



UvA-DARE (Digital Academic Repository)

Levelable Sets and the Algebraic Structure of Parameterizations

Witteveen, J.; Torenvliet, L.

Publication date

2018

Document Version

Submitted manuscript

License

Other

[Link to publication](#)

Citation for published version (APA):

Witteveen, J., & Torenvliet, L. (2018). *Levelable Sets and the Algebraic Structure of Parameterizations*. arXiv.org. <https://arxiv.org/abs/1709.04699>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Levelable Sets and the Algebraic Structure of Parameterizations

Jouke Witteveen Leen Torenvliet

April 23, 2018

Abstract

Asking which sets are fixed-parameter tractable for a given parameterization constitutes much of the current research in parameterized complexity theory. This approach faces some of the core difficulties in complexity theory. By focussing instead on the parameterizations that make a given set fixed-parameter tractable, we circumvent these difficulties. We isolate parameterizations as independent measures of complexity and study their underlying algebraic structure. Thus we are able to compare parameterizations, which establishes a hierarchy of complexity that is much stronger than that present in typical parameterized algorithms races. Among other results, we find that no practically fixed-parameter tractable sets have optimal parameterizations.

1 Introduction

Ever since the identification of efficient computability by Cobham and Edmonds (see Goldreich, 2008, Arora and Barak, 2009) and subsequent intractability results [Cook, 1971, Garey and Johnson, 1979], the computational complexity of sets has been focused on the complexity of their hardest instances, or in the case of average computational complexity on the complexity of the majority of their instances. Yet, even very hard sets have simple instances and often lots of them.

Parameterized complexity [Downey and Fellows, 1999, Flum and Grohe, 2006] was originally introduced to deal with this apparent indiscriminate judgment of computational complexity of sets. Instead of looking at the entire set as a single computational object, the complexity of the set is stratified. A parameter function is introduced that singles out one particular dimension of the set, such as the size of the desired solution, and a function of this parameter is factored into the complexity of the computation. Then, for any part of the set where this function yields a constant, membership of the set may be efficiently computable. Sets that can thus be dissected into parts that are decidable in polynomial time, where the degree of the polynomial is invariable across all parts, form the parameterized counterpart of efficient computability, *fixed-parameter tractability*. This division into polynomial time computable parts may be uniform or nonuniform (i.e., with or without a single piecewise polynomial time algorithm) and the function of the parameter in play may be

computable or non-computable. Thus, sets can be divided up into parts that are easy or tractable from one of many viewpoints.

At the same time, many sets have infinite subsets that do not have tractable algorithms. Lynch [1975] introduced *complexity cores* and showed that intractable sets have infinite subsets of which all but finitely many instances resist all efficient programs. In particular no parameter function of a fixed-parameter tractable set can take on the same value infinitely often on a complexity core.

Unfortunately, complexity cores allow finite variations and thus any specific instance can arbitrarily be made part of, or excluded from a complexity core. Therefore, complexity cores do not lead to a useful formalization of the idea of hard instances. Exclusion of an instance from a complexity core is possible by augmenting a program with table lookup for that instance. This shortcoming can be overcome by taking also the sizes of programs into account. Resource bounded *instance complexity*, as introduced by Orponen et al. [1994], does exactly that as a combination of the complexity of the set and the individual complexity of strings [Li and Vitányi, 1997].

Both instance complexity and parameterized complexity can be seen as ways of extending complexity cores into a meaningful notion of the distribution of complexity inside a set. While instance complexity focuses on sets directly, parameterized complexity instead focuses on algorithms, or, more abstractly, on the possible ways to slice up a set into tractable parts. In general, there may be many different algorithms for deciding a particular set and in that regard instance complexity is less ambiguous than parameterized complexity. However, any finite number of algorithms can be combined into a single one that boasts the best behavior of its constituents. Such a compound algorithm would be structured as a big if-then-else case distinction. On the parameterization side, this means that any finite number of parameters can be combined into a single parameter via an easy to compute pairing function. This compound parameter then represents the complexity obtained when taking all of the finitely many algorithms considered into account.

Besides treating parameterizations as measures of complexity, it may be tempting to view a classification of a set as fixed-parameter tractable as information about the computational complexity of that set. However, Witteveen and Torenvliet [2016, Corollary 3.8] have shown that the class of fixed-parameter tractable sets equals that of fixed-parameter decidable sets. Hence classifying a set as fixed-parameter tractable holds very little information about its computational complexity. When imposing additional uniformity constraints, the class of fixed-parameter tractable sets quickly becomes equal to that of the decidable sets. Again, a classification as fixed-parameter tractable is of little value from a computational complexity point of view. We therefore consider it more informative to study parameters and parameterizations first, and sets second.

In studying parameterizations, we should not ask what sets are fixed-parameter tractable with a given parameterization, even though such questions constitute much of the current research in parameterized complexity theory (e.g., Bodlaender et al., 2013). The question “is some set A (uniformly) fixed-parameter tractable with a given parameterization?” is equivalent to “are the parts of A with a fixed parameter value (uniformly) decidable in polynomial time?” If this was in general an easy question, we would not have so much trouble separating \mathbf{P} from other complexity classes. Instead, we should explore the parameterizations with which a given set is fixed-parameter tractable (e.g., Garg and Philip, 2016).

This circumvents the difficulties associated with having to separate complexity classes.

We found that in order to study parameterizations properly, we need a definition of parameterizations that is more general than most common definitions. Often, parameters are represented by integer values associated with instances. Whether a parameter represents the size of a vertex cover in a graph, a bound on the treewidth in a graph, or the number of classrooms in a scheduling problem, it is usually a function that yields a number. It has been recognized that parameters need not be restricted to one dimensional numerical values, yet to our knowledge the true limits of parameters have not been investigated before. In Section 3, we shall identify an axiomatic basis of traits required from parameters and parameterizations. This allows us to build general definitions from the ground up, enforcing just the properties we desire. The resulting definitions are rooted in order theory and reveal a rich algebraic structure that governs parameterizations. Notions from order theory required for our definitions are defined also in Section 3. Because of the role of order theory in this paper, we meet various kinds of orders that (almost) all have their own symbol. There is \leq , which we reserve for natural numbers, \mathbb{N} . In text, we refer to this as “smaller”. We use \preceq both for orders on parameter spaces and for orders on parameterizations, and refer to these orders as “before” and “below” respectively. Finally, sets of parameterizations take center stage in this paper. For such sets, the usual set inclusion ordering \subseteq is in place. When the parameterizations with which a given set is fixed-parameter tractable form a subset of the parameterizations with which some other set is fixed-parameter tractable, we could say that the latter set is “easier” than the former.

Historically, parameterized complexity theory is, at least in part, practically motivated [Downey and Fellows, 1999], and its definitions are arrived at in a pragmatic fashion. In Sections 4 and 5, continuing our foundational journey, we recover the well-known parameterized complexity classes **XP** and **FPT** from classical complexity theory primitives and our general definition of parameterizations. The first of these sections deals with nonuniform parameterized complexity, whereas the second deals with semi-uniform and fully uniform parameterized complexity. Section 2 contains the required background in classical complexity theory, as far as used for building parameterized complexity theory bottom-up. Chiefly, that section includes the definition of a *slice*, which serves as a dual to the complexity *core*.

Much of our parameterized analysis of complexity revolves around collections of parameterizations that put a given set in a parameterized complexity class. Such collections function as an interface to the complexity of the sets that gave rise to them. In Sections 4 and 5 we lift classical levelability and immunity classifications to the parameterized context. The parameterized versions of these classifications can be expressed as properties of the collection of parameterizations that put a given set in a parameterized complexity class. Doing so makes clear that levelability captures a usefulness criterion for parameterized algorithms. Intuitively, a parameterized approach to decision algorithms is only useful when there are infinitely many instances corresponding to any fixed parameter value. Levelability with respect to parameterized complexity classes is the formalization of this intuition. Another complexity aspect captured by the collection of parameterizations that put a given set in a parameterized complexity class is the existence of optimal parameterizations. As it turns out, these

collections of parameterizations form filters in a lattice of parameterizations. A parameterization that is better than (or in terms of this paper *below*) all other parameterizations exists precisely when such a filter is a principal filter. We obtain an almost complete characterization of the sets that have optimal parameterizations with respect to **FPT** and **XP** with varying uniformity constraints. From this characterization we get that uniformly fixed-parameter tractable sets for which a parameterized approach is useful admit no optimal parameterizations. A completion of the characterization and two conjectures regarding the separations of sets based on their complexity are left as open problems.

2 Preliminaries

We assume the reader is familiar with standard notation from complexity theory, as in [Papadimitriou, 2003, Arora and Barak, 2009]. In the current text we use a binary alphabet $\mathbb{2} = \{0, 1\}$. The set of nonempty finite binary sequences, *strings*, is denoted by $\mathbb{2}^+$. Complexity classes are written in boldface. We make use of \mathcal{O} -notation, where n is the free variable in the mathematical expression following \mathcal{O} . For a function f , the complexity class $\mathcal{O}(f(n))$ is the class of sets that can be decided deterministically by a Turing machine with a running time bound in $\mathcal{O}(f(n))$ using some agreed upon alphabet and number of tapes.

Of interest to us are decision procedures that not necessarily decide on membership of every element.

Definition. A Turing machine Φ is a *partial decision procedure* for a set A if we have, for all $x \in \mathbb{2}^+$:

- $\Phi(x) = 1 \implies x \in A$.
- $\Phi(x) = 0 \implies x \notin A$

The machine is said to *decide* the elements of its *domain*:

$$\text{dom}(\Phi) = \{x \mid \Phi(x) \in \{1, 0\}\}.$$

Outside its domain, a partial decision procedure either does not halt at all, or outputs anything other than 0 or 1. Limiting partial decision procedures to a polynomial running time gets us polynomial approximations as in [Ko and Moore, 1981, Balcázar and Schöning, 1985].

Definition. A procedure Φ is a **P**-*approximation* for a set A if it is a partial decision procedure for A that runs in polynomial time.

Note that although a **P**-approximation halts on every input, we do not demand that its domain is $\mathbb{2}^+$. Necessarily, however, the domain of a **P**-approximation is in **P**.

Much of our work revolves around sets that occur as domains of partial decision procedures. Therefore, we shall introduce a name for such sets.

Definition. A set S is a **P**-*slice* for a set A if there is a **P**-approximation Φ for A satisfying $\text{dom}(\Phi) = S$.

The name *slice* is inspired by the use of that term in parameterized complexity theory. Slices function as a dual to the complexity cores of Lynch [1975].

Definition. A set C is a \mathbf{P} -core for a set A if for every \mathbf{P} -slice S for A the intersection $C \cap S$ is finite.

Subsets of \mathbf{P} -cores for a set are also \mathbf{P} -cores for that set, as are finite variations. This complicates thinking of the members of a core for a set as the inherently hard instances of that set. However, for some sets the collection of \mathbf{P} -cores contains a maximal element with respect to inclusion up to finite variation. Such sets are split into an easy part and a hard part.

Theorem 1. *A set A has a maximal (up to finite variations) \mathbf{P} -slice if and only if it has a maximal \mathbf{P} -core.*

Proof. \implies . The complement of a maximal \mathbf{P} -slice for A is a \mathbf{P} -core for A and cannot be infinitely extended. Hence it is a maximal \mathbf{P} -core for A .

\impliedby . We claim that the complement of a maximal \mathbf{P} -core for A is a \mathbf{P} -slice for A . Let S_1, S_2, \dots be an enumeration of the \mathbf{P} -slices for A . Assuming that the complement of a maximal \mathbf{P} -core for A is not a \mathbf{P} -slice for A , then for all j , there are infinitely many elements in this complement outside the \mathbf{P} -slice $\bigcup_{i \leq j} S_i$. Consequently, we would be able to extend the maximal \mathbf{P} -core with infinitely many elements, one for each j , contradicting its maximality. Hence, the complement of a maximal \mathbf{P} -core for A must be a \mathbf{P} -slice for A . \square

Thus the complement of a maximal \mathbf{P} -slice is a maximal \mathbf{P} -core and because \mathbf{P} -slices were necessarily in \mathbf{P} we get the following.

Corollary 2. *A maximal \mathbf{P} -core, if it exists, is in \mathbf{P} .*

Of course, within a maximal \mathbf{P} -core C for a set A , membership of A is hard to decide in the sense that no \mathbf{P} -approximation decides membership of A for infinitely many elements from C .

Complexity cores give us an elegant definition of immunity that is nonstandard, but equivalent to the common definition in the literature based on finite subsets. The equivalence of both definitions was already observed by Balcázar and Schöning [1985] (see also Book et al., 1988).

Definition. A set A is \mathbf{P} -bi-immune if 2^+ is a \mathbf{P} -core for A .

Thus a set is \mathbf{P} -bi-immune if it has the largest \mathbf{P} -core possible. This definition is easily generalized to sets that have a maximal \mathbf{P} -core. In this case the equivalence of our definition to the common definition as a disjoint union was observed by Orponen [1986].

Definition. A set is *almost \mathbf{P} -bi-immune* if it has a maximal \mathbf{P} -core.

Note that by our definitions sets in \mathbf{P} , and finite sets in particular, have maximal \mathbf{P} -cores and are therefore almost \mathbf{P} -bi-immune. This may seem somewhat peculiar, but is in agreement with the definitions used by Orponen [1986] and of no objection in the sequel.

With \mathbf{P} -slices and \mathbf{P} -cores we make no distinction between the members and the nonmembers of a set. For this reason and by Theorem 1 a set is almost \mathbf{P} -bi-immune if it has a slice that cannot be extended by infinitely many members or nonmembers into a larger slice. In [Orponen et al., 1985, 1986] sets of which every slice can be extended by infinitely many members into a larger

slice are called **P-levelable**. We want a more general definition covering sets of which every slice can be extended by infinitely many members or nonmembers. Although it may be tempting to call such sets **P-bi-levelable**, this would more naturally describe sets of which every slice can be extended by both infinitely many members *and* infinitely many nonmembers. The equivalence of the following definition and that based on infinite extensions of slices was observed by Orponen et al. [1985].

Definition. A set is **P-omni-levelable** if it has no maximal **P**-slice.

By Theorem 1, a set is **P-omni-levelable** precisely when it is not almost **P**-bi-immune. Furthermore, every **P-levelable** set is **P-omni-levelable**. As a consequence of the observation of Orponen et al. [1986] that very many (natural) intractable sets are **P-levelable**, we find that there are many **P-omni-levelable** sets.

In all of the above definitions and theorems, **P** can be replaced, given some constant c , by $\mathcal{O}(n^c)$. Likewise, an $\mathcal{O}(n^c)$ -approximation for a set A is a partial decision procedure for A with a running time in $\mathcal{O}(n^c)$.

3 Parameter Spaces and Parameterizations

It is customary in computability theory to consider convergence of computation [Odifreddi, 1992]. Computation that terminates is said to be *convergent*, while computation that does not halt is said to be *divergent*. In computational complexity theory the central theme is convergence of computation within a bounded amount of resource (time, space, ...) usage. Parameterized complexity theory originates from the desire to have a more fine-grained analysis of computational complexity, that is, a more fine-grained analysis of convergence of computation. In the current section we shall adopt this mindset and define parameterizations from the ground up as mathematical structures.

When we are not interested in resource usage, convergence of a Turing machine on a given input is a simple matter of *yes* or *no*. We can represent convergence of a Turing machine Φ by the set $\{x \mid \Phi \text{ converges on } x\}$. In a context where resource usage matters, we are mostly interested in Turing machines that converge on all inputs, but with varying resource needs. It is these resource needs that we want to measure using parameters. With the parameter, we want to express how much of some resource is needed for a given Turing machine to converge on a given input. Thus, we want to represent convergence of a Turing machine Φ by something like the set

$$\{(x, k) \mid \Phi \text{ converges on } x \text{ with at most } k \text{ (additional) resource usage}\}.$$

From the presence of *at most* in the above criteria, it is clear that parameter values have to be taken from ordered sets. These ordered sets we shall call *parameter spaces* and the sets of tuples (x, k) as above we shall call *parameterizations*. While we have used a parameter, k , to measure resource usage, we have neglected to specify how resources should be measured. Indeed, this is not something that can be done unambiguously and therefore whenever we say a parameter measures a resource, we mean that the resource is bounded in a parameter dependent way. The precise dependence (such as, for example, exponential or double exponential) we shall not pay attention to in this paper.

In order to come up with proper definitions for parameter spaces and parameterizations, we consider some of the properties we want them to have. As ours is the study of computational complexity and not that of computability, we require that it is always possible to have sufficient parametric leeway for a computation to converge.

Axiom 1. For every input x , computation converges somewhere (i.e., there is a parameter value with which x occurs in the parameterization).

A second trait we put in place is that convergence should be stable.

Axiom 2. When some x is in a parameterization with a parameter value k , then, for all k' following k (i.e., where k comes before k' in a parameter space specific order), x is also in the parameterization with parameter value k' .

When computations produce results, such as is the case for decision procedures, we want these results to not change for some input x depending on the parameter values with which x is in a parameterization. This is relevant since we shall eventually provide Turing machines access to the value of the parameter. Demanding that the outcome of a computation is the same for all parameter values following one with which some input occurs in a parameterization is not enough to enforce convergence to a unique outcome. It could be that the order on the parameter space defines incomparable (with regard to the order) branches of parameter values. Not only do we want convergence to occur on all these branches for all inputs, we also want some enforcement of the uniqueness of the outcome of a converged computation.

Axiom 3. For every two parameter values k_1, k_2 there is a parameter value k such that k_1 and k_2 come before k in the applicable order.

This axiom internalizes both the idea that resources can always be sufficiently extended and the desire for convergence to only be possible to a single outcome.

Lastly, we want to be able to use parameters constructively.

Axiom 4. Parameter values can be effectively encoded so that they can be used in data structures.

These 4 axioms can be realized using structures known from order theory [Davey and Priestley, 2002].

Definition. A nonempty set Ω , ordered by a reflexive and transitive order \preceq , is a *directed set* if for all $a, b \in \Omega$ there is a $c \in \Omega$ such that we have $a \preceq c$ and $b \preceq c$.

Reflexive and transitive orders that are not necessarily antisymmetric are often referred to as *quasiorders*. A quasiorder that is also antisymmetric is a *partial order*. In this paper, quasiorders are the dominant type of orders so we shall not drag the term *quasi* around and instead speak simply of orders.

Definition. A subset U of a directed set Ω is an *up-set* if, for all $a \in U$ and $b \in \Omega$ with $a \preceq b$, we have $b \in U$.

Using these concepts, we can formulate the most general definitions that meet our axioms.

Definition. A *parameter space* is a directed set that can be encoded into 2^+ . The length of an element k of a parameter space is the length of its encoding and denoted by $|k|$.

We call a parameter space *decidable* if the range of its encoding is. Likely the most prevalent parameter space is the set of natural numbers ordered by the regular *smaller than* order and encoded with ordinary binary representation.

Definition. Given a parameter space Ω , a *parameterization* is a subset η of $2^+ \times \Omega$, such that for every x the set $\{k \mid (x, k) \in \eta\}$ is a nonempty up-set of Ω . For a fixed parameter value k , we denote by η_k the set $\{x \mid (x, k) \in \eta\}$.

Observe how, for a parameterization η and a fixed parameter value k , the set η_k is similar to our initial representation of convergence of a Turing machine Φ by the set $\{x \mid \Phi \text{ converges on } x\}$. We are not so much interested in finite variations on the sets of inputs on which computation converges. Instead, we are mostly interested in parameterizations for which these sets increase in infinitely large steps.

Definition. A parameterization η has *imix* (infinitely many infinite extensions) if for every parameter value k there is a parameter value k' such that the set $\eta_{k'} \setminus \eta_k$ is infinite.

This paper is not the first to define parameterizations, and ideally our definition would be equivalent to earlier definitions. Unfortunately, our axiomatic definitions differ from the definitions of both Downey and Fellows [1999] and Flum and Grohe [2006], which were obtained in a more empirical fashion.

The concept of a parameterization as used by Flum and Grohe [2006] is closest to ours as it can be considered a special case of our definition. For Flum and Grohe [2006], a parameterization is a polynomial time computable function $\kappa : 2^+ \rightarrow \mathbb{N}$ mapping inputs to a unique point of convergence in the parameter space \mathbb{N} . This can be viewed as an instance of *parameterizing by a function* $f : 2^+ \rightarrow \Omega$, where Ω is a parameter space, in general. The parameterization corresponding to such a function is the set $\{(x, k) \mid f(x) \preceq k\}$. Note that, contrary to Flum and Grohe [2006], we have not required that there is a *unique* point of convergence in parameterizations and in particular we have not required that every parameterization arises from parameterizing by some function. The benefit of our more general definition is that it allows for natural operations combining parameterizations into new ones.

With Downey and Fellows [1999], there is no distinction between the sets of which the computational complexity is studied and parameterizations. Their study is restricted to multi-dimensional sets of which the computational complexity is expressed as a function of the values along each of the dimensions of the sets. As a result, the notion of convergence of computation is lost since it is no longer possible to speak of the kind of limit behavior we demanded by our axioms. Where a point of convergence was unique with Flum and Grohe [2006], it is nonexistent with Downey and Fellows [1999]. In practice, however, most of the sets considered by Downey and Fellows [1999] are monotone in the sense that when a parameter value k comes before another parameter value k' , the inputs associated with k form a subset of those associated with k' . While this restores a point of convergence for members of a set in the form of a first parameter value with which it occurs in the parameterized set, it does no good

for nonmembers. We feel that this asymmetry is undesirable and that parameterized complexity theory should be invariant under taking the complement of a set.

Both Downey and Fellows [1999] and Flum and Grohe [2006] work mostly with \mathbb{N} as their parameter space, ordered by its canonical order. Having a common parameter space makes it easy to compare parameterizations [Komusiewicz and Niedermeier, 2012]. While our more general definitions open the door for parameter spaces with a richer structure than that of \mathbb{N} , they also necessitate a special means of comparing parameterizations. It is possible to numerically capture the point of convergence for some input given a parameterization.

Definition. Given a parameterization $\eta \subseteq \mathbb{2}^+ \times \Omega$, the minimization function $\mathbb{2}^+ \rightarrow \mathbb{N}$ of η is defined as

$$\mu_\eta(x) = \min\{|k| \mid k \in \Omega \wedge (x, k) \in \eta\}.$$

This minimization function allows us to compare the behavior of the point of convergence of multiple parameterizations with different underlying parameter spaces. Informally, we want to express how long a parameter value needs to be for convergence in one parameterization to happen on all inputs that converge with some bounded point of convergence in another parameterization.

Definition. Given parameterizations η_1 and η_2 , the gap function $\mathbb{N} \rightarrow \mathbb{N} \cup \{\infty\}$, is defined as

$$\text{gap}_{\eta_1, \eta_2}(n) = \max\{\mu_{\eta_1}(x) \mid x \in \mathbb{2}^+ \wedge \mu_{\eta_2}(x) \leq n\},$$

where we take the maximum of the empty set to be 0.

Comparing parameterizations using this gap function enables us to define a nonuniform and a uniform order on parameterizations. One parameterization is below another, when a bound on the point of convergence for the other parameterization can be turned into a bound on the point of convergence for the one parameterization. Similar orders have been considered by Komusiewicz and Niedermeier [2012] and Fellows et al. [2013].

Definition. A parameterization η_1 is below a parameterization η_2 in the *nonuniform* order \preceq_{nu} if for all $n \in \mathbb{N}$ we have $\text{gap}_{\eta_1, \eta_2}(n) < \infty$.

From a computational standpoint, a bound on the gap between two parameterizations is only useful if it is computable. The uniform variant of the order on parameterizations is therefore mainly of interest when the parameterizations involved are based on decidable parameter spaces and are themselves decidable.

Definition. A parameterization η_1 is below a parameterization η_2 in the *uniform* order \preceq if there is a computable function upper bounding $\text{gap}_{\eta_1, \eta_2}$.

We observe the relationship $\preceq \subset \preceq_{\text{nu}}$ between these orders. These definitions provide structure to the class of parameterizations.

Lemma 3. *Both \preceq_{nu} and \preceq are reflexive and transitive orders on the class of parameterizations.*

Proof. Reflexivity follows from the observation that for every parameterization η the gap function $\text{gap}_{\eta,\eta}$ is bounded by the identity function. For transitivity we need only to remark that the composition of finite bounding functions is again a finite bounding function and that the composition of computable functions is computable. \square

Neither order is antisymmetric and it is convenient to work with the associated partially ordered set of *equivalence classes* instead of with parameterizations directly. Note that every parameterization with *imix* is unequal, in the associated equivalence relation, to any parameterization without *imix*, both using the nonuniform as well as using the uniform order.

Lemma 4. *Given parameterizations η and η' , when η has *imix* but η' does not, then one of $\eta \preceq_{\text{nu}} \eta'$ and $\eta' \preceq_{\text{nu}} \eta$ fails to hold (and similarly for \preceq).*

Proof. The statement for the uniform order follows from that for the nonuniform order by the inclusion $\preceq \subset \preceq_{\text{nu}}$.

Let m be so that no parameter value for η' of length at least m has an infinite extension. Because parameter spaces are directed sets, for every parameterization that does not have *imix* such an m can be found.

Suppose we have $\eta \preceq_{\text{nu}} \eta'$. There is then a parameter value k such that η_k is a superset of all sets $\eta'_{k'}$, when we have $|k'| \leq m$. Since η has *imix*, there exists a parameter value l such that $\eta_l \setminus \eta_k$ is infinite. We find that $\eta' \preceq_{\text{nu}} \eta$ fails because $\text{gap}_{\eta',\eta}(|l|)$ must be infinite.

In case we have $\eta' \preceq_{\text{nu}} \eta$, then if $\eta \preceq_{\text{nu}} \eta'$ were to hold as well, $\text{gap}_{\eta,\eta'}(m)$ would be finite. However, there would then be a parameter value l such that η_l is infinitely larger than any $\eta'_{k'}$, when we have $|k'| \leq m$. As η' does not have *imix*, this would violate the assumed $\eta' \preceq_{\text{nu}} \eta$. \square

For a further investigation of the structure of parameterizations, we need two more definitions from order theory.

Definition. A partially ordered set is a *lattice* if every two elements have a least upper bound and a greatest lower bound. A lattice is *bounded* if it has a greatest element and a least element.

Definition. A nonempty up-set F of a lattice is a *filter* if every greatest lower bound of elements of F is also an element of F . A filter is *principal* if it contains a least element.

Already without any reference to complexity classes we can characterize the partially ordered sets corresponding to our orders on parameterizations.

Lemma 5. *The partially ordered sets on equivalence classes of parameterizations obtained from \preceq_{nu} and \preceq are lattices.*

Proof. We shall show only that the partially ordered sets contain greatest lower bounds. The presence of least upper bounds can be shown in an entirely similar fashion and is therefore omitted. The same proof works for both \preceq_{nu} and \preceq . For simplicity we state the proof for \preceq .

Let $\eta_1 \subseteq \mathcal{2}^+ \times \Omega_1$ and $\eta_2 \subseteq \mathcal{2}^+ \times \Omega_2$ be two arbitrary parameterizations. It suffices to show that there is a parameterization η such that we have $\eta \preceq \eta_1$ and $\eta \preceq \eta_2$, and for every other parameterization η' that realizes this we have

$\eta' \preceq \eta$. From this, the claimed existence of a greatest lower bound of any two equivalence classes of parameterizations follows immediately.

The parameter space underlying our parameterization η will be $\Omega_1 \times \Omega_2$, ordered such that (k_1, k_2) comes before (k'_1, k'_2) if and only if k_1 comes before k'_1 in Ω_1 and k_2 comes before k'_2 in Ω_2 . Observe that this indeed makes Ω a directed set and that an effective encoding can be derived from the effective encodings of Ω_1 and Ω_2 . When Ω_1 and Ω_2 are decidable, Ω is as well. We define our parameterization as

$$\eta = \{(x, (k_1, k_2)) \mid (x, k_1) \in \eta_1 \vee (x, k_2) \in \eta_2\}.$$

Surely, for every x the set $\{(k_1, k_2) \mid (x, (k_1, k_2)) \in \eta\}$ is a nonempty up-set of $\Omega_1 \times \Omega_2$ using the order we defined.

We shall now prove that we have $\eta \preceq \eta_1$. For this, fix some $\omega_2 \in \Omega_2$ and suppose we have some x and n such that $\mu_{\eta_1}(x) \leq n$ holds. Then, there exists a k_1 with $|k_1| \leq n$ and $(x, k_1) \in \eta_1$. Moreover, we have $(x, (k_1, \omega_2)) \in \eta$ and $|k_1, \omega_2|$ is bounded by a computable function of n , depending on the fixed ω_2 and the specifics of the way the encodings of Ω_1 and Ω_2 were combined. By the same token, we obtain that $\eta \preceq \eta_2$ holds as well.

Now suppose we have some other η' with which $\eta' \preceq \eta_1$ and $\eta' \preceq \eta_2$ hold. We need to prove that we have $\eta' \preceq \eta$, or equivalently that $\text{gap}_{\eta', \eta}$ is upper bounded by a computable function. For all n , taking $X_n = \{x \mid \mu_{\eta'}(x) \leq n\}$, we have $\text{gap}_{\eta', \eta}(n) = \max\{\mu_{\eta'}(x) \mid x \in X_n\}$. Since for any reasonable encoding of $\Omega_1 \times \Omega_2$ we get, $X_n \subseteq \{x \mid \mu_{\eta_1}(x) \leq n\} \cup \{x \mid \mu_{\eta_2}(x) \leq n\}$ and a maximum cannot increase when taken on a subset, we find $\text{gap}_{\eta', \eta}(n) \leq \max\{\text{gap}_{\eta', \eta_1}(n), \text{gap}_{\eta', \eta_2}(n)\}$. Because both members of the set on the right hand side are computable as a function of n by assumption, the maximum is computable and we obtain a computable upper bound for $\text{gap}_{\eta', \eta}$. \square

The above proof demonstrates the possibility, under our definitions, of combining parameterizations in such a way that the composite parameterization relates naturally to its constituents. In this case, it was shown that we can construct greatest lower bounds and least upper bounds. Similar constructions get convoluted when we work solely with the natural numbers as our parameter space, or when we require that points of convergence are unique. Note that in the above proof, there are no ‘first’ parameter values with which an input is a member of the constructed parameterization. In order to make the order on the combined parameter space transitive, it was based on a conjunction of orders. On the other hand, the parameterization itself was based on a disjunction of points of convergence.

In the context of orders on parameterizations, we refer to the equivalence classes of parameterizations when speaking simply of parameterizations. As we have seen in Lemma 4, parameterizations with imix will remain distinct from those without when employing this convention. Showcasing this convention, we shall augment Lemma 5 for the nonuniform order.

Lemma 6. *The lattice of parameterizations obtained from \preceq_{nu} is a bounded lattice.*

Proof. First, we construct a least element. Consider $\eta = \mathbb{2}^+ \times \Omega$, for an arbitrary Ω . Let c be the minimum length of a parameter value in Ω . For every η' , we

have, for all n ,

$$\text{gap}_{\eta, \eta'}(n) \leq c,$$

thus we have $\eta \preceq_{\text{nu}} \eta'$.

Next, we construct a greatest element. Consider $\eta \subset \mathbb{2}^+ \times \mathbb{N}$ defined by $\eta = \{(x, n) \mid |x| \leq n\}$. For every η' , the gap with η is the maximum of a finite set,

$$\text{gap}_{\eta', \eta}(n) = \max\{\mu_{\eta'}(x) \mid |x| \leq n\}.$$

Hence, the gap with η is finite and we have $\eta' \preceq_{\text{nu}} \eta$. □

So far, we have looked at parameterizations in isolation. Turning to complexity theory, we are interested in pairs of a subset of $\mathbb{2}^+$ and a parameterization. In its broadest sense, parameterized complexity classes are collections of such pairs. When a set A is in a parameterized complexity class \mathbf{C} with parameterization η , we write $(A, \eta) \in \mathbf{C}$. Parameterized complexity classes can be used to define ordered sets of parameterizations. We shall give names to two of such sets.

Definition. Given a partial order \leq on parameterizations and a parameterized complexity class \mathbf{C} , we denote the partially ordered set of parameterizations that may put a set in \mathbf{C} by

$$\mathcal{L}_{\mathbf{C}}^{\leq} = (\{\eta \mid \exists A : (A, \eta) \in \mathbf{C}\}, \leq).$$

For a fixed set A , the partially ordered set of parameterizations with which A is in \mathbf{C} is denoted by

$$\mathcal{F}_{(A, \mathbf{C})}^{\leq} = (\{\eta \mid (A, \eta) \in \mathbf{C}\}, \leq).$$

4 Nonuniform Parameterized Complexity

Parameterized complexity classes can be derived from nonparameterized complexity classes [Flum and Grohe, 2003]. In this section we shall look at a nonuniform way of doing so and study the structure of the sets of parameterizations associated with the resulting parameterized complexity classes. The complexity class for which we shall define parameterized variants will be \mathbf{P} , although other classes can be used equally well.

Lifting \mathbf{P} to different realms of analysis is not new and several ways of doing so have been studied previously. Using \exists as an operator, the nondeterministic counterpart, \mathbf{NP} , of \mathbf{P} has been obtained as $\exists\mathbf{P}$. In probabilistic complexity theory, the \mathbf{BP} operator was derived from the complexity class \mathbf{BPP} by Schöning [1989] to define various probabilistic complexity classes from their deterministic counterparts. For defining parameterized complexity classes we shall consider the nonuniform \mathbf{X}_{nu} operator, of which uniform versions will be studied in the next section. Note that we indicate nonuniformity with the nu -subscript. In general, we shall use unmodified names of parameterized complexity classes for their uniform variants. In other places (e.g., https://complexityzoo.uwaterloo.ca/Complexity_Zoo:X), special notation is used instead to denote the uniform classes.

Definition. A set A is in \mathbf{XP}_{nu} with parameterization η if for every parameter value k the set η_k is a \mathbf{P} -slice for A .

In the current century, the class \mathbf{XP} has been called *slicewise P*, giving a name to the \mathbf{X} operator [Flum and Grohe, 2003]. However, in earlier work, Downey and Fellows [1999] used “slicewise \mathbf{P} ” to denote a different complexity class (\mathbf{FPT}).

By definition of \mathbf{P} , a set is a \mathbf{P} -slice if there is a constant c such that it is an $\mathcal{O}(n^c)$ -slice. Because the exponent c is in general unbounded for members of \mathbf{XP}_{nu} , the class \mathbf{XP}_{nu} is very permissive. A bound on the exponent is obtained by changing the order of the quantifiers in the definition above.

Definition. A set A is in \mathbf{FPT}_{nu} with parameterization η if there is a constant c such that for every parameter value k the set η_k is an $\mathcal{O}(n^c)$ -slice for A .

Sets in \mathbf{FPT} are said to be *fixed-parameter tractable* as they can be decided in polynomial time with a fixed exponent. The algorithm and precise running time for the slices of a fixed-parameter tractable set vary depending on the parameter. In our nonuniform variant \mathbf{FPT}_{nu} , this dependence is unrestricted.

For every value of c , we can replace \mathbf{P} by $\mathcal{O}(n^c)$ in the definition of \mathbf{XP}_{nu} to get a definition of $\mathbf{X}\mathcal{O}(n^c)_{\text{nu}}$. With this definition in place, we get the equalities

$$\mathbf{XP}_{\text{nu}} = \mathbf{X} \left(\bigcup_c \mathcal{O}(n^c) \right)_{\text{nu}}$$

and

$$\mathbf{FPT}_{\text{nu}} = \bigcup_c \left(\mathbf{X}\mathcal{O}(n^c)_{\text{nu}} \right),$$

further illustrating the reversal of the order of dependence on the parameter and the exponent in the definitions of \mathbf{XP}_{nu} and \mathbf{FPT}_{nu} .

The sets of parameterizations relevant to \mathbf{XP}_{nu} and \mathbf{FPT}_{nu} showcase an algebraic structure similar to that of the set of all parameterizations ordered by the nonuniform order.

Lemma 7. $\mathcal{L}_{\mathbf{XP}_{\text{nu}}}^{\leq \text{nu}}$ and $\mathcal{L}_{\mathbf{FPT}_{\text{nu}}}^{\leq \text{nu}}$ are bounded lattices.

Proof. We shall first show that the partially ordered sets are lattices. In order to do so, we shall only prove that the partially ordered sets contain greatest lower bounds. The presence of least upper bounds can be shown in an entirely similar fashion and is therefore omitted.

Let \mathbf{C} be either \mathbf{XP}_{nu} or \mathbf{FPT}_{nu} and let η_1 and η_2 be parameterizations in $\mathcal{L}_{\mathbf{C}}^{\leq \text{nu}}$. The parameterization η as constructed in the proof of Lemma 5 is a greatest lower bound in the lattice of all parameterizations. Since the union of a set in $\mathcal{O}(n^{c_1})$ and a set in $\mathcal{O}(n^{c_2})$ is in $\mathcal{O}(n^{\max\{c_1, c_2\}})$, the parameterization η puts any trivial set, such as the empty set, in \mathbf{C} and thus η is contained in $\mathcal{L}_{\mathbf{C}}^{\leq \text{nu}}$. Hence $\mathcal{L}_{\mathbf{C}}^{\leq \text{nu}}$ indeed contains all greatest lower bounds.

Next, we shall show that the lattices contain a least and a greatest element. Sets in \mathbf{P} are put in both \mathbf{XP}_{nu} and \mathbf{FPT}_{nu} by the parameterization $2^+ \times \Omega$, where Ω is an arbitrary parameter space. In the proof of Lemma 6 we have seen that this parameterization is below any other. Hence it is a least element in both $\mathcal{L}_{\mathbf{XP}_{\text{nu}}}^{\leq \text{nu}}$ and $\mathcal{L}_{\mathbf{FPT}_{\text{nu}}}^{\leq \text{nu}}$. The greatest element of the lattice of parameterizations, as constructed in the proof of Lemma 6, is present in both $\mathcal{L}_{\mathbf{XP}_{\text{nu}}}^{\leq \text{nu}}$ and $\mathcal{L}_{\mathbf{FPT}_{\text{nu}}}^{\leq \text{nu}}$ and thus a greatest element for both these lattices. \square

Focussing on \mathbf{XP}_{nu} for a moment, we find that the set of parameterizations with which any particular set is put in \mathbf{XP}_{nu} sits nicely inside $\mathcal{L}_{\mathbf{XP}_{\text{nu}}}^{\lessdot{\text{nu}}}$.

Lemma 8. *For every set A , $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$ is a filter in $\mathcal{L}_{\mathbf{XP}_{\text{nu}}}^{\lessdot{\text{nu}}}$.*

Proof. Let A be an arbitrary set. We need to prove that $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$ is a nonempty up-set of $\mathcal{L}_{\mathbf{XP}_{\text{nu}}}^{\lessdot{\text{nu}}}$ that includes all greatest lower bounds of its members.

As $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$ includes all parameterizations η for which, for every parameter value k , the set η_k is finite, it is nonempty. To see that it is an up-set, first observe that for every parameterization η in $\mathcal{L}_{\mathbf{XP}_{\text{nu}}}^{\lessdot{\text{nu}}}$ and every parameter value k in the corresponding parameter space, the set η_k is a \mathbf{P} -slice of some set and hence η_k is in \mathbf{P} . Next, note that whenever there is a parameterization η' in $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$ that is below a parameterization η , then for every parameter value k in the parameter space of η there is a parameter value k' in the parameter space of η' such that we have $\eta_k \subseteq \eta'_{k'}$. Because every subset of a \mathbf{P} -slice that is in \mathbf{P} is itself a \mathbf{P} -slice, η_k is a \mathbf{P} -slice for A . Hence, as the parameter values were chosen arbitrarily, we may conclude that η is in $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$ and thus that $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$ is an up-set of $\mathcal{L}_{\mathbf{XP}_{\text{nu}}}^{\lessdot{\text{nu}}}$.

Let η_1 and η_2 be two parameterizations in $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$. To prove that $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$ includes all greatest lower bounds of its members, it suffices to show that the greatest lower bound η as constructed in the proof of Lemma 5 is in $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$. That this is the case readily follows from the ability to combine any two \mathbf{P} -approximations for A into a \mathbf{P} -approximation for A of which the domain is the union of the domains of its constituents. \square

The orders on parameterizations can be thought of as an inverse ranking of how powerful parameterizations are. When a parameterization is below another, it signifies that the convergence behavior of this parameterization is an improvement over that of the other. This improvement is of a much stronger kind than the improvements made in parameterized algorithms races, where improvements are sought within a parameterization [Komusiewicz and Niedermeier, 2012, Fellows et al., 2013]. In this regard, the best parameterizations in a filter of parameterizations such as that of the previous lemma are those that are below most others. Hence a set A admits an optimal parameterization with respect to \mathbf{XP}_{nu} if the filter $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$ is principal.

Theorem 9. *For any set A , the filter $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$ is principal.*

Proof. Let S_1, S_2, \dots be an enumeration of the \mathbf{P} -slices for A . Consider the parameterization $\eta \subseteq \mathbb{2}^+ \times \mathbb{N}$ given by

$$\eta_k = \bigcup_{i \leq k} S_i.$$

By definition A is in \mathbf{XP}_{nu} with η and by construction η is a least element in $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\lessdot{\text{nu}}}$. \square

We shall call a least element in the filter corresponding to some set a *principal parameterization* for that set. While Theorem 9 shows that all sets have principal parameterizations with respect to \mathbf{XP}_{nu} , this is not a given for arbitrary

parameterized complexity classes. When they exist, principal parameterizations provide insight into some of the computational complexity of a set. Indeed, as a consequence of Theorem 9 there is a one-to-one correspondence between the imix property of a principal parameterization with respect to \mathbf{XP}_{nu} and the levelability of a set.

Corollary 10. *A set is \mathbf{P} -omni-levelable (almost \mathbf{P} -bi-immune) if and only if a principal parameterization for the induced nonuniform filter with respect to \mathbf{XP}_{nu} has (does not have) imix.*

Note that the filter induced by a \mathbf{P} -bi-immune set consists of a single equivalence class of parameterizations, namely that of parameterizations η where for every parameter value k the set η_k is finite.

In case the existence of a principal parameterization is not a given, a statement like Corollary 10 is still valid when there is a parameterization such that it and all parameterizations below it have imix. This motivates the following definition.

Definition. A set A is \mathbf{XP}_{nu} -omni-levelable (almost \mathbf{XP}_{nu} -bi-immune) if there is a parameterization η with which A is in \mathbf{XP}_{nu} such that every parameterization $\eta' \preceq_{\text{nu}} \eta$ with which A is in \mathbf{XP}_{nu} has (does not have) imix.

In these definitions, \mathbf{P} can be replaced by other complexity classes, in particular, for any c , by $\mathcal{O}(n^c)$. However, by the nature of the definition of \mathbf{FPT}_{nu} we should not blindly replace \mathbf{XP}_{nu} by \mathbf{FPT}_{nu} in these definitions, but instead include a dependence on the exponent c .

Definition. A set is \mathbf{FPT}_{nu} -omni-levelable (almost \mathbf{FPT}_{nu} -bi-immune) if there is a constant c_0 such that for all $c \geq c_0$ the set is $\mathbf{X}\mathcal{O}(n^c)_{\text{nu}}$ -omni-levelable (almost $\mathbf{X}\mathcal{O}(n^c)_{\text{nu}}$ -bi-immune).

It is an open problem whether this definition is different from the alternative obtained by replacing \mathbf{XP}_{nu} by \mathbf{FPT}_{nu} in our initial definition of parameterized levelability. If the two notions are the same, then for every \mathbf{FPT}_{nu} -omni-levelable set A there is a constant c such that every \mathbf{P} -slice for A can be infinitely extended to an $\mathcal{O}(n^c)$ -slice for A . We do not expect every set we wish to call \mathbf{FPT}_{nu} -omni-levelable to show this behavior and shall go with the tailored definition of levelability for \mathbf{FPT}_{nu} . This definition works the way we want it to and we feel that keeping the exponent fixed in the analysis reflects the spirit of fixed-parameter tractability.

Our definitions are so that from Theorem 9 and Corollary 10 we get an identification of levelability with respect to \mathbf{P} and levelability with respect to \mathbf{XP}_{nu} .

Corollary 11. *A set is \mathbf{P} -omni-levelable (almost \mathbf{P} -bi-immune) if and only if it is \mathbf{XP}_{nu} -omni-levelable (almost \mathbf{XP}_{nu} -bi-immune).*

The case for levelability with respect to \mathbf{FPT}_{nu} is more complicated. As with \mathbf{XP}_{nu} , we find that the set of parameterizations with which any particular set is put in \mathbf{FPT}_{nu} is a filter.

Lemma 12. *For every set A , $\mathcal{F}_{(A, \mathbf{FPT}_{\text{nu}})}^{\preceq_{\text{nu}}}$ is a filter in $\mathcal{L}_{\mathbf{FPT}_{\text{nu}}}^{\preceq_{\text{nu}}}$.*

Proof. This lemma can be proven similarly to Lemma 8. For the current lemma, however, we need to keep track of the exponent in the running time of the \mathbf{P} -approximations involved in the proof. In general, when multiple \mathbf{P} -approximations are at play, it is possible to compute all of them within a polynomial running time of which the exponent is the maximum of the exponents of the individual polynomial running times. \square

The existence of optimal parameterizations is of interest with respect to $\mathbf{FPT}_{\mathbf{nu}}$ too. Contrary to the case for $\mathbf{XP}_{\mathbf{nu}}$, not every filter of parameterizations with which a set is put in $\mathbf{FPT}_{\mathbf{nu}}$ is principal. For some sets, however, principality of the corresponding filter is easily shown.

Theorem 13. *For any set A that is almost \mathbf{P} -bi-immune, the filter $\mathcal{F}_{(A, \mathbf{FPT}_{\mathbf{nu}})}^{\mathbf{nu}}$ is principal.*

Proof. By definition of being almost \mathbf{P} -bi-immune, A has a maximal \mathbf{P} -slice S . For some constant c , this \mathbf{P} -slice S is also an $\mathcal{O}(n^c)$ -slice. Any parameterization η with which A is in $\mathbf{FPT}_{\mathbf{nu}}$ and for which there is a parameter value k such that η_k equals S is equivalent to any parameterization below it, hence such an η is principal. \square

Of course, a principal parameterization with respect to $\mathbf{FPT}_{\mathbf{nu}}$ for any almost \mathbf{P} -bi-immune set does not have imix and it follows that such sets are also almost $\mathbf{FPT}_{\mathbf{nu}}$ -bi-immune.

Corollary 14. *If a set is almost \mathbf{P} -bi-immune, it is almost $\mathbf{FPT}_{\mathbf{nu}}$ -bi-immune.*

The proof of Theorem 13 inspires alternative characterizations of the almost $\mathbf{FPT}_{\mathbf{nu}}$ -bi-immune and $\mathbf{FPT}_{\mathbf{nu}}$ -omni-levelable sets.

Lemma 15. *A set is $\mathbf{FPT}_{\mathbf{nu}}$ -omni-levelable (almost $\mathbf{FPT}_{\mathbf{nu}}$ -bi-immune) if and only if there is a constant c_0 such that for all $c \geq c_0$ the set is $\mathcal{O}(n^c)$ -omni-levelable (almost $\mathcal{O}(n^c)$ -bi-immune).*

Proof. For any c , we can replace \mathbf{P} by $\mathcal{O}(n^c)$ in Theorem 9. The current lemma then follows from the associated variants of Corollary 11. \square

While we have seen that levelability with respect to $\mathbf{XP}_{\mathbf{nu}}$ is identical to levelability with respect to \mathbf{P} , the situation with respect to $\mathbf{FPT}_{\mathbf{nu}}$ is different. We shall provide a constructive proof of the fact that the converse of Corollary 14 does not hold. The set we are about to construct has the remarkable property that it is almost $\mathcal{O}(n^c)$ -bi-immune for infinitely many values of c , yet each of the bi-immune parts is still decidable in polynomial time. Conceptually, we show that it is possible to diagonalize against polynomial time machines in polynomial time.

Theorem 16. *There are \mathbf{P} -omni-levelable sets that are not $\mathbf{FPT}_{\mathbf{nu}}$ -omni-levelable.*

Proof. We shall prove the theorem by constructing a \mathbf{P} -omni-levelable set A for which there are infinitely many $c \in \mathbb{N}$ such that A is almost $\mathcal{O}(n^c)$ -bi-immune. For this, let ϕ_1, ϕ_2, \dots be an effective enumeration of all partial decision procedures and denote by $\phi_i^{\downarrow c}$ the i th partial decision procedure restricted to running

time n^c so that for all i and all $c \leq d$ we have $\text{dom } \phi_i^{\downarrow c} \subseteq \phi_i^{\downarrow d}$. Additionally, let $\langle c, x \rangle$ be the result of a pairing function such as Cantor's.

Consider the following recursive procedure for deciding a set A . On input $\langle c, x \rangle$, the procedure takes the following steps.

1. *Determine a set I of procedures consistent with an initial segment of A :*
 - (a) Set $I = \{1, 2, \dots, |\langle c, x \rangle|\}$.
 - (b) For d in $\{1, 2, \dots, c\}$ and y in $\{1, 2, \dots, |\langle c, x \rangle|\}$ do:
Set $I = \{i \mid i \in I \wedge \phi_i^{\downarrow 2c}(\langle d, y \rangle) = A(\langle d, y \rangle)\}$.
2. *Try to make a procedure from I inconsistent with A :*
 - (a) For i in I do:
If $\phi_i^{\downarrow 2c}(\langle c, x \rangle) \in \{0, 1\}$ then return $1 - \phi_i^{\downarrow 2c}(\langle c, x \rangle)$.
 - (b) return 0. *(arbitrary)*

The first stage of this procedure performs at most $c|\langle c, x \rangle|^2$ simulations of computations, each with a running time of n^{2c} where the input length n is at most the length of $\langle c, |\langle c, x \rangle| \rangle$. Besides these simulations, this procedure needs access to a segment of A to test against. This segment can be computed recursively, with a recursion depth bounded by the iterated logarithm of x . By using dynamic programming the time required to compute the segment is insignificant with respect to the total running time of the entire procedure.

The second stage of the procedure requires the simulation of at most $|\langle c, x \rangle|$ computations, each with a running time of n^{2c} , where we have $n = |\langle c, x \rangle|$. Using efficient simulation [Arora and Barak, 2009], this puts the running time of the second stage in $\mathcal{O}(n^{2c+2})$. Note that the running time of the procedure is thus not polynomial in $|\langle c, x \rangle|$, as c appears in the exponent and is not independent of the input.

For any fixed c , the set $\{\langle d, x \rangle \mid d \leq c \wedge x \in \mathbb{2}^+\}$ is a $\mathcal{O}(n^{2c+2})$ -slice of A . It is not a maximal \mathbf{P} -slice, as for larger values of c infinitely many elements are introduced in the corresponding sets. However, we claim that it is a maximal $\mathcal{O}(n^{2c+2})$ -slice and thus that A is \mathbf{P} -omni-levelable yet not \mathbf{FPT}_{nu} -omni-levelable. Suppose towards a contradiction that there is an infinite $\mathcal{O}(n^{2c+2})$ slice $S = \{\langle d, x \rangle \mid d > c \wedge x \in \mathbb{2}^+\}$ for A that is the domain of a partial decision procedure that occurs in our enumeration with index i . When $|\langle d, x \rangle|$ outgrows i in the first stage of our procedure for deciding A will include i in I . Because there are only finitely many values for d and x such that $|\langle d, x \rangle|$ is smaller than i , we may assume that I contains i at the start of the second stage of our procedure. Now for every element of S , either the second stage invalidates i as the index of a $\mathcal{O}(n^{2c+2})$ -slice for A , or an index smaller than i is removed from I for all subsequent values of d and x . The latter of these possibilities can happen at most i times, so, since we assumed that S is infinite, eventually i must be invalidated. This contradicts our assumption that i was the index of a partial decision procedure for S with a running time in $\mathcal{O}(n^{2c+2})$. Note that our time bounds were chosen as they are so that already for $d = c + 1$ we find $2d \geq 2c + 2$. \square

Note that although the set constructed in the above proof is not \mathbf{FPT}_{nu} -omni-levelable this does not mean that it is necessarily almost \mathbf{FPT}_{nu} -bi-

immune. It could be that the set is neither \mathbf{FPT}_{nu} -omni-levelable nor almost \mathbf{FPT}_{nu} -bi-immune. We shall now turn to the existence of \mathbf{FPT}_{nu} -omni-levelable sets. Our notion of a *reduction* is that of a membership preserving polynomial time computable function, in other words, that of a Karp reduction.

Theorem 17. *Every set outside \mathbf{P} from which there is a linearly length increasing reduction to itself is \mathbf{FPT}_{nu} -omni-levelable.*

Proof. Let A be a set outside \mathbf{P} and f a linearly length increasing reduction from A to itself. Suppose that A is not \mathbf{FPT}_{nu} -omni-levelable and has a maximal $\mathcal{O}(n^c)$ -slice S , with c high enough for f to be computable in time $\mathcal{O}(n^{c-1})$. The sets

$$\begin{aligned} S' &= \{x \mid x \notin S \wedge f(x) \in S\}, \\ S_x &= \{x, f(x), f(f(x)), \dots\} \end{aligned}$$

are, by nature of f , also $\mathcal{O}(n^c)$ -slices for A . For S' this requires the linear bound on the length of the output of f , where for S_x this requires that f is length increasing. Furthermore, S' satisfies $S \cap S' = \emptyset$.

By the assumed maximality of S , for every x there are only finitely many elements in the set $S \setminus S_x$. However, since A is not in \mathbf{P} , there are infinitely many x outside S and for each of these the set S_x intersects S' . Hence S' is infinite, contradicting the maximality of S . \square

The existence of \mathbf{FPT}_{nu} -omni-levelable sets now follows from the existence of sets outside \mathbf{P} that have a linearly length increasing reduction to itself.

Lemma 18. *There are sets outside \mathbf{P} that have a linearly length increasing reduction to itself.*

Proof. Let X be a set outside \mathbf{P} and consider its cylindrification

$$A = \{\langle x, y \rangle \mid x \in X, y \in \mathbb{N}\}.$$

Note that A too is not in \mathbf{P} . For most reasonable choices of pairing functions, the function f defined by

$$f(\langle x, y \rangle) = \langle x, 2 \cdot y \rangle$$

is a linearly length increasing reduction from A to itself. \square

It should be noted that certain padding functions give rise to linearly length increasing reductions. There are padding functions, pad , for which there exists a constant c such that for all x, y we have

$$\frac{1}{c} \cdot (|x| + |y|) \leq |\text{pad}(x, y)| \leq c \cdot (|x| + |y|).$$

Padding functions that meet this enhanced honesty criterion can be turned into linearly length increasing reductions by mapping x to $\text{pad}(x, 0^{c \cdot |x|})$.

It was shown [Young, 1983] that every honest, one-one, polynomially computable function is the productive function for the complement of some k -creative set. In particular this means that there are k -creative sets where the influence of the associated productive function on the length of its input is

linear. Such functions can again be turned into linearly length increasing reductions and hence, assuming $\mathbf{P} \neq \mathbf{NP}$, there are also k -creative sets that are \mathbf{FPT}_{nu} -omni-levelable.

Starting from Lemma 15 we can augment Theorem 13 and thus answer the principality question for all sets that are not \mathbf{FPT}_{nu} -omni-levelable.

Theorem 19. *For any set A that is \mathbf{P} -omni-levelable and not \mathbf{FPT}_{nu} -omni-levelable, the filter $\mathcal{F}_{(A, \mathbf{FPT}_{\text{nu}})}^{\text{nu}}$ is nonprincipal.*

Proof. In case A is not \mathbf{FPT}_{nu} -omni-levelable, there is an unbounded set of values c such that A has a maximal $\mathcal{O}(n^c)$ -slice. When such a set A is \mathbf{P} -omni-levelable, then for every $\mathcal{O}(n^c)$ -slice S for A that is maximal, there is a constant d such that A has a maximal $\mathcal{O}(n^d)$ -slice that is infinitely larger than S . Hence $\mathcal{F}_{(A, \mathbf{FPT}_{\text{nu}})}^{\text{nu}}$ cannot be principal. \square

For almost \mathbf{FPT}_{nu} -bi-immune sets, the principality of their filters with respect to \mathbf{FPT}_{nu} is still an open problem. Next to identifying sets that admit optimal parameterizations, we can study to what degree a filter of parameterizations is unique to a set. The filter of parameterizations with respect to a parameterized complexity class is then taken as a representation of the distribution of complexity inside a set. This approach is a continuation of an idea by Orponen [1986] who represented the complexity characteristics of a set by the filter of its complexity cores. Where this idea was shown fruitless when using proper cores, our parameterized setting is promising. For sets A, B , let $A \triangle B$ denote the symmetric difference $(A \setminus B) \cup (B \setminus A)$.

Theorem 20. *For any set X in \mathbf{P} and any set A we have*

$$\mathcal{F}_{(A, \mathbf{FPT}_{\text{nu}})}^{\text{nu}} = \mathcal{F}_{(A \triangle X, \mathbf{FPT}_{\text{nu}})}^{\text{nu}}.$$

Proof. This follows from the fact that given X , any slice for A can be turned into a slice for $A \triangle X$ and vice versa. \square

Intuitively, the above theorem states that taking the symmetric difference with an easy set does not alter the distribution of complexity. Similarly, we find that the symmetric difference of two sets with the same distribution of complexity is easier than either of the initial sets.

Theorem 21. *For any two sets A, B satisfying $\mathcal{F}_{(A, \mathbf{FPT}_{\text{nu}})}^{\text{nu}} = \mathcal{F}_{(B, \mathbf{FPT}_{\text{nu}})}^{\text{nu}}$ we have*

$$\mathcal{F}_{(A, \mathbf{FPT}_{\text{nu}})}^{\text{nu}} \subseteq \mathcal{F}_{(A \triangle B, \mathbf{FPT}_{\text{nu}})}^{\text{nu}}.$$

Proof. This follows from the fact that any slice for A is also a slice for B and can thus be turned into a slice for $A \triangle B$. \square

Even though Theorem 21 asserts that the symmetric difference of two sets that share all their parameterizations is easier than either of the sets, it does not guarantee that this symmetric difference is in \mathbf{P} . If this would be the case, a filter with respect to \mathbf{FPT}_{nu} would uniquely define a set up to variations in \mathbf{P} . Of comparable flavor is the Berman–Hartmanis conjecture [Young, 1983, Li and Vitányi, 1997], which asserts that completeness for \mathbf{NP} uniquely defines a set up to isomorphisms computable in polynomial time. We conjecture that indeed the

filter of the symmetric difference collapses to that of a set in \mathbf{P} . In other words, we conjecture that for all A, B , whenever we have $\mathcal{F}_{(A, \mathbf{FPT}_{\text{nu}})}^{\mathcal{S}_{\text{nu}}} = \mathcal{F}_{(B, \mathbf{FPT}_{\text{nu}})}^{\mathcal{S}_{\text{nu}}}$ there is some X in \mathbf{P} such that we have $B = A \triangle X$. Here, we should keep in mind that taking the symmetric difference with some set is an involution and we have $A \triangle (A \triangle X) = X$. Because all sets in \mathbf{P} have the same associated filter, our conjecture can be expressed elegantly as follows.

Conjecture 1. *For any two sets A, B we have*

$$\mathcal{F}_{(A, \mathbf{FPT}_{\text{nu}})}^{\mathcal{S}_{\text{nu}}} = \mathcal{F}_{(B, \mathbf{FPT}_{\text{nu}})}^{\mathcal{S}_{\text{nu}}} \iff \mathcal{F}_{(A \triangle B, \mathbf{FPT}_{\text{nu}})}^{\mathcal{S}_{\text{nu}}} = \mathcal{F}_{(\emptyset, \mathbf{FPT}_{\text{nu}})}^{\mathcal{S}_{\text{nu}}}.$$

If true, a separation result would follow. Namely, for any two sets A, B of which the symmetric difference is outside \mathbf{P} , there would exist a set in \mathbf{P} separating A and B in the sense that the separating set contains a \mathbf{P} -core for precisely one of the two.

5 Uniform Parameterized Complexity

The classes \mathbf{XP}_{nu} and \mathbf{FPT}_{nu} were nonuniform in two ways. Firstly, we did not require the parameter dependence of the running time of the \mathbf{P} -approximations of slices to have a uniformly computable upper bound as a function of the parameter value to which the slices belong. Put differently, in the equivalent characterizations of the complexity classes on page 13, we observe that the \mathbf{X}_{nu} operator is used on (a union of) $\mathcal{O}(n^c)$ complexity classes without restrictions to the constant hidden in the \mathcal{O} -notation. Secondly, we did not require the existence of a procedure to instantiate the \mathbf{P} -approximations of the slices from the parameter values to which the slices belong.

For the majority of this section, we shall consider two additional complexity classes for fixed-parameter tractability, both exhibiting more uniformity than \mathbf{FPT}_{nu} . The first of them is a semi-uniform variant, which is known as *uniform FPT* by Downey and Fellows [1999]. In this variant the \mathbf{P} -approximations are required to form a uniform collection.

Definition. A set A is in \mathbf{FPT}_{u} with parameterization η if there is a constant c and a Turing machine Ψ taking two inputs, such that for every parameter value k the partial application of Ψ to k yields an $\mathcal{O}(n^c)$ -approximation, $\Psi(k, \cdot)$, for A with domain η_k .

The second of our classes is a fully uniform variant, for which we use no subscript. In [Downey and Fellows, 1999], this class is known as *strongly uniform FPT*.

Definition. A set A is in \mathbf{FPT} with parameterization η if it is in \mathbf{FPT}_{u} with constant c and Turing machine Ψ and there is a computable function f such that the partial application of Ψ to k has a running time bounded by $f(k) \cdot n^c$.

The parameterizations that can put sets in \mathbf{FPT}_{u} on \mathbf{FPT} can be identified. This will help us characterize the structure of $\mathcal{L}_{\mathbf{FPT}_{\text{u}}}^{\mathcal{S}_{\text{nu}}}$ and $\mathcal{L}_{\mathbf{FPT}}$.

Lemma 22. *A parameterization η is in $\mathcal{L}_{\mathbf{FPT}_{\text{u}}}^{\mathcal{S}_{\text{nu}}}$ if and only if there is a constant c such that, uniformly in k , membership of any x in η_k can be decided by a decision procedure with a running time in $\mathcal{O}(n^c)$.*

Moreover, η is in $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$ if and only if there is additionally a computable function f such that membership of any (x, k) in η can be decided by a decision procedure with a running time bounded by $f(k) \cdot |x|^c$.

Proof. When a set is in \mathbf{FPT}_u or \mathbf{FPT} with some parameterization, the Turing machine witnessing such a membership can easily be modified to decide membership of the parameterization. Conversely, a trivial set such as the empty set is put in \mathbf{FPT}_u or \mathbf{FPT} by every parameterization that meets the respective running time bound. \square

Thus the parameterizations relevant for semi-uniform and strongly uniform fixed-parameter tractability are decidable in a time bound that is polynomial in the length of their first component. This makes that convergence of computation with a given parameter value becomes less of a *promise* for the input, as it was for our nonuniform classes, and more of a *property* of the input. Moreover, it follows from the previous lemma that the minimization function μ_η of a parameterization η taken from $\mathcal{L}_{\mathbf{FPT}_u}^{\preceq_{\text{nu}}}$ or $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$ is computable. Having identified the parameterizations relevant for semi-uniform and strongly uniform fixed-parameter tractability, we explore the algebraic structure they form.

Lemma 23. $\mathcal{L}_{\mathbf{FPT}_u}^{\preceq_{\text{nu}}}$ and $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$ are bounded lattices.

Proof. We shall phrase the proof as a proof for $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$. Only the first part is relevant for $\mathcal{L}_{\mathbf{FPT}_u}^{\preceq_{\text{nu}}}$ as the rest of the proof for this partially ordered set is identical to Lemma 7.

That $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$ is a lattice follows from Lemma 22. The greatest lower bound of two decidable parameterizations as constructed in the proof of Lemma 5 is again decidable, hence $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$ contains greatest lower bounds. As before, the remainder of proving that $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$ is a lattice is now routine.

Similar to Lemma 7, we observe that $\mathbb{2}^+ \times \Omega$ is a least element in $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$ for an arbitrary parameter space Ω . The case for a greatest element is more subtle. In $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$, a parameterization is below another not simply when there is a finite bound on the gap function of the two, but only when there is such a bound that is computable. For this reason, given the parameterization $\eta = \{(x, k) \mid |x| \leq k\}$ of Lemma 6, the argument that for all slices in η are finite is not sufficient to show that η is a greatest element. Surely, it follows from Lemma 22 that η is in $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$. That it is indeed also a greatest element requires that, uniformly in k , we can compute the number of elements in η_k . Because of this, the maximum

$$\text{gap}_{\eta', \eta}(n) = \max\{\mu_{\eta'}(x) \mid |x| \leq n\}$$

is computable and η is a greatest element in $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$. \square

While all parameterizations in $\mathcal{L}_{\mathbf{FPT}_u}^{\preceq_{\text{nu}}}$ of which all slices are finite are \preceq_{nu} -equivalent, not all parameterizations in $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$ of which all slices are finite are \preceq -equivalent. In particular, the number of elements in a finite set in \mathbf{P} need not be computable. It is for this reason that the existence of a greatest element in $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$ requires a more delicate proof than the existence of a greatest element in $\mathcal{L}_{\mathbf{FPT}_u}^{\preceq_{\text{nu}}}$.

We have seen that the minimization function for parameterizations that put a set in \mathbf{FPT}_u or \mathbf{FPT} are computable. As a result of this, we find that the

sets that can be put in \mathbf{FPT}_u or \mathbf{FPT} are all decidable. Indeed, the sets that can be put in \mathbf{FPT}_u or \mathbf{FPT} are *precisely* the decidable sets.

Lemma 24. *For every decidable set A , $\mathcal{F}_{(A, \mathbf{FPT}_u)}^{\preceq \text{nu}}$ is a filter in $\mathcal{L}_{\mathbf{FPT}_u}^{\preceq \text{nu}}$ and $\mathcal{F}_{(A, \mathbf{FPT})}^{\preceq}$ is a filter in $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$.*

Proof. Let A be any decidable set. We shall first prove that $\mathcal{F}_{(A, \mathbf{FPT}_u)}^{\preceq \text{nu}}$ and $\mathcal{F}_{(A, \mathbf{FPT})}^{\preceq}$ are nonempty. For this, let Φ be a decision procedure for A and consider the parameterization

$$\eta = \{(x, k) \mid \Phi \text{ decides membership of } x \text{ in at most } k \text{ steps}\}.$$

Immediately, we see that η is in both $\mathcal{F}_{(A, \mathbf{FPT}_u)}^{\preceq \text{nu}}$ and $\mathcal{F}_{(A, \mathbf{FPT})}^{\preceq}$. The running times of the corresponding approximations for A do not depend on the length of the input at all.

Next, we shall prove that $\mathcal{F}_{(A, \mathbf{FPT})}^{\preceq}$ is an up-set of $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$. The case for $\mathcal{F}_{(A, \mathbf{FPT}_u)}^{\preceq \text{nu}}$ is simpler. Suppose we have a parameterization η in $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$ and a parameterization $\eta' \subseteq 2^+ \times \Omega'$ in $\mathcal{F}_{(A, \mathbf{FPT})}^{\preceq}$ that is below η . We need to show that A is in \mathbf{FPT} with parameterization η as well. Let Ψ be the Turing machine witnessing that A is in \mathbf{FPT} with η' , and consider the following procedure on input (x, k) .

1. If (x, k) is not in η , return nothing.
2. For ω' in (an unbounded subset of) Ω' do:
If $\Psi(x, \omega') \in \{0, 1\}$ then return $\Psi(x, \omega')$.

For precisely those (x, k) that are in η , this procedure decides membership of x in A . The time it needs to do so depends on where in the enumeration of (an unbounded subset of) Ω' we encounter an ω' such that (x, ω') is in η' . For each of the values in the enumeration up to and including this ω' , the time needed for step 2 is, for some c depending on Ψ , in $\mathcal{O}(n^c)$ with a hidden constant depending on the parameter value. In the case of \mathbf{FPT} the dependence on the parameter value is upper bounded by some computable function. Because we are considering the uniform order on parameterizations, \preceq , it is furthermore possible to compute an upper bound on the set of parameter values considered in step 2 of the procedure. Therefore, the running time of the above procedure is so that it witnesses that A is in \mathbf{FPT} with parameterization η and we have shown that $\mathcal{F}_{(A, \mathbf{FPT})}^{\preceq}$ is an up-set of $\mathcal{L}_{\mathbf{FPT}}^{\preceq}$.

It remains to show that $\mathcal{F}_{(A, \mathbf{FPT}_u)}^{\preceq \text{nu}}$ and $\mathcal{F}_{(A, \mathbf{FPT})}^{\preceq}$ include all greatest lower bounds of pairs of their respective members. This can be done in the same way as it was done in the proof of Lemma 12 as that proof is constructive insofar concerned with greatest lower bounds. \square

This lemma, together with Lemma 8 and Lemma 12 are possible because of the way we have defined the orders on parameterizations. In other words, these lemmas justify the definitions of the nonuniform and uniform order on parameterizations.

For the semi-uniform filter, we have an analogue of Theorem 13.

Theorem 25. *For any set A that is almost \mathbf{P} -bi-immune, the filter $\mathcal{F}_{(A, \mathbf{FPT}_u)}^{\lessdot{\text{nu}}}$ is principal in $\mathcal{L}_{\mathbf{FPT}_u}^{\lessdot{\text{nu}}}$.*

Proof. Identical to the proof of Theorem 13. □

For being principal in $\mathcal{L}_{\mathbf{FPT}}^{\lessdot{\text{nu}}}$, we would need an additional computable upper bound to the gap between parameterizations. Still, a parameterization with imix would be below one without it even in the nonuniform order. That is, in accordance with Lemma 4, the nonuniform order on parameterizations distinguishes parameterizations with imix from those without. Therefore, the corollary to Theorem 13 works for the uniform case too. The definitions of levelability and immunity with respect to semi-uniform and uniform parameterized complexity classes are the same as those with respect to nonuniform ones, with the sole difference that for uniform complexity classes they now refer to the uniform order on parameterizations.

Corollary 26. *If a set is almost \mathbf{P} -bi-immune, it is almost \mathbf{FPT}_u -bi-immune and almost \mathbf{FPT} -bi-immune.*

Of course, we *can* show that filters with respect to \mathbf{FPT} are principal for sets in \mathbf{P} .

Theorem 27. *For any set A that is in \mathbf{P} , the filter $\mathcal{F}_{(A, \mathbf{FPT})}^{\lessdot{\text{nu}}}$ is principal in $\mathcal{L}_{\mathbf{FPT}}^{\lessdot{\text{nu}}}$.*

Proof. Regardless of the parameter space Ω , the full parameterization $2^+ \times \Omega$ is one with which A is in \mathbf{FPT} . Since the class of this parameterization is the least element of the lattice of parameterizations, the filter must be principal. □

Whether or not there exists a set outside \mathbf{P} for which the filter with respect to \mathbf{FPT} is principal is an open problem. We shall see that any such set is necessarily almost \mathbf{P} -bi-immune. As before, the converse of Corollary 26 does not hold and not all \mathbf{P} -omni-levelable sets are \mathbf{FPT}_u -omni-levelable or \mathbf{FPT} -omni-levelable.

Theorem 28. *There are \mathbf{P} -omni-levelable sets that are not \mathbf{FPT}_u -omni-levelable and not \mathbf{FPT} -omni-levelable.*

Proof. The proof of Theorem 16 is constructive and applies in the semi-uniform and uniform situations too. □

The existence of \mathbf{FPT}_u -omni-levelable sets and \mathbf{FPT} -omni-levelable sets can be shown similarly to how the existence of \mathbf{FPT}_{nu} -omni-levelable sets was shown.

Theorem 29. *Every set outside \mathbf{P} from which there is a linearly length increasing reduction to itself is \mathbf{FPT}_u -omni-levelable and \mathbf{FPT} -omni-levelable.*

Proof. The proof of Theorem 17 is constructive and applies in the semi-uniform and uniform situations too. □

Thus, by Lemma 18, there exist \mathbf{FPT}_u -omni-levelable sets and \mathbf{FPT} -omni-levelable sets. Again, it could well be that some sets are neither almost \mathbf{FPT}_u -bi-immune nor \mathbf{FPT}_u -omni-levelable (and similarly for \mathbf{FPT}). By Corollary 14, every almost \mathbf{P} -bi-immune set is almost \mathbf{FPT}_u -bi-immune and almost \mathbf{FPT} -bi-immune. Conversely, because every \mathbf{P} -slice for a decidable set can be made part of a parameterization with which the set is fixed-parameter tractable, being \mathbf{FPT}_u -omni-levelable or \mathbf{FPT} -omni-levelable implies being \mathbf{P} -omni-levelable. This can be graphically depicted as in Figure 1. The same holds for \mathbf{FPT}_u .

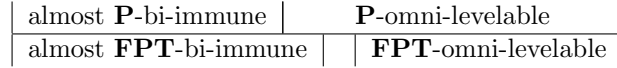


Figure 1: The universe of sets, represented by the horizontal line, can be divided according to levelability with respect to \mathbf{P} or with respect to \mathbf{FPT} . Note that \mathbf{P} is part of the almost \mathbf{P} -bi-immune sets.

As we did for \mathbf{FPT}_{nu} , we can answer the principality question for all sets that are not \mathbf{FPT}_u -omni-levelable or \mathbf{FPT} -omni-levelable.

Theorem 30. *For any set A that is \mathbf{P} -omni-levelable and not \mathbf{FPT}_u -omni-levelable (\mathbf{FPT} -omni-levelable), the filter $\mathcal{F}_{(A, \mathbf{FPT}_u)}^{\lessdot{nu}}$ ($\mathcal{F}_{(A, \mathbf{FPT})}^{\lessdot{}}$) is nonprincipal.*

Proof. Identical to the proof of Theorem 19. □

In contrast to the nonuniform setting we can extend Theorem 30 proving that the filters produced by \mathbf{FPT}_u -omni-levelable sets and \mathbf{FPT} -omni-levelable sets are also nonprincipal.

Theorem 31. *For any set A that is not almost \mathbf{FPT}_u -bi-immune (almost \mathbf{FPT} -bi-immune), the filter $\mathcal{F}_{(A, \mathbf{FPT}_u)}^{\lessdot{nu}}$ ($\mathcal{F}_{(A, \mathbf{FPT})}^{\lessdot{}}$) is nonprincipal.*

Proof. We present a proof by contradiction for the uniform setting. The semi-uniform setting is subsumed in this proof. Assuming $\mathcal{F}_{(A, \mathbf{FPT})}^{\lessdot{}}$ is the principal filter induced by a parameterization $\eta \subseteq \mathbb{2}^+ \times \Omega$ we construct a parameterization η' such that we have $(A, \eta') \in \mathbf{FPT}$ and $\eta \not\leq \eta'$.

Let c and Ψ be the constant and the Turing machine witnessing that A is in \mathbf{FPT} with parameterization η and let $\omega_1, \omega_2, \dots$ be an effective enumeration of (an unbounded subset of) Ω . Denote by ψ_{ω_i} the partial application of Ψ to ω_i . Thus, ψ_{ω_i} is a $\mathcal{O}(n^c)$ -approximation for A with domain η_{ω_i} . Consider the following partial decision procedure for A , uniformly defined for all $k \in \mathbb{N}$. On input x , the procedure takes the following steps.

1. Spend $k \cdot |x|^{c+2}$ time computing the values $\psi_{\omega_1}(x), \psi_{\omega_2}(x), \dots$
2. If any computed value is either 1 or 0, return it, otherwise return nothing.

This partial decision procedure is a $\mathcal{O}(n^{c+2})$ -approximation for A and moreover it defines a parameterization $\eta' \subseteq \mathbb{2}^+ \times \mathbb{N}$ with which A is in \mathbf{FPT} when we interpret k as the parameter value.

Because the exponent $c + 2$ is more than necessary to simulate any fixed number of $\mathcal{O}(n^c)$ -approximations for A , we are able to compute an increasing

number of values $\psi_{\omega_1}, \psi_{\omega_2}, \dots$ with increasing $|x|$ for any constant value of k . Because η necessarily has imix, this enables us to decide membership for elements from arbitrary high slices in η . Hence the gap between η and η' is always infinite, proving $\eta \not\stackrel{\text{nu}}{\approx} \eta'$ and also $\eta \not\approx \eta'$. \square

Combining the above theorem with Theorem 25 and Theorem 30, we get a complete picture of which principal filters in $\mathcal{L}_{\mathbf{FP}\mathbf{T}_u}^{\stackrel{\text{nu}}{\approx}}$ can occur as filters of the form $\mathcal{F}_{(A, \mathbf{FP}\mathbf{T}_u)}^{\stackrel{\text{nu}}{\approx}}$ with some set A .

Corollary 32. *For any set A , the filter $\mathcal{F}_{(A, \mathbf{FP}\mathbf{T}_u)}^{\stackrel{\text{nu}}{\approx}}$ is principal if and only if A is almost \mathbf{P} -bi-immune.*

For the uniform filters, we lack a uniform equivalent to Theorem 25. However, Theorem 31 holds that whenever $\mathcal{F}_{(A, \mathbf{FP}\mathbf{T})}^{\stackrel{\text{nu}}{\approx}}$ is principal, a principal parameterization of that filter does not have imix. Conversely, every parameterization that has imix does not occur as a principal parameterization in any filter of the form $\mathcal{F}_{(A, \mathbf{FP}\mathbf{T})}^{\stackrel{\text{nu}}{\approx}}$. As parameterizations with imix are the most interesting from an applications point of view, we find that sets for which parameterized algorithms are attractive do not admit optimal parameterizations.

The proof of Theorem 31 has a clear kinship to that of the time hierarchy theorem [Hartmanis and Stearns, 1965]. Observe though, that the time hierarchy theorem constitutes a hierarchy of *sets*, whereas the current theorem is about a hierarchy of *algorithms*. The proof of Theorem 31 is made possible in its current form by our choice of Turing machines as a model of computation. For other machine models, random access machines for example, different, but at the same time similar, proofs can be given. To make the paper entirely machine model independent, we would have to take the jumps in the time bounds a bit bigger, though still polynomial of small degree as warranted by the sequential computation thesis [Van Emde Boas, 2014], also known as the extended Church–Turing thesis [Parberry, 1986].

We have made use of the time hierarchy theorem to break out of the constant exponent in the running time of fixed-parameter tractable algorithms. When the exponent is allowed to vary, the diagonalization in the proof of Theorem 31 fails. An immediate class that allows such variations in the exponent of the polynomial running time is the semi-uniform variant of \mathbf{XP}_{nu} .

Definition. A set A is in \mathbf{XP}_u with parameterization η if there is a Turing machine Ψ taking two inputs, such that for every parameter value k the partial application of Ψ to k yields an \mathbf{P} -approximation for A with domain η_k .

Not only does the proof of Theorem 31 fail on \mathbf{XP}_u , the proof of Theorem 9 fails on \mathbf{XP}_u as well. The enumeration of slices in that proof is in general not effective and the resulting parameterization need not put the set under consideration in \mathbf{XP}_u . When focusing only on parameterizations that provably put a set in \mathbf{XP}_u , we regain something akin to Theorem 9.

Theorem 33. *Given any formal proof system, for any set A there is a least parameterization among those provably, via a witnessing Turing machine, putting A in \mathbf{XP}_u .*

Proof. Let \mathfrak{F} be the fixed deductive system. A form of universal search, along the lines of Hutter [2002], through \mathbf{P} -approximations is possible in the parameterized setting. Consider the following partial decision procedure for A , uniformly defined for all $k \in \mathbb{N}$. On input x , the procedure takes the following steps.

1. (a) Set $M = \emptyset$.
- (b) For all proofs P in \mathfrak{F} and all machines Φ both of length at most k do:
 If P proves that Φ is a \mathbf{P} -approximation for A then:
 Set $M = M \cup \{\Phi\}$.
2. (a) For Φ in M do:
 If $\Phi(x) \in \{0, 1\}$ then return $\Phi(x)$.
- (b) return nothing.

Since, for all values of k , the set M is finite throughout the execution of this partial decision procedure, step 2a can be executed in polynomial time and the procedure is a \mathbf{P} -approximation for A . Moreover, when we interpret k as the parameter value, the above procedure defines a parameterization η with which A is in \mathbf{XP}_u .

It remains to show that η lies below every other parameterization that provably puts A in \mathbf{XP}_u . Let Ψ be the witness Turing machine corresponding to some parameterization $\eta' \subseteq \mathbb{2}^+ \times \Omega$ with which A is in \mathbf{XP}_u . For any parameter value $\omega \in \Omega$, the slice η'_ω is included in the slice $\eta_{|\langle \Psi, \omega \rangle| + \mathcal{O}(1)}$, where the hidden constant depends on the length of an \mathfrak{F} -proof of A being in \mathbf{XP}_u with parameterization η' and the overhead of a program applying Ψ to ω . Hence we have $\eta \preceq \eta'$ as desired. \square

Thus, adding a provability requirement offsets the limitations we incurred by moving to a uniform setting. The provability requirement in Theorem 33 enforces the effectiveness that was not present in the proof of Theorem 9. Additionally, from the existence of least parameterizations that provably put a set in \mathbf{XP}_u it follows that sets are either almost \mathbf{XP}_u -bi-immune or \mathbf{XP}_u -omni-levelable.

Corollary 34. *When restricting to parameterizations that provably put a set in \mathbf{XP}_u , sets are either almost \mathbf{XP}_u -bi-immune or \mathbf{XP}_u -omni-levelable.*

While a set is \mathbf{P} -omni-levelable precisely when it is $\mathbf{XP}_{\mathbf{nu}}$ -omni-levelable, it could be that the \mathbf{XP}_u -omni-levelable sets form a strict subset of the \mathbf{P} -omni-levelable sets.

Without accounting for proofs, the partial decision procedure constructed in the proof of Theorem 33 would consider only the lengths of the \mathbf{P} -approximations Φ . Then, parameter values are much akin to instance complexity [Orponen et al., 1994].

Definition. The time t bounded *instance complexity* of a string x with respect to a set A is

$$\text{ic}^t(x : A) = \min\{|\Phi| \mid \Phi \text{ is a } t\text{-approximation for } A \text{ and } x \in \text{dom}(\Phi)\},$$

where a t -approximation for A is a partial decision procedure for A that runs in time t .

The following theorem gives a time bounded version of Lemma 6.1 in [Witteveen and Torenvliet, 2016].

Theorem 35. *For any polynomial p , any set A is in \mathbf{FPT}_{nu} with parameterization $\eta = \{(x, k) \mid \text{ic}^p(x : A) \leq k\}$.*

Proof. Let c be such that p is in $\mathcal{O}(n^c)$. For any value of k , there are only finitely many p -approximations for A of length at most k . These can be combined into a single $\mathcal{O}(n^c)$ -approximation for A for which the domain is exactly η_k . Hence, A is in \mathbf{FPT}_{nu} with parameterization η . \square

Observe how this theorem is similar to Theorem 9. Stretching our definition of time bounded instance complexity a little, we can define a parameterization $\{(x, k) \mid \text{ic}^P(x : A) \leq k\}$. The previous theorem can be adapted to show that A is in \mathbf{XP}_{nu} with this parameterization and as seen in the proof of Theorem 9 this parameterization is a principal parameterization in $\mathcal{F}_{(A, \mathbf{XP}_{\text{nu}})}^{\llcorner \text{nu}}$.

Based on Theorem 33 and Theorem 35, for a parameterization η that puts a set in \mathbf{FPT} , we think of μ_η as conveying a sense of complexity. Unlike instance complexity, the sense of complexity conveyed by μ_η can be called *uniform* as it requires a uniform way of deriving approximations. Indeed, there is a sort of converse to Theorem 35.

Theorem 36. *For any set A that is in \mathbf{FPT} with parameterization $\eta \subseteq \mathbb{2}^+ \times \Omega$ there is a function f and polynomial p such that for all x we have, up to an additive constant independent of x ,*

$$\text{ic}^{f(\mu_\eta(x))p}(x : A) \leq \mu_\eta(x).$$

Proof. It suffices to show that for some c and every $\omega \in \Omega$ there is a $\mathcal{O}(n^c)$ -approximation for A of which the size is, up to an additive constant, bounded by $|\omega|$. Let Ψ be the Turing machine witnessing that A is in \mathbf{FPT} with parameterization η . By definition, there is a c such that for every value ω the partial application of Ψ to ω yields a $\mathcal{O}(n^c)$ -approximation for A . As this $\mathcal{O}(n^c)$ -approximation can be constructed from Ψ and ω , the length of its specification can, up to an independent additive constant, be bounded by $|\langle \Psi, \omega \rangle|$. Because Ψ is fixed for all x , the theorem follows. \square

The behavior of the complexity measure embodied, for a parameterization η , by μ_η can be somewhat untangible. When η is not a principal parameterization for a given set, there are sharper parameterizations and hence sharper complexity measures possible with respect to that set. Many sets, however, do not allow for principal parameterizations at all. Furthermore, even when a parameterization η is principal, there can be other parameterizations equal to it that give rise to a measure of complexity lower than μ_η . This improvement though, is of a bounded nature and a sense of optimality is still given to the complexity behavior of principal parameterizations.

Somewhat more abstract, for a set A the filter $\mathcal{F}_{(A, \mathbf{FPT})}^{\llcorner}$ itself can be considered a representation of the distribution of complexity of instances with respect to A . This view has the added benefit that it is applicable not only when the filter is principal. As we did previously in the nonuniform context, we shall classify sets based on their filters with respect to \mathbf{FPT} . Where polynomial isomorphism of sets indicates a *comparable* distribution of difficulty, having the

same filter with respect to **FPT** signifies that *exactly the same* elements are difficult.

Theorem 37. *For any set X in \mathbf{P} and any set A we have*

$$\mathcal{F}_{(A,\mathbf{FPT})}^{\preceq} = \mathcal{F}_{(A\Delta X,\mathbf{FPT})}^{\preceq}.$$

Proof. This follows from the fact that given X , the approximations for A corresponding to a parameterization can be uniformly transformed into approximations for $A \Delta X$ with the same domain, and vice versa. \square

Compared to Theorem 20, we needed the additional remark that the transformation used in the proof is uniform in the parameter. With a similar addition we retrieve a version of Theorem 21 for **FPT**.

Theorem 38. *For any two sets A, B satisfying $\mathcal{F}_{(A,\mathbf{FPT})}^{\preceq} = \mathcal{F}_{(B,\mathbf{FPT})}^{\preceq}$ we have*

$$\mathcal{F}_{(A,\mathbf{FPT})}^{\preceq} \subseteq \mathcal{F}_{(A\Delta B,\mathbf{FPT})}^{\preceq}.$$

Proof. Given a parameterization η with which A and B are in **FPT**, let Ψ_A and Ψ_B be Turing machines that yield the respective approximations on partial application to a parameter value. From Ψ_A and Ψ_B it is possible to define an approximation for $A \Delta B$ uniformly in a parameter value k . On input x , when (x, k) is a member of η this approximation simply outputs the exclusive disjunction of $\Psi_A(k, x)$ and $\Psi_B(k, x)$. It follows that $A \Delta B$ is also in **FPT** with parameterization η . \square

We conjecture that the filter of the symmetric difference collapses to that of a set in \mathbf{P} .

Conjecture 2. *For any two sets A, B we have*

$$\mathcal{F}_{(A,\mathbf{FPT})}^{\preceq} = \mathcal{F}_{(B,\mathbf{FPT})}^{\preceq} \iff \mathcal{F}_{(A\Delta B,\mathbf{FPT})}^{\preceq} = \mathcal{F}_{(\emptyset,\mathbf{FPT})}^{\preceq}.$$

As with Conjecture 1, this conjecture implies a separation result. However, uniformity constraints make this separation result more intricate than that obtained from Conjecture 1.

6 Conclusion

We have explored the algebraic structure of parameterizations underlying the parameterized analysis of computational complexity. Under the general definitions of parameter spaces and parameterizations of Section 3, parameterizations generate a lattice. We found that for several parameterized complexity classes the parameterizations that put a given set in that parameterized complexity class form a filter in this lattice. These filters hold information about the complexity make up of the sets that produce them. Thus, in the examination of sets these filters can act as a proxy to their internal distribution of complexity.

Solely based on the filters induced by given sets with respect to parameterized complexity classes we could extend the classifications “almost \mathbf{P} -bi-immune” and “ \mathbf{P} -omni-levelable” into the parameterized context. We have seen that the classical classifications coincide with the parameterized classifications

“almost \mathbf{XP}_{nu} -bi-immune” and “ \mathbf{XP}_{nu} -omni-levelable”. Furthermore we have seen that for a parameterized complexity class $\mathbf{C} \in \{\mathbf{XP}_{\text{nu}}, \mathbf{XP}_{\mathbf{u}}, \mathbf{FPT}_{\text{nu}}, \mathbf{FPT}_{\mathbf{u}}, \mathbf{FPT}\}$ there are sets that are almost \mathbf{C} -bi-immune as well as sets that are \mathbf{C} -omni-levelable. In particular, when we denote the class of sets outside \mathbf{P} that admit a linearly length increasing reduction to itself by \mathbf{LLI} , we have the following relations between classifications of sets.

$$\text{in } \mathbf{P} \implies \text{almost } \mathbf{P}\text{-bi-immune} \implies \text{almost } \mathbf{C}\text{-bi-immune}$$

and

$$\text{in } \mathbf{LLI} \implies \mathbf{C}\text{-omni-levelable} \implies \mathbf{P}\text{-omni-levelable},$$

where the last implications on each of these two lines become equivalences when \mathbf{C} is \mathbf{XP}_{nu} . Indeed, there exists almost \mathbf{C} -bi-immune sets and \mathbf{C} -omni-levelable sets because \mathbf{P} is nonempty and, by Lemma 18, \mathbf{LLI} is nonempty. These implications expand on Figure 1 and it should be noted that no set can be both almost \mathbf{C} -bi-immune and \mathbf{C} -omni-levelable, and with respect to $\mathbf{C} \in \{\mathbf{FPT}_{\text{nu}}, \mathbf{FPT}_{\mathbf{u}}, \mathbf{FPT}\}$ it is not ruled out that some set is neither.

In defining levelability with respect to classes of fixed-parameter tractable sets we have made explicit the role of the exponent in the involved polynomial running time bounds. Although the alternative characterizations of \mathbf{XP} and \mathbf{FPT} on page 13 provide a good motivation for doing so, we wonder whether any set would meet the naive definition of being \mathbf{FPT} -omni-levelable.

Open problem 1. *Is there a set A and a parameterization η with which A is in \mathbf{FPT} such that every parameterization $\eta' \preceq \eta$ with which A is in \mathbf{FPT} has *imix*?*

Another aspect of parameterized complexity that is captured by filters for sets with respect to parameterized complexity classes is the existence of optimal parameterizations. A parameterization that puts a set in one of our parameterized complexity classes is optimal when it is below all other such parameterizations in the order relevant for the particular parameterized complexity class. It should be noted that the improvement signified by one parameterization being below another is far stronger than the improvements related to typical races in parameterized algorithmics, where a parameterization is held fixed. For our nonuniform and semi-uniform parameterized complexity classes we use the nonuniform order on parameterizations. For our strongly uniform parameterized complexity class the uniform order on parameterizations is the most natural. An optimal parameterization for a set with respect to a parameterized complexity class then exists when the induced filter with respect to that class is principal. Our results on principality of filters with respect to the different parameterized complexity classes can be summarized as in Table 1.

Every value in a cell of Table 1 is backed by one or more theorems or corollaries. By extending the uniformity constraints of $\mathbf{XP}_{\mathbf{u}}$ with provability constraints, we were able to obtain the same principality results as for \mathbf{XP}_{nu} . Regarding fixed-parameter tractability we were able to obtain a necessary and sufficient condition for the principality of filters in the semi-uniform case. A filter for a set with respect to $\mathbf{FPT}_{\mathbf{u}}$ is principal if and only if the set is almost \mathbf{P} -bi-immune. Consequently, no $\mathbf{FPT}_{\mathbf{u}}$ -omni-levelable set has an optimal parameterization with respect to $\mathbf{FPT}_{\mathbf{u}}$ as the $\mathbf{FPT}_{\mathbf{u}}$ -omni-levelable sets are all \mathbf{P} -omni-levelable. This is noteworthy because from a practical point

	<i>almost \mathbf{P}-bi-immune</i>	<i>\mathbf{P}-omni-levelable</i>
\mathbf{XP}_{nu}	principal	principal
$\mathbf{XP}_{\mathbf{u}}$ (provably)	principal	principal
\mathbf{FPT}_{nu}	principal	Theorem 19
$\mathbf{FPT}_{\mathbf{u}}$	principal	nonprincipal
\mathbf{FPT}	Theorem 27	nonprincipal

Table 1: Principality of filters with respect to parameterized complexity classes depending on the classification of the set inducing the filter.

of view a parameterized approach is only worthwhile for sets that are $\mathbf{FPT}_{\mathbf{u}}$ -omni-levelable. In many cases attention is limited even further to only strongly uniform parameterized algorithms. Such algorithms are of practical use in particular for \mathbf{FPT} -omni-levelable sets. As \mathbf{FPT} -omni-levelable are again also \mathbf{P} -omni-levelable, in Table 1 we see that no \mathbf{FPT} -omni-levelable set admits an optimal parameterization with respect to \mathbf{FPT} .

When the filter induced by a set is nonprincipal, it is not possible to capture the structure responsible for the computational hardness of the set by a parameterization. In that case, there are infinitely many distinct structural properties an element may have that can be used to decide membership of the element in the set in a way that defies the computational hardness of the set in general.

For two cells in Table 1 our results are incomplete. The filter of a \mathbf{P} -omni-levelable set with respect to \mathbf{FPT}_{nu} is nonprincipal when the set is not \mathbf{FPT}_{nu} -omni-levelable. However, for \mathbf{FPT}_{nu} -omni-levelable sets the principality of the corresponding filters with respect to \mathbf{FPT}_{nu} is still unknown.

Open problem 2. *Is there a \mathbf{FPT}_{nu} -omni-levelable set A for which $\mathcal{F}_{(A, \mathbf{FPT}_{\text{nu}})}^{\mathcal{S}_{\text{nu}}}$ is principal?*

We have a similar incomplete picture for the filter of an almost \mathbf{P} -bi-immune set with respect to \mathbf{FPT} . Such a filter is known to be principal for the rather trivial case when the set is in \mathbf{P} . Otherwise we only know that filters of \mathbf{P} -omni-levelable sets with respect to \mathbf{FPT} are nonprincipal.

Open problem 3. *Is there a set A outside \mathbf{P} for which $\mathcal{F}_{(A, \mathbf{FPT})}^{\mathcal{S}}$ is principal?*

References

- Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.
- José L. Balcázar and Uwe Schöning. Bi-immune sets for complexity classes. *Theory of Computing Systems*, 18(1):1–10, 1985.
- Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. In *International Colloquium on Automata, Languages, and Programming*, pages 196–207, 2013.
- Ronald V. Book, D-Z Du, and David A. Russo. On polynomial and generalized complexity cores. In *Proceedings of the third annual Structure in Complexity Theory conference*, pages 236–250. IEEE, 1988.

- Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM Symposium on Theory of Computing*, pages 151–158. ACM, 1971.
- Brian A. Davey and Hilary A. Priestley. *Introduction to lattices and order*. Cambridge University Press, 2002.
- Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Springer, 1999.
- Peter van Emde Boas. Machine models and simulations. *Handbook of Theoretical Computer Science, volume A*, pages 1–66, 2014.
- Michael R. Fellows, Bart M.P. Jansen, and Frances Rosamond. Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. *European Journal of Combinatorics*, 34(3):541–566, 2013.
- Jörg Flum and Martin Grohe. Describing parameterized complexity classes. *Information and Computation*, 187(2):291–319, 2003.
- Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Springer, 2006.
- Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.
- Shivam Garg and Geevarghese Philip. Raising the bar for vertex cover: fixed-parameter tractability above a higher guarantee. In *Proceedings of the twenty-seventh annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1152–1166. Society for Industrial and Applied Mathematics, 2016.
- Oded Goldreich. Computational complexity: a conceptual perspective. *ACM SIGACT News*, 39(3):35–39, 2008.
- Juris Hartmanis and Richard E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.
- Marcus Hutter. The fastest and shortest algorithm for all well-defined problems. *International Journal of Foundations of Computer Science*, 13(3):431–443, 2002.
- Ker-I Ko and Daniel Moore. Completeness, approximation and density. *SIAM Journal on Computing*, 10(4):787–796, 1981.
- Christian Komusiewicz and Rolf Niedermeier. New races in parameterized algorithmics. In *Mathematical Foundations of Computer Science*, volume 12, pages 19–30. Springer, 2012.
- Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer, 1997.
- Nancy Lynch. On reducibility to complex or sparse sets. *Journal of the ACM*, 22(3):341–345, 1975.

- Piergiorgio Odifreddi. *Classical Recursion Theory*. Studies in Logic and the Foundations of Mathematics. Elsevier, 1992.
- Pekka Orponen. A classification of complexity core lattices. *Theoretical Computer Science*, 47:121–130, 1986.
- Pekka Orponen, David A. Russo, and Uwe Schöning. Polynomial levelability and maximal complexity cores. In *International Colloquium on Automata, Languages, and Programming*, pages 435–444. Springer, 1985.
- Pekka Orponen, David A. Russo, and Uwe Schöning. Optimal approximations and polynomially levelable sets. *SIAM Journal on Computing*, 15(2):399–408, 1986.
- Pekka Orponen, Ker-I Ko, Uwe Schöning, and Osamu Watanabe. Instance complexity. *Journal of the ACM*, 41(1):96–121, 1994.
- Christos H. Papadimitriou. Computational complexity. In *Encyclopedia of Computer Science*, pages 260–265. Wiley, 2003.
- Ian Parberry. Parallel speedup of sequential machines: A defense of parallel computation thesis. *ACM SIGACT News*, 18(1):54–67, 1986.
- Uwe Schöning. Probabilistic complexity classes and lowness. *Journal of Computer and System Sciences*, 39(1):84–100, 1989.
- Jouke Witteveen and Leen Torenvliet. Fixed-parameter decidability: Extending parameterized complexity analysis. *Mathematical Logic Quarterly*, 62(6):596–607, 2016.
- Paul Young. Some structural properties of polynomial reducibilities and sets in NP. In *Proceedings of the fifteenth annual ACM Symposium on Theory of Computing*, pages 392–401. ACM, 1983.