On the Information Carried by Programs About the Objects they Compute

Mathieu Hoyrup, Cristóbal Rojas

Noname manuscript No.
(will be inserted by the editor)

On the information carried by programs about the objects they compute

Mathieu Hoyrup and Cristóbal Rojas

the date of receipt and acceptance should be inserted later

Abstract In computability theory and computable analysis, finite programs can compute infinite objects. Such objects can then be represented by finite programs. Can one characterize the additional useful information contained in a program computing an object, as compared to having the object itself? Having a program immediately gives an upper bound on the Kolmogorov complexity of the object, by simply measuring the length of the program, and such an information cannot usually be derived from an infinite representation of the object. We prove that bounding the Kolmogorov complexity of the object is the only additional useful information. Hence we identify the exact relationship between Markov-computability and Type-2-computability. We then use this relationship to obtain several results characterizing the computational and topological structure of Markov-semidecidable sets.

This article is an extended version of [8], including complete proofs and a new result (Theorem 9).

Keywords Markov-computable · representation · Kolmogorov complexity · Ershov topology

Acknowledgements

This work was partially supported by Inria program "Chercheur invité". C.R. was partially supported by a FONDECYT grant No. 1150222, a Universidad

M. Hoyrup
Inria Nancy Grand Est
615 rue du jardin botanique, 54600 Villers-lès-Nancy, France
E-mail: hoyrup@inria.fr

C. Rojas
Departamento de Matemáticas, Universidad Andres Bello
República 220, Santiago, Chile.
E-mail: crojas@mat-unab.cl

Andres Bello grant No. DI-782-15/R and a BASAL grant No. PFB-03 CMM-Universidad de Chile.

1 Introduction

We assume that the reader is familiar with Turing machines and basic computability theory over the natural numbers. To define computability over infinite objects, one still uses Turing Machines but has to set up a way for them to access such objects. In any case, the input of the machine is a finite or infinite sequence of symbols written on the input tape and one has to choose a suitable way to describe infinite objects by such symbolic sequences. We now briefly describe the two main approaches that have been developed.

The first one was introduced and studied by Turing [22], Grzegorczyk [6], Lacombe [11] and later Kreitz and Weihrauch [23] and is nowadays known as Type-2-computability. In this model, the description itself is completely written on the input tape of the machine. At any time, the machine can read a finite portion of this description. We will call this the ***Type-2-model*** (the machine computes a functional of order type 2, i.e. taking a function of order type 1 as input). The second approach, promoted by the Russian school led by Markov [12,10], gives an alternative. In this model one restricts the action of the machine to operate on computable (infinite) objects only, in the sense that they have computable descriptions. Instead of having access to the description itself as in the Type-2-model, the machine here has access to a *program* computing a description. We will call this the ***Markov-model*** (it could also be called Tpye-1 model). These two approaches provide a priori different computability notions, and their comparison has been an important subject of study [15,13,20,9,1,4,14,7,21].

It is clear that the Markov-model is at least as powerful as the Type-2-model, so the question is: does it allow to compute strictly more than the Type-2-model? The answer depends on the input objects that we consider, and the algorithmic tasks we want to perform on them. The computational power of these models can therefore be classified according to these parameters. Table 1 summarizes the most celebrated results in this direction. The computable objects considered are the partial computable functions and the total computable functions. The algorithmic tasks considered are decidability and semidecidability of properties about these objects.

Table 1 Some celebrated results comparing Markov-computability to Type-2-computability.

Objects	Decidability	Semidecidability
Partial computable functions	Markov ≡ Type-2 *Rice*	Markov ≡ Type-2 *Rice−Shapiro*
Total computable functions	Markov ≡ Type-2 *Kreisel et al/Čeitin*	Markov > Type-2 *Friedberg*

Kreisel-Lacombe-Shœnfield/Čeitin's Theorem [9, 1] for instance, states that over total computable functions, Markov-decidability is equivalent to Type-2-decidability[1]. This means that the machine trying to decide a property, when provided with a program p for a function f, cannot do better than just running p to evaluate f. The machine gains no additional information about f from p. We note that Čeitin's version of this result shows that over the real line, Markov-computable functions and Type-2-computable functions coincide.

On the other hand, Friedberg [4] exhibited properties about total computable functions that are Markov-semidecidable but that are not Type-2-semidecidable. So that for semidecidability, a program p for a function f *does* give some additional information that can be exploited by the machine. The main question we raise in this paper is the following:

Can we characterize the additional useful information contained in a program computing an object, as compared to having the object itself ?

To get some intuition, consider the following fundamental difference between the two models. In the Type-2-model, at any given time only a finite portion of the description of x is provided, which corresponds to a finite approximation of x. Clearly, this approximation is also good for infinitely many other objects – all the ones that are "close enough" to x. In particular, *x is never completely specified.* In the Markov-model on the other hand, the program provided to the machine completely specifies x from the beginning of the calculation! This increases the predictive power of M, which might therefore be able to perform stronger calculations. The point is to understand in which situations this fact can be exploited. A trivial example is obtained when one considers the partially relativized setting: *every* function is Markov-computable relative to an appropriate (powerful enough) oracle. Whereas whatever oracle A we consider, Type-2-computable functions relative to A must always be continuous. This is a partial relativization as the programs used as Markov names are not relativized.

This observation takes us to another interesting point that separates the Markov-model from the Type-2-model, namely their topological structure. It is well known that Type-2-computability and topology are closely related: e.g. the Type-2-computable functions are exactly the effectively continuous ones, and the Type-2-semidecidable properties exactly correspond to the effectively open sets. The connection between Markov-computability and topology, on the other hand, appears to be much less clear. In particular, Friedberg's construction provides a Markov-semidecidable set which is not open (for the standard topology restricted to computable elements).

An obvious solution to relate Markov-computability to topology is to consider precisely the topology generated by all the Markov-semidecidable sets – the so called Ershov's topology. The question then becomes:

How do Markov-semidecidable sets look like? can we characterize Ershov's open sets ?

[1] In its original form due to Kreisel-Lacombe-Shœnfield, this theorem is stated for functionals.

In the present paper we make use of Kolmogorov Complexity to provide a fairly complete answer to these and other questions in different settings. Our main result is a characterization of the additional information provided by a program, when the class of objects considered are the computable points of an effective topological space. It can be informally stated as follows (see Section 3):

Theorem A *Over effective topological spaces, a program computing x provides as much information as (i) a description of x itself (ii) plus* ***any upper bound on the Kolmogorov complexity of x.***

Here, the Kolmogorov complexity $K(x)$ of a computable infinite object x is to be understood as the size of the shortest program computing a description of x (Kolmogorov complexity of infinite objects was first defined by Schnorr [17]). Obviously, any program for x trivially provides, in addition to a description, an upper bound on its Kolmogorov complexity. Theorem A says that this bound is all the exploitable additional information it provides.

Thus, we have a third model to deal with computable infinite objects. In this model, input x is presented to the machine as a pair (d, k), where d is a description for x and k a bound on the Kolmogorov complexity of x. We shall call this the ***K-model.*** In these terms, a particular case of Theorem A can be stated as follows: if $\mathcal{X}$, $\mathcal{Y}$ are *effective topological spaces* (not necessarily metric) and X_c, Y_c are the corresponding sets of computable points, then a function $f : X_c \to Y_c$ is Markov-computable if and only if it is K-computable.

A simple observation (see Proposition 1 below) shows that one can not in general compute a program for x from a K-description of x, meaning that the two notions are not fully equivalent. Despite this fact, Theorem A states that the same algorithmic tasks can be performed from these two representations. We consider a wide range of possible tasks: deciding a property of the represented object, semideciding a property, computing the image of the object under a function and also other weaker tasks. In proving this we make a fundamental use of the Recursion Theorem. Interestingly, although the Recursion Theorem does not relativize (a well known fact), Theorem A does in many cases.

The K-model also sheds light into the structure of the open sets of Ershov's topology, providing a nice characterization in terms of Kolmogorov complexity, at least in the particular case of the extended natural numbers $\overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$. We recall that Ershov's topology is generated by the Markov-semidecidable properties.

Theorem B *On the extended natural numbers, the Markov-semidecidable sets are unions of c.e. subsets of $\mathbb{N}$ and sets $\{n \in \mathbb{N} : K(n) < h(n)\} \cup \{\infty\}$ for some computable order h.*

Here $K(n)$ is (any version, like prefix or plain) of the usual Kolmogorov complexity of natural numbers.

With the same techniques, we are able to prove several other related results that are interesting on their own. For example, we show that there is no effective enumeration of the Markov-semidecidable sets of $\overline{\mathbb{N}}$ and that there is a Markov-semidecidable subset of $\{0,1\}^{\mathbb{N}}$ that is not Σ^0_2.

Finally, in the search of the limitations of our techniques, we turn our attention to more general spaces and analyze functions with values on topological spaces that have an admissible representation but are not countably-based. In particular, when this is the space of open subsets of Baire space $\mathcal{O}(\mathbb{B})$, we show that Markov-computability can be strictly stronger than K-computability:

Theorem C *For functionals taking partial computable functions as inputs and elements of $\mathcal{O}(\mathbb{B})$ as outputs, one has that:*

$$\textit{Markov-computability} > \textit{K-computability} > \textit{Type-2-computability.}$$

One of the main questions that remains open is whether the first strict inequality in Theorem C holds if we replace the *partial* computable functions by the *total* ones. The situation is summarized in Table 2.

Table 2 Some results comparing Markov-computability, K-computability, and Type-2-computability. $\mathbb{S} = \{\bot, \top\}$ is the Sierpiński space whose topology is generated by $\{\top\}$.

Space $\mathcal{X}$	Semidecidable	$\emptyset'$-Semidecidable	F: $\mathcal{X} \to \mathcal{O}(\mathbb{B})$
$\mathbb{S}$	Markov $\equiv$ K $\equiv$ Type-2	Markov $>$ K $\equiv$ Type-2	Markov $>$ K $\equiv$ Type-2
Part. func.	Markov $\equiv$ K $\equiv$ Type-2	Markov $>$ K $\equiv$ Type-2	Markov $>$ K $>$ Type-2
Tot. func.	Markov $\equiv$ K $>$ Type-2	Markov $\equiv$ K $>$ Type-2	Markov **?** K $>$ Type-2

The paper is organized as follows. We start by providing the basic notions and definitions in Section 2. In Section 3 we introduce the K-model and present our main results. Section 4 contains several results that shed light on the structure of Markov-semidecidable sets and in Section 5 we present the announced negative results. Finally, Section 6 contains a list of related problems for possible future work.

2 Background

2.1 Notations and basic definitions.

We assume the reader is familiar with computability theory. Let $\{\varphi_e\}_{e\in\mathbb{N}}$ be an effective enumeration of the set of computable partial functions. We denote by $P_c(\mathbb{N})$ the collection of c.e. subsets of $\mathbb{N}$ and $W_e = \mathrm{dom}(\varphi_e)$ the induced effective enumerations of its elements. If $A \in P_c(\mathbb{N})$, an ***index*** of A is a number e such that $W_e = A$. If A is a c.e. set, implicitly given by an index, $A[s]$ is the finite subset of A enumerated by stage s, so that $A[s] \subseteq A[s+1]$ and $A = \bigcup_s A[s]$. We use the notation $A[\text{at } s] = A[s] \setminus A[s-1]$ if $s \geq 1$ and $A[\text{at } 0] = A[0]$. If F is a finite subset of $\mathbb{N}$ then $[F]$ is the collection of supersets of F. $\mathbb{B} = \mathbb{N}^{\mathbb{N}}$ will denote Baire space. We will often consider two particular elements of the Cantor space $\{0,1\}^{\mathbb{N}}$ of infinite binary sequences: if $a \in \{0,1\}$ then a^ω denotes the infinite sequence with a's only.

We will use the Recursion Theorem, which says that one can define a program computing a function (or more generally other classes of objects, like c.e. sets) and use the program in its own definition. See [16].

Theorem 1 (Recursion Theorem) *For every total computable function f, there exists e such that $\varphi_e = \varphi_{f(e)}$ and $W_e = W_{f(e)}$. Moreover, e can be computed from an index of f.*

2.2 Effective topological spaces.

An ***effective topological space*** is a tuple $(\mathcal{X}, \tau, \mathcal{B})$ where $(\mathcal{X}, \tau)$ is a non-empty T_0 topological space, $\mathcal{B} = \{\mathcal{B}_i\}_{i\in\mathbb{N}}$ is numbered basis such that there exists a computable function $f : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ satisfying $\mathcal{B}_i \cap \mathcal{B}_j = \bigcup_{k\in W_{f(i,j)}} \mathcal{B}_k$.

Given an effective topological space $\mathcal{X}$, the standard representation of points is defined as a surjective map $\rho : \mathrm{dom}(\rho) \subseteq \mathbb{B} \to \mathcal{X}$ satisfying $\rho(f) = x$ whenever $\{f(n) : n \in \mathbb{N}\} = \{i : x \in \mathcal{B}_i\}$. We will call any $f \in \rho^{-1}(x)$ a ***Type-2-name*** of x. A point $x \in \mathcal{X}$ is ***computable*** if it has a computable Type-2-name. We denote by $\boldsymbol{X_c}$ the set of computable points.

The countable set X_c has a canonical numbering ν defined by $\nu(e) = x$ if φ_e is a name of x. We will call such an e a ***Markov-name*** of x and we write $x_e = x$. Another numbering η of X_c is ***admissible*** if it is equivalent to the canonical numbering ν in the sense that there exist partial computable functions f and g such that $\nu = \eta \circ f$ on $\mathrm{dom}(\nu)$ and $\eta = \nu \circ g$ on $\mathrm{dom}(\eta)$. We will often use the admissible numbering η of X_c defined by $\eta(e) = x$ whenever $W_e = \{i \in \mathbb{N} : x \in \mathcal{B}_i\}$.

Example 1 Let $\mathbb{B} = \mathbb{N}^{\mathbb{N}}$ be the Baire space. For each finite sequence u, let $[u]$ be the set of infinite extensions of u, called a cylinder. We endow $\mathbb{B}$ with the topology generated by the cylinders, which is an effective topology. The standard numbering φ_e of partial computable functions, restricted to the indices of total functions is an admissible numbering of $\mathbb{B}_c$.

Example 2 Let $\mathcal{P}(\mathbb{N})$ be the space of subsets of $\mathbb{N}$. For each finite set $F \subseteq \mathbb{N}$, let $[F]$ be the set of supersets of F. We endow $\mathcal{P}(\mathbb{N})$ with the Scott topology, generated by the sets $[F]$, which is an effective topology. The standard numbering $W_e = \mathrm{dom}(\varphi_e)$ of c.e. sets is an admissible numbering of $P_c(\mathbb{N})$.

To facilitate the reading of the paper, we will use the normal font (as in A, N, U) when working on the space X_c, and the script font (as in $\mathcal{A}, \mathcal{N}, \mathcal{U}$) when working on $\mathcal{X}$.

2.2.1 Type-2-computability and Markov-computability

Let $(\mathcal{X}, \tau, \mathcal{B})$ and $(\mathcal{Y}, \tau', \mathcal{B}')$ be effective topological spaces. In what follows R stands for both *Type-2* and *Markov*. A set $A \subseteq X_c$ is R-***semidecidable*** if there is a Turing machine M which, when provided with an R-name of x, halts if and only if $x \in A$. A function $f : X_c \to Y_c$ is R-***computable*** if there is a Turing machine M which, when provided with an R-name of x, writes an R-name for $f(x)$ on its one-way output tape. It is not hard to see that a function $f : X_c \to Y_c$ is R-computable if and only if the sets $f^{-1}(\mathcal{B}'_i)$ are uniformly R-semidecidable.

The Markov-computability notions do not depend on the choice of the admissible numbering.

Remark 1 It is worth noting that for a function $f : X_c \to Y_c$, being Markov-computable is equivalent to having a machine M which, provided with a Markov-name of x, outputs a *Type-2-name* of $f(x)$. Indeed, combining the program for x with the program for M gives a program for $f(x)$. We also note that a function $f : X_c \to Y_c$ which is Type-2-computable does not necessarily extend to a Type-2-computable function $\overline{f} : \mathcal{X} \to \mathcal{Y}$.

Type-2-computability and topology are closely related. A set $\mathcal{U} \subseteq \mathcal{X}$ is an ***effective open set*** if there exists $e \in \mathbb{N}$ such that $\mathcal{U} = \bigcup_{i \in W_e} \mathcal{B}_i$. If $A = \mathcal{U} \cap X_c$, we will then say that A is effectively open ***in*** $\boldsymbol{X_c}$. The connection is established by the following result (see [23]).

Theorem 2 *A set $A \subseteq X_c$ is Type-2-semidecidable if and only if it is effectively open in X_c. Therefore, a function $f : X_c \to Y_c$ is Type-2-computable if and only if it is effectively continuous, i.e. the sets $f^{-1}(\mathcal{B}'_i)$ are uniformly effectively open in X_c.*

As mentioned in the introduction, in order to have an analogous result for Markov-computability, we have to use Ershov's topology on X_c, which may be different from the topology of $\mathcal{X}$ restricted to X_c.

3 Main results

In this section, $(\mathcal{X}, \tau, \mathcal{B})$ is always an effective topological space and $X_c \subseteq \mathcal{X}$ is the set of computable elements. We start by explaining the main idea behind our results. Let $x \in X_c$ be a fixed element. From a machine Type-2-semideciding a set A containing x, one can compute a neighborhood $\mathcal{N}$ of x such that for every element $y \in X_c$ the following implication holds:

$$y \in \mathcal{N} \implies y \in A. \tag{1}$$

Now assume that A has the weaker property of being Markov-semidecidable, and still contains x. From a machine Markov-semideciding A one cannot in general compute such a neighborhood, which may not exist as shown by Friedberg's example. However, from the Markov-name of any element $y \in X_c$ one can still compute a neighborhood $\mathcal{N}_y$ of x such that implication (1) holds. Further, as a finite intersection of neighborhoods is still a neighborhood, one can compute a neighborhood $\mathcal{N}$ satisfying implication (1) for all y in a given finite set. Using this argument we can show that the problem $x \in A$ can be Type-2-semidecided *as soon as we know, in addition, a finite list of programs containing at least one for x.* This additional information is equivalent to having any upper bound on the Kolmogorov complexity of x, which leads us to the notion of K-computability that we now introduce.

3.1 K-computability

Definition 1 The ***Kolmogorov complexity K(x)*** of a computable element $x \in X_c$ is the length of a shortest program computing a Type-2-name of x.

As in the case of finite objects, there are several ways of formalizing the notion of a program computing an infinite object, leading to several variants of the notion of Kolmogorov complexity. However the choice of a formulation will not make any difference so we do not need to specify the definition any further.

Definition 2 A ***K-name*** of a computable element $x \in X_c$ consists of a pair (f, k) where f is a Type-2-name of x and $k \geq K(x)$.

This representation does not change the notion of computable element: the elements that have computable K-names are exactly the computable elements. However this representation is useful when considering computations that take K-names as inputs.

Remark 2 Note that k is only an upper bound on the Kolmogorov complexity of x and not necessarily of f, which may even be non computable. Note also that knowing any such k is effectively equivalent to knowing any upper bound on a Markov-name of x. This is what we will rather use in our proofs.

The ***K-computability*** notions are defined in the same way as in the previous section. We will denote by $X_c(k)$ the set of computable elements whose Kolmogorov complexity is at most k. Note that $X_c = \bigcup_k X_c(k)$ and that K-computability is the same as Type-2-computability on $X_c(k)$, uniformly in k. In particular, a set $A \subseteq X_c$ is K-semidecidable iff there exists uniformly effective open sets $\mathcal{U}_k$ such that $A \cap X_c(k) = \mathcal{U}_k \cap X_c(k)$.

Thus, for each notion of computability we have so far three versions, depending on the way the objects are represented.

It is clear that one can compute K-names from Markov-names. An important first observation is the fact that the converse does not necessarily hold. In other words, the representations underlying Markov-computability and K-computability are not equivalent.

Proposition 1 *In general, it is not possible to compute Markov-names from K-names.*

Proof We are on $\{0,1\}^{\mathbb{N}}_c$. We denote by x_e the computable sequence with Markov-name e. There is a computable function $f:\mathbb{N}\to\mathbb{N}$ such that

$$x_{f(i)} = \begin{cases} 0^\omega & \text{if } \varphi_i(i) \text{ does not halt,} \\ 0^t 1^\omega & \text{if } \varphi_i(i) \text{ halts in time } t. \end{cases}$$

Assume that there is an oracle Turing machine M that converts each K-name of a sequence into a program computing the sequence (a Markov-name). There is a computable function $k:\mathbb{N}\to\mathbb{N}$ such that $k(i)$ is an upper bound on the complexity of $x_{f(i)}$ and of 0^ω. For each i, $(0^\omega, k(i))$ is a K-name of 0^ω so M must halt on it and output an index of 0^ω. Let u_i be the number of bits of 0^ω read by the machine when it halts. If $x_{f(i)}$ starts with u_i zeroes then M will output the same on $(x_{f(i)}, k(i))$, which is a K-name of $x_{f(i)}$, which implies that $x_{f(i)}$ must be 0^ω.

As a result, for each i, if $\varphi_i(i)$ halts it must halt in time u_i, which enables one to decide the Halting problem, a contradiction.

In other words, a machine cannot convert a computable binary sequence x and a finite list of programs, one of them computing x, into a program computing x. Actually the proof shows that it is true even when the list contains two programs only.

One can show that on Cantor space, Markov-names are limit-computable (can be *learned*) from K-names: given x and $k\geq K(x)$, one can compute a sequence of natural numbers converging to an index of x (this problem was investigated in the context of inductive inference [3]). To see this, given a finite list of programs build a new program that on each input $n\in\mathbb{N}$ runs all the programs of the list in parallel and outputs the result of the first program that halts. Now, start proposing the program (or index) built from the finite list of programs of size $\leq k$. Progressively remove from the list the programs that output something different from x. Each time one of the programs is rejected, propose a new program built from the smaller list. In finite time, the list will stabilize and the proposed program will compute x.

One can moreover show that relative to the halting set, Markov-names are uniformly computable from K-names (the previous procedure is a limit-computation, hence a computation relative to the halting set).

A c.e. set, however, cannot be learned. Actually one can prove a stronger statement.

Proposition 2 *There is no Turing functional Φ that, given an index e and a Type-2-name of a set W which is either $\mathbb{N}$ or W_e, computes a sequence of numbers converging to an index of W.*

Proof Here a Type-2-name of a c.e. set W is a function $t:\mathbb{N}\to\mathbb{N}$ such that $W=\{n:\exists k, t(k)=n+1\}$. When we say that a number n appears in t, we mean that $n+1=t(k)$ for some k.

Assume that such a Φ exists. Using the Recursion theorem, we define an index a in the way described below. At the same time we enumerate a c.e. set W_a and we build an oracle t.

We will define W_a as a union $\bigcup_i F_i$ where $(F_i)_{i\in\mathbb{N}}$ is a computable sequence of finite sets and $F_i \subseteq F_{i+1}$. At the same time we will define a Type-2-name t pf W_a as the limit of a computable sequence of finite strings t_i such that t_{i+1} extends t_i. The finite string t_i contains exactly the elements of F_i.

The strings t_i are built such that $\Phi^{t_i}(a)$ outputs a (finite or infinite) sequence of indices of length at least i and changing after position i, before reaching the end of t_i. Hence $\Phi^t(a)$ outputs an infinite sequence of indices that does not converge, contradicting the assumptions about Φ.

To define a using the Recursion Theorem (Theorem 1), given e we define a c.e. set $W_{f(e)}$. The construction being uniform, f is a computable function and we take for a a fixed-point of f, i.e. a satisfies $W_a = W_{f(a)}$.

Let $e \in \mathbb{N}$. We start with $F_0 = \emptyset$ and t_0 is the empty string. Assume F_i and t_i have been defined. Look for a finite extension u of t_i such that before reaching the end of u, $\Phi^u(e)$ outputs a sequence containing an index j occurring after position i in the sequence, such that W_j contains some number that is not in u (such a u must be found: on a representation of $\mathbb{N}$ starting with t_i, Φ must output an infinite sequence converging to an index j of $\mathbb{N}$, let u be the finite part of the oracle that is read when such an index is output after position i). Let F_{i+1} be the set of elements enumerated in u. Run $\Phi^{u0^\omega}(e)$ and look for a number $k \neq j$ appearing later than j in the output sequence. If such a k is found then let t_{i+1} be the part of the oracle that is used in the computation of k, otherwise $t_{i+1}, t_{i+2}, \dots$ and $F_{i+2}, F_{i+3}, \dots$ are undefined.

As the construction is effective, there exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that $W_{f(e)}$ is the union of the sets F_i that are defined. Let a be a fixed-point of f, i.e. $W_a = W_{f(a)}$. We prove that the sequences t_i and F_i are entirely defined. Assume otherwise that for some i, t_i is defined but not t_{i+1}. One has $W_a = W_{f(a)} = F_{i+1}$ and $u0^\omega$ is a Type-2-name of W_a, so $\Phi^{u0^\omega}(a)$ should output an infinite sequence converging to an index k of W_a. As W_j contains a number which is not in u, $W_j \neq W_a$ so $k \neq j$, and occurrences of k appear later than j in the sequence. As a result, the search procedure will terminate and t_{i+1} will be defined, contradicting the assumption.

Now by construction $\Phi^{t_i}(a)$ outputs a sequence that changes after position i (from j to $k \neq j$) so the sequence produced by $\Phi^t(a)$ does not converge. We get a contradiction with the assumption about Φ, so no such Φ exist.

The rest of this section is devoted to show that, despite the facts above, the notions of Markov-computability and K-computability are indeed equivalent to a large extent.

3.2 Equivalence between Markov-computability and K-computability

The following Lemma contains the main technical arguments.

Lemma 1 *Let A be a c.e. subset of $\mathbb{N}$. There exist uniformly effective Scott open sets $\mathcal{U}_k \subseteq \mathcal{P}(\mathbb{N})$, such that for every c.e. set E the following hold:*

i) if all the indices of E belong to A then $E \in \mathcal{U}_k$ for every k, *ii) if no index of E belongs to A then $E \notin \mathcal{U}_k$ for every $k \geq K(E)$.*

The argument is uniform: the open sets $\mathcal{U}_k$ are effective, uniformly in a c.e. index of A.

We first reproduce the proof given in [8] which uses the Recursion Theorem.

Proof Using the Recursion theorem, there is a computable function $e(a, b)$ such that for all $a, b \in \mathbb{N}$,

$$W_{e(a,b)} = \begin{cases} W_a & \text{if } e(a,b) \notin A, \\ W_a[t] \cup W_b & \text{if } e(a,b) \in A[\text{at } t]. \end{cases}$$

Let $k \in \mathbb{N}$. We define an effective open set $\mathcal{U}_k$. Compute b_k such that every element whose complexity is less than k has an index less than b_k. If a is such that for all $b \leq b_k$, $e(a,b) \in A$ then let t be minimal such that $e(a,b) \in A[t]$ for all $b \leq b_k$, enumerate $[W_a[t]]$ into $\mathcal{U}_k$.

We now check the two announced conditions. *i)* Let $E \subseteq \mathbb{N}$ be a c.e. set. Assume that every index of E belongs to A and let a be an index of E. For all b, $e(a,b) \in A$ (otherwise $e(a,b)$ is an index of $W_a = E$ but $e(a,b) \notin A$, contradiction), so $\mathcal{U}_k$ contains $[W_a[t]]$ for some t, which contains E. *ii)* Assume that $K(E) \leq k$, that no index of E belongs to A and that $E \in \mathcal{U}_k$. Let $b \leq b_k$ be an index of E. As $E \in \mathcal{U}_k$, E belongs to some $[W_a[t]]$ enumerated into $\mathcal{U}_k$ (here a is not the same as above and is not assumed to be an index of E). As $e(a,b) \in A$, $W_{e(a,b)} = W_a[t'] \cup W_b$ for some $t' \leq t$. As $W_a[t'] \subseteq W_a[t] \subseteq W_b$, $e(a,b)$ is an index of E that belongs to A, contradicting the assumption.

We now present an alternative proof that does not use the Recursion Theorem.

Proof There is a computable function $e(a,b,c)$ such that

$$W_{e(a,b,c)} = \begin{cases} W_a & \text{if } \varphi_c(c) \text{ does not halt}, \\ W_a[t] \cup W_b & \text{if } \varphi_c(c) \text{ halts in time } t. \end{cases}$$

Given $k \in \mathbb{N}$ we define an effective open set $\mathcal{U}_k$. Compute b_k such that every element whose complexity is less than k has an index less than b_k. If a is such that for all $b \leq b_k$, there exists $c = c(a,b)$ such that $e(a,b,c) \in A$ and $\varphi_c(c)$ halts then let t be the maximal halting time of $\varphi_c(c)$ for $c = c(a,b)$ with $b \leq b_k$, enumerate $[W_a[t]]$ into $\mathcal{U}_k$.

We now check the two announced conditions. Let $E \subseteq \mathbb{N}$ be a c.e. set.

i) Assume that every index of E belongs to A and let a be an index of E. Let us prove by contradiction that for each $b \in \mathbb{N}$ there exists c such that $e(a,b,c) \in A$ and $\varphi_c(c)$ halts, which implies that W_a is contained in $\mathcal{U}_k$ for all k. Let us assume that it is false for some $b \in \mathbb{N}$, i.e. that for each c such that $\varphi_c(c)$ halts, $e(a,b,c) \notin A$. Observe that for each c such that $\varphi_c(c)$ does not halt, $e(a,b,c)$ is an index of E so it belongs to A. Hence the complement of the halting set is many-one reducible to A via the function $c \mapsto e(a,b,c)$, which is a contradiction.

ii) Assume that $K(E) \leq k$, that no index of E belongs to A and that $E \in \mathcal{U}_k$. Let $b \leq b_k$ be an index of E. As $E \in \mathcal{U}_k$, E belongs to some $[W_a[t]]$ enumerated into $\mathcal{U}_k$ (here a is not the same as above and is not assumed to be an index of E). There is c such that $e(a,b,c) \in A$ and $\varphi_c(c)$ halts in $t' \leq t$ steps. One has $W_{e(a,b,c)} = W_a[t'] \cup W_b = W_b$ as $W_a[t'] \subseteq W_a[t] \subseteq W_b$, hence $e(a,b,c)$ is an index of $W_b = E$ that belongs to A, contradicting the assumption.

We now state the main explicit versions of Theorems A and B.

Theorem 3 *Let $\mathcal{X}$ be an effective topological space. A set $A \subseteq X_c$ is Markov-semidecidable iff it is K-semidecidable. The equivalence is uniform.*

Proof Every effective topological space is Type-2-computably homeomorphic to a subspace of $\mathcal{P}(\mathbb{N})$: to $x \in \mathcal{X}$, associate $\{i \in \mathbb{N} : x \in \mathcal{B}_i\}$ where $\mathcal{B}_i$ is the canonical enumeration of the basis of $\mathcal{X}$. Hence we can assume that $\mathcal{X}$ is a subspace of $\mathcal{P}(\mathbb{N})$. Let $I \subseteq \mathbb{N}$ be a c.e. set such that for all $e \in \mathbb{N}$ for which $W_e \in X_c$, it holds $W_e \in A \iff e \in I$. Each c.e. set $E \in X_c$ either has all its indices in I or has no index in I, so the effective open sets $\mathcal{U}_k$ provided by Lemma 1 coincide with A on the set of elements of X_c whose complexity is at most k. Now, a machine K-semideciding A works as follows: given a Type-2-name of $E \in X_c$ and $k \geq K(E)$, it tests whether $E \in \mathcal{U}_k$ and halts in this case only.

Corollary 1 *Let $\mathcal{X}, \mathcal{Y}$ be effective topological spaces. A function $f : X_c \to Y_c$ is Markov-computable iff f is K-computable. The equivalence is uniform.*

Proof Let $\mathcal{B}_i$ be the numbered basis of $\mathcal{Y}$. f is Markov-computable iff the sets $f^{-1}(\mathcal{B}_i)$ are uniformly Markov-semidecidable iff these sets are uniformly K-semidecidable (Theorem 3) iff f is K-computable.

We now show that the argument in the proof of Lemma 1 can be extended from semidecidability to weaker classes of properties, showing that for most algorithmic tasks, the additional information given by programs is indeed just an upper bound on the Kolmogorov complexity.

3.2.1 Hierarchies.

Let $\mathcal{X}$ be an effective topological space. We consider the finite levels of the effective Borel hierarchy, defined as follows. The class Σ^0_1 consists of the effective open sets. The class Σ^0_{n+1} consists of the effective unions of differences of Σ^0_n-sets (in a Polish space it is equivalent to the more common effective unions of Π^0_1-sets). The class Π^0_n consists of complements of Σ^0_n-sets. The class Δ^0_n is the intersection of Σ^0_n and Π^0_n. Inside the class Δ^0_2 we consider the finite levels of the effective difference hierarchy. For $n \in \mathbb{N}$, an element of the class $\mathcal{D}_n$ is the difference of n effective open sets $\mathcal{U}_0 \supseteq \ldots \supseteq \mathcal{U}_{n-1}$, i.e. $(\mathcal{U}_0 \setminus \mathcal{U}_1) \cup \ldots \cup (\mathcal{U}_{n-2} \setminus \mathcal{U}_{n-1})$ if n is even and $(\mathcal{U}_0 \setminus \mathcal{U}_1) \cup \ldots \cup \mathcal{U}_{n-1}$ if n is odd. We denote this difference by $D_n(\mathcal{U}_0, \ldots, \mathcal{U}_{n-1})$. In the case $\mathcal{X} = \mathbb{N}$ with the discrete topology, the effective Borel hierarchy is exactly the arithmetical hierarchy, the class $\mathcal{D}_n$ of the effective difference hierarchy is exactly the class of n-c.e. sets.

Theorem 4 *A set $A \subseteq X_c$ is Markov-n-c.e. iff it is K-n-c.e.*

More precisely, it means that the following statements are equivalent, where $\nu : \mathrm{dom}(\nu) \to X_c$ is the standard numbering of X_c:

- there exist c.e. sets $I_0, \dots, I_{n-1} \subseteq \mathbb{N}$ such that
$$\nu^{-1}(A) = D_n(I_0, \dots, I_{n-1}) \cap \mathrm{dom}(\nu),$$
- there exist uniformly effective open sets $\mathcal{U}_0^k, \dots, \mathcal{U}_{n-1}^k$ such that for all k,
$$A \cap X_c(k) = D_n(\mathcal{U}_0^k, \dots, \mathcal{U}_{n-1}^k) \cap X_c(k).$$

Proof One direction is trivial: if a set is K-n-c.e. then it is Markov-n-c.e., as every Markov-name can be converted into a K-name.

We prove the other direction. Again we can assume w.l.o.g. that $\mathcal{X}$ is a subspace of $\mathcal{P}(\mathbb{N})$. Let $I_0 \supseteq I_1 \supseteq \dots \supseteq I_{n-1}$ be c.e. sets such that if $W_e \in X_c$ then $W_e \in A \iff e \in D_n(I_0, \dots, I_{n-1})$. It is convenient to define $I_n = \emptyset$, so that $e \in D_n(I_0, \dots, I_{n-1})$ if and only if $e \in I_0$ and for each odd $i < n$, $e \in I_i$ implies $e \in I_{i+1}$. We denote a tuple $(a_0, \dots, a_n) \in \mathbb{N}^{n+1}$ by $\overline{a}$.

Using the Recursion theorem, we define a computable function $e(\overline{a})$ in the following way. Let
$$W_{e(\overline{a})} = \bigcup_{j<i} W_{a_j}[t_j] \cup W_{a_i},$$
where

- $i \leq n$ is minimal such that $e(\overline{a}) \notin I_i$,
- for $j < i$, t_j is such that $e(\overline{a}) \in I_i[\text{at } t_j]$.

The enumeration of $W_{e(\overline{a})}$ is indeed effective. Start with $i = 0$ and enumerate W_{a_0}. When one discovers that $e(\overline{a}) \in I_i$, stop enumerating W_{a_i}, start enumerating $W_{a_{i+1}}$ and increment i.

Given k, we define effective open sets $\mathcal{U}_0^k, \dots, \mathcal{U}_{n-1}^k$. Let b_k be an upper bound on the indices of elements whose Kolmogorov complexity is at most k. If a_0 is such that for all $a_1, \dots, a_n \leq b_k$, $e(a_0, \dots, a_n) \in I_0$ then let t_0 be minimal such that all these numbers belong to $I_0[t_0]$ and enumerate $[W_{a_0}[t_0]]$ in $\mathcal{U}_0^k$. By induction, let $1 \leq i < n$ and assume $a_0, \dots, a_{i-1}$ have been accepted with $t_0, \dots, t_{i-1}$. If a_i is such that for all $a_{i+1}, \dots, a_n \leq b_k$, $e(a_0, \dots, a_n) \in I_i$ then let t_i be minimal such that all these numbers belong to $I_i[t_i]$ and enumerate $[W_{a_0}[t_0] \cup \dots \cup W_{a_i}[t_i]]$ in $\mathcal{U}_i^k$. For convenience we also define $\mathcal{U}_n^k = \emptyset$.

Let $E \in X_c(k)$. We now prove that $E \in A \iff E \in D_n(\mathcal{U}_0, \dots, \mathcal{U}_{n-1})$.

(i) We show that if $E \in A$ then $E \in \mathcal{U}_0^k$ and for all odd $i < n$, if $E \in \mathcal{U}_i^k$ then $E \in \mathcal{U}_{i+1}^k$. Assume $E \in A$ and let a_0 be an index of E. For all $a_1, \dots, a_n$, $e(\overline{a}) \in I_0$ (otherwise $e(\overline{a})$ is an index of E that does not belong to I_0) so $\mathcal{U}_0$ contains some $[W_{a_0}[t_0]]$ which contains E. Let $i < n$ be odd. If $E \in \mathcal{U}_i^k$ then there is some $[W_{a_0}[t_0] \cup \dots \cup W_{a_i}[t_i]]$ (not necessarily the same a_0 as before) containing E enumerated in $\mathcal{U}_i^k$. Let $a_{i+1} \leq b_k$ be an index of E. For all $a_{i+2}, \dots, a_n$, $e(\overline{a}) \in I_i$ hence $e(\overline{a})$ must belong to I_{i+1}, otherwise it is an index of E but does not belong to $D_n(I_0, \dots, I_{n-1})$. As a result, some neighborhood of E is enumerated in $\mathcal{U}_{i+1}^k$.

(ii) We now show that if $E \in D_n(\mathcal{U}_0^k, \dots, \mathcal{U}_{n-1}^k)$ then $E \in A$. Let i be maximal such that $E \in \mathcal{U}_i^k$: $E \in D_n(\mathcal{U}_0^k, \dots, \mathcal{U}_{n-1}^k)$ means that i is even. As $E \in \mathcal{U}_i^k$, E is in some $[W_{a_0}[t_0] \cup \dots \cup W_{a_i}[t_i]]$ enumerated in $\mathcal{U}_i^k$. Let $a_{i+1} \leq b_k$ be an index of E. As $E \notin \mathcal{U}_{i+1}^k$, there exist $a_{i+2}, \dots, a_n \leq b_k$ such that $e(a_0, \dots, a_n) \notin I_{i+1}$, otherwise some $[W_{a_0}[t_0] \cup \dots \cup W_{a_i}[t_i] \cup W_{a_{i+1}}[t_{i+1}]]$ would be enumerated in $\mathcal{U}_{i+1}^k$ which would contain E, a contradiction. But then $e(\overline{a})$ is an index of E that belongs to $I_i \setminus I_{i+1} \subseteq D_n(I_0, \dots, I_{n-1})$, hence $E \in A$.

Friedberg's example shows a difference between Markov-semidecidability and Type-2-semidecidability on the Baire space, or the space of total functions. Rice-Shapiro theorem shows that such a difference does not hold on the space of partial functions and on $\mathcal{P}(\mathbb{N})$. One has to consider 2-c.e. sets rather than semidecidable sets to see a difference between the two computation models on $\mathcal{P}(\mathbb{N})$. Indeed, Selivanov [19] constructed a Markov-2-c.e. subset of $\mathcal{P}(\mathbb{N})$ that is not even Π_2^0. However Grassin [5] proved that every Markov-n-c.e. subset of $\mathcal{P}(\mathbb{N})$ always lies in the corresponding level of the *non-effective* difference hierarchy, i.e. it is a difference $D_n(\mathcal{U}_0, \dots, \mathcal{U}_{n-1})$ of n open sets, which by the result of Selivanov are not necessarily *effective* open sets. Selivanov's result also implies that a Markov-2-c.e. set is not in general the difference of two Markov-semidecidable sets.

In the following theorem, we need to assume an additional property on the space $\mathcal{X}$. Namely, that the domain of the standard representation on $\mathcal{X}$ is a Π_2^0 set. This is the case for example for the so called *quasi-Polish spaces* (see [2]).

Theorem 5 *Assume that the domain of the standard representation of $\mathcal{X}$ is Π_2^0. A set $A \subseteq X_c$ is Markov-Σ_2^0 iff it is K-Σ_2^0.*

More precisely, it means that the following statements are equivalent:

- there exists a Σ_2^0 set $I \subseteq \mathbb{N}$ such that

$$\nu^{-1}(A) = I \cap \mathrm{dom}(\nu),$$

- there exist uniformly effective open sets $\mathcal{U}_n^k, \mathcal{V}_n^k$ such that for all k,

$$A \cap X_c(k) = \bigcup_n (\mathcal{U}_n^k \setminus \mathcal{V}_n^k) \cap X_c(k).$$

Proof We show that if A is Markov-Π_2^0 then A is K-Π_2^0, which is equivalent to the statement by replacing A with its complement. We use the numbering $x_e = x$ if W_e is the set of indices of basic neighborhoods of x. Let $I = \bigcap_n I_n \subseteq \mathbb{N}$ be Π_2^0 (I_n are uniformly c.e.) such that if $\nu^{-1}(A) = I \cap \mathrm{dom}(\nu)$. The assumption about the space implies that $\mathrm{dom}(\nu)$ is a Π^0_2-set $D = \bigcap_n D_n \subseteq \mathbb{N}$, where D_n are uniformly c.e. sets.

Given i, let $C_i = \{x_i\}$ if $i \in I \cap D$, $C_i = \emptyset$ otherwise. C_i is Π^0_2, uniformly in i. Indeed, for each n, define the uniformly effective open sets

$$(\mathcal{U}_{2n}, \mathcal{V}_{2n}) = \begin{cases} (B_n, \emptyset) & \text{if } n \notin W_i, \\ (X, B_n) & \text{if } n \in W_i, \end{cases}$$

and

$$(\mathcal{U}_{2n+1}, \mathcal{V}_{2n+1}) = \begin{cases} (X, \emptyset) & \text{if } i \notin I_n \cap D_n, \\ (X, X) & \text{if } i \in I_n \cap D_n. \end{cases}$$

One has $C_i = \bigcap_n (\mathcal{U}_n \setminus \mathcal{V}_n)^c$ (we use the fact that $\mathcal{X}$ is T_0).

Given $k \in \mathbb{N}$, compute b_k such that every element of Kolmogorov complexity at most k has an index $\leq b_k$. The set $\bigcup_{i \leq b_k} C_i$ is Π^0_2, uniformly in k. This set intersected with $X_c(k)$ is exactly $A \cap X_c(k)$.

Remark 3 In case $\mathcal{X}$ is a Polish space, the sets $\mathcal{U}^k_n$ are not needed and therefore the last part of the statement reads $A \cap X_c(k) = \bigcup_n (\mathcal{X} \setminus \mathcal{V}^k_n) \cap X_c(k)$.

4 Structure of Markov-semidecidable sets

Here we provide several results that shed light on the computational and topological structure of Markov-semidecidable sets. Our first result shows that Markov-semidecidable sets share some of the nice properties of Type-2-semidecidable sets.

Proposition 3 *Assume that $\mathcal{X}$ contains a dense computable sequence. Given a Markov-semidecidable set A, it is semi-decidable whether A is non-empty. If A is non-empty, one can compute a sequence of points $\{x_i\} \subseteq A$ which is dense in A.*

Proof Let I be c.e. subset of $\mathbb{N}$ such that for every index i, $x_i \in A \iff i \in I$. Using the Recursion theorem, there is a computable function $e(a)$ such that $x_{e(a)} = x_a$ if $e(a) \notin I$, and $x_{e(a)}$ is some point from the dense sequence if $e(a) \in I$. Indeed, the program computing $x_{e(a)}$ starts enumerating the neighborhoods of x_a, testing $e(a) \in I$ in parallel. If that test eventually halts then the program looks for some point from the dense sequence in the intersection of all the neighborhoods of x_a enumerated so far, and then enumerates all the neighborhoods of that point.

Now A is non-empty iff there is a such that $e(a) \in I$. When A is non-empty, one can compute an element in A: look for a such that $e(a) \in I$, $x_{e(a)}$ is such a point. To get a computable dense sequence, apply this argument to the intersection of A with each basic open set $\mathcal{B}_i$.

The following result provides an upper bound on the effective Borel complexity of Markov-semidecidable sets.

Proposition 4 *Let $A \subseteq X_c$ be Markov-semidecidable. There exist uniformly effective open sets $\mathcal{U}_k \subseteq \mathcal{X}$ such that $A = \bigcap_k \mathcal{U}_k \cap X_c$.*

Proof Let $\mathcal{U}_k$ be the effective open sets from the proof of Theorem 3. We already know that $A \subseteq \mathcal{U}_k$ for all k. If $x \in \bigcap_k \mathcal{U}_k \cap X_c$ then let $k \geq K(x)$: since $x \in \mathcal{U}_k \cap X_c(k) = A \cap X_c(k)$, we conclude that $x \in A$.

The result above is actually tight, as we will now show. For the sake of completeness, let us recall original Friedberg's example. We present it in a way that is more convenient for our purposes. For a natural number n, $K(n)$ is its Kolmogorov complexity (in any of its versions).

Theorem 6 (Friedberg) *On the Cantor space, the set*

$$\mathcal{A} = \{0^\omega\} \cup \bigcup_{n:K(n)<\log(n)-1} [0^n 1].$$

is Markov-semidecidable but not open. Hence the Ershov topology is strictly stronger than the Cantor topology.

Proof We show that $\mathcal{A}$ is K-semidecidable. There is a constant c such that for every $n \in \mathbb{N}$ and every $x \in [0^n 1]$, $K(n) \leq K(x) + c$ (n can be easily computed from a program computing x). Given an infinite binary sequence x (a Type-2-description) and an upper bound k on $K(x)$, we only need to read the first $e = 2^{k+c+2}$ bits of x. If we see only zeros, we accept (this is safe as if $x \in [0^n 1]$ for some n, $K(n) \leq K(x) + c \leq k + c$ and $n \geq 2^{k+c+2}$ so $K(n) < \log(n) - 1$ hence $x \in A$). Otherwise one gets $0^n 1 \dots$ for some $n < e$, then test whether $K(n) < \log(n) - 1$.

Remark 4 Friedberg's example is not Σ_1^0 as it is not open but it is still low in the effective Borel hierarchy. It happens to be Δ_2^0 and even in co$\mathcal{D}_2$, i.e. its complement is the difference of two effective open sets. It is an effective open set appended with a limit point. We strengthen Friedberg's example by constructing a Markov-semidecidable set which is far from being open and higher in the effective Borel hierarchy.

For a finite binary string u, let us define the monotone complexity $Km(u)$ of u as the length of a shortest program computing a (finite or infinite) binary sequence extending u. The program writes its output on a one-way output tape and may never halt. Again the precise definition of $Km(u)$ (Levin or Schnorr monotone or process complexity) does not make any difference for our purposes. The only important property is that for a computable sequence x, $Km(x{\restriction_n}) \leq Km(x)$ for all n.

Theorem 7 *There is a Markov-semidecidable subset of $\{0,1\}_c^{\mathbb{N}}$ that is not Σ_2^0. It is a non-empty closed subset of $\{0,1\}_c^{\mathbb{N}}$ with empty interior, defined by*

$$A = \{x \in \{0,1\}_c^{\mathbb{N}} : \forall n, Km(x{\restriction_n}) < n/2 + a\}$$

for some sufficiently large $a \in \mathbb{N}$.

Proof We choose a such that for some computable sequence x, $K(x) \leq a$, hence A is non-empty as it contains x. We first show that A is K-semidecidable. First, the function $u \mapsto Km(u)$ is right-c.e. Now, given x and some $k \geq K(x)$, $x \in A$ iff for all $n \leq 2(k-a)$, $Km(x\restriction_n) < n/2 + a$, as for all $n > 2(k-a)$, $Km(x\restriction_n) \leq K(x) \leq k < n/2 + a$.

Here we denote $\{0,1\}^{\mathbb{N}}$ by $\mathcal{X}$ and the set of computable sequences by X_c. A is a subset of X_c. We show that there is no Σ^0_2-subset of $\mathcal{X}$ whose intersection with X_c is A. Let $\overline{A}$ be the closure of A in $\mathcal{X}$ (it might not be $\{x \in \mathcal{X} : \forall n, Km(x\restriction_n) < n/2 + a\}$). Here is the argument:

1. A has empty interior in X_c, i.e. there is no cylinder $[u]$ such that $[u] \cap X_c \subseteq A$. Indeed, given a finite string u and a sufficiently large k, for most words v of length k, $Km(uv) \geq |uv|/2 + a$ so $[uv]$ is disjoint from A.
2. If $\mathcal{P} \subseteq \mathcal{X}$ is a Π^0_1-set and $\mathcal{P} \cap X_c \subseteq A$ then $\mathcal{P}$ is nowhere dense in A. Indeed, if there exists a cylinder $[u]$ such that $\emptyset \neq A \cap [u] \subseteq \mathcal{P}$ then $A \cap [u] = \mathcal{P} \cap [u] \cap X_c$ is both Markov-semidecidable and Markov-co-semidecidable hence by Kreisel-Lacombe-Shœnfield/Čeitin theorem it is clopen on X_c, so A has non-empty interior in X_c, contradicting the first point.
3. By Proposition 3, $\overline{A}$ is a c.e. closed subset of $\mathcal{X}$ (it contains a dense computable sequence) hence a Π^0_2-set. Let $\mathcal{S} \subseteq \mathcal{X}$ be the Π^0_2-set given by Proposition 4, satisfying $A = \mathcal{S} \cap X_c$. Let $\mathcal{S}' = \overline{A} \cap \mathcal{S}$. $\mathcal{S}'$ is a Π^0_2-set which contains a dense computable sequence, and A is exactly the set of computable elements of $\mathcal{S}'$. From this it follows that the computable Baire theorem holds on $\mathcal{S}'$: if the sets $\mathcal{P}_i$ are uniformly Π^0_1-sets that have empty interior in $\mathcal{S}'$ then one can compute some x in $\mathcal{S}' \setminus \bigcup_i \mathcal{P}_i$.
4. Now, if $\mathcal{P}_i$ are uniformly Π^0_1-sets such that each $\mathcal{P}_i \cap X_c$ is contained in A then by the second point $\mathcal{P}_i$ has empty interior in A, and also in $\mathcal{S}' \subseteq A$, so by the third point one can compute some x in $\mathcal{S}' \setminus \bigcup_i \mathcal{P}_i$. As x is computable and belongs to $\mathcal{S}'$, x belongs to A so $\bigcup_i \mathcal{P}_i$ does not cover A.

4.1 The extended natural numbers

For the following results, we restrict our attention to the space $\mathcal{X} = \overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$ whose topology is generated by the singletons $\{n\}$ and the semi-lines $[n, \infty]$, for $n \in \mathbb{N}$. Note that every point in this space is computable, so that $\mathcal{X} = X_c$.

Observe that for a finite $x \in \overline{\mathbb{N}}$, $K(x)$ here coincides with the usual notion of Kolmogorov complexity of natural numbers, and $K(\infty)$ is some finite number.

Friedberg's example translated to this space reads $\{x \in \overline{\mathbb{N}} : K(x) < \log(x) - 1\}$, which inspires the following definition.

Definition 3 We define the ***Friedberg sets*** of $\overline{\mathbb{N}}$ to be the ones of the form $\{x \in \overline{\mathbb{N}} : K(x) < h(x)\}$, where $h : \mathbb{N} \to \mathbb{N}$ is any computable order, namely, any non decreasing unbounded computable function.

Note that a computable order can always be extended to a computable function $h : \overline{\mathbb{N}} \to \overline{\mathbb{N}}$, with $h(\infty) = \infty$.

Friedberg sets are Markov-semidecidable just like the set from Theorem 6. The next two results show that the only Markov-semidecidable sets over $\overline{\mathbb{N}}$ which are not Type-2-semidecidable are essentially the Friedberg sets.

Proposition 5 *If $A \subseteq \overline{\mathbb{N}}$ is Markov-semidecidable and contains ∞ then there is a computable order h such that A contains a Friedberg set.*

Proof Since A is K-semidecidable, for each k one can compute $p(k)$ such that $[p(k), \infty] \cap \{x : K(x) \leq k\} \subseteq A$. One can assume that $p(k)$ is increasing. Let $h(n) = \min\{i : p(i) > n\}$. If $n \notin A$ then $p(K(n)) > n$ (just take $k = K(n)$), so one has $h(n) \leq K(n)$.

Proposition 5 provides a nice characterization of the Ershov's open sets.

Corollary 2 *The Ershov topology is generated by the singletons $\{n\}$ and the Friedberg sets.*

Whether or not one can find such a characterization on other spaces such as the Cantor space is an interesting question.

We end this section by observing that, unlike Type-2-semidecidable sets, Markov-semidecidable sets cannot be effectively enumerated.

Proposition 6 *There is no effective enumeration of the Markov-semidecidable subsets of $\overline{\mathbb{N}}$.*

Proof Let ν be some admissible numbering of $\overline{\mathbb{N}}$. Let A_i be a sequence of uniformly Markov-semidecidable sets, coming with uniformly c.e. sets $E_i \subseteq \mathbb{N}$ such that $E_i \cap \mathrm{dom}(\nu) = \nu^{-1}(A_i)$. One can extract the sets that contain ∞ (let e_0 be some index of ∞, one can enumerate the numbers i such that E_i contains e_0) and rename the subsequence, so we assume that each A_i contains ∞. For each i one can compute an increasing computable function $f_i : \mathbb{N} \to \mathbb{N}$ whose range is contained in A_i. Now we build a Markov-semidecidable set A that contains ∞ and differs from each A_i. Let f be a computable order such that for each i and all sufficiently large k, $f_i(k) < f(k)$ (for instance, $f(k) = \max(f_0(k), \dots, f_k(k)) + 1$). Here we use another version of Kolmogorov complexity: $C(x)$ is the minimal index of x. We now define $A = \{x \in \overline{\mathbb{N}} : f(C(x)) \leq x\}$. A is Markov-semidecidable by the usual argument.

We show that A differs from each A_i. Let $i \in \mathbb{N}$. As $C \circ f_i$ is one-to-one, there exist infinitely many $k \in \mathbb{N}$ such that $C(f_i(k)) \geq k$. Moreover if k is sufficiently large then $f_i(C(f_i(k))) < f(C(f_i(k)))$. Hence there exists k such that $f_i(k) \leq f_i(C(f_i(k))) < f(C(f_i(k))$. Let $x = f_i(k)$: $x \in A_i$ by construction of f_i and $x < f(C(x))$ so $x \notin A$.

5 When Markov beats Kolmogorov

In this section we explore the limits of our results. We first look at the relativized case, and show that there are simple cases that separate Markov-computability from K-computability. However, we also show that, interestingly, the equivalence persists if the space has a Polish structure.

5.1 Relativization

Let $\mathbb{S} = \{\perp, \top\}$ be the Sierpiński space with topology given by $\{\emptyset, \{\top\}, \{\perp, \top\}\}$. Note that as $\mathbb{S}$ is finite, K-computability is trivially equivalent to Type-2-computability simply because all the elements share a common upper bound on their Kolmogorov complexities, which therefore provides no interesting information. Relativizing w.r.t. the Halting set, we can then separate Markov-decidability from Type-2-decidability, and therefore from K-decidability.

Remark 8 *The set $\{\perp\} \subseteq \mathbb{S}$ is Markov-decidable relative to the Halting set but is not Type-2-decidable relative to any oracle.*

Proof It is not decidable relative to any oracle simply because it is not clopen.

Note that this is only a partial relativization as the Markov-names are not relativized. For this reason, Remark 8 is not very interesting, but will be used to prove a deeper result (Theorem 10).

Similarly, over $\mathcal{P}(\mathbb{N})$, $\emptyset''$ separates K-semidecidability from Type-2-semidecidability (without oracle, Rice-Shapiro theorem shows that the two notions coincide with Markov-semidecidability).

Proposition 7 *The set $\{\mathbb{N}\} \subseteq \mathcal{P}(\mathbb{N})$ is K-semidecidable relative to $\emptyset''$ but is not Type-2-semidecidable relative to any oracle.*

Proof Let $E \subseteq \mathbb{N}$ be a c.e. set and k an upper bound on its Kolmogorov complexity. From k we know that A has an index in some finite set F. Using $\emptyset''$ we, for each $e \in F$, decide whether $W_e = \mathbb{N}$ and compute, when $W_e \neq \mathbb{N}$, an element outside W_e. We then wait that each one of this finite set of elements appears in A, enumerated on the input tape, and accept A if it is the case.

The set $\{\mathbb{N}\}$ is not Type-2-semidecidable relative to any oracle because it contains the set $\mathbb{N}$ but no finite subset.

However, metric spaces behave differently. Although stated on Cantor space, the next result extends to any computable metric space [23].

Proposition 8 *Let $O \subseteq \mathbb{N}$. A subset of $\{0,1\}^{\mathbb{N}}_c$ is Markov-semidecidable relative to O if and only if it is K-semidecidable relative to O.*

Proof There are two cases, depending on whether O computes the halting set or not.

If O computes the halting set then by the remark following Proposition 1, Markov-names can be uniformly computed from K-names relative to O, which gives the result. This part only works in the case of the Cantor space.

If O does not compute the halting set then Lemma 1 and Theorem 3 still hold relative to O on any effective topological space. Indeed, in the proof of Lemma 1 we used the non-computability of the halting set. As long as O does not compute the halting set, the argument remains valid. This part holds on every effective topological space.

Let us state this result differently. Every subset A of $\{0,1\}^{\mathbb{N}}_c$ is Markov-semidecidable relative to some oracle, in an obvious way: let O encode an enumeration of an index set of A, i.e. a set containing all the indices of the elements of A but no index of elements of $\{0,1\}^{\mathbb{N}}_c \setminus A$, and let M be the machine with oracle O accepting its input i iff i belongs to the set encoded by O. M Markov-semidecides A with oracle O.

Proposition 8 can be reformulated this way: every subset A of $\{0,1\}^{\mathbb{N}}_c$ is K-semidecidable relative to any enumeration of an index set of A. Is it uniform? Does the machine K-semideciding A relative to an index set of A depend on the particular index set?

Observe that the proof of Proposition 8 is not uniform as two cases are treated separately. We now show that we cannot get rid of this distinction, i.e. that the result is not uniform.

Theorem 9 *While every subset A of $\{0,1\}^{\mathbb{N}}_c$ is K-semidecidable relative to any enumeration of any index set of A, it is not so uniformly.*

Proof Assume that there is a single machine M that for each set $A \subseteq \{0,1\}^{\mathbb{N}}_c$ and each enumeration of an index set of A, K-semidecides A using the enumeration as oracle.

Let us consider the element $x_0 \in \{0,1\}^{\mathbb{N}}_c$ which is the sequence with only 0's, and the set $A_0 = \{x_0\} \subseteq \{0,1\}^{\mathbb{N}}_c$. For each $k \geq K(x_0)$, M accepts the K-name (x_0, k) of x_0 provided any enumeration of the index set of A_0.

Hence for each $k \geq K(x_0)$ there exists a finite string u and a number n such that M with oracle u accepts $(0^n, k)$, and such that u encodes the enumeration of a finite set $F \subseteq \mathbb{N}$, such that each $i \in F$ is an index of a (finite or infinite) sequence extending 0^{n+1}. Such a pair (u, n) can be effectively found, so to each $k \geq K(x_0)$ one can associate some $n = n(k)$, in a computable way.

The Kolmogorov complexity of a computable sequence x_i can be bounded computably in i: let $k(i)$ be a total computable function such that $K(x_i) \leq k(i)$ for every index i. Using the Recursion Theorem, we now define some i such that $x_i = 0^{n(k(i))}1^{\omega}$. By definition of the function n, there is some finite string u such that on oracle u, M accepts $(0^{n(k(i))}, k(i))$, and such that u enumerates a finite set F such that every $i \in F$ is an index of a sequence extending $0^{n(k(i))+1}$. As a result, F contains no index of x_i but M accepts the K-name $(x_i, k(i))$ of x_i.

Let $A = \{0,1\}^{\mathbb{N}}_c \setminus x_i$. As u encodes no index of x_i, u can be extended so that it encodes an enumeration of the index set of A. On that oracle, M accepts a

K-name of x_i. However x_i does not belong to A, so M does not K-semidecide A on that oracle.

5.2 Functions to non-effective topological spaces

Corollary 1 states that Markov-computable functions are the same as K-computable ones. This result assumes that the underlying spaces are effective topological spaces, which are essentially countably-based spaces. Does the result still hold when the spaces are not countably-based? We investigate what happens when the target space is not countably-based, but still has an admissible representation, as defined in [18]. We show that Corollary 1 breaks in that case.

One of the simplest examples of a non-countably-based space is the space $\mathcal{O}(\mathbb{B})$ of open subsets of the Baire space. The topology is generated by the following sets: given a compact set $K \subseteq \mathbb{B}$, the class of open subsets of $\mathbb{B}$ containing K is open. This topology is not countably-based and hence is not an effective topology. However it does have an admissible representation: an open set is represented by an enumeration of cylinders whose union is the open set. Computing an element U of $\mathcal{O}(\mathbb{B})$ means enumerating the open set and is equivalent to semideciding, given $f \in \mathbb{B}$ as oracle, whether f belongs to U. Hence the computation of an element of $\mathcal{O}(\mathbb{B})$ amounts to a semidecision procedure, relative to an oracle f. This observation enables us to use the results from the previous section.

We now present the details of the simplest case, a uniform version of Remark 8. This result contrasts with Corollary 1.

Theorem 10 *There exists a Markov-computable function $F : \mathbb{S} \to \mathcal{O}(\mathbb{B})$ that is not K-computable.*

Proof We use the admissible numbering $\nu_{\mathbb{S}}$ of $\mathbb{S}$ defined by $\nu_{\mathbb{S}}(e) = \top$ if $\varphi_e(e)\downarrow$, $\nu_{\mathbb{S}}(e) = \bot$ otherwise. We define two effective open sets $U_\bot, U_\top$ and define $F(\bot) = U_\bot$ and $F(\top) = U_\top$. First, let $U_\bot = \mathbb{B}$. Let $T : \mathbb{N} \to \mathbb{N}$ be defined as follows: $T(n)$ is the halting time of $\varphi_n(n)$ if it halts, $T(n) = 0$ otherwise. The open set $U_\top := \mathbb{B} \setminus \{T\}$ happens to be effective. First the function F is not Type-2-computable because it is not continuous: indeed, F is not monotonic as $\bot \leq \top$ but $U_\bot = \mathbb{B}$ is not contained in $U_\top \subsetneq \mathbb{B}$. As $\mathbb{S}$ is finite, F is not K-computable neither. However F is Markov-computable. Given an index e of $s \in \mathbb{S}$, enumerate $U_\top$ and enumerate the set of functions f such that $\varphi_e(e)$ does not halt in exactly $f(e)$ steps. The latter set of functions is effectively open, uniformly in e. If $\varphi_e(e)\uparrow$ then the whole space $\mathbb{B}$ is enumerated. If $\varphi_e(e)\downarrow$ then nothing more than $U_\top$ is enumerated. Intuitively, given e and f, from T one can decide whether $\varphi_e(e)$ halts, i.e. whether $\nu_{\mathbb{S}}(e) = \bot$.

A similar construction, based on Proposition 7, yields a function $F : \mathcal{P}(\mathbb{N}) \to \mathcal{O}(\mathbb{B})$ which is K-computable but not Type-2-computable by replacing the function T from Theorem 10 by a function T' computing $\emptyset''$ and such that $\mathbb{B} \setminus \{T'\}$ is effectively open.

Combining all these results, and using that fact that Theorem 10 can clearly also be realized using $\mathcal{P}(\mathbb{N})$ in place of $\mathbb{S}$, we obtain our announced Theorem C.

Theorem 11 *For functions from $\mathcal{P}(\mathbb{N})$ with values on $\mathcal{O}(\mathbb{B})$ one has that:*

$$\text{Markov-computability} > \text{K-computability} > \text{Type-2-computability}.$$

While Type-2-computable functions are always Scott continuous (i.e. monotone and compact), one can show that K-computable functions are always monotone but not necessarily compact. Markov-computable functions may even not be monotone.

Let us now briefly discuss whether Theorem 11 holds for functions from the Cantor space to $\mathcal{O}(\mathbb{B})$. Friedberg's example of a Markov (hence K)-semidecidable set that is not Type-2-semidecidable directly implies the second inequality. However the idea behind the proof of the first inequality cannot be applied on Cantor space. Indeed, using Proposition 8 one can show that the analog of the function of Theorem 10 is actually K-computable.

Proposition 9 *The function $G : \{0,1\}^{\mathbb{N}} \to \mathcal{O}(\mathbb{B})$ mapping 0^ω to $\mathbb{B}$ and any other sequence to $\mathbb{B} \setminus \{T\}$ is K-computable.*

Proof Given x, k and f, apply the algorithm given by Proposition 8 to semidecide, if $f = T$, whether $x = 0^\omega$. In parallel, semidecide whether $f \neq T$.

We leave the following question open: is there a Markov-computable function from the Cantor space to $\mathcal{O}(\mathbb{B})$ that is not K-computable?

6 Future work

We list a few problems for future work.

- Find a characterization of the Ershov topology on other spaces than $\overline{\mathbb{N}}$, like the Cantor space.
- Determine for which levels of the effective difference hierarchy the Markov-model and the K-model are equivalent. We know from Theorem 4 that the equivalence holds for the *finite* levels. What about the level ω?
- All our results hold when the space $\mathcal{X}$ is an effective topological space. However the three models also make sense on any represented space. It seems like an interesting research program to study the extent to which our results are valid in this case.
- Compare the effective Borel hierarchy induced by the Markov-semidecidable sets, the hierarchy induced by the arithmetical hierarchy on the indices and the effective Borel hierarchy induced by the standard topology.

References

1. G. S. Čeitin. Algorithmic operators in constructive metric spaces. *Trudy Matematiki Instituta Steklov*, 67:295–361, 1962. English translation: *American Mathematical Society Translations*, series 2, 64:1-80, 1967.
2. Matthew de Brecht. Quasi-polish spaces. *Ann. Pure Appl. Logic*, 164(3):356–381, 2013.
3. Rusins Freivalds and Rolf Wiehagen. Inductive inference with additional information. *Journal of Information Processing and Cybernetics*, 15:179–185, 1979.
4. Richard M. Friedberg. Un contre-exemple relatif aux fonctionnelles récursives. *Comptes Rendus de l'Académie des Sciences*, 247:852–854, 1958.
5. Jacques Grassin. Index sets in Ershov's hierarchy. *The Journal of Symbolic Logic*, 39:97–104, 3 1974.
6. Andrzej Grzegorczyk. On the definitions of computable real continuous functions. *Fundamenta Mathematicae*, 44:61–71, 1957.
7. Peter Hertling. Computable real functions: Type 1 computability versus Type 2 computability. In *CCA*, 1996.
8. Mathieu Hoyrup and Cristobal Rojas. On the information carried by programs about the objects they compute. In Ernst W. Mayr and Nicolas Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPIcs*, pages 447–459. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.
9. G. Kreisel, D. Lacombe, and J.R. Shœnfield. Fonctionnelles récursivement définissables et fonctionnelles récursives. *Comptes Rendus de l'Académie des Sciences*, 245:399–402, 1957.
10. Boris A. Kushner. The constructive mathematics of A. A. Markov. *Amer. Math. Monthly*, 113(6):559–566, 2006.
11. Daniel Lacombe. Extension de la notion de fonction récursive aux fonctions d?une ou plusieurs variables réelles I-III. *Comptes Rendus Académie des Sciences Paris*, 240,241:2478–2480,13–14,151–153, 1955.
12. A. A. Markov. On the continuity of constructive functions (russian). *Uspekhi Mat. Nauk*, 9:226–230, 1954.
13. J. Myhill and J. C. Shepherdson. Effective operations on partial recursive functions. *Mathematical Logic Quarterly*, 1(4):310–317, 1955.
14. Marian B. Pour-El. A comparison of five "computable" operators. *Mathematical Logic Quarterly*, 6(15-22):325–340, 1960.
15. H. G. Rice. Classes of recursively enumerable sets and their decision problems. *Transactions of the American Mathematical Society*, 74(2):pp. 358–366, 1953.
16. Hartley Jr. Rogers. *Theory of Recursive Functions and Effective Computability*. MIT Press, Cambridge, MA, USA, 1987.
17. C.P. Schnorr. Optimal enumerations and optimal gÃűdel numberings. *Mathematical systems theory*, 8(2):182–191, 1974.
18. Matthias Schröder. Extended admissibility. *Theoretical Computer Science*, 284(2):519–538, 2002.
19. Victor L. Selivanov. Index sets in the hyperarithmetical hierarchy. *Siberian Mathematical Journal*, 25:474–488, 1984.
20. N. Shapiro. Degrees of computability. *Transactions of the American Mathematical Society*, 82:281–299, 1956.
21. Dieter Spreen. Representations versus numberings: on the relationship of two computability notions. *Theoretical Computer Science*, 262(1):473–499, 2001.
22. Alan Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2, 42:230–265, 1936.
23. Klaus Weihrauch. *Computable Analysis*. Springer, Berlin, 2000.