho_{c'} \text{ and } c' = c + |\{w \in N_G(v) \cap X_t \mid x[\,w\,] \in \mathbb{S}\}|, \\ \sigma_d & \text{if } x[\,v\,] = \sigma_{d'} \text{ and } d' = d + |\{w \in N_G(v) \cap X_t \mid x[\,w\,] \in \mathbb{S}\}| \end{cases}$$

and add $\widehat{x}$ to the set $L'_{t_1,i}$. Observe that, for each $i \in [\,0\,..\,\mathrm{m}\,)$, we can compute the set $L'_{t_1,i}$ in time $\tau^{\mathsf{tw}} \cdot (t_{\text{top}} + \mathsf{tw})^{O(1)}$ using Equation (4.14).

Now, we easily observe that

$$L_{t,i} = \bigcup_{j \in [\,0\,..\,\mathrm{m}\,)} L'_{t_1,j} \oplus L_{t_2,i-j},$$

where the index $i - j$ is taken modulo m. We iterate over all choices of $i, j \in [0\,..\,\mathrm{m})$ and compute $L'_{t_1,j} \oplus L_{t_2,i-j}$ using Theorem 4.18. To that end, we need to ensure that the requirements of Theorem 4.18 are satisfied. We have already argued that $L_{t_2,i-j} \times L_{t_2,i-j} \subseteq \mathcal{R}^{|X_t|}$. Further, $L'_{t_1,j}$ is realized by $(G[V_t] - E(X_t, X_t), X_t)$ (that is, the graph obtained from $G[V_t]$ by removing all edges within X_t), and hence, $L'_{t_1,j} \times L'_{t_1,j} \subseteq \mathcal{R}^{|X_t|}$ using Lemma 4.3.

Overall, this allows us to compute the join in time $\tau^{\mathsf{tw}} \cdot (t_{\mathrm{top}} + \mathsf{tw})^{O(1)}$ using Theorem 4.18.

Since processing a single node of T takes time $\tau^{\mathsf{tw}} \cdot (t_{\mathrm{top}} + \mathsf{tw})^{O(1)}$ and we have $|V(T)| = O(\mathsf{tw} \cdot |V(G)|)$, we see that all sets $L_{t,i}$ can be computed in the desired time.

To decide whether G has a (σ, ρ)-DomSet, we consider the root node $t \in V(T)$ for which $X_t = \varnothing$ and $V_t = V(G)$. Then, G has a (σ, ρ)-DomSet if and only if $\varepsilon \in L_{t,i}$ for some $i \in [0\,..\,\mathrm{m})$, which completes the proof. □

Next, we explain how to extend the algorithm to the optimization and counting version of the problem. For the optimization version, it is easy to see that we can keep track of the size of partial solutions in the dynamic programming tables. This increases the size of all tables by a factor of $|V(G)|$. Hence, we obtain the following theorem for the optimization version.

Theorem 4.28. *Let (σ, ρ) denote finite, m-structured sets for some $\mathrm{m} \geq 2$. Then, there is an algorithm $\mathcal{A}$ that, given a graph G, an integer k, and a nice tree decomposition of G of width tw, decides whether G has a (σ, ρ)-DomSet of size at most (at least) k.*

If $\mathrm{m} \geq 3$ or t_{top} is odd or $\min(r_{\mathrm{top}}, s_{\mathrm{top}}) < t_{\mathrm{top}}$, then algorithm $\mathcal{A}$ runs in time

$$(t_{\mathrm{top}} + 1)^{\mathsf{tw}} \cdot (t_{\mathrm{top}} + \mathsf{tw})^{O(1)} \cdot |V(G)|^2.$$

If $\mathrm{m} = 2$, t_{top} is even, and $r_{\mathrm{top}} = s_{\mathrm{top}} = t_{\mathrm{top}}$, then algorithm $\mathcal{A}$ runs in time

$$(t_{\mathrm{top}} + 2)^{\mathsf{tw}} \cdot (t_{\mathrm{top}} + \mathsf{tw})^{O(1)} \cdot |V(G)|^2.$$

For the counting version of the problem (that is, we wish to compute the number of solution sets), the situation is more complicated. The main challenge for the counting version stems from the application of Fact 4.19 for which we need to find an appropriate prime p as well as certain roots of unity. In Remark 4.20, we explained how to find these objects in time roughly linear in p (ignoring various lower-order terms that are not relevant to the discussion here). However, for the counting version, we would need p to be larger than the number of solutions, which results in a running time that is exponential in n in the worst case.

Luckily, we can circumvent this problem using the Chinese Remainder Theorem. The basic idea is to compute the number of solutions modulo p_i for a sufficiently large number of distinct small primes p_i. Assuming $\prod_i p_i > 2^n$, the number of solutions can be uniquely recovered using the Chinese Remainder Theorem.

Theorem 4.29 (Chinese Remainder Theorem)**.** *Let $m_1, \ldots, m_\ell$ denote a sequence of integers that are pairwise coprime, and define $M := \prod_{i \in [1\,..\,\ell]} m_i$. Also, let $0 \leq a_i < m_i$ for all $i \in [1\,..\,\ell]$. Then, there is a unique number $0 \leq s < M$ such that*

$$s \equiv a_i \pmod{m_i}$$

for all $i \in [1\,..\,\ell]$. Moreover, there is an algorithm that, given $m_1, \ldots, m_\ell$ and $a_1, \ldots, a_\ell$, computes the number s in time $O((\log M)^2)$.

More generally, we can build on the following extension of Fact 4.19 which may also be interesting in its own right.

Theorem 4.30. *Let $d_1, \dots, d_n \geq 2$, and let $D \coloneqq \prod_{i=1}^{n} d_i$. Also, let $f, g \colon \mathbb{Z}_{d_1} \times \cdots \times \mathbb{Z}_{d_n} \to \mathbb{Z}$ denote a function, and let $h \colon \mathbb{Z}_{d_1} \times \cdots \times \mathbb{Z}_{d_n} \to \mathbb{F}_p$ denote the convolution*

$$h(a) \coloneqq \sum_{a_1+a_2=a} f(a_1) \cdot g(a_2).$$

Moreover, let M denote a non-negative integer such that all images of f, g and h are contained in $\{0, \dots, M\}$. Then, the function h can be computed in time $D \cdot (\log D + n + \log M)^{O(1)}$.

Proof. Let $m \coloneqq \lceil \log M \rceil$. We compute the list of the first m primes $p_1 < \cdots < p_m$ such that $p_i \equiv 1 \pmod{D}$ for all $i \in [\![1 \mathinner{..} m]\!]$. By the Prime Number Theorem for Arithmetic Progressions (see, e.g., [5]) we get that $p_m = O(\varphi(D) \cdot m \cdot \log m)$, where φ denotes Euler's totient function. In particular, $p_m = O(D \cdot m \cdot \log m)$ because $\varphi(D) \leq D$. Since prime testing can be done in polynomial time, we can find the sequence $p_1, \dots, p_m$ in time $O(D \cdot m \cdot (\log m)^c)$ for some constant c.

Next, for every $i \in [\![1 \mathinner{..} m]\!]$ and $j \in [\![1 \mathinner{..} n]\!]$, we compute a d_j-th root of unity in $\mathbb{F}_{p_i}$ as follows. First, observe that such a root of unity exists since d_j divides $p_i - 1$. Now, we simply iterate over all elements $x \in \mathbb{F}_{p_i}$ and test whether a given element x is a d_j-th root of unity in time $(d_j + \log p_i)^{O(1)}$. So, overall, computing all roots of unity can be done in time

$$p_m \cdot (n + m + \log D)^{O(1)} = D \cdot (n + m + \log D)^{O(1)}.$$

Now, for every $i \in [\![1 \mathinner{..} m]\!]$ and $a \in \mathbb{Z}_{d_1} \times \cdots \times \mathbb{Z}_{d_n}$, we compute

$$h_i(a) \coloneqq h(a) \pmod{p_i}$$

using Fact 4.19 taking $O(m \cdot D \cdot \log D)$ many arithmic operations. Since each arithmetic operation can be done time $(\log p_m)^{O(1)}$, we obtain a total running time of

$$D \cdot (m + \log D)^{O(1)}.$$

Finally, we can recover all numbers $h(a)$ by the Chinese Remainder Theorem in time

$$O(D \cdot m^2).$$

Note that $\prod_{i \in [\![1 \mathinner{..} m]\!]} p_i > 2^m \geq M$, which implies that all numbers are indeed uniquely recovered. In total, this achieves the desired running time. □

Now, to obtain an algorithm for the counting version, we follow the algorithm from Theorem 4.1 and replace the application of Fact 4.19 by Theorem 4.30. Also, we change the definition of the functions f_i in Equation (4.13) to give the number of partial solutions. Note that we can set $M \coloneqq 2^n$ since the number of solutions is always bounded by 2^n. Additionally, similarly to the optimization version, we keep track of the size of solutions. Overall, we obtain the following theorem for the counting version.

Theorem 4.31. *Let (σ, ρ) denote finite, m-structured sets for some $\mathrm{m} \geq 2$. Then, there is an algorithm $\mathcal{A}$ that, given a graph G, an integer k, and a nice tree decomposition of G of width tw, computes the number of solution sets of size exactly k in G for (σ, ρ)-DomSet.*

If $\mathrm{m} \geq 3$ or t_{top} is odd or $\min(r_{\mathrm{top}}, s_{\mathrm{top}}) < t_{\mathrm{top}}$, then algorithm $\mathcal{A}$ runs in time

$$(t_{\mathrm{top}} + 1)^{\mathsf{tw}} \cdot (t_{\mathrm{top}} + \mathsf{tw} + |V(G)|)^{O(1)}.$$

If $\mathrm{m} = 2$, t_{top} is even, and $r_{\mathrm{top}} = s_{\mathrm{top}} = t_{\mathrm{top}}$, then algorithm $\mathcal{A}$ runs in time

$$(t_{\mathrm{top}} + 2)^{\mathsf{tw}} \cdot (t_{\mathrm{top}} + \mathsf{tw} + |V(G)|)^{O(1)}.$$

We obtain Theorem 1.3 by combining Theorems 1.1 and 4.31.

5 Faster Algorithms via Representative Sets

Next, we present a second algorithm for (σ,ρ)-DomSet which is designed for the decision version and the case that one of the sets σ,ρ is cofinite. More precisely, the aim of this section is to prove Theorem 1.6. Let us stress again that the algorithm given in this section works for all finite or cofinite sets σ,ρ, but in the case where both ρ and σ are finite, it is slower than existing algorithms (see Theorem 1.1). The algorithm is based on representative sets. Intuitively speaking, for a graph G and a set $U \subseteq V(G)$, the idea is to not compute the set entire set $L \subseteq \mathbb{A}^U$ of strings that are compatible with (G,U), but only a *representative set* $R \subseteq L$ such that, if there is a partial solution for some $x \in L$ that can be extended to a full solution via some $y \in \mathbb{A}^U$, then there is also a partial solution $x' \in R$ that can be extended to a full solution via y. If one of the sets σ,ρ is cofinite, then it is possible to obtain representative sets $R \subseteq L$ that are much smaller than the number of partial solutions that are maintained by standard dynamic programming algorithms (see, e.g., [48]).

For technical reasons, it turns out to be more convenient to work with the alphabet $\mathbb{A}_n = \{\rho_0,\dots,\rho_n,\sigma_0,\dots,\sigma_n\}$, where n denotes the number of vertices of the graph G under investigation.

To obtain the representative sets, we build on ideas that were already used in [38]. In the following, let $\omega < 2.37286$ denote the matrix multiplication exponent [2].

Let us first restrict ourselves to the case where both ρ and σ are cofinite. Let $k \geq 1$ and let $F_1,\dots,F_k \subseteq \mathbb{Z}_{\geq 0}$ denote finite sets of *forbidden elements*. Intuitively speaking, for a set $X \subseteq V(G)$ consisting of k vertices $v_1,\dots,v_k$, we set $F_i := \mathbb{Z}_{\geq 0} \setminus \sigma$ if v_i is selected into a partial solution, and $F_i := \mathbb{Z}_{\geq 0} \setminus \rho$ otherwise.

Definition 5.1. *The* compatibility graph for forbidden sets $F_1,\dots,F_k$ *is the infinite graph* $\mathcal{C} = \mathcal{C}(F_1,\dots,F_k)$ *with*

- $V(\mathcal{C}) := U^k \cup V^k$ *where* U,V *are disjoint sets both identified with* $\mathbb{Z}_{\geq 0}$*, and*
- $E(\mathcal{C}) := \{((a_1,\dots,a_k),(b_1,\dots,b_k)) \mid \forall i \in [1\,..\,k]\colon a_i + b_i \notin F_i\}$.

Let $\mathcal{S} \subseteq {\mathbb{Z}_{\geq 0}}^k$ *denote a finite set. We say that* $\mathcal{S}' \subseteq \mathcal{S}$ *is an* $(F_1,\dots,F_k)$-representative set of $\mathcal{S}$ *if, for every* $b \in {\mathbb{Z}_{\geq 0}}^k$*, we have that*

$$\exists a \in \mathcal{S}\colon (a,b) \in E(\mathcal{C}(F_1,\dots,F_k)) \iff \exists a' \in \mathcal{S}'\colon (a',b) \in E(\mathcal{C}(F_1,\dots,F_k)).$$

Now, the basic idea is that, for a set $X = \{v_1,\dots,v_k\}$ and a fixed σ-vector $\vec{s} \in \{0,1\}^{|X|}$, it suffices to keep an $(F_1,\dots,F_k)$-representative set of the weight vectors of those partial solutions that have σ-vector $\vec{s}$. Here, $F_i := \mathbb{Z}_{\geq 0} \setminus \sigma$ if $\vec{s}[\,v_i\,] = 1$, and $F_i := \mathbb{Z}_{\geq 0} \setminus \rho$ if $\vec{s}[\,v_i\,] = 0$.

Lemma 5.2 ([38, Lemma 3.3 & 3.6])**.** *Let* $F_1,\dots,F_k \subseteq \mathbb{Z}_{\geq 0}$ *denote finite sets such that* $|F_i| \leq t$ *for all* $i \in [1\,..\,k]$*. Further, let* $\mathcal{S} \subseteq {\mathbb{Z}_{\geq 0}}^k$ *denote a finite set. Then, one can compute an* $(F_1,\dots,F_k)$*-representative set* $\mathcal{S}'$ *of* $\mathcal{S}$ *such that* $|\mathcal{S}'| \leq (t+1)^k$ *in time* $O(|\mathcal{S}| \cdot (t+1)^{k(\omega-1)}k)$.

We can use Lemma 5.2 to compute representative sets of small size if both ρ and σ are cofinite. To also cover finite sets, we need to extend the above results as follows. Consider a $k,\ell \in \mathbb{Z}_{\geq 0}$ such that $k+\ell \geq 1$ and let $F_1,\dots,F_k,P_1,\dots,P_\ell \subseteq \mathbb{Z}_{\geq 0}$ denote finite sets of *forbidden elements* and *positive elements*. The basic intuition is similar to before. Consider a set $X \subseteq V(G)$ consisting of $k+\ell$ vertices $v_1,\dots,v_{k+\ell}$ and a fixed σ-vector $\vec{s} \in \{0,1\}^{|X|}$ that has k entries corresponding to an infinite set, and ℓ entries corresponding to a finite set (e.g., if σ is infinite and ρ is finite, then there are k many 1-entries since they correspond to the infinite set σ). With this intuition in mind, we generalize Definition 5.1 as follows.

Definition 5.3. *The* compatibility graph for forbidden sets $F_1, \ldots, F_k$ and positive sets $P_1, \ldots, P_\ell$ *is the infinite graph* $\mathcal{C} = \mathcal{C}(F_1, \ldots, F_k; P_1, \ldots, P_\ell)$ *with*

- $V(\mathcal{C}) \coloneqq U^{k+\ell} \cup V^{k+\ell}$ *where* U, V *are disjoint sets both identified with* $\mathbb{Z}_{\geq 0}$*, and*
- $E(\mathcal{C}) \coloneqq \{((a_1, \ldots, a_{k+\ell}), (b_1, \ldots, b_{k+\ell})) \mid \forall i \in [\![1 \mathinner{..} k]\!] \colon a_i + b_i \notin F_\ell \text{ and } \forall j \in [\![1 \mathinner{..} \ell]\!] \colon a_{k+j} + b_{k+j} \in P_j\}$.

Let $\mathcal{S} \subseteq {\mathbb{Z}_{\geq 0}}^{k+\ell}$ *denote a finite set. We say that* $\mathcal{S}' \subseteq \mathcal{S}$ *is an* $(F_1, \ldots, F_k; P_1, \ldots, P_\ell)$-representative set of $\mathcal{S}$ *if, for every* $b \in {\mathbb{Z}_{\geq 0}}^k$*, we have that*

$$\exists a \in \mathcal{S} \colon (a, b) \in E(\mathcal{C}) \iff \exists a' \in \mathcal{S}' \colon (a', b) \in E(\mathcal{C}).$$

By taking a brute-force approach to positions with positive sets, we can also generalize Lemma 5.2.

Lemma 5.4. *Let* $F_1, \ldots, F_k, P_1, \ldots, P_\ell \subseteq \mathbb{Z}_{\geq 0}$ *denote finite sets such that* $|F_i| \leq t$ *for all* $i \in [\![1 \mathinner{..} k]\!]$ *and* $\max(P_j) \leq t$ *for all* $j \in [\![1 \mathinner{..} \ell]\!]$*. Further, write* $\mathcal{S} \subseteq {\mathbb{Z}_{\geq 0}}^k$ *for a finite set. Then, one can compute an* $(F_1, \ldots, F_k; P_1, \ldots, P_\ell)$*-representative set* $\mathcal{S}'$ *of* $\mathcal{S}$ *such that* $|\mathcal{S}'| \leq (t+1)^{k+\ell}$ *in time* $O(|\mathcal{S}| \cdot (t+1)^{\ell + k(\omega-1)}(k+\ell))$.

Proof. We proceed in two steps. We first compute the set

$$\mathcal{S}'' \coloneqq \{(a_1, \ldots, a_{k+\ell}) \in \mathcal{S} \mid \forall j \in [\![1 \mathinner{..} \ell]\!] \colon a_{k+j} \leq t\}.$$

Clearly, the set $\mathcal{S}''$ can be computed in time $O(|\mathcal{S}| \cdot \ell)$. Also, every element $(a_1, \ldots, a_{k+\ell}) \in \mathcal{S} \setminus \mathcal{S}''$ is an isolated vertex in $\mathcal{C} = \mathcal{C}(F_1, \ldots, F_k; P_1, \ldots, P_\ell)$ because $\max(P_j) \leq t$ for all $j \in [\![1 \mathinner{..} \ell]\!]$. Hence, it suffices to compute a $(F_1, \ldots, F_k; P_1, \ldots, P_\ell)$-representative set of $\mathcal{S}_1$.

We say that two elements $(a_1, \ldots, a_{k+\ell}), (a_1, \ldots, a_{k+\ell}) \in \mathcal{S}''$ are *positive-equivalent* if $a_{k+j} = b_{k+j}$ for all $j \in [\![1 \mathinner{..} \ell]\!]$. Let $\mathcal{S}_1, \ldots, \mathcal{S}_p$ denote the equivalence classes of this relation. Note that, by the definition of the set $\mathcal{S}''$, we have $p \leq (t+1)^\ell$. We can compute the sets $\mathcal{S}_1, \ldots, \mathcal{S}_p$ in time $O(|\mathcal{S}| \cdot (t+1)^\ell (k+\ell))$. For each $i \in [\![1 \mathinner{..} p]\!]$, we can compute a $(F_1, \ldots, F_k; P_1, \ldots, P_\ell)$-representative set $\mathcal{S}_i'$ of $\mathcal{S}_i$ using Lemma 5.2 in time $O(|\mathcal{S}| \cdot (t+1)^{k(\omega-1)} k)$. Then, $|\mathcal{S}_i'| \leq (t+1)^k$. We define

$$\mathcal{S}' \coloneqq \bigcup_{i \in [\![1 \mathinner{..} p]\!]} \mathcal{S}_i'.$$

Observe that $|\mathcal{S}'| \leq p \cdot (t+1)^k \leq (t+1)^{k+\ell}$. Moreover, computing $\mathcal{S}'$ overall takes time $O(|\mathcal{S}| \cdot \ell + |\mathcal{S}| \cdot (t+1)^\ell (k+\ell) + p \cdot |\mathcal{S}| \cdot (t+1)^{k(\omega-1)} k) = O(|\mathcal{S}| \cdot (t+1)^{\ell + k(\omega-1)}(k+\ell))$. □

We have all the tools to present an algorithm for (σ, ρ)-DomSet on graphs of small tree-width based on representative sets. To state its running time, let us introduce the following cost measure for sets of natural numbers. We write $\varnothing \neq \tau \subseteq \mathbb{Z}_{\geq 0}$ for a finite or cofinite set. If τ is finite, then we define $\mathrm{cost}(\tau) \coloneqq \max(\tau)$. Otherwise, τ is cofinite and we define $\mathrm{cost}(\tau) \coloneqq |\mathbb{Z}_{\geq 0} \setminus \tau|$.

Theorem 5.5 (Theorem 1.6 restated)**.** *Suppose* $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ *are finite or cofinite. Also, let* $t_{\mathrm{cost}} \coloneqq \max(\mathrm{cost}(\rho), \mathrm{cost}(\sigma))$*. Then, there is an algorithm* $\mathcal{A}$ *that, given a graph* G *and a nice tree decomposition of* G *of width* tw*, decides whether* G *has a* (σ, ρ)-DomSet *in time*

$$2^{\mathsf{tw}} \cdot (t_{\mathrm{cost}} + 1)^{\mathsf{tw}(\omega+1)} \cdot (t_{\mathrm{cost}} + \mathsf{tw})^{O(1)} \cdot |V(G)|.$$

Before we dive into the proof, let us again compare the running times from this algorithm and the existing algorithm by van Rooij (Theorem 1.1). To this end, we define a modified cost measure. Write $\tau \subseteq \mathbb{Z}_{\geq 0}$ for a finite or cofinite set. If τ is finite, then we define $\mathrm{cost}'(\tau) \coloneqq \max(\tau)$. Otherwise, τ is cofinite and we define $\mathrm{cost}'(\tau) \coloneqq \max(\mathbb{Z}_{\geq 0} \setminus \tau) + 1$ if $\mathbb{Z}_{\geq 0} \setminus \tau \neq \varnothing$, and $\mathrm{cost}'(\mathbb{Z}_{\geq 0}) = 0$. Observe the difference in the definition for cofinite sets. Also, note that $\mathrm{cost}(\tau) \leq \mathrm{cost}'(\tau)$ for all finite or cofinite sets $\tau \subseteq \mathbb{Z}_{\geq 0}$. Moreover, for a cofinite set $\tau \subseteq \mathbb{Z}_{\geq 0}$, we have that $\mathrm{cost}(\tau) = \mathrm{cost}'(\tau)$ if and only if $\tau = \{c, c+1, c+2, \dots\}$ for some number $c \in \mathbb{Z}_{\geq 0}$.

Using this cost measure, the running time of the algorithm from Theorem 1.1 is

$$\Big(\mathrm{cost}'(\rho) + \mathrm{cost}'(\sigma) + 2\Big)^{\mathsf{tw}} n^{O(1)}.$$

On the other hand, the algorithm from Theorem 5.5 runs in time

$$\Big(2 \cdot \big(\max(\mathrm{cost}(\rho), \mathrm{cost}(\sigma)) + 1\big)^{\omega+1}\Big)^{\mathsf{tw}} n^{O(1)}.$$

If σ and ρ are both finite, then the algorithm by van Rooij is clearly faster using that $\mathrm{cost}'(\rho) = \mathrm{cost}(\rho)$ and $\mathrm{cost}'(\sigma) = \mathrm{cost}(\sigma)$ (but, in this case, Theorem 4.1 provides an improved algorithm for structured sets). However, if one the sets σ, ρ is cofinite, then our algorithm may be substantially faster since $\mathrm{cost}(\tau)$ can be arbitrarily smaller than $\mathrm{cost}'(\tau)$ for a cofinite set τ. As a concrete example, suppose that $\rho = \mathbb{Z}_{\geq 0} \setminus \{c\}$ and $\sigma = \mathbb{Z}_{\geq 0} \setminus \{d\}$. Then, $\mathrm{cost}(\rho) = \mathrm{cost}(\sigma) = 1$, but $\mathrm{cost}'(\rho) = c + 1$ and $\mathrm{cost}'(\sigma) = d + 1$. Hence, van Rooij's algorithm runs in time $(c + d + 4)^{\mathsf{tw}} n^{O(1)}$, where the algorithm from Theorem 5.5 takes time $2^{\mathsf{tw}(\omega+2)} n^{O(1)}$ which is at most $20.72^{\mathsf{tw}} n^{O(1)}$. Observe that the second running time is independent of c and d.

Proof. Let (T, β) denote the nice tree decomposition of G and suppose $n = |V(G)|$. For ease of notation, let us set $\mathbb{A} \coloneqq \mathbb{A}_n = \{\rho_0, \dots, \rho_n, \sigma_0, \dots, \sigma_n\}$ for the remainder of this proof. For $t \in V(T)$, we denote $X_t \coloneqq \beta(t)$ and V_t the set of vertices contained in bags below t (including t itself). For each node $t \in V(T)$ and each $\vec{s} \in \{0,1\}^{X_t}$ we denote by $L_{t,\vec{s}} \subseteq \mathbb{A}^{X_t}$ the set of all strings $x \in \mathbb{A}^{X_t}$ that are compatible with $(G[V_t], X_t)$ and $\vec{\sigma}(x) = \vec{s}$.

Now, let us fix some $t \in V(T)$ and $\vec{s} \in \{0,1\}^{X_t}$. Also, suppose that $v \in X_t$. We say that v is an *F-position* if $\vec{s}[v] = 1$ and σ is cofinite, or $\vec{s}[v] = 0$ and ρ is cofinite. In the former case, we define $F_v \coloneqq \mathbb{Z}_{\geq 0} \setminus \sigma$, and in the latter case we define $F_v \coloneqq \mathbb{Z}_{\geq 0} \setminus \rho$. If v is not an *F-position*, then we say that v is a *P-position*. Note that v is a *P-position* if $\vec{s}[v] = 1$ and σ is finite, or $\vec{s}[v] = 0$ and ρ is finite. In the former case, we define $P_v \coloneqq \sigma$, and in the latter case we define $P_v \coloneqq \rho$. By ordering elements in X_t accordingly, we may assume that $X_t = \{v_1, \dots, v_k, v_{k+1}, \dots, v_{k+\ell}\}$ such that $v_1, \dots, v_k$ are F-positions and $v_{k+1}, \dots, v_{k+\ell}$ are P-positions.

For two words $x, y \in \mathbb{A}^{X_t}$ such that $\vec{\sigma}(x) = \vec{\sigma}(y) = \vec{s}$, we write $x \sim_{t,\vec{s}} y$ if

$$(\vec{w}(x), \vec{w}(y)) \in E(\mathcal{C}(F_{v_1}, \dots, F_{v_k}; P_{v_{k+1}}, \dots, P_{v_{k+\ell}})).$$

We say that a set $R_{t,\vec{s}} \subseteq L_{t,\vec{s}}$ is a *$(t,\vec{s})$-representative set of $L_{t,\vec{s}}$* if

$$\mathcal{S}' \coloneqq \{\vec{w}(x) \mid x \in R_{t,\vec{s}}\}$$

is an $(F_{v_1}, \dots, F_{v_k}; P_{v_{k+1}}, \dots, P_{v_{k+\ell}})$-representative set of

$$\mathcal{S} \coloneqq \{\vec{w}(x) \mid x \in L_{t,\vec{s}}\}.$$

Equivalently, $R_{t,\vec{s}} \subseteq L_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$ if, for every $y \in \mathbb{A}^{X_t}$ such that $\vec{\sigma}(y) = \vec{s}$, we have

$$\exists x \in L_{t,\vec{s}} \colon x \sim_{t,\vec{s}} y \quad \Longleftrightarrow \quad \exists x' \in R_{t,\vec{s}} \colon x' \sim_{t,\vec{s}} y.$$

The algorithm computes, for each $t \in V(T)$ and $\vec{s} \in \{0,1\}^{X_t}$, a $(t,\vec{s})$-representative set $R_{t,\vec{s}}$ of $L_{t,\vec{s}}$ such that $|R_{t,\vec{s}}| \leq (t_{\text{cost}}+1)^{|X_t|}$. To compute these sets we proceed in a bottom-up fashion starting at the leaves of T. Suppose t is a leaf of T. Then, $X_t = V_t = \varnothing$ and $L_{t,\vec{s}} = \{\varepsilon\}$. We set $R_{t,\vec{s}} \coloneqq \{\varepsilon\}$.

Next, let t denote an internal node and suppose the algorithm already computed all sets $R_{t',\vec{s}}$ for all children t' of t.

Forget: First, suppose t is a forget-node and write t' for the unique child of t. Also, assume that $X_{t'} = X_t \cup \{v\}$, i.e., $v \in V(G)$ is the vertex forgotten at t. We say that a string $x \in \mathbb{A}^{X_{t'}}$ *is happy at* v if $x[\,v\,] = \rho_c$ and $c \in \rho$, or $x[\,v\,] = \sigma_d$ and $d \in \sigma$. Fix some $\vec{s} \in \{0,1\}^{X_t}$. For $i \in \{0,1\}$, we write $\vec{s}_i \in \{0,1\}^{X_t \cup \{v\}}$ for the extension of $\vec{s}$ for which $\vec{s}[\,v\,] = i$. It is easy to see that

$$\begin{aligned} L_{t,\vec{s}} = \ & \{x[\,X_t\,] \mid x \in L_{t',\vec{s}_0} \text{ such that } x \text{ is happy at } v\} \\ & \cup \{x[\,X_t\,] \mid x \in L_{t',\vec{s}_1} \text{ such that } x \text{ is happy at } v\}. \end{aligned}$$

We set

$$\begin{aligned} \widehat{R}_{t,\vec{s}} = \ & \{x[\,X_t\,] \mid x \in R_{t',\vec{s}_0} \text{ such that } x \text{ is happy at } v\} \\ & \cup \{x[\,X_t\,] \mid x \in R_{t',\vec{s}_1} \text{ such that } x \text{ is happy at } v\}. \end{aligned}$$

We show that $\widehat{R}_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$. Let $x \in L_{t,\vec{s}}$ and $y \in \mathbb{A}^{X_t}$ such that $\vec{\sigma}(y) = \vec{s}$ and $x \sim_{t,\vec{s}} y$. We need to show that there is some $x' \in \widehat{R}_{t,\vec{s}}$ such that $x' \sim_{t,\vec{s}} y$. Let $z \in L_{t',\vec{s}_i}$ such that z is happy at v and $x = z[\,X_t\,]$. Let $y_i \in \mathbb{A}^{X_{t'}}$ such that $\vec{\sigma}(y_i) = \vec{s}_i$ and $\vec{w}(y_i)[\,v\,] = 0$ and $\vec{w}(y_i)[\,w\,] = \vec{w}(y)[\,w\,]$ for all $w \in X_t$. Then, $z \sim_{t',\vec{s}_i} y_i$. So, there is some $z' \in R_{t',\vec{s}_i}$ such that $z' \sim_{t',\vec{s}_i} y_i$. Observe that z' is happy at v since $\vec{w}(y_i)[\,v\,] = 0$. We set $x' \coloneqq z'[\,X_t\,]$. Then, $x' \in \widehat{R}_{t,\vec{s}}$ and $x' \sim_{t,\vec{s}} y$ as desired.

Observe that $\widehat{R}_{t,\vec{s}}$ can be computed in time $O((t_{\text{cost}}+1)^{|X_{t'}|} \cdot |X_t|) = O((t_{\text{cost}}+1)^{\mathsf{tw}} \cdot (t_{\text{cost}} + \mathsf{tw})^{O(1)})$.

Finally, we obtain $R_{t,\vec{s}}$ by computing a $(t,\vec{s})$-representative set of $\widehat{R}_{t,\vec{s}}$ using Lemma 5.4. Note that $|R_{t,\vec{s}}| \leq (t_{\text{cost}}+1)^{|X_t|}$ as desired. Also, $R_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$ since $\widehat{R}_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$. This step takes time

$$\begin{aligned} & O(|\widehat{R}_{t,\vec{s}}| \cdot (t_{\text{cost}}+1)^{|X_t|(\omega-1)} \cdot |X_t|) \\ &= (t_{\text{cost}}+1)^{\mathsf{tw}} \cdot (t_{\text{cost}}+1)^{\mathsf{tw}(\omega-1)} \cdot (t_{\text{cost}} + \mathsf{tw})^{O(1)} \\ &= (t_{\text{cost}}+1)^{\mathsf{tw}\cdot\omega} \cdot (t_{\text{cost}} + \mathsf{tw})^{O(1)}. \end{aligned}$$

So overall, computing $R_{t,\vec{s}}$ for every $\vec{s} \in \{0,1\}^{X_t}$ takes time $2^{\mathsf{tw}} \cdot (t_{\text{cost}}+1)^{\mathsf{tw}\cdot\omega} \cdot (t_{\text{cost}} + \mathsf{tw})^{O(1)}$.

Introduce: Next consider the case that t is an introduce-node and write t' for the unique child of t. Suppose that $X_t = X_{t'} \cup \{v\}$, that is, $v \in V(G)$ is the vertex introduced at t. Note that $N_G(v) \cap V_t \subseteq X_{t'}$. We say a string $x \in \mathbb{A}^{X_t}$ *ρ-extends* a string $y \in \mathbb{A}^{X_{t'}}$ if

1. $x[\,v\,] = \rho_c$ for some $c \in \mathbb{Z}_{\geq 0}$,
2. $c = |\{w \in N_G(v) \mid y[\,w\,] \in \mathbb{S}\}|$, and
3. $x[\,X_{t'}\,] = y[\,X_{t'}\,]$.

A string $x \in \mathbb{A}^{X_t}$ *σ-extends* a string $y \in \mathbb{A}^{X_{t'}}$ if

1. $x[\,v\,] = \sigma_d$ for some $d \in \mathbb{Z}_{\geq 0}$,
2. $d = |\{w \in N_G(v) \mid y[\,w\,] \in \mathbb{S}\}|$,
3. $\vec{\sigma}(x)[\,w\,] = \vec{\sigma}(y)[\,w\,]$ for all $w \in X_{t'}$,
4. $\vec{w}(x)[\,w\,] = \vec{w}(y)[\,w\,]$ for all $w \in X_{t'} \setminus N_G(v)$, and
5. $\vec{w}(x)[\,w\,] = \vec{w}(y)[\,w\,] + 1$ for all $w \in X_{t'} \cap N_G(v)$.

Finally, a string $x \in \mathbb{A}^{X_t}$ *extends* a string $y \in \mathbb{A}^{X_{t'}}$ if it ρ-extends y or it σ-extends y.

Now, fix some $\vec{s} \in \{0,1\}^{X_t}$. We have that

$$L_{t,\vec{s}} = \{x \in \mathbb{A}^{X_t} \mid x \text{ extends some } y \in L_{t',\vec{s}[\,X_{t'}\,]}\}.$$

Similarly to the previous case, the algorithm computes

$$\widehat{R}_{t,\vec{s}} \coloneqq \{x \in \mathbb{A}^{X_t} \mid x \text{ extends some } y \in R_{t',\vec{s}[\,X_{t'}\,]}\}.$$

Since, for every string $y \in \mathbb{A}^{X_{t'}}$, there are at most two strings $x \in \mathbb{A}^{X_t}$ such that x extends y, this can be done in time $(t_{\mathrm{cost}} + 1)^{\mathsf{tw}} \cdot (t_{\mathrm{cost}} + \mathsf{tw})^{O(1)}$.

Again, we claim that $\widehat{R}_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$. Let $x \in L_{t,\vec{s}}$ and $y \in \mathbb{A}^{X_t}$ such that $\vec{\sigma}(y) = \vec{s}$ and $x \sim_{t,\vec{s}} y$. We need to show that there is some $x' \in \widehat{R}_{t,\vec{s}}$ such that $x' \sim_{t,\vec{s}} y$. Let $z \in L_{t',\vec{s}[\,X_{t'}\,]}$ such that x extends z.

First, suppose that x ρ-extends z, i.e., $\vec{s}[\,v\,] = 0$. Let $y' \coloneqq y[\,X_{t'}\,]$. Then, $z \sim_{t',\vec{s}[\,X_{t'}\,]} y'$ since $x[\,X_{t'}\,] = y[\,X_{t'}\,]$. So, there is some $z' \in R_{t',\vec{s}[\,X_{t'}\,]}$ such that $z' \sim_{t',\vec{s}[\,X_{t'}\,]} y'$. We define $x' \in \mathbb{A}^{X_t}$ such that $x'[\,X_{t'}\,] = z'$ and $x'[\,v\,] = x[\,v\,]$. Then, x' ρ-extends z', and hence, $x' \in \widehat{R}_{t,\vec{s}}$. Also, $x' \sim_{t,\vec{s}} y$ as desired.

Otherwise, x σ-extends z, that is, $\vec{s}[\,v\,] = 1$. Define $y' \in \mathbb{A}^{X_{t'}}$ via

1. $\vec{\sigma}(y') = \vec{s}[\,X_{t'}\,]$,
2. $\vec{w}(y')[\,w\,] = \vec{w}(y)[\,w\,]$ for all $w \in X_{t'} \setminus N_G(v)$, and
3. $\vec{w}(y')[\,w\,] = \vec{w}(y)[\,w\,] + 1$ for all $w \in X_{t'} \cap N_G(v)$.

Then, $z \sim_{t',\vec{s}[\,X_{t'}\,]} y'$. So, there is some $z' \in R_{t',\vec{s}[\,X_{t'}\,]}$ such that $z' \sim_{t',\vec{s}[\,X_{t'}\,]} y'$. We define $x' \in \mathbb{A}^{X_t}$ such that

1. $x'[\,v\,] = x[\,v\,]$,
2. $\vec{\sigma}(x') = \vec{s}$,
3. $\vec{w}(x')[\,w\,] = \vec{w}(z')[\,w\,]$ for all $w \in X_{t'} \setminus N_G(v)$, and
4. $\vec{w}(x')[\,w\,] = \vec{w}(z')[\,w\,] + 1$ for all $w \in X_{t'} \cap N_G(v)$.

Then, x' σ-extends z', and hence, $x' \in \widehat{R}_{t,\vec{s}}$. Also, $x' \sim_{t,\vec{s}} y$ as desired.

So, overall, $\widehat{R}_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$. We obtain $R_{t,\vec{s}}$ by computing a $(t,\vec{s})$-representative set of $\widehat{R}_{t,\vec{s}}$ using Lemma 5.4. Note that $|R_{t,\vec{s}}| \leq (t_{\mathrm{cost}} + 1)^{|X_t|}$ as desired. Also,

$R_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$ since $\widehat{R}_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$. This step takes time

$$\begin{aligned}&O(|\widehat{R}_{t,\vec{s}}| \cdot (t_{\text{cost}}+1)^{|X_t|(\omega-1)} \cdot |X_t|)\\ &= (t_{\text{cost}}+1)^{\mathsf{tw}} \cdot (t_{\text{cost}}+1)^{\mathsf{tw}(\omega-1)} \cdot (t_{\text{cost}}+\mathsf{tw})^{O(1)}\\ &= (t_{\text{cost}}+1)^{\mathsf{tw}\cdot\omega} \cdot (t_{\text{cost}}+\mathsf{tw})^{O(1)}.\end{aligned}$$

So, in total, computing $R_{t,\vec{s}}$ for every $\vec{s} \in \{0,1\}^{X_t}$ takes time $2^{\mathsf{tw}} \cdot (t_{\text{cost}}+1)^{\mathsf{tw}\cdot\omega} \cdot (t_{\text{cost}}+\mathsf{tw})^{O(1)}$.

Join: Finally, suppose that t is a join-node and write t_1, t_2 for the two children of t. Note that $X_t = X_{t_1} = X_{t_2}$.

Again, let us fix some $\vec{s} \in \{0,1\}^{X_t}$. For two strings $x, y \in \mathbb{A}^{X_t}$ such that $\vec{\sigma}(x) = \vec{\sigma}(y) = \vec{s}$, we define the string $x \oplus_{\vec{s}} y$ via

$$(x \oplus_{\vec{s}} y)[v] = \begin{cases} \rho_c & \text{if } x[v] = \rho_{c_1}, y[v] = \rho_{c_2}, c = c_1 + c_2 - |\{w \in N_G(v) \cap X_t \mid \vec{s}[w] = 1\}| \\ \sigma_d & \text{if } x[v] = \sigma_{d_1}, y[v] = \sigma_{d_2}, d = d_1 + d_2 - |\{w \in N_G(v) \cap X_t \mid \vec{s}[w] = 1\}| \end{cases}$$

Now,

$$L_{t,\vec{s}} = \{x_1 \oplus_{\vec{s}} x_2 \mid x_1 \in L_{t_1,\vec{s}}, x_2 \in L_{t_2,\vec{s}}\}.$$

Again, the algorithm computes

$$\widehat{R}_{t,\vec{s}} \coloneqq \{x \oplus_{\vec{s}} y \mid x \in R_{t_1,\vec{s}}, y \in R_{t_2,\vec{s}}\}.$$

This can be done in time $|R_{t_1,\vec{s}}| \cdot |R_{t_1,\vec{s}}| \cdot (t_{\text{cost}}+\mathsf{tw})^{O(1)} = (t_{\text{cost}}+1)^{2\mathsf{tw}} \cdot (t_{\text{cost}}+\mathsf{tw})^{O(1)}$.

As before, we claim that $\widehat{R}_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$. Let $x \in L_{t,\vec{s}}$ and $y \in \mathbb{A}^{X_t}$ such that $\vec{\sigma}(y) = \vec{s}$ and $x \sim_{t,\vec{s}} y$. We need to show that there is some $x' \in \widehat{R}_{t,\vec{s}}$ such that $x' \sim_{t,\vec{s}} y$. We have that $x = x_1 \oplus_{\vec{s}} x_2$ for some $x_1 \in L_{t_1,\vec{s}}$ and $x_2 \in L_{t_2,\vec{s}}$. Let $y_1 \coloneqq y \oplus_{\vec{s}} x_2$. Then, $x_1 \sim_{t_1,\vec{s}} y_1$, and hence, there is some $x_1' \in R_{t_1,\vec{s}}$ such that $x_1' \sim_{t_1,\vec{s}} y_1$. Let $x'' \coloneqq x_1' \oplus_{\vec{s}} x_2$. Then, $x'' \sim_{t,\vec{s}} y$. By applying the same argument to the second term, we obtain an $x' \in \widehat{R}_{t,\vec{s}}$ such that $x' \sim_{t,\vec{s}} y$.

So, $\widehat{R}_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$. As usual, we obtain $R_{t,\vec{s}}$ by computing a $(t,\vec{s})$-representative set of $\widehat{R}_{t,\vec{s}}$ using Lemma 5.4. Note that $|R_{t,\vec{s}}| \leq (t_{\text{cost}}+1)^{|X_t|}$ as desired. Also, $R_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$ since $\widehat{R}_{t,\vec{s}}$ is a $(t,\vec{s})$-representative set of $L_{t,\vec{s}}$. This step takes time

$$\begin{aligned}&O(|R_{t_1,\vec{s}}| \cdot |R_{t_2,\vec{s}}| \cdot (t_{\text{cost}}+1)^{|X_t|(\omega-1)} \cdot |X_t|)\\ &= (t_{\text{cost}}+1)^{2\mathsf{tw}} \cdot (t_{\text{cost}}+1)^{\mathsf{tw}(\omega-1)} \cdot (t_{\text{cost}}+\mathsf{tw})^{O(1)}\\ &= (t_{\text{cost}}+1)^{\mathsf{tw}\cdot(\omega+1)} \cdot (t_{\text{cost}}+\mathsf{tw})^{O(1)}.\end{aligned}$$

So, in total, computing $R_{t,\vec{s}}$ for every $\vec{s} \in \{0,1\}^{X_t}$ takes time $2^{\mathsf{tw}} \cdot (t_{\text{cost}}+1)^{\mathsf{tw}\cdot(\omega+1)} \cdot (t_{\text{cost}}+\mathsf{tw})^{O(1)}$.

Since processing a single node of T takes time $2^{\mathsf{tw}} \cdot (t_{\text{cost}}+1)^{\mathsf{tw}(\omega+1)} \cdot (t_{\text{cost}}+\mathsf{tw})^{O(1)}$ and $|V(T)| = O(\mathsf{tw} \cdot |V(G)|)$, it follows that all sets $L_{t,\vec{s}}$ can be computed in the desired time.

To decide whether G has a (σ,ρ)-DomSet, the algorithm considers the root node $t \in V(T)$ for which $X_t = \varnothing$ and $V_t = V(G)$. Then, G has a (σ,ρ)-DomSet if and only if $\varepsilon \in R_{t,\vec{s}}$, where $\vec{s}$ denotes the empty vector. □

Similarly to the previous section, we can also obtain an algorithm for the optimization version by incorporating the size of solution sets.

Theorem 5.6. *Suppose $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ are finite or cofinite. Also, let $t_{\mathrm{cost}} \coloneqq \max(\mathrm{cost}(\rho), \mathrm{cost}(\sigma))$. Then, there is an algorithm $\mathcal{A}$ that, given a graph G, an integer k, and a nice tree decomposition of G of width tw, decides whether G has a (σ, ρ)-DomSet of size at most (at least) k in time*

$$2^{\mathsf{tw}} \cdot (t_{\mathrm{cost}} + 1)^{\mathsf{tw}(\omega+1)} \cdot (t_{\mathrm{cost}} + \mathsf{tw})^{O(1)} \cdot |V(G)|^2.$$

However, in contrast to the previous section, the representative set approach cannot be extended to the counting version of the problem. Note that this is by design, since the fundamental idea of this approach is to not keep all the partial solutions which would be necessary for the counting version. We shall see in the following sections that this is actually not a shortcoming of our algorithmic approach, but holds in general assuming #SETH.

Part II
Lower Bounds

6 High-level Constructions for Proving Lower Bounds for GenDomSet

In this part, we prove the advertised lower bounds for (σ, ρ)-DomSet. More precisely, we formally prove Theorems 1.4 and 1.5.

For the decision problem, we divide the proof into three parts. We first construct a special type of gadget (later called *manager*) which allows us to give certain numbers of (selected) neighbors to vertices. In this step, we exploit the properties of σ and ρ to show different versions of the manager which results in stronger or weaker lower bounds. Once we have said managers at hand, the second step is a lower bound for a generalization of (σ, ρ)-DomSet where we additionally allow relations to be present in the graph (they generalize the constraints enforced by σ and ρ). The third and last step is to show how to remove the relations, while preserving their properties, such that we get a lower bound for (σ, ρ)-DomSet.

For the counting problem, we follow essentially the same outline. In fact, we can reuse some results for the decision problem: we show that some constructions directly transfer to the case when we count the number of solutions. An overview of the steps to obtain the lower bounds can be found in Figure 6.1.

Before starting with the proof, we first need to define some notation in order to formally state our results.

Definition 6.1 (Graph with Relations)**.** *A* graph with relations *is a tuple $G = (V_S, V_C, E, (R_v)_{v \in V_C})$ where $(V_S \cup V_C, E)$ forms a graph and $R_v \subseteq 2^{N(v)}$ for every $v \in V_C$. We call the vertices in V_S* simple*, and the vertices in V_C* complex*. Slightly abusing notation, we usually do not distinguish between G and its underlying graph $(V_S \cup V_C, E)$, and use G to refer to both objects depending on the context.*

Given a graph with relations $G = (V_S, V_C, E, (R_v)_{v \in V_C})$, a (σ, ρ)-set *of G is a set $S \subseteq V_S$ such that S is a (σ, ρ)-set of the underlying graph $(V_S \cup V_C, E)$, and $S \cap N(v) \in R_v$ for every $v \in V_C$.*

The treewidth of a graph with relations *G, is the treewidth of the graph G' obtained from G by turning each complex vertex and its neighborhood into a clique. Formally, we define $G' := (V_S \cup V_C, E')$ where $E' := E \cup \bigcup_{v \in V_C} \{(u, w) \mid u, w \in N[v], u \neq w\}$.*

Note that when using graphs with relations in a treewidth preserving reduction (see Definition 3.1), this new variant of treewidth is used instead of the "normal" treewidth.

As a first example of a graph with relations, we define a special gadget, called *manager*, which allows us to add selected neighbors to possibly selected vertices. In fact, a manager additionally ensures that the vertices to which it is connected get a valid number of neighbors such that the σ-constraints and ρ-constraints are always satisfied.

Definition 6.2 (A-manager)**.** *Given $A \subseteq \mathbb{A}$, an* A-manager *is an infinite family $((G_\ell, U_\ell))_{\ell \geq 1}$ of pairs (G_ℓ, U_ℓ) where G_ℓ is a graph with relations and $U_\ell = \{u_1, \ldots, u_\ell\} \subseteq V(G_\ell)$ is a set of ℓ distinguished simple vertices. Moreover, there is some non-negative integer b (depending only on $s_{\text{top}}, r_{\text{top}}$) such that the following holds for every $\ell \geq 1$:*

- *The vertices from $V(G_\ell) \setminus U_\ell$ can be partitioned into 2ℓ vertex-disjoint subgraphs $B_1, \ldots, B_\ell$ and $\overline{B}_1, \ldots, \overline{B}_\ell$, which we refer to as* blocks*, such that*

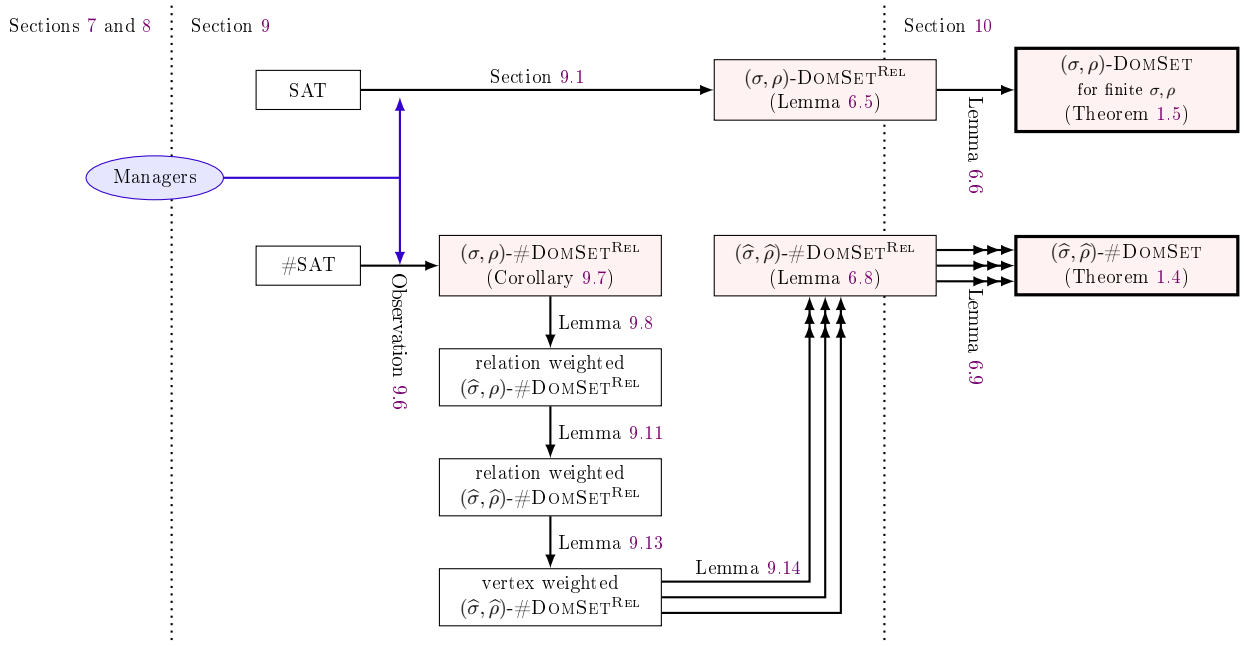


Figure 6.1: Outline of the proof of the lower bounds. The upper sequence considers the decision version. The lower sequence considers the counting version and includes the chain of reductions behind the proof of Lemma 6.8.
For the problems in the filled boxes, the statement refers to the formal lower bound. Multiple parallel edges indicate Turing-reductions.
The sets σ, ρ are finite or simple cofinite sets unless mentioned otherwise. The sets $\widehat{\sigma}$ and $\widehat{\rho}$ are finite or cofinite sets (not necessarily simple cofinite).

  - $|B_i| \leq b$ *and* $|\overline{B}_i| \leq b$ *for all* $i \in [1..\ell]$,
  - $N(u_i) \subseteq B_i \cup \overline{B}_i$ *for all* $i \in [1..\ell]$,
  - *there are edges only between the following pairs of blocks:* B_i *and* B_{i+1}, $\overline{B}_i$ *and* $\overline{B}_{i+1}$, *for each* $i \in [1..\ell-1]$, *and* B_ℓ *and* $\overline{B}_\ell$.

- *Each* $x \in A^\ell \subseteq \mathbb{A}^\ell$ *is* managed *in the sense that there is a unique* (σ,ρ)*-set* S_x *of* G_ℓ[7] *such that for all* $i \in [1..\ell]$*:*
  - *If* $x[i] = \sigma_s$, *then* $u_i \in S_x$. *Moreover,* u_i *has exactly* s *neighbors in* $B_i \cap S_x$, *and exactly* $s_{\text{top}} - s$ *neighbors in* $\overline{B}_i \cap S_x$.
  - *If* $x[i] = \rho_r$, *then* $u_i \notin S_x$. *Moreover,* u_i *has exactly* r *neighbors in* $B_i \cap S_x$, *and exactly* $r_{\text{top}} - r$ *neighbors in* $\overline{B}_i \cap S_x$.

We refer to G_ℓ *as the* A*-manager of rank* ℓ.

Recall that the first step for the lower bound is to show that there are managers. Formally, this is achieved by the following lemma.

[7] Note that all vertices in G_ℓ already have a feasible number of neighbors. The graph does not have any potentially "unsatisfied" portals. In this sense, it is different from the graphs with portals that we use in other gadget constructions.

Lemma 6.3 (Existence of Managers). *Let σ and ρ be two finite or cofinite sets of non-negative integers with $\rho \neq \{0\}$.*

1. *There is an A-manager with $|A| = r_{\text{top}} + 1$.*
2. *If (σ, ρ) is 2-structured, but not m-structured for any $\mathrm{m} \geq 3$, and s_{top} and $r_{\text{top}} \geq 1$ are even, then there is an A-manager with $|A| = (s_{\text{top}} + r_{\text{top}})/2 + 2$.*
3. *If $s_{\text{top}} \geq r_{\text{top}} \geq 1$, then there is an A-manager with $|A| = s_{\text{top}} + 1$.*
4. *If (σ, ρ) is not m-structured for any $\mathrm{m} \geq 2$, then there is an A-manager with $|A| = s_{\text{top}} + r_{\text{top}} + 2$.*

Moreover, each A-manager is such that A is closed under the inverse with respect to σ, ρ.

Section 8 is dedicated to the proof of Lemma 6.3. We prove each case separately; the proof of Lemma 6.3 then follows from Lemmas 8.1, 8.2, 8.5 and 8.6.

Observe that the bounds for the managers precisely coincide with the bounds for the languages provided in Example 4.5. Hence, these managers are the key ingredient to obtain matching lower bounds.

6.1 Decision Problem

While the first step is the same for the decision problem and for the counting problem, the remaining two steps need problem specific results. Before this, we define the intermediate problem for which we show a lower bound as the next step.

Definition 6.4 ((σ,ρ)-DomSet$^{\text{Rel}}$). *In the (σ,ρ)-Dominating Set w. Relations ((σ,ρ)-DomSet$^{\text{Rel}}$) problem, we are given a graph with relations $G = (V_S, V_C, E, (R_v)_{v \in V_C})$, and the task is to decide whether there is a (σ,ρ)-set of G.*

We define the counting version (σ,ρ)-#DomSet$^{\text{Rel}}$ analogously.

As mentioned before, our lower bound depends on the "quality" of the manager. Thus, we treat the manager as input and prove a lower bound based on it. Although a manager is a graph with relations, this assumption is useful as the intermediate problem has relations too.

Lemma 6.5 (Lower Bound for (σ,ρ)-DomSet$^{\text{Rel}}$). *Let $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ be two fixed, non-empty, and finite or simple cofinite sets with $0 \notin \rho$. Suppose there is an $A \subseteq \mathbb{A}$ that is closed under the inverse with respect to σ, ρ such that there is an A-manager.*

Then, (σ,ρ)-DomSet$^{\text{Rel}}$ cannot be solved in time $(|A| - \varepsilon)^{k+O(1)} \cdot n^{O(1)}$, even if we are given a path decomposition of width k, and all relations have arity at most $O(1)$, unless SETH fails.

The proof of this lemma is given in Section 9.

The third and last step for the lower bound shows how to remove these relations. This is made formal with the following lemma.

Lemma 6.6 (Removing Relations in the Decision Version). *Let σ, ρ denote finite, non-empty sets with $0 \notin \rho$. If all relations have arity at most $O(1)$, then (σ,ρ)-DomSet$^{\text{Rel}}$ $\leq_{\mathsf{tw}}$ (σ,ρ)-DomSet.*

In Section 10.1, we show how to remove relations for the decision version. The lemma follows almost immediately from Lemma 10.7.

Now we have all the results we need to show the lower bound for the decision version.

Proof of Theorem 1.5. By Definition 1.2, we have to consider three different cases. We give the proof for the first case in full detail. The remaining cases are analogous. Observe that all managers from Lemma 6.3 satisfy the constraints in Lemma 6.5.

- (σ, ρ) is not m-structured for any $\mathrm{m} \geq 2$.

  Let G be an instance of (σ, ρ)-DomSet$^{\text{Rel}}$ where the arity of the relations is $O(1)$. By the treewidth-preserving reduction from Lemma 6.6, this instance G can be transformed into an instance H of (σ, ρ)-DomSet such that $\mathsf{tw}(H) \leq \mathsf{tw}(G) + O(1)$. Toward a contradiction, assume that there is a faster algorithm for (σ, ρ)-DomSet with $c_{\sigma,\rho} = s_{\text{top}} + r_{\text{top}} + 2$. Then, use this faster algorithm to solve H and afterward recover the solution for G in time

  $$(s_{\text{top}} + r_{\text{top}} + 2 - \varepsilon)^{\mathsf{tw}(G)+O(1)} \cdot |G|^{O(1)}.$$

  By Lemma 6.3 Case 4, we know that there is an A-manager with $|A| = s_{\text{top}} + r_{\text{top}} + 2$ which then contradicts SETH by our intermediate lower bound from Lemma 6.5.

- (σ, ρ) is 2-structured, but not m-structured for any $\mathrm{m} \geq 3$, and $s_{\text{top}} = r_{\text{top}}$ is even.

  In this case, we use the same arguments as before with the difference being that for the algorithm we have $c_{\sigma,\rho} = \max(s_{\text{top}}, r_{\text{top}}) + 2$. As a consequence, we use the A-manager from Lemma 6.3 Case 2 with $|A| = (s_{\text{top}} + r_{\text{top}})/2 + 2 = \max(s_{\text{top}}, r_{\text{top}}) + 2$.

  Observe that we can use this encoder as ρ being 2-structured implies that ρ is finite but $\rho \neq \{0\}$, by assumption. Hence, we directly get $r_{\text{top}} \geq 1$.

- (σ, ρ) is m-structured for some $\mathrm{m} \geq 3$, or 2-structured with $s_{\text{top}} \neq r_{\text{top}}$, or 2-structured with $s_{\text{top}} = r_{\text{top}}$ being odd.

  Once more we use the same approach as for the first case but now we have an algorithm with $c_{\sigma,\rho} = \max(s_{\text{top}}, r_{\text{top}}) + 1$.

  If $r_{\text{top}} \geq s_{\text{top}}$, then we can use the A-manager from Lemma 6.3 Case 1 with $|A| = r_{\text{top}} + 1$. If $s_{\text{top}} \geq r_{\text{top}}$, then we want to use the A-manager from Lemma 6.3 Case 3 with $|A| = s_{\text{top}} + 1$. For this case to be applicable we need $r_{\text{top}} \geq 1$. Whenever $r_{\text{top}} = 0$, we either have $\rho = \{0\}$ or $\rho = \mathbb{Z}_{\geq 0}$. The first case is not possible by assumption. The second case implies that ρ is 1-structured but not m-structured for any $\mathrm{m} \geq 2$. This immediately contradicts our assumption. □

We note that it is indeed a reasonable assumption that $0 \notin \rho$. Otherwise the empty set is a trivial solution and the problem becomes trivial.

Observation 6.7. *For all $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$, (σ, ρ)-DomSet can be solved in constant time if $0 \in \rho$.*

6.2 Counting Problem

We have mentioned earlier that the lower bounds for the counting version follows the same ideas as the bounds for the decision version. By the definition of the managers, their constructions transfer directly to the counting version. Moreover, the lower bound for the intermediate problem is based on the result for the decision version. As we allow cofinite sets for the counting version, further investigation is needed to obtain the following result where we again use the managers in a black-box style.

Lemma 6.8 (Lower Bound for (σ,ρ)-#DomSet$^{\text{Rel}}$). *Let $\sigma,\rho \subseteq \mathbb{Z}_{\geq 0}$ be two fixed, non-empty and finite or cofinite sets. Suppose there is an $A \subseteq \mathbb{A}$ that is closed under the inverse with respect to σ,ρ such that there is an A-manager.*

Then, (σ,ρ)-#DomSet$^{\text{Rel}}$ cannot be solved in time $(|A|-\varepsilon)^{k+O(1)} \cdot n^{O(1)}$, even if we are given a path decomposition of width k, and all relations have arity at most $O(1)$, unless #SETH fails.

We show in Section 9.2 how to obtain this bound by proving the lemma.

While the procedure for the decision and counting versions was so far very similar, this does not hold for the last step where we remove the relations. Although the result is still comparable, the following reduction is much more involved than the result for the decision version.

Lemma 6.9 (Removing Relations in the Counting Version). *Let σ,ρ denote finite or cofinite, non-empty sets such that neither $\rho = \{0\}$ nor σ is cofinite with $\rho = \mathbb{Z}_{\geq 0}$.*

If all relations have arity at most $O(1)$, then (σ,ρ)-#DomSet$^{\text{Rel}}$ $\leq_{\mathsf{tw}}$ (σ,ρ)-#DomSet.

Section 10.2 gives the formal proof for this lemma.

Now we have all the results we need to prove the lower bound for the counting version.

Proof of Theorem 1.4. The proof follows in the same way as the proof of Theorem 1.5, with the only difference being that we now use the intermediate lower bound for (σ,ρ)-#DomSet$^{\text{Rel}}$ from Lemma 6.8 and the reduction from Lemma 6.9 to remove the relations. □

Similarly as for the decision version, we cannot make the assumptions about σ and ρ weaker (see Fact 3.3) except for the open case when σ is cofinite with $\rho = \mathbb{Z}_{\geq 0}$.

7 Constructing Providers

In this section, we provide the basis for the construction of the managers that we introduced earlier. Their formal construction is then given in Section 8.

In Section 7.1, we first build providers which have either σ-states or ρ-states. We use them later to construct the managers from Cases 1 and 3 in Lemma 6.3, that is, A-managers where we have that either $A \subseteq \mathbb{R}$ or $A \subseteq \mathbb{S}$.

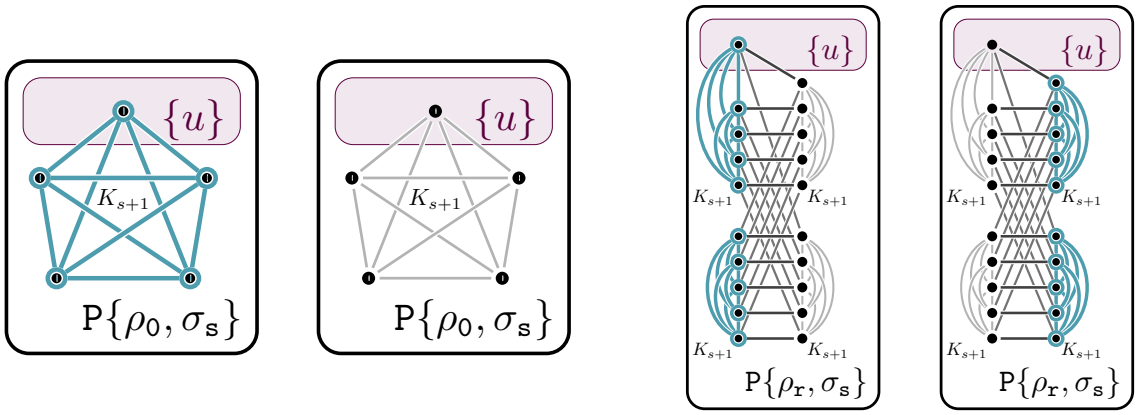
In Section 7.2, we then construct providers which contain σ-states together with ρ-states. They then allow us to construct A-managers with A containing states from $\mathbb{S}$ and simultaneously from $\mathbb{R}$. These managers correspond to Cases 2 and 4 in Lemma 6.3.

The following basic provider is a useful building block for many of our constructions.

Lemma 7.1. *For any $r \in \rho$ and $s \in \sigma$, there is a $\{\rho_r,\sigma_s\}$-provider $(\mathtt{P}\{\rho_{\mathtt{r}},\sigma_{\mathtt{s}}\},u)$.*

Proof. We define a graph $G \coloneqq \mathtt{P}\{\rho_{\mathtt{r}},\sigma_{\mathtt{s}}\}$ with a single portal u. We consider two cases depending on whether $r = 0$ or not.

Case 1: $r = 0$. We choose G to be a clique on $s+1$ vertices, and we declare any of its vertices to be u. Observe that both selecting all vertices and selecting no vertices constitute valid partial solutions, and hence, the strings $\rho_0 = \rho_r$ and σ_s are compatible with $(G,\{u\})$. Hence, $(G,\{u\})$ is a $\{\rho_r,\sigma_s\}$-provider. Consult Figure 7.1a for a visualization of the construction and the aforementioned (partial) solutions.



(a) Valid partial solutions for the provider for Case 1, when $r = 0$.

(b) Valid partial solutions for the provider for Case 2, when $r \neq 0$ ($r = 2$ in the depicted example).

Figure 7.1: The gadget constructions from Lemma 7.1.

Case 2: $r \neq 0$. We construct the graph G as follows. For each $i \in [1..r]$, we add two $(s+1)$-cliques $K^{(i)}_{s+1}$ (with the vertices $v_{i,1}, \ldots, v_{i,s+1}$) and $\overline{K}^{(i)}_{s+1}$ (with the vertices $\overline{v}_{i,1}, \ldots, \overline{v}_{i,s+1}$). Next, for each $j \in [1..s+1]$, we connect all vertices $v_{\star,j}$ and $\overline{v}_{\star,j}$ into a complete bipartite graph B_j (where $V(B_j) = (\{v_{\star,j}\} \cup \{\overline{v}_{\star,j}\})$. Finally, we choose u to be any vertex in $K^{(1)}_{s+1}$. Formally, we set

$$\begin{aligned} V(G) &:= \{v_{i,j}, \overline{v}_{i,j} \mid i \in [1..r], j \in [1..s+1]\}, \\ E(G) &:= \{v_{i,j}v_{i,j'}, \overline{v}_{i,j}\overline{v}_{i,j'} \mid i \in [1..r], j \neq j' \in [1..s+1]\} \\ &\quad \cup \{v_{i,j}\overline{v}_{i',j} \mid i, i' \in [1..r], j \in [1..s+1]\}, \quad \text{and} \\ u &:= v_{1,1}. \end{aligned}$$

Observe that $(G, \{u\})$ has the following partial solutions $\{\overline{v}_{i,j} \mid i \in [1..r], j \in [1..s+1]\}$ (the cliques $\overline{K}^{(i)}_{s+1}$ for $i \in [1..r]$) and $\{v_{i,j} \mid i \in [1..r], j \in [1..s+1]\}$ (the cliques $K^{(i)}_{s+1}$ for $i \in [1..r]$, where $K^{(1)}_{s+1}$ contains u). Consult Figure 7.1b for a visualization of the construction and the aforementioned (partial) solutions. Hence, $(G, \{u\})$ is a $\{\rho_r, \sigma_s\}$-provider, completing the proof. □

7.1 Providers Having Either σ-States or ρ-States

In this section, we establish L-providers for select languages L that are either completely contained in $\mathbb{R}$ or completely contained in $\mathbb{S}$. These gadgets exist without special requirements on ρ and σ. We start with a provider for all possible elements from $\mathbb{R}$.

Lemma 7.2. *For non-empty sets ρ and σ, there is a $\{\rho_0, \rho_1, \ldots, \rho_{r_{\text{top}}}\}$-provider $(\mathtt{P}\mathbb{R}, u)$.*

Proof. Fix $r \in \rho$ and $s \in \sigma$. We define a graph $G := \mathtt{P}\mathbb{R}$ with a single portal u. To that end, we take r_{top} independent copies $P^{(i)} := ((\mathtt{P}\{\rho_r, \sigma_s\})^{(i)}, u^{(i)})$ of the $\{\rho_r, \sigma_s\}$-provider from Lemma 7.1 and connect each vertex $u^{(i)}$ to the vertex u. Consult Figure 7.2 for a visualization.

By Lemma 7.1, each provider $P^{(i)}$ has at least a partial solution that selects $u^{(i)}$ and a partial solution that does not select $u^{(i)}$. Hence, for each $j \in [0..r_{\text{top}}]$, the constructed graph G has at least one partial solution that selects exactly j neighbors of u; this completes the proof. □

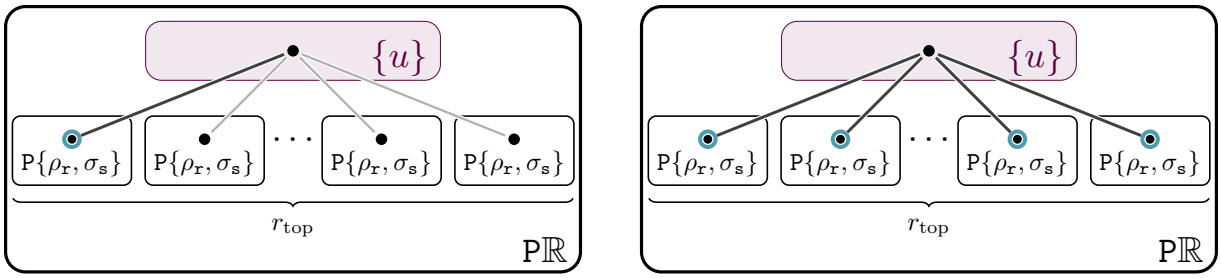


Figure 7.2: The gadget constructions from Lemma 7.2 with exemplary partial solutions that correspond to ρ_1 and $\rho_{r_{\text{top}}}$. For providers used in the construction, we depict only their portal vertices.

For our next construction, we need regular (bipartite) graphs, which are fortunately easy to construct.

Lemma 7.3. *For any nonnegative integers $d \leq n$, the bipartite graph $M_n^{(d)}$,*

$$V(M_n^{(d)}) \coloneqq \{v_0, \dots, v_{n-1}\} \cup \{w_0, \dots, w_{n-1}\},$$
$$E(M_n^{(d)}) \coloneqq \{\{v_i, w_j\} \mid ((i-j) \bmod n) \in [0\,..\,d)\}$$

is d-regular and contains an edge $\{v_i, w_i\}$ for each $i \in [0\,..\,n)$.

Proof. Observe that for each $b \in [0\,..\,d)$ and each $i \in [0\,..\,n)$, there is exactly one solution each to the equations $i - x \equiv_n b$ and $x - i \equiv_n b$; the claim follows. □

Further, recall from Section 3 that, for a string x and a character $a \in \mathbb{A}$, we write $\#_a(x)$ to denote the number of occurrences of a in x.

Lemma 7.4. *For an $r \geq 1$, consider the language*

$$L_r \coloneqq \{x \in \{\sigma_0, \sigma_1\}^{4r} \mid \#_{\sigma_1}(x) \in \{0, 2r\}\}.$$

For any set ρ that contains r and any set σ that contains an $s \geq r$, there is an L-provider $(\mathtt{P}L_r, U)$. Moreover, the closed neighborhoods of portals in U are pairwise disjoint.

Proof. Let $U = \{u_1, \dots, u_{4r}\}$ denote the set of portal vertices. We define an L-provider $(\mathtt{P}L_r, U)$ as follows. For every $A \subseteq [1\,..\,4r]$ such that $|A| = 2r$, $G \coloneqq \mathtt{P}L_r$ has the following vertices and edges in addition to its portals. It has $(2s+2)r$ vertices which are partitioned into blocks $V_A^1, \dots, V_A^{2s+2}$, where each block V_A^q contains r vertices $v_{A,1}^q, \dots, v_{A,r}^q$. Moreover, G has $2sr + 2(r-1)$ vertices which are partitioned into blocks $W_A^1, \dots, W_A^{2s+2}$, where the first two blocks W_A^q, $q \in \{1, 2\}$, contain $r-1$ vertices $w_{A,1}^q, \dots, w_{A,r-1}^q$, and all other blocks W_A^q contain r vertices $w_{A,1}^q, \dots, w_{A,r}^q$. We define $V_A \coloneqq \bigcup_{q \in [1\,..\,2s+2]} V_A^q$ and $W_A \coloneqq \bigcup_{q \in [1\,..\,2s+2]} W_A^q$. Suppose $A = \{k_1, \dots, k_{2r}\}$. Then, G has edges to form the following:

- A complete bipartite graph between V_A^q and W_A^q for all $q \in [1\,..\,2s+2]$.
- A graph on V_A such that every $v \in V_A^1 \cup V_A^2$ has degree $s-1$, and every $v \in V_A \setminus (V_A^1 \cup V_A^2)$ has degree s. This is possible by Lemma 7.3. Indeed, $|V_A|$ is even and $|V_A|/2 \geq s$. So, we can build an s-regular graph on V_A. Additionally, by numbering vertices appropriately, there is a matching between V_A^1 and V_A^2, and we may simply omit the corresponding edges to decrease the degree of every vertex from $V_A^1 \cup V_A^2$ by one to achieve the desired result.

- An s-regular graph on W_A which is possible by Lemma 7.3 since $|W_A|$ is even and $|W_A|/2 \geq s$.
- $u_{k_p} v^1_{A,p}$ for all $p \in [\![1 \,..\, r]\!]$.
- $u_{k_p} v^2_{A,p-r}$ for all $p \in [\![r+1 \,..\, 2r]\!]$.

This completes the description of the graph G. From the last two items, it follows that no two portals share an edge and that their neighborhoods are disjoint.

Let us start with some basic observations. Let $A \subseteq [\![1 \,..\, 4r]\!]$ such that $|A| = 2r$. Then, $G[W_A]$ is s-regular. We have $|N_G(v) \cap V_A| = s - 1$ for all $v \in V^1_A \cup V^2_A$, and $|N_G(v) \cap V_A| = s$ for all $v \in V_A \setminus V^1_A \cup V^2_A$. Also, $|N_G(v) \cap W_A| = r - 1$ for all $v \in V^1_A \cup V^2_A$, and $|N_G(v) \cap W_A| = r$ for all $v \in V_A \setminus V^1_A \cup V^2_A$. Moreover, $|N_G(w) \cap V_A| = r$ for all $w \in W_A$. Finally, $|N_G(v) \cap U| = 1$ for every $v \in V^1_A \cup V^2_A$.

Now, let $x \in \{\sigma_0, \sigma_1\}^{4r}$ such that $\#_{\sigma_1}(x) \in \{0, 2r\}$. Let $x = \sigma_{i_1}\sigma_{i_2}\ldots\sigma_{i_{4r}}$. We need to show that $x \in L(G, U)$. Let $A_x \coloneqq \{p \in [\![1 \,..\, 4r]\!] \mid i_p = 1\}$.

First, suppose that $A_x = \varnothing$. Then, we define a solution set

$$S_x \coloneqq U \cup \bigcup_{A \subseteq [\![1 \,..\, 4r]\!] \colon |A| = 2r} W_A.$$

Using the basic observations outlined above, it is easy to verify that this solution set indeed witnesses that $x \in L(G, U)$.

Otherwise, $|A_x| = 2r$. Now, we define a solution set

$$S_x \coloneqq U \cup V_{A_x} \cup \bigcup_{A \subseteq [\![1 \,..\, 4r]\!] \colon |A| = 2r, A \neq A_x} W_A.$$

Again, using the basic observations outlined above, it is easy to verify that this solution set indeed witnesses that $x \in L(G, U)$. □

7.2 Providers Having σ-States Together with ρ-States

Conversely to the previous section, we establish L-providers in this section, where L contains states from both $\mathbb{R}$ and $\mathbb{S}$. These providers are relevant if (σ, ρ) is m-structured for $\mathrm{m} \leq 2$, but not m-structured for any $\mathrm{m} \geq 3$.

We start by obtaining two auxiliary providers that are used in later constructions.

Lemma 7.5. *Let $r \in \rho$, $s \in \sigma$ with $r, s \geq 1$. There is an L-provider where*

$$L \coloneqq \{(\sigma_{s-1}, \sigma_{s-1}, \rho_{r-1}), (\rho_r, \rho_r, \rho_r), (\rho_{r-1}, \rho_r, \sigma_s)\}.$$

Proof. We define a graph G with three portals as follows. The vertices of G are partitioned into three sets $X = \{x^j_i \mid i \in [\![1 \,..\, r]\!], j \in [\![1 \,..\, s+1]\!]\}$, $Y = \{y^j_i \mid i \in [\![1 \,..\, r]\!], j \in [\![1 \,..\, s+1]\!]\}$, and $Z = \{z^j_i \mid i \in [\![1 \,..\, r]\!], j \in [\![1 \,..\, s+1]\!]\}$. Intuitively, each of these sets consists of r cliques of size $s+1$ each. Then, between each pair of sets from X, Y, Z, the vertices with the same index $j \in [\![1 \,..\, s+1]\!]$ form a complete bipartite graph (with r vertices in each part).

Formally, for each $i \in [\![1 \,..\, r]\!]$, the vertices $\{x^j_i \mid j \in [\![1 \,..\, s+1]\!]\}$ ($\{y^j_i \mid j \in [\![1 \,..\, s+1]\!]\}$ and $\{z^j_i \mid j \in [\![1 \,..\, s+1]\!]\}$, respectively) form a clique on $s+1$ vertices. There are no other edges within the set X (Y and Z, respectively). For each $j \in [\![1 \,..\, s+1]\!]$, the vertices $\{x^j_i \mid i \in [\![1 \,..\, r]\!]\}$ and $\{y^j_i \mid i \in [\![1 \,..\, r]\!]\}$ form the two parts of a complete bipartite graph (with r vertices in each part). Similarly, for each $j \in [\![1 \,..\, s+1]\!]$, there is a complete bipartite graph between X and Z, as well as

between Y and Z. Finally, two edges are omitted from the previous construction: there is a missing edge between x_1^1 and x_1^2, and there is a missing edge between x_1^1 and z_1^1. We use x_1^1, x_1^2, z_1^1 as the three portal vertices of G (in that order).

It remains to show that $(G, \{x_1^1, x_1^2, z_1^1\})$ is an L-provider. We give partial solutions corresponding to the three elements of L.

$(\sigma_{s-1}, \sigma_{s-1}, \rho_{r-1})$ Select the vertices from X. Thus, x_1^1 and x_1^2 are selected, but z_1^1 is not. To satisfy the σ-constraints, each vertex in X, with the exception of x_1^1 and x_1^2, obtains s selected neighbors from (their respective clique in) X. Because of the missing edge, both x_1^1 and x_1^2 only have $s-1$ selected neighbors each. To satisfy the ρ-constraints, each vertex from Y and Z, with the exception of z_1^1, obtains r selected neighbors from X. Because of the missing edge to x_1^1, z_1^1 only has $r-1$ selected neighbors.

(ρ_r, ρ_r, ρ_r) Select the vertices from Y. As before, we can check that this selection witnesses (ρ_r, ρ_r, ρ_r).

$(\rho_{r-1}, \rho_r, \sigma_s)$ Select the vertices from Z. As before, we can check that this selection witnesses $(\rho_{r-1}, \rho_r, \sigma_s)$. □

In the statement and proof of Lemma 7.6 we use, for $a \in \mathbb{A}$, the expression (a^δ) to denote the vector in $\mathbb{A}^\delta$ with $(a^\delta)[i] = a$ for all $i \in [1..\delta]$.

Lemma 7.6. *Let $r \in \rho$, $s, s' \in \sigma$ with $r, s \geq 1$ and $s' > s_{\min}$, where $s_{\min} := \min(\sigma)$. Let $k \in \mathbb{Z}_{>0}$ and $\delta := k(s' - s_{\min})$. Then, there is a $\{(\rho_r^\delta), (\rho_{r-1}^\delta), (\sigma_s^\delta)\}$-provider (with δ portals).*

Proof. We define a graph G with $\delta = k(s' - s_{\min})$ portals as follows.

- There are three disjoint sets of vertices $A = \{a_i \mid i \in [1..\delta]\}$, $B = \{b_i \mid i \in [1..\delta]\}$, and $C = \{c_i \mid i \in [1..\delta]\}$, where the vertices of C form the δ portals of G.
- For each $i \in [1..\delta]$, (a_i, b_i, c_i) are (in this order) the portals of an attached $\{(\sigma_{s-1}, \sigma_{s-1}, \rho_{r-1}), (\rho_r, \rho_r, \rho_r), (\rho_{r-1}, \rho_r, \sigma_s)\}$-provider J_i, which exists by Lemma 7.5 and the fact that $r, s \geq 1$.
- There is a set D_A of size k, where each vertex of D_A is the portal of an attached $\{\rho_r, \sigma_{s_{\min}}\}$-provider, which exists by Lemma 7.1. We introduce edges between A and D_A in a way that each vertex in A has precisely 1 neighbor in D_A, and each vertex of D_A has precisely $s' - s_{\min}$ neighbors in A. This is possible as $\delta = k \cdot (s' - s_{\min})$.
- Similarly, there is a set D_B of size k, where each vertex of D_B is the portal of an attached $\{\rho_r, \sigma_{s_{\min}}\}$-provider, each vertex in B has precisely 1 neighbor in D_B, and each vertex of D_B has precisely $s' - s_{\min}$ neighbors in B.

We show that (G, C) is a $\{(\rho_r^\delta), (\rho_{r-1}^\delta), (\sigma_s^\delta)\}$-provider. We give solutions corresponding to the three states.

(ρ_r^δ) None of the vertices in A, B, C, D_A, D_B are selected. For each $i \in [1..\delta]$, select vertices in J_i according to the solution that witnesses the state (ρ_r, ρ_r, ρ_r). In the $\{\rho_r, \sigma_{s_{\min}}\}$-providers attached to D_A and D_B, select vertices according to the ρ_r-state.

For each i, the vertices a_i and b_i, as well as the portal c_i, are unselected and each obtain r selected neighbors from J_i. Each vertex in D_A and D_B is also unselected and obtains r selected neighbors from the attached $\{\rho_r, \sigma_{s_{\min}}\}$-provider.

$(\boldsymbol{\rho^{\delta}_{r-1}})$ Select the vertices $A \cup B \cup D_A \cup D_B$. For each $i \in [1..\delta]$, select vertices in J_i according to the solution that witnesses the state $(\sigma_{s-1}, \sigma_{s-1}, \rho_{r-1})$. In the $\{\rho_r, \sigma_{s_{\min}}\}$-providers attached to D_A and D_B, select vertices according to the $\sigma_{s_{\min}}$-state.

For each i, the portal c_i is unselected and obtains $r-1$ selected neighbors from J_i. The vertices a_i and b_i are selected and each obtain $s-1$ selected neighbors from J_i, and a_i also obtains 1 selected neighbor from D_A, and b_i obtains 1 selected neighbor from D_B. Thus, both obtain a total of $s \in \sigma$ selected neighbors. The vertices in D_A are selected and each obtain $s' - s_{\min}$ selected neighbors from A, each of them also obtains $s_{\min}$ selected neighbors from the attached $\{\rho_r, \sigma_{s_{\min}}\}$-provider for a total of $s' \in \sigma$. Similarly, the vertices in D_B are selected and obtain $s' \in \sigma$ selected neighbors.

$(\boldsymbol{\sigma^{\delta}_{s}})$ Select the vertices $C \cup D_A$. For each $i \in [1..\delta]$, select vertices in J_i according to the solution that witnesses the state $(\rho_{r-1}, \rho_r, \sigma_s)$. In the $\{\rho_r, \sigma_{s_{\min}}\}$-providers attached to D_A, select vertices according to the $\sigma_{s_{\min}}$-state. In the $\{\rho_r, \sigma_{s_{\min}}\}$-providers attached to D_B, select vertices according to the ρ_r-state.

For each i, the portal c_i is selected and obtains s selected neighbors from J_i. The vertex a_i is unselected, obtains $r-1$ selected neighbors from J_i, and obtains 1 selected neighbor from D_A. The vertex b_i is unselected and obtains r selected neighbors from J_i. The vertices in D_A are selected and each obtain $s_{\min}$ selected neighbors from the attached $\{\rho_r, \sigma_{s_{\min}}\}$-provider. The vertices in D_B are unselected and each obtain r selected neighbors from the attached $\{\rho_r, \sigma_{s_{\min}}\}$-provider. □

Before we continue with the next gadget, we first prove some technical lemma which ensures the existence of bipartite graphs with certain degree sequences.

Lemma 7.7. *Let $c_1, \ldots, c_\ell, d_1, \ldots, d_k \in \mathbb{Z}_{\geq 0}$ such that*

$$\sum_{i \in [\ell]} c_i = \sum_{j \in [k]} d_j.$$

Also, suppose a is another natural number such that $a \geq \max\{c_1, \ldots, c_\ell, d_1, \ldots, d_k\}$. Then, there is some $s_0 \geq 0$ such that, for every $s \geq s_0$, there is a bipartite graph $G = (V, W, E)$ such that

1. *$V = \{v_1, \ldots, v_\ell, x_1, \ldots, x_s\}$,*
2. *$\deg_G(v_i) = c_i$ for all $i \in [\ell]$, and $\deg_G(x_i) = a$ for all $i \in [s]$,*
3. *$W = \{w_1, \ldots, w_k, y_1, \ldots, y_s\}$, and*
4. *$\deg_G(w_i) = d_i$ for all $i \in [k]$, and $\deg_G(y_i) = a$ for all $i \in [s]$.*

Proof. Pick $s_0 \coloneqq a$. We give an iterative construction for G. Initially, $E(G) \coloneqq \varnothing$. We define $c(v_i) \coloneqq c_i$ for all $i \in [\ell]$, and $c(x_j) \coloneqq a$ for $j \in [s]$. Similarly, define $c(w_i) \coloneqq d_i$ for all $i \in [k]$, and $c(y_j) \coloneqq a$ for $j \in [s]$. Throughout the construction, $c(u)$ denotes the number of neighbors still to be added to $u \in V \cup W$. We shall maintain the property that

(i) $c(u) \geq 0$ for all $u \in V \cup W$,

(ii) $\sum_{v \in V} c(v) = \sum_{w \in W} c(w)$, and

(iii) $|\{v \in V \mid c(v) \in \{c_{\max}, c_{\max} - 1\}\}| \geq a$, where $c_{\max} \coloneqq \max_{v \in V} c(v)$.

Observe that this property is initially satisfied.

If $c(w) = 0$ for all $w \in W$, then $c(v) = 0$ for all $v \in V$ by Conditions (i) and (ii), and the construction is complete. So, fix some $w \in W$ such that $c(w) > 0$. Let $d \coloneqq c(w)$. Let $v'_1, \dots, v'_{\ell+s}$ be a list of all the vertices from V ordered according to the current capacities, i.e., $c(v'_i) \geq c(v'_{i+1})$ for all $i \in [\ell + s - 1]$.

First, we claim that $c(v'_i) \geq 1$ for all $i \in [d]$. If $c_{\max} \geq 2$, then this follows directly from Condition (iii) because $d \leq a$. Otherwise, $c_{\max} = 1$. But, then $|\{v \in V \mid c(v) = c_{\max}\}| \geq d$ by Condition (ii). We add an edge between w and v'_i for all $i \in [d]$. Also, we update $c(w) \coloneqq 0$ and decrease $c(v'_i)$ by one for all $i \in [d]$. Clearly, Conditions (i) and (ii) are still satisfied. For Condition (iii), it can be observed that every $v \in V$ such that $c(v) \in \{c_{\max}, c_{\max} - 1\}$ before the update still satisfies this Condition after the update.

Repeating this process until all capacities are equal to zero, we obtain the desired graph G. □

The following lemma crucially exploits that (σ, ρ) is 1-structured or 2-structured. While in the former case we can obtain a provider with only one portal, this is not possible for the latter case. On an intuitive level, the property of being 2-structured raises parity issues which does not allow us to have two independent copies of the provider for the 1-structured case. Hence, we design a provider where the state of one portal depends on the state of the other portal, and in fact, both states must be equal for this provider.

Lemma 7.8. *Let $r_{\text{top}} \geq 1$. Suppose there is a maximum value m such that (σ, ρ) is m-structured.*

1. *If $\mathrm{m} = 1$, then there is a $\{\rho_0, \rho_1, \sigma_0\}$-provider.*
2. *If $\mathrm{m} = 2$, then there is a $\{\rho_0, \rho_2, \sigma_0\}$-provider. There is also a $\{(\rho_0, \rho_0), (\rho_1, \rho_1), (\sigma_0, \sigma_0)\}$-provider.*

Proof. We first claim that there are finite subsets $\sigma' \subseteq \sigma$ and $\rho' \subseteq \rho$ such that there exists a maximum value m' such that (σ', ρ') is m'-structured, and moreover $\mathrm{m}' = \mathrm{m}$. Indeed, if σ is finite, we simply set $\sigma' \coloneqq \sigma$. Otherwise, σ is cofinite which means there is some $s \in \mathbb{Z}_{\geq 0}$ such that $s, s+1 \in \sigma$. In particular, it follows that $\mathrm{m} = 1$. We set $\sigma' \coloneqq \{s, s+1\}$ which implies that σ' is 1-structured, but not m'-structured for any $\mathrm{m} \geq 2$. The set ρ' is defined analogously.

Since every L-provider with respect to (σ', ρ') is also an L-provider with respect to (σ, ρ), it suffices to prove the lemma for the pair (σ', ρ'). For ease of notation, let us suppose that $(\sigma, \rho) = (\sigma', \rho')$, i.e., in the remainder of the proof, we suppose that both σ and ρ are finite.

Let $\rho = \{r_1, \dots, r_{|\rho|}\}$ and $\sigma = \{s_1, \dots, s_{|\sigma|}\}$, where we assume that elements are ordered with respect to size. The fact that m is the maximum value such that (σ, ρ) is m-structured is equivalent to

$$\gcd(\{r - r_1 \mid r \in \rho\} \cup \{s - s_1 \mid s \in \sigma\}) = \mathrm{m}. \tag{7.1}$$

We claim that there are non-negative integers $x_i, \widetilde{x}_i$ (for $i \in [\![2 \,..\, |\rho|]\!]$) and non-negative integers $y_j, \widetilde{y}_j$ (for $j \in [\![1 \,..\, |\sigma|]\!]$) such that

$$\mathrm{m} + \sum_{i=2}^{|\rho|} (r_i - r_1)(x_i - \widetilde{x}_i) + \sum_{j=2}^{|\sigma|} (s_j - s_1)(y_j - \widetilde{y}_j) = 0 \tag{7.2}$$

We argue that such a choice is possible. Note that (7.2) corresponds to a Diophantine equation with variables $(x_i - \widetilde{x}_i)$ and $(y_j - \widetilde{y}_j)$ (for $2 \leq i \leq |\rho|$, $j \in [\![1 \,..\, |\sigma|]\!]$), which has solutions by (7.1). We can choose non-negative values for the indeterminates accordingly.

Using these integer solutions, we define a graph G with one portal as follows.

- The vertices of G are partitioned into two sets L and R together with some attached providers.
- L consists of the following.
  - The portal vertex u which is intended to have m neighbors in R.
  - For each $i \in [\,2\,..\,|\rho|\,]$, a set A_i of x_i vertices, each of which are intended to have $r_i - r_1$ neighbors in R.
  - For some yet to be determined positive integer k, a set A^* of k vertices, each of which are intended to have $r_{|\rho|} - r_1$ neighbors in R.
  - For each $j \in [\,2\,..\,|\sigma|\,]$, a set B_j of $b_j \coloneqq (s_j - s_1)y_j$ vertices, each of which is intended to have 1 neighbor in R.

  We set $A = A^* \cup \bigcup_{i \in [\,2\,..\,|\rho|\,]} A_i$ and $B = \bigcup_{i \in [\,2\,..\,|\sigma|\,]} B_i$.
- There are some providers attached to L:
  - Each vertex in A is portal to an attached $\{\rho_{r_1}, \sigma_{s_1}\}$-provider (exists by Lemma 7.1).
  - If $|\sigma| \geq 2$, then $s_{\text{top}} \geq 1$. Also, $r_{\text{top}} \geq 1$ by assumption. So, we can choose some $r \in \rho$ and $s \in \sigma$ with $r, s \geq 1$. For each $j \in [\,2\,..\,|\sigma|\,]$, the vertices of B_j are the b_j portals of an attached $\{(\rho_r^{b_j}), (\rho_{r-1}^{b_j}), (\sigma_s^{b_j})\}$-provider J_j (exists by Lemma 7.6 using the fact that $r, s \geq 1$).
- Analogously, R consists of the following.
  - For each $i \in [\,2\,..\,|\rho|\,]$, a set $\widetilde{A}_i$ of $\widetilde{x}_i$ vertices, each of which are intended to have $r_i - r_1$ neighbors in L.
  - A set $\widetilde{A}^*$ of k vertices, each of which are intended to have $r_{|\rho|} - r_1$ neighbors in L.
  - For each $j \in [\,2\,..\,|\sigma|\,]$, a set $\widetilde{B}_j$ of $\widetilde{b}_j \coloneqq (s_j - s_1)\widetilde{y}_j$ vertices, each of which is intended to have 1 neighbor in L.

  We set $\widetilde{A} = \widetilde{A}^* \cup \bigcup_{i \in [\,2\,..\,|\rho|\,]} \widetilde{A}_i$ and $\widetilde{B} = \bigcup_{i \in [\,2\,..\,|\sigma|\,]} \widetilde{B}_i$.
- There are some providers attached to R:
  - Each vertex in $\widetilde{A}$ is portal of an attached $\{\rho_{r_1}, \sigma_{s_1}\}$-provider.
  - For each $j \in [\,2\,..\,|\sigma|\,]$, the vertices in $\widetilde{B}_j$ are the $\widetilde{b}_j$ portals of an attached $\{(\rho_r^{\widetilde{b}_j}), (\rho_{r-1}^{\widetilde{b}_j}), (\sigma_s^{\widetilde{b}_j})\}$-provider $\widetilde{J}_j$. (Here we can use the same r and s as in the definition of L.)
- Using (7.2), we verify that the intended number of edges going from L to R is the same as the intended number of edges going from R to L:

$$\begin{aligned} \mathrm{m} + \sum_{i=2}^{|\rho|} (r_i - r_1)x_i + (r_{|\rho|} - r_1)k + \sum_{i=j}^{|\sigma|} (s_i - s_1)y_j \\ = \sum_{i=2}^{|\rho|} (r_i - r_1)\widetilde{x}_i + (r_{|\rho|} - r_1)k + \sum_{j=2}^{|\sigma|} (s_i - s_1)\widetilde{y}_j \end{aligned}$$

  Note that $r_{|\rho|} - r_1$ is the maximum of the intended degrees. Therefore, by Lemma 7.7 applied to $a = r_{|\rho|} - r_1$, for sufficiently large k, edges can be introduced such that the intended degrees in the bipartite graph induced by $L \cup R$ are met.

We show that $(G, \{u\})$ is a $\{\rho_0, \rho_m, \sigma_0\}$-provider. We give solutions corresponding to the three states.

$\boldsymbol{\rho_0}$ Both L and R are unselected. In each of the $\{\rho_{r_1}, \sigma_{s_1}\}$-providers attached to A and $\widetilde{A}$, select vertices according to the ρ_{r_1}-state. For each $i \in [2..s_{|\sigma|}]$, select vertices in J_j and $\widetilde{J_j}$ according to the $(\rho_r^{b_j})$- and $(\rho_r^{\widetilde{b}_j})$-state, respectively.

The portal u is unselected and has no selected neighbors, as required. Moreover, the vertices in A and $\widetilde{A}$ are unselected and obtain $r_1 \in \rho$ selected neighbors from the attached $\{\rho_{r_1}, \sigma_{s_1}\}$-providers. The vertices in B and $\widetilde{B}$ are unselected and obtain $r \in \rho$ selected neighbors from the attached J_j's and $\widetilde{J_j}$'s, respectively.

$\boldsymbol{\rho_m}$ R is selected, but L is unselected. In each of the $\{\rho_{r_1}, \sigma_{s_1}\}$-providers attached to A, select vertices according to the ρ_{r_1}-state. In each of the $\{\rho_{r_1}, \sigma_{s_1}\}$-providers attached to $\widetilde{A}$, select vertices according to the σ_{s_1}-state. For each $i \in [2..s_{|\sigma|}]$, select vertices in J_j according to the $(\rho_{r-1}^{b_j})$-state, and select vertices in $\widetilde{J_j}$ according to the $(\sigma_s^{\widetilde{b}_j})$-state.

The portal u is unselected and has m selected neighbors in R, as required. For $i \in [2..|\rho|]$, each vertex in A_i is unselected, obtains $r_1 \in \rho$ selected neighbors from the attached $\{\rho_{r_1}, \sigma_{s_1}\}$-provider, and additionally obtains $r_i - r_1$ selected neighbors in R, for a total of $r_i \in \rho$ selected neighbors. The vertices in $\widetilde{A}$ are selected and obtain $s_1 \in \sigma$ selected neighbors from the attached $\{\rho_{r_1}, \sigma_{s_1}\}$-providers. They have no further selected neighbors. For $i \in [2..|\sigma|]$, each vertex in B_i is unselected, obtains $r - 1 \in \rho$ selected neighbors from J_j, and additionally obtains 1 selected neighbor from R, for a total of $r \in \rho$. The vertices in $\widetilde{B}$ are selected and obtain $s \in \sigma$ selected neighbors from the attached $\widetilde{J_j}$'s.

$\boldsymbol{\sigma_0}$ The selection is symmetric to the previous state: L is selected but R is unselected. In each of the $\{\rho_{r_1}, \sigma_{s_1}\}$-providers attached to A, select vertices according to the σ_{s_1}-state. In each of the $\{\rho_{r_1}, \sigma_{s_1}\}$-providers attached to $\widetilde{A}$, select vertices according to the ρ_{r_1}-state. For each $i \in [2..s_{|\sigma|}]$, select vertices in J_j according to the $(\sigma_s^{b_j})$-state, and select vertices in $\widetilde{J_j}$ according to the $(\rho_{r-1}^{\widetilde{b}_j})$-state.

The portal u is selected and has no selected neighbors, as required. The analysis is symmetric to the previous case.

So, for $\mathrm{m} \in \{1, 2\}$, we have shown the existence of a $\{\rho_0, \rho_\mathrm{m}, \sigma_0\}$-provider. Note that, for $\mathrm{m} = 2$, if we replace the single portal u of degree 2 with two portals, each of degree 1, then the construction yields a $\{(\rho_0, \rho_0), (\rho_1, \rho_1), (\sigma_0, \sigma_0)\}$-provider. □

In the remaining part, we extend these providers to a setting where the selected portals can also get neighbors. That is, we also have the state σ_1 for the portal. We first handle the case when (σ, ρ) is 1-structured, and then independently the case when the sets are 2-structured.

7.2.1 The Sets are 1-Structured

We first construct an auxiliary provider which we combine afterward with the provider from Lemma 7.8.

Lemma 7.9. *Suppose there is a $\{\rho_0, \rho_1, \sigma_0\}$-provider. If $s_{\mathrm{top}} \geq 1$, then there is a $\{(\rho_0, \rho_0), (\rho_0, \sigma_0), (\sigma_0, \sigma_0), (\sigma_1, \sigma_1)\}$-provider. Moreover, the closed neighborhoods of both portals are disjoint.*

Proof. Let $(J, \{\widehat{u}\})$ be a $\{\rho_0, \rho_1, \sigma_0\}$-provider. We define a graph G with two portals u_1 and u_2 as follows. Let K be a clique with vertices $v_1, \ldots, v_{s_{\text{top}}+1}$ where we remove the edge between v_1 and v_2. (Note that K contains at least 2 vertices since $s_{\text{top}} \geq 1$.) Then, G consists of K where we add the two edges u_1v_1 and u_2v_2. Moreover, for every $i \in [1 \,..\, s_{\text{top}} + 1]$, we add r_{top} copies of J to G, where we use v_i as the portal vertex for all copies.

It remains to show that $(G, \{u_1, u_2\})$ is a $\{(\rho_0, \rho_0), (\rho_0, \sigma_0), (\sigma_0, \sigma_0), (\sigma_1, \sigma_1)\}$-provider. We give solutions corresponding to the four states.

(ρ_0, ρ_0) Do not select any vertices from K, so u_1 and u_2 have no selected neighbors. To satisfy the ρ-constraints of the vertices in K, each v_i obtains r_{top} selected neighbors from its attached copies of J (for each of the r_{top} attached $\{\rho_0, \rho_1, \sigma_0\}$-providers, we select vertices corresponding to the ρ_1-state).

(ρ_0, σ_0) Do not select any vertices from K. Since u_2 is selected, v_2 already has one neighbor. From the copies of J attached to v_2, it receives another $r_{\text{top}} - 1$ selected neighbors, for a total of r_{top} ($r_{\text{top}} - 1$ of the attached $\{\rho_0, \rho_1, \sigma_0\}$-providers are in the ρ_1-state, one is in the ρ_0-state). The remaining vertices of K get r_{top} neighbors via the attached copies of J.

(σ_0, σ_0) Do not select any vertices from K. Now both v_1 and v_2 already have one neighbor and receive another $r_{\text{top}} - 1$ selected neighbors through their attached copies of J. The remaining vertices of K again get r_{top} neighbors by the copies of J.

(σ_1, σ_1) Select all vertices from K. The copies of J do not provide further neighbors to these vertices (all of them are in the σ_0-state). By construction, $v_2, \ldots, v_{s_{\text{top}}+1}$ have s_{top} selected neighbors in K, and thus, in total. Both v_1 and v_2 have $s_{\text{top}} - 1$ selected neighbors in K (since the edge between v_1 and v_2 is missing), and they are adjacent to the selected u_1 and u_2, respectively. Hence, they also have s_{top} selected neighbors in total. □

Combining the previous lemma with Lemma 7.8, we get the following result.

Lemma 7.10. *Let $s_{\text{top}} \geq 1$. Suppose that $\mathrm{m} = 1$ is the maximum value such that (σ, ρ) is m-structured.*

If $r_{\text{top}} \geq 1$, then there is a $\{(\rho_0, \sigma_0), (\rho_1, \sigma_0), (\sigma_0, \sigma_0), (\sigma_1, \sigma_1)\}$-provider. If $r_{\text{top}} = 0$ but $\rho \neq \{0\}$, then there is a $\{(\rho_0, \sigma_0), (\sigma_0, \sigma_0), (\sigma_1, \sigma_1)\}$-provider. Moreover, the closed neighborhoods of the portals in both providers are disjoint.

Proof. Assume $r_{\text{top}} \geq 1$. Let $(G_1, \{u_1\})$ and $(G_2, \{u_2\})$ be two copies of a $\{\rho_0, \rho_1, \sigma_0\}$-provider as constructed in Lemma 7.8. Let $(G_3, \{u_1, u_2\})$ be a $\{(\rho_0, \rho_0), (\rho_0, \sigma_0), (\sigma_0, \sigma_0), (\sigma_1, \sigma_1)\}$-provider which exists by Lemma 7.9. It is easy to check that $(G_1 \cup G_2 \cup G_3, \{u_1, u_2\})$ is a $\{(\rho_0, \sigma_0), (\rho_1, \sigma_0), (\sigma_0, \sigma_0), (\sigma_1, \sigma_1)\}$-provider.

Observe that the construction from Lemma 7.9 does not use any $\{\rho_0, \rho_1, \sigma_0\}$-provider if $r_{\text{top}} = 0$. Hence, the exactly same proof works in the case when $r_{\text{top}} = 0$ and gives us a $\{(\rho_0, \rho_0), (\rho_0, \sigma_0), (\sigma_0, \sigma_0), (\sigma_1, \sigma_1)\}$-provider. □

7.2.2 The Sets are 2-Structured

For the case when (σ, ρ) are 2-structured, we again first show an intermediate provider.

Lemma 7.11. *Let $s_{\text{top}} \geq 1$. If a $\{\rho_0, \rho_2, \sigma_0\}$-provider and a $\{(\rho_0, \rho_0), (\rho_1, \rho_1), (\sigma_0, \sigma_0)\}$-provider exist, then there is a $\{\rho_0, \rho_2, \sigma_0, \sigma_2\}$-provider.*

Proof. Let $(G', \{u'\})$ be a $\{\rho_0, \rho_2, \sigma_0\}$-provider and assume we can construct a $\{\rho_0, \sigma_0, \sigma_2\}$-provider $(G, \{u\})$. By setting $u = u'$, $(G \cup G', \{u\})$ is a $\{\rho_0, \rho_2, \sigma_0, \sigma_2\}$-provider. In the remainder of this proof, we show how to obtain a $\{\rho_0, \sigma_0, \sigma_2\}$-provider $(G, \{u\})$.

Let K be a biclique with vertex bipartition $\{w_1, \dots, w_{s_{\text{top}}}\}$ and $\{w'_1, \dots, w'_{s_{\text{top}}}\}$. We remove the edge between w_1 and w'_1 (observe that $s_{\text{top}} \geq 1$, so these vertices always exist). For each $i \in [1 \,..\, s_{\text{top}}]$, let w_i and w'_i be the two portals of r_{top} attached copies of a $\{(\rho_0, \rho_0), (\rho_1, \rho_1), (\sigma_0, \sigma_0)\}$-provider. The portal u is adjacent to w_1 and w'_1. Then, G is the graph consisting of K, the $s_{\text{top}} \cdot r_{\text{top}}$ attached $\{(\rho_0, \rho_0), (\rho_1, \rho_1), (\sigma_0, \sigma_0)\}$-providers, and the portal u. We show that $(G, \{u\})$ is a $\{\rho_0, \sigma_0, \sigma_2\}$-provider by giving solutions corresponding to the three states.

$\boldsymbol{\rho_0}$ No vertex from K is selected, so u has no selected neighbor. To satisfy the ρ-constraints of the vertices in K, all vertices w_i and w'_i receive r_{top} selected neighbors from the attached $\{(\rho_0, \rho_0), (\rho_1, \rho_1), (\sigma_0, \sigma_0)\}$-providers (they are all in state (ρ_1, ρ_1)).

$\boldsymbol{\sigma_0}$ No vertex from K is selected, so u has no selected neighbor. However, now u is selected, so w_1 and w'_1 already have one selected neighbor. Therefore, w_1 and w'_1 receive only $r_{\text{top}} - 1$ selected neighbors from the attached $\{(\rho_0, \rho_0), (\rho_1, \rho_1), (\sigma_0, \sigma_0)\}$-providers ($r_{\text{top}} - 1$ of these are in state (ρ_1, ρ_1), the remaining one is in state (ρ_0, ρ_0)). For $i \geq 2$, w_i and w'_i receive r_{top} selected neighbors from the attached $\{(\rho_0, \rho_0), (\rho_1, \rho_1), (\sigma_0, \sigma_0)\}$-providers, as before.

$\boldsymbol{\sigma_2}$ All vertices from K are selected. Since K is a biclique (minus one edge), the vertices $w_2, \dots, w_{s_{\text{top}}}$ and $w'_2, \dots, w'_{s_{\text{top}}}$ are adjacent to s_{top} selected neighbors. The vertices w_1 and w'_1 have $s_{\text{top}} - 1$ selected neighbors in K and, additionally, the selected portal u as a neighbor. All of the $\{(\rho_0, \rho_0), (\rho_1, \rho_1), (\sigma_0, \sigma_0)\}$-providers are in state (σ_0, σ_0) and do not give any further selected neighbors to vertices in K. □

As a last step, we combine the previous lemma with Lemma 7.8 to obtain the final provider for the 2-structured case.

Lemma 7.12. *Let $r_{\text{top}} \geq 1$. Suppose that $\mathrm{m} = 2$ is the maximum value such that (σ, ρ) is m-structured. Suppose further that the elements of ρ and σ are even. Then, for $L = \{\rho_i \in \mathbb{R} \mid i \in 2\mathbb{Z}_{\geq 0}\} \cup \{\sigma_i \in \mathbb{S} \mid i \in 2\mathbb{Z}_{\geq 0}\}$, there is an L-provider.*

Proof. First assume that $s_{\text{top}} \geq 1$. Combining Lemmas 7.8 and 7.11, there is a $\{\rho_0, \rho_2, \sigma_0, \sigma_2\}$-provider $(G, \{u\})$. Create $t_{\text{top}}/2$ copies $(G_i, \{u_i\})_{i \in [1 \,..\, t_{\text{top}}/2]}$ of $(G, \{u\})$. We set $u_1 = \cdots = u_{t_{\text{top}}/2}$, i.e., we identify all portals with the same vertex. It is straightforward to verify that $(G_1 \cup \cdots \cup G_{t_{\text{top}}/2}, \{u_1\})$ is an L-provider.

If we have that $s_{\text{top}} = 0$, then we can use the same construction except that we use the $\{\rho_0, \rho_2, \sigma_0\}$-provider from Lemma 7.8 instead. □

8 Constructing Managers

The goal of this section is to establish four A-managers with different $A \subseteq \mathbb{A}$. Formally, we prove Lemma 6.3 which we restate here for convenience.

Lemma 6.3 (Existence of Managers)**.** *Let σ and ρ be two finite or cofinite sets of non-negative integers with $\rho \neq \{0\}$.*

1. *There is an A-manager with $|A| = r_{\text{top}} + 1$.*

2. *If (σ, ρ) is 2-structured, but not* m*-structured for any* $\mathrm{m} \geq 3$*, and s_{top} and $r_{\text{top}} \geq 1$ are even, then there is an A-manager with $|A| = (s_{\text{top}} + r_{\text{top}})/2 + 2$.*

3. *If $s_{\text{top}} \geq r_{\text{top}} \geq 1$, then there is an A-manager with $|A| = s_{\text{top}} + 1$.*

4. *If (σ, ρ) is not* m*-structured for any* $\mathrm{m} \geq 2$*, then there is an A-manager with $|A| = s_{\text{top}} + r_{\text{top}} + 2$.*

Moreover, each A-manager is such that A is closed under the inverse with respect to σ, ρ.

Before starting with the construction, we restate the definition of a manager here.

Definition 6.2 (A-manager)**.** *Given $A \subseteq \mathbb{A}$, an* A-manager *is an infinite family $((G_\ell, U_\ell))_{\ell \geq 1}$ of pairs (G_ℓ, U_ℓ) where G_ℓ is a graph with relations and $U_\ell = \{u_1, \ldots, u_\ell\} \subseteq V(G_\ell)$ is a set of ℓ distinguished simple vertices. Moreover, there is some non-negative integer b (depending only on $s_{\text{top}}, r_{\text{top}}$) such that the following holds for every $\ell \geq 1$:*

- *The vertices from $V(G_\ell) \setminus U_\ell$ can be partitioned into 2ℓ vertex-disjoint subgraphs $B_1, \ldots, B_\ell$ and $\overline{B}_1, \ldots, \overline{B}_\ell$, which we refer to as* blocks*, such that*
  - *$|B_i| \leq b$ and $|\overline{B}_i| \leq b$ for all $i \in [1..\ell]$,*
  - *$N(u_i) \subseteq B_i \cup \overline{B}_i$ for all $i \in [1..\ell]$,*
  - *there are edges only between the following pairs of blocks: B_i and B_{i+1}, $\overline{B}_i$ and $\overline{B}_{i+1}$, for each $i \in [1..\ell - 1]$, and B_ℓ and $\overline{B}_\ell$.*
- *Each $x \in A^\ell \subseteq \mathbb{A}^\ell$ is* managed *in the sense that there is a unique (σ, ρ)-set S_x of G_ℓ[8] such that for all $i \in [1..\ell]$:*
  - *If $x[i] = \sigma_s$, then $u_i \in S_x$. Moreover, u_i has exactly s neighbors in $B_i \cap S_x$, and exactly $s_{\text{top}} - s$ neighbors in $\overline{B}_i \cap S_x$.*
  - *If $x[i] = \rho_r$, then $u_i \notin S_x$. Moreover, u_i has exactly r neighbors in $B_i \cap S_x$, and exactly $r_{\text{top}} - r$ neighbors in $\overline{B}_i \cap S_x$.*

We refer to G_ℓ as the A-manager of rank ℓ.

See Figure 8.1 for an illustration of this definition. Note that this definition does not a priori rule out solutions that do *not* correspond to values $x \in A^\ell$.

The proof of Lemma 6.3 is split into four cases, each case corresponding to one manager. Lemmas 8.1, 8.2, 8.5 and 8.6 handle each one of these cases and together imply the above lemma.

For the construction of the managers, we mostly rely on the providers introduced in Section 7. In the following, we first construct the managers from the first two cases as they follow rather directly from the constructed providers. Then, we show how to obtain the remaining two managers assuming a general construction for managers based on specific providers. As a last step, we provide this general construction in Section 8.1.

Lemma 8.1. *For non-empty sets σ and ρ, there is an $\mathbb{R}$-manager.*

[8] Note that all vertices in G_ℓ already have a feasible number of neighbors. The graph does not have any potentially "unsatisfied" portals. In this sense, it is different from the graphs with portals that we use in other gadget constructions.

Proof. Let ℓ be a fixed rank of the manager. For each $i \in [1..\ell]$, G contains a distinguished vertex u_i that is the portal of two independent copies of a $\{\rho_0, \rho_1, \dots, \rho_{r_{\text{top}}}\}$-provider (which exists by Lemma 7.2). The providers neighboring u_i in the first and second copy form the sets B_i and $\overline{B}_i$, respectively. To each copy, we add a relation which just ensures that the selection of vertices for each state ρ_i is unique. This defines an $\mathbb{R}$-manager of rank ℓ since, for each $r \in [0..r_{\text{top}}]$, both ρ_r and $\rho_{r_{\text{top}}-r}$ are compatible with the $\{\rho_0, \rho_1, \dots, \rho_{r_{\text{top}}}\}$-provider. □

As a next step, we construct the first manager with σ and ρ-states.

Lemma 8.2. *Let $r_{\text{top}} \geq 1$. Suppose that $\mathrm{m} = 2$ is the maximum value such that (σ, ρ) is m-structured. Suppose further that all elements of ρ and σ are even. Then, there is an L-manager where*

$$L \coloneqq \{\rho_i \in \mathbb{R} \mid i \in 2\mathbb{Z}_{\geq 0}\} \cup \{\sigma_i \in \mathbb{S} \mid i \in 2\mathbb{Z}_{\geq 0}\}.$$

Proof. The construction is analogous to that in the proof of Lemma 8.1, but we use the L-provider from Lemma 7.12 instead of the $\{\rho_0, \rho_1, \dots, \rho_{r_{\text{top}}}\}$-provider. Note that r_{top} and s_{top} are even, and thus, $r_{\text{top}} - k$ and $s_{\text{top}} - k$ are even if k is even. □

Before providing the last two managers, we first introduce some notation.

Let x be a vector (string) of elements from $\mathbb{A}$. Recall that, for $a \in \mathbb{A}$, $\#_a(x)$ denotes the number of occurrences of a in x. Similarly, for $A \subseteq \mathbb{A}$, $\#_A(x)$ denotes the number of occurrences of elements from A in x.

Definition 8.3. *Let $d \geq 1$ and $\beta \in [0..2d]$. We define a set*

$$L_\beta^{(2d)} \coloneqq \{x \in \{\rho_0, \rho_1, \sigma_0, \sigma_1\}^{2d} \mid \#_{\sigma_1}(x) \in \{0, d\}, \#_{\mathbb{R}}(x) \leq \beta\}.$$

Using this definition, we can state the following general construction for a manager.

Lemma 8.4. *Suppose for some $d \geq 1$ and $\beta \in \{0, 1\}$ that there is an $L_\beta^{(2d)}$-provider where the closed neighborhoods of the portals are disjoint. Then, there is an $\mathbb{S}$-manager, and if $\beta = 1$, then there is also an $\mathbb{A}$-manager.*

Before proving Lemma 8.4 in Section 8.1, we show how to use it to prove Lemmas 8.5 and 8.6, that is, the remaining two managers from Lemma 6.3. For the first manager, the distinguished vertices have to always be selected.

Lemma 8.5. *If there is $c \in \rho$ with $s_{\text{top}} \geq c \geq 1$, then there is an $\mathbb{S}$-manager.*

Proof. By Lemma 7.4, there is an L-provider where

$$L \coloneqq \{x \in \{\sigma_0, \sigma_1\}^{4r} \mid \#_{\sigma_1}(x) \in \{0, 2r\}\}.$$

Note that this precisely corresponds to an $L_0^{(4r)}$-provider. Moreover, we get from the lemma that the closed neighborhoods of the portals are disjoint. Then, the claim follows by Lemma 8.4. □

The following manager concludes the series of constructions by providing an $\mathbb{A}$-manager.

Lemma 8.6. *Let $\rho \neq \{0\}$. Suppose that $\mathrm{m} = 1$ is the maximum value such that (σ, ρ) is m-structured. Then, there is an $\mathbb{A}$-manager.*

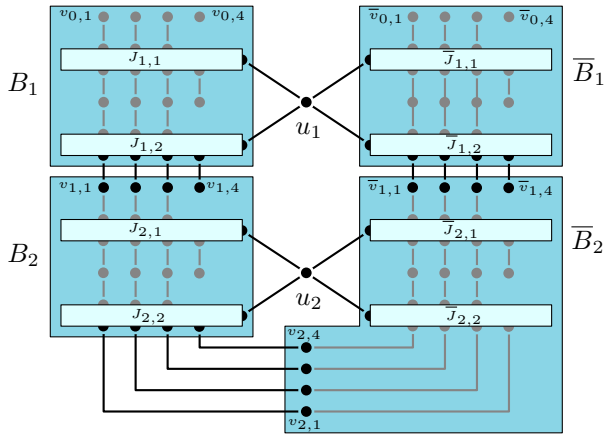


Figure 8.1: Construction of the manager from Lemma 8.4. The big, dark blue boxes represent the blocks, while the smaller, light boxes represent the L-providers. The L-providers $F^j_{i,z}, \overline{F}^j_{i,z}$ are not shown to keep the figure simple.

Proof. We first handle the case when $s_{\text{top}}, r_{\text{top}} \geq 1$. By Lemma 7.10, there is a Z-provider with $Z = \{(\rho_0, \sigma_0), (\rho_1, \sigma_0), (\sigma_0, \sigma_0), (\sigma_1, \sigma_1)\}$. Moreover, the closed neighborhoods of both portals are disjoint. From this we construct an $L_1^{(4)}$-provider as follows. We introduce four vertices $u_1, \ldots, u_4$ that serve as portals of the $L_1^{(4)}$-provider. For each pair (u_i, u_j) of distinct portals, we attach an independent copy of a Z-provider where u_i is the first portal, and u_j is the second portal vertex. It is straightforward to verify that the resulting gadget is an $L_1^{(4)}$-provider where the closed neighborhoods of the portals are disjoint. The lemma then follows from Lemma 8.4.

The next case is when, $r_{\text{top}} \geq 1$ and $s_{\text{top}} = 0$, i.e., $\sigma = \{0\}$. By Lemma 7.8, there is a $\{\rho_0, \rho_1, \sigma_0\}$-provider J. For all i, the blocks B_i and $\overline{B}_i$ consist of t_{top} copies of J each using u_i as a portal. Since $\sigma = \{0\}$, this finishes the construction of the $\mathbb{A}$-manager of rank ℓ, after we added relations to ensure that the solutions are unique, and proves its correctness.

It remains to handle the case when $r_{\text{top}} = 0$. If we have $s_{\text{top}} = 0$, then we are only interested in the states σ_0 and ρ_0. In this case the empty graph already gives an $\mathbb{A}$-manager. Thus, assume $s_{\text{top}} \geq 1$. We claim that it suffices to have a Z'-provider with $Z' = \{(\rho_0, \sigma_0), (\sigma_0, \sigma_0), (\sigma_1, \sigma_1)\}$. Then we could use the construction for $r_{\text{top}}, s_{\text{top}} \geq 1$ as we never need the combination (ρ_1, σ_0). The desired Z'-provider follows from Lemma 7.10 and finishes the proof. □

8.1 Proof of Lemma 8.4: A Blueprint for Managers

Now, we turn to the proof of Lemma 8.4. That is, given an $L_\beta^{(2d)}$-provider for some $d \geq 1$ and $\beta \in \{0, 1\}$, we construct an $\mathbb{S}$-manager, and if $\beta = 1$, then also an $\mathbb{A}$-manager.

To simplify notation let $L = L_\beta^{(2d)}$ in the following.

Construction of the Graph. Let ℓ be a fixed rank of the manager. For ease of notation, we omit ℓ from G_ℓ and U_ℓ in the following and just write G and U. We first define the graph G with its set of distinguished vertices $U = \{u_1, \ldots, u_\ell\}$. An illustration is given in Figure 8.1. Afterward, we establish that it is an $\mathbb{S}$-manager or $\mathbb{A}$-manager for $\beta = 0$ or $\beta = 1$, respectively.

Apart from the vertices in U, G also contains $\ell^* - \ell$ auxiliary vertices $u_{\ell+1}, \ldots, u_{\ell^*}$, which are used to circumvent some parity issues that would otherwise prevent us from obtaining proper solutions. In the construction of G, we treat these auxiliary vertices in the same way as the vertices

in U. However, for the encoding that we want to model (see Definition 6.2), we are only interested in the vertices $u_1, \dots, u_\ell$ and their selected neighbors. Later, we set the number ℓ^* (as a number depending on β). For each $i \in [1..\ell^*]$, there are some subgraphs X_i and $\overline{X}_i$, which we define subsequently. They provide the neighbors for u_i in G, and with some minor modifications, these subgraphs X_i and $\overline{X}_i$ form the blocks B_i and $\overline{B}_i$ of the manager G.

For all $i \in [0..\ell^*]$ and all $j \in [1..d]$, we create vertices $v_{i,j}$ and $\overline{v}_{i,j}$. For $j \in [1..d]$, we identify $v_{\ell^*,j} = \overline{v}_{\ell^*,d+1-j}$. Intuitively, the case $i = \ell^*$ is special as it is the connection between the $v_{i,j}$'s and the $\overline{v}_{i,j}$'s. For each $i \in [0..\ell^*]$, the vertices $v_{i,1}, \dots, v_{i,d}$ serve as portals of s_{top} copies of the given L-provider. Let us name these providers $F_i^1, \dots, F_i^{s_{\text{top}}}$. (We use "$F$" as in "fill" since these providers are used to fill up the count of selected neighbors of their portals as far as possible.) Note that the L-provider has $2d$ portals, and therefore, each vertex in $v_{i,1}, \dots, v_{i,d}$ serves as two portals for each of these L-providers. Identifying two portals of an L-provider is possible without forming loops or multiple edges by the assumption that the portals of the given L-provider are independent and have no common neighbors. Analogously, for each $i \in [0..\ell^*]$, the vertices $\overline{v}_{i,1}, \dots, \overline{v}_{i,d}$ serve as portals of s_{top} L-providers $\overline{F}_i^1, \dots, \overline{F}_i^{s_{\text{top}}}$.

For each $i \in [1..\ell^*]$, there is a subgraph X_i defined as follows. It contains a set of vertices $\{x_{i,z,j} \mid z \in [0..t_{\text{top}}], j \in [1..d]\}$. For $z \notin \{0, t_{\text{top}}\}$, these $x_{i,z,j}$'s are new vertices, but for all $j \in [1..d]$, we set $x_{i,0,j} = v_{i-1,j}$ and $x_{i,t_{\text{top}},j} = v_{i,j}$. Just like the $v_{i,j}$'s, for each $z \in [1..t_{\text{top}} - 1]$, the vertices $x_{i,z,1}, \dots, x_{i,z,d}$ serve as portals of s_{top} copies $F_{i,z}^1, \dots, F_{i,z}^{s_{\text{top}}}$ of the given L-provider. We set $F_{i,0}^y = F_{i-1}^y$ and $F_{i,t_{\text{top}}}^y = F_i^y$ for all $y \in [1..s_{\text{top}}]$. Furthermore, for all $z \in [1..t_{\text{top}}]$, X_i contains an L-provider $J_{i,z}$ with portals u_i, $x_{i,z-1,1}, \dots, x_{i,z-1,d-1}$, and $x_{i,z,1}, \dots, x_{i,z,d}$.

For each i, we similarly define a subgraph $\overline{X}_i$ with vertices $\{\overline{x}_{i,z,j} \mid z \in [0..t_{\text{top}}], j \in [1..d]\}$, and $\overline{x}_{i,0,j} = \overline{v}_{i-1,j}$, $\overline{x}_{i,t_{\text{top}},j} = \overline{v}_{i,j}$ (for all $j \in [1..d]$). Moreover, for all $z \in [1..t_{\text{top}}]$, $\overline{X}_i$ contains an L-provider $\overline{J}_{i,z}$ with portals u_i, $\overline{x}_{i,z-1,1}, \dots, \overline{x}_{i,z-1,d-1}$, and $\overline{x}_{i,z,1}, \dots, \overline{x}_{i,z,d}$. For all $z \in [1..t_{\text{top}} - 1]$, we create s_{top} L-providers $\overline{F}_{i,z}^1, \dots, \overline{F}_{i,z}^{s_{\text{top}}}$, and additionally set for all $y \in [1..s_{\text{top}}]$, $\overline{F}_{i,0}^y = \overline{F}_{i-1}^y$ and $\overline{F}_{i,t_{\text{top}}}^y = \overline{F}_i^y$. As a last step, we make the vertices in X_i together with u_i subject to a relation which ensures that there is exactly one solution for each state the vertex u_i can get. For this we use the solution which we construct in the following. We apply the same to the vertices of $\overline{X}_i$ together with u_i. This completes the definition of G.

Partitioning the Graph. We use G for both cases $\beta = 0$ and $\beta = 1$. We only adjust the value of ℓ^* accordingly. We have to show that there is an integer b, and a partition of G into blocks that fulfill the requirements stated in Definition 6.2.

Let δ_L denote the number of vertices of the L-provider (including the $2d$ portal vertices). We set

$$b \coloneqq (\ell^* - \ell + 1) \cdot (s_{\text{top}} + 1) \cdot (t_{\text{top}} + 1) \cdot \delta_L. \tag{8.1}$$

For all $i \in [1..\ell - 1]$, the block B_i consists of the vertices of X_i minus the ones from X_{i+1} (and minus u_i). Note that the common vertices of X_i and X_{i+1} are $v_{i,1}, \dots, v_{i,d}$ together with the L-providers $F_i^1, \dots, F_i^{s_{\text{top}}}$. Likewise, the block $\overline{B}_i$ consists of the vertices of $\overline{X}_i$ minus the vertices from $\overline{X}_{i+1}$. The final blocks B_ℓ and $\overline{B}_\ell$ hold all the remaining vertices of $V(G) \setminus U$, that is, B_ℓ consists of all X_i with $i \in [\ell..\ell^*]$ minus the vertices from $X_{\ell^*} \cap \overline{X}_{\ell^*}$. (These are $v_{\ell^*,1} = \overline{v}_{\ell^*,d}, \dots, v_{\ell^*,d} = \overline{v}_{\ell^*,1}$ together with the L-providers $F_{\ell^*}^1, \dots, F_{\ell^*}^{s_{\text{top}}}$.) Then, $\overline{B}_\ell$ consists of all $\overline{X}_i$ with $i \in [\ell..\ell^*]$.

It is straightforward to verify that these blocks form a partition of the vertices of G. Each set X_i contains $s_{\text{top}} \cdot (t_{\text{top}} + 1)$ many L-providers of the form $F_{i,z}^s$, and t_{top} many L-providers of the form $J_{i,z}$. So in total, this gives at most $(s_{\text{top}} + 1) \cdot (t_{\text{top}} + 1)$ many L-providers. Since each vertex $x_{i,z,j}$ is a portal of at least one of those L-providers, we conclude that X_i contains at most $(s_{\text{top}} + 1) \cdot (t_{\text{top}} + 1) \cdot \delta_L$ many vertices. The same bound also holds for all subgraphs $\overline{X}_i$ for all

$i \in [1..\ell^*]$. Since each block B_i or $\overline{B}_i$, $i \in [1..\ell]$, contains vertices from at most $(\ell^* - \ell + 1)$ many subgraphs X_i or $\overline{X}_i$, it follows that $|B_i| \leq b$ and $|\overline{B}_i| \leq b$ for all $i \in [1..\ell]$. Finally, by construction, it is immediately clear that $N(u_i) \subseteq B_i \cup \overline{B}_i$ for all $i \in [1..\ell]$, and there are edges only between B_i and B_{i+1}, $\overline{B}_i$ and $\overline{B}_{i+1}$, for each $i \in [1..\ell-1]$, and B_ℓ and $\overline{B}_\ell$.

Constructing a Solution. To show that the corresponding graph G together with the distinguished vertices $U = \{u_1, \ldots, u_\ell\}$ is an $\mathbb{A}$-manager (for $\beta = 1$) or an $\mathbb{S}$-manager (for $\beta = 0$) of rank ℓ, we additionally need to prove the second property from Definition 6.2.

For a given $x \in \mathbb{A}^\ell$,[9] we define an $x^* \in \mathbb{A}^{\ell^*}$ with $x^*[i] = x[i]$ for all $i \in [1..\ell]$, while the remaining entries are set depending on β (and x). We iteratively construct (partial) solutions $S_{i,z}$ for all $i \in [1..\ell^*]$ and $z \in [0..t_{\text{top}}]$ such that $S_x \coloneqq S_{\ell^*,t_{\text{top}}}$ corresponds to the final solution. Based on these $S_{i,z}$, we define two functions $B, \overline{B} : [1..\ell^*] \times [0..t_{\text{top}}] \to [0..d-1]$ such that $B(i,z) = k$ if and only if the k vertices $x_{i,z,1}, \ldots, x_{i,z,k}$ need (exactly) one more neighbor in $S_{i,z}$, respectively for $\overline{B}(i,z) = k$ and the vertices $\overline{x}_{i,z,1}, \ldots, \overline{x}_{i,z,k}$. Due to the construction of G, we set $S_{i,t_{\text{top}}} = S_{i+1,0}$, and therefore, $B(i, t_{\text{top}}) = B(i+1, 0)$ and $\overline{B}(i, t_{\text{top}}) = \overline{B}(i+1, 0)$ for all $i \in [1..\ell^* - 1]$. We later set ℓ^* and x^* such that the partial solutions maintain the following invariant for all $i \in [1..\ell^*]$:

$$B(i, t_{\text{top}}) + \overline{B}(i, t_{\text{top}}) \equiv_d s_{\text{top}} \cdot \#_{\mathbb{S}}(x^*[1..i]) \quad \text{and} \quad B(1,0) = \overline{B}(1,0) = 0 \tag{8.2}$$

$S_{1,0}$ consists of the vertices $x_{1,0,j}$ and $\overline{x}_{1,0,j}$ for all $j \in [1..d]$. Moreover, the L-providers $F^1_{1,0}, \ldots, F^{s_{\text{top}}}_{1,0}$ and $\overline{F}^1_{1,0}, \ldots, \overline{F}^{s_{\text{top}}}_{1,0}$ give each of these vertices s_{top} neighbors. We directly get $B(1,0) = \overline{B}(1,0) = 0$ which satisfies the invariant.

Now, for each i from 1 to ℓ^*, we iterate over all z from 1 to t_{top} and apply the following selection process. Unless mentioned otherwise, the portals of the L-providers get state σ_0.

- $S_{i,z}$ consists of all selected vertices from $S_{i,z-1}$. If $x^*[i] \in \mathbb{S}$, then we select vertex u_i. Moreover, we select the vertices $x_{i,z,j}$ and $\overline{x}_{i,z,j}$ for all $j \in [1..d]$. We use the L-providers $F^y_{i,z}, \overline{F}^y_{i,z}$ for all $y \in [1..s_{\text{top}} - 1]$ to add $s_{\text{top}} - 1$ neighbors to each of these vertices.
- Selection process for B_i if $x^*[i] = \sigma_s$.
  - If u_i has $< s$ neighbors in $B_i \cap S_{i,z-1}$, then we give it one more neighbor in $S_{i,z}$. The L-provider $J_{i,z}$ gives state σ_1 to u_i and each of $x_{i,z-1,1}, \ldots, x_{i,z-1,B(i,z-1)}$ and $x_{i,z,B(i,z-1)+2}, \ldots, x_{i,z,d}$. Hence, the vertices $x_{i,z,1}, \ldots, x_{i,z,B(i-1)+1}$ need one more neighbor.

    If $B(i-1,z) + 1 = d$, then we use the additional L-provider $F^{s_{\text{top}}}_{i,z}$ to give every vertex one additional neighbor. Therefore, $B(i,z) = B(i,z-1) + 1 \mod d$.
  - If u_i has s neighbors in $B_i \cap S_{i,z-1}$, then we do not give it one more neighbor in $S_{i,z}$. The L-provider $J_{i,z}$ gives state σ_0 to u_i and state σ_1 to each of $x_{i,z-1,1}, \ldots, x_{i,z-1,B(i,z-1)}$ and $x_{i,z,B(i,z-1)+1}, \ldots, x_{i,z,d}$. Hence, $B(i,z) = B(i,z-1)$.
- Selection process for $\overline{B}_i$ if $x^*[i] = \sigma_s$.
  - If u_i has $< s_{\text{top}} - s$ neighbors in $\overline{B}_i \cap S_{i,z-1}$, then we give it one more neighbor in $S_{i,z}$. The L-provider $\overline{J}_{i,z}$ gives state σ_1 to u_i and each of $\overline{x}_{i,z-1,1}, \ldots, \overline{x}_{i,z-1,\overline{B}(i,z-1)}$ and $\overline{x}_{i,z,\overline{B}(i,z-1)+2}, \ldots, \overline{x}_{i,z,d}$. Hence, the $\overline{B}(i,z-1)+1$ vertices $\overline{x}_{i,z,1}, \ldots, \overline{x}_{i,z,\overline{B}(i,z-1)+1}$ need one more neighbor.

[9] For $\beta = 0$ we restrict ourself to strings from $\mathbb{S}^\ell$.

If $\overline{B}(i-1,z)+1 = d$, then we use the additional L-provider $\overline{F}_{i,z}^{s_{\text{top}}}$ to give every vertex one additional neighbor. Therefore, $\overline{B}(i,z) = \overline{B}(i,z-1)+1 \mod d$.

- If u_i has $s_{\text{top}} - s$ neighbors in $\overline{B}_i \cap S_{i,z-1}$, then we do not give it one more neighbor in $S_{i,z}$. The L-provider $\overline{J}_{i,z}$ gives state σ_0 to u_i and state σ_1 to each of $\overline{x}_{i,z-1,1}, \dots, \overline{x}_{i,z-1,\overline{B}(i,z-1)}$ and $\overline{x}_{i,z,\overline{B}(i,z-1)+1}, \dots, \overline{x}_{i,z,d}$. Hence, $\overline{B}(i,z) = \overline{B}(i,z-1)$.

- Selection process for B_i if $x^*[\,i\,] = \rho_r$. (Note that this case does not occur when $\beta = 0$)
  - If u_i has $< r$ neighbors in $B_i \cap S_{i,z-1}$, then we give it one more neighbor in $S_{i,z}$. The L-provider $J_{i,z}$ gives state ρ_1 to u_i and state σ_1 to each of $x_{i,z-1,1}, \dots, x_{i,z-1,B(i,z-1)}$ and $x_{i,z,B(i,z-1)+1}, \dots, x_{i,z,d}$. Since the vertices $x_{i,z,1}, \dots, x_{i,z,B(i-1)}$ need one more neighbor, $B(i,z) = B(i,z-1)$.
  - If u_i has r neighbors in $B_i \cap S_{i,z-1}$, then we do not give it one more neighbor in $S_{i,z}$. The L-provider $J_{i,z}$ gives state ρ_0 to u_i and state σ_1 to each of $x_{i,z-1,1}, \dots, x_{i,z-1,B(i,z-1)}$ and $x_{i,z,B(i,z-1)+1}, \dots, x_{i,z,d}$. Hence, $B(i,z) = B(i,z-1)$.

- Selection process for $\overline{B}_i$ if $x^*[\,i\,] = \rho_r$. (Note that this case does not occur when $\beta = 0$)
  - If u_i has $< r_{\text{top}} - r$ neighbors in $\overline{B}_i \cap S_{i,z-1}$, then we give it one more neighbor in $S_{i,z}$. The L-provider $\overline{J}_{i,z}$ gives state ρ_1 to u_i and state σ_1 to each of $\overline{x}_{i,z-1,1}, \dots, \overline{x}_{i,z-1,\overline{B}(i,z-1)}$ and $\overline{x}_{i,z,\overline{B}(i,z-1)+1}, \dots, \overline{x}_{i,z,d}$. Since the vertices $\overline{x}_{i,z,1}, \dots, \overline{x}_{i,z,\overline{B}(i,z-1)}$ need one more neighbor, $\overline{B}(i,z) = \overline{B}(i,z-1)$.
  - If u_i has $r_{\text{top}} - r$ neighbors in $\overline{B}_i \cap S_{i,z-1}$, then we do not give it one more neighbor in $S_{i,z}$. The L-provider $\overline{J}_{i,z}$ gives state ρ_0 to u_i and state σ_0 to each of $\overline{x}_{i,z-1,1}, \dots, \overline{x}_{i,z-1,\overline{B}(i,z-1)}$ and $\overline{x}_{i,z,\overline{B}(i,z-1)+1}, \dots, \overline{x}_{i,z,d}$. Hence, $\overline{B}(i,z) = \overline{B}(i,z-1)$.

Correctness of the Solution. We first show that these steps preserve the invariant in Equation (8.2). The claim immediately follows if $x^*[\,i\,] = \rho_r$ because in this case $B(i,z-1) = B(i,z)$ for all $z \in [\,1\,..\,t_{\text{top}}\,]$. Likewise, we get $\overline{B}(i,z-1) = \overline{B}(i,z)$.

For the case $x^*[\,i\,] = \sigma_s$, we get that $B(i,z) \equiv_d B(i,0) + z$ for all $z \in [\,1\,..\,s\,]$, and $B(i,z) = B(i,z-1)$ for all $z \in (\,s\,..\,t_{\text{top}}\,]$. Conversely, we get $\overline{B}(i,z) \equiv_d \overline{B}(i,0) + z$ for all $z \in [\,1\,..\,s_{\text{top}} - s\,]$, and $\overline{B}(i,z) = \overline{B}(i,z-1)$ for all $z \in (\,s_{\text{top}} - s\,..\,t_{\text{top}}\,]$. This directly yields

$$\begin{aligned} B(i,t_{\text{top}}) + \overline{B}(i,t_{\text{top}}) &\equiv_d B(i,0) + \overline{B}(i,0) + s + (s_{\text{top}} - s) \\ &\equiv_d \#_{\mathbb{S}}(x^*[\,1\,..\,i-1\,]) \cdot s_{\text{top}} + s_{\text{top}} \\ &\equiv_d \#_{\mathbb{S}}(x^*[\,1\,..\,i\,]) \cdot s_{\text{top}}, \end{aligned}$$

proving that the invariant is preserved.

Setting the Parameters. Assume we can set ℓ^* and $x^*(\,\ell\,..\,\ell^*\,]$ such that the solution satisfies $B(\ell^*,t_{\text{top}}) + \overline{B}(\ell^*,t_{\text{top}}) \in \{0,d\}$. We then show that $S_{\ell^*,t_{\text{top}}}$ is indeed a valid solution. From $B(\ell^*,t_{\text{top}}) = k$, we know that the vertices $v_{\ell^*,1}, \dots, v_{\ell^*,k}$ need one more neighbor. From $\overline{B}(\ell^*,t_{\text{top}}) = k'$, we know that the vertices $\overline{v}_{\ell^*,1}, \dots, \overline{v}_{\ell^*,k'}$ need one more neighbor. Recall that $v_{\ell^*,j} = \overline{v}_{\ell^*,d+1-j}$ for all j. If $B(\ell^*,t_{\text{top}}) + \overline{B}(\ell^*,t_{\text{top}}) = d$, then $k' = d - k$ and the vertices $v_{\ell^*,1}, \dots, v_{\ell^*,k}$ get a neighbor from $\overline{J}_{\ell^*,t_{\text{top}}}$. Conversely, the vertices $\overline{v}_{\ell^*,1} = v_{\ell^*,d}, \dots, \overline{v}_{\ell^*,d-k} = v_{\ell^*,k+1}$ get a neighbor from $J_{\ell^*,t_{\text{top}}}$. Thus, none of these vertices need an additional neighbor.

If $B(\ell^*,t_{\text{top}}) + \overline{B}(\ell^*,t_{\text{top}}) = 0$, then the L-provider $F_{\ell^*}^{s_{\text{top}}} = \overline{F}_{\ell^*}^{s_{\text{top}}}$ already provided one additional neighbor for these vertices.

It remains to set ℓ^* and $x^*(\ell \,..\, \ell^*]$ depending on β such that the claim holds.

- $\beta = 1$: We set $\ell^* = \ell + d$. Let $x^*[i] = \sigma_0$ for all $i \in (\ell \,..\, \ell + \ell')]$ and $x^*[i] = \rho_0$ for all $i \in (\ell + \ell' \,..\, \ell^*]$, where $\ell' = d - (\#_{\mathbb{S}}(x) \mod d)$. From the invariant in Equation (8.2), we get:

$$\begin{aligned} B(\ell^*, t_{\text{top}}) + \overline{B}(\ell^*, t_{\text{top}}) &\equiv_d s_{\text{top}} \cdot \#_{\mathbb{S}}(x^*) \\ &\equiv_d s_{\text{top}} \cdot (\#_{\mathbb{S}}(x) + \#_{\mathbb{S}}(x^*(\ell \,..\, \ell^*])) \\ &\equiv_d s_{\text{top}} \cdot (\#_{\mathbb{S}}(x) + \ell') \\ &\equiv_d s_{\text{top}} \cdot (\#_{\mathbb{S}}(x) + d - (\#_{\mathbb{S}}(x) \mod d)) \equiv_d 0 \end{aligned}$$

- $\beta = 0$: We set $\ell^* = \ell + d - (\ell \mod d)$ and $x^*[i] = \sigma_0$ for all $i \in (\ell \,..\, \ell^*]$. By the invariant in Equation (8.2) and the fact that $x^* \in \mathbb{S}^{\ell^*}$, we get:

$$\begin{aligned} B(\ell^*, t_{\text{top}}) + \overline{B}(\ell^*, t_{\text{top}}) &\equiv_d s_{\text{top}} \cdot \#_{\mathbb{S}}(x^*) \\ &\equiv_d s_{\text{top}} \cdot \ell^* \\ &\equiv_d s_{\text{top}} \cdot (\ell + d - (\ell \mod d)) \equiv_d 0 \end{aligned}$$

We complete the proof of Lemma 8.4 by observing that $\ell^* - \ell + 1 \leq d + 1$ in both cases, which means that b (see Equation (8.1)) only depends on s_{top} and r_{top}.

9 Lower Bound for the Problem with Relations

As mentioned earlier, our lower bounds follow in three steps. In Section 7, we provided the basis for the managers we constructed in Section 8. In this section, we proceed with the second step that is an intermediate lower bound for (σ, ρ)-DomSet$^{\text{Rel}}$, which is defined as the generalization of (σ, ρ)-DomSet on a graph with relations, cf. Definitions 6.1 and 6.4. Unless mentioned otherwise, the considered vertices are simple from now on.

In Section 9.1, we prove the intermediate lower bound for the decision version, that is, we prove Lemma 6.5. We achieve this by giving a reduction from k-SAT to (σ, ρ)-DomSet$^{\text{Rel}}$ assuming a suitable A-manager is given. In Section 9.2, we reuse this construction to show the lower bound for the counting version, that is, Lemma 6.8. As the construction for the decision version is only for the case when σ and ρ are finite or simple cofinite, we provide some modifications and extensions (exploiting properties of the counting version) such that the sets might also be cofinite without being simple cofinite.

Observe that each A-manager from Lemmas 8.1, 8.2, 8.5 and 8.6, satisfies our conditions on A from the lower bounds. Hence, we can use them in the following constructions.

9.1 Decision Problem

To prove the lower bound, we reduce from k-SAT to (σ, ρ)-DomSet$^{\text{Rel}}$, and keep pathwidth, and thus, also treewidth low. For this we follow previous approaches as in [18, 36, 37].

We first introduce some types of relations that we use in the following reduction.

Definition 9.1. *Let $d \geq 1$ denote a positive integer and $\ell \geq 0$ denote a non-negative integer.*

We write $\texttt{HW}^{(d)}_{=\ell} \coloneqq \{S \subseteq [1 \,..\, d] \mid |S| = \ell\}$ for the d-ary Hamming Weight ℓ *relation, that is, the relation whose corresponding partial solutions contain exactly ℓ selected portal vertices.*

Further, we write $\texttt{EQ}^{(d)} \coloneqq \{\varnothing, [1 \,..\, d]\}$ for the d-ary Equality *relation, that is, the relation whose corresponding partial solutions either select all portal vertices or none of them.*

In Section 9.1.1, we give the construction of the (σ, ρ)-DomSet$^{\text{Rel}}$ instance. Then, in Section 9.1.2, we prove the correctness of these constructions, and finalize the proof in Section 9.1.3.

9.1.1 Construction of the Graph

Let ϕ be a given k-SAT formula in CNF with n variables $x_1 \dots, x_n$ and m clauses $C_1, \dots, C_m$. We group the variables of ϕ into $t := \lceil n/q \rceil$ groups $F_1, \dots, F_t$ of q variables each, where q is chosen later. Later, we also choose some g with $2^q \leq |A|^g$.

We now define a corresponding graph G_ϕ that serves as an instance of (σ, ρ)-DomSet$^{\text{Rel}}$. The construction is illustrated in Figure 9.1.

In the following, the indices we use are superscripts if they refer to "columns" of the construction, and they are subscripts if they refer to "rows" of the construction. Whenever we say that some set of vertices V' is subject to a relation R, we mean that there is a complex vertex v to which the relation R is assigned and the neighborhood of v is V'. In a separate step, we show how to remove these complex vertices and their relations. Here is the construction of G_ϕ:

- Introduce vertices $w^j_{i,\ell}$ for all $i \in [1..t]$, $\ell \in [1..g]$, and $j \in [0..m]$.
- Introduce vertices c^j_i for all $i \in [0..t]$ and $j \in [1..m]$.
- For all $j \in [0..m]$, we create a copy J^j of the given A-manager of rank tg for which $\{w^j_{i,\ell} \mid i \in [1..t], \ell \in [1..g]\}$ act as distinguished vertices. Let $B^j_{i,\ell}$ and $\overline{B}^j_{i,\ell}$ be the corresponding blocks.
- For each $w^j_{i,\ell}$, $N_B(w^j_{i,\ell})$ and $N_{\overline{B}}(w^j_{i,\ell})$ are the neighborhoods of $w^j_{i,\ell}$ in $B^j_{i,\ell}$ and $\overline{B}^j_{i,\ell}$, respectively.
- For all $i \in [0..t]$ and $j \in [1..m]$, there is a $\{\sigma_{s_{\min}}, \rho_{r_{\min}}\}$-provider Q^j_i (exists by Lemma 7.1) with c^j_i as portal. We add a relation to the vertices of Q^j_i and c^j_i which ensures that the provider only has two solutions, namely one corresponding to the state when c^j_i is selected, and one to the state when c^j_i is not selected.
- For all $j \in [1..m]$, c^j_0 is additionally subject to a relation $\mathtt{HW}^{(1)}_{=0}$, and c^j_t is additionally subject to $\mathtt{HW}^{(1)}_{=1}$.
- For each $i \in [1..t]$ and $j \in [1..m]$, the set of vertices

$$Z^j_i := \{c^j_{i-1}, c^j_i\} \cup \bigcup_{\ell \in [1..g]} \left(w^{j-1}_{i,\ell} \cup N_{\overline{B}}(w^{j-1}_{i,\ell}) \cup w^j_{i,\ell} \cup N_B(w^j_{i,\ell})\right)$$

  is subject to a relation R^j_i, which we define in a moment.
- Similarly, for each $i \in [1..t]$,

$$Z^0_i := \bigcup_{\ell \in [1..g]} \left(w^1_{i,\ell} \cup N_B(w^1_{i,\ell})\right) \quad \text{and} \quad Z^{m+1}_i := \bigcup_{\ell \in [1..g]} \left(w^m_{i,\ell} \cup N_{\overline{B}}(w^m_{i,\ell})\right),$$

  are subject to relations R^0_i and R^{m+1}_i, respectively.

This completes the definition of G_ϕ. In order to define the relations R_i^j, we need some more notation. For each group of variables F_i, there are at most 2^q corresponding partial assignments. We encode these assignments by states from A^g. For technical reasons, instead of using all states, we only use those states that have a certain property. For now it suffices to think of this property as having a certain weight. To this end, we first define the weight $\mathrm{wt}(x)$ of a vector $x \in \mathbb{A}^g$ using the definition of a weight vector $\vec{w}(\cdot)$ from Definition 4.2:

$$\mathrm{wt}(x) = \sum_{i=1}^{g} \vec{w}(x)[\,i\,].$$

One can also think of $\mathrm{wt}(\cdot)$ as the 1-norm of $\vec{w}(\cdot)$. Let $\max A = \max\{i \mid \sigma_i \in A \vee \rho_i \in A\}$. Observe that the vectors in A^g can have at most $g \cdot \max A + 1$ possible weights between 0 and $g \cdot \max A$. There are $|A^g|$ vectors in total.

Claim 9.2. *There is a weight w and a set $\mathfrak{A} \subseteq A^g$ such that $|\mathfrak{A}| \geq \lceil |A^g|/(g \cdot \max A + 1) \rceil$ and, for all $x \in \mathfrak{A}$, we have $\mathrm{wt}(x) = w$.*

Proof. If $g \cdot \max A + 1 \geq |A^g|$, then we only need that there is one weight which appears at least once. This is true as A is non-empty. Otherwise, there are more vectors than weights, and $g \cdot \max A + 1 \leq |A^g|$. The pigeon-hole principle provides some weight w such that at least the claimed fraction of vectors has weight w. □

Recall the definition of the inverse of a state from Definition 3.8. Using this definition, we define a new set $\widehat{\mathfrak{A}} \subseteq \mathbb{A}^g$ as the inverse of $\mathfrak{A}$ with respect to σ and ρ if the sets σ and ρ are simple cofinite, respectively. We initially set $\widehat{\mathfrak{A}} = \mathfrak{A}$.

- If σ is simple cofinite, then update $\widehat{\mathfrak{A}} \coloneqq \mathsf{inv}^\sigma(\widehat{\mathfrak{A}})$.
- If ρ is simple cofinite, then update $\widehat{\mathfrak{A}} \coloneqq \mathsf{inv}^\rho(\widehat{\mathfrak{A}})$.

Since A is closed under inversion with respect to σ, ρ, we get $\widehat{\mathfrak{A}} \subseteq A^g$. Moreover, as the inverse is a bijective function, we have $|\widehat{\mathfrak{A}}| = |\mathfrak{A}|$.

We set $q = \lfloor g \log|A| - \log(g \cdot \max A + 1) \rfloor$ such that $2^q \leq |A|^g/(g \cdot \max A + 1) \leq |\widehat{\mathfrak{A}}|$, that is, even when using vectors from $\widehat{\mathfrak{A}}$ for the encodings, we can still encode all partial assignments. As a last step, we choose an (injective) mapping $e\colon 2^q \to \widehat{\mathfrak{A}}$ to fix the encoding.

Given a selection S of vertices from G_ϕ, for each $i \in [\,1\,..\,t\,]$, $\ell \in [\,1\,..\,g\,]$, and $j \in [\,0\,..\,m\,]$, we define two states $a_{i,\ell}^j$ and $\overline{a}_{i,\ell}^j$ as follows. Let $T_{i,\ell}^j \coloneqq S \cap N_B(w_{i,\ell}^j)$ be the number of selected neighbors of $w_{i,\ell}^j$ in $B_{i,\ell}^j$, and let $\overline{T}_{i,\ell}^j \coloneqq S \cap N_{\overline{B}}(w_{i,\ell}^j)$ be the number of selected neighbors of $w_{i,\ell}^j$ in $\overline{B}_{i,\ell}^j$.

$$\begin{aligned} &\text{If } w_{i,\ell}^j \text{ is selected, then we set } a_{i,\ell}^j = \sigma_{T_{i,\ell}^j} \text{ and } \overline{a}_{i,\ell}^j = \sigma_{\overline{T}_{i,\ell}^j}. \\ &\text{If } w_{i,\ell}^j \text{ is } \textit{not} \text{ selected, then we set } a_{i,\ell}^j = \rho_{T_{i,\ell}^j} \text{ and } \overline{a}_{i,\ell}^j = \rho_{\overline{T}_{i,\ell}^j}. \end{aligned} \tag{9.1}$$

Note that, for each $i \in [\,1\,..\,t\,]$ and $j \in [\,1\,..\,m\,]$, a selection of vertices S_i^j from Z_i^j determines the states $a_{i,\ell}^j$ and $\overline{a}_{i,\ell}^{j-1}$ for each $\ell \in [\,1\,..\,g\,]$.

Using these states, we now have everything in place to conveniently define the relations R_i^j. For each $i \in [\,1\,..\,t\,]$ and $j \in [\,1\,..\,m\,]$, S_i^j is in R_i^j if and only if all of the following hold:

- $a_{i,1}^j \ldots a_{i,g}^j$ is in $\widehat{\mathfrak{A}} \subseteq A^g$, and it is the encoding $e(\pi_i)$ of a partial assignment π_i of the group of variables F_i.

- For each $\ell \in [1..g]$, the state $\overline{a}_{i,\ell}^{j-1}$ complements $a_{i,\ell}^{j}$ in the sense that $\overline{a}_{i,\ell}^{j-1} = \mathsf{inv}^{\sigma,\rho}(a_{i,\ell}^{j})$. Note that therefore $w_{i,\ell}^{j}$ is selected if and only if $w_{i,\ell}^{j-1}$ is selected.
- If the vertex c_{i-1}^{j} is selected, then the vertex c_{i}^{j} must also be selected.
- If c_{i-1}^{j} is not selected, then c_{i}^{j} is selected if and only if the encoded assignment π_i satisfies the clause C_j.

Similarly, we define R_i^0 and R_i^{m+1} for each $i \in [1..t]$. A selection of vertices S_i^0 from Z_i^0 determines, for each $\ell \in [1..g]$, the state $a_{i,\ell}^0$; and a selection of vertices S_i^m from Z_i^m determines, for each $\ell \in [1..g]$, the state $\overline{a}_{i,\ell}^m$ where we follow the above procedure.

S_i^0 is in R_i^0 if and only if

- $a_{i,1}^0 \dots a_{i,g}^0$ is in $\widehat{\mathfrak{A}} \subseteq A^g$, and it is the encoding $e(\pi_i)$ of a partial assignment π_i of the group of variables F_i.

In order to define R_i^{m+1}, we define, for each $\ell \in [1..g]$, the auxiliary state $a_{i,\ell}^{m+1}$ with $a_{i,\ell}^{m+1} := \mathsf{inv}^{\sigma,\rho}(\overline{a}_{i,\ell}^m)$. Then, S_i^{m+1} is in R_i^{m+1} if and only if

- $a_{i,1}^{m+1} \dots a_{i,g}^{m+1}$ is in $\widehat{\mathfrak{A}} \subseteq A^g$, and it is the encoding $e(\pi_i)$ of a partial assignment π_i of the group of variables F_i.

9.1.2 Correctness of the Construction

In the following two claims, we show the correctness of the construction. Afterward, we prove some properties of the graph which we need later to obtain tight bounds for the lower bounds.

Claim 9.3. *If ϕ is satisfiable, then (σ,ρ)-DomSet$^{\text{Rel}}$ has a solution on G_ϕ.*

Proof. Let π be a satisfying assignment for ϕ, and let $\pi_1, \dots, \pi_t$ be the corresponding partial assignments to the variable groups $F_1, \dots, F_t$. For each $i \in [1..t]$, $e_i := e(\pi_i) \in \widehat{\mathfrak{A}} \subseteq A^g \subseteq \mathbb{A}^g$ denotes the corresponding encoding. So, these encodings as a whole form an element $x \in A^{tg}$. By the definition of an A-manager of rank gt, for each $x \in A^{tg}$, there is a unique solution S_x that manages x.

We now define a selection S of vertices of the graph G_ϕ, and afterward show that S is a solution for this instance of (σ,ρ)-DomSet$^{\text{Rel}}$. For each $j \in [1..m]$, the following vertices are selected:

- From the A-manager of rank gt J^j, we select vertices according to the solution S_x.
- Let $i^* \in [1..t]$ be the smallest index such that F_{i^*} contains a variable that satisfies the clause C_j under the assignment π. A vertex c_i^j is selected if and only if $i \geq i^*$.
- We lift the selection to the $\{\sigma_{s_{\min}}, \rho_{r_{\min}}\}$-providers: if c_i^j is selected, then we select vertices according to the $\sigma_{s_{\min}}$-state of the attached $\{\sigma_{s_{\min}}, \rho_{r_{\min}}\}$-provider, and otherwise, according to the $\rho_{r_{\min}}$-state.

In order to show that this selection of vertices is a feasible solution of (σ,ρ)-DomSet$^{\text{Rel}}$, we have to verify that

1. for each $j \in [1..m]$, the corresponding $\mathtt{HW}_{=0}^{(1)}$- and $\mathtt{HW}_{=1}^{(1)}$-relations are satisfied, i.e., c_0^j is unselected, and c_t^j is selected,

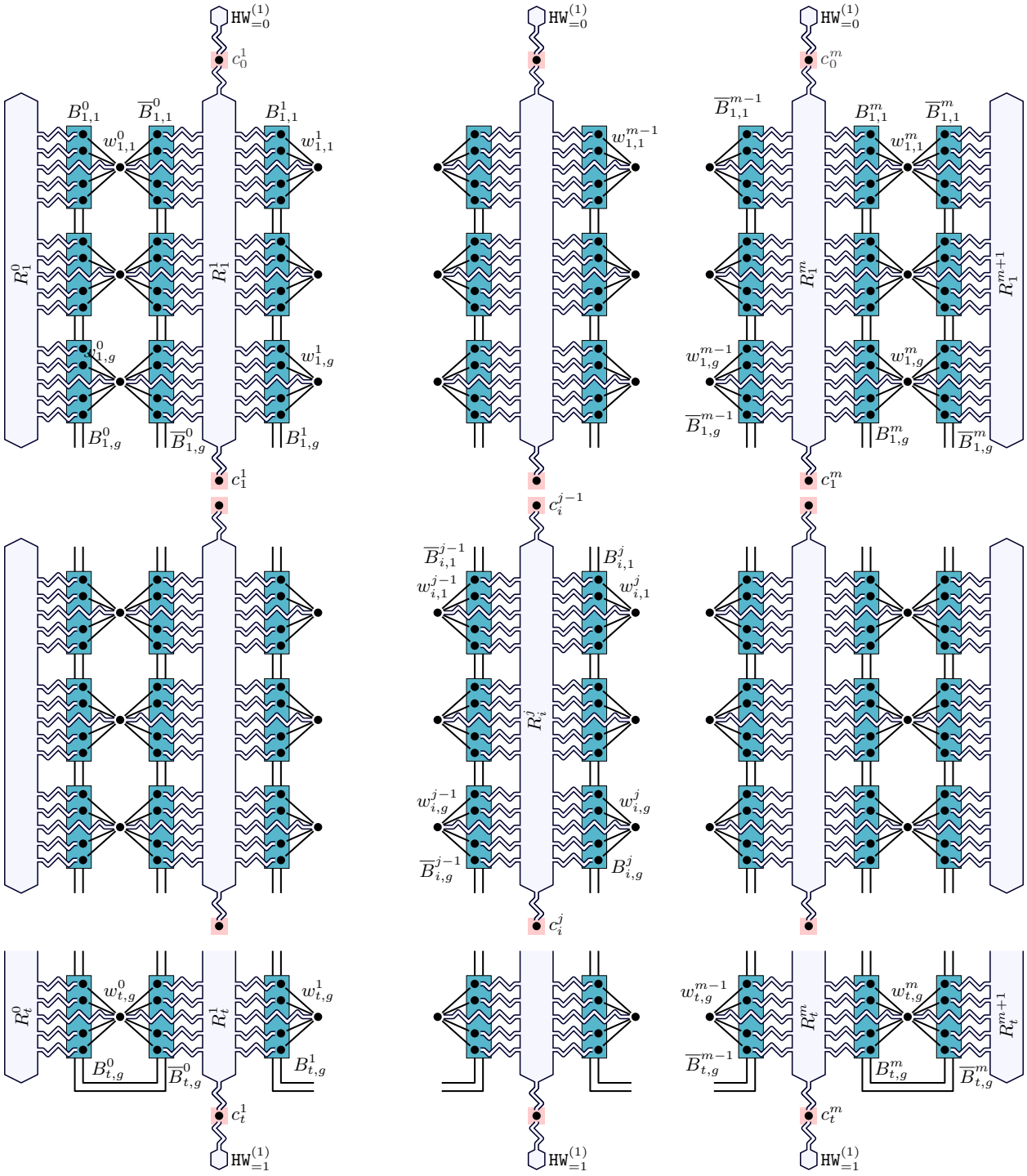


Figure 9.1: The construction in Section 9.1 for the lower bound for the problem with relations. Vertices are shown by dots, $\{\sigma_{s_{\min}}, \rho_{r_{\min}}\}$-providers are shown in red, the blocks from the A-manager are shown in blue, and relations by hexagons.

2. for each i and j, the relation R_i^j is satisfied, and

3. all simple vertices of G_ϕ have a feasible number of selected neighbors, according to their own selection status.

Since π is a satisfying assignment, for each $j \in [1..m]$, there is an index $i^* \in [1..t]$ such that π_{i^*} satisfies C_j. This gives Item 1.

In order to verify Item 2, let $i \in [1..t]$ and $j \in [1..m]$. We check that R_i^j is satisfied. The relevant selected vertices are $S_i^j = S \cap Z_i^j$. Recall from Equation (9.1) that S_i^j determines the states $a_{i,\ell}^j$ and $\overline{a}_{i,\ell}^{j-1}$ for each $\ell \in [1..g]$. The states $a_{i,1}^j, \ldots, a_{i,g}^j$ are determined by the selected vertices of J^j, i.e. an A-manager of rank gt, and they are determined by the selection status of its distinguished vertices and their selected neighbors in the corresponding B-blocks. As we selected vertices of J^j according to the solution S_x, the fact that $a_{i,1}^j \ldots a_{i,g}^j$ is the encoding e_i of π_i follows from the fact that S_x manages x (see Definition 6.2).

Similarly, the states $\overline{a}_{i,1}^{j-1}, \ldots, \overline{a}_{i,g}^{j-1}$ are determined by the selected vertices of J^{j-1}, and in particular, by the selection status of its distinguished vertices and their selected neighbors in the corresponding $\overline{B}$-blocks. Let $\ell \in [1..g]$. Since we selected vertices of J^{j-1} according to S_x as well, we have $a_{i,\ell}^{j-1} = a_{i,\ell}^j$. By Definition 6.2, $\overline{a}_{i,\ell}^{j-1}$ complements the state $a_{i,\ell}^{j-1} = a_{i,\ell}^j$ as required by R_i^j.

It is clear that in our selection we include c_i^j into S whenever we include c_{i-1}^j. Finally, suppose c_{i-1}^j is not selected. If c_i^j is selected, then we have $i = i^*$, which means that π_i satisfies C_j by the choice of i^*. Conversely, if we do not select c_i^j, then $i < i^*$, which means that π_i does not satisfy C_j. This shows that the selection S satisfies R_i^j.

It remains to show Item 3. Let v be a simple vertex in G_ϕ. If v is a non-portal vertex of a $\{\sigma_{s_{\min}}, \rho_{r_{\min}}\}$-provider, then, by the definition of such providers (Definition 3.7), v has a feasible number of selected neighbors (and, as it is not a portal, it does not have any neighbors outside of the respective provider). In the remaining cases, v is either a vertex of some J^j, i.e., an A-manager of rank gt, or it is a vertex of the form c_i^j. Suppose v is in some J^j. By Definition 6.2, v has a feasible number of neighbors within J^j. If v has neighbors in G_ϕ that are not part of J^j, then v is in some set Z_i^j and each of its additional neighbors is a complex vertex which is not selected by assumption. This means that v does not have any selected neighbors other than the feasible number within J^j. Finally, the same argument holds for $v = c_i^j$: it has a feasible number of selected neighbors within the attached $\{\sigma_{s_{\min}}, \rho_{r_{\min}}\}$-provider, but as all other neighbors are complex vertices with relations (some R_i^j, $\mathtt{HW}_{=0}^{(1)}$, or $\mathtt{HW}_{=1}^{(1)}$), it does not receive any more selected neighbors. □

In the following, we show the reverse direction from the proof of correctness.

Claim 9.4. *If (σ, ρ)-DomSet$^{\text{Rel}}$ has a solution on G_ϕ, then ϕ is satisfiable.*

Proof. Let S be a selection of vertices from G_ϕ that is a solution of (σ, ρ)-DomSet$^{\text{Rel}}$. Recall that S determines the states $a_{i,\ell}^j$, $\overline{a}_{i,\ell}^j$ and numbers $T_{i,\ell}^j$, $\overline{T}_{i,\ell}^j$, as described in Equation (9.1).

Observe that, for all i, ℓ, we have that $w_{i,\ell}^j$ is selected if and only if $w_{i,\ell}^{j-1}$ is selected. This follows from the definition of the relation R_i^j. First, suppose the vertices $w_{i,\ell}^0, \ldots, w_{i,\ell}^m$ are selected.

Assume further that σ is finite. As S is a solution, $w_{i,\ell}^j$ can have at most s_{top} selected neighbors. Hence, for each $j \in [0..m]$, we have $T_{i,\ell}^j + \overline{T}_{i,\ell}^j \leq s_{\text{top}}$. Moreover, by the definition of the relations R_i^j, for each $j \in [1..m]$, we get $\overline{T}_{i,\ell}^{j-1} + T_{i,\ell}^j = s_{\text{top}}$. Combining both constraints yields $T_{i,\ell}^0 \leq \cdots \leq T_{i,\ell}^m$ and $\overline{T}_{i,\ell}^0 \geq \cdots \geq \overline{T}_{i,\ell}^m$.

Now, assume σ is simple cofinite. As S is a solution, $w^j_{i,\ell}$ has at least s_{top} selected neighbors. Hence, for each $j \in [0..m]$, we have $T^j_{i,\ell} + \overline{T}^j_{i,\ell} \geq s_{\text{top}}$. As before, we get by the definition of the relations R^j_i that, for each $j \in [1..m]$, $\overline{T}^{j-1}_{i,\ell} + T^j_{i,\ell} = s_{\text{top}}$. Combining both constraints yields $T^0_{i,\ell} \geq \cdots \geq T^m_{i,\ell}$ and $\overline{T}^0_{i,\ell} \leq \cdots \leq \overline{T}^m_{i,\ell}$ in the simple cofinite case.

It is easy to check that we get the same constraints if the vertices $w^j_{i,\ell}$ are not selected.

We define new values $\widehat{T}^j_{i,\ell}$.

$$\widehat{T}^j_{i,\ell} \coloneqq \begin{cases} s_{\text{top}} - T^j_{i,\ell} & w^j_{i,\ell} \text{ is selected } \wedge \sigma \text{ is simple cofinite} \\ r_{\text{top}} - T^j_{i,\ell} & w^j_{i,\ell} \text{ is not selected } \wedge \rho \text{ is simple cofinite} \\ T^j_{i,\ell} & \text{else} \end{cases} \tag{9.2}$$

One can easily check that we get $\widehat{T}^0_{i,\ell} \leq \cdots \leq \widehat{T}^m_{i,\ell}$ independently from σ or ρ being finite or cofinite.

Recall that we only use encodings from $\widehat{\mathfrak{A}}$ (all other encodings are not accepted by the relations R^j_i), and hence, they have a one-to-one correspondence to the encodings in $\mathfrak{A}$ which all have the same weight. Therefore, for all $i \in [1..t]$ and $j \in [0..m+1]$, we get: $\sum_{\ell=1}^{g} \widehat{T}^j_{i,\ell} = w$. Now, assume that for some $i \in [1..t]$, $j \in [1..m]$, and $\ell \in [1..g]$, we have $\widehat{T}^j_{i,\ell} < \widehat{T}^{j+1}_{i,\ell}$. This implies that there is some ℓ' such that $\widehat{T}^j_{i,\ell'} > \widehat{T}^{j+1}_{i,\ell'}$, which contradicts the above assumption. Hence, we obtain $\widehat{T}^0_{i,\ell} = \cdots = \widehat{T}^m_{i,\ell}$ for each i and ℓ. Combined with Equation (9.2) this implies

$$T^0_{i,\ell} = \cdots = T^m_{i,\ell} \quad \text{and} \quad \overline{T}^0_{i,\ell} = \cdots = \overline{T}^m_{i,\ell} \quad \text{for each } i \text{ and } \ell. \tag{9.3}$$

We define an assignment π of ϕ as follows. For each i, $a^1_{i,1} \dots a^1_{i,g} \in A^g$ is subject to R^1_i, and therefore, it is the encoding of a partial assignment π_i of the group of variables F_i. Let π be the assignment comprised of these partial assignments.

It remains to verify that π satisfies ϕ. For $j \in [1..m]$, we verify that π satisfies the clause C_j. Consider the vertices $c^j_0, \dots, c^j_t$. Since c^j_0 and c^j_t are subject to a $\mathtt{HW}^{(1)}_{=0}$- and $\mathtt{HW}^{(1)}_{=1}$-relation, respectively, we have $c^j_0 \notin S$ and $c^j_t \in S$. Hence, there is an $i \in [1..t]$ for which c^j_{i-1} is not selected, but c^j_i is. As c^j_{i-1} and c^j_i are subject to R^j_i, it follows that $a^j_{i,1} \dots a^j_{i,g}$ encodes a partial assignment that satisfies the clause C_j. The equalities from Equation (9.3) imply $a^1_{i,1} \dots a^1_{i,g} = a^j_{i,1} \dots a^j_{i,g}$. Therefore, π_i satisfies C_j, and consequently π satisfies C_j. □

This finishes the correctness of the construction. As a last step we analyze the graph and its tree decomposition.

Claim 9.5. *There is some function f that depends only on g, s_{top}, and r_{top} such that the graph G_ϕ has at most $m \cdot t \cdot f(g, s_{\text{top}}, r_{\text{top}})$ vertices, pathwidth at most $gt + f(g, s_{\text{top}}, r_{\text{top}})$, and each relation has arity at most $f(g, s_{\text{top}}, r_{\text{top}})$.*

Proof. Every vertex in G_ϕ is part of (at least) one of the following:

- an A-manager of rank gt (and there are $m+1$ of these) including the complex vertices therein,
- a $\{\sigma_{s_{\min}}, \rho_{r_{\min}}\}$-provider Q^j_i (and there are $(t+1)m$ of these),
- the complex vertices with relations; there are $2m$ of the form $\mathtt{HW}^{(1)}_{=0}$ or $\mathtt{HW}^{(1)}_{=1}$, $t(m+2)$ of the form R^j_i, and $(t+1)m$ relations attached to Q^j_i and c^j_i.

We analyze the number of vertices in these components.

- By Definition 6.2, the size of an A-manager of rank gt is of the form $gt \cdot b$, where b is an upper bound on the size of the blocks. Thus, b depends only on s_{top} and r_{top}.
- The size of the $\{\sigma_{s_{\min}}, \rho_{r_{\min}}\}$-providers Q_i^j depends only on $s_{\min}$ and $r_{\min}$ (see the construction in Lemma 7.1).
- Note that the arity of the realizations we use is upper bounded by the size of the sets Z_i^j or some function depending on the size of the blocks from the manager. Recall that

$$Z_i^j = \{c_{i-1}^j, c_i^j\} \cup \bigcup_{\ell \in [1\,..\,g]} \left(w_{i,\ell}^{j-1} \cup N_{\overline{B}}(w_{i,\ell}^{j-1}) \cup w_{i,\ell}^j \cup N_B(w_{i,\ell}^j)\right).$$

  The size of the sets $N_{\overline{B}}(w_{i,\ell}^{j-1})$ and $N_B(w_{i,\ell}^j)$ is bounded by that of the blocks of the A-manager of rank gt. So, once again, their size depends only on s_{top} and r_{top}. So, there is some function f'' such that, for each i and j, we have $|Z_i^j| \leq g \cdot f''(s_{\text{top}}, r_{\text{top}})$.

This proves the bound on the size of G_ϕ. Moreover, it implies the bound of the arity of the relations.

We use a node search strategy to bound the pathwidth of G_ϕ (see [20, Section 7.5]). For each $i \in [1\,..\,t]$ and $j \in [1\,..\,m]$, we define a set Y_i^j that contains the following vertices:

- The vertices subject to relation R_i^j and the complex vertex the relation is assigned to. (This includes the set Z_i^j.)
- The vertices in $\overline{B}_{i,\ell}^{j-1}$ and $B_{i,\ell}^j$ for all $\ell \in [1\,..\,g]$. (Some of these vertices are also contained in Z_i^j.)
- The vertices of the $\{\sigma_{s_{\min}}, \rho_{r_{\min}}\}$-providers Q_{i-1}^j and Q_i^j attached to c_{i-1}^j and c_i^j and the vertices of the corresponding relations.
- If $i - 1 = 0$, the vertex of the $\mathtt{HW}_{=0}^{(1)}$-relation attached to c_{i-1}^j.
- If $i = t$, the vertex of the $\mathtt{HW}_{=1}^{(1)}$-relation attached to c_i^j.

We similarly define, for each $i \in [1\,..\,t]$, the sets Y_i^0 and Y_i^{m+1}. To simplify notation, we consider Y_i^j to be the empty set if $i \notin [1\,..\,t]$ or $j \notin [0\,..\,m+1]$.

We now describe $m+2$ stages of selecting vertices as positions for searchers, where each stage consists of t rounds. Let $i \in [1\,..\,t]$. In the ith round of the jth stage, where $j \in [0\,..\,m+1]$, the selected vertices are $Y_i^j \cup Y_{i+1}^j \cup Y_t^{j-1}$ together with, for each $z \in [1\,..\,i-1]$ and $\ell \in [1\,..\,g]$, the vertex $w_{z,\ell}^j$ (if it exists), and for each $z \in [i+1\,..\,t]$ and $\ell \in [1\,..\,g]$, the vertex $w_{z,\ell}^{j-1}$ (if it exists).

Using Figure 9.1, it is straightforward to verify that selecting vertices according to the described stages cleans the graph as required. Note that we select the set Y_t^{j-1} in all rounds of the jth stage to prevent "recontamination" via the edges between the blocks $B_{t,g}^{j-1}$ and $\overline{B}_{t,g}^{j-1}$. Moreover, for each complex vertex, there is some stage in some round where this vertex together with its neighborhood are covered (that is, they are in the same bag of the decomposition).

Now, consider the number of searchers/the selected vertices in each stage. By the previously specified bounds on the size of blocks, and $\{\sigma_{s_{\min}}, \rho_{r_{\min}}\}$-providers, there is some function f that depends only on g, s_{top}, and r_{top}, such that the size of Y_i^j is bounded from above by $f(g, s_{\text{top}}, r_{\text{top}})$ for each i and j. Hence, at each stage we select at most $gt + O(1) \cdot f(g, s_{\text{top}}, r_{\text{top}})$ vertices. This is an upper bound on the node search number of G_ϕ, and consequently an upper bound on its pathwidth. □

The following observation follows directly from the construction and the previous proofs.

Observation 9.6. *There is a one-to-one correspondence between satisfying assignments for ϕ and solutions for G_ϕ.*

In all solutions for G_ϕ, each vertex $w^j_{i,\ell}$ has exactly s_{top} neighbors if it is selected, and exactly r_{top} neighbors if it is not selected.

9.1.3 Finalizing the Lower Bound

It remains to prove Lemma 6.5, which we restate here for convenience.

Lemma 6.5 (Lower Bound for (σ,ρ)-DomSet$^{\text{Rel}}$)**.** *Let $\sigma,\rho \subseteq \mathbb{Z}_{\geq 0}$ be two fixed, non-empty, and finite or simple cofinite sets with $0 \notin \rho$. Suppose there is an $A \subseteq \mathbb{A}$ that is closed under the inverse with respect to σ,ρ such that there is an A-manager.*

Then, (σ,ρ)-DomSet$^{\text{Rel}}$ cannot be solved in time $(|A|-\varepsilon)^{k+O(1)} \cdot n^{O(1)}$, even if we are given a path decomposition of width k, and all relations have arity at most $O(1)$, unless SETH fails.

Proof. Toward a proof by contradiction, assume that there is such an algorithm for (σ,ρ)-DomSet$^{\text{Rel}}$. We show that there is an algorithm solving SAT in time $(2-\delta)^n$ for some $\delta > 0$, where n is the number of variables.

To use the construction from the previous sections, it remains to choose the values for q and g. Let $\varepsilon' = \log_{|A|}(|A|-\varepsilon) < 1$. We choose some $\alpha > 1$ such that $\alpha \cdot \varepsilon' \leq \delta' = \log(2-\delta) < 1$ for some $\delta > 0$. We choose g large enough such that it satisfies $g \log |A| \leq \alpha \lfloor g \log |A| - \log(g \cdot \max A + 1) \rfloor$. Finally, set $q = \lfloor g \log |A| - \log(g \cdot \max A + 1) \rfloor$. Observe that we can actually encode all partial assignments of one group. Using these parameters, we can construct a (σ,ρ)-DomSet$^{\text{Rel}}$ instance. For the size bound and pathwidth, observe that $|A| \leq |\mathbb{A}|$, and thus, $|A|$ can be bounded in terms of s_{top} and r_{top}. Moreover, g only depends on ε, δ, and $|A|$. Since σ,ρ are fixed, any term only depending on ε, δ, s_{top}, and r_{top} can be treated as a constant. Based on this, from Claim 9.5 it follows that there is some fixed function f such that G_ϕ has pathwidth at most $tg + f(g, s_{\text{top}}, r_{\text{top}}) = tg + O(1)$, where $t = \lceil \frac{n}{q} \rceil$, and that a path decomposition of this size can be computed efficiently. The graph has size

$$O(\lceil \tfrac{n}{q} \rceil \cdot m \cdot f(g, s_{\text{top}}, r_{\text{top}})) = O(n \cdot m \cdot f'(g, s_{\text{top}}, r_{\text{top}})) = O(n \cdot m).$$

Then, we run the fast (σ,ρ)-DomSet$^{\text{Rel}}$ algorithm on this example:

$$\begin{aligned}(|A|-\varepsilon)^{k+O(1)} \cdot n^{O(1)} \leq& (|A|-\varepsilon)^{tg+O(1)} \cdot n^{O(1)} \\ \leq& (|A|-\varepsilon)^{\lceil \frac{n}{q} \rceil g+O(1)} \cdot n^{O(1)} \\ \leq& (|A|-\varepsilon)^{\frac{n}{q} g+g+O(1)} \cdot n^{O(1)}\end{aligned}$$

By the same argument as above, we can ignore the term of $g + O(1)$ in the exponent as it only contributes a constant factor to the runtime.

$$\begin{aligned}\leq& (|A|-\varepsilon)^{\frac{n}{q} \cdot g} \cdot n^{O(1)} \\ \leq& 2^{\log |A| \cdot \varepsilon' \cdot \frac{n}{q} \cdot g} \cdot n^{O(1)} \\ \leq& 2^{\varepsilon' \cdot \frac{n}{\lfloor g \log |A| - \log(g \cdot \max A + 1) \rfloor} \cdot g \cdot \log |A|} \cdot n^{O(1)}\end{aligned}$$

By our choice of g we get:

$$\leq 2^{\varepsilon' \alpha n} \cdot n^{O(1)} \leq 2^{\delta' n} \cdot n^{O(1)} = (2-\delta)^n \cdot n^{O(1)}.$$

But, this directly contradicts SETH. □

9.2 Counting Problem

Now we move to the proof of the intermediate lower bound for the counting version. By Observation 9.6, the previous reduction is parsimonious. Hence, the lower bound for the decision version from Lemma 6.5 directly transfers to the counting version.

Corollary 9.7. *Let $\sigma, \rho \subseteq \mathbb{Z}_{\geq 0}$ be two fixed and non-empty sets which are* finite *or* simple cofinite*. Suppose there is an $A \subseteq \mathbb{A}$ that is closed under the inverse with respect to σ, ρ such that there is an A-manager.*

Then, (σ, ρ)-#DomSet$^{\text{Rel}}$ cannot be solved in time $(|A| - \varepsilon)^{k+O(1)} \cdot n^{O(1)}$, even if we are given a path decomposition of width k, and all relations have arity at most $O(1)$, unless #SETH fails.

In what follows, we extend this result to (σ, ρ)-#DomSet$^{\text{Rel}}$, where we allow the sets σ and ρ to also be arbitrary cofinite sets and not only *simple cofinite* sets. See Figure 6.1 for an illustration of the process.

Our first step is to allow σ to be cofinite. For this we introduce the *relation weighted* version of (σ, ρ)-#DomSet$^{\text{Rel}}$. For this problem, each accepted input of a relation might be assigned a weight such that the relation contributes by this factor to the number of solutions, rather than just contributing 1 in the unweighted case or 0 if the input is not accepted.

Lemma 9.8. *Let ρ be a fixed and non-empty set. Let σ, σ' be two fixed and non-empty sets where σ is cofinite and σ' is simple cofinite such that* $\max(\mathbb{Z}_{\geq 0} \setminus \sigma) = \max(\mathbb{Z}_{\geq 0} \setminus \sigma')$.

There is a treewidth-preserving reduction from (relation weighted) (σ', ρ)-#DomSet$^{\text{Rel}}$ to relation weighted (σ, ρ)-#DomSet$^{\text{Rel}}$ introducing at most $O(1)$ new weights.

Proof. We can obviously assume that $\sigma \neq \sigma'$. We use the following auxiliary provider in the proof.

Claim 9.9. *Let σ, ρ be fixed and non-empty. If σ is cofinite, then there is a parsimonious $\{\sigma_0, \sigma_1, \rho_0\}$-provider which uses relations.*

Proof of Claim. Without loss of generality, we can assume that $s_{\text{top}} \geq r_{\text{top}}$. Let u be the portal vertex to which we add a single neighbor, say v. We make v adjacent to s_{top} additional vertices $v_1, \ldots, v_{s_{\text{top}}}$ each serving as the portal of a new $\{\sigma_{s_{\text{top}}}, \rho_{r_{\min}}\}$-provider. For ease of notation we let $F = \{\sigma_{s_{\text{top}}}, \rho_{r_{\min}}\}$. We show that there is a partial solution for each state.

- **$\boldsymbol{\rho_0}$** The vertices $v_1, \ldots, v_{r_{\text{top}}}$ are selected, which is extended to the $\sigma_{s_{\text{top}}}$-state of the F-provider. The remaining vertices, including v, are unselected, and this is extended to the $\rho_{r_{\min}}$-states of the F-providers. Hence, the vertices $v_1, \ldots, v_{r_{\text{top}}}$ have s_{top} neighbors from the providers and no other neighbors. The vertices $v_{r_{\text{top}}+1}, \ldots, v_{s_{\text{top}}}$ have $r_{\min}$ neighbors from the provider and no other neighbors.

- **$\boldsymbol{\sigma_0}$** The vertices $v_1, \ldots, v_{r_{\text{top}}-1}$ are selected and the selection is extended to the F-providers. The remaining vertices are unselected, which is again extended to the F-provider. The vertex v is unselected and gets $r_{\text{top}} - 1$ neighbors from the F-provider, but already has u as a selected neighbor. The vertices $v_{r_{\text{top}}}, \ldots, v_{s_{\text{top}}}$ get $r_{\min}$ neighbors from the F-provider and no other neighbors since u is unselected.

- **$\boldsymbol{\sigma_1}$** The vertex v and all the vertices v_i are selected, and this is extended to the F-providers. Hence, they have s_{top} neighbors in the provider and one neighbor outside, namely v. Moreover, u is selected and has $s_{\text{top}} + 1$ neighbors.[10]

[10] This also works for state ρ_1.

We add a relation which ensures that there are exactly three partial solutions, each corresponding to one of the mentioned states. By this, all other potential partial solutions are ruled out. ◁

Observe that when we just replace the set σ' by the set σ, all previous solutions are still valid. However, the converse is not true; there might be new solutions which are not valid before as the degree of some vertex might be less than s_{top}. We modify the graph, or rather each vertex v, such that the degree of v cannot be smaller than s_{top}. As we work with the counting version, we additionally introduce (negative) weights which allow us to compensate for some other contribution. Hence, we change the vertex such that the invalid combinations cancel out.

The basic idea is to add neighbors to v which can be selected almost arbitrarily. By the degree constraints of v based on the set σ, some selections are allowed and some are not. We add weights w_i based on the number of additional vertices v gets as neighbors such that in total the vertex behaves as if it would have set σ'.

Formally, we apply the following modification for each vertex $v \in V(G)$. To simplify notation set $Z = \{\sigma_0, \sigma_1, \rho_0\}$. Note that, by Claim 9.9, a Z-provider exists (if we allow additional relations). We introduce s_{top} copies of the Z-provider with v as portal. Let $v_1, \ldots, v_{s_{\text{top}}}$ be the neighbors of v in these copies. We add a relation R observing $v, v_1, \ldots, v_{s_{\text{top}}}$ which ensures that if v is selected, then there is always a $\gamma \in [0 \,..\, s_{\text{top}}]$ such that the vertices $v_1, \ldots, v_\gamma$ are selected and the vertices $v_{\gamma+1}, \ldots, v_{s_{\text{top}}}$ are not selected. Moreover, if v is unselected, then the relation R enforces that $v_1, \ldots, v_{s_{\text{top}}}$ are unselected. If exactly the vertices $v, v_1, \ldots, v_\gamma$ are selected, then the relation accepts with a weight w_γ which is defined shortly. To conclude the construction, it remains to define the weights w_i.

If v is adjacent to α selected neighbors in the graph (excluding the vertices $v_1, \ldots, v_{s_{\text{top}}}$), then by the degree constraints of σ, we can only select vertices $v_1, \ldots, v_\gamma$ such that $\alpha + \gamma \in \sigma$. Moreover, these valid weights have to sum up to 0 if $\alpha < s_{\text{top}}$, and sum up to 1 if $\alpha \geq s_{\text{top}}$. Formally, we have to define the weights w_i such that the following equation holds for all $\alpha \in [0 \,..\, s_{\text{top}}]$:

$$\sum_{\substack{\gamma=0: \\ \alpha+\gamma \in \sigma}}^{s_{\text{top}}} w_\gamma = \begin{cases} 1 & \alpha \geq s_{\text{top}} \\ 0 & \text{else} \end{cases}. \tag{9.4}$$

Claim 9.10. *We can (efficiently) find the values for the w_i's such that they satisfy the constraints from Equation (9.4).*

Proof of Claim. Observe that there are $s_{\text{top}} + 1$ unknown values and $s_{\text{top}} + 1$ many constraints. We get that $s_{\text{top}} - 1 \notin \sigma$ since $s_{\text{top}} > 0$. From the definition in Equation (9.4), the sums for $\alpha = s_{\text{top}}$ and $\alpha = s_{\text{top}} - 1$ differ by exactly one weight, namely w_0. Hence, we can determine the value for w_0 and eliminate it from the constraints. Repeating this procedure yields a solution for the equations and provides the values for all w_i. ◁

It remains to show that we can actually select the vertices correspondingly. First, assume that v is unselected. Then, all vertices v_i must be unselected. That is, each Z-provider gives state ρ_0 to v.

Now, assume that v is selected. Then, the first γ many Z-providers give state σ_1 to v, and the remaining providers give state σ_0 to v. This precisely corresponds to the case when the vertices $v_1, \ldots, v_\gamma$ are selected and the vertices $v_\gamma, \ldots, v_{s_{\text{top}}}$ are unselected. The weight of the relation is then w_γ.

The relation R has arity $s_{\text{top}} + 1$. Hence, we can duplicate one bag containing v and add all s_{top} vertices $v_1, \ldots, v_{s_{\text{top}}}$ to that bag. Then, the closed neighborhood of the relation is contained in one bag. As σ is fixed, s_{top} is a constant, and hence, treewidth increases by a constant only. □

Similarly to the previous result, we show the following dual result, that is, we allow ρ to be cofinite.

Lemma 9.11. *Let σ be a fixed and non-empty set. Let ρ, ρ' be two fixed sets where ρ is cofinite and ρ' is simple cofinite such that $\max(\mathbb{Z}_{\geq 0} \setminus \rho) = \max(\mathbb{Z}_{\geq 0} \setminus \rho')$.*

There is a treewidth-preserving reduction from (relation weighted) (σ, ρ')-#DomSet$^{\text{Rel}}$ to relation weighted (σ, ρ)-#DomSet$^{\text{Rel}}$ introducing at most $O(1)$ new weights.

Proof. We use the following auxiliary provider in the proof which complements the result from Claim 9.9.

Claim 9.12. *Let σ, ρ be fixed and non-empty. If ρ is cofinite, then there is a parsimonious $\{\sigma_0, \rho_0, \rho_1\}$-provider which uses relations.*

Proof of Claim. Let u be the portal vertex and make it adjacent to a new vertex v which serves as the portal of a $\{\sigma_{s_{\min}}, \rho_{r_{\text{top}}}\}$-provider. Once more we let $F = \{\sigma_{s_{\min}}, \rho_{r_{\text{top}}}\}$ to simplify notation.

For states σ_0 and ρ_0, v is unselected which is extended to the F-provider. Hence, v has at least r_{top} neighbors.

For state ρ_1, v is selected which is again extended to the F-provider. Hence, v has $s_{\min}$ neighbors as u is not selected.

We add a relation which ensures that there are exactly three partial solutions, each corresponding to one of the mentioned states. By this all other potential partial solutions are ruled out. ◁

The proof follows precisely the same idea as before. The most notable difference is that we use Z-providers where we now have $Z = \{\sigma_0, \rho_0, \rho_1\}$, that is, the state σ_1 is not allowed, but the state ρ_1 is allowed. Observe that this is not an issue as we never need to add neighbors to v if the vertex is selected, actually the relation forbids to add neighbors to v in this case.

Then, the remaining part of the proof follows in precisely the same way. □

In the next step, we remove the weighted relations by allowing weighted vertices. In this setting, the vertices contribute with their weight to the solution whenever they are selected.

Lemma 9.13. *There is a treewidth-preserving reduction from relation weighted (σ, ρ)-#DomSet$^{\text{Rel}}$ with $O(1)$ different weights for each relation to vertex weighted (σ, ρ)-#DomSet$^{\text{Rel}}$ with the same weights.*

Proof. Let R be a weighted relation assigned to a complex vertex v such that R uses q many weights $w_1, \ldots, w_q$. We make v adjacent to q many new vertices $v_1, \ldots, v_q$. Each of these vertices is the portal of a parsimonious $\{\rho_{r_{\min}}, \sigma_{s_{\min}}\}$-provider with relations. For all $i \in [1 \,..\, q]$, we assign weight w_i to vertex v_i. Moreover, we replace relation R by an unweighted relation R' which observes the same vertices as R and additionally also the vertices $v_1, \ldots, v_q$. The new relation R' behaves (i.e., accepts or rejects) in exactly the same way as R on the vertices observed by R, but whenever R accepts with weight w_i, R' forces the vertex v_i to be selected and all the vertices v_j with $i \neq j$ to be unselected.

It remains to argue that the reduction increases the treewidth only by a constant. By assumption there is a bag containing v and its closed neighborhood. We duplicate this bag and add the new vertices $v_1, \ldots, v_q$ to the copy. For each v_i, we duplicate this new bag and add all the vertices from the provider to it. Observe that the size of each $\{\rho_{r_{\min}}, \sigma_{s_{\min}}\}$-provider with relation is bounded by a constant, as the size depends only on σ and ρ, and these two sets are fixed. Hence, the width of the decomposition increases by an additive term of at most $O(1)$. □

The last step now finally removes the weights from the vertices, and thus, also from the instance.

Lemma 9.14. *There is a treewidth-preserving reduction from vertex weighted* (σ,ρ)-#DomSet$^{\text{Rel}}$ *with* $O(1)$ *different weights to unweighted* (σ,ρ)-#DomSet$^{\text{Rel}}$ *without changing the arity of the relations.*

Proof. We use the known interpolation techniques to remove the weights. Assume there are q different weights. We replace each weight w_i by a variable x_i and treat the output as a polynomial P in the q variables $x_1,\dots,x_q$. Observe that there are at most n vertices, and hence, the total degree is at most n for each variable. Hence, if we can realize $n+1$ different weights for each variable x_i, then we can use Lemma 2.5 in [17] to recover the coefficients of P in time polynomial in n. Then, we can output $P(w_1,\dots,w_q)$ to recover the original solution.

It remains to realize $n+1$ different weights for each variable. For this it certainly suffices to realize weights of the form 2^i.

Let v be the vertex for which we want to realize weight 2^i. For this we introduce i new vertices $v_1,\dots,v_i$ which are all portals of a parsimonious $\{\rho_{r_{\min}},\sigma_{s_{\min}}\}$-provider with relations. Moreover, we add complex vertices $u_1,\dots,u_i$. Vertex u_j is adjacent to v and v_j, and we assign relation R to it. This relation R ensures that if v is unselected, then v_j must also be unselected. Otherwise, v_j can be selected or unselected. Whenever v is selected, there are two options to select v_j, and hence, this part contributes a factor of 2^i to the solution. If v is unselected, then there is only one solution which is enforced by the relations, and hence, contributes a factor of 1.

It easily follows that this modification does not change the treewidth too much. Indeed, whenever v is in a bag, we can add each u_j, v_j, and the corresponding $\{\sigma_{s_{\min}},\rho_{r_{\text{top}}}\}$-provider together with the relations one after the other to the bag. By the size bound for the $\{\sigma_{s_{\min}},\rho_{r_{\text{top}}}\}$-provider, the claim follows. □

Combining all the previous results, we get the intermediate lower bound for the counting version in Lemma 6.8 which we restate here for convenience.

Lemma 6.8 (Lower Bound for (σ,ρ)-#DomSet$^{\text{Rel}}$)**.** *Let* $\sigma,\rho\subseteq\mathbb{Z}_{\geq 0}$ *be two fixed, non-empty and finite or cofinite sets. Suppose there is an* $A\subseteq\mathbb{A}$ *that is closed under the inverse with respect to* σ,ρ *such that there is an* A*-manager.*

Then, (σ,ρ)-#DomSet$^{\text{Rel}}$ *cannot be solved in time* $(|A|-\varepsilon)^{k+O(1)}\cdot n^{O(1)}$*, even if we are given a path decomposition of width* k*, and all relations have arity at most* $O(1)$*, unless #SETH fails.*

Proof. Observe that we only have to consider the case when one of the sets is cofinite. Otherwise, the lower bound follows directly from Corollary 9.7.

Assume, without loss of generality, that σ and ρ are cofinite, but not simple cofinite. Let $\sigma'\subseteq\sigma$ and $\rho'\subseteq\rho$ be simple cofinite sets such that $\max(\mathbb{Z}_{\geq 0}\setminus\sigma)=\max(\mathbb{Z}_{\geq 0}\setminus\sigma')$ and $\max(\mathbb{Z}_{\geq 0}\setminus\rho)=\max(\mathbb{Z}_{\geq 0}\setminus\rho')$.

We start with an instance G of (σ',ρ')-#DomSet$^{\text{Rel}}$ where all relations have arity $O(1)$. We sequentially apply Lemmas 9.8, 9.11, 9.13 and 9.14 to obtain polynomially (as we allow Turing-reductions) many instances H_i of (σ,ρ)-#DomSet$^{\text{Rel}}$. By the properties of the treewidth-preserving reductions, we get that $\mathsf{tw}(H_i)\leq\mathsf{tw}(G)+O(1)$ and $|H_i|\leq|G|^{O(1)}$. Moreover, the arity of the relations increased by at most a constant.

Now, assume that the algorithm from the lemma exists for some $\varepsilon>0$ and apply it to all (σ,ρ)-#DomSet$^{\text{Rel}}$ instances H_i. Recovering the solution for G takes in total time

$$\sum_i(|A|-\varepsilon)^{\mathsf{tw}(H_i)+O(1)}\cdot|H_i|^{O(1)}+|G|^{O(1)}=(|A|-\varepsilon)^{\mathsf{tw}(G)+O(1)}\cdot|G|^{O(1)}.$$

By Corollary 9.7, this immediately contradicts #SETH, and hence, finishes the proof. □

10 Realizing Relations

As the third and last step for the lower bound, we wish to express arbitrary relations using graphs with portals. In particular, for a given d-ary relation R, we wish to construct a graph with portals whose compatible language is *equivalent* to R. We call such a graph a *realization of* R. Further, we wish to be able to add a (realization of) a relation to a set of vertices, without invalidating the σ- and ρ-constraints of these vertices. Hence, we require that a realization does not add any *selected* neighbors to the portal vertices.

We first prove the result for the decision version, that is Lemma 6.6, where we ensure that the size of a realization of some relation is always bounded by a function in the arity d, r_{top}, and s_{top}. For the counting version, the situation is not that simple. Indeed, for the proof of Lemma 6.9 we do not give a self-contained construction as for the decision version, but a sequence of reductions from (σ,ρ)-#DomSet$^{\text{Rel}}$ to (σ,ρ)-#DomSet. We prove this result in Section 10.2.

10.1 Decision Problem

As mentioned above, we want to replace a relation R by a graph that realizes R. Recalling Definition 3.7, we can define a realization of R as an L_R-realizer for a language $L_R \subseteq \{\rho_0,\sigma_0\}^d$ such that each string $x \in L_R$ 1-to-1 corresponds with an element in $r \in R$, where $x[\,i\,] = \sigma_0$ if and only if $i \in r$.

Definition 10.1 (Realization). *For an integer $d \geq 1$, let $R \subseteq 2^{[\,1\,..\,d\,]}$ denote a d-ary relation. For an element $r \in R$, write x_r for the length-d string that is σ_0 at every position $i \in r$ and ρ_0 at the remaining positions,*

$$x_r[\,i\,] \coloneqq \begin{cases} \sigma_0 & \text{if } i \in r, \\ \rho_0 & \text{otherwise.} \end{cases}$$

Now, define L_R as

$$L_R \coloneqq \{x_r \mid r \in R\}.$$

Let $H = (G, \{u_1,\ldots,u_d\})$ denote a graph with d portals. Slightly overloading notation, we say that H realizes *R if H realizes L_R, that is, if $L(H) = L_R$. We say that R is realizable if there is a graph with d portals that realizes R.*

10.1.1 Realizing Simple Auxiliary Relations

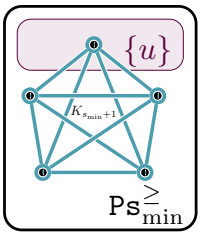
As a first major step, we show how to realize simple auxiliary relations. We then use said realizations as crucial building blocks when realizing arbitrary relations.

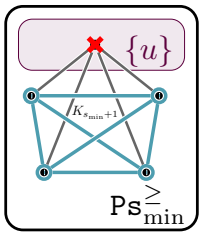
We start with a helpful gadget to ensure that a vertex v is selected (with s_{top} selected neighbors) in any partial solution, that is, how to realize the language $\{\sigma_{s_{\text{top}}}\}$.

Lemma 10.2. *Let σ and ρ denote finite and non-empty sets with $0 \notin \rho$. Then, there is a $\{\sigma_{s_{\text{top}}}\}$-realizer $(\texttt{Rs}_{\text{top}}, u)$.*[11]

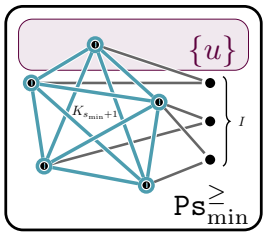
Proof. We start with a useful helper gadget.

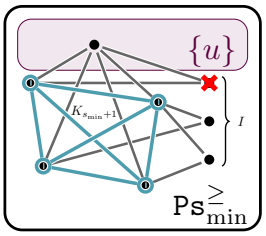
[11] It is also possible to obtain a $\{\sigma_{s_{\text{top}}}\}$-realizer for all non-empty (but not necessarily finite) ρ with (i) $0 \notin \rho$ and for some $r \in \rho$ we have $r+1 \notin \rho$ or (ii) $t_{\min} \geq 1$. However, as we need $\{\sigma_{s_{\text{top}}}\}$-realizers for finite ρ only, we instead opted for simplifying our construction.



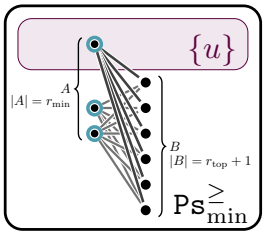


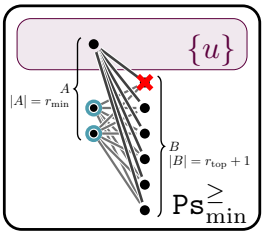
(a) The $\{\sigma_{s_{\min}}\}$-provider for Case 1, when $0 < s_{\min} < r_{\min}$. Observe that not selecting the portal vertex violates the ρ-constraints of the portal vertex.





(b) The $\{\sigma_{s_{\min}}\}$-provider for Case 2, when $s_{\min} \geq r_{\min} \geq 1$. Observe that not selecting the portal vertex violates the ρ-constraints of at least one vertex from the independent set I.





(c) The $\{\sigma_{s_{\min}}\}$-provider for Case 3, when $s_{\min} = 0, r_{\min} \geq 1$. Observe that not selecting the portal vertex violates the ρ-constraints of at least one vertex from B.

Figure 10.1: The gadget constructions from Claim 10.3.

Claim 10.3. *There is a $\{\sigma_{s_{\min}}\}$-provider $(\mathtt{Ps}^{\geq}_{\min}, u)$ with the additional property that $L(G, \{u\}) \subseteq \{\sigma_i \in \mathbb{S} \mid s_{\min} \leq i\}$.*

Proof of Claim.

Recall that we need to construct a graph $G := \mathtt{Ps}^{\geq}_{\min}$ with a single portal vertex u that satisfies $\{\sigma_{s_{\min}}\} \subseteq L(G, \{u\}) \subseteq \{\sigma_i \in \mathbb{S} \mid s_{\min} \leq i\}$. We consider three cases.

Case 1: $0 < s_{\min} < r_{\min}$. We choose G to be a clique of size $s_{\min} + 1$; we declare any of the vertices to be u; consult Figure 10.1a for a visualization.

Now, suppose that one of the vertices of G is unselected and call said vertex v. Then, v needs at least $r_{\min}$ selected neighbors. However, v can have at most $s_{\min} < r_{\min}$ selected neighbors, and thus, v cannot be unselected. Symmetrically, all other vertices of G need to be selected as well.

Case 2: $s_{\min} \geq r_{\min} \geq 1$. We construct G as follows. Starting from a clique $C = K_{s_{\min}+1}$ (of $s_{\min}+1$ vertices) and an independent set $I = I_{\lceil (s_{\min}+1)/r_{\min} \rceil}$ (of $\lceil (s_{\min}+1)/r_{\min} \rceil$ vertices), we add $r_{\min} \cdot \lceil (s_{\min}+1)/r_{\min} \rceil$ edges between C and I such that each vertex in I has a degree of exactly $r_{\min}$, and such that every vertex in C is adjacent to at least one vertex in I. It is readily verified that this is always possible. Finally, we pick any vertex from C to be the portal vertex u. Consult Figure 10.1b for a visualization.

Now, first observe that we cannot select any vertex from I in any partial solution, as by $s_{\min} \geq r_{\min}$, any selected vertex in I may never have at least $s_{\min}$ selected neighbors. However, if all vertices in I are not selected, then any vertex neighboring a vertex in I must be selected (as the degree of any vertex in I is exactly $r_{\min}$). Hence, every vertex in C must be selected, and thus, has $s_{\min}$ selected neighbors.

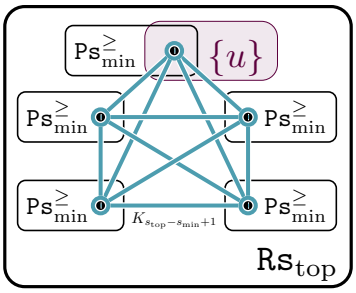


Figure 10.2: The main construction of Lemma 10.2: we use the providers from Claim 10.3 to obtain $(\mathtt{Rs}_{\text{top}}, \{u\})$. Observe that u has exactly $s_{\min} + (s_{\text{top}} - s_{\min}) = s_{\text{top}}$ selected neighbors, $s_{\min}$ selected neighbors from inside the provider and $(s_{\text{top}} - s_{\min})$ selected neighbors from the selected portals of the other instances of the provider. For providers, only their portal vertex is depicted.

Case 3: $s_{\min} = 0, r_{\min} \geq 1$. We choose G to be a complete bipartite graph with bipartition (A, B) such that $|A| = r_{\min}, |B| = r_{\text{top}} + 1$.[12] We pick an arbitrary vertex u in A as the portal vertex u. Consult Figure 10.1c for a visualization.

Suppose that u is not selected. Then, every vertex in B can have at most $r_{\min} - 1$ selected neighbors, which means that every vertex in B must be selected. In this case, u would have $r_{\text{top}} + 1$ selected neighbors, which is not feasible. Therefore, u must be selected. If u is indeed selected, then selecting every vertex in A and leaving every vertex in B unselected yields a solution for which u has $s_{\min} = 0$ selected neighbors. ◁

Finally, we use Claim 10.3 to construct a $\{\sigma_{s_{\text{top}}}\}$-realizer $(\mathtt{Rs}_{\text{top}}, \{u\})$ as follows. As our graph $G := \mathtt{Rs}_{\text{top}}$, we use $s_{\text{top}} - s_{\min} + 1 \geq 1$ independent copies $((\mathtt{Ps}^{\geq}_{\min})^{(i)}, \{u^{(i)}\})$ of the provider from Claim 10.3, and connect the vertices $u^{(i)}$ to a $(s_{\text{top}} - s_{\min} + 1)$-clique. We choose any of the vertices $u^{(i)}$ to be the portal vertex u.

By Claim 10.3, all vertices $u^{(i)}$ are selected in any partial solution for $(G, \{u\})$ with the additional guarantee that each vertex $u^{(i)}$ already has at least $s_{\min}$ selected neighbors inside of the corresponding helper gadgets. Hence, each vertex $u^{(i)}$ (and in particular u) has exactly s_{top} selected neighbors, which completes the proof. □

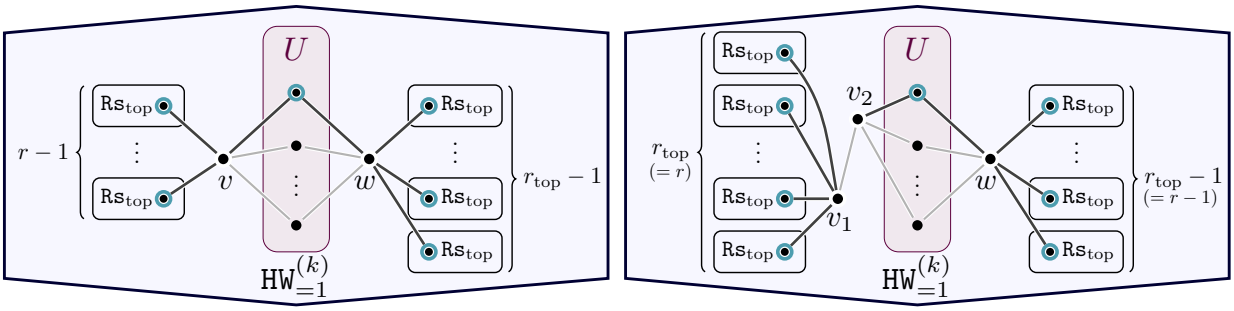
Recall from Definition 9.1 that we denote by $\mathtt{HW}^{(d)}_{=1}$ the d-ary *Hamming Weight One* relation where exactly one portal must be selected, and by $\mathtt{EQ}^{(d)}$ the d-ary *Equality* relation where either all or no portals must be selected. We first realize these two types of relations which we then use to construct arbitrary relations.

Lemma 10.4. *Let σ, ρ denote finite non-empty sets with $0 \notin \rho$.*[13] *Then, for all $k \geq 1$, $\mathtt{HW}^{(k)}_{=1}$ is realizable.*

Proof. We construct a graph with portals $H = (G, U)$ with $L(H) = \mathtt{HW}^{(k)}_{=1}$. Write $U = \{u_1, \ldots, u_k\}$ to denote the k portals of H. We distinguish three cases.

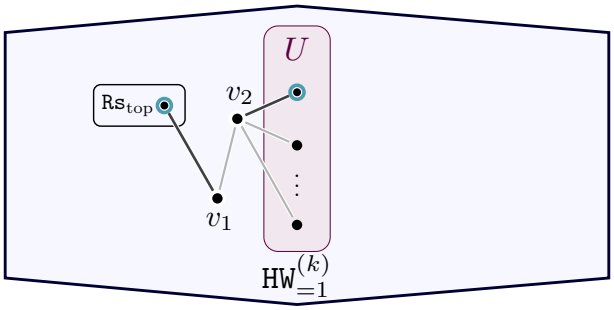
[12] Instead of r_{top}, any value $r \in \rho$ with $r + 1 \notin \rho$ suffices.

[13] Technically, the existence of a $\{\sigma_{s_{\text{top}}}\}$-realizer, as well as $1 \leq r, r' \in \rho$ with $r - 1, r' + 1 \notin \rho$, are enough to realize $\mathtt{HW}^{(k)}_{=1}$.



(a) The construction in Case 1, when there is an $r \in \rho$ with $r \geq 2$ and $r-1 \notin \rho$.

(b) The construction in Case 2, when $\rho = [1..r]$ for some $r \geq 2$.



(c) The construction in Case 3, when $\rho = \{1\}$.

Figure 10.3: The gadget constructions from Lemma 10.4 for realizing $\mathtt{HW}^{(k)}_{=1}$. For realizers, we depict only their portal vertices; we depict an arbitrary vertex from U as being selected.

Case 1: There is an $r \in \rho$ with $r \geq 2$ and $r-1 \notin \rho$. We create two additional vertices v and w, and make both adjacent to each portal u_i. Further, we take $r-1$ independent copies $((\mathtt{Rs}_{\text{top}})^{(i,1)}, u^{(i,1)})$ of the realizer from Lemma 10.2, and connect v to the $r-1$ vertices $u^{(i,1)}$. Next, we take $r_{\text{top}}-1$ independent copies $((\mathtt{Rs}_{\text{top}})^{(i,2)}, u^{(i,2)})$ of the realizer from Lemma 10.2, and connect w to the $r_{\text{top}}-1$ vertices $u^{(i,2)}$. Consult Figure 10.3a for a visualization.[14]

To see that H realizes $\mathtt{HW}^{(k)}_{=1}$, first observe that by Lemma 10.2, any vertex $u^{(\star,\star)}$ is always selected with s_{top} selected neighbors (inside of the corresponding realizer). Hence, no neighbor of any $u^{(\star,\star)}$ outside the corresponding realizer can be selected. In particular, v and w have to be unselected.

Since $r-1 \notin \rho$, at least one portal vertex $u_1, \ldots, u_k$ must be selected into the solution to satisfy the ρ-constraint of v. Further, as ρ is finite and w already has $r_{\text{top}}-1$ selected neighbors, at most one of the k portal vertices $u_1, \ldots, u_k$ can be selected. Thus, we must select exactly one portal vertex into the solution, completing the proof for this case.

Case 2: $\rho = [1..r]$ for $r \geq 2$. We create three additional vertices v_1, v_2, and w, and make v_2 and w adjacent to each portal u_i. Further, we take r independent copies $((\mathtt{Rs}_{\text{top}})^{(i,1)}, u^{(i,1)})$ of the realizer from Lemma 10.2, and connect v_1 to the r vertices $u^{(i,1)}$; we also connect v_1 and v_2. Next, we take $r_{\text{top}}-1 = r-1$ independent copies $((\mathtt{Rs}_{\text{top}})^{(i,2)}, u^{(i,2)})$ of the realizer from Lemma 10.2, and connect w to the $r-1$ vertices $u^{(i,2)}$. Consult Figure 10.3b for a visualization.

[14] If ρ is not finite, we can use the following construction: we create $\binom{k}{2}+1$ additional vertices v and w_i for all $i \subseteq [1..k]$ with $|i| = 2$. Vertex v is adjacent to all portal vertices and to $r-1$ pendent nodes where each node is the portal of an attached realizer from Lemma 10.2. For each $i \subseteq [1..k]$ with $|i| = 2$, say $i = \{i_1, i_2\}$, we make w_i adjacent to the portals u_{i_1} and u_{i_2}. Additionally, w_i is adjacent to $r'-1$ pendant nodes where each node is the portal of an attached realizer from Lemma 10.2.

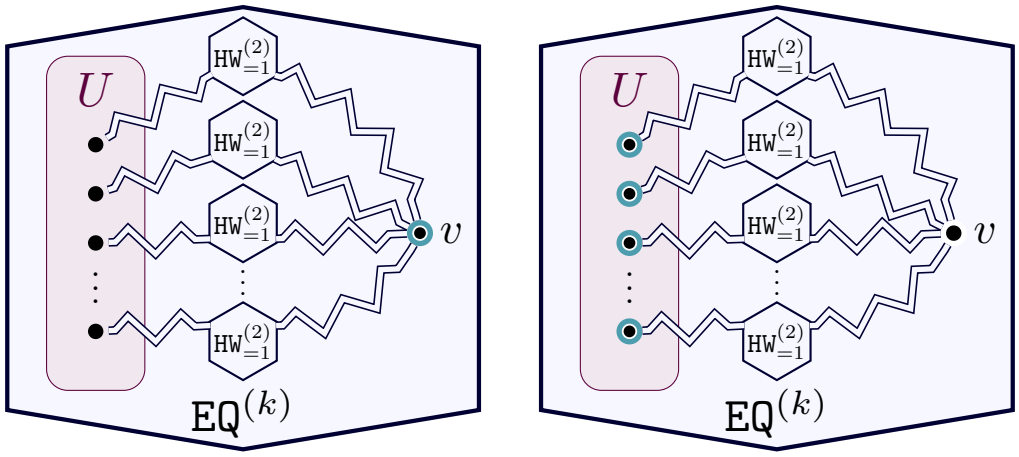


Figure 10.4: The two solutions of the gadget construction from Lemma 10.5 and Corollary 10.6 for realizing $\mathtt{EQ}^{(k)}$. For realizers, we depict only their portal vertices; we use hexagons to depict that some relation is realized between the vertices connected to said hexagons.

As in Case 1, v_1 and w must be unselected. Now, observe that v_2 must also be unselected as v_1 already has r selected neighbors and $r+1 \notin \rho$. As $0 \notin \rho$, at least one of the non-v_1 neighbors of v_2 must be selected, that is, at least one of $u_1, \ldots, u_k$ must be selected. Finally, observe that $r_{\text{top}} = r$, so as in Case 1, the vertex w ensures that at most one of the vertices $u_1, \ldots, u_k$ is selected. This completes the proof in this case.

Case 3: $\rho = \{1\}$**.** We use the same construction as in Case 2, but we remove the vertex w. Consult Figure 10.3c for a visualization.

The constraint that exactly one portal is selected follows already from the properties of v_2 as it needs exactly one selected neighbor. This completes the proof for the final case, and hence, also the proof of the lemma. □

Lemma 10.5. *Let σ, ρ denote non-empty sets. If $\mathtt{HW}^{(2)}_{=1}$ can be realized (for σ, ρ), then $\mathtt{EQ}^{(k)}$ can be realized for any $k \geq 1$ (for σ, ρ).*

Proof. First, observe that we can easily realize $\mathtt{EQ}^{(1)}$ by a graph that consists of a single portal vertex.

Now suppose that $k \geq 2$. We construct a graph with portals (H, U); write $U = \{u_1, \ldots, u_k\}$. We add a single $\{\rho_{r_{\min}}, \sigma_{s_{\min}}\}$-provider $(G, \{u\})$ from Lemma 7.1.[15] Finally, we (separately) realize $\mathtt{HW}^{(2)}_{=1}$ between u and each of $u_1, \ldots, u_k$ using some construction (which exists by assumption). Consult Figure 10.4 for a visualization.

To see that H realizes $\mathtt{EQ}^{(k)}$, first observe that if at least one portal vertex u_i is selected, then v is forced to be unselected by the $\mathtt{HW}^{(2)}_{=1}$ relations. This forces the remaining portal vertices to be selected. (Also recall that, by Definition 10.1, neither v nor the k portals u_i may have any selected neighbors in the gadgets realizing the $\mathtt{HW}^{(2)}_{=1}$ relations.)

Second, if no portal u_i is selected, then we can get a valid solution by selecting v, which completes the proof. □

[15] Here it is not important that we use $r_{\min}$ and $s_{\min}$, that is, any pair of $r \in \rho$ and $s \in \sigma$ would suffice.

Under the assumptions of Lemma 10.4, $\mathtt{HW}^{(2)}_{=1}$ can be realized, and hence, we get the following result.

Corollary 10.6. *Let σ, ρ denote finite non-empty sets with $0 \notin \rho$. Then, $\mathtt{EQ}^{(k)}$ is realizable for any $k \geq 1$.*

10.1.2 Realizing Arbitrary Relations

As promised, we proceed to realize arbitrary relations.

Lemma 10.7. *There is a fixed non-decreasing function $f : {\mathbb{Z}_{\geq 0}}^3 \to \mathbb{Z}_{\geq 0}$ that satisfies the following. Let σ, ρ denote finite non-empty sets with $0 \notin \rho$, and let $Q \subseteq 2^{[1\,..\,d]}$ denote an arbitrary relation given as a truth table. Then, Q is realizable by a graph of size at most $f(d, s_{\text{top}}, r_{\text{top}})$.*

Proof. We intend to follow the construction from [18]. However, we need to modify their construction to account for the fact that we are selecting vertices rather than edges.

Write $Q = \{q_1, \ldots, q_{|Q|}\}$, where $q_i \subseteq [1\,..\,d]$. We construct a graph with portals $H = (G, U)$ with $U = \{u_1, \ldots, u_d\}$. For each set q_i, we add $|[1\,..\,d] \backslash q_i|$ independent copies $(\mathtt{P}\{\rho_{r_{\min}}, \sigma_{s_{\min}}\}, s^{(i,j)})$ and an extra copy $(\mathtt{P}\{\rho_{r_{\min}}, \sigma_{s_{\min}}\}, t^{(i)})$ of a $\{\rho_{r_{\min}}, \sigma_{s_{\min}}\}$-provider from Lemma 7.1 to G. Next, using Corollary 10.6, we realize an $\mathtt{EQ}^{(|[1\,..\,d]\backslash q_i|+1)}$ relation between the vertices $s^{(i,\star)}$ and t^i. Using Lemma 10.4, we realize a $\mathtt{HW}^{(|Q|)}_{=1}$ relation between the vertices $t^\star$. Finally, using Lemma 10.4, for each $j \in [1\,..\,d]$, we realize a $\mathtt{HW}_{=1}$ relation between u_j and all vertices $s^{\star,j}$. Consult Figure 10.5 for a visualization of an example.

Claim 10.8. *The constructed graph H realizes Q.*

Proof of Claim. We first define a solution for a given $q_i \in Q$. We select the portal vertices u_j with $j \in q_i$. In addition, we select $t^{(i)}$ and all vertices $s^{(i,\star)}$. We do no select any other vertices $u_\star, s^{(\star,\star)}$ or $t^{(\star)}$. Observe that the $\{\rho_{r_{\min}}, \sigma_{s_{\min}}\}$-providers ensure that both selecting or not selecting a vertex $s^{(\star,\star)}$ or $t^{(\star)}$ satisfies the σ-constraints and ρ-constraints on these vertices. Hence, it remains to convinces ourselves that our selection satisfies all relation constraints on the vertices of H that we introduced.

To that end, observe that we selected exactly one vertex $t^{(\star)}$, which satisfies the $\mathtt{HW}^{(|Q|)}_{=1}$ relation between the vertices $t^{(\star)}$. Next, observe that we selected all vertices $s^{(i,\star)}$ together with $t^{(i)}$, which satisfies the $\mathtt{EQ}^{(\star)}$-relation on these vertices; for all remaining $\mathtt{EQ}^{(\star)}$ relations, all corresponding vertices are unselected (which in turn also satisfies said $\mathtt{EQ}^{(\star)}$ relations). Finally, for each $j \in [1\,..\,d]$, we select either u_j (if $j \in q_i$) or $s^{(i,j)}$ (if $j \notin q_i$); as we select no other vertices $u_\star$ or $s^{(\star,\star)}$, the remaining $\mathtt{HW}^{(\star)}_{=1}$ are also satisfied. In total, we conclude that our selection is a valid partial solution for H.

Now, assume that we are given a partial solution (and a corresponding set of selected vertices). As the relations are satisfied, there is exactly one vertex $t^{(i)}$ that is selected. By the $\mathtt{EQ}^{(\star)}$ relations, we also get that all $s^{(i,j)}$ are selected for $j \in [1\,..\,d] \setminus q_i$, and that all remaining $s^{(\star,\star)}$ are unselected. Finally, from the remaining $\mathtt{HW}_{=1}$ relations, we get that, for each $j \in [1\,..\,d] \setminus q_i$, the portal u_j is unselected (as $s^{(i,j)}$ is already selected), and that, for each $j \in q_i$, the portal u_j is selected (as all vertices $s^{(i',j)}$ are unselected in this case, as $i' \neq i$). In particular, the indices of the selected portals correspond to q_i, which completes the proof of the claim. ◁

For the bound on the size of H, observe that $|Q| \leq 2^d$, the size of a $\{\rho_{r_{\min}}, \sigma_{s_{\min}}\}$-provider is upper-bounded by a function in r_{top} and s_{top} (see Lemma 7.1), the size of a realization of $\mathtt{HW}^{(k)}_{=1}$, according to Lemma 10.4, is at most some function in r_{top}, s_{top}, k, and the size of a realization

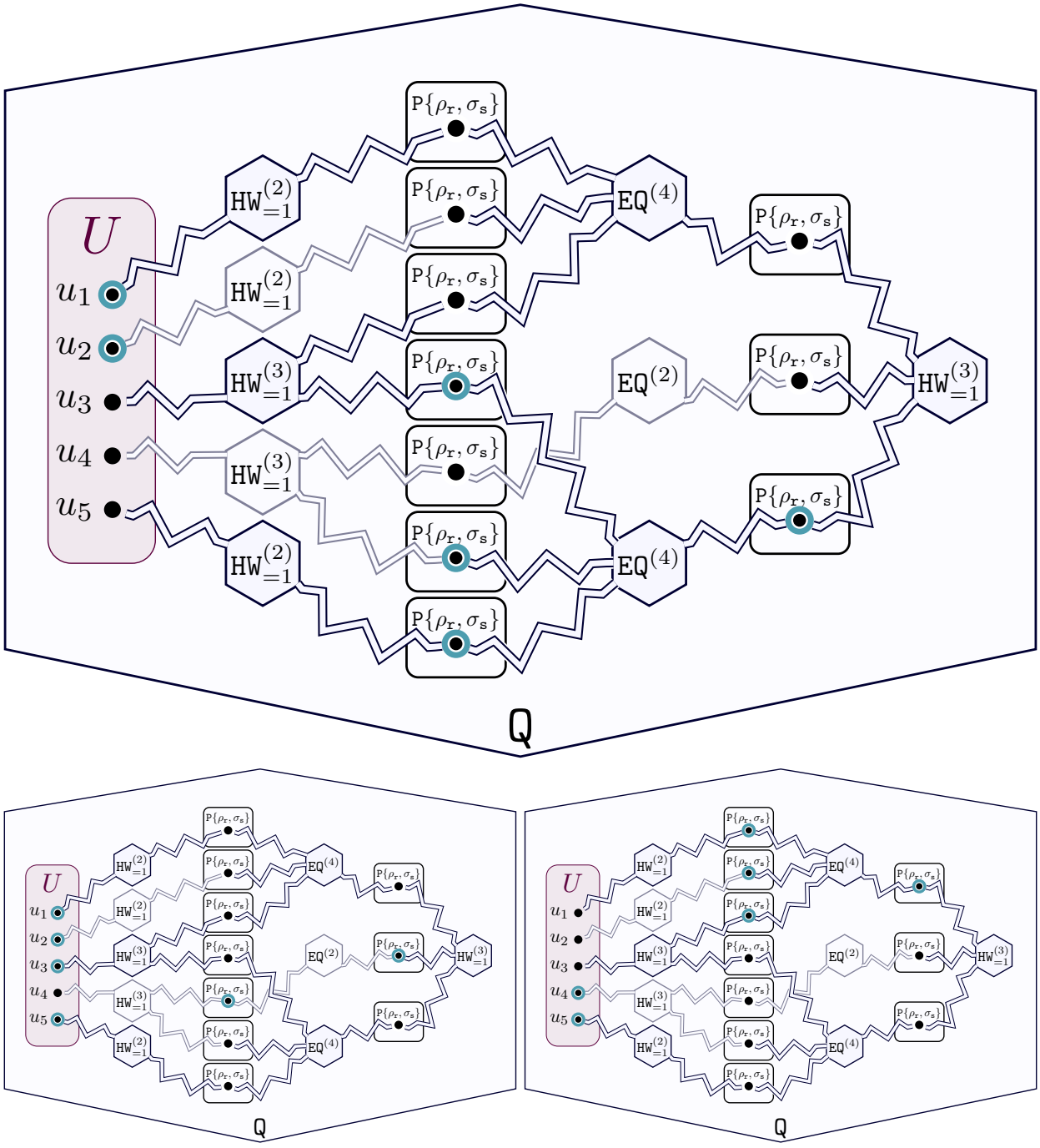


Figure 10.5: The realization of $Q = \{\{1,2\},\{1,2,3,5\},\{4,5\}\}$ and its three partial solutions. For providers, we depict only their portal vertices; also we write $\mathrm{P}\{\rho_{\mathtt{r}}, \sigma_{\mathtt{s}}\}$ for $\mathrm{P}\{\rho_{\mathtt{r}_{\min}}, \sigma_{\mathtt{s}_{\min}}\}$. Further, hexagons depict relations that are realized between vertices.

of $\mathtt{EQ}^{(k)}$ is upper-bounded by a function in k, the size of an $\mathtt{HW}^{(2)}_{=1}$-gadget, and the size of the $\{\rho_{r_{\min}}, \sigma_{s_{\min}}\}$-provider. In total, this concludes the proof. □

As the final step we have to prove Lemma 6.6.

Lemma 6.6 (Removing Relations in the Decision Version)**.** *Let σ, ρ denote finite, non-empty sets with $0 \notin \rho$. If all relations have arity at most $O(1)$, then (σ,ρ)-DomSet$^{\text{Rel}} \leq_{\mathsf{tw}} (\sigma,\rho)$-DomSet.*

Proof. Let G be an instance of (σ,ρ)-DomSet$^{\text{Rel}}$ where all relations have arity at most $O(1)$.

We use Lemma 10.7 to replace each relation by its realization. Let H be the resulting graph. A given tree decomposition of G is modified as follows. For each relation R, there is a bag X_R containing R and the neighborhood of R. We duplicate the bag X_R and add all the vertices of the realization to the copy of the bag.

As the arity of each relation is constant and we only add vertices of one realization to a bag, we get $\mathsf{tw}(H) \leq \mathsf{tw}(G) + O(1)$. Hence, the reduction is treewidth-preserving. □

10.2 Counting Problem

In this section, we show that if (σ,ρ)-#DomSet is not polynomial-time solvable, then it is sufficiently expressive to realize arbitrary relations (unless σ is cofinite with $\rho = \mathbb{Z}_{\geq 0}$). In order to formally state this result in Theorem 10.9, we first need some notation.

Let Ω_0 be the set of all finite and cofinite subsets of $\mathbb{Z}_{\geq 0}$, and let $\Omega \coloneqq \Omega_0 \setminus \{\varnothing\}$. Let $(\sigma,\rho) \in \Omega^2$. For some non-negative integer k, let $\mathcal{P}$ be a set of pairs $\{(\sigma^{(i)},\rho^{(i)}) \in \Omega_0^2 \mid i \in [1\,..\,k]\}$. For ease of notation, let $(\sigma^{(0)},\rho^{(0)}) \coloneqq (\sigma,\rho)$. We now define a generalization of (σ,ρ)-#DomSet that allows us to augment the problem with additional constraints specified by $\mathcal{P}$ in order to obtain more modeling power.

The problem (σ,ρ)-#DomSet$^{\mathcal{P}}$ takes as input some graph G together with a mapping $\lambda\colon V(G) \to [0\,..\,k]$, and outputs the number of sets $S \subseteq V(G)$ with the following properties:

- For each $v \in S$, we have $|N(v) \cap S| \in \sigma^{(\lambda(v))}$.
- For each $v \notin S$, we have $|N(v) \cap S| \in \rho^{(\lambda(v))}$.

If $\mathcal{P}$ is the empty set, then we also drop the superscript and simply write (σ,ρ)-#DomSet. In this case, it also suffices to take as input only the graph G and assume that λ is constant 0. We also refer to S as a *solution* of (σ,ρ)-#DomSet$^{\mathcal{P}}$ (on input G,λ). For a less formal but more convenient specification of λ, we say that a vertex v with $\lambda(v) = i$ is a $(\sigma^{(i)},\rho^{(i)})$-*vertex*.

Sometimes we restrict the problem (σ,ρ)-#DomSet$^{\mathcal{P}}$ to instances for which certain pairs from $\mathcal{P}$ are used only a constant number of times (w.r.t. the number of vertices in G). To this end, for some positive integer c, we say that a pair $P \in \mathcal{P}$ is *c-bounded* if we restrict (σ,ρ)-#DomSet$^{\mathcal{P}}$ to instances G,λ with $|\lambda^{-1}(P)| \leq c$.

When working with (σ,ρ)-#DomSet$^{\mathcal{P}}$, it is helpful to generalize some definitions from Section 3. For a graph G with portals U and a mapping $\lambda\colon V(G) \to [0\,..\,k]$, the tuple $\mathcal{G} = (G,U,\lambda)$ is a *gadget* for (σ,ρ)-#DomSet$^{\mathcal{P}}$. Again, we drop λ if $\mathcal{P} = \varnothing$. Then, a *partial solution* of (G,U,λ) is a set $S \subseteq V(G)$ with:

- For each $v \in S \setminus U$, we have $|N(v) \cap S| \in \sigma_{\lambda(v)}$.
- For each $v \notin S \cup U$, we have $|N(v) \cap S| \in \rho_{\lambda(v)}$.

Let $n = |V(G)|$. Recall that $\mathbb{A}_n \coloneqq \{\sigma_0, \cdots, \sigma_n, \rho_0, \cdots, \rho_n\}$. Recall that a partial solution S *witnesses* the string $x \in \mathbb{A}_n^U$ if, for each $u \in U$, we have

$$x[\,u\,] = \begin{cases} \sigma_z & \text{if } u \in S \\ \rho_z & \text{otherwise} \end{cases}$$

where $z \coloneqq |N(u) \cap S|$.

Then, for $x \in \mathbb{A}_n^U$, $\mathrm{ext}_{\mathcal{G}}(x)$ is the number of partial solutions of $\mathcal{G}$ that witness x. We also refer to this as the number of *extensions* of x (to the gadget $\mathcal{G}$).

With an eye to Fact 3.3, recall that a pair $(\sigma, \rho) \in \Omega_0^2$ is *trivial* if $\rho = \{0\}$ or $\sigma = \rho = \mathbb{Z}_{\geq 0}$. Otherwise, we say that (σ, ρ) is *non-trivial*. Sometimes the following notation is useful. For a set τ of non-negative integers (think of σ or ρ) and an integer i, let $\tau - i \coloneqq \{k - i \mid k \in \tau, k - i \geq 0\}$.

Now we can state the main result of this section. Recall the definition of (σ, ρ)-#DomSet$^{\text{Rel}}$ as defined in Section 6.

Theorem 10.9. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial. If $\rho \neq \mathbb{Z}_{\geq 0}$ or σ is finite, then (σ, ρ)-#DomSet$^{\text{Rel}}$ $\leq_{\mathsf{tw}}$ (σ, ρ)-#DomSet.*

We state some intermediate results. Recall Figure 2.4 from Section 2.2 for a more detailed overview of the steps involved in order to reduce from the problem with relations to the one without.

Lemma 10.10. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial. If $\rho = \mathbb{Z}_{\geq 0}$ and σ is finite, then (σ, ρ)-#DomSet$^{\text{Rel}}$ $\leq_{\mathsf{tw}}$ (σ, ρ)-#DomSet.*

Lemma 10.11. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial. If $\rho \neq \mathbb{Z}_{\geq 0}$, then we have (σ, ρ)-#DomSet$^{\text{Rel}}$ $\leq_{\mathsf{tw}}$ (σ, ρ)-#DomSet$^{\{(\varnothing,\rho),(\sigma,\varnothing)\}}$.*

Given a set of pairs $\mathcal{P}$ from Ω_0^2 and a pair $(\sigma', \rho') \in \Omega_0^2$, we use the shorthand $\mathcal{P} + (\sigma', \rho')$ for the set $\mathcal{P} \cup \{(\sigma', \rho')\}$.

Lemma 10.12. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial with $\rho \neq \mathbb{Z}_{\geq 0}$. Let $\mathcal{P}$ be some (possibly empty) set of pairs from Ω_0^2. Then, (σ, ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\rho)+(\sigma,\varnothing)}$ $\leq_{\mathsf{tw}}$ (σ, ρ)-#DomSet$^{\mathcal{P}}$.*

Proof of Theorem 10.9. For the case $\rho = \mathbb{Z}_{\geq 0}$, the proof of Theorem 10.9 directly follows from Lemma 10.10, which we prove in Section 10.2.2. Otherwise, it follows from Lemma 10.11 together with Lemma 10.12 (applied to an empty set of pairs $\mathcal{P}$), which we prove in Sections 10.2.1 and 10.2.3, respectively. □

10.2.1 Realizing Relations by Forcing Selection Status

The main result of this section is to prove Lemma 10.11 which shows that if we assume that we can force (σ, ρ)-vertices to be selected, and if we additionally assume that we can force such vertices to be unselected, then we can model arbitrary relations.

In this section, we use $\mathcal{P}$ to refer to a set of pairs from Ω_0^2. We first start with an auxiliary result that we use several times in the following.

Lemma 10.13. *Let $(\sigma, \rho) \in \Omega^2$ with $s \in \sigma$, $r \in \rho$, and $r \geq 1$. There is a graph G with a vertex u such that there are disjoint (σ, ρ)-sets X and Y of G such that $X \cup Y = V(G)$, $u \in X$ with $|N(u) \cap X| = s$, and $u \notin Y$ with $|N(u) \cap Y| = r$.*

Proof. Let G consist of $2r$ cliques $X_1, \dots, X_r, Y_1, \dots Y_r$, each of size $s+1$. For $i \in [1..r]$, let $x_0^{(i)}, \dots, x_s^{(i)}$ be the vertices of X_i, and let $y_0^{(i)}, \dots, y_s^{(i)}$ be the vertices of Y_i. These cliques are connected in a way that they form bicliques indexwise, that is, for each $i \in [1..r]$ and each $j \in [0..s]$, the vertex $x_j^{(i)}$ is adjacent not only to the vertices in its clique X_i, but also to the vertices $y_j^{(1)}, \dots, y_j^{(r)}$. Let $u = x_1^{(1)}$. Note that both $X = \bigcup_{i=1}^{r} X_i$ and $Y = \bigcup_{i=1}^{r} Y_i$ are (σ, ρ)-sets of G, where $u \in X$ with $|N(u) \cap X| = s$, and $u \notin Y$ with $|N(u) \cap Y| = r$. □

Note that the requirements of Lemma 10.13 are always fulfilled if (σ, ρ) is a non-trivial pair in Ω^2. In this case, both σ and ρ are non-empty, and $\rho \neq \{0\}$.

In a first step, we show how to remove arbitrary relations using only $\mathtt{HW}_{=1}$ relations. Recall that (σ, ρ)-#DomSet$^{\text{Rel}}$ allows arbitrary relations to appear. For some relation R, we write (σ, ρ)-#DomSet^{R} as the restricted problem where only the relation R might appear, that is, other relations are not allowed.

Lemma 10.14. *Let $(\sigma, \rho) \in \Omega^2$ be a non-trivial pair. Then, (σ, ρ)-#DomSet$^{\text{Rel}} \leq_{\mathsf{tw}} (\sigma, \rho)$-#DomSet$^{\mathtt{HW}_{=1}}$. If the arity of the relations is initially bounded by $O(1)$, then the reduction preserves this property.*

Proof. Recall that in Lemma 10.5, we realize $\mathtt{EQ}^{(k)}$ relations just using $\mathtt{HW}_{=1}$ relations and $\{\sigma_s, \rho_r\}$-providers for some $s \in \sigma$ and $r \in \rho$. Observe that this construction is actually parsimonious whenever the $\{\sigma_s, \rho_r\}$-providers are parsimonious. In Lemma 10.7, this result is extended to realize arbitrary relations. Once more this construction is parsimonious whenever the $\{\sigma_s, \rho_r\}$-providers are parsimonious. To use this construction we claim that there is actually such a provider.

Claim 10.15. *There are $s \in \sigma$ and $r \in \rho$ such that there is a parsimonious $\{\sigma_s, \rho_r\}$-provider that uses $\mathtt{HW}_{=1}$ relations.*

Proof of Claim. There are $s \in \sigma$ and $r \in \rho$ with $s \geq 0$ and $r \geq 1$ since both σ and ρ are non-empty, and additionally $\rho \neq \{0\}$ because of non-triviality. Then, we can use Lemma 10.13 to get a graph G with a vertex w such that there are two solutions S_0 and S_1 partitioning the vertex set of G satisfying $w \notin S_0$, $|N(w) \cap S_0| = r$, $w \in S_1$, and $|N(w) \cap S_1| = s$.

For each pair of vertices $v \in S_0$ and $u \in S_1$, we add a complex vertex with relation $\mathtt{HW}_{=1}$ that is adjacent to v and u. The $\mathtt{HW}_{=1}$ relations ensure that in each partial solution either all vertices from S_0 are selected and none from S_1, or all vertices from S_1 are selected and none from S_0. The complex vertices are unselected by definition, and hence, do not give any neighbors to other vertices. As the gadget does not contain any other vertices, it has precisely these two partial solutions. ◁

Using Claim 10.15, we can use the construction from Lemmas 10.5 and 10.7 to replace all relations by appropriate gadgets. As the arity of each relation is constant, the size of the graph replacing the relations is also constant. □

In the following, we show how to remove the $\mathtt{HW}_{=1}$ relations. By doing so, we essentially are able to remove arbitrary relations when applying Lemma 10.14 first.

We show for the following pairs in $\mathcal{P}$ that we can realize $\mathtt{HW}_{=1}$:

- $(\varnothing, \{1\})$ in Lemma 10.16,
- $(\varnothing, \{0, 1\})$ in Lemma 10.17,
- $(\varnothing, \mathbb{Z}_{\geq 1})$ in Lemma 10.19,
- $(\{0\}, \mathbb{Z}_{\geq 0})$ in Lemma 10.20,

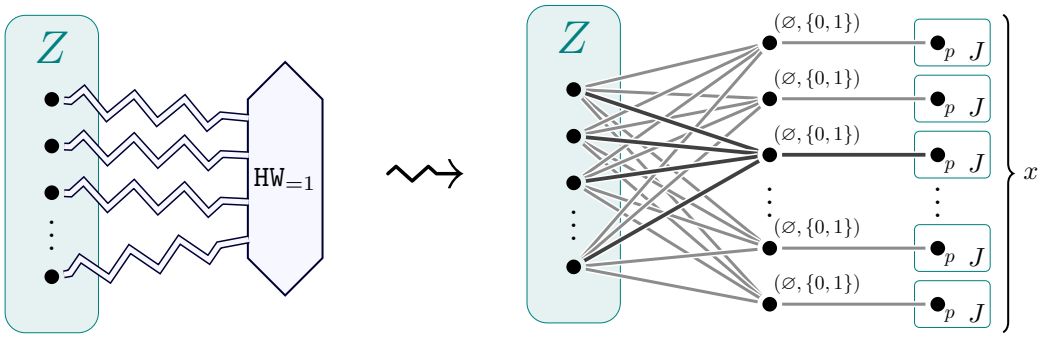


Figure 10.6: The construction of instance I'_x from Lemma 10.17 to replace a $\texttt{HW}_{=1}$ relation between the vertices in Z.

- $\{(\varnothing, \rho), (\sigma, \varnothing)\}$ in Lemma 10.11.

We start with the easiest case.

Lemma 10.16. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial. If $(\varnothing, \{1\}) \in \mathcal{P}$, then (σ, ρ)-#DomSet$^{\text{Rel}} \leq_{\text{tw}}$ (σ, ρ)-#DomSet$^{\mathcal{P}}$.*

Proof. We use a simple reduction from (σ, ρ)-#DomSet$^{\texttt{HW}_{=1}}$ and apply Lemma 10.14. From an instance I of (σ, ρ)-#DomSet$^{\texttt{HW}_{=1}}$, we create an instance I' of (σ, ρ)-#DomSet$^{\mathcal{P}}$ as follows. For each complex vertex z with relation $\texttt{HW}_{=1}$, we completely connect $N(z)$ to a new $(\varnothing, \{1\})$-vertex z'. Then, we remove z. It is straightforward to see that the solutions of I and I' are in a one-to-one-correspondence. Since the neighborhood of a complex vertex in I is considered as a clique, by the definition of the treewidth of I, the treewidth of I' is at most that of I. □

Lemma 10.17. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial. If $(\varnothing, \{0,1\}) \in \mathcal{P}$, then (σ, ρ)-#DomSet$^{\text{Rel}} \leq_{\text{tw}}$ (σ, ρ)-#DomSet$^{\mathcal{P}}$.*

Proof. We show the reduction (σ, ρ)-#DomSet$^{\texttt{HW}_{=1}} \leq_{\text{tw}}$ (σ, ρ)-#DomSet$^{\mathcal{P}}$. Then, the statement of the lemma follows from Lemma 10.14.

Let I be an instance of (σ, ρ)-#DomSet$^{\texttt{HW}_{=1}}$. Let U be the set of (complex) $\texttt{HW}_{=1}$-vertices in I, and let $\mathcal{Z} = \{N(u) \mid u \in U\}$. Let I' be the instance I without the vertices in U. I' can be cast as an instance of (σ, ρ)-#DomSet$^{\mathcal{P}}$. Then, the number of solutions of I is identical to the number of those solutions of I' that select precisely one vertex from each $Z \in \mathcal{Z}$.

By Lemma 10.13, there is a graph J that contains a vertex p such that J has $\alpha \geq 1$ (σ, ρ)-sets that contain p, and it has $\beta \geq 1$ (σ, ρ)-sets that do not contain p. It is not clear whether or not $\alpha = \beta$. For a positive integer x, let I'_x be the instance of (σ, ρ)-#DomSet$^{\mathcal{P}}$ obtained from I' by attaching to each set $Z \in \mathcal{Z}$ a total of x $(\varnothing, \{0,1\})$-vertices $v_1^Z, \ldots, v_x^Z$ each of which is completely connected to Z. In addition, there are x copies $J_1^Z, \ldots, J_x^Z$ of the graph J, where each v_i^Z is adjacent to the copy of p in J_i^Z. Consult Figure 10.6 for a visualization of this construction (for a single set Z).

Note that a solution S' of I' for (σ, ρ)-#DomSet$^{\mathcal{P}}$ can only be extended to a solution of I'_x if, in each set $Z \in \mathcal{Z}$, at most one vertex is selected (because of the attached $(\varnothing, \{0,1\})$-vertices). Let us say that solutions of I' with this property are *good*. Suppose in a good solution S', a set $Z \in \mathcal{Z}$ is entirely unselected. Then, there are $f_0 := (\alpha + \beta)^x$ feasible extensions to its attached copies of J ($\alpha + \beta$ for each graph J_i since the copy of p can be either selected or not). If in S' exactly 1 vertex

of Z is selected, then there are $f_1 \coloneqq \beta^x$ extensions to the attached copies of J — this time we can only use the selections of the J_i's for which the copy of p is unselected.

Let a_i be the number of good solutions of I' in which precisely i of the sets in $\mathcal{Z}$ are entirely unselected. (In the remaining sets from $\mathcal{Z}$ exactly one vertex is selected.) Let $\#S(I'_x)$ denote the number of solutions of I'_x. Then,

$$\begin{aligned}\#S(I'_x) &= \sum_{i=0}^{|\mathcal{Z}|} a_i \cdot f_0^i f_1^{|\mathcal{Z}|-i} \\ &= f_1^{|\mathcal{Z}|} \cdot \sum_{i=0}^{|\mathcal{Z}|} a_i \cdot \left(\frac{f_0}{f_1}\right)^i \\ &= (\alpha+\beta)^{x|\mathcal{Z}|} \cdot \sum_{i=0}^{|\mathcal{Z}|} a_i \cdot \left(\left(\frac{\alpha+\beta}{\beta}\right)^x\right)^i.\end{aligned}$$

Note that $\#S(I'_x)/(\alpha+\beta)^{x|\mathcal{Z}|}$ is a polynomial in $((\alpha+\beta)/\beta)^x$ of degree $|\mathcal{Z}|$. Since $(\alpha+\beta)/\beta > 1$, it suffices to choose $|\mathcal{Z}|$ different values of $x > 0$ for the interpolation. This way we can recover the coefficients a_i. In particular, we can recover a_0, which corresponds to the number of good solutions of I' in which none of the sets in $\mathcal{Z}$ are entirely unselected, i.e., in which all of these sets contain precisely one selected vertex. This is exactly the sought-for number of solutions of I.

Since the neighborhood of a complex vertex u in I is considered as a clique in the definition of the treewidth of I, for $Z = N(u)$, we can add all vertices from $J_1^Z, \ldots, J_x^Z$ together with the vertices $v_1^Z, \ldots, v_x^Z$ one after another to a copy of the original bag containing the clique Z. Hence, the treewidth of I_x is at most that of I plus an additive term in $O(1)$. □

We define the relation $\mathtt{HW}_{\geq 1}$ to be the *Hamming Weight at least One* relation which requires that not all portals are unselected, that is, at least one of the portals must be selected.

Lemma 10.18. *Let $(\sigma,\rho) \in \Omega^2$ be non-trivial. Then, (σ,ρ)-#DomSet$^{\mathrm{Rel}} \leq_{\mathsf{tw}} (\sigma,\rho)$-#DomSet$^{\mathtt{HW}_{\geq 1}}$.*

Proof. The proof is similar to that of Lemma 10.17, but uses a different gadget, which leads to different numbers of extensions that have to be considered. We show the reduction (σ,ρ)-#DomSet$^{\mathtt{HW}=1} \leq_{\mathsf{tw}}$ (σ,ρ)-#DomSet$^{\mathtt{HW}_{\geq 1}}$. Then, the statement of the lemma follows from Lemma 10.14.

Let I be an instance of (σ,ρ)-#DomSet$^{\mathtt{HW}=1}$. We define U, $\mathcal{Z}$, I', J, α, and β precisely as we did in the proof of Lemma 10.17. When defining I_x, we divert from the proof of Lemma 10.17. For a positive integer x, let I'_x be the instance of (σ,ρ)-#DomSet$^{\mathtt{HW}_{\geq 1}}$ obtained from I' by attaching, for each $Z \in \mathcal{Z}$, a $\mathtt{HW}_{\geq 1}$-vertex v^Z. In addition, we attach to each subset Z' of Z with $|Z'| = |Z| - 1$ a total of x $\mathtt{HW}_{\geq 1}$-vertices $v_1^{(Z,Z')}, \ldots, v_x^{(Z,Z')}$ each of which is completely connected to Z'. Again, there are also x copies $J_1^{(Z,Z')}, \ldots, J_x^{(Z,Z')}$ of the graph J, where each $v_i^{(Z,Z')}$ is adjacent to the copy of p in $J_i^{(Z,Z')}$. Note that some set Z' may receive multiple such attachments for different supersets Z.

Note that a solution S' of I' for (σ,ρ)-#DomSet$^{\mathcal{P}}$ can only be extended to a solution of I'_x if, in each set $Z \in \mathcal{Z}$, at least one vertex is selected (because of the attached $\mathtt{HW}_{\geq 1}$-vertex v^Z). Let us say that solutions of I' with this property are *good*. If, in a good solution S' of I', some Z' has at least one selected vertex, then there are $f_1 \coloneqq (\alpha+\beta)^x$ feasible extensions to the graphs $J_1^{(Z,Z')}, \ldots, J_x^{(Z,Z')}$ ($\alpha+\beta$ for each graph $J_i^{(Z,Z')}$ since the copy of p can either be selected or not). If in S' no vertex of Z' is selected, then there are only $f_0 \coloneqq (\alpha+\beta)^x - \beta^x$ extensions (all copies of p unselected is not possible). Continuing from this, we say that if a set $Z \in \mathcal{Z}$ contains at least 2

selected vertices (from S'), then it is *undesired*, and if it contains precisely 1 selected vertex, then it is *desired*.

For an undesired set $Z \in \mathcal{Z}$, there is at least one vertex selected in each of the $|Z|$ corresponding subsets Z' (this gives a total of $f_1^{|Z|}$ extensions to the graphs $J_\star^{(Z,\star)}$), whereas, for a desired set Z, there is exactly one subset Z' that is entirely unselected (which gives a total of $f_1^{(|Z|-1)} \cdot f_0$ extensions). Now, let a_i be the number of good solutions of I' in which precisely i of the sets $Z \in \mathcal{Z}$ are undesired. Let $\#S(I'_x)$ denote the number of solutions of I'_x. Then,

$$\begin{aligned}
\#S(I'_x) &= \sum_{i=0}^{|\mathcal{Z}|} a_i \cdot \Big[\prod_{\text{undesired } Z \in \mathcal{Z}} f_1^{|Z|} \cdot \prod_{\text{desired } Z \in \mathcal{Z}} f_1^{(|Z|-1)} f_0 \Big] \\
&= f_1^{\sum_{Z \in \mathcal{Z}} (|Z|-1)} \cdot \sum_{i=0}^{|\mathcal{Z}|} a_i \cdot f_1^i f_0^{|\mathcal{Z}|-i} \\
&= \underbrace{f_1^{\sum_{Z \in \mathcal{Z}} (|Z|-1)} f_0^{|\mathcal{Z}|}}_{=:F} \cdot \sum_{i=0}^{|\mathcal{Z}|} a_i \cdot \left(\frac{(\alpha+\beta)^x}{(\alpha+\beta)^x - \beta^x} \right)^i .
\end{aligned}$$

Note that $\#S(I'_x)/F$ is a polynomial in $(\alpha+\beta)^x/((\alpha+\beta)^x - \beta^x)$ of degree $|\mathcal{Z}|$. Since $\alpha, \beta \geq 1$, the values of $(\alpha+\beta)^x/((\alpha+\beta)^x - \beta^x)$ are defined and distinct for different $x > 0$. It suffices to choose $|\mathcal{Z}|$ different values of $x > 0$ for the interpolation. This way we can recover the coefficients a_i. In particular, we can recover a_0, which corresponds to the number of good solutions of I' in which all of the sets in $\mathcal{Z}$ are desired, i.e., in which all of these sets contain precisely one selected vertex. This is exactly the sought-for number of solutions of I.

As in the proof of Lemma 10.17, we can argue that the treewidth of I_x is at most that of I plus an additive term in $O(1)$. □

As a next step, we replace the $\mathtt{HW}_{\geq 1}$ relations by some appropriate vertices as we did previously for the $\mathtt{HW}_{=1}$ relations.

Lemma 10.19. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial. If $(\varnothing, \mathbb{Z}_{\geq 1}) \in \mathcal{P}$, then (σ,ρ)-$\#\textsc{DomSet}^{\textsc{Rel}} \leq_{\mathsf{tw}}$ (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$.*

Proof. This is a straightforward reduction from (σ,ρ)-$\#\textsc{DomSet}^{\mathtt{HW}_{\geq 1}}$ to (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$. The statement of the lemma then follows from Lemma 10.18.

From an instance I of (σ,ρ)-$\#\textsc{DomSet}^{\mathtt{HW}_{\geq 1}}$, we create an instance I' of (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$ by completely connecting, for each (complex) $\mathtt{HW}_{=1}$-vertex z, the neighborhood $N(z)$ to a new $(\varnothing, \mathbb{Z}_{\geq 1})$-vertex z'. Then, we remove z.

It is straightforward to see that the solutions of I and I' are in a one-to-one correspondence. Since the neighborhood of a complex vertex in I is considered as a clique in the definition of the treewidth of I, the treewidth of I' is at most that of I. □

Lemma 10.20. *If $(\{0\}, \mathbb{Z}_{\geq 0}) \in \mathcal{P}$, then (σ,ρ)-$\#\textsc{DomSet}^{\textsc{Rel}} \leq_{\mathsf{tw}}$ (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$.*

Proof. We show the reduction (σ,ρ)-$\#\textsc{DomSet}^{\mathtt{HW}_{\geq 1}} \leq_{\mathsf{tw}}$ (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$. Then, the statement of the lemma follows from Lemma 10.18.

Let I be an instance of (σ,ρ)-$\#\textsc{DomSet}^{\mathtt{HW}_{\geq 1}}$, and let V_S be the simple vertices of I. Let U be the set of (complex) $\mathtt{HW}_{\geq 1}$-vertices in G, and let $\mathcal{Z} = \{N(u) \mid u \in U\}$. Let I' be the instance I without the vertices in U. I' can be cast as an instance of (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$. Then, the number

of solutions of I is identical to the number of those solutions of I' that select at least one vertex from each $Z \in \mathcal{Z}$.

For a positive integer x, let I'_x be the instance of (σ,ρ)-#DomSet$^{\mathcal{P}}$ obtained from I' by attaching to each set $Z \in \mathcal{Z}$ a total of x $(\{0\},\mathbb{Z}_{\geq 0})$-vertices $v_1^Z,\ldots,v_x^Z$ each of which is completely connected to Z.

Let S' be some subset of V_S. Note that in order for S' to be extended to a solution of I'_x, it is required that all vertices in V_S that are *not* in some $Z \in \mathcal{Z}$ already have a feasible number of selected neighbors ((σ,ρ)-constraint) in S'. Let us say that subsets S' with this property are *good*. Suppose that in some good S' a set $Z \in \mathcal{Z}$ is entirely unselected. Then, there are 2^x feasible extensions to the attached vertices $v_1^Z,\ldots,v_x^Z$ (each of them can be selected or not). On the contrary, if at least 1 vertex of Z is selected, then, in a feasible extension of S' to a solution of I_x, all of the vertices $v_1^Z,\ldots,v_x^Z$ have to be unselected.

Let a_i be the number of good subsets of V_S in which precisely i of the sets in $\mathcal{Z}$ are entirely unselected. Let $\#S(I'_x)$ denote the number of solutions of I'_x. Then,

$$\#S(I'_x) = \sum_{i=0}^{|\mathcal{Z}|} a_i \cdot 2^{xi}.$$

Let n be the number of vertices in I. Then, for each i we have $a_i \leq 2^n$. Thus, we can choose $x \in O(n)$ sufficiently large such that $a_0 < 2^x$. Then, $a_0 = \#S(I'_x) \mod 2^x$ can be computed by a single (σ,ρ)-#DomSet$^{\mathcal{P}}$-oracle call. Now note that a_0 is the number of good subsets S' of V_S such that each set $Z \in \mathcal{Z}$ contains at least one vertex from S'. In this case, all of the attached vertices ($v_1^Z,\ldots,v_x^Z$ for each $Z \in \mathcal{Z}$) have to be unselected, which implies that all vertices in Z obtain a feasible number of selected neighbors already from S'. Therefore, every set S' that contributes to a_0 is actually a (σ,ρ)-set of I' with the additional property that each set Z in $\mathcal{Z}$ contains at least one selected vertex. This shows that a_0 is precisely the number of solutions of I.

It remains to argue that the treewidth does not change too much. For each clique Z, there is a bag B containing Z. We duplicate this bag B a total of $x \in O(n)$ times, and each vertex v_i^Z is added to exactly one of these copies. Hence, the treewidth of I_x is at most that of I plus an additive term in $O(1)$. □

Now we have everything ready to prove the following lemma, which is the main result of this section.

Lemma 10.11. *Let $(\sigma,\rho) \in \Omega^2$ be non-trivial. If $\rho \neq \mathbb{Z}_{\geq 0}$, then we have (σ,ρ)-#DomSet$^{\text{Rel}} \leq_{\mathsf{tw}}$ (σ,ρ)-#DomSet$^{\{(\varnothing,\rho),(\sigma,\varnothing)\}}$.*

Proof. First, suppose that ρ is finite. Then, r_{top} is the maximum element of ρ. Because of non-triviality, we have $\rho \neq \{0\}$, and therefore, $r_{\text{top}} \geq 1$. If $r_{\text{top}} = 1$, then ρ is one of $\{1\}$ or $\{0,1\}$, and the statement follows from Lemma 10.16 or Lemma 10.17, respectively. Otherwise, we have $r_{\text{top}} \geq 2$. Let $s_{\min}$ be the minimum of σ. In this case, notice that a $(s_{\min}+1)$-clique of $(\sigma,\varnothing)$-vertices gives a $(\sigma',\varnothing)$-vertex with $0 \in \sigma'$, and a vertex p that is subject to $(\varnothing,\rho)$ and that is adjacent to $i \geq 1$ $(\sigma',\varnothing)$-vertices models a $(\varnothing,\rho-i)$-vertex. Using this construction, we obtain, for $\mathcal{P} = \{(\varnothing,\rho),(\sigma,\varnothing)\}$ and for each $i \geq 1$, the reduction (σ,ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\rho-i)} \leq_{\mathsf{tw}} (\sigma,\rho)$-#DomSet$^{\mathcal{P}}$. In particular, we obtain (σ,ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\rho-(r_{\text{top}}-1))} \leq_{\mathsf{tw}} (\sigma,\rho)$-#DomSet$^{\mathcal{P}}$, where we use that $r_{\text{top}} \geq 2$. Now again, $\rho-(r_{\text{top}}-1)$ is one of $\{1\}$ or $\{0,1\}$, and we can conclude as before using Lemma 10.16 or Lemma 10.17, respectively.

Second, suppose that ρ is cofinite. Then, r_{top} is the largest integer missing from ρ plus 1. Note that $r_{\text{top}} \geq 1$ as $\rho \neq \mathbb{Z}_{\geq 0}$. If $r_{\text{top}} = 1$, then $\rho = \mathbb{Z}_{\geq 1}$, and the statement follows from

Lemma 10.19. Otherwise, suppose that $r_{\text{top}} \geq 2$. As in the case with finite ρ, there is a reduction (σ,ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\rho-i)} \leq_{\mathsf{tw}} (\sigma,\rho)$-#DomSet$^{\mathcal{P}}$ for each $i \geq 1$. In particular, we obtain (σ,ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\rho-(r_{\text{top}}-1))} \leq_{\mathsf{tw}} (\sigma,\rho)$-#DomSet$^{\mathcal{P}}$, where we use that $r_{\text{top}} \geq 2$. Now again, $\rho-(r_{\text{top}}-1) = \mathbb{Z}_{\geq 1}$, and we can conclude as before using Lemma 10.19. □

10.2.2 Realizing Relations if $\rho = \mathbb{Z}_{\geq 0}$

The goal of this section is to prove Lemma 10.10. We start with some intermediate results.

Lemma 10.21. *Let $(\sigma,\rho) \in \Omega^2$ be non-trivial with $\rho = \mathbb{Z}_{\geq 0}$. If σ is finite, then (σ,ρ)-#DomSet$^{\{(\sigma,\varnothing)\}} \leq_{\mathsf{tw}}$ (σ,ρ)-#DomSet.*

Proof. Let $I = (G,\lambda)$ be an instance of (σ,ρ)-#DomSet$^{(\sigma,\varnothing)}$, and let U be the set of $(\sigma,\varnothing)$-vertices of I. For a positive integer x, we define an instance I_x of (σ,ρ)-#DomSet. Let $\mathcal{J} = (J,\{p\})$ be a gadget for (σ,ρ)-#DomSet that consists of p together with x cliques, each of size $s_{\min}+1$. The portal p is adjacent to precisely one vertex from each clique. Then, we obtain I_x by making the vertices in U (σ,ρ)-vertices, and attaching to each vertex $u \in U$ a copy of $\mathcal{J}$, where u is identified with the portal of $\mathcal{J}$.

Let $\#S(I_x)$ be the number of solutions of (σ,ρ)-#DomSet for the instance I_x. Let $p(x) \coloneqq \#S(I_x) \mod 2^x$. We show that $p(x)$ is a polynomial whose degree depends on σ, and whose constant term is the number of solutions of I for which the vertices in U are selected, which is precisely what we want. We recover the constant term by interpolation.

Consider a solution S of I_x and let $S_G = S \cap V(G)$ be the corresponding selection of the original vertices. If one of the vertices in U is unselected in S_G, then consider the attached gadget $\mathcal{J}$. Note that in the solution S, each of the cliques in $\mathcal{J}$ can only be entirely selected or entirely unselected (as selecting only a few vertices would give them less than $s_{\min}$ selected neighbors). If u is unselected, each combination of entirely selected or unselected cliques is feasible since $\rho = \mathbb{Z}_{\geq 0}$. This means that the gadget $\mathcal{J}$ has 2^x feasible extensions for the selection S_G. Consequently, selections of the vertices in G for which of the vertices in U is unselected do not contribute to $p(x)$. So let$\#S'(I_x)$ be the number of solutions of I_x for which all vertices in U are selected. We have $p(x) = \#S'(I_x) \mod 2^x$.

Suppose that $U \subseteq S_G$ for some selection S_G of vertices from $V(G)$. For each vertex u in U, the number of feasible extensions to the attached gadget $\mathcal{J}$ now depends on the number of selected neighbors of u. For i selected neighbors, there are $f_i(x) = \sum_{s+i\in\sigma} \binom{x}{s+i}$ feasible extensions to the corresponding copy of $\mathcal{J}$. Note that $f_i(x)$ is a polynomial of degree at most s_{top}. Also observe that the constant of $f_i(x)$ is 1 if $i \in \sigma$, and otherwise it is 0. As $\#S'(I_x)$ is a sum of products of terms of the form $f_i(x)$ for different i, $\#S'(I_x)$ is a polynomial in x of degree at most $s_{\text{top}} \cdot |U|$. Consequently, for sufficiently large x, $p(x) = \#S'(I_x)$. This means that we can recover $p(x)$ by polynomial interpolation using at most $s_{\text{top}} \cdot |U|$ different sufficiently large values of x and corresponding oracle calls to (σ,ρ)-#DomSet on instance I_x.

Now note that a selection S_G of vertices from $V(G)$ contributes 1 to the constant term of $p(x)$ if and only for each vertex in U the corresponding number i of selected neighbors is in σ, otherwise it contributes 0 to the constant. So, it contributes 1 if and only if S_G is a solution of I in which all vertices of U are selected. The constant of $p(x)$ is the sought-for number of solutions of (σ,ρ)-#DomSet$^{(\sigma,\varnothing)}$ on instance I. □

Lemma 10.22. *Let $(\sigma,\rho) \in \Omega^2$ be non-trivial with $\rho = \mathbb{Z}_{\geq 0}$. If σ is finite with $s_{\text{top}}-1 \notin \sigma$, then (σ,ρ)-#DomSet$^{\{(\{0\},\mathbb{Z}_{\geq 0})\}} \leq_{\mathsf{tw}} (\sigma,\rho)$-#DomSet$^{\{(\sigma,\varnothing)\}}$.*

Proof. Let $I = (G, \lambda)$ be an instance of (σ, ρ)-#DomSet$^{(\{0\},\mathbb{Z}_{\geq 0})}$, and let U be the set of $(\{0\}, \mathbb{Z}_{\geq 0})$-vertices of I. We define a gadget $\mathcal{J}$ for (σ, ρ)-#DomSet$^{\{(\sigma,\varnothing)\}}$. $\mathcal{J}$ has a single portal p, which is a (σ, ρ)-vertex. The portal p is fully adjacent to an s_{top}-clique C of $(\sigma, \varnothing)$-vertices. There is an additional vertex v which is also adjacent to all vertices in C, but not to p. Note that there are precisely two partial solutions for this gadget. The vertices of C are $(\sigma, \varnothing)$-vertices, and therefore, they are always selected. If p is selected, then the vertices in C cannot have another selected neighbor, and therefore, v is unselected (which is fine as $\rho = \mathbb{Z}_{\geq 0}$). If p is unselected, then the vertices in C have $s_{\text{top}} - 1$ selected neighbors within C. Since $s_{\text{top}} - 1 \notin \sigma$, the vertex v must be selected (which is fine as it now also has s_{top} selected neighbors). In either case, p has s_{top} selected neighbors within $\mathcal{J}$.

From I we define an instance I' of (σ, ρ)-#DomSet$^{\{(\sigma,\varnothing)\}}$ by making the vertices in U (σ, ρ)-vertices, and attaching to each vertex u in U a copy of the gadget $\mathcal{J}$, where u is identified with the portal p. It is straightforward to see that the number of solutions of instance I for (σ, ρ)-#DomSet$^{(\{0\},\mathbb{Z}_{\geq 0})}$ is the same as the number of solutions of I' for (σ, ρ)-#DomSet$^{\{(\sigma,\varnothing)\}}$. □

Lemma 10.23. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial with $\rho = \mathbb{Z}_{\geq 0}$. If σ is finite with $s_{\text{top}} - 1 \in \sigma$, then* (σ, ρ)-#DomSet$^{\{(\{0\},\mathbb{Z}_{\geq 0})\}} \leq_{\mathsf{tw}} (\sigma, \rho)$-#DomSet$^{\{(\sigma,\varnothing)\}}$.

Proof. We make a case distinction depending on s_{top}. If $s_{\text{top}} = 0$, then $\sigma = \{0\}$ and we are done. If $s_{\text{top}} = 1$, then, as $s_{\text{top}} - 1 \in \sigma$, we have $\sigma = \{0, 1\}$. In this case, a (σ, ρ)-vertex with an attached $(\sigma, \varnothing)$-vertex models a $(\{0\}, \rho)$-vertex. (Recall that $\rho = \mathbb{Z}_{\geq 0}$.) So, given some instance I of (σ, ρ)-#DomSet$^{(\{0\},\mathbb{Z}_{\geq 0})}$, we replace every $(\{0\}, \mathbb{Z}_{\geq 0})$-vertex by a (σ, ρ)-vertex with an attached $(\sigma, \varnothing)$-vertex to obtain an instance I' of (σ, ρ)-#DomSet$^{\{(\sigma,\varnothing)\}}$ with the same number of solutions.

Finally, consider the case $s_{\text{top}} \geq 2$. Let $\mathcal{J}$ be a gadget for (σ, ρ)-#DomSet$^{\{(\sigma,\varnothing)\}}$ with a single portal p (a (σ, ρ)-vertex) that is adjacent to two vertices v_1, v_2 of an $s_{\text{top}} + 1$-clique C of $(\sigma, \varnothing)$-vertices from which the edge between v_1 and v_2 is removed. Note that v_1 and v_2 have degree $s_{\text{top}} - 1$ in C, whereas the remaining vertices of C have degree s_{top}. Hence, selecting all vertices of C gives a partial solution independently of the selection status of p. In either case, p obtains two selected neighbors from $\mathcal{J}$. Thus, a (σ, ρ)-vertex that acts as portal of $\lfloor s_{\text{top}}/2 \rfloor$ attached copies of $\mathcal{J}$ models a (σ', ρ)-vertex, where σ' is one of $\{0\}$ (if s_{top} is even) or $\{0, 1\}$ (if s_{top} is odd, and using the fact that $s_{\text{top}} - 1 \in \sigma$). So we obtain a reduction (σ, ρ)-#DomSet$^{\{(\sigma',\mathbb{Z}_{\geq 0}),(\sigma,\varnothing)\}} \leq_{\mathsf{tw}} (\sigma, \rho)$-#DomSet$^{\{(\sigma,\varnothing)\}}$. If $\sigma' = \{0\}$, then we are done, and if $\sigma' = \{0, 1\}$, then we can proceed as in the case for $s_{\text{top}} = 1$. □

Lemma 10.10. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial. If $\rho = \mathbb{Z}_{\geq 0}$ and σ is finite, then* (σ, ρ)-#DomSet$^{\text{Rel}} \leq_{\mathsf{tw}}$ (σ, ρ)-#DomSet.

Proof. From Lemma 10.21 together with Lemmas 10.22 and 10.23, it follows that (σ, ρ)-#DomSet$^{\{(\{0\},\mathbb{Z}_{\geq 0})\}} \leq_{\mathsf{tw}}$ (σ, ρ)-#DomSet. Then, from Lemma 10.20, it follows that (σ, ρ)-#DomSet$^{\text{Rel}} \leq_{\mathsf{tw}} (\sigma, \rho)$-#DomSet. □

10.2.3 Forcing Both Selected and Unselected Vertices for $\rho \neq \mathbb{Z}_{\geq 0}$

Let $(\sigma, \rho) \in \Omega^2$ and let $\mathcal{P}$ be some possibly empty set of pairs from Ω_0^2. Consider a gadget $\mathcal{G}$ for (σ, ρ)-#DomSet$^{\mathcal{P}}$ with a single portal. We say that $\mathcal{G}$ is a *candidate* if it has the following properties:

(I) $\text{ext}_{\mathcal{G}}(\rho_0), \text{ext}_{\mathcal{G}}(\sigma_0) \geq 1$,

(II) $\text{ext}_{\mathcal{G}}(\rho_0) \neq \text{ext}_{\mathcal{G}}(\sigma_0)$,

(III) $\mathrm{ext}_{\mathcal{G}}(\sigma_i) = \mathrm{ext}_{\mathcal{G}}(\rho_i) = 0$ for all $i \geq 2$.

We say that a candidate $\mathcal{G}$ is a *winner* if additionally it holds that

(IV) $\mathrm{ext}_{\mathcal{G}}(\rho_1) = \mathrm{ext}_{\mathcal{G}}(\sigma_1) = 0$.

We say that a candidate $\mathcal{G}$ is a *strong candidate* if (instead of Item (IV)) it additionally holds that

(iv) $\mathrm{ext}_{\mathcal{G}}(\rho_1) \geq 1$.

(v) $\mathrm{ext}_{\mathcal{G}}(\rho_0) \neq \mathrm{ext}_{\mathcal{G}}(\sigma_0) + \mathrm{ext}_{\mathcal{G}}(\sigma_1)$.

The properties of winner, candidate, and strong candidate gadgets are useful in the proof of Lemma 10.25. We start by showing that such gadgets exist.

Lemma 10.24. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial. If $\rho \neq \mathbb{Z}_{\geq 0}$, then there is a gadget $\mathcal{G} = (G, \{p\})$ for (σ, ρ)-#DomSet that is either a winner or a strong candidate.*

We defer the proof of Lemma 10.24 to Section 10.2.4.

Lemma 10.25. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial, and let $\mathcal{P}$ be some (possibly empty) set of pairs from Ω_0^2. Suppose that there is a strong candidate $\mathcal{J}$ for (σ, ρ)-#DomSet. Then, (σ, ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\{0\})} \leq_{\mathsf{tw}}$ (σ, ρ)-#DomSet$^{\mathcal{P}}$, where $(\varnothing, \{0\})$ is 1-bounded in the source problem.*

We defer the proof of Lemma 10.25 to Section 10.2.5, and first discuss how we use it to obtain Lemma 10.12.

Lemma 10.26. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial. Let $\mathcal{P}$ be some (possibly empty) set of pairs from Ω_0^2. Then, (σ, ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\rho)} \leq_{\mathsf{tw}}$ (σ, ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\{0\})}$, even if $(\varnothing, \{0\})$ is 1-bounded in the target problem.*

Proof. The construction is straightforward: it suffices for all vertices that should be unselected (that should be $(\varnothing, \rho)$-vertices) to be adjacent to a single $(\varnothing, \{0\})$-vertex p. The key observations are that p forces its neighbors to be unselected, but since p itself is unselected, it does not otherwise alter the original solutions. □

Lemma 10.12. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial with $\rho \neq \mathbb{Z}_{\geq 0}$. Let $\mathcal{P}$ be some (possibly empty) set of pairs from Ω_0^2. Then, (σ, ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\rho)+(\sigma,\varnothing)} \leq_{\mathsf{tw}}$ (σ, ρ)-#DomSet$^{\mathcal{P}}$.*

Proof. Let $\mathcal{J} = (J, \{v\})$ be the gadget given by Lemma 10.24 (using that $\rho \neq \mathbb{Z}_{\geq 0}$). Then, $\mathcal{J}$ is either a winner or a strong candidate.

We first give the reduction assuming that $\mathcal{J}$ is a winner, and consequently $\mathrm{ext}_{\mathcal{J}}(\rho_1) = \mathrm{ext}_{\mathcal{J}}(\sigma_1) = 0$. Afterward, we give a modified reduction for the case where $\mathcal{J}$ is a strong candidate.

If $\mathcal{J}$ is a winner, then the only states of $\mathcal{J}$ with non-zero extensions are ρ_0 and σ_0. In particular, v never receives any selected neighbors within $\mathcal{J}$. We start by showing that (σ, ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\rho)} \leq_{\mathsf{tw}}$ (σ, ρ)-#DomSet$^{\mathcal{P}}$. Let $I = (G, \lambda)$ be an instance of (σ, ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\rho)}$, and let $U = \{u_1, \ldots, u_k\}$ be the set of $(\varnothing, \rho)$-vertices in G. For a positive integer x, we define an instance $I_x = (G_x, \lambda_x)$ of (σ, ρ)-#DomSet$^{\mathcal{P}}$ by attaching x new copies of the gadget $\mathcal{J}$ to each vertex $u \in U$, where u is identified with the portal of each attached copy. The function λ_x is identical to λ on the vertices in $V(G) \setminus U$, and maps the remaining vertices to 0 (i.e., the vertices in U and in their attached copies of $\mathcal{J}$ are (σ, ρ)-vertices).

Let $\#S(I_x)$ denote the number of solutions of (σ, ρ)-#DomSet$^{\mathcal{P}}$ on input I_x. Further, let z_i be the number of solutions of (σ, ρ)-#DomSet$^{\mathcal{P}}$ on instance $I' = (G, \lambda_x|_{V(G)})$ for which precisely

i of the vertices in U are selected. Intuitively, the instance I' is obtained from I by replacing the $(\varnothing, \rho)$-vertices in U by (σ, ρ)-vertices. Then, our goal is to compute z_0 since this is precisely the number of solutions on instance I.

We recover z_0 from $\#S(I_x)$ using interpolation. We observe that

$$\begin{aligned}\#S(I_x) &= \sum_{i=0}^{|U|} z_i \cdot \operatorname{ext}_{\mathcal{J}}(\sigma_0)^{xi} \operatorname{ext}_{\mathcal{J}}(\rho_0)^{x(|U|-i)} \\ &= \operatorname{ext}_{\mathcal{J}}(\rho_0)^{x|U|} \cdot \sum_{i=0}^{|U|} z_i \cdot (\operatorname{ext}_{\mathcal{J}}(\sigma_0)/\operatorname{ext}_{\mathcal{J}}(\rho_0))^{xi}.\end{aligned}$$

Hence, $\#S(I_x)/\operatorname{ext}_{\mathcal{J}}(\rho_0)^{x|U|}$ is a polynomial of degree $|U|$ with coefficients $z_0, \ldots, z_{|U|}$ and indeterminates $(\operatorname{ext}_{\mathcal{J}}(\sigma_0)/\operatorname{ext}_{\mathcal{J}}(\rho_0))^x$. Since $\mathcal{J}$ is a winner, we have $\operatorname{ext}_{\mathcal{J}}(\rho_0), \operatorname{ext}_{\mathcal{J}}(\sigma_0) \geq 1$, and $\operatorname{ext}_{\mathcal{J}}(\rho_0) \neq \operatorname{ext}_{\mathcal{J}}(\sigma_0)$. Thus, we can recover the coefficients using polynomial interpolation using $|U|$ distinct values of x, which give $|U|$ distinct values of the indeterminates. In particular, we obtain z_0 as required.

Now, note that the coefficient $z_{|U|}$ corresponds to the number of solutions for which all vertices in U are selected. This is precisely what we need to model the vertices in U as $(\sigma, \varnothing)$-vertices. Hence, with the same proof we obtain the reduction (σ, ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\sigma,\varnothing)} \leq_{\mathsf{tw}} (\sigma, \rho)$-$\#\textsc{DomSet}^{\mathcal{P}}$. So applying this reduction to $\mathcal{P}' = \mathcal{P} + (\varnothing, \rho)$ and combining it with the established reduction (σ, ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\varnothing,\rho)} \leq_{\mathsf{tw}} (\sigma, \rho)$-$\#\textsc{DomSet}^{\mathcal{P}}$, we obtain the statement of the lemma.

Now, we revisit our assumption about the gadget $\mathcal{J}$ and consider the remaining case where $\mathcal{J}$ is not a winner, but a strong candidate. In this case, $\mathcal{J}$ fulfills the requirements of Lemma 10.25 with which we obtain (σ, ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\varnothing,\{0\})} \leq_{\mathsf{tw}} (\sigma, \rho)$-$\#\textsc{DomSet}^{\mathcal{P}}$, where $(\varnothing, \{0\})$ is 1-bounded in the source problem. By Lemma 10.26, we obtain (σ, ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\varnothing,\rho)} \leq_{\mathsf{tw}} (\sigma, \rho)$-$\#\textsc{DomSet}^{\mathcal{P}}$.

In order to show the sought-for (σ, ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\sigma,\varnothing)+(\varnothing,\rho)} \leq_{\mathsf{tw}} (\sigma, \rho)$-$\#\textsc{DomSet}^{\mathcal{P}}$, it now suffices to show (σ, ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\sigma,\varnothing)+(\varnothing,\rho)} \leq_{\mathsf{tw}} (\sigma, \rho)$-$\#\textsc{DomSet}^{\mathcal{P}+(\varnothing,\rho)}$. Note that if in the gadget $\mathcal{J}$ we replace all neighbors of its portal by $(\varnothing, \rho)$-vertices, then we obtain a gadget $\mathcal{J}'$ for (σ, ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\varnothing,\rho)}$ that inherits the properties $\operatorname{ext}_{\mathcal{J}'}(\rho_0), \operatorname{ext}_{\mathcal{J}'}(\sigma_0) \geq 1$, and $\operatorname{ext}_{\mathcal{J}'}(\rho_0) \neq \operatorname{ext}_{\mathcal{J}'}(\sigma_0)$ from $\mathcal{J}$. But now, since all neighbors of the portal have to be unselected, we also have $\operatorname{ext}_{\mathcal{J}'}(\rho_i) = \operatorname{ext}_{\mathcal{J}'}(\sigma_i) = 0$ for all $i \geq 1$. Thus, $\mathcal{J}'$ is now a *winner*. The same interpolation as before, but using $\mathcal{J}'$ instead of $\mathcal{J}$, yields the reduction (σ, ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\sigma,\varnothing)+(\varnothing,\rho)} \leq_{\mathsf{tw}}$ (σ, ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\varnothing,\rho)}$, which completes the proof. □

10.2.4 Proof of Lemma 10.24: Constructing Strong Candidates and Winners

Let us restate Lemma 10.24.

Lemma 10.24. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial. If $\rho \neq \mathbb{Z}_{\geq 0}$, then there is a gadget $\mathcal{G} = (G, \{p\})$ for (σ, ρ)-$\#\textsc{DomSet}$ that is either a winner or a strong candidate.*

Claim 10.27. *Let $(\sigma, \rho) \in \Omega^2$. Then, there is a gadget $\mathcal{J} = (J, \{p\})$ for (σ, ρ)-$\#\textsc{DomSet}$ such that $\operatorname{ext}_{\mathcal{J}}(\rho_0), \operatorname{ext}_{\mathcal{G}}(\rho_1) \geq 1$, but $\operatorname{ext}_{\mathcal{J}}(\rho_i) = 0$ for each $i \geq 2$.*

Proof of Claim. Let G and u be as given by Lemma 10.13. Then, J is obtained by attaching a new vertex p to the vertex u in G. Note that each partial solution of the gadget $(J, \{p\})$ in which p is unselected is a solution of (σ, ρ)-$\#\textsc{DomSet}$ on input G. Since such solutions exist independently of the selection status of u (Lemma 10.13), both states ρ_0 and ρ_1 can be extended by $(J, \{p\})$. Since p only has a single neighbor in J, there are no extensions for ρ_i if $i \geq 2$. ◁

By Claim 10.27, the gadget $\mathcal{J}$ has $f_i := \mathrm{ext}_{\mathcal{J}}(\rho_i) \geq 1$ if and only if $i \in \{0,1\}$. For a yet to be determined value x, let $(J_1, \{u_1\}), \ldots, (J_x, \{u_x\})$ be x copies of the gadget $\mathcal{J}$. From this we obtain a graph Z by identifying the vertices $u_1, \ldots, u_x$ to a single vertex u, which then is adjacent to the portal p of $\mathcal{Z} = (Z, \{p\})$.

Now, consider extensions of the states ρ_0 and σ_0 of $\mathcal{Z}$ depending on whether σ and ρ are finite or cofinite. Since in both cases u is unselected, each of the attached copies of $\mathcal{J}$ is in state ρ_0 or ρ_1. Then, we obtain a solution whenever u obtains a feasible number of selected neighbors (from p and its neighbors in the attached copies of $\mathcal{J}$).

We have

$$\mathrm{ext}_{\mathcal{Z}}(\rho_0) = \begin{cases} \sum_{r \in \rho} \binom{x}{r} f_1^r f_0^{x-r} = f_0^x \cdot \sum_{r \in \rho} \binom{x}{r} (f_1/f_0)^r & \text{if } \rho \text{ is finite.} \\ (f_0+f_1)^x - \sum_{r \notin \rho} \binom{x}{r} f_1^r f_0^{x-r} = (f_0+f_1)^x - f_0^x \cdot \sum_{r \notin \rho, r \geq 0} \binom{x}{r} (f_1/f_0)^r & \text{if } \rho \text{ is cofinite.} \end{cases}$$

and analogously

$$\mathrm{ext}_{\mathcal{Z}}(\sigma_0) = \begin{cases} f_0^x \cdot \sum_{r \in \rho, r \geq 1} \binom{x}{r-1} (f_1/f_0)^{r-1} & \text{if } \rho \text{ is finite.} \\ (f_0+f_1)^x - f_0^x \cdot \sum_{r \notin \rho, r \geq 1} \binom{x}{r-1} (f_1/f_0)^{r-1} & \text{if } \rho \text{ is cofinite.} \end{cases}$$

Claim 10.28. *There is an integer x_0 (depending on ρ) such that, for all $x \geq x_0$, $\mathcal{Z}$ is a candidate.*

Proof of Claim. First, consider the case where ρ is finite. Then, $\mathrm{ext}_{\mathcal{Z}}(\rho_0)/f_0^x$ is a polynomial in x of degree r_{top}, whereas $\mathrm{ext}_{\mathcal{Z}}(\sigma_0)/f_0^x$ is a polynomial of degree $r_{\mathrm{top}} - 1$. Since $r_{\mathrm{top}} \geq 1$ by the fact that (σ, ρ) is non-trivial, it follows that these polynomials are not constant 0. Hence, there is a value of x that only depends on ρ for which $\mathrm{ext}_{\mathcal{Z}}(\rho_0) \neq \mathrm{ext}_{\mathcal{Z}}(\sigma_0)$ and $\mathrm{ext}_{\mathcal{Z}}(\rho_0), \mathrm{ext}_{\mathcal{Z}}(\sigma_0) \geq 1$, as required.

Second, suppose that ρ is cofinite. Then, $\sum_{r \notin \rho} \binom{x}{r} (f_1/f_0)^r$ is a polynomial in x. Its degree is the largest non-negative integer r^* that is missing from ρ. Since $\rho \neq \mathbb{Z}_{\geq 0}$, such an integer exists and the polynomial is not constant 0. Similarly, $\sum_{r \notin \rho, r \geq 1} \binom{x}{r-1} (f_1/f_0)^{r-1}$ is a polynomial in x with degree $r^* - 1$. In this case, the sum might be empty, but in any case the expressions for $\mathrm{ext}_{\mathcal{Z}}(\rho_0)$ and $\mathrm{ext}_{\mathcal{Z}}(\sigma_0)$ are positive and distinct for all sufficiently large x.

Finally, since p only has a single neighbor in Z, there are no extensions in $\mathcal{Z}$ for states σ_i or ρ_i if $i \geq 2$. ◁

Now, we are in one of three cases.

Case 1: $\mathrm{ext}_{\mathcal{Z}}(\rho_1) = \mathrm{ext}_{\mathcal{Z}}(\sigma_1) = 0$.

Case 2: $\mathrm{ext}_{\mathcal{Z}}(\rho_1) \geq 1$.

Case 3: $\mathrm{ext}_{\mathcal{Z}}(\rho_1) = 0$ and $\mathrm{ext}_{\mathcal{Z}}(\sigma_1) \geq 1$.

In case 1, the candidate $\mathcal{Z}$ is a winner, the statement of Lemma 10.24 holds, and we are done. In case 2, the candidate $\mathcal{Z}$ has the property from Item (iv) and it remains to show $\mathrm{ext}_{\mathcal{Z}}(\rho_0) \neq \mathrm{ext}_{\mathcal{Z}}(\sigma_0) + \mathrm{ext}_{\mathcal{Z}}(\sigma_1)$ in order for $\mathcal{Z}$ to be a strong candidate. In case 3, $\mathcal{Z}$ is a candidate, but not a strong candidate. In this case we need another gadget.

We define a new gadget from $\mathcal{Z}$ as follows. Let $\mathcal{Z}' = (Z', \{p'\})$ be another copy of $\mathcal{Z}$. Then, we attach to the vertex u in $\mathcal{Z}$ the gadget $\mathcal{Z}'$ by identifying u and p'. This forms the new gadget $\mathcal{Z}^* = (Z^*, \{p\})$.

Claim 10.29. *Suppose that $\mathrm{ext}_{\mathcal{Z}}(\rho_1) = 0$ and $\mathrm{ext}_{\mathcal{Z}}(\sigma_1) \geq 1$. Then, $\mathcal{Z}^*$ is a candidate with $\mathrm{ext}_{\mathcal{Z}^*}(\rho_1) \geq 1$. Moreover, $\mathrm{ext}_{\mathcal{Z}^*}(\rho_0) = \mathrm{ext}_{\mathcal{Z}}(\rho_0)^2$ and $\mathrm{ext}_{\mathcal{Z}^*}(\sigma_0) = \mathrm{ext}_{\mathcal{Z}}(\sigma_0) \cdot \mathrm{ext}_{\mathcal{Z}}(\rho_0)$.*

Proof of Claim. As $\mathcal{Z}$ is a candidate with $\mathrm{ext}_{\mathcal{Z}}(\rho_1) = 0$, we have $\mathrm{ext}_{\mathcal{Z}}(\rho_i) = 0$ for $i \geq 1$. Note that $\mathrm{ext}_{\mathcal{Z}^*}(\rho_0) = \mathrm{ext}_{\mathcal{Z}}(\rho_0)^2$ because if u is unselected, then $\mathcal{Z}'$ has to be in state ρ_0 as $\mathcal{Z}'$ is a copy of $\mathcal{Z}$ and $\mathrm{ext}_{\mathcal{Z}}(\rho_i) = 0$ for $i \geq 1$. This does not affect the number of selected neighbors of u, so every partial solution of $\mathcal{Z}$ that witnesses ρ_0 can be extended by every such partial solution of the copy $\mathcal{Z}'$ to a partial solution of $\mathcal{Z}^*$ that witnesses ρ_0. Analogously, $\mathrm{ext}_{\mathcal{Z}^*}(\sigma_0) = \mathrm{ext}_{\mathcal{Z}}(\sigma_0) \cdot \mathrm{ext}_{\mathcal{Z}}(\rho_0)$.

Thus, we still have $\mathrm{ext}_{\mathcal{Z}^*}(\rho_0) \neq \mathrm{ext}_{\mathcal{Z}^*}(\sigma_0)$ and $\mathrm{ext}_{\mathcal{Z}^*}(\rho_0), \mathrm{ext}_{\mathcal{Z}^*}(\sigma_0) \geq 1$. Also, as before, p only has a single neighbor in Z^* and consequently there are no extensions for states σ_i or ρ_i if $i \geq 2$.

We now verify that $\mathrm{ext}_{\mathcal{Z}^*}(\rho_1) \geq 1$. Since $\mathrm{ext}_{\mathcal{Z}}(\sigma_1) \geq 1$ there is a partial solution S of $\mathcal{Z}$ that witnesses σ_1. Let S' be a copy of this set for the gadget $\mathcal{Z}'$. Then, $S^* \coloneqq S' \cup S \setminus \{p\}$ is a partial solution of $\mathcal{Z}^*$ that witnesses ρ_1. The key observation is that u has the same number of selected neighbors as in the partial solution S (where now instead of p it has a single selected neighbor in $\mathcal{Z}'$). ◁

In order to complete the proof of Lemma 10.24 for cases 2 and 3, we set

$$\mathcal{G} \coloneqq \begin{cases} \mathcal{Z} & \text{if } \mathrm{ext}_{\mathcal{Z}}(\rho_1) \geq 1. \\ \mathcal{Z}^* & \text{if } \mathrm{ext}_{\mathcal{Z}}(\rho_1) = 0 \text{ and } \mathrm{ext}_{\mathcal{Z}}(\sigma_1) \geq 1. \end{cases}$$

We have already established that $\mathcal{G}$ is a candidate with $\mathrm{ext}_{\mathcal{G}}(\rho_1) \geq 1$. It remains to show that $\mathrm{ext}_{\mathcal{G}}(\rho_0) \neq \mathrm{ext}_{\mathcal{G}}(\sigma_0) + \mathrm{ext}_{\mathcal{G}}(\sigma_1)$ (Item (v)) for $\mathcal{G}$ to be a strong candidate.

Claim 10.30. *If ρ is cofinite, then there is an integer x_0 (depending on ρ) such that, for all $x \geq x_0$, $\mathcal{G}$ is a strong candidate.*

Proof of Claim. It remains to show that $\mathrm{ext}_{\mathcal{G}}(\rho_0) \neq \mathrm{ext}_{\mathcal{G}}(\sigma_0) + \mathrm{ext}_{\mathcal{G}}(\sigma_1)$. For cofinite ρ, recall that

$$\mathrm{ext}_{\mathcal{Z}}(\rho_0) = (f_0 + f_1)^x - f_0^x \cdot \sum_{r \notin \rho, r \geq 0} \binom{x}{r} (f_1/f_0)^r$$

and

$$\mathrm{ext}_{\mathcal{Z}}(\sigma_0) = (f_0 + f_1)^x - f_0^x \cdot \sum_{r \notin \rho, r \geq 1} \binom{x}{r-1} (f_1/f_0)^{r-1}.$$

This shows that $\mathrm{ext}_{\mathcal{Z}}(\rho_0) < \mathrm{ext}_{\mathcal{Z}}(\sigma_0)$ (using that x is sufficiently large) and consequently $\mathrm{ext}_{\mathcal{Z}}(\rho_0) \neq \mathrm{ext}_{\mathcal{Z}}(\sigma_0) + \mathrm{ext}_{\mathcal{Z}}(\sigma_1)$. This gives $\mathrm{ext}_{\mathcal{G}}(\rho_0) \neq \mathrm{ext}_{\mathcal{G}}(\sigma_0) + \mathrm{ext}_{\mathcal{G}}(\sigma_1)$, as required, where we use Claim 10.29 if $\mathcal{G} = \mathcal{Z}^*$. ◁

Now suppose that ρ is finite and $\mathrm{ext}_{\mathcal{G}}(\rho_0) = \mathrm{ext}_{\mathcal{G}}(\sigma_0) + \mathrm{ext}_{\mathcal{G}}(\sigma_1)$, which means that $\mathcal{G}$ is not a strong candidate. In this case, we make one last modification and define $\mathcal{G}' = (G', \{q\})$ as follows.

For a yet to be determined value y, let $(G_1, \{p_1\}), \ldots, (G_y, \{p_y\})$ be y copies of the gadget $\mathcal{G}$. We use the same trick as before. We obtain a graph G' by identifying the vertices $p_1, \ldots, p_y$ to a single vertex p, which then is adjacent to the portal q of $\mathcal{G}' = (G', \{q\})$. In order to complete the proof of Lemma 10.24, we show the following claim.

Claim 10.31. *If ρ is finite and $\mathrm{ext}_{\mathcal{G}}(\rho_0) = \mathrm{ext}_{\mathcal{G}}(\sigma_0) + \mathrm{ext}_{\mathcal{G}}(\sigma_1)$, then, for sufficiently large y, $\mathcal{G}'$ is a strong candidate.*

Proof of Claim. Using that $\mathrm{ext}_{\mathcal{G}}(\rho_0), \mathrm{ext}_{\mathcal{G}}(\rho_1) \geq 1$, the proof that $\mathcal{G}'$ is a candidate for large enough y is analogous to the proof that $\mathcal{Z}$ is a candidate for sufficiently large x.

Since $\mathcal{G}$ is a candidate, we have $\mathrm{ext}_{\mathcal{G}}(\sigma_0) \geq 1$ and $\mathrm{ext}_{\mathcal{G}}(\rho_0) \neq \mathrm{ext}_{\mathcal{G}}(\sigma_0)$. From the fact that $\mathrm{ext}_{\mathcal{G}}(\rho_0) = \mathrm{ext}_{\mathcal{G}}(\sigma_0) + \mathrm{ext}_{\mathcal{G}}(\sigma_1)$, it then follows that $\mathrm{ext}_{\mathcal{G}}(\sigma_1) \geq 1$. To shorten notation, let $g_0 \coloneqq$

$\mathrm{ext}_{\mathcal{G}}(\rho_0)$, $g_1 \coloneqq \mathrm{ext}_{\mathcal{G}}(\rho_1)$, $h_0 \coloneqq \mathrm{ext}_{\mathcal{G}}(\sigma_0)$, and $h_1 \coloneqq \mathrm{ext}_{\mathcal{G}}(\sigma_1)$. From the properties of $\mathcal{G}$, we have $g_0, g_1, h_0, h_1 \geq 1$, $g_0 \neq h_0$, and $g_0 = h_0 + h_1$.

Then, as before, using the fact that ρ is finite, we have

$$\mathrm{ext}_{\mathcal{G}'}(\rho_0) = g_0^y \cdot \sum_{r \in \rho} \binom{y}{r} (g_1/g_0)^r \quad \text{and} \quad \mathrm{ext}_{\mathcal{G}'}(\sigma_0) = g_0^y \cdot \sum_{r \in \rho, r \geq 1} \binom{y}{r-1} (g_1/g_0)^{r-1}.$$

If σ is finite, then analogously

$$\mathrm{ext}_{\mathcal{G}'}(\rho_1) = h_0^y \cdot \sum_{s \in \sigma} \binom{y}{s} (h_1/h_0)^s \quad \text{and}$$
$$\mathrm{ext}_{\mathcal{G}'}(\sigma_1) = h_0^y \cdot \sum_{s \in \sigma, s \geq 1} \binom{y}{s-1} (g_1/g_0)^{s-1}.$$

So $\mathrm{ext}_{\mathcal{G}'}(\rho_1) \geq 1$ as $h_0, h_1 \geq 1$, and the corresponding sum is not empty ($\mathcal{G}'$ has property Item (iv)). Note that $\mathrm{ext}_{\mathcal{G}'}(\rho_0) = g_0^y \cdot p_1(y)$, $\mathrm{ext}_{\mathcal{G}'}(\sigma_0) = g_0^y \cdot p_2(y)$, and $\mathrm{ext}_{\mathcal{G}'}(\sigma_1) = h_0^y \cdot p_3(y)$ for some polynomials p_1, p_2, p_3 in y, where $p_1(y) > p_2(y)$ (for y sufficiently large) since $\rho \neq \{0\}$ as (σ, ρ) is non-trivial. Thus, using the fact that $g_0 = h_0 + h_1$ with $h_1 \geq 1$ and consequently $g_0 > h_0$, for sufficiently large y we have

$$\frac{\mathrm{ext}_{\mathcal{G}'}(\sigma_0) + \mathrm{ext}_{\mathcal{G}'}(\sigma_1)}{\mathrm{ext}_{\mathcal{G}'}(\rho_0)} = \frac{g_0^y \cdot p_2(y) + h_0^y \cdot p_3(y)}{g_0^y \cdot p_1(y)} = \frac{p_2(y)}{p_1(y)} + \frac{h_0^y \cdot p_3(y)}{g_0^y \cdot p_1(y)} < 1.$$

This shows that $\mathcal{G}'$ is a strong candidate if σ is finite.

It remains to consider the case where σ is cofinite. The expressions for $\mathrm{ext}_{\mathcal{G}'}(\rho_0)$ and $\mathrm{ext}_{\mathcal{G}'}(\sigma_0)$ are the same as in the previous case, but

$$\mathrm{ext}_{\mathcal{G}'}(\rho_1) = (h_0+h_1)^y - h_0^y \cdot \sum_{s \notin \sigma} \binom{y}{s} (h_1/h_0)^s \quad \text{and} \quad \mathrm{ext}_{\mathcal{G}'}(\sigma_1) = (h_0+h_1)^y - h_0^y \cdot \sum_{s \notin \sigma, s \geq 1} \binom{y}{s-1} (g_1/g_0)^{s-1}.$$

Clearly, $\mathrm{ext}_{\mathcal{G}'}(\rho_1) \geq 1$ (since y is sufficiently large). Moreover, $\mathrm{ext}_{\mathcal{G}'}(\sigma_1) < (h_0 + h_1)^y = g_0^y$. Hence, for sufficiently large y,

$$\frac{\mathrm{ext}_{\mathcal{G}'}(\sigma_0) + \mathrm{ext}_{\mathcal{G}'}(\sigma_1)}{\mathrm{ext}_{\mathcal{G}'}(\rho_0)} \leq \frac{g_0^y \cdot p_2(y) + g_0^y}{g_0^y \cdot p_1(y)} = \frac{p_2(y) + 1}{p_1(y)} < 1.$$

This shows that $\mathcal{G}'$ is a strong candidate if σ is cofinite. ◁

10.2.5 Proof of Lemma 10.25: Forcing a Single Unselected Vertex with no Selected Neighbors

Let us restate Lemma 10.25 for convenience.

Lemma 10.25. *Let $(\sigma, \rho) \in \Omega^2$ be non-trivial, and let $\mathcal{P}$ be some (possibly empty) set of pairs from Ω_0^2. Suppose that there is a strong candidate $\mathcal{J}$ for (σ, ρ)-#DomSet. Then, (σ, ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\{0\})}$ $\leq_{\mathsf{tw}}$ (σ, ρ)-#DomSet$^{\mathcal{P}}$, where $(\varnothing, \{0\})$ is 1-bounded in the source problem.*

Let $I = (G, \lambda)$ be an instance of (σ, ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\{0\})}$ and let $n = |V(G)|$. If G contains no $(\varnothing, \{0\})$-vertex, then we can directly call the (σ, ρ)-#DomSet$^{\mathcal{P}}$-oracle on I. So suppose that p is a single $(\varnothing, \{0\})$-vertex in G. In this proof, the usual definitions of r_{top} and s_{top} are not convenient.

Instead, we use r^* as the maximum element of ρ if ρ is finite, and the maximum missing integer from ρ if ρ is cofinite. Analogously, we define s^*.

For a positive integer x, we define an instance $I_x = (G_x, \lambda_x)$ of (σ,ρ)-#DomSet$^{\mathcal{P}}$. Intuitively, in I_x the vertex p is replaced by a (σ,ρ)-vertex to which we attach x copies of the strong candidate $\mathcal{J}$, where p acts as portal for each of them. Note that $\mathcal{J}$ as a gadget for (σ,ρ)-#DomSet can also be cast as a gadget for (σ,ρ)-#DomSet$^{\mathcal{P}}$ that happens to use only (σ,ρ)-vertices.

More formally, for each $i \in [\,1\,..\,x\,]$, let $(J_i, \{v_i\})$ be a copy of $\mathcal{J}$. The graph G_x is obtained from G by identifying all vertices in $\{p, v_1, \ldots, v_x\}$ to a single vertex. Then, λ_x is identical to λ on the vertices in $V(G) \setminus \{p\}$, and maps the remaining vertices to 0 (i.e., the vertices in $\bigcup_{i=1}^{x} V(J_i)$ are (σ,ρ)-vertices).

Note that $\mathcal{G} = (G, \{p\}, \lambda_x \mid_{V(G)})$ can be interpreted as a gadget for (σ,ρ)-#DomSet$^{\mathcal{P}}$. Then, our goal is to compute $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ as this corresponds to the number of partial solutions of $\mathcal{G}$ for which p is unselected and has no selected neighbors. These are precisely the solutions of (σ,ρ)-#DomSet$^{\mathcal{P}+(\varnothing,\{0\})}$ on input I.

Let $J^{(x)}$ be the graph induced by the union of the x copies of $\mathcal{J}$ that are attached to p in G_x. Then, $\mathcal{J}^{(x)} := (J^{(x)}, \{p\})$ can also be interpreted as a gadget for (σ,ρ)-#DomSet$^{\mathcal{P}}$. (Here we dropped the λ-term since λ_x is constant 0 on $J^{(x)}$.)

For $r, s \in [\,0\,..\,x\,]$, we have

$$\begin{aligned}\mathrm{ext}_{\mathcal{J}^{(x)}}(\rho_r) &= \binom{x}{r} \mathrm{ext}_{\mathcal{J}}(\rho_1)^r \, \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r} \\ &= \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*} \cdot \underbrace{\binom{x}{r} \mathrm{ext}_{\mathcal{J}}(\rho_1)^r \, \mathrm{ext}_{\mathcal{J}}(\rho_0)^{r^*-r}}_{=:a_r(x)} \end{aligned} \tag{10.1}$$

and analogously

$$\begin{aligned}\mathrm{ext}_{\mathcal{J}^{(x)}}(\sigma_s) &= \binom{x}{s} \mathrm{ext}_{\mathcal{J}}(\sigma_1)^s \, \mathrm{ext}_{\mathcal{J}}(\sigma_0)^{x-s} \\ &= \mathrm{ext}_{\mathcal{J}}(\sigma_0)^{x-s^*} \cdot \underbrace{\binom{x}{s} \mathrm{ext}_{\mathcal{J}}(\sigma_1)^s \, \mathrm{ext}_{\mathcal{J}}(\sigma_0)^{s^*-s}}_{=:b_s(x)} \end{aligned} \tag{10.2}$$

We later use the fact that $a_r(x)$ is a polynomial of degree r since $\mathrm{ext}_{\mathcal{J}}(\rho_0)$ and $\mathrm{ext}_{\mathcal{J}}(\rho_0)$ are positive integers by our assumptions about the gadget $\mathcal{J}$. Moreover, if $x \geq r$ and $r^* \geq r$, then $a_r(x)$ is a positive integer. Note that an analogous statement for $b_s(x)$ does not necessarily hold as we may have $\mathrm{ext}_{\mathcal{J}}(\sigma_1) = 0$.

Let k be the number of neighbors of p in G. Let S' be some set of selected vertices in $V(G) \setminus \{p\}$ such that p already has $i \in [\,0\,..\,k\,]$ selected neighbors in S'. We define $f_i(x)$ as the number of possible partial solutions S'' of $\mathcal{J}^{(x)}$ that extend S' to a solution of (σ,ρ)-#DomSet$^{\mathcal{P}}$ on input I_x in which p is *unselected* — $g_i(x)$ is the corresponding number of partial solutions in which p is selected.

Now, depending on whether or not σ and ρ are finite or cofinite, we obtain different expressions for $f_i(x)$ and $g_i(x)$. We assume that $x \geq \max\{s^*, r^*\}$. Setting $\alpha := \mathrm{ext}_{\mathcal{J}}(\rho_1) + \mathrm{ext}_{\mathcal{J}}(\rho_0)$ and using that

$$\sum_{r=0}^{x} \mathrm{ext}_{\mathcal{J}^{(x)}}(\rho_r) = (\mathrm{ext}_{\mathcal{J}}(\rho_1) + \mathrm{ext}_{\mathcal{J}}(\rho_0))^x = \alpha^x,$$

we obtain

$$f_i(x) = \begin{cases} \sum_{i+r\in\rho} \mathrm{ext}_{\mathcal{J}^{(x)}}(\rho_r) & \text{if } \rho \text{ is finite} \\ \alpha^x - \sum_{i+r\notin\rho} \mathrm{ext}_{\mathcal{J}^{(x)}}(\rho_r) & \text{if } \rho \text{ is cofinite.} \end{cases} \tag{10.3}$$

Analogously, for $\beta \coloneqq \mathrm{ext}_{\mathcal{J}}(\sigma_1) + \mathrm{ext}_{\mathcal{J}}(\sigma_0)$, we have

$$g_i(x) = \begin{cases} \sum_{i+s\in\sigma} \mathrm{ext}_{\mathcal{J}^{(x)}}(\sigma_s) & \text{if } \sigma \text{ is finite} \\ \beta^x - \sum_{i+s\notin\sigma} \mathrm{ext}_{\mathcal{J}^{(x)}}(\sigma_s) & \text{if } \sigma \text{ is cofinite.} \end{cases} \tag{10.4}$$

For $\#S(I_x)$ denoting the number of solutions of (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$ on input I_x, we have

$$\#S(I_x) = \underbrace{\sum_{i=0}^{k} \mathrm{ext}_{\mathcal{G}}(\rho_i)\cdot f_i(x)}_{=:A_x} + \underbrace{\sum_{i=0}^{k} \mathrm{ext}_{\mathcal{G}}(\sigma_i)\cdot g_i(x)}_{=:B_x}. \tag{10.5}$$

Plugging Equations (10.1) and (10.2) into the expression for A_x we obtain

$$A_x = \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*} \cdot \underbrace{\sum_{i=0}^{k} \mathrm{ext}_{\mathcal{G}}(\rho_i)\cdot\Big[\sum_{i+r\in\rho} a_r(x)\Big]}_{a(x)} \quad \text{if } \rho \text{ is finite,} \tag{10.6}$$

and

$$A_x = \alpha^x \cdot \sum_{i=0}^{k} \mathrm{ext}_{\mathcal{G}}(\rho_i) - \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*} \cdot \underbrace{\sum_{i=0}^{k} \mathrm{ext}_{\mathcal{G}}(\rho_i)\cdot\Big[\sum_{i+r\notin\rho} a_r(x)\Big]}_{a(x)} \quad \text{if } \rho \text{ is cofinite.} \tag{10.7}$$

In Equation (10.6) together with Equation (10.7), we define an expression $a(x)$ depending on whether ρ is finite or cofinite. We later use the crucial fact that $a(x)$ is a polynomial in x because, for each r, $a_r(x)$ is a polynomial in x.

Analogously, we obtain

$$B_x = \mathrm{ext}_{\mathcal{J}}(\sigma_0)^{x-s^*} \cdot \underbrace{\sum_{i=0}^{k} \mathrm{ext}_{\mathcal{G}}(\sigma_i)\cdot\Big[\sum_{i+s\in\sigma} b_s(x)\Big]}_{b(x)} \quad \text{if } \sigma \text{ is finite} \tag{10.8}$$

and

$$B_x = \beta^x \cdot \sum_{i=0}^{k} \mathrm{ext}_{\mathcal{G}}(\sigma_i) - \mathrm{ext}_{\mathcal{J}}(\sigma_0)^{x-s^*} \cdot \underbrace{\sum_{i=0}^{k} \mathrm{ext}_{\mathcal{G}}(\sigma_i)\cdot\Big[\sum_{i+s\notin\sigma} b_s(x)\Big]}_{b(x)} \quad \text{if } \sigma \text{ is cofinite.} \tag{10.9}$$

Again, $b(x)$ is a polynomial in x.

At this point, let us recall that our goal is to compute $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ which equals the number of solutions of (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\varnothing,\{0\})}$ on input I. We aim to use polynomial interpolation by using A_x to obtain a polynomial in x with $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ as a coefficient. Given the different expressions for A_x depending on whether σ and ρ are finite or cofinite, we split the proof into two cases at this point. We start with the substantially easier case where both sets are finite.

Recovering $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ if both σ and ρ are finite

In order to do polynomial interpolation on A_x, we first show how to isolate the term A_x from the value of $\#S(I_x)$, which we can compute by using an oracle call.

Claim 10.32. *For sufficiently large $x \in O(n)$ and given $\#S(I_x)$, the term A_x can be computed (in time polynomial in n).*

Proof of Claim. From the definition of a candidate, we know that $\mathrm{ext}_{\mathcal{J}}(\sigma_0)$ and $\mathrm{ext}_{\mathcal{J}}(\rho_0)$ are distinct positive values. We show how to recover A_x from $\#S(I_x)$ if $\mathrm{ext}_{\mathcal{J}}(\rho_0) > \mathrm{ext}_{\mathcal{J}}(\sigma_0)$. In the case $\mathrm{ext}_{\mathcal{J}}(\rho_0) < \mathrm{ext}_{\mathcal{J}}(\sigma_0)$, the term B_x can be recovered analogously, and then $A_x = \#S(I_x) - B_x$.

Plugging Equations (10.6) and (10.8) into Equation (10.5), we obtain

$$\#S(I_x) = \underbrace{\mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*} \cdot \sum_{i=0}^{k} \mathrm{ext}_{\mathcal{G}}(\rho_i) \cdot \Big[\sum_{i+r \in \rho} a_r(x) \Big]}_{=A_x} + \underbrace{\mathrm{ext}_{\mathcal{J}}(\sigma_0)^{x-s^*} \cdot \sum_{i=0}^{k} \mathrm{ext}_{\mathcal{G}}(\sigma_i) \cdot \Big[\sum_{i+s \in \sigma} b_s(x) \Big]}_{=B_x}$$

Note that the terms $b_s(x)$ are polynomials in x, and that each of the terms $\mathrm{ext}_{\mathcal{G}}(\sigma_i)$ is upper bounded by 2^n. Thus, we can choose $x \in O(n)$ such that $B_x / \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*} < 1$ (using the fact that $\mathrm{ext}_{\mathcal{J}}(\rho_0) > \mathrm{ext}_{\mathcal{J}}(\sigma_0)$). Then, we observe that $A_x / \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*}$ is an integer.

Therefore, $\lfloor \#S(I_x) / \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*} \rfloor = A_x / \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*}$, and thus, A_x can be recovered by computing $A_x = \lfloor \#S(I_x) / \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*} \rfloor \cdot \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*}$. ◁

Recall that our ultimate goal is to compute $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ using calls to a (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$-oracle. With Claim 10.32 in hand, we can compute $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ using standard interpolation. To this end, note that $p(x) \coloneqq A_x / \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*}$ is a polynomial in x whose highest degree term is $\mathrm{ext}_{\mathcal{G}}(\rho_0)\,\mathrm{ext}_{\mathcal{J}}(\rho_1)^{r^*} \cdot x^{r^*}$. Using polynomial interpolation, we can recover the coefficients of $p(x)$ by evaluating the polynomial for r^* distinct values of x. This can be done since, according to Claim 10.32, we can compute A_x using a (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$-oracle call as long as x is sufficiently large in $O(n)$; and we can compute $\mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*}$ as $\mathrm{ext}_{\mathcal{J}}(\rho_0)$ does not depend on n. This can be done in time polynomial in $x \in O(n)$, i.e., in time polynomial in n. Using the fact that $\mathrm{ext}_{\mathcal{J}}(\rho_1)$ is non-zero by assumption of the lemma, we can recover $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ from the coefficient of the highest degree term x^{r^*}. Summarizing, we have shown that we can solve (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\varnothing,\{0\})}$ on instance I using r^* oracle calls to (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$ on instances of the form I_x.

Recovering $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ if one of σ or ρ is cofinite

In the cofinite case, there is an additional obstacle to computing $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ from A_x by polynomial interpolation. If ρ is cofinite, then A_x contains the unwanted exponential expression α^x, see Equation (10.7). We eliminate the leading exponential terms by considering

$$\psi(x) \coloneqq \#S(I_{x+2}) - (\alpha+\beta)\#S(I_{x+1}) + \alpha\beta\#S(I_x) = \underbrace{A_{x+2} - (\alpha+\beta)A_{x+1} + \alpha\beta A_x}_{=:A'_x} + \underbrace{B_{x+2} - (\alpha+\beta)B_{x+1} + \alpha\beta B_x}_{=:B'_x}. \tag{10.10}$$

So we aim to recover $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ from A'_x rather than from A_x directly. The following claim establishes that we can isolate the term A'_x from the value of $\psi(x)$, which we can compute using oracle calls.

Claim 10.33. *For sufficiently large $x \in O(n)$ and given $\#S(I_{x+2})$, $\#S(I_{x+1})$, and $\#S(I_x)$, the term A'_x can be computed (in time polynomial in n).*

We postpone the proof of Claim 10.33 for now, and first show how to compute $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ from A'_x. To shorten notation, let us set

$$a'(x) \coloneqq \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x+2-r^*} \cdot a(x+2) - (\alpha+\beta)\,\mathrm{ext}_{\mathcal{J}}(\rho_0)^{x+1-r^*} \cdot a(x+1) + \alpha\beta\,\mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*} \cdot a(x). \quad (10.11)$$

We use Equations (10.6) and (10.7) to expand the expression A'_x. If ρ is finite, then

$$A'_x = a'(x). \quad (10.12)$$

If ρ is cofinite, then

$$\begin{aligned} A'_x &= \left(\alpha^{x+2} - (\alpha+\beta)\alpha^{x+1} + \alpha\beta\alpha^x\right) \cdot \left(\sum_{i=0}^{k} \mathrm{ext}_{\mathcal{G}}(\rho_i)\right) - a'(x) \\ &= -a'(x). \end{aligned} \quad (10.13)$$

As noted previously, the terms $a(x)$ are polynomials in x. Therefore, it is straightforward to verify that $p(x) \coloneqq A'_x / \mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*}$ is also a polynomial in x (whether ρ is finite or cofinite).

Recall that our ultimate goal is to compute $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ using calls to a (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$-oracle. With Claim 10.33 in hand, we can compute $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ using standard interpolation.

Claim 10.34. $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ *can be computed from the coefficients of $p(x)$.*

Proof of Claim 10.34. We are interested in the highest degree monomial of $p(x)$. Let us first investigate the polynomial $a(x)$. Recall that $a_r(x)$ is a polynomial of degree r. Using this fact, we observe the highest degree monomial of $a(x)$ is $\mathrm{ext}_{\mathcal{G}}(\rho_0)\,\mathrm{ext}_{\mathcal{J}}(\rho_1)^{r^*} \cdot x^{r^*}$, where r^* has a different meaning depending on whether ρ is finite or cofinite.

Therefore, using Equation (10.11), the highest degree monomial of $p(x)$ is the same as the highest-degree monomial of

$$\mathrm{ext}_{\mathcal{G}}(\rho_0)\,\mathrm{ext}_{\mathcal{J}}(\rho_1)^{r^*} \cdot \left[\mathrm{ext}_{\mathcal{J}}(\rho_0)^2 (x+2)^{r^*} - (\alpha+\beta)\,\mathrm{ext}_{\mathcal{J}}(\rho_0)(x+1)^{r^*} + \alpha\beta x^{r^*}\right],$$

which is

$$\mathrm{ext}_{\mathcal{G}}(\rho_0)\,\mathrm{ext}_{\mathcal{J}}(\rho_1)^{r^*} \cdot \left[\mathrm{ext}_{\mathcal{J}}(\rho_0)^2 - (\alpha+\beta)\,\mathrm{ext}_{\mathcal{J}}(\rho_0) + \alpha\beta\right] \cdot x^{r^*}.$$

So the coefficient of x^{r^*} in $p(x)$ is $c = \mathrm{ext}_{\mathcal{G}}(\rho_0)\,\mathrm{ext}_{\mathcal{J}}(\rho_1)^{r^*} \cdot c'$, where $c' = (\mathrm{ext}_{\mathcal{J}}(\rho_0) - \alpha)(\mathrm{ext}_{\mathcal{J}}(\rho_0) - \beta)$. Recall that $\mathcal{J}$ is a strong candidate, and therefore, we have

- $\mathrm{ext}_{\mathcal{J}}(\rho_1) \geq 1$, and consequently
- $\mathrm{ext}_{\mathcal{J}}(\rho_0) \neq \mathrm{ext}_{\mathcal{J}}(\rho_0) + \mathrm{ext}_{\mathcal{J}}(\rho_1) = \alpha$, and also
- $\mathrm{ext}_{\mathcal{J}}(\rho_0) \neq \mathrm{ext}_{\mathcal{J}}(\rho_0) + \mathrm{ext}_{\mathcal{J}}(\rho_1) = \alpha$ by Item (v).

This shows that $c' \neq 0$ and $\mathrm{ext}_{\mathcal{J}}(\rho_1)^{r^*} \neq 0$, and consequently, by computing the coefficient c, the sought-for value $\mathrm{ext}_{\mathcal{G}}(\rho_0)$ can be computed as well. This finishes the proof of Claim 10.34. ◁

Finally, the coefficients of $p(x)$ can be computed by polynomial interpolation by evaluating $p(x)$ for r^* distinct values of x. This can be done since, according to Claim 10.33, we can compute A'_x using three (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$-oracle calls as long as x is sufficiently large in $O(n)$; and we can efficiently compute $\mathrm{ext}_{\mathcal{J}}(\rho_0)^{x-r^*}$ as $\mathrm{ext}_{\mathcal{J}}(\rho_0)$ does not depend on n. So computing the coefficients

can be done in time polynomial in $x \in O(n)$, and from the coefficients one can compute $\operatorname{ext}_{\mathcal{G}}(\rho_0)$ according to Claim 10.34.

Summarizing, we have shown that we can solve (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}+(\varnothing,\{0\})}$ on instance I using $3r^*$ (three per A'_x) oracle calls to (σ,ρ)-$\#\textsc{DomSet}^{\mathcal{P}}$ on instances of the form I_x.

We finish the proof of Lemma 10.25 by paying our debts and showing Claim 10.33.

Proof of Claim 10.33. By the fact that $\mathcal{J}$ is a candidate, we know that $\operatorname{ext}_{\mathcal{J}}(\sigma_0)$ and $\operatorname{ext}_{\mathcal{J}}(\rho_0)$ are distinct positive values. We show how to recover A'_x from $\psi(x)$ if $\operatorname{ext}_{\mathcal{J}}(\rho_0) > \operatorname{ext}_{\mathcal{J}}(\sigma_0)$. In the case $\operatorname{ext}_{\mathcal{J}}(\rho_0) < \operatorname{ext}_{\mathcal{J}}(\sigma_0)$, the term B'_x can be recovered analogously, and then $A'_x = \psi(x) - B'_x$.

Recall that $A'(x) = \pm a'(x)$, where the sign depends on whether ρ is finite or cofinite. If, analogously to Equation (10.11), we define

$$b'(x) \coloneqq \operatorname{ext}_{\mathcal{J}}(\sigma_0)^{x+2-s^*} \cdot b(x+2) - (\alpha+\beta)\operatorname{ext}_{\mathcal{J}}(\sigma_0)^{x+1-r^*} \cdot b(x+1) + \alpha\beta \operatorname{ext}_{\mathcal{J}}(\sigma_0)^{x-r^*} \cdot b(x),$$

then again $B'(x) = \pm b'(x)$, where the sign depends on whether σ is finite or cofinite. Then, we recall that $b(x)$ is a polynomial in x and that $\operatorname{ext}_{\mathcal{G}}(\sigma_0)$, which contributes to the coefficients of $b(x)$, is upper bounded by 2^n. Thus, we can choose $x \in O(n)$ such that $B'_x / \operatorname{ext}_{\mathcal{J}}(\rho_0)^{x-r^*} < 1$, where we use the crucial fact that $\operatorname{ext}_{\mathcal{J}}(\rho_0) > \operatorname{ext}_{\mathcal{J}}(\sigma_0)$. Then, we observe that $A'_x / \operatorname{ext}_{\mathcal{J}}(\rho_0)^{x-r^*}$ is an integer as $a_r(x)$ is an integer for $r \leq r^*$, and consequently, $a(x)$ is an integer.

Therefore, $\lfloor \psi(x) / \operatorname{ext}_{\mathcal{J}}(\rho_0)^{x-r^*} \rfloor = A'_x / \operatorname{ext}_{\mathcal{J}}(\rho_0)^{x-r^*}$, and A'_x can be recovered by computing $A'_x = \lfloor \psi(x) / \operatorname{ext}_{\mathcal{J}}(\rho_0)^{x-r^*} \rfloor \cdot \operatorname{ext}_{\mathcal{J}}(\rho_0)^{x-r^*}$. ◁

References

[1] Jochen Alber and Rolf Niedermeier. Improved tree decomposition based algorithms for domination-like problems. In *Proceedings of the 5th Latin American Symposium on Theoretical Informatics (LATIN 2002)*, volume 2286 of *Lecture Notes in Computer Science*, pages 613–628. Springer, 2002.

[2] Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 522–539. SIAM, 2021. doi: 10.1137/1.9781611976465.32. URL https://doi.org/10.1137/1.9781611976465.32.

[3] Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for np-hard problems restricted to partial k-trees. *Discret. Appl. Math.*, 23(1):11–24, 1989. doi: 10.1016/0166-218X(89)90031-0. URL https://doi.org/10.1016/0166-218X(89)90031-0.

[4] Sanjeev Arora, Michelangelo Grigni, David R. Karger, Philip N. Klein, and Andrzej Woloszyn. A polynomial-time approximation scheme for weighted planar graph TSP. In Howard J. Karloff, editor, *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, 25-27 January 1998, San Francisco, California, USA*, pages 33–41. ACM/SIAM, 1998. URL http://dl.acm.org/citation.cfm?id=314613.314632.

[5] Michael A. Bennett, Greg Martin, Kevin O'Bryant, and Andrew Rechnitzer. Explicit bounds for primes in arithmetic progressions. *Illinois J. Math.*, 62(1-4):427–532, 2018. ISSN 0019-2082. doi: 10.1215/ijm/1552442669. URL https://doi.org/10.1215/ijm/1552442669.

[6] Marshall W. Bern, Eugene L. Lawler, and A.L. Wong. Linear-time computation of optimal subgraphs of decomposable graphs. *Journal of Algorithms*, 8(2):216–235, 1987. ISSN 0196-6774. doi: https://doi.org/10.1016/0196-6774(87)90039-3. URL https://www.sciencedirect.com/science/article/pii/0196677487900393.

[7] Umberto Bertelè and Francesco Brioschi. On non-serial dynamic programming. *J. Comb. Theory, Ser. A*, 14(2):137–148, 1973. doi: 10.1016/0097-3165(73)90016-2. URL https://doi.org/10.1016/0097-3165(73)90016-2.

[8] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Fourier meets Möbius: fast subset convolution. In *Proceedings of the 39th Annual ACM on Symposium on Theory of Computing (STOC 2007)*, pages 67–74, 2007.

[9] Hans L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In Timo Lepistö and Arto Salomaa, editors, *Automata, Languages and Programming, 15th International Colloquium, ICALP88, Tampere, Finland, July 11-15, 1988, Proceedings*, volume 317 of *Lecture Notes in Computer Science*, pages 105–118. Springer, 1988. doi: 10.1007/3-540-19488-6_110. URL https://doi.org/10.1007/3-540-19488-6_110.

[10] Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Inf. Comput.*, 243:86–111, 2015. doi: 10.1016/j.ic.2014.12.008. URL https://doi.org/10.1016/j.ic.2014.12.008.

[11] Glencora Borradaile and Hung Le. Optimal dynamic program for r-domination problems over tree decompositions. In Jiong Guo and Danny Hermelin, editors, *11th International Symposium on Parameterized and Exact Computation, IPEC 2016, August 24-26, 2016, Aarhus, Denmark*, volume 63 of *LIPIcs*, pages 8:1–8:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi: 10.4230/LIPIcs.IPEC.2016.8. URL https://doi.org/10.4230/LIPIcs.IPEC.2016.8.

[12] Glencora Borradaile, Philip N. Klein, and Claire Mathieu. An $O(n \log n)$ approximation scheme for steiner tree in planar graphs. *ACM Trans. Algorithms*, 5(3):31:1–31:31, 2009. doi: 10.1145/1541885.1541892. URL https://doi.org/10.1145/1541885.1541892.

[13] Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Boolean-width of graphs. *Theor. Comput. Sci.*, 412(39):5187–5204, 2011. doi: 10.1016/j.tcs.2011.05.022. URL https://doi.org/10.1016/j.tcs.2011.05.022.

[14] Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Fast dynamic programming for locally checkable vertex subset and vertex partitioning problems. *Theor. Comput. Sci.*, 511:66–76, 2013. doi: 10.1016/j.tcs.2013.01.009. URL https://doi.org/10.1016/j.tcs.2013.01.009.

[15] Mathieu Chapelle. Parameterized complexity of generalized domination problems on bounded tree-width graphs. *CoRR*, abs/1004.2642, 2010. URL http://arxiv.org/abs/1004.2642.

[16] Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. ISSN 0890-5401. doi: https://doi.org/10.1016/0890-5401(90)90043-H. URL https://www.sciencedirect.com/science/article/pii/089054019090043H.

[17] Radu Curticapean. Block interpolation: A framework for tight exponential-time counting complexity. *Inf. Comput.*, 261:265–280, 2018. doi: 10.1016/j.ic.2018.02.008. URL https://doi.org/10.1016/j.ic.2018.02.008.

[18] Radu Curticapean and Dániel Marx. Tight conditional lower bounds for counting perfect matchings on graphs of bounded treewidth, cliquewidth, and genus. In Robert Krauthgamer, editor, *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1650–1669. SIAM, 2016. doi: 10.1137/1.9781611974331.ch113. URL https://doi.org/10.1137/1.9781611974331.ch113.

[19] Radu Curticapean, Nathan Lindzey, and Jesper Nederlof. A tight lower bound for counting Hamiltonian cycles via matrix rank. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1080–1099. SIAM, 2018. doi: 10.1137/1.9781611975031.70. URL https://doi.org/10.1137/1.9781611975031.70.

[20] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. ISBN 978-3-319-21274-6. doi: 10.1007/978-3-319-21275-3. URL https://doi.org/10.1007/978-3-319-21275-3.

[21] Erik D. Demaine and Mohammad Taghi Hajiaghayi. The bidimensionality theory and its algorithmic applications. *Comput. J.*, 51(3):292–302, 2008.

[22] Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and -minor-free graphs. *J. ACM*, 52(6):866–893, 2005.

[23] László Egri, Dániel Marx, and Paweł Rzążewski. Finding list homomorphisms from bounded-treewidth graphs to reflexive graphs: a complete complexity characterization. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, volume 96 of *LIPIcs*, pages 27:1–27:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi: 10.4230/LIPIcs.STACS.2018.27. URL https://doi.org/10.4230/LIPIcs.STACS.2018.27.

[24] Jacob Focke, Dániel Marx, and Paweł Rzążewski. Counting list homomorphisms from graphs of bounded treewidth: tight complexity bounds. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 431–458. SIAM, 2022. doi: 10.1137/1.9781611977073.22. URL https://doi.org/10.1137/1.9781611977073.22.

[25] Fedor V. Fomin, Petr A. Golovach, Jan Kratochvíl, Dieter Kratsch, and Mathieu Liedloff. Sort and search: Exact algorithms for generalized domination. *Inf. Process. Lett.*, 109(14):795–798, 2009. doi: 10.1016/j.ipl.2009.03.023. URL https://doi.org/10.1016/j.ipl.2009.03.023.

[26] Fedor V. Fomin, Petr A. Golovach, Jan Kratochvíl, Dieter Kratsch, and Mathieu Liedloff. Branch and recharge: Exact algorithms for generalized domination. *Algorithmica*, 61(2):252–273, 2011. doi: 10.1007/s00453-010-9418-9. URL https://doi.org/10.1007/s00453-010-9418-9.

[27] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. doi: 10.1145/2886094. URL https://doi.org/10.1145/2886094.

[28] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Representative families of product families. *ACM Trans. Algorithms*, 13(3):36:1–36:29, 2017. doi: 10.1145/3039243. URL https://doi.org/10.1145/3039243.

[29] Petr A. Golovach, Jan Kratochvíl, and Ondrej Suchý. Parameterized complexity of generalized domination problems. *Discret. Appl. Math.*, 160(6):780–792, 2012. doi: 10.1016/j.dam.2010.11.012. URL https://doi.org/10.1016/j.dam.2010.11.012.

[30] Rudolf Halin. S-functions for graphs. *Journal of Geometry*, 8(1-2):171–186, 1976.

[31] Magnús M. Halldórsson, Jan Kratochvíl, and Jan Arne Telle. Independent sets with domination constraints. *Discret. Appl. Math.*, 99(1-3):39–54, 2000. doi: 10.1016/S0166-218X(99)00124-9. URL https://doi.org/10.1016/S0166-218X(99)00124-9.

[32] Lars Jaffke, O joung Kwon, Torstein J. F. Strømme, and Jan Arne Telle. Generalized Distance Domination Problems and Their Complexity on Graphs of Bounded mim-width. In Christophe Paul and Michal Pilipczuk, editors, *13th International Symposium on Parameterized and Exact Computation (IPEC 2018)*, volume 115 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-084-2. doi: 10.4230/LIPIcs.IPEC.2018.6. URL http://drops.dagstuhl.de/opus/volltexte/2019/10207.

[33] Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structural parameters, tight bounds, and approximation for (k,r)-center. *Discret. Appl. Math.*, 264:90–117, 2019. doi: 10.1016/j.dam.2018.11.002. URL https://doi.org/10.1016/j.dam.2018.11.002.

[34] Philip N. Klein. A linear-time approximation scheme for planar weighted TSP. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 647–657. IEEE Computer Society, 2005. doi: 10.1109/SFCS.2005.7. URL https://doi.org/10.1109/SFCS.2005.7.

[35] Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *J. ACM*, 67(3):16:1–16:50, 2020. doi: 10.1145/3390887. URL https://doi.org/10.1145/3390887.

[36] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018. doi: 10.1145/3170442. URL https://doi.org/10.1145/3170442.

[37] Dániel Marx, Govind S. Sankar, and Philipp Schepper. Degrees and gaps: Tight complexity results of general factor problems parameterized by treewidth and cutwidth. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 95:1–95:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPIcs.ICALP.2021.95. URL https://doi.org/10.4230/LIPIcs.ICALP.2021.95.

[38] Dániel Marx, Govind S. Sankar, and Philipp Schepper. Anti-factor is FPT parameterized by treewidth and list size (but counting is hard). *CoRR*, abs/2110.09369, 2021. URL https://arxiv.org/abs/2110.09369.

[39] Dániel Marx. A parameterized view on matroid optimization problems. *Theoretical Computer Science*, 410(44):4471–4479, 2009. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2009.07.027. URL https://www.sciencedirect.com/science/article/pii/S030439750900512X. Automata, Languages and Programming (ICALP 2006).

[40] Burkhard Monien. How to find long paths efficiently. In *Analysis and design of algorithms for combinatorial problems (Udine, 1982)*, volume 109 of *North-Holland Math. Stud.*, pages 239–254. North-Holland, Amsterdam, 1985.

[41] Karolina Okrasa and Paweł Rzążewski. Fine-grained complexity of the graph homomorphism problem for bounded-treewidth graphs. *SIAM J. Comput.*, 50(2):487–508, 2021. doi: 10.1137/20M1320146. URL https://doi.org/10.1137/20M1320146.

[42] Karolina Okrasa, Marta Piecyk, and Paweł Rzążewski. Full complexity classification of the list homomorphism problem for bounded-treewidth graphs. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 74:1–74:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi: 10.4230/LIPIcs.ESA.2020.74. URL https://doi.org/10.4230/LIPIcs.ESA.2020.74.

[43] Jürgen Plehn and Bernd Voigt. Finding minimally weighted subgraphs. In Rolf H. Möhring, editor, *Graph-Theoretic Concepts in Computer Science, 16rd International Workshop, WG '90, Berlin, Germany, June 20-22, 1990, Proceedings*, volume 484 of *Lecture Notes in Computer Science*, pages 18–29. Springer, 1990. doi: 10.1007/3-540-53832-1_28. URL https://doi.org/10.1007/3-540-53832-1_28.

[44] Neil Robertson and Paul D. Seymour. Graph minors. III. planar tree-width. *J. Comb. Theory, Ser. B*, 36(1): 49–64, 1984. doi: 10.1016/0095-8956(84)90013-3. URL https://doi.org/10.1016/0095-8956(84)90013-3.

[45] Jan Arne Telle. Complexity of domination-type problems in graphs. *Nordic J. of Computing*, 1(1):157–171, mar 1994. ISSN 1236-6064.

[46] Jan Arne Telle and Andrzej Proskurowski. Practical algorithms on partial k-trees with an application to domination-like problems. In *Proceedings of the 3rd Workshopon Algorithms and Data Structures (WADS 1993)*, volume 709 of *Lecture Notes in Computer Science*, pages 610–621. Springer, 1993.

[47] Jan Arne Telle and Andrzej Proskurowski. Algorithms for vertex partitioning problems on partial k-trees. *SIAM J. Discret. Math.*, 10(4):529–550, 1997. doi: 10.1137/S0895480194275825. URL https://doi.org/10.1137/S0895480194275825.

[48] Johan M. M. van Rooij. Fast algorithms for join operations on tree decompositions. In Fedor V. Fomin, Stefan Kratsch, and Erik Jan van Leeuwen, editors, *Treewidth, Kernels, and Algorithms - Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, volume 12160 of *Lecture Notes in Computer Science*, pages 262–297. Springer, 2020. doi: 10.1007/978-3-030-42071-0_18. URL https://doi.org/10.1007/978-3-030-42071-0_18.

[49] Johan M. M. van Rooij. A generic convolution algorithm for join operations on tree decompositions. In Rahul Santhanam and Daniil Musatov, editors, *Computer Science - Theory and Applications - 16th International Computer Science Symposium in Russia, CSR 2021, Sochi, Russia, June 28 - July 2, 2021, Proceedings*, volume 12730 of *Lecture Notes in Computer Science*, pages 435–459. Springer, 2021. doi: 10.1007/978-3-030-79416-3_27. URL https://doi.org/10.1007/978-3-030-79416-3_27.

[50] Johan M. M. van Rooij, Hans L. Bodlaender, and Peter Rossmanith. Dynamic programming on tree decompositions using generalised fast subset convolution. In *Proceedings of the 17th Annual European Symposium on Algorithms (ESA 2009)*, volume 5757 of *Lecture Notes in Computer Science*, pages 566–577. Springer, 2009.