Lower Bounds for Conjunctive and Disjunctive Turing Kernels

Elisabet Burjons
Department of Computer Science, RWTH Aachen, Germany

Peter Rossmanith
Department of Computer Science, RWTH Aachen, Germany

Abstract

The non-existence of polynomial kernels for OR- and AND-compositional problems is now a well-established result. Some of these problems have adaptive or non-adaptive polynomial Turing kernels. Up to now, most known polynomial Turing kernels are non-adaptive and most of them are of the conjunctive or disjunctive kind. For some problems it has been conjectured that the existence of polynomial Turing kernels is unlikely. For instance, either all or none of the WK[1]-complete problems have polynomial Turing kernels. While it has been conjectured that they do not, a proof tying their non-existence to some complexity theoretic assumption is still missing and seems to be beyond the reach of today's standard techniques.

In this paper, we prove that OR-compositional problems and all WK[1]-hard problems do not have conjunctive polynomial kernels, a special type of non-adaptive Turing kernels, under the assumption that coNP $\not\subseteq$ NP/poly. Similarly, it is unlikely that AND-compositional problems have disjunctive polynomial kernels. Moreover, we present a way to prove that the parameterized versions of some $\oplus P$-hard problems, for instance, Odd Path on planar graphs, do not have conjunctive or disjunctive polynomial kernels, unless coNP $\subseteq$ NP/poly. Finally, we identify a problem that is unlikely to have either a conjunctive or disjunctive polynomial kernel, unless coNP $\subseteq$ NP/poly, due to a reduction from an NP-hard problem instead: Weighted Odd Path on planar graphs.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Parameterized Complexity, Turing kernels

Digital Object Identifier 10.4230/LIPIcs.IPEC.2021.12

Funding *Elisabet Burjons*: This work was supported by the Swiss National Science Foundation postdoc mobility grant P2EZP2-191861.

1 Introduction

Kernelization [12] is a central notion in parameterized complexity. Given a parameterized problem, a kernelization procedure reduces an instance of size n and parameter k to a size that only depends on the parameter in polynomial time. A *polynomial kernel* is the result of a kernelization procedure where one can find a single instance equivalent to the original instance but of size polynomial in k.

However, there are problems that do not admit polynomial kernels under common complexity theoretic assumptions. In particular, Fortnow and Santhanam [13] together with Bodlaender et al. [2] proved that, so called, OR-compositional problems do not have polynomal kernels, unless NP $\subseteq$ coNP/poly, which means that the polynomial hierarchy collapses to the third level [25]. Polynomial parameter transformations (PPT) also allowed to further extend the class of problems unlikely to have polynomial kernels [4]. Some of these problems are, for instance, Longest Path, Clique(VC), Set Cover($|U|$), and Connected Vertex Cover. If nothing is otherwise indicated, these problems are parameterized by the targeted solution size k. Here (VC) indicates that the problem is parameterized by the size of the vertex cover, and $|U|$ indicates that the problem is parameterized by the universe size. The exact definitions of all problems in this paper can be found in [12], unless otherwise stated.

Later, Drucker [11] extended these results by proving that AND-compositional problems do not have polynomial kernels, unless there are statistical zero knowledge proofs for all the problems in NP, which is a stronger condition than NP $\subseteq$ coNP/poly. Thus, the list of problems that do not have polynomial kernels also includes Treewidth, Independent Set(ω), and 3-coloring(ω), where ω indicates parameterization by size of the treewidth. Moreover, Drucker extended the previous results to include the unlikely existence of probabilistic polynomial kernels, for both OR and AND-compositional problems, with a small enough error probability.

Guo was the first to ask the question whether one could extend the notion of efficient kernelization to include polynomial Turing kernelizations [5], where instead of building one reduction, an algorithm builds many reductions from the original problem into instances of smaller size (only depending polynomially on the parameter), then the algorithm has access to an oracle solving such instances, and outputs an answer for the original instance in polynomial time. The first example of such a kernel, by Raible et al. [1], is the Leaf Out-Branching problem, in which, given a directed graph G and an integer k, one needs to find whether G contains a directed tree with at least k leaves. If the root of the tree is fixed, this problem admits a kernel of size $O(k^3)$, thus, by fixing all possible n vertices one can build the kernel for each possible root and use the oracle to find out whether at least one of these roots extends to a tree with at least k vertices, if none of them do, the graph does not contain such a tree, and otherwise the answer is yes. We call such a Turing kernel formed as the disjunction of n (or more) different instances a disjunctive Turing kernel, or a disjunctive kernel for short. Analogously one can find in the literature other examples of disjunctive kernels for OR-compositional problems such as Clique with bounded degree, Clique(VC), etc. One can analogously find conjunctive Turing kernelizations, or conjunctive kernelizations, for AND-compositional problems in some cases, however, not all such problems seem to admit one of these kernelizations. The first instance of an adaptive Turing kernelization that is neither conjunctive nor disjunctive is for Longest Path in planar graphs by Jansen [15]. In this paper, unless otherwise specified, all mentions of conjunctive, disjunctive, or Turing kernelizations refer to polynomial conjunctive, disjunctive, or Turing kernelizations.

Hermelin et al. [14] propose a completeness theory for Turing kernelization. Using the class of problems WK[1], characterized among others by Set Cover($|U|$), the authors conjecture that problems that are hard for this class do not admit Turing kernelizations, however, the authors do not provide complexity-theoretic consequences for the non-existence of Turing kernels for those problems.

On the other hand, Witteveen, Bottesch, and Torenvliet [24] show that the kernelization hierachy is strict by using diagonalization arguments to show that there exist problems that have Turing kernels but do not have non-adaptive Turing kernels such as disjunctive or conjunctive kernels, and problems that have non-adaptive Turing kernels but do not admit polynomial kernels. However, the presented problems to prove the strictness of the hierarchy are not natural.

It is worth mentioning that there are problems for which polynomial Turing kernels are unlikely due to lower bounds on their FPT running time. Any problem with a polynomial Turing kernel, whose non-parameterized version is in EXPTIME, will have a single exponential FPT algorithm consisting of running the algorithm for the polynomial Turing kernel and solving any calls to the oracle with the non-parameterized algorithm. For instance, Cygan, Pilipczuk and Pilipczuk [7] proved that the Edge Clique Cover problem has no $2^{2^{o(k)}}\mathrm{poly}(n)$ algorithm unless the ETH hypothesis fails. Thus, it is equally unlikely that there exist subexponential Turing kernels for Edge Clique Cover.

In this paper, we extend the proofs of nonexistence of polynomial kernels to the nonexistence of conjunctive kernels in the case of OR-compositional problems and disjunctive kernels in the case of AND-compositional problems under the same complexity theoretic assumptions also in the probabilistic case. We prove, thus, that WK[1]-hard problems do not admit conjunctive kernels, unless NP $\subseteq$ coNP/poly. Using these results, we consider XOR-compositional problems, which naturally correspond to problems counting parity, and prove that if a reduction from an NP-hard problem is possible these types of problems do not have conjunctive or disjunctive kernels, unless coNP $\subseteq$ NP/poly. We also prove that if a reduction from a $\oplus$P-hard problem is possible XOR-cross-compositional problems do not have a conjunctive or disjunctive kernel, unless NP $\subseteq$ coNP/poly. By way of an example, we find that Odd Path on planar graphs, the problem of finding whether a planar graph contains an odd number of paths of length k, is unlikely to have conjunctive or disjunctive kernels.

Furthermore, we find an example of problem that does not have conjunctive or disjunctive kernels using the traditional complexity-theoretic approach to lower bounds, i.e., through a reduction from an NP-hard problem, in this case, Longest Path. The problem Weighted Odd Path on planar graphs, a weighted version of the Odd Path problem, does not have conjunctive or disjunctive kernels, unless NP $\subseteq$ coNP/poly. We hope that this approach, using the classic complexity-theoretic results to achieve the lower bound, can be used to prove similar lower bounds on the existence of conjunctive and disjunctive kernels for more classical FPT problems in the future.

These lower bounds do not provide lower bounds for the existence of Turing kernels in general, but they do provide lower bounds for some of the most common kinds of Turing kernelizations.

2 Lower Bounds for Conjunctive Kernels

First we want to prove that there are no conjunctive kernels for OR-compositional parameterized problems. In order to do that, we first formally define the notion of conjunctive kernelization.

▶ **Definition 1** (Conjunctive Compression)**.** Given two parameterized problems L and R, a conjunctive compression is an algorithm that given an instance (x, k) for L of size n, it outputs in polynomial time a set of instances (x'_i, k'_i) for R with $1 \leq i \leq p(n)$ and $|x'_i|, k'_i \leq q(k)$ for every i for some polynomials $p(\cdot)$ and $q(\cdot)$, such that $(x, k) \in L$ if and only if $(x'_i, k'_i) \in R$ for every i.

2.1 Conjunctive OR-distillations

To prove that this type of kernels do not exist we have to go to the original proof of nonexistence of polynomial compressions for OR-distillable problems. Given a problem L, we define the problem OR(L), where the input is a set of instances $x_1, \dots, x_t$ of length at most n, and the task is to decide whether there exists an i such that $x_i \in L$.

First we define distillation algorithm.

▶ **Definition 2** (OR-Distillation Algorithm [2]). Let $L, R \subseteq \{0,1\}^*$ be a pair of languages and let $t\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ be a function. Then a t-bounded OR-distillation algorithm from L into R is an algorithm A that for every n, given $t(n)$ input strings $x_1, \ldots, x_{t(n)}$, with $|x_i| = n$ for all i, A runs in polynomial time and outputs a string y of length at most $t(n) \cdot \log n$ such that $y \in R$ if and only if $x_i \in L$ for some $i \in \{1, \ldots, t(n)\}$.

Essentially, an OR-distillation for a problem L is a reduction from OR(L) into R. Recall that, an NP-hard problem cannot have an OR-distillation, unless NP $\subseteq$ coNP/poly [13]. A Conjunctive OR-Distillation Algorithm is an extension of this definition.

▶ **Definition 3** (Conjunctive OR-Distillation Algorithm). Let $L, R \subseteq \{0,1\}^*$ be a pair of languages and let $t\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ be a function. Then a t-bounded conjunctive OR-distillation algorithm from L into R is an algorithm A that for every n, given $t(n)$ input strings $x_1, \ldots, x_{t(n)}$, with $|x_i| = n$ for all i, A runs in polynomial time and outputs $p(n)$ strings $y_1, \ldots, y_{p(n)}$ of length at most $t(n) \cdot \log n$ such that $y_i \in R$ for every $i \in \{1, \ldots, p(n)\}$ if and only if $x_j \in L$ for some $j \in \{1, \ldots, t(n)\}$.

We can use the same procedure as in [13] to rule out the existence of Conjunctive OR-Distillation Algorithms for NP-hard languages.

We take the pigeonhole lemma from Fomin et al. [12].

▶ **Lemma 4.** *Let X, Y be finite sets, t be a natural number and $\beta\colon X^t \to Y$ be a mapping. We say that $y \in Y$ covers $x \in X$ if there exist $x_1, \ldots, x_t$ such that $x_i = x$ for some i, $\beta((x_1, \ldots, x_t)) = y$. Then at least one element from Y covers at least $|X|/\sqrt[t]{|Y|}$ elements of X.*

Now we are ready to prove the following theorem.

▶ **Theorem 5.** *If there is a t-bounded Conjunctive OR-distillation algorithm from a language $L \subseteq \{0,1\}^*$ into a language $R \subseteq \{0,1\}^*$ for some polynomially bounded function t, then $\bar{L} \in$ NP/poly. In particular if L is NP-hard, then coNP $\subseteq$ NP/poly.*

Proof. Let $n \in \mathbb{N}$ and $\bar{L}_n = \{x \in \bar{L} \mid |x| = n\}$. Let us assume that there is a t-bounded Conjunctive OR-distillation algorithm A from L into R. Then, for each input $(x_1, \ldots, x_{t(n)}) \in \bar{L}_n^{t(n)}$, A outputs $y_1, \ldots, y_{p(n)}$ such that $|y_i| \le t(n) \cdot \log n$ and $y_i \in \bar{R}$ for at least one i. Define $\beta\colon \bar{L}_n^{t(n)} \to \bar{R}_{\le t(n)\log(n)}$ as follows: If $A(x_1, \ldots, x_{t(n)}) = y_1, \ldots, y_{p(n)}$ then define $\beta(x_1, \ldots, x_{t(n)}) = y_i$ where i is the smallest index for which $y_i \in \bar{R}$. Now apply Lemma 4 to this function β with $t = t(n)$, $X = \bar{L}$ and $Y = \bar{R}$. The lemma then states that there is a $z_1 \in \bar{R}$ that covers at least $|\bar{L}_n|/|\bar{R}_{t(n)\log(n)}|^{1/t(n)} \ge |\bar{L}_n|/n$ elements from $\bar{L}$.

Let Z_1 the set of $x \in \bar{L}_n$ covered by z_1. Then we consider the set $\bar{L}_n \setminus Z_1$ and find in the same way a z_2 that covers as many instances from $\bar{L}_n \setminus Z_1$ as possible. Repeating this for $2n$ rounds we get a set of z_i's that together cover all of $\bar{L}_n$ just as in the proof by Fortnow and Santhanam [13]: In each round the number of uncovered elements is reduced by a factor of $1 - 1/n$ and $(1 - 1/n)^{2n} < 2^{-n}$. As $|\bar{L}| \le 2^n$ the number of uncovered elements in the end is less than one and hence it is exactly zero. In conclusion we have shown that there is a polynomial size subset S_n of elements of $\bar{R}$ that covers the whole $\bar{L}_n$.

We want to show next how a nondeterministic TM can decide $\bar{L}$ in polynomial time with the help of polynomial advice. For an input x of size n the TM does the following: First, it guesses $x_1, \ldots, x_{t(n)}$ such that $x = x_i$ for some i and $x_j \in \bar{L}$ for every $j \ne i$. Then it computes $A(x_1, \ldots, x_{t(n)}) = y_1, \ldots, y_{p(n)}$ and checks if $y_i \in S_n$ for at least one i. The set S_n is read from the advice tape. If there is indeed such a $y_i \in S_n$ the machine accepts x and rejects it otherwise.

If $x \in \bar{L}$ then the TM will accept: As x is covered by some $y \in S_n$ there must be such an $x_1, \dots, x_{t(n)}$ that lets the machine accept x and it can guess it nondeterministically. On the other hand, if $x \in L$ it is easy to see that the TM cannot accept for any guessed $x_1, \dots, x_{t(n)}$.

If L is NP-hard then $\bar{L}$ is coNP-hard. This means that every problem in coNP can be reduced to $\bar{L}$. Thus, coNP $\subseteq$ NP/poly. ◀

2.2 Conjunctive Kernels

In order to use Theorem 5 to prove that some parameterized problems are unlikely to have conjunctive kernelizations, we use OR-cross-compositions and polynomial parameter transformations.

▶ **Definition 6** (Bodlaender, Jansen, and Kratsch [3]). An equivalence relation R on the set Σ^* is a *polynomial equivalence relation* if there exists an algorithm that given two strings $x, y \in \Sigma^*$ resolves whether x is equivalent to y in polynomial time in $|x| + |y|$, and for any finite set $S \subseteq \Sigma^*$, R partitions the elements of S into at most $(\max_{x \in S} |x|)^{O(1)}$ classes.

▶ **Definition 7** (Bodlaender, Jansen, and Kratsch [3]). Let $L \subseteq \Sigma^*$ be a language and $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized language. We say that L *OR-cross-composes* into Q if there exists a polynomial equivalence relation R and an algoritm A called an OR-cross-composition such that A takes as input $x_1, \dots, x_t \in \Sigma^*$ equivalent with respect to R, and outputs one instance $(y, k) \in \Sigma^* \times N$ such that k is polynomial in the size of x_i and $\log t$, and $(y, k) \in Q$ if and only if there exists one index i such that $x_i \in L$.

In particular, for any problem L, we have a trivial OR-cross-composition to OR(L) parameterized by the size of the largest instance $|x_i| = n$.

▶ **Theorem 8.** *Let $L \subseteq \Sigma^*$ be an NP-hard language. If L OR-cross-composes into a parameterized problem Q and Q has a conjunctive kernelization, then* coNP $\subseteq$ NP/poly*.*

Proof Sketch. The proof is equivalent to that in Bodlaender et al. [2], for polynomial compressions. Given an NP-hard language L and an OR-cross-composition from L into a parameterized problem Q, if Q had a conjunctive kernelization, one would be able to build a conjunctive distillation for L. ◀

From here we are only left to list problems that are unlikely to have a conjunctive kernelization because an NP-hard problem OR-cross-composes into them. These problems, called *OR-compositional*, form an extensive list, a subset of them would be.

▶ **Corollary 9.** *Unless* coNP $\subseteq$ NP/poly*, none of the following FPT problems have conjunctive kernelizations: Longest Path, Longest Cycle, Exact Cycle, Short Cheap Tour, Graph Minor Order Test, Bounded Treewidth Subgraph Test, Steiner Tree, and Clique(VC).*

Proof. Exact Cycle, Short Cheap Tour, Graph Minor Order Test and Bounded Treewidth Subgraph Test are defined in Bodlaender et al. [2], the authors provide OR-cross-compositions from their own unparameterized versions for these problems together with Longest Path and Longest Cycle. Dom, Lokshtanov, and Saurabh provide an OR-cross-composition for Steiner Tree [8], and finally, the OR-cross-composition for Clique(VC) is due to Bodlaender Jansen and Kratsch [3]. ◀

Most of the problems on this list cross-compose from their own unparameterized versions, or from a similar problem. But that is not the only option. Once one has a list of problems unlikely to have conjunctive kernelizations one can extend it using Polynomial parameter transformations (PPT).

▶ **Definition 10** (Bodlaender, Thomassé, and Yeo [4])**.** Given two parameterized problems P and Q an algorithm A is a PPT from P to Q if given an instance (x, k) for P, A transforms it in polynomial time into an instance (x', k') for Q such that k' is polynomial in k and $(x, k) \in P$ if and only if $(x', k') \in Q$.

At this point it is easy to see that, if a problem Q has a conjunctive kernelization, and there is a PPT from P to Q then P also has a conjunctive kernelization (or compression).

Due to the existence of appropriate PPTs the following problems are also unlikely to have conjunctive kernelizations:

▶ **Corollary 11.** *Unless* coNP $\subseteq$ NP/poly*, none of the following FPT problems have conjunctive kernelizations: PATH PACKING, CYCLE PACKING, RED BLUE DOMINATING SET, SET COVER($|U|$), CONNECTED VERTEX COVER, and CAPACITADED VERTEX COVER.*

Proof. There are PPTs for PATH PACKING and CYCLE PACKING from LONGEST PATH and LONGEST CYCLE respectively by Fomin et al. [12]. RED BLUE DOMINATING SET has a PPT from its own colored version due to Dom Lokstanov and Saurabh [8], furthermore SET COVER($|U|$), CONNECTED VERTEX COVER and CAPACITADED VERTEX COVER have PPTs to RED BLUE DOMINATING SET, as was also shown in [8, 12]. ◀

The list of problems unlikely to have conjunctive kernelization through PPTs is longer, we only presented some examples.

3 Lower Bounds for Disjunctive Kernels

From the results we have seen so far for OR-compositional problems, it would seem reasonable to find similar results for AND-compositional problems, however, the proof that AND-distillable problems are unlikely to have compressions was not as straightforward. Drucker [11], proved that it is also unlikely that OR-distillable problems have probabilistic compressions and more importantly, that AND-distillable problems are also unlikely to have deterministic or probabilistic compressions.

3.1 Disjunctive AND-distillations

A simplified version of Drucker's approach [11] was presented in a conference version of the paper [10] and in the following book [9]. This simplified version does not have the same scope as the full proof but it can be used to prove that AND-distillable problems are unlikely to have deterministic compressions.

We now extend this result to state that AND-distillable problems are unlikely to have deterministic disjunctive compressions.

The notion of a disjunctive algorithm can be defined analogously the to the conjunctive version.

▶ **Definition 12** (Disjunctive Distillation Algorithm)**.** Let $L, R \subseteq \{0,1\}^*$ be a pair of languages and let $t\colon \mathbb{N} \to \mathbb{N}\backslash\{0\}$ be a function. Then a t-bounded disjunctive AND-distillation algorithm from L into R is an algorithm A that for every n, given $t(n)$ input strings $x_1, \ldots, x_{t(n)}$, with

$|x_i| = n$ for all i, A runs in polynomial time and outputs $p(n)$ strings $y_1, \dots, y_{p(n)}$, where each y_i has length at most $t(n) \log n$, such that $y_i \in R$ for some $i \in \{1, \dots, p(n)\}$ if and only if $x_j \in L$ for every $j \in \{1, \dots, t(n)\}$.

Also, given a problem L, we define the problem AND(L), where the input is a set of instances $x_1, \dots, x_t$ of length at most n, and the task is to decide whether $x_i \in L$ for every $i \in \{1, \dots, t\}$.

Equivalently, we can consider the algorithm A as a set of $p(n)$ polynomially computable functions $f_i \colon \{0,1\}^{t(n)\times n} \to \{0,1\}^{t(n) \log n}$ in such a way that the required property is satisfied.

We start by introducing some basic notions and lemmas.

Given a statistical distribution $\mathcal{D}$, the *support* of $\mathcal{D}$, $\sup(\mathcal{D})$ is the set of values that $\mathcal{D}$ assumes with nonzero probability, and $\mathcal{D}(u) = \text{Prob}[\mathcal{D} = \text{u}]$. We can assume from now on that we talk about distributions with finite supports.

▶ **Definition 13** (Statistical Distance). The *statistical distance* of distributions $\mathcal{D}$ and $\mathcal{D}'$ is

$$\|\mathcal{D} - \mathcal{D}'\|_{\text{stat}} = \frac{1}{2} \sum_{u \in \sup(\mathcal{D}) \cup \sup(\mathcal{D}')} |\mathcal{D}(u) - \mathcal{D}'(u)|$$

Just to get some intuition, if two distributions have completely different supports, their distance is 1. If two distributions share some support their distance is smaller.

Now, given a function $f \colon \{0,1\}^{t\times n} \to \{0,1\}^{t'}$, given a subset $A \subseteq \{0,1\}^n$ we can define the distribution $\mathbf{F}_A = f(\mathcal{U}_A^{\otimes t})$, where $\mathcal{U}_A^{\otimes t}$ is the uniform distribution over t-tuples of elements of A. Moreover, for any $a \in \{0,1\}^n$ we define the distribution

$$\mathbf{F}_A[a] = f(\mathcal{U}_A^{\otimes(j-1)}, a, \mathcal{U}_A^{\otimes(t-j)}) \ ,$$

where j is sampled from the uniform distribution over the integers from 1 to t, $\mathcal{U}_{[t]}$.

Finally, we define the *standout factor* of a with respect to A as

$$\beta(a, A) = \|\mathbf{F}_A - \mathbf{F}_A[a]\|_{\text{stat}} \ .$$

The Disguising-Distribution lemma states

▶ **Lemma 14** (Drucker [10]). *Let $f \colon \{0,1\}^{t\times n} \to \{0,1\}^{t'}$ be any (possibly randomized) function for $t, t' \in \mathbb{N}^+$. Let $S \subseteq \{0,1\}^n$, and fix $d > 0$. Given any $\varepsilon > 0$, let $s = \lceil (0.5 \ln 2)n/\varepsilon^2 \rceil$. Let*

$$\hat{\delta} = \min\left\{ \sqrt{(\ln 2)(t'+1)/2t}, 1 - 2^{-3-t'/t} \right\} .$$

Then there exists a collection $K_1, \dots, K_s$ of multisets of size d contained in S, such that for every $y \in S$, we have

$$\mathbb{E}_{i \sim \mathcal{U}_{[s]}}[\beta(y, K_i \setminus y)] \le \hat{\delta} + 2t/(d+1) + \varepsilon \ .$$

Intuitively, given a language L, this lemma allows us to pick multisets of L of size d (potentially polynomial), and given any other element of L, the expected statistical distance to one of these multisets will be small, allowing us to distinguish potentially, if a particular element is in L or not. In order for these statistical distances to give meaningful complexity theoretic consequences, we turn to promise problems, in particular circuits.

For a boolean circuit $\mathcal{C}$ let $\mathcal{D}_C$ be the output distribution for a random input. Given two parameters $0 \le d \le D \le 1$, we define the promise problem $\text{SD}^{\ge \text{D}}_{\le \text{d}}$ with the yes instances being $\Pi_Y = \{\langle C, C'\rangle \colon \|\mathcal{D}_C - \mathcal{D}_{C'}\|_{\text{stat}} \ge D\}$, and the subset of no instances $\Pi_N = \{\langle C, C'\rangle \colon \|\mathcal{D}_C - \mathcal{D}_{C'}\|_{\text{stat}} \le d\}$.

We will use the class of promise problems having statistical zero-knowledge proofs (pr-SZK). In particular

▶ **Theorem 15** (Sahai and Vadhan [20]). *Let $0 \leq d = d(n) \leq D = D(n) \leq 1$ be (not necessarily computable) parameters.*

1. *If $D > d + 1/\mathrm{poly(n)}$, then $\mathrm{SD}^{\geq \mathrm{D}}_{\leq \mathrm{d}} \in \text{pr-NP/poly}$.*
2. *If we have the stronger gap $D^2 > d + 1/\mathrm{poly(n)}$, then $\mathrm{SD}^{\geq \mathrm{D}}_{\leq \mathrm{d}}$ is many-to-one reducible to $\mathrm{SD}^{\geq 2/3}_{\leq 1/3} \in \text{pr-SZK}$.*

From Lemma 14 and Theorem 15, Drucker [10] proves that (probabilistic) AND-compositional and OR-compositional problems are unlikely to have polynomial kernels. We adapt the proof to our case and state the following theorem for disjunctive kernels for deterministic AND-compositional problems. In Subsection 4.1 we show how one can analogously adapt the proof to work in the probabilistic case, this allows us to present first a self-contained proof that focuses on the difference between AND-distillations and disjunctive AND-distillations without worrying about the probabilistic aspect. This result is already sufficient to prove that AND-compositional problems are unlikely to have disjunctive kernels.

▶ **Theorem 16.** *Let L be any language. Suppose that $t(n)\colon \mathbb{N}^+ \to \mathbb{N}^+$ is a function. Suppose that there exists a t-bounded disjunctive AND-distillation defined by $p(n)$ functions $f_i\colon \{0,1\}^{t(n)\times n} \to \{0,1\}^{t(n)\log n}$ for L with parameter $t(n)$, and some target language R. Then, $L \in \text{coNP/poly}$.*

Proof. We prove this theorem by reducing $\overline{L}$ to $\mathrm{SD}^{\geq \mathrm{D}}_{\leq \mathrm{d}}$. First of all, let us define $L_n = \{0,1\}^n \cap L$, and let us assume that $L_n \subsetneq \{0,1\}^n$. For every function f_i we apply the disguising distribution lemma to L_n. This means that for every i, we define $S = L_n$, $f = f_i$, $t = t(n)$, $t' = t(n)\log n$, $\varepsilon = 1/4n^c$, $d = 8t(n)n^c$ and $\hat{\delta} = \min\left\{\sqrt{(\ln 2)(t\log n + 1)/2t}, 1 - 2^{-3-\log n}\right\}$. Lemma 14 states that for every f_i there exists a collection $K^i_1, \dots, K^i_s$ of multisets of size d contained in L_n such that for every $y \in L_n$ we have

$$\mathbb{E}_{j\sim\mathcal{U}_{[s]}}[\beta(y, K^i_j \setminus y)] \leq \hat{\delta} + 2t/(d+1) + \varepsilon\ .$$

Observe, that in this case as long as $t(n)$ is polynomial in n, both s and d are polynomial in n, too. Thus, as long as $p(n)$ is also polynomial in n, which by construction it is if this condition is satisfied for $t(n)$, the collection of all subsets for every i is polynomial in size, too. This will be our advice.

On input $y \in \{0,1\}^n$, our reduction outputs for every mapping f_i, $\langle C_i, C'_i\rangle$ of the following randomized circuits. Circuit C_i samples $j \sim \mathcal{U}_{[s]}$, then samples $\hat{x}_i \sim \mathcal{U}^{\otimes t(n)}_{K^i_j}$, and outputs $z_i = f_i(\hat{x}_i)$. Circuit C'_i samples $j \sim \mathcal{U}_{[s]}$, and $k \sim \mathcal{U}_{t(n)}$, then samples $\hat{x}_i \sim \left(\mathcal{U}^{\otimes(k-1)}_{K^i_j}, y, \mathcal{U}^{\otimes(t(n)-k)}_{K^i_j}\right)$ and outputs $z_i = f_i(\hat{x}_i)$.

The idea is that we have a control circuit C that only samples elements from L, the other circuit C' takes also the original input, if both outputs are in the target language this should be reflected in their statistical distance being small. On the other hand, if the input is not in L, then the output generated by circuit C' is not in the target language, thus, the support of C' is completely disjunct from the support of the distribution of control circuit C and their statistical distance is 1. This however, has to be tempered by the disjunctive property of the reduction.

▷ Claim 17. The following holds:

1. If $y \notin L$, then for every i, we have $\|\mathcal{D}_{C_i} - \mathcal{D}_{C'_i}\|_{\mathrm{stat}} = D(n) = 1$;
2. If $y \in L$ then there exists an i such that, $\|\mathcal{D}_{C_i} - \mathcal{D}_{C'_i}\|_{\mathrm{stat}} \leq d(n) = \hat{\delta} + 1/2n^c$

The first part follows from the AND-respecting and disjunctive properties of the reduction. If there is an element that is not in L, there cannot be any mapping that outputs an element of the target language. Thus, every possible output for C'_i will not be in R, while every output from C_i is, thus, these two distributions have completely different supports and they do not intersect, thus their statistical distance must be 1. The second part follows from Lemma 14 and the disjunctive properties of the reduction, the formal proof will follow.

Now, to prove Theorem 16, if $1 - \hat{\delta} \geq 1/n^c$ for sufficiently large n, then $1 - d(n) = D(n) - d(n) \leq 1/n^c$, thus the gap is inversely polynomial in n and also inversely polynomial in the length of the output pairs $\langle C_i, C'_i \rangle$. Thus, given an instance y of the decision problem L, there exists at least one i, for which the pair $\langle C_i, C'_i \rangle$ is a valid instance of the promise problem $\mathrm{SD}^{\geq 1}_{\leq \mathrm{d(n)}}$. We know by Theorem 15 that for the given values of $d(n)$ and $D(n)$, $\mathrm{SD}^{\geq \mathrm{D(n)}}_{\leq \mathrm{d(n)}} \in$ pr-NP/poly. Thus, assume we are given an algorithm A that, with advice a_n, solves $\mathrm{SD}^{\geq \mathrm{D(n)}}_{\leq \mathrm{d(n)}}$ in nondeterministic polynomial time. Given an instance y, we can run the reduction and then apply this algorithm to every $\langle C_i, C'_i \rangle$ given by instance y and correctly decide if $y \notin L$. In particular, if $y \notin L$, for every i, $\langle C_i, C'_i \rangle \in \Pi_Y$, otherwise $y \in L$ and there exists one i such that $\langle C_i, C'_i \rangle \in \Pi_N$. Note here, that there might be pairs that are not a valid instance for the promise problem, but this is not a concern, as there is at least one which will be solved by A together with the advice. Observe that this procedure runs in polynomial time, as one can run A in parallel for every pair $\langle C_i, C'_i \rangle$, and the amount of pairs is $t(n)$ which we have assumed to be polynomial in n. Thus, $\overline{L} \in$ NP/poly, and $L \in$ coNP/poly. ◀

Proof of Claim 17. The first part of the claim is already proven after the claim statement. For the second part, given a disjunctive AND-distillation defined by $p(n)$ functions $f_i \colon \{0,1\}^{t(n) \times n} \to \{0,1\}^{t(n) \log n}$ for L with parameter $t(n)$, and some target language R, and $y \in L$ then, given $(x_1, \dots, x_t)$ such that every $x_m \in L$ for every $m \in [t(n)]$ and $x_j = y$ for some $j \in [t(n)]$, there exists an $i \in [p(n)]$ such that $f_i(x_1, \dots x_t) \in R$. Thus, by Lemma 14 if we take the same values of d, $\hat{\delta}$ and ε as in the main theorem, there exists a collection $K^i_1, \dots, K^i_s$ of multisets of size d contained in L_n such that for every $y \in L_n$ we have

$$\mathbb{E}_{j \sim \mathcal{U}_{[s]}}[\beta(y, K^i_j \setminus y)] \leq \hat{\delta} + 2t/(d+1) + \varepsilon \, .$$

But, recall that $\beta(y, K^i_j \setminus y)$ is nothing else than the statistical distance between the two distributions, which means that the expected value for all possible values of j is the same as $\|\mathcal{D}_{C_i} - \mathcal{D}_{C'_i}\|_{\mathrm{stat}}$ by construction, and also $\hat{\delta} + 2t/(d+1) + \varepsilon = \hat{\delta} + 1/2n^c = d(n)$. ◁

3.2 Disjunctive Kernels

In order to use this result to prove that some parameterized problems do not have conjunctive kernelizations, we use AND-cross-compositions and polynomial parameter transformations just as we did in Subsection 2.2.

We are going to define AND-cross-compositions analogously to OR-cross-compositions using equivalence relations.

▶ **Definition 18.** Let $L \subseteq \Sigma^*$ be a language and $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized language. We say that L AND-cross-composes into Q if there exists a polynomial equivalence relation R and an algoritm A called a cross-composition such that A takes as input $x_1, \dots, x_t \in \Sigma^*$ equivalent with respect to R, and outputs one instance $(y, k) \in \Sigma^* \times N$ such that k is polynomial in the size of x_i and $\log t$, and $(y, k) \in Q$ if and only if for every index i, $x_i \in L$.

Now we are ready to state the following theorem.

▶ **Theorem 19.** *Let $L \subseteq \Sigma^*$ be an NP-hard language. If L AND-cross-composes into a parameterized problem Q and Q has a disjunctive kernelization, then* coNP $\subseteq$ NP/poly*.*

Proof Sketch. The proof is analogous to the one for Theorem 8. The main idea is that if we had an AND-cross-composition into such a parameterized problem, and we had a disjunctive kernelization for that problem, we would be able to build a disjunctive distillation for our original problem, which by Theorem 16 would imply that coNP $\subseteq$ NP/poly. ◀

From here we are only left to list problems that are unlikely to have a disjunctive kernelization because an NP-hard problem cross-composes into them. These problems, sometimes called AND-compositional, form an extensive list, a subset of them would be:

▶ **Corollary 20.** *Unless* coNP $\subseteq$ NP/poly*, none of the following FPT problems have disjunctive kernelizations: Treewidth, Pathwidth, Cutwidth, and Modified Cutwidth.*

Proof. For the definitions of these problems together with their the AND-compositions we refer to Bodlaender et al. [2]. ◀

Just as in the conjunctive case, the list of problems unlikely to have disjunctive kernelizations be extended using polynomial parameter transformations (PPT).

Other problems that admit AND-compositions or admit PPT to AND-compositional problems are some problems parameterized by treewidth (ω).

▶ **Corollary 21.** *Unless* coNP $\subseteq$ NP/poly*, none of the following FPT problems have disjunctive kernelizations: Independent Set(ω) and 3-coloring(ω).*

Proof. Bodlaender et al. present AND-compositions for these two problems in [2]. ◀

4 Problems without Conjunctive or Disjunctive Kernels

In the previous two sections we have proven that there are problems that are unlikely to have conjunctive kernels and disjunctive kernels, respectively. However, none of these problems intersect. In this section, we prove that if a $\oplus$P-hard problem is in coNP/poly then coNP $\subseteq$ NP/poly. Furthermore, we use this result to find a class of problems without conjunctive or disjunctive kernels if they reduce from a problem that is NP-hard or $\oplus$P-hard. For each class, we present an FPT problem unlikely to have conjunctive or disjunctive kernels. The problem Odd Path on planar graphs does not have conjunctive or disjunctive kernels, unless coNP $\subseteq$ NP/poly, due to a reduction from its own unparameterized version and its weighted version, the Weighted Odd Path problem on planar graphs, does not have conjunctive or disjunctive kernels, unless coNP $\subseteq$ NP/poly, due to a reduction from Longest Path, an NP-hard problem.

4.1 Probabilistic Compressions

First we provide the framework for Probabilistic compressions.

▶ **Definition 22** (Probabilistic conjunctive OR-distillation)**.** Let $L, R \subseteq \{0,1\}^*$ be a pair of languages and let $t\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ be a function. Then a probabilistic t-bounded conjunctive OR-distillation algorithm from L into R with bounded error $0 \leq \xi < 0.5$ is an algorithm A that for every n, given $t(n)$ input strings $x_1, \dots, x_{t(n)}$, with $|x_i| = n$ for all i, A runs in polynomial time and outputs $t(n)$ strings $y_1, \dots, y_{t(n)}$ of length at most $t(n) \cdot \log n$ such that with a probability greater than $1 - \xi$, $y_i \in R$ for every $i \in \{1, \dots, t(n)\}$ if and only if $x_j \in L$ for some $j \in \{1, \dots, t(n)\}$.

Essentially one can make mistakes with probability smaller than ξ but otherwise the algorithm should produce a conjunctive OR-distillation. And analogously we can define probabilistic disjunctive AND-distillations.

▶ **Definition 23** (Probabilistic disjunctive AND-distillation). Let $L, R \subseteq \{0,1\}^*$ be a pair of languages and let $t\colon \mathbb{N} \to \mathbb{N} \setminus \{0\}$ be a function. Then a probabilistic t-bounded disjunctive AND-distillation algorithm from L into R with bounded error $0 \leq \xi < 0.5$ is an algorithm A that for every n, given $t(n)$ input strings $x_1, \ldots, x_{t(n)}$, with $|x_i| = n$ for all i, A runs in polynomial time and outputs $t(n)$ strings $y_1, \ldots, y_{t(n)}$, where each y_i has length at most $t(n) \log n$, such that with probability greater than $1 - \xi$ $y_i \in R$ for some $i \in \{1, \ldots, t(n)\}$ if and only if $x_j \in L$ for every $j \in \{1, \ldots, t(n)\}$.

We can now prove that if such distillations exist then the language is in coNP/poly by adapting the proof by Drucker [10] to work as well for the probabilistic version of conjunctive and disjunctive distillations.

▶ **Theorem 24.** *Let L be any language. Suppose that $t(n)\colon \mathbb{N}^+ \to \mathbb{N}^+$ is a function. Suppose that there exists a probabilistic conjunctive OR-distillation or a probabilistic disjunctive AND-distillation defined by $p(n)$ functions $f_i\colon \{0,1\}^{t(n)\times n} \to \{0,1\}^{t(n)\log n}$ for L with parameter $t(n)$, error bound $\xi(n) < 0.5$, and some target language R. Let*

$$\hat{\delta} = \min\left\{\sqrt{(\ln 2)(t'+1)/2t}, 1 - 2^{-3-t'/t}\right\}.$$

If, for some constant $c > 0$, we have $1 - 2\xi(n) - \hat{\delta} \geq 1/n^c$, then $L \in$ coNP/poly.

The proof of this theorem is completely analogous to the proof of Theorem 16, where the treatment of the probabilistic aspect is analogous to the one by Drucker [10]. One needs to be careful to state an appropriate version of Claim 17 for both the conjunctive and disjunctive cases, which should take into account the probabilistic aspect of the distillation, the OR or AND properties of the reduction itself, as well as the conjunctive or disjunctive aspect. These claims read as follows.

▷ Claim 25. If L is a language with a probabilistic conjunctive OR-distillation as determined in Theorem 24, the following holds:

1. If $y \notin L$, then there exists an i such that, $\|\mathcal{D}_{C_i} - \mathcal{D}_{C'_i}\|_{\text{stat}} \geq D(n) = 1 - 2\xi(n)$;
2. If $y \in L$ then for every i, we have $\|\mathcal{D}_{C_i} - \mathcal{D}_{C'_i}\|_{\text{stat}} \leq d(n) = \hat{\delta} + 1/2n^c$

▷ Claim 26. If L is a language with a probabilistic disjunctive AND-distillation as determined in Theorem 24, the following holds:

1. If $y \notin L$, then for every i, we have $\|\mathcal{D}_{C_i} - \mathcal{D}_{C'_i}\|_{\text{stat}} \geq D(n) = 1 - 2\xi(n)$;
2. If $y \in L$ then there exists an i such that, $\|\mathcal{D}_{C_i} - \mathcal{D}_{C'_i}\|_{\text{stat}} \leq d(n) = \hat{\delta} + 1/2n^c$

These claims can be proven using Lemma 14 and then Theorem 24 follows also from applying the same circuit reduction, now with the probabilistic aspect, and using Theorem 15.

4.2 XOR-compositional Problems

In the same way that we defined for a problem L, the problem OR(L) and AND(L) one can easily imagine defining problems for other boolean operations, for example XOR(L) would count the parity of the number of yes instances. Formally,

▶ **Definition 27.** Given a problem L, we define the boolean variable $b_L(x)$ such that $b_L(x) = 1$ if and only if $x \in L$. We say that a set of instances $(x_1, \ldots, x_t)$ each having a length of at most n is in XOR(L) if and only if $\text{XOR}(b_L(x_1), \ldots, b_L(x_t)) = 1$.

Analogously we define XOR-cross-composition.

▶ **Definition 28.** Let $L \subseteq \Sigma^*$ be a language and $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized language. We say that L XOR-cross-composes into Q if there exists a polynomial equivalence relation R and an algoritm A called a cross-composition such that A takes as input $x_1, \ldots, x_t \in \Sigma^*$ equivalent with respect to R, and outputs one instance $(y, k) \in \Sigma^* \times N$ such that k is polynomial in the size of x_i and $\log t$, and $(y, k) \in Q$ if and only if there is an odd number of instances $x_i \in L$.

The goal is to prove that for some problems L, if one has an XOR-cross-composition from L into a parameterized problem Q, and a conjunctive or disjunctive kernelization for Q, then one would be able to build a conjunctive or disjunctive distillation for L, which by Theorem 24 means that $L \subseteq \text{NP/poly}$. Thus, if L is for instance NP-hard, Q is unlikely to have conjunctive or disjunctive kernels.

In general, when we think of problems that XOR-cross-compose to their own parameterized versions, or problems where L is equivalent to XOR(L), an instance will be in L then, if we have an odd number of solutions, which in general means, that they belong to $\oplus$P, or are $\oplus$P-hard instead of NP-hard, making it difficult to find problems that are unlikely to have conjunctive and disjunctive, or even polynomial kernels through the usual complexity-theoretic consequence coNP $\subseteq$ NP/poly. To avoid this problem, we establish a collapse of the polynomial hierarchy in the case of $\oplus$P problems. First, we formally define some complexity classes and state a few complexity theoretic results.

We first define the general complexity class $\text{BP} \cdot \mathcal{C}$.

▶ **Definition 29** (Schöning [21]). Given a complexity class $\mathcal{C}$, the bounded error probabilistic-$\mathcal{C}$ class ($\text{BP} \cdot \mathcal{C}$), is defined as the class of all languages L such that for some language $L' \in \mathcal{C}$, for some polynomial p and for all inputs x of length n, we have

$$\text{Prob}[(x, y) \in L' \text{ if } x \in L] \geq 2/3$$
$$\text{Prob}[(x, y) \in L' \text{ if } x \notin L] \leq 1/3$$

where y is chosen uniformly at random from all strings of length $p(n)$.

With this definition we prove the following theorem.

▶ **Theorem 30.** $\text{BP} \cdot (\text{NP/poly}) \subseteq \text{NP/poly}$.

Proof. Given the defintion of $\text{BP} \cdot (\mathcal{C})$, and using amplification, we know that a language L is in $\text{BP} \cdot (\text{NP/poly})$ if there exists a language $L' \in \text{NP/poly}$ and a polynomial p such that for all inputs of length n, we have

$$\text{Prob}[(x, y) \in L' \text{ if } x \in L] \geq 1 - 2^{-n^2}$$
$$\text{Prob}[(x, y) \in L' \text{ if } x \notin L] \leq 2^{-n^2}$$

where y is chosen uniformly at random from all strings of length $p(n)$.

Let us consider such a language $L \in \text{BP} \cdot (\text{NP/poly})$, with its corresponding language $L' \in \text{NP/poly}$ and polynomial p. There are only 2^n many inputs of length n for any given n, thus, by the union bound, the probability that a string string y provides a bad outcome for at least one input x of length n is at most $2^n \cdot 2^{-n^2} < 1$. Thus, there must be a fixed string y of length $p(n)$, such that $x \in L$ if and only if $(x, y) \in L'$ for every input x of length n.

A nondeterministic Turing machine M with polynomial advice can decide whether $x \in L$ in polynomial time as follows. Given a nondeterministic Turing machine N with polynomial advice solving L', M obtains first y from the advice tape and then tests whether $(x, y) \in L'$ by simulating N on (x, y) by using the subsequent advice that it also obtains from the same advice tape. Observe that both the advice and the running time of M are polynomial. ◀

We now formally define $\oplus$P introduced by Papadimitriou and Zachos [19].

▶ **Definition 31.** $\oplus$P is the class of decision problems solvable by a nondeterminsitic Turing machine in polynomial time, with an odd number of accepting computation paths.

An example of a $\oplus$P problem is Odd Path, where given a graph G one has to decide whether G contains an odd number of k-paths. For the purposes of this paper, we are interested on Odd Path on planar graphs, which can be formally defined as:

Input: A planar graph G and an integer k.
Task: Decide whether G contains an odd number of k-paths.

There are many other examples of problems in $\oplus$P or problems that are complete for $\oplus$P, a compillation can be found in [23].

An important property of $\oplus$P as it relates to the polynomial hierarchy (PH) is that PH $\subseteq$ BP $\cdot$ $\oplus$P, which is an important ingredient in the proof of Toda's theorem [22]. This property together with Theorem 30 allows us to prove that $\oplus P$-hard problems are unlikely to be in coNP/poly we can do it in the following way

▶ **Theorem 32.** *If a problem L is both $\oplus$P-hard and $L \in$ coNP/poly, then* coNP $\subseteq$ NP/poly.

Proof. If a language L is both $\oplus$P-hard and $L \in$ coNP/poly, this means that $\oplus$P $\subseteq$ coNP/poly, in particular, because $\oplus$P is closed under complement it implies that $\oplus$P $\subseteq$ NP/poly.

Using the fact that PH $\subseteq$ BP $\cdot$ $\oplus$P [22, Theorem 3.1] and Theorem 30, we can see that $\oplus$P $\subseteq$ NP/poly implies that

$$\mathrm{PH} \subseteq \mathrm{BP} \cdot \oplus\mathrm{P} \subseteq \mathrm{BP} \cdot (\mathrm{NP/poly}) \subseteq \mathrm{NP/poly}\ ,$$

which is unlikely because it makes the polynomial hierarchy colapse. In particular, coNP $\subseteq$ PH $\subseteq$ NP/poly brings us to our usual complexity theoretic consequence for this kind of lower bounds. A result by Yap [25] shows that coNP $\subseteq$ NP/poly means that the polynomial hierarchy collapses to the third level. A more refined result by Köbler and Watanabe [16] shows a stronger collapse, that is, if coNP $\subseteq$ NP/poly then the polynomial hierarchy collapses to ZPP(Σ_2^{P}). ◀

Moreover, we are able to prove that problems which XOR-cross compose into parameterized problems that have conjunctive or disjunctive polynomial kernels are in NP/poly.

▶ **Theorem 33.** *Let $L \subseteq \Sigma^*$ be a language. If L XOR-cross-composes into a parameterized problem Q and Q has a conjunctive or disjunctive kernelization, then $L \in$* coNP/poly.

Proof. We prove for conjunctive kernelizations, as we will see, the case for disjunctive kernelizations is analogous.

Let $L \subseteq \Sigma^*$ be a language, and let A be an XOR-cross-composes into a parameterized problem Q, which has a conjunctive kernelization. Let us build a probabilistic conjunctive OR-distillation for L. Given a set of inputs $x_1, \ldots x_t$ for L, the first step in our distillation is to delete every input x_i at random with probability 1/2. This way, if there are any $x_i \in L$, after this procedure there will be an odd number of them with probability 1/2 and if there were none to begin with, there are still none after this procedure. Then, to the remaining set of instances $x'_1, \ldots, x'_{t'}$ we apply the XOR-cross-composition and from the result we build a conjunctive kernel. The probability of success is 1 if the initial input contained no $x_i \in L$, and otherwise the probability of sucess is 1/2. This is not good enough, as we need a probabilistic conjunctive OR-distillation with an error bound smaller than 1/2 to be able to apply Theorem 24. In order to do this we sacrifice the onesided error in a clever way. With a probability of 1/4 we substitute the remaining set of instances $x'_1, \ldots, x'_{t'}$ for a single instance $x \in L$ before applying the XOR-cross-composition. The probability of success is now $3/8 + 1/4 = 5/8$ if $x_i \in L$ for some i, and 3/4 otherwise, making the total error bound $\xi \leq 3/8$. We have thus, a probabilistic conjunctive OR-distillation satisfying all of the conditions to apply Theorem 24, which means that $L \in \text{coNP/poly}$. The reduction for disjunctive kernelizations is analogous. ◀

Thus, by Theorems 32 and 33 if a problem is hard for NP or ⊕P and reduces to a parameterized problem, then the parameterized problem is unlikely to have polynomial conjunctive or disjunctive kernels.

▶ **Corollary 34.** *The* Odd Path *problem on planar graphs is FPT and does not have conjunctive or disjunctive kernels, unless* $\text{coNP} \subseteq \text{NP/poly}$*.*

Proof. The Odd Path problem is FPT on planar graphs because it can be solved by dynamic programming on graphs of bounded tree-depth using standard techniques. Reducing the problem from planar graphs to graphs of bounded tree-depth can be done with low tree-depth colorings [18].

The problem trivially XOR-cross-composes to itself and is in ⊕P. Moreover, Odd Path on planar graphs is ⊕P-hard because it can be reduced to the problem of finding the parity of the number of Hamiltonian paths ($k = n$) on planar graphs with maximum degree 3, which is ⊕P-complete [23]. ◀

The question whether Odd Path on general graphs is FPT remained open until very recently, when Curticapean, Dell and Husfeldt proved that Odd Path is ⊕-W[1]-complete, thus is unlikely to be fixed parameter tractable [6].

4.3 The Weighted Odd Path Problem

In order to find an FPT problem without conjunctive or disjunctive kernels, with a reduction from an NP-hard problem instead of a reduction from a ⊕P-hard problem, we consider the Weighted Odd Path problem with parameter k on planar graphs, i.e., the weighted version of the Odd Path problem:

Input: A planar graph G with edge weights and two integers k and m.
Task: Decide whether G contains an odd number of k-paths with weight m.

This problem is in FPT on planar graphs because it can be solved by dynamic programming on graphs of bounded tree-depth using standard techniques. Reducing the problem from planar graphs to graphs of bounded tree-depth can be done with low tree-depth colorings [18].

We now prove, through a reduction from Longest Path, that Weighted Odd Path is unlikely to have conjunctive or disjunctive kernels. The reduction uses a probabilistic XOR-cross-composition and then proves that conjunctive and disjunctive kernels are still unlikely following the same techniques as used in Theorem 33.

▶ **Theorem 35.** *Unless* coNP ⊆ NP/poly*,* Weighted Odd Path *on planar graphs has no disjunctive or conjunctive kernelizations.*

Proof. The XOR-cross-composition from Longest Path into Weighted Odd Path relies on the following observation. Let G be a graph of order n. If we turn G into an edge-weighted graph by equipping each edge with a random weight from $\{1, \ldots, n^3\}$ then the Isolation Lemma [17] shows that a k-path with minimum weight is unique with a failure probability of at most $1/n$. If, on the other hand, G does not contain a k-path, then its weighted version will not contain a k-path of any weight.

Thus, the reduction runs as follows. First we turn G into a weighted graph G' as explained above. Then create $k(n^3 - 1) + 1$ different instances $(G', k, k), (G', k + 1, k), \ldots, (G', kn^3, k)$ of the Weighted Odd Path problem. If G does contain a k-path, then with probability of at least $1 - 1/n$ one of the instances is a yes-instance. Second, we delete each instance from this collection independently with a probability of $1/2$. Let us call the remaining set of instances I. Now I contains an odd number of yes-instances with probabilty of at least $1/2 - 1/2n$ if G contains at least one k-path. Now, to offset the one-sided error, just as we did in the previous subsection, with a probability of $1/4$ we redefine I to be a single (trivial) yes-instance. If G contained no k-path, the probability that I does not contain a k-path is now $3/4$, whereas if G contains at least one k-path, now I contains an odd number of yes-instances with probabilty of at least $1/4 + 3/8(1 - 1/n) = 5/8 - 3/8n > 1/2$.

Third, we turn I into a single graph as follows: If I contains (G', i, k) we double each edge weight in G' and call the resulting weighted graph G''_i. Then we attach to each vertex in G''_i a new pendant vertex via a new edge with weight $n^7/2 - i$. Note that G''_i contains a $k + 2$-path of weight n^7 iff G' contains a k-path of weight i. Finally we create a new graph G''' that consists of all G''_i for $i \in \{k, \ldots, kn^3\}$.

We arrived at the following situation: If G does not contain a k-path, then with a probability of $3/4$ the graph G''' does not contain a $k + 2$-path of weight n^7. Otherwise, if G contains a k-path, G''' contains a $k + 2$-path of weight n^7 with a probability of at least $5/8 - 3/8n$.

Let us now assume that Weighted Odd Path has a polynomial conjunctive kernel. Then we can use this kernel on G''' and get a randomized conjunctive OR-distillation for Longest Path, which implies, by Theorem 24 that NP ∈ coNP/poly. Hence, the Weighted Odd Path problem should not have a conjunctive polynomial kernel. Because there is an almost trivial self reduction to its complement it should also have no polynomial disjunctive kernel. ◀

5 Conclusion

Lower bounds were part of parameterized complexity from the very begining in terms of W-completeness. However, it took more than 20 years for the first strong bounds on non-existence of many-one polynomial kernels, which was finally achieved by the concerted efforts of research in classical and parameterized complexity theory. Today, the next big open question are similar lower bounds for polynomial Turing kernels, a task which has so far withstood every solution attempt. In this paper, we went a little beyond many-one kernels by considering disjunctive and conjunctive kernels, which are arguably the most common types of Turing kernels in parameterized algorithmics. Besides the big question about general Turing kernels we pose the following open questions: Can we prove the same lower bounds for Longest Path as we did for Odd Path and Weighted Odd Path? Can we prove the nonexistence of disjunctive kernels for any of the WK[1]-hard problems?

References

1 Daniel Binkele-Raible, Henning Fernau, Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Yngve Villanger. Kernel(s) for problems with no kernel: On out-trees with many leaves. *ACM Trans. Algorithms*, 8(4):38:1–38:19, 2012. doi:10.1145/2344422.2344428.

2 Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009. doi:10.1016/j.jcss.2009.04.001.

3 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discret. Math.*, 28(1):277–305, 2014. doi:10.1137/120880240.

4 Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011. doi:10.1016/j.tcs.2011.04.039.

5 L. Bodlaender, Hans, Erik D. Demaine, Michael R. Fellows, Jiong Guo, Danny Hermelin, Daniel Lokshtanov, Moritz Müller, Venkatesh Raman, Johan van Rooij, and Frances A. Rosamond. Open problems in parameterized and exact computation – IWPEC 2008. Technical Report UU-CS-2008-017, Dept. of Informatics and Computing Sciences, Utrecht University, 2008.

6 Radu Curticapean, Holger Dell, and Thore Husfeldt. Modular Counting of Subgraphs: Matchings, Matching-Splittable Graphs, and Paths. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms (ESA 2021)*, volume 204 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 34:1–34:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2021.34.

7 Marek Cygan, Marcin Pilipczuk, and Michal Pilipczuk. Known algorithms for edge clique cover are probably optimal. *SIAM J. Comput.*, 45(1):67–83, 2016. doi:10.1137/130947076.

8 Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Kernelization lower bounds through colors and ids. *ACM Trans. Algorithms*, 11(2):13:1–13:20, 2014. doi:10.1145/2650261.

9 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

10 Andrew Drucker. New limits to classical and quantum instance compression. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 609–618. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.71.

11 Andrew Drucker. New limits to classical and quantum instance compression. *SIAM J. Comput.*, 44(5):1443–1479, 2015. doi:10.1137/130927115.

12 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing.* Cambridge University Press, 2019. doi:10.1017/9781107415157.

13 Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011. doi:10.1016/j.jcss.2010.06.007.

14 Danny Hermelin, Stefan Kratsch, Karolina Soltys, Magnus Wahlström, and Xi Wu. A completeness theory for polynomial (Turing) kernelization. *Algorithmica*, 71(3):702–730, 2015. doi:10.1007/s00453-014-9910-8.

15 Bart M. P. Jansen. Turing kernelization for finding long paths and cycles in restricted graph classes. *J. Comput. Syst. Sci.*, 85:18–37, 2017. doi:10.1016/j.jcss.2016.10.008.

16 Johannes Köbler and Osamu Watanabe. New collapse consequences of NP having small circuits. In Zoltán Fülöp and Ferenc Gécseg, editors, *Automata, Languages and Programming, 22nd International Colloquium, ICALP95, Szeged, Hungary, July 10-14, 1995, Proceedings*, volume 944 of *Lecture Notes in Computer Science*, pages 196–207. Springer, 1995. doi:10.1007/3-540-60084-1_74.

17 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Comb.*, 7(1):105–113, 1987. doi:10.1007/BF02579206.

18 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity – Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. `doi:10.1007/978-3-642-27875-4`.

19 Christos H. Papadimitriou and Stathis Zachos. Two remarks on the power of counting. In Armin B. Cremers and Hans-Peter Kriegel, editors, *Theoretical Computer Science, 6th GI-Conference, Dortmund, Germany, January 5-7, 1983, Proceedings*, volume 145 of *Lecture Notes in Computer Science*, pages 269–276. Springer, 1983. `doi:10.1007/BFb0009651`.

20 Amit Sahai and Salil P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, 2003. `doi:10.1145/636865.636868`.

21 Uwe Schöning. Probabilistic complexity classes and lowness. *J. Comput. Syst. Sci.*, 39(1):84–100, 1989. `doi:10.1016/0022-0000(89)90020-2`.

22 Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991. `doi:10.1137/0220053`.

23 Leslie G. Valiant. Completeness for parity problems. In Lusheng Wang, editor, *Computing and Combinatorics, 11th Annual International Conference, COCOON 2005, Kunming, China, August 16-29, 2005, Proceedings*, volume 3595 of *Lecture Notes in Computer Science*, pages 1–8. Springer, 2005. `doi:10.1007/11533719_1`.

24 Jouke Witteveen, Ralph Bottesch, and Leen Torenvliet. A hierarchy of polynomial kernels. In Barbara Catania, Rastislav Královic, Jerzy R. Nawrocki, and Giovanni Pighizzini, editors, *SOFSEM 2019: Theory and Practice of Computer Science - 45th International Conference on Current Trends in Theory and Practice of Computer Science, Nový Smokovec, Slovakia, January 27-30, 2019, Proceedings*, volume 11376 of *Lecture Notes in Computer Science*, pages 504–518. Springer, 2019. `doi:10.1007/978-3-030-10801-4_39`.

25 Chee-Keng Yap. Some consequences of non-uniform conditions on uniform classes. *Theor. Comput. Sci.*, 26:287–300, 1983. `doi:10.1016/0304-3975(83)90020-8`.