Kolmogorov Generic Bases

Extended Abstract

Ana Pereira, João Rasga (Adviser)

October 2011

Abstract

We start by studying the algorithmic complexity of numbers and of strings (the size of the smallest description of the object in question). We prove that there is a correspondence between these two complexities and go on to study algorithmic prefix complexity (a restriction of plain algorithmic complexity). We also define and study a form of logical proof complexity based on the minimum number of lines necessary to demonstrate a formula. Here, we adapt some known results and present some new ones. We then use the similarities between these four types of complexity to define a new structure: the Kolmogorov generic basis. We verify that instances of it coincide with the complexities previously mentioned. Finally, we present some new results, namely that there are classes of Kolmogorov generic bases that induce: incomputable functions (some of which can be approximated by a computable function), computable functions and unbounded functions.

1 Kolmogorov Based Complexities

1.1 Kolmogorov Complexity Based on Partial Recursive Functions

This first presentation of Kolmogorov Complexity is based on Li-Vitányi's book [1], where the proofs of the results in this subsection can be found.

Let $\Sigma = \{0, 1\}$ be the alphabet we will be using throughout. As such, when we refer to a string we will mean an element of Σ^*. We shall use the symbol ε to denote the empty string. We will also need to consider a one-to-one map between the natural numbers and strings in which we associate each string with its index in the length-increasing lexicographic ordering: $(\varepsilon, 0), (0, 1), (1, 2), (00, 3), (01, 4), (10, 5), (11, 6), \ldots$.

Moreover, we define $l(x)$ as the function that returns the length (number of bits) of x, if it is a string, or the length of the string corresponding to x, if it is a natural number. Also, let $[\cdot, \cdot]$ be such that $[x, y] = 1^{l(x)}0\,x\,y$, if x and y are strings, or $[x, y] = 1^{l(x)}0\,x'\,y'$, where x' and y' are the strings corresponding to x and y, if they are natural numbers.

Definition 1.1. Let $x \in \mathbb{N}$, $S \subset \mathbb{N}$ finite and ϕ be a partial recursive function. Define $C^F_\phi(x|S) = \min\{l(p) : \phi([S,p]) = x, p \in \mathbb{N}\}$ and $C^F_\phi(x|S) = \infty$ if there is no such p. ∇

Since there is an additively optimal universal partial recursive function, we can define Kolmogorov complexity specifically for this class of description methods.

Definition 1.2. Consider an $x \in \mathbb{N}$ and $S \subset \mathbb{N}$ finite. Fix an additively optimal universal partial recursive function ϕ_0 and define the *conditional Kolmogorov complexity based on partial recursive functions* $C^F(\cdot|\cdot)$ as $C^F(x|S) = C^F_{\phi_0}(x|S)$. The *unconditional* version $C^F(\cdot)$ is defined by $C^F(x) = C^F(x|\{\})$ for all x. ∇

In computer science terms, we would say that p is a program and ϕ_0 a computer, so $C^F(x)$ is the minimal length of a program for ϕ_0 to compute x without additional input.

Theorem 1.3. *There exist constants c_1 and c_2 such that for all $x \in \mathbb{N}$ and $S \subset \mathbb{N}$, $C^F(x) \leq l(x) + c_1$ and $C^F(x|S) \leq C^F(x) + c_2$.* $\diamondsuit$

1.2 Kolmogorov Complexity Based on Turing Machines

The second presentation of Kolmogorov complexity is based on [8]. Here, instead of using natural numbers or partial recursive functions, strings and Turing machines are used.

Definition 1.4. Let $x \in \Sigma^*$, $S \subset \Sigma^*$ finite and T be a universal Turing machine. Define $C^M_T(x|S) = \min\{l(p) : T([S,p]) = x, p \in \Sigma^*\}$ and $C^M_T(x|S) = \infty$ if there is no such p. ∇

We can prove that there is an additively optimal universal Turing machine, allowing us to give the definition below.

Definition 1.5. Consider an $x \in \Sigma^*$ and $S \subset \Sigma^*$ finite. Fix an additively optimal universal Turing machine T_0 and define the *conditional Kolmogorov complexity based on Turing machines* $C^M(\cdot|\cdot)$ as $C^M(x|S) = C^M_{T_0}(x|S)$. The *unconditional* version $C^M(\cdot)$ is defined by $C^M(x) = C^M(x|\{\})$ for all x. ∇

Essentially, $C^M(x)$ gives the size of the smallest string p which when inputted into Turing machine T_0, outputs x.

It is easy to prove that the two presentations of Kolmogorov complexity we have looked at are equivalent, in the sense that for any Turing machine T, $x \in \Sigma^*$ and $S \subset \Sigma^*$ finite, there is a partial recursive function ϕ such that $C^M_T(x|S) = C^F_\phi(x'|S')$, for some x' and S'. We just need to keep in mind that there is a unique correspondence between partial recursive functions and Turing machines and between natural numbers and strings.

As such, we will be able to prove any statement regarding the Kolmogorov complexity based on partial recursive functions for the one based on Turing machines and vice-versa.

1.3 Prefix Kolmogorov Complexity

In this subsection, we will present a different kind of Kolmogorov complexity, also in [13].

Definition 1.6. Let $x \in \mathbb{N}$, $S \subset \mathbb{N}$ finite and φ be a partial recursive prefix function. Define $K_\varphi(x|S) = \min\{l(p) : \varphi([S, p]) = x, p \in \mathbb{N}\}$ and $K_\varphi(x|S) = \infty$ if there is no such p. ∇

Since there exists an additively optimal universal partial recursive prefix function, we are able to define prefix complexity.

Definition 1.7. Consider an $x \in \mathbb{N}$ and $S \subset \mathbb{N}$ finite. Fix an additively optimal universal partial recursive prefix function φ_0 and define the *conditional prefix complexity* $K(\cdot|\cdot)$ as $K(x|S) = K_{\varphi_0}(x|S)$. The *unconditional* version $K(\cdot)$ is defined by $K(x) = K(x|\{\})$ for all x. ∇

The statement shown below is a corollary of propositions proven in [13].

Proposition 1.8. *There exist constants c_1 and c_2 such that for all $x \in \mathbb{N}$ and $S \subset \mathbb{N}$, $K(x) \leq l(x) + 2log_2(l(x)) + c_1$ and $K(x|S) \leq K(x) + 2log_2(K(x)) + c_2$.* $\diamondsuit$

1.4 Kolmogorov Proof Complexity

This subsection is based on the definitions and results of [1, 3, 6].

Consider the set of formulas over a set of propositional variables Ξ, denoted by L. Denote by L^+ the set of all finite sequences over L, except for the empty sequence. Furthermore, consider two distinct $\xi_1, \xi_2 \in \Xi$. Define $\langle \cdot, \cdot \rangle : L^* \times L^+ \to L^+$ such that $\langle \psi_1 ... \psi_m, \phi_1 ... \phi_n \rangle = \xi_1^m \, \xi_2 \, \psi_1 ... \psi_m \, \phi_1 ... \phi_n$. Also, define $| \cdot | : L^+ \to \mathbb{N}$ as the length of the sequence in question, i.e., the number of formulas it has.

Definition 1.9. Consider a deductive system D. The *propositional proof system for L induced by D* is a function $\Delta_D : L^+ \to L$ that expects an input of the form $\langle \psi_1 ... \psi_m, \phi_1 ... \phi_n \rangle$, where $\psi_1 ... \psi_m \in L^*$and $\phi_1 ... \phi_n \in L^+$, and returns ϕ_n if $(\phi_1, ..., \phi_n)$ is a derivation sequence for ϕ_n in D with hypothesis in $\{\psi_1, ..., \psi_m\}$. Otherwise, Δ_D is undefined. ∇

Notation 1.10. Let $\varphi \in L$, $\Gamma \subseteq L$, d be a derivation and Δ_D the propositional proof system induced by a deductive system D. We write $\Gamma \vdash^d_{\Delta_D} \varphi$ if there exists an $\Gamma' \subseteq \Gamma$ finite such that $\Delta_D(\langle \Gamma', d \rangle) = \varphi$. We also write $\Gamma \vdash_{\Delta_D} \varphi$ if there is a $d \in L^+$ such that $\Gamma \vdash^d_{\Delta_D} \varphi$.

Definition 1.11. Let $\varphi \in L$, $S \subseteq L$ and $\mathcal{F}$ be a Frege system. Define $CL_\mathcal{F}(\varphi|S) = \min\{|d| : S \vdash^d_{\Delta_\mathcal{F}} \varphi, d \in L^+\}$ and $CL_\mathcal{F}(\varphi|S) = \infty$ if there is no such d. ∇

Since the size of the smallest proof for a formula only varies with the Frege system in question by a polynomial amount, we are able to define proof complexity as below.

Definition 1.12. Consider an $\varphi \in L$ and $S \subseteq L$. Fix a Frege system $\mathcal{F}_0$ and define the *conditional Kolmogorov proof complexity* $CL(\cdot|\cdot)$ as $CL(\varphi|S) = CL_{\mathcal{F}_0}(\varphi|S)$. The *unconditiona*l version $CL(\cdot)$ is defined as $CL(\varphi) = CL(\varphi|\{\})$ for all formulas φ. ∇

As such, $CL(\varphi|S)$ is the size of the smallest derivation for φ in a Frege system $\mathcal{F}_0$, using the elements of S as hypotheses. From now on, we will write $\vdash$ instead of $\vdash_{\Delta_{\mathcal{F}_0}}$.

Proposition 1.13. *Let ψ be a formula in L and Γ a set of formulas in L. Then, $CL(\psi|\Gamma) \leq CL(\psi)$.* $\diamondsuit$

Proof. Any derivation for a formula given no hypotheses is also a derivation for the same formula given any set of hypotheses. As such, if d is a derivation sequence such that $CL(\psi) = |d|$ with $\vdash^d \psi$, then d also witnesses $\Gamma \vdash \psi$ and $|d| \geq CL(\psi|\Gamma)$. □

For any derivation d, define $\rho(d)$ as the number of distinct subformulas in d. We can prove that the number of subformulas in the smallest proof for a formula only varies with the Frege system used by a polynomial amount, allowing us to give the definition below.

Definition 1.14. Let $\mathcal{F}_0$ be the Frege system used for CL. For any formula φ in L, define $\tau(\varphi) = \min\{\rho(d) : \vdash^d \varphi\}$ and $\tau(\varphi) = \infty$ if there is no such d. ∇

Proposition 1.15. *For every formula φ in L, $CL(\varphi) \leq \tau(\varphi)$.* $\diamondsuit$

Proof. For any derivation d' such that $\vdash^{d'} \varphi$, note that there is at least one new subformula for each line. As such, $\rho(d') \geq |d'|$. Let d be a derivation sequence such that $|d| = CL(\varphi)$ and $\vdash^d \varphi$. It follows from this that $|d| \leq \rho(d')$. Hence, $|d|$ is smaller or equal to the number of subformulas in any derivation that witnesses $\vdash \varphi$, which means $|d| = CL(\varphi) \leq \tau(\varphi)$. □

2 Kolmogorov Generic Bases

2.1 Definition and Examples

In this subsection, we introduce the concept of Kolmogorov Generic Basis, a structure designed to embody all four types of complexity found in the previous section but that is not limited to these.

Definition 2.1. A *Kolmogorov generic basis* is a tuple (A, D, M, F, v) such that:

- A and D are two non-empty recursively enumerable sets[1];
- $M : D \to \mathbb{N}$ is a computable function[2];

[1] A is the set of objects we want to describe and D is the set of objects we use to describe the elements of A.

[2] M is the function of D we want to minimize.

- F is a non-empty subset of the functions from D to A that can be simulated by deterministic Turing machines[3];
- $v : A \times 2^A \times F \times D \to \{0,1\}$ is a function[4].

This tuple induces another function $g : A \times 2^A \times F \to \mathbb{N} \cup \{\infty\}$ such that $g(a, S, f) = \min\{M(d) : d \in D, v(a, S, f, d) = 1\}$, also written as $g_f(a|S)$. If no such d exists, then we define $g_f(a|S)$ as ∞. Also, we will abbreviate $g_f(a|\{\})$ as $g_f(a)$.

A Kolmogorov generic basis must also have the following properties:

1. The particular specification method f used only affects the value of $g_f(a|S)$ by additive and multiplicative constants, allowing us to fix one and define $g(\cdot|\cdot) = g_f(\cdot|\cdot)$.
2. $g(\cdot)$ has an $h(\cdot)$, function of its input, as an upper bound;
3. $g(\cdot|\cdot)$ has the unconditional version as an upper bound.

We call $g(\cdot|\cdot)$ the *conditional Kolmogorov generic function for basis* (A, D, M, F, v) and $g(\cdot)$ the *unconditional* version. ∇

Looking at this definition, we can see that a Kolmogorov generic function g returns the smallest possible value for an element of D over M satisfying verification function v. This is how C^F, C^M, K and CL work.

Example 2.2. To get the Kolmogorov complexity based on partial recursive functions, we have to consider $A = D = \mathbb{N}$, F as the set of all partial recursive functions from over $\mathbb{N}$, $M(d) = l(d)$ and $v(a, S, f, d) = \begin{cases} 1, & S \text{ is finite and } f([S, d]) = a \\ 1, & S \text{ is infinite and } f([S', d]) = a \text{ for some finite } S' \subset S \\ 0, & \text{otherwise} \end{cases}$. Properties 1 to 3 of a Kolmogorov generic basis are ensured by the fact that there exists an additively optimal universal partial recursive function and by Theorem 1.3.

We then get that $g_f(a|S) = \begin{cases} C^F_f(a|S), & S \text{ is finite} \\ \min\{C^F_f(a|S') : S' \subset S \text{ is finite}\}, & S \text{ is infinite} \end{cases}$. □

Example 2.3. To arrive at the Kolmogorov complexity based on Turing machines, we have to consider $A = D = \{0,1\}^*$, F as the set of deterministic Turing Machines, $M(d) = l(d)$ and $v(a, S, f, d) = \begin{cases} 1, & S \text{ is finite and } f([S, d]) = a \\ 1, & S \text{ is infinite and } f([S', d]) = a \text{ for some finite } S' \subset S \\ 0, & \text{otherwise} \end{cases}$. The

[3] F is the set of description methods we use to interpret each element of D and arrive at an element of A.

[4] v is the function that tells us whether the element of D in question effectively describes the inputted element of A.

properties of a Kolmogorov generic basis are ensured the fact that there exists an additively optimal universal partial recursive function and by Theorem 1.3 since C^F and C^M are equivalent, in the sense we explained in Subsection 1.2.

As such, $g_f(a|S) = \begin{cases} C_f^M(a|S), & S \text{ is finite} \\ \min\{C_f^M(a|S') : S' \subset S \text{ is finite}\}, & S \text{ is infinite} \end{cases}$. □

Example 2.4. To get the prefix Kolmogorov complexity we have to consider $A = D = \mathbb{N}$, F as the set of all partial recursive prefix functions from $\mathbb{N}$ to $\mathbb{N}$, $M(d) = l(d)$ and $v(a, S, f, d) = \begin{cases} 1, & S \text{ is finite and } f([S, d]) = a \\ 1, & S \text{ is infinite and } f([S', d]) = a \text{ for some finite } S' \subset S \\ 0, & \text{otherwise} \end{cases}$. Properties 1 to 3 of a Kolmogorov generic basis are ensured the fact that there exists an additively optimal universal partial recursive prefix function and by Proposition 1.8.

Then, $g_f(a|S) = \begin{cases} K_f(a|S), & S \text{ is finite} \\ \min\{K_f(a|S') : S' \subset S \text{ is finite}\}, & S \text{ is infinite} \end{cases}$. □

Example 2.5. To arrive at the Kolmogorov proof complexity, we use A as the set of formulas over a set of propositional variables Ξ, $D = A^+$, F as the set of all propositional proof systems induced for A by Frege systems, $M(d) = |d|$ and $v(a, S, f, d) = \begin{cases} 1, & S \vdash_f^d a \\ 0, & \text{otherwise} \end{cases}$. The properties of a Kolmogorov generic basis are ensured by the fact that the size of the smallest proof for the same formula only varies with the Frege system in question by a polynomial amount and by Propositions 1.13 and 1.15.

Using the basis above, we get that $g_f(a|S) = CL_f(a|S)$. □

2.2 Results

Having defined this new structure, we now work on proving some results for it.

For any recursively enumerable set S, let $i_S : S \to \mathbb{N}$ be the function such that $i_S(s)$ is the index of s in a fixed enumeration of S. As such, $i_S^{-1}(n)$ returns the n^{th} element of the enumeration.

Define $\mathcal{C}_1$ as the class of Kolmogorov generic bases such that, for any enumeration of A, there exist constants k_1 and k_2 such that, for any $a \in A$, $g(a) \leq k_1 C^F(i_A(a)) + k_2$.

Theorem 2.6. *For any basis in $\mathcal{C}_1$, the induced unconditional Kolmogorov generic function g is not computable.* ♢

Proof. For each basis in $\mathcal{C}_1$, suppose there exists a partial recursive function ϕ defined on an infinite set of points that coincides with the induced $g \circ i_A^{-1}$ over the whole of its domain of definition, for some enumeration of A. Consider an infinite recursive subset of its domain of definition $B \subseteq \mathbb{N}$ and the function $\psi : \mathbb{N} \to B$ such that $\psi(m) = \min\{n \in B : g(i_A^{-1}(n)) \geq m\}$. This function is partial recursive since $g \circ i_A^{-1}$ coincides with ϕ in B. By the definition of ψ, $g(i_A^{-1}(\psi(m))) \geq m$. Since ψ is a partial recursive function, we can apply the invariance theorem for C^F and get that $C^F(\psi(m)) \leq C_\psi^F(\psi(m)) + c \leq l(m) + c' = \log(m) + c''$, for some constants c, c', c''. As such, we get that there exist constants c_1 and c_2 such that for all $m \in \mathbb{N}$, $m \leq c_1 \log(m) + c_2$, which is false from some m onward. Hence, our assumption was incorrect and g is not computable. □

It is obvious that the Kolmogorov generic basis that induces C^F is in $\mathcal{C}_1$ and it can also be shown that the bases for C^M and K are in this class as well. However, bases inducing Kolmogorov proof complexity may not to be a part of this class since we cannot specify a relation between CL and C^F.

Define $\mathcal{C}_2$ as the class of Kolmogorov generic bases such that: for all $f \in F$, f is defined for all inputs; for all $a \in A, f \in F$ and $d \in D$, $v(a, \{\}, f, d) = 1$ if and only if $f(d) = a$; for all $i \in \mathbb{N}$, the set $\{d \in D : M(d) = i\}$ is finite.

Theorem 2.7. *A Kolmogorov generic function induced by a basis in $\mathcal{C}_2$ is computable.* ♢

Proof. Since D is recursively enumerable and, for all $i \in \mathbb{N}$, the set $\{d \in D : M(d) = i\}$ is finite, we can define an enumeration of D such that its elements are organized in ascending order of $M(d)$. Let i_D be given by one such enumeration. Note that $g_f(i_A^{-1}(n)) = min\{M(d) : v(i_A^{-1}(n), \{\}, f, d) = 1\} = min\{M(d) : f(d) = i_A^{-1}(n)\}$ and since any f in F is defined for all inputs, we can just test whether for a given n and d $f(d) = i_A^{-1}(n)$ and always have an answer. Furthermore, since the elements of F are either partial recursive functions or deterministic Turing machines, each can be uniquely linked to a Turing machine. As such, each f in F is associated to the index in an enumeration of all Turing machines of the one it is linked to. Denote this index by $i_F(f)$.

Now, consider a Turing machine that receives the strings corresponding to an $n \in \mathbb{N}$ and $i_F(f)$ for an $f \in F$ and follows the algorithm below in order to return $g_f(i_A^{-1}(n))$.

```
Set $j = 0$, $r = 0$;
While ($r == 0$) do {If {the Turing machine associated with $f$ outputs $i_A^{-1}(n)$ on input $i_D^{-1}(j)$} then {set $r = 1$}; otherwise {set $j = j + 1$}};
Return $M(i_D^{-1}(j))$.
```

As such, the induced $g_f \circ i_A^{-1}$ is computable. □

Class $\mathcal{C}_2$ doesn't have any of the examples we studied: for C^F, C^M and K the first condition doesn't hold and for CL the second and third conditions don't hold. However, inside $\mathcal{C}_2$ there is a simple Kolmogorov generic basis described below.

Example 2.8. Consider the basis (A, D, M, F, v) where $A = D = \mathbb{N}$, $F = \{f(x) = x\}$, $M(d) = d$ and $v(a, S, f, d) = \begin{cases} 1, & d = a \\ 0, & \text{otherwise} \end{cases}$. Under these conditions, $g_f(a|S) = \min\{d : d \in \mathbb{N}, d = a\} = a$, which obviously verifies all three properties of a Kolmogorov generic basis and is in $\mathcal{C}_2$. We call the Kolmogorov generic function induced by this basis g^* . □

We now consider the class $\mathcal{C}_3$ of Kolmogorov generic bases such that for all $i \in \mathbb{N}$, the set $\{d \in D : M(d) = i\}$ is finite. Obviously, $\mathcal{C}_2 \subset \mathcal{C}_3$.

Theorem 2.9. *For any basis in $\mathcal{C}_3$, define $m : \mathbb{N} \to \mathbb{N}$ such that $m(x) = min\{g(i_A^{-1}(y)) : y \geq x\}$ for some enumeration of A. The function m is unbounded.* ♢

Proof. For each i there is a least an x_i such that for all $x > x_i$, $g(i_A^{-1}(x)) \geq i$. Clearly, for all i we have $x_{i+1} \geq x_i$. Consider an $x_i < x \leq x_{i+1}$. Then, $g(i_A^{-1}(x)) = i$ and $m(x) = \min\{g(i_A^{-1}(y)) : y \geq x\} = g(i_A^{-1}(x)) = i$ so m is unbounded. □

As a corollary of this result, we now know that for $\mathcal{C}_3$ the induced Kolmogorov generic functions are unbounded. Within this class, we have the bases for C^F, C^M and K but not for CL. However, a modified version of it can be found in $\mathcal{C}_3$.

Consider a Kolmogorov generic basis for CL, but with the difference that the number of propositional variables is finite and there is a maximum number of connectives and propositional variables for each formula. Then there would be a finite number of derivation sequences for each length and this basis would be in $\mathcal{C}_3$. For purposes of Figure 1, we shall call the induced Kolmogorov generic function CL^*. Furthermore, since $\mathcal{C}_2$ is a subset of $\mathcal{C}_3$, g^* belongs to $\mathcal{C}_3$ as well.

Define $\mathcal{C}_4$ as the class of Kolmogorov generic bases (A, D, M, F, v) in $\mathcal{C}_3$ such that the function h from property 2 of Definition 2.1 is total recursive and for all $a \in A, f \in F$ and $d \in D$, $v(a, \{\}, f, d) = 1$ if and only if $f(d) = a$.

Theorem 2.10. *For any Kolmogorov generic basis in $\mathcal{C}_4$, there is a total recursive function $\phi(t, a)$, monotonic and decreasing in t, such that $lim_{t\to\infty}\phi(t, a) = g(a)$.* ♢

Proof. Consider a Kolmogorov generic basis in $\mathcal{C}_4$. Since for any f in F, its behaviour can be simulated by a deterministic Turing machine, consider one such f and the machine that simulates it, M. Also, the induced unconditional Kolmogorov function will be $g_f(a) = \min\{M(d) : d \in D, v(a, \{\}, f, d) = 1\} = \min\{M(d) : d \in D, f(d) = a\}$, since v is such that $v(a, \{\}, f, d) = 1$ if and only if $f(d) = a$.

As such, we can construct ϕ on input t and a such that we run machine M for t steps on each element of $\{d \in D : M(d) \leq h(a)\}$ and if for any such input d the computation halts with output a, then define the value of $\phi(t, a)$ as the smallest value of $M(d)$ for those d's. Otherwise, define it as equal to $h(a)$.

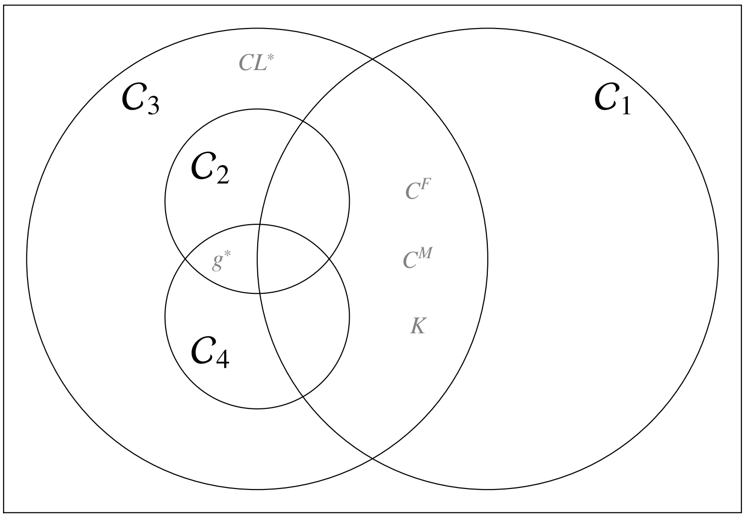

Figure 1: Some Kolmogorov generic bases notable classes: $\mathcal{C}_1$ has non computable functions, $\mathcal{C}_2$ has computable functions, $\mathcal{C}_3$ has unbounded functions and $\mathcal{C}_4$ has functions that can be approximated by a computable function.

It is easy to observe that $\phi(t, a)$ is recursive, total, and monotonically decreasing with t. Looking at $\lim_{t\to\infty}\phi(t, a)$, the limit exists, since for each a there exists a t such that M halts with output a after computing t steps starting with input d and satisfying $M(d) = g_f(a)$. □

Inside this new class, we have the basis from Example 2.8. However, none of the other instances of Kolmogorov generic bases we looked at are in $\mathcal{C}_4$.

Figure 1 summarizes the results we presented in this subsection and shows how these classes are related with each other. When looking at the figure, one should keep in mind that, for brevity's sake, we did not write the bases we have talked about but the induced functions.

References

[1] In Samuel R. Buss, editor, *Handbook of Proof Theory*. Elsevier, 1998.

[2] Maria Luisa Bonet and Samuel R. Buss. The deduction rule and linear and near-linear proof simulations. *Journal of Symbolic Logic*, 58:688–709, 1993.

[3] Samuel R. Buss. Some remarks on lengths of propositional proofs. *Archive for Mathematical Logic*, 34:377–394, 1995.

[4] Samuel R. Buss. Towards NP-P via proof complexity and search. Technical report, 2011. To appear.

[5] Stephen Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity.* Cambridge University Press, 1st edition, 2010.

[6] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44:36–50, 1979.

[7] Lance Fortnow. Kolmogorov complexity and computational complexity. In *Complexity of Computations and Proofs. Quaderni di Matematica*, 2004.

[8] Lance Fortnow, John M. Hitchcock, A. Pavan, N. V. Vinodchandran, and Fengming Wang. Extracting kolmogorov complexity with applications to dimension zero-one laws. In *In procedings of the 33rd International Colloquium on Automara, Languages and Programing*, pages 335–345. Springer-Verlag, 2006.

[9] Nicola Galesi. On the complexity of propositional proof systems. Technical report, 2000. PhD Thesis.

[10] Alexander Hertel. Propositional proof complexity: A depth oral survey. Technical report, 2005. Depth Oral (PhD Candidacy) Paper.

[11] Jan Krajíček. Speed-up for propositional frege systems via generalizations of proofs. *Commentationes Mathematicae Universitatis Carolina*, 30:137–140, 1989.

[12] Jan Krajíček. *Bounded arithmetic, propositional logic, and complexity theory.* Cambridge University Press, 1995.

[13] Ming Li and Paul Vitányi. *An introduction to Kolmogorov Complexity and Its Applications.* Springer, 2008.

[14] John Alan Robinson. A machine oriented logic based on the resolution principle. *Journal of the Association for the Computing Machinery*, 12:627–631, 1965.

[15] Amílcar Sernadas and Cristina Sernadas. *Foundations of Logic and Theory of Computation.* College Publications, 2008.

[16] Michael Sipser. *Introduction to the Theory of Computation.* International Thomson Publishing, 1st edition, 1996.

[17] Osamu Watanabe. *Kolmogorov Complexity and Computational Complexity.* Springer-Verlag, 1992.