Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 14 (2011)

# Towards an axiomatic system for Kolmogorov complexity

Antoine Taveneaux

LIAFA, CNRS & Université de Paris 7

**Abstract.** In [She82], it is shown that four of its basic functional properties are enough to characterize plain Kolmogorov complexity, hence obtaining an axiomatic characterization of this notion. In this paper, we try to extend this work, both by looking at alternative axiomatic systems for plain complexity and by considering potential axiomatic systems for other types of complexity. First we show that the axiomatic system given by Shen cannot be weakened (at least in any natural way). We then give an analogue of Shen's axiomatic system for conditional complexity. In the second part of the paper, we look at prefix-free complexity and try to construct an axiomatic system for it. We show however that the natural analogues of Shen's axiomatic systems fail to characterize prefix-free complexity.

## 1 Introduction

The concept of Kolmogorov complexity was introduced independently by Kolmogorov (in [Kol65]) and Chaitin (in [Cha66]). The aim of Kolmogorov complexity is to quantify the amount of "information" contained in finite objects, such as binary strings. This idea can be used to give an answer to the philosophical question, "What does it mean for a single object to be random?"

The usual definition of Kolmogorov complexity uses the existence of an optimal Turing machine. However, it is not immediate from that definition that Kolmogorov complexity is satisfactory as a measure of information. One is only convinced after deriving certain fundamental facts about it, such as: most strings have maximal complexity, the complexity of a pair $(x, y)$ is not (much) greater than the sum of the complexities of $x$ and $y$, etc. Therefore, a natural question is to ask whether there exists an axiomatic system characterizing Kolmogorov complexity uniquely via some of its functional properties. And of course, as with any axiomatic system, we want the axiomatic system to be minimal, i.e. to contain no superfluous axiom.

Such a characterization was given by Shen in [She82] for plain complexity. In Section 2 we recall this characterization and adapt it to provide an axiomatic system for conditional complexity. We then study whether we can weaken the hypotheses of this characterization of plain complexity in a natural way and show that it is indeed not possible. In particular, one of the hypotheses in the theorem states that applying a partial computable function to a string does not increase

ISSN 1433-8092

its Kolmogorov complexity (up to an additive constant), and we show that this hypothesis cannot be restricted to total computable functions. To show that we need the power of partial computable functions to characterize plain complexity, we introduce a notion of complexity for functions that are total on a initial segment of the integers; this notion of complexity is robust under the application of total computable functions, but differs from Kolmogorov complexity.

A second natural question would be, "Is there a similar axiomatic system for prefix-free Kolmogorov complexity?" Unlike the plain complexity case, we show that the classical properties of prefix-free complexity are not sufficient to characterize it. Since prefix-free complexity is greater than plain complexity, we have to choose a larger upper bound and a tighter lower bound to characterize $\mathrm{K}$ (where $\mathrm{K}(x)$ is prefix-free complexity of the string $x$, see below). Actually, all basic upper bounds on prefix-free complexity fail to characterize it. To show that our the classical properties of prefix-free complexity do not characterize it, we construct a counter-example defined by $\mathrm{A} = \mathrm{K} + f$ with $f$ a very slow growing function. To build such a slow function, we define an operator that slows down sub-linear non-decreasing functions while preserving their computational properties (computability or semicomputability).

Throughout the paper we will identify natural numbers and finite strings in a natural way (the set of finite strings is denoted by $2^{<\omega}$). We denote by $\log(x)$ the discrete binary logarithm of $x$. We fix an effective enumeration of the Turing machines and we denote by $\psi_e$ the function computed by the $e^{\mathrm{th}}$ machine. For each machine $T$ and string $x$, the complexity of $x$ relatively to $T$ is:

$$\mathrm{C}_T(x) = \min\{n \mid \exists y \in 2^{<\omega} \text{ such that } |y| = n \text{ and } T(y) = x\}$$

and throughout the paper we fix an optimal machine $\mathbb{U}$ (i.e. a machine such that for all machines $T$ we have $\mathrm{C}_{\mathbb{U}} \leq \mathrm{C}_T + O(1)$, see [Nie09] for a existence proof of a such a machine) and set $\mathrm{C} = \mathrm{C}_{\mathbb{U}}$. In the same way, we fix an optimal prefix-free machine $\mathbb{U}'$ (i.e. a machine with prefix-free domain) and set $\mathrm{K} = \mathrm{C}_{\mathbb{U}'}$. $\mathrm{C}(x)$ and $\mathrm{K}(x)$ denote the plain complexity and prefix-free complexity of $x$, respectively.

Conditional Kolmogorov complexity is an extension of the above notions which quantifies the information of a string $x$ *relative* to another string $y$. More precisely, the complexity of $x$ given $y$, relative to the machine $T$, is:

$$\mathrm{C}_T(x|y) = \min\{n \mid \exists z \in 2^{<\omega} \text{ such that } |z| = n \text{ and } T(\langle z, y\rangle) = x\}$$

As above we can define $\mathrm{C}(.|.) = \mathrm{C}_{\mathbb{U}}(.|.)$ and $\mathrm{K}(.|.) = \mathrm{C}_{\mathbb{U}'}(.|.)$.

## 2 Plain complexity

As mentioned above, Shen showed in [She82] that four basic properties are sufficient to fully characterize plain Kolmogorov complexity:

1. Upper-semicomputability: $\mathrm{C}$ is not computable but it is upper semicomputable (i.e. the predicate $\mathrm{C}(x) \leq k$ is uniformly computably enumerable in $x$ and $k$).

2. Stability: a recursive function cannot increase the complexity of a string by more than an additive constant.
3. Explicit description: the length of the smallest description of a string (i.e. its plain complexity) is not much bigger than the string itself.
4. Counting: no more than $2^n$ of the strings have a complexity less than $n$.

Formally, Shen's theorem states the following.

**Theorem 1.** *[She82] Let* $\mathrm{A} : 2^{<\omega} \to \mathbb{N}$ *be some function. Suppose* $\mathrm{A}$ *satisfies the following four properties:*

1. $\mathrm{A}$ *is upper semi-computable.*
2. *For every partial computable function* $f : 2^{<\omega} \to 2^{<\omega}$ *there exists a constant* $c_f$ *such that for each* $\mathrm{A}(f(x)) \leq \mathrm{A}(x) + c_f$ *for each* $x \in 2^{<\omega}$*.*
3. $\mathrm{A}(x) \leq |x| + O(1)$ *for all* $x \in 2^{<\omega}$*.*
4. $|\{x|\mathrm{A}(x) \leq n\}| = O(2^n)$*.*

*Then* $\mathrm{A}(x) = \mathrm{C}(x) + O(1)$*.*

*Proof.* We give a quick sketch of the proof.

To show $\mathrm{A} \leq \mathrm{C} + O(1)$, let $x^*$ denote the shortest description of $x$ (for the complexity C ). By hypotheses 2 and 3 we have:

$$\mathrm{A}(x) = \mathrm{A}(\mathbb{U}(x^*)) \leq \mathrm{A}(x^*) + O(1) \leq |x^*| + O(1) = \mathrm{C}(x) + O(1).$$

To show $\mathrm{C} \leq \mathrm{A} + O(1)$, we consider $y$ and $n$ such that $\mathrm{A}(y) = n$. Since A is upper semi-computable, the set $\{x|\mathrm{A}(x) \leq n\}$ is uniformly computably enumerable. Since there exists a uniform $d$ such that $|\{x|\mathrm{A}(x) \leq n\}| = 2^{n+d}$, we can describe $y$ with only $n + d$ bits (this description $\overline{y}$ such that $|\overline{y}| = n + d$ represents the rank of $y$ in an enumeration of $\{x|\mathrm{A}(x) \leq |\overline{y}| - d\}$). So, for all $y$ we have $\mathrm{C}(y) \leq n + d + O(1) = \mathrm{A}(y) + O(1)$. □

**Remark 1:** The authors of [USV10] show that conditions 3 and 4 can be replaced by "There exists a constant $c$ such that $|\{x|\mathrm{A}(x) \leq n\}| \in [2^{n-c}, 2^{n+c}]$." We can also replace conditions 2 and 3 by "For every partial computable function $f$ there exists a constant $c_f$ such that $\mathrm{A}(f(x)) \leq |x| + c_f$ for each $x \in 2^{<\omega}$." Finally condition 4 can be replaced by the stronger version "$|\{x|\mathrm{A}(x) \leq n-k\}| = O(2^{n-k})$."
**Remark 2:** With essentially the same proof, one can show a similar result for conditional plain complexity. The following system characterizes of conditional plain complexity:

- Uniformly in $x, y \in 2^{<\omega}$, $\mathrm{B}(x|y)$ is computable from above.
- For all $x, y \in 2^{<\omega}$, $\mathrm{B}(x|y) \leq |x| + O(1)$.
- For each $y \in 2^{<\omega}$ we have $|\{x|\mathrm{B}(x|y) \leq n\}| = O(2^n)$ (such that $O(2^n)$ do not depend of $y$).
- For all $y$ and for every partial computable function $f$ from $2^{<\omega}$ to $2^{<\omega}$ there exists a constant $c_f$ such that for each $x \in 2^{<\omega}$:

$$\mathrm{B}(f(x)|y) \leq \mathrm{B}(x|y) + c_f.$$

To characterize the conditional aspect, we add to the four previous items the hypothesis "$\mathrm{B}(\langle x,y\rangle|y) \leq \mathrm{B}(x|y)+O(1)$". Note however that replacing this last condition by $\mathrm{B}(x|x) = O(1)$ would not be sufficient.

## 2.1 Weakening the hypotheses

Shen's theorem raises a natural question: Are all 4 conditions actually needed? In this subsection we discuss this question. First, it is not hard to see that none of the hypotheses can be removed.

- We need the hypothesis 3 because the function 2C satisfies the three others hypotheses.
- The hypothesis 4 is necessary because the function 0 satisfies the three others hypotheses.
- The hypothesis 1 cannot be removed since $\mathrm{C}^{\emptyset'}$ (plain Kolmogorov complexity relativised to the halting problem oracle) satisfies each of three others hypotheses (and clearly differs from the unrelativized version C).
- The hypothesis 2 cannot be removed because the length function satisfies the three others hypotheses.

It could however be the case that hypothesis 2 be replaced by the weaker "for all total computable functions $f$ there exists $c_f$ such that $\mathrm{A}(f(x)) \leq \mathrm{A}(x)+c_f$". Our first main result is that this is not case.

**Theorem 2.** *There exists a function* $\mathrm{A} : 2^{<\omega} \to \mathbb{N}$ *satisfying hypotheses 1, 3, 4 (of Theorem 4) and:*

- *For every total computable function $f$ from $2^{<\omega}$ to $2^{<\omega}$ there exists a constant $c_f$ such that $\mathrm{A}(f(x)) \leq \mathrm{A}(x)+c_f$ for each $x \in 2^{<\omega}$.*
- $|\mathrm{A}(x)-\mathrm{C}(x)|$ *is not bounded.*

*Proof.* In [MMSV10], the authors define a notion of total conditional complexity $\overline{\mathrm{C}}(x|y)$ as the smallest length of a program for a total function $f$ code such that $f(y)=x$. Of course, $\overline{\mathrm{C}}$ is stable over all total computable functions (i.e. $\overline{\mathrm{C}}(f(x)|y) \leq \overline{\mathrm{C}}(x|y)+c_f$ for all total computable functions $f$) and the authors show that $\overline{\mathrm{C}}$ significantly differs from the plain conditional complexity. However, the function $\overline{\mathrm{C}}$ is not quite suitable for our purposes, for two reasons. First, it is not upper semi-computable and second, its non-conditional version $\overline{\mathrm{C}}(x|\lambda)$ is equal to C up to a constant.

In order to construct our counter-example, we first define a way to encode compositions of partial computable functions by a set of strings having the prefix-free property. This encoding is not at all optimal, which is precisely what will make our proof work. We define:

$$P = \{1^{p_1}0001^{p_2}000\dots1^{p_k}01| \ \forall k,\ p_k > 0\}.$$

Notice that $P$ is a prefix-free set. For $\tau = 1^{p_1}0001^{p_2}000\dots1^{p_k}01 \in P$ we now denote by $\varphi_\tau$ the function $\varphi_\tau = \psi_{p_1} \circ \psi_{p_2} \circ \dots \circ \psi_{p_i}$ (recall that $(\psi_i)$ is a standard enumeration of partial computable functions). We now define a function $V$, as follows. For all $x \in 2^{<\omega}$ and $\tau \in P$, set

$$V(\tau x) = \begin{cases} \varphi_\tau(x) & \text{if for all } y \text{ such that } |y| \leq |x| \text{ we have } \varphi_\tau(y)\downarrow \\ \uparrow & \text{otherwise} \end{cases}$$

$P$ is prefix-free, so $V$ is defined without ambiguity and clearly $V$ is a (partial) computable function. If $\varphi_\tau$ is not a total function, there are only a finite number of strings $x$ such that $V(\tau x)\downarrow$. We shall prove that $\mathrm{A} = \mathrm{C}_V$ satisfies the conditions of the theorem. First, $\mathrm{C}_V$ is upper semicomputable and satisfies the counting condition, as it is just the Kolmogorov complexity function associated to the machine $V$. Moreover, let $i$ be an index for the identity function (i.e. $\psi_i = id$). By definition of $V$, one has $V(1^i 01x) = x$, hence $\mathrm{A}(x) \leq |x| + (i+2)$. To see that A is stable over all total computable functions, let $f$ be a total computable function and let $e$ be an index for $f$. Now, for any string $x$, let $\tau y$ be such the shortest description of $x$ for $V$ with $\tau \in P$. By definition of $V$, this means that $\varphi_\tau(z)\downarrow$ for all $|z| \leq |y|$. And since $f = \psi_e$ is total, we also know that $\psi_e \circ \varphi_\tau(z)\downarrow$ for all $|z| \leq |y|$. Therefore $\sigma = 1^e 000\tau$ is a description of $x$ for $V$. We have proven that $\mathrm{C}_V(f(x)) \leq \mathrm{C}_V(x) + e + 3$ for all $x$.

It remains to prove that $\mathrm{C}_V$ differs from C, i.e. that $\mathrm{C}_V - \mathrm{C}$ takes arbitrarily large values. We prove this by contradiction: Suppose that $|\mathrm{A}(x) - \mathrm{C}(x)|$ is bounded by a constant. For $x \in 2^{<\omega}$, we denote by $\widehat{x}$ the smallest description of $x$ for $V$ (by definition this means that $\mathrm{C}_V(x) = |\widehat{x}|$).

Let $x$ be a string. Let us first write

$$\widehat{x} = 1^{p_1}0001^{p_2}0000 \ldots 1^{p_k}01y.$$

It is easy to see that

$$\mathrm{C}(\widehat{x}) \leq 2\log(p_1) + 2\log(p_2) + \cdots + 2\log(p_k) + 2k + |y| + O(1),$$

and since $x$ can be computed from $\widehat{x}$, this implies a fortiori:

$$\mathrm{C}(x) \leq 2\log(p_1) + 2\log(p_2) + \cdots + 2\log(p_k) + 2k + |y| + O(1).$$

Moreover, by definition of $V$,

$$\mathrm{C}_V(x) = |\widehat{x}| = p_1 + p_2 + \cdots + p_k + 3(k-1) + 2 + |y|.$$

Thus, since we have assumed that $\mathrm{C}_V(x) - \mathrm{C}(x)$ bounded, this shows two things:

- the $(p_i)$ appearing in the $\widehat{x}$'s are bounded, and
- the number of $p_i$'s used in each $\widehat{x}$ is bounded.

Formally, we have proven that $\{\tau \in P \mid \exists x \in 2^{<\omega} \text{ such that } \widehat{x} = \tau y\}$ is a finite set and that for each $\tau$ in this set, either $\varphi_\tau$ is a total function or for $y$ large enough, $\tau y$ is not in the domain of $V$. Thus, this $\tau$ appears only in a finite number of $\widehat{x}$.

Finally for $|\widehat{x}|$ large enough (and hence for $|x|$ large enough because $\{x|\mathrm{A}(x) \leq n\}$ is finite for all $n$), $\widehat{x} = \tau x$ with $\tau \in P$, and $\varphi_\tau$ is a total computable function. So

$$Q = \{\tau \in P | \exists^\infty x \in 2^{<\omega} \text{ such that } \widehat{x} = \tau y\}.$$

is a finite set of codes of total functions and thus there is only a finite number of $\tau \in P$ in the prefixes of $\widehat{x}$'s.

Therefore, for $x$ large enough, $\widehat{x}$ is of the form $\tau y$ with $\tau \in Q$ and hence:

$$\mathrm{A}(x) = \min\{|\tau y| \mid \tau \in Q \text{ and } \varphi_\tau(y) = x\}.$$

Since $Q$ is finite and all $(\varphi_\tau)_{\tau \in Q}$ are total, this makes A computable, contradicting $\mathrm{A} = \mathrm{C} + O(1)$ because no non-trivial lower-bound of C is computable.

□

## 3 An axiomatic system for prefix complexity

As we have seen in the last section, there exists a minimal set of simple properties that characterize plain complexity. One may ask whether it is possible to obtain a similar characterization of prefix-free complexity K .

It is natural to keep the hypotheses 1 and 2, but the other two hypotheses need to be adapted. Indeed, hypothesis 3 fails to hold for K (i.e. $\mathrm{K}(x) \nleq |x| + O(1)$), and the sharpest classical upper bound is $\mathrm{K}(x) \leq |x| + \mathrm{K}(|x|) + O(1)$ (see [DH10]).

Accordingly, the hypothesis 4 (i.e. $|\{x|\mathrm{K}(x) \leq n\}| = O(2^n)$) is too weak. The analogue of that counting argument for K is the classical

$$|\{x||x| = n \text{ and } \mathrm{K}(x) \leq n + \mathrm{K}(n) - k\}| = O(2^{n-k}).$$

Another property of K that is very often used is $\sum_{x \in 2^{<\omega}} 2^{-\mathrm{K}(x)} < \infty$ (in fact, any upper semi-computable function A satisfying $\sum_{x \in 2^{<\omega}} 2^{-\mathrm{A}(x)} < \infty$ is such that $\mathrm{K} \leq \mathrm{A} + O(1)$). Perhaps surprisingly, this set of properties alone is not enough to characterize K.

**Theorem 3.** *There exists a function* A *satisfying the following:*

1. A *is upper semi-computable.*
2. *For every partial computable function $f$ from $2^{<\omega}$ to $2^{<\omega}$ there exists a constant $c_f$ such that for each $\mathrm{A}(f(x)) \leq \mathrm{A}(x) + c_f$ for each $x \in 2^{<\omega}$.*
3. $\sum_{x \in 2^{<\omega}} 2^{-\mathrm{A}(x)} < \infty$.
4. $\mathrm{A}(x) \leq |x| + \mathrm{A}(|x|) + O(1)$ *for each* $x \in 2^{<\omega}$.
5. $|\{x \in 2^n \mid \mathrm{A}(x) \leq |n| + \mathrm{A}(n) - b\}| \leq O(2^{n-b})$.
6. $|\mathrm{A} - \mathrm{K}|$ *is not bounded.*

**Remark:** Since hypothesis 3 guarantees the inequality $\mathrm{K} \leq \mathrm{A} + O(1)$, it would be sufficient, in order to obtain a full characterization of K, to add the property: "For every $f$ partial computable prefix-free function there exists $c_f$ such that

$\mathrm{A}(f(x)) \leq |x| + c_f$". Indeed, for all $x$ if we denote by $x^*$ a shortest string such that $\mathbb{U}'(x^*) = x$ then $\mathrm{A}(x) = \mathrm{A}(\mathbb{U}'(x*)) \leq |x^*| + c_{\mathbb{U}'} = \mathrm{K}(x) + c_{\mathbb{U}'}$. However, such a system would not be very satisfactory because it uses the prefix-freeness of functions and thus is mostly a rewording of the definition of K.

*Proof.* We will construct A by taking $\mathrm{A} = \mathrm{K} + \beta$ with $\beta$ an unbounded function with certain nice properties. The function $\beta$ will be upper semicomputable, non-decreasing, unbounded, such that

$$\beta(x) = \beta(|x|) + O(1),$$

and such that for $f$ partial computable function, there is $c_f$ such that

$$\beta(f(n)) \leq \beta(n) + c_f. \tag{1}$$

Simple considerations show that $\beta$ has to have a very low growth speed. First let us define Solovay's $\alpha$-function:

**Definition 1.** *The Solovay's $\alpha$-function is defined by:*

$$\alpha(n) = \min\{\mathrm{K}(i) | i > n\}.$$

*We call order a total, non-decreasing and unbounded (not necessarily computable) function $f : \mathbb{N} \to \mathbb{N}$.*

Equivalently $\alpha(n)$ is the length of the shortest string $\tau$ such that $\mathbb{U}'(\tau) > n$ since $\mathbb{U}'$ is the optimal optimal prefix-free machine chosen to define K.

$\alpha$ is an order with a very low rate of growth, and actually one can show that it grows more slowly than any computable order.

**Lemma 1.** *For each $h$ computable order:*

*(i) for all $n$, $\alpha(h(n)) = \alpha(n) + O(1)$*
*(ii) for all $n$, $\alpha(n) \leq h(n) + O(1)$*

*Proof.* To prove this lemma, we need the following list of trivial facts.

- By the definition of $\alpha$, there exists $j > n$ such that $\alpha(n) = \mathrm{K}(j)$.
- There exists $c_h$ such that for all $n$, $\mathrm{K}(h(n)) \leq \mathrm{K}(n) + c_h$.
- $\mathrm{K}(n) \geq \alpha(n)$ for all $n$.
- Since $h$ and $\alpha$ are order functions, $h(j) \geq h(n)$ and $\alpha(h(j)) \geq \alpha(h(n))$.

Now, one can apply these facts in order to get:

$$\alpha(n) = \mathrm{K}(j) \geq \mathrm{K}(h(j)) - c_f \geq \alpha(h(j)) - c_f \geq \alpha(h(n)) - c_f.$$

To prove $\alpha(n) \leq \alpha(h(n)) + O(1)$, it suffices to consider an inverse order $\widehat{h}$ of the order function $h$ defined by: $\widehat{h}(n) = \max\{i | h(i) \leq n\}$. Since $\widehat{h}$ is a computable order we have:

$$\alpha(n) \leq \alpha(\widehat{f}(h(n))) \leq \alpha(h(n)) + c.$$

To show that $\alpha(n) \leq h(n) + O(1)$, notice that $\mathrm{K}(n) \leq n + O(1)$ and so there exists $c$ such that $\alpha(n) \leq n + c$. Finally, by the previous point, we have:

$$\alpha(n) \leq \alpha\left(h(n)\right) + c_h \leq h(n) + d.$$

□

We can show that $\alpha$ satisfies 1 for each total computable function, but there exists some partial computable functions such that $\alpha$ does not satisfy 1. In the same way we can show that $\mathrm{K} + \alpha$ does not satisfy condition 2 in the statement of the theorem. However, we have a weaker version for partial functions:

**Lemma 2.** *For each partial computable function $f : \mathbb{N} \to \mathbb{N}$ there exists $c_f$ such that for all $n$*

$$\alpha(\alpha(f(n))) \leq \alpha(n) + c_f.$$

*Proof.* This follows from the following simple fact. For each $f$ there exists $c_f$ such that:

$$\alpha(f(n)) \leq \mathrm{K}(f(n)) \leq \mathrm{K}(n) + c_f \leq n + c_f.$$

Since $\alpha$ is a sub-linear order:

$$\alpha(\alpha(f(n))) \leq \alpha(n + c_f) \leq \alpha(n) + c_f + O(1).$$

□

As stated above, the partial computable functions can increase too quickly to satisfy the second condition of the theorem. For this reason we introduce a general operator to slow down sub-linear and upper semi-computable orders:

**Definition 2 (Star-operator).** *Let $f$ be a sub-linear (i.e. $f(n) = o(n)$) order function. If we set*

$$p_f = \max\{n | f(n) \geq n\}$$

*which is well-defined by sub-linearity of $f$, then, $f^*$ is defined by:*

$$f^*(n) = \min\{k | f^{(k)}(n) \leq p_f\}.$$

This operator is a generalization of the so-called $\log^*$, which is precisely the function one gets by taking $f = \log$ in our definition of $f^*$.

**Remark:** A simpler definition could be $f^*(n) = \min\{k | f^{(k)}(n) = f^{(k+1)}(n)\}$ but for small values of $n$ and for some function $f$ (for functions with more than one fixed point, for example) this definition is not exactly the same and is in fact less natural. This star operator will suit our purposes because it possesses some nice properties.

**Lemma 3.** *Let $f$ a sub-linear order function. The following properties hold:*

1. *$f^*$ is a sub-linear order function.*
2. *If $f$ is a computable function then so is $f^*$.*
3. *If $f$ is a upper semi-computable function then so is $f^*$.*

*4.* $0 \leq f^*(n) - f^*(f^{(i)}(n)) \leq i$.

*Proof.* (1) The last claim will ensure sub-linearity. To see that $f^*$ is non-decreasing, if $x \leq y$ then $f^{(i)}(x) \leq f^{(i)}(y)$ for all $i$ because $f$ is non-decreasing. Finally, $f^*$ is unbounded, for if $f^*$ had a finite limit $d$, then $f^{(d)}$ would be bounded. But this is not possible because $\widehat{f}$ tends to infinity.

(2) If $f$ is computable then to determine $f^*(n)$ we can compute the sequence $f^{(1)}(n), f^{(2)}(n), \ldots, f^{(k)}(n), \ldots$ until we find the first $j$ such that $f^{(j)}(n) \leq p_f$ and we return $j$.

(3) If $f$ is upper semi-computable then we compute in parallel the approximations of $f^{(k)}(n)$ for all $k$, and we return the least $k$ such that $f^{(k)}(n) \leq p_f$.

(4) If $f^*(n) \leq i$ then $f^*(f^{(i)})(n) = 0$ because necessarily, $f^{(i)}(n) \leq p_f$. So $f^*(n) \leq f^*(f^{(i)}(n)) + i$.
If $f^*(n) > i$ then $f^*(n) = f^*(f^{(i)})(n) + i$ by definition of the star-operator. In both cases $f^*(n) \geq f^*(f^{(i)}(n))$.

We shall use Solovay's $\alpha$ function transformed by the star-operator. We will show that the function $\mathrm{A}(x) = \mathrm{K}(x) + \alpha^*(x)$ has all the necessary properties to prove the theorem.

By Lemma 3 the function $\alpha^*$ is upper semicomputable, and thus $\mathrm{K} + \alpha^*$ is as well.

By Lemma 2 we have that for each partial computable function $f : \mathbb{N} \to \mathbb{N}$ there exists $c_f$ such that for all $n$

$$\alpha(\alpha(f(n))) \leq \alpha(n) + c_f$$

and by Lemma 3 (claim 4), if we apply $\alpha^*$ on each term of the previous inequality we have (since $\alpha^*$ is sub-linear)

$$\alpha^*(f(n)) - 2 \leq \alpha^*(\alpha(n) + c_f) \leq \alpha^*(n) + O(1).$$

This proves the second condition of the theorem.

The property 3 is clear because we have $\sum_{x \in 2^{<\omega}} 2^{-\mathrm{K}(x)} < \infty$ and $\mathrm{A}(x) \geq \mathrm{K}(x)$ (because $\alpha^*(x) \geq 0$).

Finally, since $|.|$ is a computable order function, by Lemma 1 we have $\alpha(x) = \alpha(|x|) + O(1)$. By condition 4 of Lemma 3, the equality

$$\alpha^*(x) = \alpha^*(|x|) + O(1)$$

holds. This equality shows that A satisfies hypotheses 4 and 5.

□

It is interesting to notice that the counter-example we produced also invalidates several similar attempts for an axiomatization. For example, one could add the condition:

$$\mathrm{K}(xy) \leq \mathrm{K}(x, y) \leq \mathrm{K}(x) + \mathrm{K}(y) + O(1).$$

But $\mathrm{K} + \alpha^*$ also satisfies this. One could then ask whether the more precise inequality $\mathrm{K}(x,y) \leq K(x) + K(y|x) + O(1)$ could help characterizing conditional prefix-free Kolmogorov complexity, but then again, defining $\alpha(x|y)$ by

$$\alpha(n|m) = \min\{\mathrm{K}(i|m) | i > n\}$$

and then $\alpha^*(.|y)$ for each $y$, we get a counter-example by taking $A(.|.) = K(.|.) + \alpha^*(.|.)$.

This, together with Theorem 3, shows that the situation is more subtle in the prefix-free complexity case than in the plain complexity case. Finding a natural characteristic set of properties for K is left as an open question.

## 4 Acknowledgements

I would like to express my gratitude to Laurent Bienvenu without whom this paper would never had existed. Thanks also to Serge Grigorieff for our numerous discussions during which he helped me progress on this work. Finally, thanks to the Chris Porter and three anonymous reviewers for their help in preparing the final version of this paper.

## References

[Cha66] Gregory J. Chaitin On the Length of Programs for Computing Finite Binary Sequences 1966, J. ACM 13(4): 547-569.

[DH10] Rod G. Downey and Denis Hirschfeldt. *Algorithmic Randomness and Complexity.* Springer, 2010.

[Kol65] Andreï N. Kolmogorov. Three approaches to the definition of the concept “quantity of information”. *Problemy Peredači Informacii*, pages 3–11, 1965.

[MMSV10] Andrej A. Muchnik, Ilya Mezhirov, Alexander Shen, and Nikolay Vereshchagin. Game interpretation of Kolmogorov complexity. Draft version, 2010.

[Nie09] André Nies. *Computability and Randomness.* Oxford University Press, 2009.

[She82] Alexander Shen. Axiomatic description of the entropy notion for finite objects. In *Logika i metodologija nauki*, Vilnjus, 1982. VIII All-USSR conference. Paper in Russian.

[USV10] Vladimir A. Uspensky, Alexander Shen, and Nikolai K. Vereshchagin. Kolmogorov complexity and randomness. Book draft version, 2010.

## Appendix

### 4.1 Theorem 4's full proof

**Theorem 4.** *[She82] Let* A *be a function of* $2^{<\omega} \to \mathbb{N}$. *If* A *verifies:*

1. A *is computable from above.*
2. *For every partial computable function* $f : 2^{<\omega} \to 2^{<\omega}$ *there exists a constant* $c_f$ *such that for each* $\mathrm{A}(f(x)) \leq \mathrm{A}(x) + c_f$ *for each* $x \in 2^{<\omega}$
3. $\mathrm{A}(x) \leq |x| + O(1)$ *for all* $x \in 2^{<\omega}$.
4. $|\{x|\mathrm{A}(x) \leq n\}| = O(2^n)$

   *Then* $\mathrm{A}(x) = \mathrm{C}(x) + O(1)$

This theorem shows that these four properties define exactly plain Kolmogorov complexity (up to an additive constant, of course).

*Proof.* First we show $\mathrm{C}(x) \leq \mathrm{A}(x) + O(1)$. We set $S_n = \{x|\mathrm{A}(x) \leq n\}$. A is computable from above hence $S_n$ is a uniformly computably enumerable set. And by property 4 there exists $d$ independent of $n$) such that $|S_n| < 2^{n+d-1}$.

For each $y$ such that $\mathrm{A}(y) = n$ we can describe $y$ in $S_n$ with a string $\overline{y}$ of length exactly $n + d$, this string represents the rank of $y$ in an enumeration of $S_n$ ($\overline{y}$ is this number with a padding if number does not use $n + d$ bits).

Now we describe an algorithm $E$ to compute $y$ from $\overline{y}$. $E(\overline{y})$ is the $\overline{y}^{\text{th}}$ element in the enumeration of $S_{|\overline{y}|-d}$. To compute $E$ we enumerate the set $\{\langle n, x\rangle | x \in S_n\}$ and count only elements of with the form $\langle |\overline{y}| - d, x\rangle$ and output $x$ for the $\overline{y}^{\text{th}}$. So we have a machine $E$ such that for all $x$ there exists $p$ such that $E(p) = x$ and $|p| \leq \mathrm{A}(x) + O(1)$. By optimality of C we have $\mathrm{C}(x) \leq \mathrm{A}(x) + O(1)$.

Now we prove that $\mathrm{A}(x) \leq \mathrm{C}(x) + O(1)$. If we apply properties 2 and 3 in this order we have the next inequalities (we note $x^*$ the shortest string such that $\mathbb{U}(x^*) = x$):

$$\mathrm{A}(x) = \mathrm{A}(\mathbb{U}(x^*)) \leq \mathrm{A}(x^*) + c_U \leq |x^*| + O(1) = \mathrm{C}(x) + O(1).$$

□

### 4.2 Conditional complexity

In this section we give a more general axiomatic system for conditional plain complexity.

**Theorem 5.** *Let* $\mathrm{B} : 2^{<\omega} \times 2^{<\omega} \to \mathbb{N}$. *If* B *satisfies:*

1. *Uniformly in* $x$, $y$, $\mathrm{B}(x|y)$ *is upper semi-computable.*
2. *For all* $x, y \in 2^{<\omega}$, $\mathrm{B}(x|y) \leq |x| + O(1)$.
3. *For each* $y \in 2^{<\omega}$ *we have* $|\{x|\mathrm{B}(x|y) \leq n\}| = O(2^n)$.
4. $\mathrm{B}(\langle x, y\rangle|y) \leq \mathrm{B}(x|y) + O(1)$.

5. *For all $y$ and for every partial computable function $f$ from $2^{<\omega}$ to $2^{<\omega}$ there exists a constant $c_f$ such that for each $x \in 2^{<\omega}$:*

$$\mathrm{B}(f(x)|y) \leq \mathrm{B}(x|y) + c_f.$$

*Then* $|\mathrm{B}(x|y) - \mathrm{C}(x|y)| = O(1)$.

**Proof.**
For $\mathrm{C}(x|y) \leq \mathrm{B}(x|y) + c$ the proof is very similar to the unconditional version (theorem 4). In the enumeration of $\{\langle n, x\rangle | \mathrm{B}(x|y) < n\}$ if we know $y$, then we can find $x$ given its index in the set hence $\mathrm{C}(x|y) \leq n + O(1)$.

For the converse. We define $x^*_y$ to be the shortest (and first in lexicographic order) string such that $\mathbb{U}(\langle y, x^*_y\rangle) = x$ so $\mathrm{C}(x|y) = |x^*_y|$. And if we apply 5, 4 and 2 (in this order for each inequality) we have:

$$\begin{aligned}\mathrm{B}(x|y) &= \mathrm{B}(\mathbb{U}(\langle x^*_y, y\rangle)|y) \leq \mathrm{B}(\langle x^*_y, y\rangle|y) + O(1)\\ &\leq \mathrm{B}(x^*_y|y) + O(1) \leq |x^*_y| + O(1) = \mathrm{C}(x|y) + O(1)\end{aligned}$$

□

One can ask whether this theorem can be proven with slightly weaker hypotheses. For example we can hope that the hypothesis $\mathrm{B}(x|x) = O(1)$ instead of hypothesis 4 would be sufficient to show the theorem. The next theorem shows that this is not the case.

**Theorem 6.** *There exists a function* $\mathrm{B} : 2^{<\omega} \times 2^{<\omega} \to 2^{<\omega}$ *such that* $\mathrm{B}$ *satisfies the hypotheses 1, 2, 3, 5 (in theorem 5),* $\mathrm{B}(x|x) = O(1)$ *and* $|\mathrm{B}(x|y) - \mathrm{C}(x|y)|$ *is not bounded.*

**Proof.**
We will show that

$$\mathrm{B}(x|y) = \min(\mathrm{C}(x), 2\mathrm{C}(x|y))$$

verifies all these properties but of course $|\mathrm{B}(x|y) - \mathrm{C}(x|y)| \neq O(1)$.

- $\mathrm{B}(x|y)$ is upper semi-computable because it is the min of two functions that are upper semi-computable.
- We know that $\mathrm{C}(x) \leq |x| + O(1)$ so $\mathrm{B}(x|y) \leq |x| + O(1)$.
- We have $|\{x|\mathrm{C}(x|y) \leq n\}| = O(2^n)$ so we have $|\{x|2\mathrm{C}(x|y) \leq n\}| = O(2^{n/2})$. And $|\{x|\mathrm{C}(x) \leq n\}| = O(2^n)$ hence we have $|\{x|\mathrm{B}(x|y) \leq n\}| = O(2^n)$.
- $\mathrm{B}(x|x) = O(1)$ comes from $2C(x|y) \leq 2O(1) = O(1)$.
- For $f : 2^{<\omega} \to 2^{<\omega}$ we must show that there exists $c_f$ such that $\mathrm{B}(f(x)|y) \leq \mathrm{B}(x|y) + c_f$. We know already that there exists $c_f$ such that $\mathrm{C}(f(x)) \leq \mathrm{C}(x) + c_f$ and $2\mathrm{C}(f(x)|y) \leq 2\mathrm{C}(x|y) + c_f$ so we have:
  - If $\mathrm{B}(f(x)|y) = \mathrm{C}(f(x))$ then by definition of B (it is a min):

  $$\mathrm{B}(f(x)|y) = \mathrm{C}(f(x)) \leq 2\mathrm{C}(f(x)|y) \leq 2\mathrm{C}(x|y) + c_f.$$

  In that case we have $\mathrm{B}(f(x)|y) = \mathrm{C}(f(x)) \leq \mathrm{B}(x|y) + O(1)$

- In a same way, if $\mathrm{B}(f(x)|y) = 2\mathrm{C}(f(x)|y)$ by definition of B we have:

$$\mathrm{B}(f(x)|y) = 2\mathrm{C}(f(x)|y) \leq \mathrm{C}(f(x)) \leq \mathrm{C}(x) + c_f.$$

So in that case $\mathrm{B}(f(x)|y) = 2\mathrm{C}(f(x)|y) \leq \mathrm{B}(x|y) + O(1)$

So $\mathrm{B}(f(x)|y) \leq \mathrm{B}(x|y) + c_f$

So B verifies the hypotheses of the theorem and $|\mathrm{B}(x|y) - \mathrm{C}(x|y)|$ is unbounded. □

## Properties of the Solovay's $\alpha$-function

In the paper we have shown that for each $h$ computable order:

- for all $n$, $\alpha(h(n)) = \alpha(n) + O(1)$
- for all $n$, $\alpha(n) \leq h(n) + O(1)$

More generally the proof show that for each total computable function $f$:

$$\alpha(f(n)) \leq \alpha(n) + O(1)$$

because it suffice to consider the total computable order $h_f$ defined by:

$$h_f(n) = \max\{f(k)|n \geq k \geq 0\}$$

with the non-decreasing property of $\alpha$ and Lemma 1 we have:

$$\alpha(f(n)) \leq \alpha(h_f(n)) \leq \alpha(n) + c_{h_f}.$$

As mentioned in the paper this property is not true for partial functions.

**Proposition 1 (Folklore).**
*There exists a partial computable function $f : \mathbb{N} \to \mathbb{N}$ such that*

$$\forall k \in \mathbb{N} \;\; \exists x_k \in 2^{<\omega} \;\; \alpha(f(x_k)) \geq \alpha(x_k) + k.$$

*Proof.* We prove this proposition by contradiction. Let $f$ be a partial computable function such that:

$$f(n) = \begin{cases} \text{time of the computation of } \mathbb{U}'(n) & \text{if } \mathbb{U}'(n) \downarrow \\ \uparrow & \textit{otherwise} \end{cases}$$

where $\mathbb{U}'$ be an optimal machine for K . Let $n_k$ be the integer $i \in [0, 2^k - 1]$ such that the computation time of $\mathbb{U}'(i)$ is maximal (but finite) among all $i \in \mathrm{Dom}(\mathbb{U}') \cap [0, 2^k - 1]$.

With $m$ such that $m \geq f(n_k)$ we can compute a string $x$ such that $\mathrm{K}(x) \geq k$ because we can compute $\mathrm{Dom}(\mathbb{U}') \cap [0, 2^k - 1]$ (because we know an upper bound of the biggest computation time) and take $x \notin (\mathrm{Dom}(\mathbb{U}') \cap [0, 2^k - 1])$, let a function such that $g(m, k) = x$.

Hence for $m \geq f(n_k)$ we have:

$$\mathrm{K}(m) + \mathrm{K}(k) + O(1) \geq \mathrm{K}(g(m,k)) \geq \mathrm{K}(x) \geq k$$

and

$$\mathrm{K}(m) \geq \log(n_k) - O(\log(\log(k))).$$

And finally by definition of $\alpha$ we have $\alpha(f(n_k)) \geq \log(n_k) - O(\log(\log(k)))$

So we cannot have

$$\log(n_k) - O(\log(\log(k))) \leq \alpha(f(n_k)) \leq \alpha(n_k) + c_f$$

because $\alpha(n) \leq \log(\log(n))$ (with Lemma 1 in the proof of Theorem 3).

$\square$

With the same idea we can show that $\mathrm{A} = \mathrm{K} + \alpha$ do not verifies $\mathrm{A}(f(x)) \leq \mathrm{A}(x) + c_f$ because for $n_k$:

$$\mathrm{K}(f(n_k)) = \mathrm{K}(n_k) + O(1).$$

ECCC ISSN 1433-8092

http://eccc.hpi-web.de