



UvA-DARE (Digital Academic Repository)

Fixed-parameter decidability

Extending parameterized complexity analysis

Witteveen, J.; Torenvliet, L.

DOI

[10.1002/malq.201500077](https://doi.org/10.1002/malq.201500077)

Publication date

2016

Document Version

Final published version

Published in

Mathematical Logic Quarterly

License

CC BY

[Link to publication](#)

Citation for published version (APA):

Witteveen, J., & Torenvliet, L. (2016). Fixed-parameter decidability: Extending parameterized complexity analysis. *Mathematical Logic Quarterly*, 62(6), 596-607. <https://doi.org/10.1002/malq.201500077>

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.

Fixed-parameter decidability: Extending parameterized complexity analysis

Jouke Witteveen* and Leen Torenvliet

Institute for Logic, Language, and Computation, Universiteit van Amsterdam, Science Park 107,
1098 XG Amsterdam, The Netherlands

Received 2 November 2015, revised 3 June 2016, accepted 1 August 2016

Published online 21 November 2016

We extend the reach of fixed-parameter analysis by introducing classes of parameterized sets defined based on decidability instead of complexity. Known results in computability theory can be expressed in the language of fixed-parameter analysis, making use of the landscape of these new classes. On the one hand this unifies results that would not otherwise show their kinship, while on the other it allows for further exchange of insights between complexity theory and computability theory. In the landscape of our fixed-parameter decidability classes, we recover part of the classification of real numbers according to their computability. From this, using the structural properties of the landscape, we get a new proof of the existence of \mathbf{P} -selective bi-immune sets. Furthermore, we show that parameter values in parameterized sets in our uniformly fixed-parameter decidability classes interact with both instance complexity and Kolmogorov complexity. By deriving a parameter based upper bound on instance complexity, we demonstrate how parameters convey a sense of randomness. Motivated by the instance complexity conjecture, we show that the upper bound on the instance complexity is infinitely often also an upper bound on the Kolmogorov complexity.

© 2016 The Authors. *Mathematical Logic Quarterly* published by WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim.

1 Introduction

In the 1990s, four papers of Downey and Fellows [1, 5–7] enriched the field of computational complexity with parameterized complexity. In particular, their class, \mathbf{FPT} , of fixed-parameter tractable sets became a popular alternative to the long-standing representative of tractability, the class \mathbf{P} . The addition of parameters enables new ways of using limits and recursive definitions. Of particular interest are properties of elements that are preserved in the limit. For such properties, parameterized reasoning is especially suitable.

As it turns out, parameterized analysis was present in computability theory long before it found its way to complexity theory, despite the strong ties between both areas. However, for computability theory, a parameterized framework was never explicitly constructed. We shall make a start in filling this gap by showing how some classical results in computability theory can be neatly phrased in a context with parameters.

It will become clear that a naive formulation of decidability in the presence of parameters, *fixed-parameter decidability*, wherein sets are included if the subsets corresponding to fixed parameter values are decidable, is too broad a concept to be meaningful. For example, any set for which the subsets for fixed parameter values are finite or cofinite would be considered fixed-parameter decidable. We shall explore two ways of augmenting the above definition in order to make fixed-parameter decidability applicable to settings of interest. The first is by posing additional uniformity conditions, which we do in § 3. The second is by confining uniformity into oracles, which we do in § 4. Wrapping nonuniformity in oracles inspires a novel type of fixed-parameter reducibility, which is of interest not only for fixed-parameter decidability, but also takes a natural place in the fixed-parameter tractability landscape.

Applications of fixed-parameter decidability to domains other than fixed-parameter tractability are given in §§ 5 & 6. A link with the computability of real numbers is shown in § 5. There, we also use the structural arrangement of fixed-parameter decidability classes in a novel proof of the existence of a \mathbf{P} -selective bi-immune set. In § 6, typical measures of randomness are connected to the landscape of fixed-parameter decidability classes.

* Corresponding author; e-mail: j.e.witteveen@uva.nl

Parameter values are shown to be related to both Kolmogorov complexity and instance complexity. This sheds new light on the instance complexity conjecture.

2 Preliminaries

In this text, we shall use a binary alphabet $\mathcal{2} = \{0, 1\}$ and denote by $\mathcal{2}^+$ the set of finite non-empty binary strings. The set of natural numbers will be denoted by \mathbb{N} . Central to the discussion will be the notion of a parameterized set.

Definition 2.1 A *parameterized set* is a subset of $\mathcal{2}^+ \times \mathbb{N}$. Given a *parameter* $k \in \mathbb{N}$, the *kth slice* of a parameterized set A is the set

$$A_k = \{x \mid (x, k) \in A\}.$$

A parameterized set A is *convergent* if we have:

$$\forall x \exists k_0 \forall k \geq k_0 (x \in A_{k_0} \iff x \in A_k). \quad (1)$$

A convergent parameterized set A is said to converge to the unique *limit* A_∞ extending (1).

Throughout this text, a set named A will be a parameterized set, whereas a set named B will be an arbitrary set.

Parameterized sets that are *not* convergent can be interesting. In fact, the theory we shall develop in § 4 makes very little use of convergence. Nevertheless, fixed-parameter analysis is normally involved with convergent sets. Moreover, the implied convergence when considering fixed-parameter tractability is often of a particularly strong kind.

Definition 2.2 A parameterized set A is *monotonically non-decreasing*, or simply *monotonic*, if, for all $k \in \mathbb{N}$, we have $A_k \subseteq A_{k+1}$. Every monotonic parameterized set A is convergent and it converges to the set $A_\infty = \bigcup_k A_k$.

The alternative definition of a parameterized set that is used by Flum and Grohe [13] internalizes monotonicity and represents a kind of modulus of convergence. As a result, it poses a substantial restriction on what sets can be considered fixed-parameter tractable. Since most of the current work would not be possible with a monotonicity requirement in the definition of a parameterized set, we argue that the definition of Flum and Grohe is overly restrictive, even when focussing on computational complexity.

We shall use the standard notion of computability derived from accepting the *Turing machine* as a model for effective computation. Familiarity with Turing machines and oracle Turing machines is assumed. For a short overview of the relevant definitions, we refer to [19] and [9]. Notational conventions are mostly taken from [8, 9], in particular, the set of halting Turing machines, as encoded in $\mathcal{2}^+$, is denoted by \emptyset' , the Turing jump of the empty set.

As is common, we shall define *computable partial functions* by associating, through coding, the input and output of a Turing machine with arguments and function values, respectively. Here, we leave the function value undefined if the Turing machine does not halt. Thus the class of Turing machines corresponds to the class of computable partial functions. A computable partial function is simply a *computable function* if it is nowhere undefined. We also define some related notions, which have to do with deciding the membership question of elements of sets.

Definition 2.3 A Turing machine Φ is a *partial decision procedure* for a given set B if we have, for all x :

1. $\Phi(x) = 0 \Rightarrow x \notin B$.
2. $\Phi(x) = 1 \Rightarrow x \in B$.

Note that no requirements are in place for the situation where, given x as input, Φ does not halt or outputs anything other than 0 or 1. A set B for which a partial decision procedure exists that outputs 1 on all members of B , in other words, a procedure that correctly identifies the members of B , is called *semidecidable*. Semidecidable sets are also known as computably enumerable sets or recursively enumerable sets. A partial decision procedure for a

set B that always halts is called a *consistent procedure* with respect to B . Such a procedure can be interpreted as one that, when asked whether some x is a member of B answers ‘yes’, ‘no’, or ‘unresolved’. Further strengthening the definition by requiring that the output of Φ is always in $\mathcal{2}$, in other words, forbidding the ‘unresolved’ answer, yields the definition of a *decision procedure*. Sets for which a decision procedure exist are called *decidable*.

Turning to oracle Turing machines instead of plain Turing machines, we obtain familiar reducibility notions.

Definition 2.4 An oracle Turing machine Φ is a *Turing reduction* from a set B to a set H if we have, for all x :

1. $\Phi^H(x) = 0 \iff x \notin B$.
2. $\Phi^H(x) = 1 \iff x \in B$.

Note that Φ is not required to halt on any input x when equipped with an oracle different from H . Moreover, it is permitted to make queries to the oracle indefinitely. Restricting the latter, demanding that Φ makes only finitely many queries to the oracle, regardless the input, we get the definition of a *weak truth-table reduction*. Restricting the former, demanding that Φ halts on any input, regardless the oracle, we get the definition of a *truth-table reduction*. Indeed, a truth-table reduction is also a weak truth-table reduction. We can interpret the finitely many queries a truth-table reduction makes to the oracle on a given input x as variables in a truth table, where the operational value of any given assignment to the variables in that truth table is the output of the reduction. Hence the name truth-table reduction.

3 Uniform and strongly uniform fixed-parameter decidability

A parameterized study of decidability was started already in 1965 [14, 21], some 30 years before a parameterized investigation of computational complexity took off. Using the new terminology of fixed-parameter tractability [8], we can recast several old results into a unified framework. In particular, this opens the door for crossbreeding between parameterized decidability and computability research. One immediate result is that we observe that the most general definitions of fixed-parameter tractability classes [8] make no mention of any convergence criteria, whereas such criteria are part of all parameterized approaches to decidability from the onset. It appears to us that the tractability classes too would benefit from a convergence requirement. For our definition of uniform fixed-parameter decidability, we choose to restrict to convergent parameterized sets.

Definition 3.1 A convergent parameterized set A is *uniformly fixed-parameter decidable* if there is a computable function $\Phi : 2^+ \times \mathbb{N} \rightarrow \mathcal{2}$ such that $\Phi(x, k) = 1$ if we have $x \in A_k$ and $\Phi(x, k) = 0$ otherwise.

The set is *strongly uniformly fixed-parameter decidable* if there is an additional computable *bounding function* $g : 2^+ \rightarrow \mathbb{N}$ such that, for all $x \in 2^+$, we have $\|\{k \mid \Phi(x, k) \neq \Phi(x, k + 1)\}\| \leq g(x)$.

The classes of uniformly and strongly uniformly fixed-parameter decidable sets are designated by **uFPD** and **suFPD** respectively.

Note that a convergent parameterized set is uniformly fixed-parameter decidable precisely when it is decidable. However, decidability of the set is independent of decidability of its limit. Partial application of some Φ as in the above definition yields decision procedures for the slices of a parameterized set A , but this does not extend to A_∞ . For monotonic sets, though, fixed-parameter decidability is related to semidecidability of the limit.

Lemma 3.2 For a set B , the following are equivalent:

1. There exists a monotonic parameterized set in **suFPD** that converges to B .
2. The set B is semidecidable.

Proof. $1 \implies 2$. We specify a partial decision procedure for B that halts precisely on the members of B , based on a decision procedure Φ of the monotonic parameterized set. On input x , start iterating over possible values of k until one is found such that we have $\Phi(x, k) = 1$. If one is eventually found, halt and output 1. Otherwise, by necessity, we keep computing forever.

$2 \implies 1$. We construct a monotonic parameterized set via its decision procedure, using the partial decision procedure ψ of B . On input (x, k) , we simulate the computation of $\psi(x)$ for k steps. If ψ halts within k steps, we output accordingly. Otherwise, we output 0. \square

As monotonic parameterized sets in **suFPD** are members of **suFPD** where the bounding function is the constant function yielding 1 (note that the converse is not true), Lemma 3.2 inspires a structure inside **uFPD** extending semidecidability. The class of semidecidable sets is closed under countable unions, provided the indices of the constituents of the union form a semidecidable set. The same does not go for countable intersections or difference. Already for a finite collection of semidecidable sets, the symmetric difference need not be semidecidable. Closing the semidecidable sets under an increasing number of applications of symmetric difference, we obtain the finite levels of Ershov's *difference hierarchy*, which stems back to the 1960s [11, 21]. To be precise, the members of the n th level of the difference hierarchy are the sets that can be written as the symmetric difference of n semidecidable sets. We recover this part of the hierarchy within our framework.

Lemma 3.3 *The levels of the difference hierarchy coincide with the classes of limits of parameterized sets in **suFPD** for which the bounding function is constant and the first slice is empty.*

The last requirement, the first slice being empty, is of little importance. Dropping it gives us the *weak* levels of the difference hierarchy [10], which are positioned so that the n th level of the weak difference hierarchy is between the n th and the $(n + 1)$ th level of the difference hierarchy. From a parameterized view on decidability and with Lemma 3.2 at hand, the above lemma is as much a definition of the difference hierarchy as it is a lemma in its own right [9]. Already in the standard proofs of the strictness of the inclusions in the difference hierarchy [3, 11, 21] we can see the parameterized approach at work. It is possible to extend the difference hierarchy to infinite computable ordinals [3, 9, 12], but of interest in the context of fixed-parameter decidability is only level ω , which neatly coincides with the class of limits of parameterized sets in **suFPD**. Within **suFPD**, a fine grained structure exists driven by unbounded computable functions [3, 10].

Lemma 3.4 *Let $f, g : 2^+ \rightarrow \mathbb{N}$ be computable functions such that $f(x)$ is less than $g(x)$ for infinitely many x . Then there is a set that occurs as the limit of a parameterized set in **suFPD** with bounding function g that does not occur as the limit of any parameterized set in **suFPD** with bounding function f .*

The above lemma decomposes the class of limits of sets in **suFPD** into a distributive lattice of subclasses based on bounding functions. This lattice lacks a greatest element as there is no pointwise greatest computable function. Nevertheless, **suFPD** does emerge as a closure of a certain set with respect to a type of reductions [3, 9].

Theorem 3.5 *A set truth-table reduces to \emptyset' if and only if it is the limit of some parameterized set in **suFPD**.*

Proof. \implies . Note that parameterized sets converging to \emptyset' exist in **suFPD**. Now, when, on input (x, k) , we evaluate the truth-table corresponding to x on slice k of some chosen parameterized set converging to \emptyset' , we obtain a parameterized set in **suFPD** that converges to the desired set.

\longleftarrow . Given a decision function Φ of a parameterized set in **suFPD**, we can effectively construct a computable partial function that is defined precisely for those x and c for which we have $\|\{k \mid \Phi(x, k) \neq \Phi(x, k + 1)\}\| \geq c$. By repeated queries to the halting set, we can determine the least c for which such a function does not halt for a given x . Based on this information we can decide membership in the limit of the parameterized set that we began with. In the case of **suFPD** the number of queries necessary is bounded, hence a truth-table reduction to the halting set, \emptyset' , suffices. \square

Regarding Theorem 3.5 it is worth noting that when it comes to reducibility to the halting set, truth-table reducibility is equivalent to weak truth-table reducibility [9].

A theorem similar to Theorem 3.5, but applicable to **uFPD** instead of **suFPD**, was proven already in 1959 [24] and is known as the *limit lemma* [9].

Theorem 3.6 *A set Turing reduces to \emptyset' if and only if it is the limit of some parameterized set in **uFPD**.*

Proof. \implies . A similar approach as in Theorem 3.5 is applicable, but we have to limit the running time of the computation to ensure we end up with a terminating computation. Limiting by the parameter value k is sufficient as from some k on, the queried elements of \emptyset' will be correctly enumerated and in particular only finitely many queries will be made before the procedure terminates within k steps. From that k on, membership of x will be

settled correctly. The decision in case of an aborted computation is irrelevant, as only finitely many values of k will lead to an aborted computation on a given x .

←. The same proof as in Theorem 3.5 works. For the present case, there need not be a computable bound on the number of queries necessary, but it will still be finite for every x . \square

We note that the limits of sets in **uFPD** thus form the limit of the infinite Ershov hierarchy [10, 12]. The class of limits of sets in **uFPD** is closed under countable symmetric difference, given the indices of the constituents of the symmetric difference form a semidecidable set and the symmetric difference is well-defined. Furthermore, by Post's theorem, this class equals the level Δ_2^0 of the arithmetic hierarchy [9].

Because the truth-table degree of the halting set is strictly smaller than the Turing degree of the halting set, we conclude from Theorem 3.5 and Theorem 3.6 the strict inclusion **suFPD** \subset **uFPD** [9, 10]. This inclusion holds in a strong sense, namely that there are parameterized sets in **uFPD** for which the limit is not the limit of any parameterized set in **suFPD**.

With respect to complexity classes of parameterized sets, we observe an absence of theorems such as Theorem 3.5 or Theorem 3.6. We have not only characterized members of our fixed-parameter decidability classes based on their internal structure, but also based on properties of their limits. A similar characterization cannot separate fixed-parameter complexity classes including but not limited to **FPT** and the levels of the **W**-hierarchy [6]. In fact, we can place bounds on the running time already for **uFPD**, comparable to what we did in the left-to-right part of Theorem 3.6.

Lemma 3.7 *For every parameterized set in **uFPD** there is a parameterized set in **uFPD** with the same limit and with a decision procedure of which the running time depends linearly on the parameter and on nothing else.*

Proof. Let A be a parameterized set in **uFPD** and Φ a decision procedure for A . Define a procedure Φ' with the desired properties as follows. On input (x, k) , spend k steps in total on simulating $\Phi(x, j)$ for each $j \in \mathbb{N}$ in ascending order. After this, return the decision of the last completed computation, or 0 if no simulation finished.

For too small k , it might not even be possible to fully read x , but for every x there is some, possibly huge, value of k with which there is enough time to compute the limit decision. Thus Φ' is the decision procedure of a parameterized set in **uFPD** with the same limit as A . By construction, the running time of Φ' depends only on k . \square

We shall call a parameterized sets with a decision procedure of which the running time depends solely on the parameter *slowly convergent*. Note that similar lemmas can be stated for other resources, such as space, and for resource dependence complying with constructible functions of the parameter.

Combined, Theorem 3.6 and Lemma 3.7 give a characterization of the limits of members of fixed-parameter complexity classes.

Corollary 3.8 *A set Turing reduces to \emptyset' if and only if it is the limit of some parameterized set in any fixed-parameter complexity class.*

This can be interpreted either as an indication of the inherent importance of the particular ways a parameterized set converges to its limit, or as a motivation for further refinement of the definitions of the fixed-parameter tractability classes.

4 Relativized fixed-parameter decidability

Following terminology of fixed-parameter tractability classes [8], the naive version of fixed-parameter decidability put forward in the introduction should be called nonuniform fixed parameter decidability.

Definition 4.1 A parameterized set A is *nonuniformly fixed-parameter decidable* if each of its slices A_k , with $k \in \mathbb{N}$, is decidable.

The class of nonuniformly fixed-parameter decidable sets is designated by **nuFPD**.

From a convergence point of view, nonuniform fixed-parameter tractability is not interesting because every set can occur as the limit of a parameterized set in **nuFPD**. An aspect of the study of parameterized sets that is of interest when looking at **nuFPD** is kernelizability. In the realm of fixed-parameter tractability, kernelizations are

a fruitful research topic. For our nonuniform fixed-parameter decidability, a definition similar to the complexity theoretic one [8] is possible.

Definition 4.2 A parameterized set A is *kernelizable* if for every $k \in \mathbb{N}$ there is a computable function $f_k : 2^+ \rightarrow 2^+ \times \mathbb{N}$ and a constant c_k such that the following hold for all $x \in 2^+$:

1. $x \in A_k \iff f_k(x) \in A$.
2. $|f_k(x)| \leq c_k$.

Note that we do not require either f_k or c_k to be computable from k . While this definition is a natural variation on its fixed-parameter tractability counterpart, we know of no natural examples of kernelizability in our nonuniform fixed-parameter decidability context. Yet, the fundamental theorem of kernelizability in fixed-parameter tractability carries over to our context.

Theorem 4.3 A parameterized set is in **nuFPD** if and only if it is kernelizable.

Proof. \implies . We shall give a construction of f_k for arbitrary k . On input x , use the decision procedure of A_k to settle the membership question of x in A_k . Next, output (y, k) , where y is the least element of 2^+ that has the same membership of A_k as x . Observe that f_k is computable given the decision procedure of A_k and that its range consists of at most two elements, so $|f_k(x)|$ is bounded by a constant.

\impliedby . We consider the case for an arbitrary k . The requirement $|f_k(x)| \leq c_k$ implies that the range of f_k is finite. By hardcoding which of those finitely many elements in the range of f_k belong to A , we obtain, from f_k , a decision procedure for A_k . \square

The above proof hinges on the fact that all finite sets are decidable. Indeed, the definition of **nuFPD** allows for undecidable collections of decidable sets. Ideally, we want to have a handle on the amount of nonuniformity required for deciding the slices of a parameterized set and **nuFPD** is too permissive for that. In the previous section, we addressed this permissiveness by imposing uniformity constraints. A gentler treatment is possible by controlling the amount of nonuniformity accepted. This is particularly relevant when studying preservation of any structure captured in a parameter under reductions from one parameterized set to another. Although the dominant type of reduction in fixed-parameter tractability research, the *fpt-reduction*, is a many-one reduction, we feel a more general reduction is more natural in the context of fixed-parameter decidability. Note, though, that the analog of the *fpt-reduction* for our fixed-parameter decidability classes works precisely as one would expect. The more natural reduction we propose for a study of fixed-parameter decidability where we can control nonuniformity is based on oracle Turing machines.

Definition 4.4 A parameterized set A is *fixed-parameter reducible* to a parameterized set H if the following hold:

1. There is an oracle Turing machine Φ that decides A when given H as oracle.
2. There is a computable function $g : \mathbb{N} \rightarrow \mathbb{N}$ such that on input (x, k) , no elements of slices of H above the $g(k)$ th are queried by Φ when given H as oracle.

In that case, we write $A \preceq_{\text{FP}} H$.

This reducibility notion defines a preorder on the parameterized sets and we obtain decidability classes as subclasses in the class of parameterized sets that are closed under this preorder. The principal relativized fixed-parameter decidability classes are then defined with respect to a single oracle that, accordingly, becomes complete for the defined class.

Definition 4.5 A parameterized set A is in **FPD** ^{H} if we have $A \preceq_{\text{FP}} H$.

When we maximally limit the power of the oracle, we obtain **FPD** ^{\emptyset} . This class is essentially just **uFPD**, without the requirement of convergence. Moreover, we can replace \emptyset by any decidable parameterized set. Consequently, **FPD** ^{\emptyset} is the class of decidable parameterized sets and in particular it is closed with respect to fixed-parameter reducibility. It also follows that, when we restrict our attention to convergent parameterized sets, the class **uFPD** is closed with respect to fixed-parameter reducibility. This does not extend to **suFPD**, as it cannot be distinguished

from **uFPD**, which is the smallest non-empty class of convergent parameterized sets definable from \preceq_{FP} , by fixed-parameter reducibility.

Our biggest class, **nuFPD**, is also closed under fixed-parameter reducibility. This can be shown similar to Theorem 4.3, but instead of hardcoding membership of elements, we hardcode finitely many decision procedures for slices. We observe though, that every permutation of some infinite set of decidable sets gives rise to a different parameterized set in **nuFPD**, hence **nuFPD** is uncountable and cannot have a complete parameterized set. Notably, this means that there is no parameterized set H for which we get $\mathbf{nuFPD} = \mathbf{FPD}^H$. Whereas \mathbf{FPD}^\emptyset coincides with a minimal fixed-parameter degree, **nuFPD** cannot be written as a countable union of fixed-parameter degrees.

When we limit fixed-parameter reducibility to the usual fixed-parameter tractability running time, we get something of a relativized fpt-reducibility that also defines a preorder. To this reducibility notion, oracles picked from **FPT** provide no additional computational strength. Thus, we find that **FPT** is closed under this relativized fpt-reducibility. This provides a new characterization of **FPT**. With this new characterization, we get the potentially easy way to prove membership of **FPT** by showing relativized fpt-reducibility to a known member of **FPT**.

5 Real numbers and parameters

Real numbers can be encoded as subsets of a countable set in more than one way. It turns out that the classification of real numbers according to their computability is often sensitive to the chosen encoding [25]. We shall consider two prominent encodings, Dedekind cuts and binary expansions, and investigate computability notions embodied by our fixed-parameter decidability classes. For both these encodings, a real number is *computable* if and only if the encoding, the Dedekind cut or binary expansion, is decidable [2, 9, 25]. Sensitivity to the encoding arises only with less restrictive notions of computability of real numbers.

Through any effective encoding of the rational numbers, \mathbb{Q} , into 2^+ we can interpret parameterized sets as subsets of $\mathbb{Q} \times \mathbb{N}$. This enables us to look at Dedekind cuts of real numbers that arise as limits of parameterized sets. In particular, for a real number a we are interested in its *left cut* $\{q \in \mathbb{Q} \mid q \leq a\}$ occurring as the limit of a parameterized set in one of our fixed-parameter decidability classes. A real number with a semidecidable left cut is called *left computable* [2, 9] (and also *lower semicomputable* [19]) and thus by Lemma 3.2 a real number is left computable if and only if its left cut is the limit of some monotonic parameterized set in **suFPD**. Real numbers for which either their left cut, or the complement of their left cut is semidecidable are called *semicomputable* [2, 9, 25]. Alternatively, the semicomputable real numbers can be characterized as those for which there exists either a non-decreasing or a non-increasing computable sequence of rational numbers that converges to them [2, 9, 25]. Exploiting this characterization, we can locate the semicomputable real numbers in our framework.

Theorem 5.1 *A real number is semicomputable if and only if its left cut is the limit of some parameterized set in **suFPD** with constant bounding function 1.*

Proof. We only need to prove the right-to-left direction. Therefore, we shall show that a parameterized set A as in the theorem converges to the left cut of a real number a that is semicomputable. For that, we may assume the left cut is not itself decidable. This means that at least one of $A_\infty \setminus A_k$ and $A_k \setminus A_\infty$ has, for all k , an infinite intersection with every non-empty interval around a . We shall assume the former and construct a non-decreasing computable sequence of rational numbers converging to a . For the latter case, a similar construction yields a non-increasing computable sequence of rational numbers converging to a .

Define the sequence $(q_i)_{i \in \mathbb{N}}$ starting from q_0 , which we set to some rational number below a , by setting q_{i+1} to the largest rational number that meets the following requirements.

1. $q_i < q_{i+1} \leq q_i + i$.
2. The denominator of q_{i+1} is at most i .
3. For some $k \leq i$ we have $q_{i+1} \in A_{k+1} \setminus A_k$.

If no rational number meets these requirements, we set q_{i+1} equal to q_i . Observe that q_{i+1} is computable, as there are only finitely many candidates for which finitely many checks have to be carried out. Also, $(q_i)_{i \in \mathbb{N}}$ is non-decreasing. Furthermore, for every q in A_∞ there is some i such that we have $q \leq q_i < a$, hence $(q_i)_{i \in \mathbb{N}}$ converges to a . \square

This theorem can be extended to arbitrary constant bounding functions [25], showing that no finite level of the difference hierarchy properly extends the semicomputable real numbers when treated as Dedekind cuts.

In computer science research, the representation of real numbers by means of binary expansions is more traditional than that by means of Dedekind cuts. Without loss of generality, we focus on the real numbers between 0 and 1 and represent them as subsets of \mathbb{N} , linking characteristic functions to binary expansions. Note that this representation is not unique for dyadic rationals, but this is a technicality that can often be ignored.

Unfortunately, for binary expansions Theorem 5.1 fails dramatically. Even the union of all finite levels of the difference hierarchy is incomparable to the class of semicomputable real numbers treated as binary expansions [25]. In other words, when working with binary expansions we cannot easily describe the class of semicomputable real numbers in terms of limits of certain sets in **suFPD**. However, the class of real numbers corresponding to limits of sets in **suFPD**, with no further restriction on the computable bounding functions, does properly extend the semicomputable real numbers [25]. This class of real numbers is referred to as the class of ω -computably enumerable real numbers [2, 25]. Even more general is the class of real numbers corresponding, using binary expansions, to limits of sets in **uFPD**. This is the class of *computably approximable* real numbers [2, 25], which have also been called *limit computable* real numbers [14]. When a is a computably approximable real number corresponding to the limit of a parameterized set A in **uFPD**, the computable reals a_k corresponding to the slices A_k of A form a sequence $(a_k)_{k \in \mathbb{N}}$ approximating a . Such a parameterized set and such an approximation are called *normed* if the computable reals a_k are dyadic rational numbers with denominators at most 2^k [2].

Lemma 5.2 *Every computably approximable real number has a normed approximation.*

Proof. Let A be the parameterized set corresponding to the computably approximable real number a . Consider the parameterized set $A' = \{(x, k) \mid x \in A_k \wedge x \leq k\}$, which is a slice-wise truncated version of A . The slices A'_k correspond to dyadic rationals with denominators at most 2^k . Furthermore, A' is decidable and it converges to the same limit as A , hence A' corresponds to a normed approximation of a . \square

Parameterized sets are a natural environment for the application of the priority method, where sets are built in a countably infinite number of stages [9]. Many of the structural results about our fixed-parameter decidability classes incorporate a priority argument in their proofs somewhere. These arguments are sometimes quite involved. Therefore, linking fixed-parameter decidability to other contexts, such as that of the computability of real numbers, can add powerful tools to those contexts.

As an example of this additional power, we shall proceed by giving a simple proof of a theorem of which the published proof is markedly convoluted. The theorem asserts the existence of a set with two properties, the first of which being **P**-selectivity [23].

Definition 5.3 A set $B \subseteq 2^+$ is **P-selective** if there exists a function $f : 2^+ \times 2^+ \rightarrow 2^+$ that is computable in polynomial time so that the following hold for all $x, y \in 2^+$:

1. $f(x, y) \in \{x, y\}$.
2. If we have $x \in B$ or $y \in B$ then we have $f(x, y) \in B$.

An earlier analogue of this property without a restriction on the running time has been used in the study of reducibilities on semidecidable sets [17]. Versions with a constrained running time, such as our **P**-selectivity, have been most successfully used in the study of feasibility of algorithms and computational simplicity of sets [16].

The second property is the more commonplace bi-immunity [9, 19].

Definition 5.4 A set is *bi-immune* if neither it nor its complement has an infinite semidecidable subset.

It may appear that **P**-selectivity is a restrictive property with respect to the computability of a set and, in particular, its subsets. However, while bi-immunity requires a set and its subsets to be far from computable, it is possible to combine the two properties in one set [15].

Theorem 5.5 *There exists a set that is both P-selective and bi-immune.*

For a more concise proof of this theorem than the one that was published, we shall make use of the observation that Lemma 3.7 and Lemma 5.2 can be combined. That is, a set is the limit of a parameterized set in **uFPD** if and only if it is the limit of some slowly convergent normed parameterized set.

Proof. Let A be a slowly convergent normed parameterized set of which the limit is not the limit of a member of **suFPD**. Such a set exists because the inclusion of **suFPD** in **uFPD** is proper in the strong sense we observed in § 3.

Let a be the real number corresponding to the limit of A and let $a_{|z|}$ be the dyadic rational number represented by $A_{|z|}$. We claim that the set $B = \{z \mid a_{|z|} < a\}$ is **P**-selective and bi-immune.

For **P**-selectivity, consider the function

$$f(x, y) = \begin{cases} x & \text{if } a_{|x|} \leq a_{|y|} \\ y & \text{otherwise.} \end{cases}$$

This function meets the two listed requirements in Definition 5.3. It remains to show that it is computable in polynomial time. Because A is normed, computing f only requires to decide on membership of at most $\min\{|x|, |y|\}$ pairs of elements. Because A is slowly convergent, each pair can be decided within a running time of $|x| + |y|$. Hence, f witnesses the **P**-selectivity of B .

For bi-immunity, we first consider infinite semidecidable subsets of B . If there was an infinite semidecidable subset B' of B , then from $(a_{|z|})_{z \in B'}$ we could distill a non-decreasing computable sequence of rational numbers converging to a . Thus, a would be semicomputable, which it cannot be because it is not the limit of any member of **suFPD**. Conversely, infinite semidecidable subsets of the complement of B would lead to non-increasing computable sequences of rational numbers converging to a . Again, this would contradict our choice of A . \square

Essentially, the above proof comes down to the observation that computably approximable real numbers with slowly convergent normed approximations exist also outside the ω -computably enumerable real numbers.

We note that the link between parameter values and computable real numbers represented by slices provides an alternative ordering of parameter values. This alternative ordering has more structure than the standard ordering of parameter values. In the above proof of bi-immunity, it is crucial that the alternative order has precisely one accumulation point. This perspective demonstrates the potential for parameters that showcase an additional structure different from \mathbb{N} .

6 Algorithmic randomness and parameters

In isolation, an element x of 2^+ may be called *random* if its length does not significantly exceed the length of any complete description of x . This idea is made formal by using the *Kolmogorov complexity* [9, 19], $K(x)$, as a measure of the shortest complete description of x . Loosely speaking, the Kolmogorov complexity is defined as

$$K(x) = \min\{|\Phi| \mid \Phi \text{ is a Turing machine that takes no input and outputs } x\}.$$

By referring to the length of a Turing machine, the Kolmogorov complexity is only defined up to an additive constant depending on the encoding of Turing machines used.

When investigating the computability of sets, we can ask ourselves what elements are responsible for the level of computability a set is in. This intuition of inherent difficulty in deciding membership of certain instances in a set is made precise by the notion of *instance complexity* [20]. With respect to a set B , the instance complexity of an element x of 2^+ is defined as

$$\text{ic}(x : B) = \min\{|\Phi| \mid \Phi \text{ is a consistent procedure with respect to } B \text{ and } \Phi(x) \in 2\}.$$

Thus, the instance complexity of an element x with respect to a set B is a measure of the shortest procedure that decides the membership of x in B without making mistakes on other elements. Like Kolmogorov complexity, instance complexity too is only defined up to an additive constant. Still, both notions are rigid enough to be of great use [4, 18–20]. We remind ourselves of the fact that a consistent procedure was defined in § 2 as a Turing machine that always halts and outputs either ‘yes’, ‘no’, or ‘unresolved’. Broadening the above definition of instance complexity to allow for partial decision procedures, which are permitted to not halt, we obtain a weaker notion of instance complexity [18]. We shall not use this weaker notion and stick with the definition based on consistent procedures.

For any set B and any element x of 2^+ we have, up to an additive constant independent of B and x , the inequality $\text{ic}(x : B) \leq K(x)$ [19]. This bound is witnessed by a procedure that first generates x and then checks

its input against x , yielding an embedded membership decision in case of a match and ‘unresolved’ otherwise. When the set B is decidable, we can do a lot better, since a decision procedure for B is in particular a consistent procedure for B . Hence, the instance complexity of any element with respect to a decidable set is bounded. The converse is also true, as membership of any element is then correctly decided by one or more members of a finite collection of consistent procedures. Still, for sets that are not decidable questions concerning the tightness of the inequality between instance complexity and Kolmogorov complexity have been of great interest [4, 18–20]. It has been conjectured that with respect to any set that is not decidable the instance complexity is within a fixed additive distance from the Kolmogorov complexity for infinitely many elements of 2^+ [4, 18, 20]. Such elements are called *hard* instances with respect to the set in much the same way elements of which the Kolmogorov complexity is close to the length are called random.

Note that there is a significant gap between the instance complexity with respect to decidable sets and the conjectured instance complexity of infinitely many elements with respect to sets that are not decidable. With respect to decidable sets the instance complexity is bounded, while with respect to sets that are not decidable the instance complexity is conjectured to be close to the Kolmogorov complexity infinitely often. While classes of sets are known for which indeed with respect to any set in the class there are infinitely many hard instances, the conjectured lower bound fails in general. Using a priority argument, a semidecidable but not decidable set has been constructed with respect to which the instance complexity is always less than the logarithm of the Kolmogorov complexity, up to an additive constant [18]. This result is tight in the sense that with respect to any set that is not decidable the instance complexity must infinitely often be at least as large as the logarithm of the Kolmogorov complexity [18].

Parameterized analysis provides new means of tackling questions related to the existence of hard instances. For parameterized sets A in \mathbf{FPD}^\varnothing , a computable partial function μ can be defined as

$$\mu(x : A) = \min\{k \mid x \in A_k\},$$

where the minimum of the empty set is left undefined. This partial function is especially meaningful when A is monotonic. For a monotonic set A that is in \mathbf{FPD}^\varnothing , and thus in \mathbf{suFPD} , the domain of the partial function equals A_∞ , which is semidecidable by Lemma 3.2. An immediate relationship between instance complexity and the value of μ is available for monotonic parameterized sets in \mathbf{suFPD} .

Lemma 6.1 *For any monotonic parameterized set A in \mathbf{suFPD} , for all $x \in A_\infty$, we have, up to an additive constant independent of x , the inequality $\text{ic}(x : A_\infty) \leq |\mu(x : A)|$.*

Proof. Let Φ be a decision procedure for A . From Φ we can get decision procedures for the slices of A . If we modify a decision procedure for a slice of A so that it outputs ‘unresolved’ whenever it would output 0, we obtain a consistent procedure with respect to A_∞ . The specification of such a consistent procedure, for a slice k , is possible within length $|k| + c$, where c is a constant depending on Φ . For all $x \in A_\infty$, taking $k = \mu(x : A)$, we thus obtain a decision procedure that halts on x and satisfies the required inequality. \square

Although with respect to arbitrary undecidable sets the conjectured existence of infinitely many hard instances need not be true, we can prove that the Kolmogorov complexity gets below our μ infinitely often. Together with Lemma 6.1, this bounds the distance between the instance complexity and the Kolmogorov complexity.

Theorem 6.2 *For a parameterized set with infinitely many different slices A in \mathbf{FPD}^\varnothing , there are infinitely many x for which we have $\mathbf{K}(x) \leq |\mu(x : A)|$.*

Proof. Observe that for any parameterized set A in \mathbf{FPD}^\varnothing that has infinitely many different slices and any constant n , we can find an x such that we have $\mu(x : A) \geq n$ or even $|\mu(x : A)| \geq n$. This can be made effective, be it tedious, by means of a computable function $f_A : \mathbb{N} \rightarrow 2^+$ so that we have $|\mu(f_A(n) : A)| \geq n$. It follows that there is a constant c depending on f_A such that for all n we have $\mathbf{K}(f_A(n)) \leq |n| + c$.

Without loss of generality, we may assume that f_A is injective. Now suppose that for almost all n we have $\mathbf{K}(f_A(n)) > |\mu(f_A(n) : A)|$. By construction we would then find, for almost all n , the inequalities $n \leq |\mu(f_A(n) : A)| < \mathbf{K}(f_A(n)) \leq |n| + c$, which cannot be true. Hence there are infinitely many n for which we have $\mathbf{K}(f_A(n)) \leq |\mu(f_A(n) : A)|$, proving the theorem. \square

Of course, there may be infinitely many x sharing a fixed value under μ , whereas only finitely many of these x could have a Kolmogorov complexity less than this value under μ . As a corollary to the above theorem in

combination with Lemma 6.1, we obtain that if we can find a monotonic parameterized set A in **suFPD** such that for all x we have that $|\mu(x : A)|$ gets within an additive constant of $\text{ic}(x : A_\infty)$, the instance complexity conjecture holds for A_∞ . A direct attempt at applying this corollary would, given an arbitrary set B , look at the monotonic parameterized set

$$A = \{(x, k) \mid x \in B \wedge \text{ic}(x : B) \leq |k|\}.$$

With this parameterized set we have, for all $x \in B$, $|\mu(x : A)| = \text{ic}(x : A_\infty)$ and $A_\infty = B$. However, A need not be in **suFPD**. In order to get a grip on the fixed-parameter decidability of A , consider the auxiliary monotonic parameterized set

$$H = \{(\Phi, k) \mid \Phi \text{ is a consistent procedure with respect to } B \wedge |\Phi| \leq |k|\}.$$

This parameterized set is so that A is in \mathbf{FPD}^H . Additionally, because the slices of H are all finite, we find that A is in **nuFPD**. Yet, by Rice's theorem [22], H_∞ is undecidable and by the relationship between $|\Phi|$ and $|k|$ in the definition of H this can be extended to H itself being undecidable. Thus \mathbf{FPD}^H is strictly bigger than \mathbf{FPD}^\emptyset .

7 Conclusion

We have introduced a number of classes of parameterized sets, defined by decidability properties of their slices. The classes form a chain

$$\mathbf{suFPD} \subset \mathbf{uFPD} \subset \mathbf{FPD}^\emptyset \subset \mathbf{nuFPD}.$$

To the left of this chain, there is also the class of monotonic sets in **suFPD**. Inside **suFPD** is an entire subhierarchy based on bounding functions. Classes of the form \mathbf{FPD}^H , where H is a parameterized set, exist between \mathbf{FPD}^\emptyset and **nuFPD**. These classes are inspired by parameterized analysis in complexity theory, chiefly by the class **FPT** of fixed-parameter tractable sets, and likewise come naturally with a fixed-parameter reducibility notion. Unique to the setting of decidability is the fact that **suFPD** and **uFPD** can be characterized by means of reducibility of the limits of convergent parameterized sets to the halting set. For **suFPD** the reducibility involved is truth-table reducibility or, equivalently, weak truth-table reducibility, whereas for **uFPD** the specific reducibility notion is Turing reducibility. Fixed-parameter complexity classes cannot be distinguished based on the reducibility of the limits of their members to the halting set. A set that is Turing reducible to the halting set can occur as the limit of a convergent parameterized set in any fixed-parameter complexity class.

For the computability of real numbers, we located some notable classes of real numbers in our parameterized framework. In doing so, we illustrated the dependence of the classification on the encoding of real numbers, considering both Dedekind cuts and binary expansions. Using properties of the above chain of parameterized decidability classes we were able to state a concise proof of a theorem that until now only had a convoluted published proof. This proof demonstrates the ability of parameterized analysis to subsume a tacit priority argument.

Lastly, we have shown that in **suFPD**, parameters give rise to a computable partial upper bound to the instance complexity of members of limit sets. Moreover, this upper bound is shown to upper bound the Kolmogorov complexity as well, infinitely often. The latter behavior is interesting because the class of parameterized sets for which it holds allows the specific upper bound to be as slowly increasing as any unbounded computable partial function. Similarly, as an upper bound to the instance complexity of members of a limit set, it is possible to exercise control over the tightness of the bound via the choice of a parameterized set. These upper bounds thus constitute a new take on the instance complexity conjecture.

Acknowledgements We are grateful for the supportive comments of an anonymous reviewer.

References

- [1] K. A. Abrahamson, R. G. Downey, and M. R. Fellows, Fixed-parameter tractability and completeness IV: On completeness for **W[P]** and **PSPACE** analogs, *Ann. Pure Appl. Log.* **73**(3), 235–276 (1995).
- [2] K. Ambos-Spies, K. Weihrauch, and X. Zheng, Weakly computable real numbers, *J. Complex.* **16**(4), 676–690 (2000).

- [3] M. Arslanov, Degree structures in local degree theory, in: Complexity, Logic, and Recursion Theory, edited by A. Sorbi, Lecture Notes in Pure and Applied Mathematics Vol. 187, (CRC Press, 1997), pp. 49–74.
- [4] H. Buhrman and P. Orponen, Random strings make hard instances, *J. Comput. Syst. Sci.* **2**(53), 261–266 (1996).
- [5] R. G. Downey and M. R. Fellows, Fixed-parameter tractability and completeness III: Some structural aspects of the W hierarchy, in: Complexity Theory, Current Research, edited by K. Ambos-Spies, S. Homer, and U. Schöning (Cambridge University Press, 1993), pp. 191–225.
- [6] R. G. Downey and M. R. Fellows, Fixed-parameter tractability and completeness I: Basic results, *SIAM J. Comput.* **24**(4), 873–921 (1995).
- [7] R. G. Downey and M. R. Fellows, Fixed-parameter tractability and completeness II: On completeness for $W[1]$, *Theor. Comput. Sci.* **141**(1), 109–131 (1995).
- [8] R. G. Downey and M. R. Fellows, Parameterized Complexity, Monographs in Computer Science (Springer, 1999).
- [9] R. G. Downey and D. R. Hirschfeldt, Algorithmic Randomness and Complexity, Theory and Applications of Computability (Springer, 2010).
- [10] R. L. Epstein, R. Haas, and R. L. Kramer, Hierarchies of sets and degrees below \emptyset' , in: Logic Year 1979–80, Proceedings of Seminars and of the Conference on Mathematical Logic held at the University of Connecticut, Storrs, Conn., November 11–13, 1979, edited by M. Lerman, J. H. Schmerl, and R. I. Soare, Lecture Notes in Mathematics Vol. 859, (Springer, 1981), pp. 32–48.
- [11] Y. L. Ershov, A hierarchy of sets I, *Algebr. Log.* **7**(1), 25–43 (1968).
- [12] Y. L. Ershov, On a hierarchy of sets II, *Algebr. Log.* **7**(4), 212–232 (1968).
- [13] J. Flum and M. Grohe, Parameterized Complexity Theory, Texts in Theoretical Computer Science (Springer, 2006).
- [14] M. E. Gold, Limiting recursion, *J. Symb. Log.* **30**(1), 28–48 (1965).
- [15] J. Goldsmith, D. Joseph, and P. Young, A note on bi-immunity and p -closeness of p -cheatable sets in \mathbf{P}/Poly , *J. Comput. Syst. Sci.* **46**(3), 349–362 (1993).
- [16] L. Hemaspaandra and L. Torenlvliet, Theory of Semi-Feasible algorithms (Springer Science & Business Media, 2013).
- [17] C. G. Jockusch, Semirecursive sets and positive reducibility, *Trans. Amer. Math. Soc.* **131**(2), 420–436 (1968).
- [18] M. Kummer, Kolmogorov complexity and instance complexity of recursively enumerable sets, *SIAM J. Comput.* **25**(6), 1123–1143 (1996).
- [19] M. Li and P. Vitányi, An Introduction to Kolmogorov Complexity and Its Applications, third edition, Texts in Computer Science (Springer, 2008).
- [20] P. Orponen, K. I. Ko, U. Schöning, and O. Watanabe, Instance complexity, *J. ACM* **41**(1), 96–121 (1994).
- [21] H. Putnam, Trial and error predicates and the solution to a problem of Mostowski, *J. Symb. Log.* **30**(1), 49–57 (1965).
- [22] H. G. Rice, Classes of recursively enumerable sets and their decision problems, *Trans. Amer. Math. Soc.* **74**(2), 358–366 (1953).
- [23] A. L. Selman, \mathbf{P} -selective sets, tally languages, and the behavior of polynomial time reducibilities on \mathbf{NP} , *Math. Syst. Theory* **13**(1), 55–65 (1979).
- [24] J. R. Shoenfield, On degrees of unsolvability, *Ann. Math.* **69**(3), 644–653 (1959).
- [25] X. Zheng, R. Rettinger, and R. Gengler, Ershov’s hierarchy of real numbers, in: Mathematical Foundations of Computer Science 2003, 28th International Symposium, MFCS 2003, Bratislava, Slovakia, August 25–29, 2003, Proceedings, edited by B. Rován and P. Vojtáš, Lecture Notes in Computer Science Vol. 2747, (Springer, 2003), pp. 681–690.