Notre Dame Journal of Formal Logic

Program Size Complexity for Possibly Infinite Computations

Verónica Becher, Santiago Figueira, André Nies, and Silvana Picchi

Abstract We define a program size complexity function H^∞ as a variant of the prefix-free Kolmogorov complexity, based on Turing monotone machines performing possibly unending computations. We consider definitions of randomness and triviality for sequences in $\{0,1\}^\omega$ relative to the H^∞ complexity. We prove that the classes of Martin-Löf random sequences and H^∞-random sequences coincide, and that the H^∞-trivial sequences are exactly the recursive ones. We also study some properties of H^∞ and compare it with other complexity functions. In particular, H^∞ is different from H^A, the prefix-free complexity of monotone machines with oracle A.

1 Introduction

We consider monotone Turing machines (a one-way read-only input tape and a one-way write-only output tape) performing possibly infinite computations, and we define a program size complexity function $H^\infty : \{0,1\}^* \to \mathbb{N}$ as a variant of the classical Kolmogorov complexity: given a universal monotone machine $\mathcal{U}$, for any string $x \in \{0,1\}^*$, $H^\infty(x)$ is the length of a shortest string $p \in \{0,1\}^*$ read by $\mathcal{U}$, which produces x via a possibly infinite computation (either a halting or a non halting computation), having read exactly p from the input.

The classical prefix-free complexity H [2; 10] is an upper bound of the function H^∞ (up to an additive constant), since the definition of H^∞ does not require that the machine $\mathcal{U}$ halts. We prove that H^∞ differs from H in that it has no monotone decreasing recursive approximation and it is not subadditive.

The complexity H^∞ is closely related with the monotone complexity Hm, independently introduced by Levin [8] and Schnorr [13] (see [15] and [11] for historical details and differences between various monotone complexities).

Printed February 9, 2004

Levin defines $Hm(x)$ as the length of the shortest halting program that provided with n ($0 \le n \le |x|$), outputs $x \restriction n$. Equivalently $Hm(x)$ can be defined as the least number of bits read by a monotone machine $\mathcal{U}$ which via a possibly infinite computation produces any finite or infinite extension of x.

Hm is a lower bound of H^∞ (up to an additive constant) since the definition of H^∞ imposes that the machine $\mathcal{U}$ reads exactly the input p and produces exactly the output x. Every recursive $A \in \{0,1\}^\omega$ is the output of some monotone machine with no input, so there is some c such that $\forall n\ Hm(A \restriction n) \le c$. Moreover, there exists n_0 such that $\forall n, m \ge n_0$, $Hm(A \restriction n) = Hm(A \restriction m)$. We show this is not the case with H^∞, since for every infinite $B = \{b_1, b_2, \ldots\} \subseteq \{0,1\}^*$, $\lim_{n\to\infty} H^\infty(b_n) = \infty$. This is also a property of the classical prefix-free complexity H, and we consider it as a decisive property that distinguishes H^∞ from Hm.

The prefix-free complexity of a universal machine with oracle $\varnothing'$, the function $H^{\varnothing'}$, is also a lower bound of H^∞ (up to an additive constant). We prove that for infinitely many strings x, the complexities $H(x)$, $H^\infty(x)$ and $H^{\varnothing'}(x)$ separate as much as we want. This already proves that these three complexities are different. In addition we show that for every oracle A, H^∞ differs from H^A, the prefix-free complexity of a universal machine with oracle A.

For sequences in $\{0,1\}^\omega$ we consider definitions of randomness and triviality based on the H^∞ complexity. A sequence is H^∞-random if its initial segments have maximal H^∞ complexity. Since Hm gives a lower bound of H^∞ and Hm-randomness coincides with Martin-Löf randomness [9], the classes of Martin-Löf random, H^∞-random and Hm-random coincide.

We argue for a definition of H^∞-trivial sequences as those whose initial segments have minimal H^∞ complexity. While every recursive $A \in \{0,1\}^\omega$ is both H-trivial and H^∞-trivial, we show that the class of H^∞-trivial sequences is strictly included in the class of H-trivial sequences. Moreover, in Theorem 5.6, the main result of the paper, we characterize the recursive sequences as those which are H^∞-trivial.

2 Definitions

$\mathbb{N}$ is the set of natural numbers, and we work with the binary alphabet $\{0,1\}$. As usual, a string is a finite sequence of elements of $\{0,1\}$, λ is the empty string and $\{0,1\}^*$ is the set of all strings. $\{0,1\}^\omega$ is the set of all infinite sequences of $\{0,1\}$, i.e. the Cantor space, and $\{0,1\}^{\le\omega} = \{0,1\}^* \cup \{0,1\}^\omega$ is the set of all finite or infinite sequences of $\{0,1\}$.

For $s \in \{0,1\}^*$, $|s|$ denotes the length of s. If $s \in \{0,1\}^*$ and $A \in \{0,1\}^\omega$ we denote by $s \restriction n$ the prefix of s with length $\min\{n, |s|\}$ and by $A \restriction n$ the length n prefix of the infinite sequence A. We consider the prefix ordering $\preceq$ over $\{0,1\}^*$, i.e, for $s, t \in \{0,1\}^*$ we write $s \preceq t$ if s is a prefix of t. We assume the recursive bijection $string : \mathbb{N} \to \{0,1\}^*$ such that $string(i)$ is the i-th string in the length and lexicographic order over $\{0,1\}^*$.

If f is any partial map then, as usual, we write $f(p)\downarrow$ when it is defined, and $f(p)\uparrow$ otherwise.

2.1 Possibly infinite computations on monotone machines A monotone machine is a Turing machine with a one-way read-only input tape, some work tapes, and a one-way write-only output tape. The input tape contains a first dummy cell (representing the empty input) and then a one-way infinite sequence of 0's and 1's, and initially the input head scans the leftmost dummy cell. The output tape is written one symbol of $\{0,1\}$ at a time (the output grows with respect to the prefix ordering in $\{0,1\}^*$ as the computational time increases).

A possibly infinite computation is either a halting or a non halting computation. If the machine halts, the output of the computation is the finite string written on the output tape. Else, the output is either a finite string or an infinite sequence written on the output tape as a result of a never ending process. This leads us to consider $\{0,1\}^{\leq\omega}$ as the output space.

In this work we restrict ourselves to possibly infinite computations on monotone machines which read just finitely many symbols from the input tape.

Definition 2.1 Let $\mathcal{M}$ be a monotone machine. $M(p)[t]$ is the *current* output of $\mathcal{M}$ on input p at stage t if it has not read beyond the end of p. Otherwise, $M(p)[t]\uparrow$. Notice that $M(p)[t]$ does not require that the computation on input p halts.

Remark 2.2

1. If $M(p)[t]\uparrow$ then $M(q)[u]\uparrow$ for all $q \preceq p$ and $u \geq t$
2. If $M(p)[t]\downarrow$ then $M(q)[u]\downarrow$ for any $q \succeq p$ and $u \leq t$. Also, if at stage t, $\mathcal{M}$ reaches a halting state without having read beyond the end of p, then $M(p)[u]\downarrow = M(p)[t]$ for all $u \geq t$.
3. Since $\mathcal{M}$ is monotone, $M(p)[t] \preceq M(p)[t+1]$, in case $M(p)[t+1]\downarrow$
4. $M(p)[t]$ has recursive domain

Definition 2.3 Let $\mathcal{M}$ be a monotone machine.

1. The input/output behavior of $\mathcal{M}$ for *halting computations* is the partial recursive map $M : \{0,1\}^* \to \{0,1\}^*$ given by the usual computation of $\mathcal{M}$, i.e., $M(p)\downarrow$ iff $\mathcal{M}$ enters into a halting state on input p without reading beyond p. If $M(p)\downarrow$ then $M(p) = M(p)[t]$ for some stage t at which $\mathcal{M}$ entered a halting state.
2. The input/output behavior of $\mathcal{M}$ for *possibly infinite computations* is the map $M^\infty : \{0,1\}^* \to \{0,1\}^{\leq\omega}$ given by $M^\infty(p) = \lim_{t\to\infty} M(p)[t]$

Proposition 2.4

1. *domain*(M) *is closed under extensions and its syntactical complexity is* Σ^0_1
2. *domain*(M^∞) *is closed under extensions and its syntactical complexity is* Π^0_1
3. M^∞ *extends* M

Proof

1. is trivial.
2. $M^\infty(p)\downarrow$ iff $\forall t$ $\mathcal{M}$ on input p does not read $p0$ and does not read $p1$. Clearly, *domain*(M^∞) is closed under extensions since if $M^\infty(p)\downarrow$ then $M^\infty(q)\downarrow = M^\infty(p)$ for every $q \succeq p$.

3. Since the machine $\mathcal{M}$ is not required to halt, M^∞ extends M.

□

Remark 2.5 An alternative definition of the functions M and M^∞ would be to consider them with prefix-free domains (instead of closed under extensions):

- $M(p)\downarrow$ iff at some stage t $\mathcal{M}$ enters a halting state having read exactly p. If $M(p)\downarrow$ then its value is $M(p)[t]$ for such stage t.
- $M^\infty(p)\downarrow$ iff $\exists t$ at which $\mathcal{M}$ has read exactly p and for every t' $\mathcal{M}$ does not read $p0$ nor $p1$. If $M^\infty(p)\downarrow$ then its value is $\lim_{t\to\infty} M(p)[t]$.

We fix an effective enumeration of all tables of instructions. This gives an effective $(\mathcal{M}_i)_{i\in\mathbb{N}}$. We also fix the usual monotone universal machine $\mathcal{U}$, which defines the functions $U(0^i1p) = M_i(p)$ and $U^\infty(0^i1p) = M_i^\infty(p)$ for halting and possibly infinite computations respectively. As usual, $i+1$ is the coding constant of $\mathcal{M}_i$. Recall that U^∞ is an extension of U. We also fix $\mathcal{U}^{\varnothing'}$ a monotone universal machine with an oracle for $\varnothing'$.

By Shoenfield's Limit Lemma every $M^\infty : \{0,1\}^* \to \{0,1\}^*$ is recursive in $\varnothing'$. However, possibly infinite computations on *monotone* machines cannot compute all $\varnothing'$-recursive functions. For instance, the characteristic function of the halting problem cannot be computed in the limit by a monotone machine. In contrast, the Busy Beaver function in unary notation $bb : \mathbb{N} \to 1^*$:

$bb(n)$ = the maximum number of 1's produced by any Turing machine with n states which halts with no input

is just $\varnothing'$-recursive and $bb(n)$ is the output of a non halting computation which on input n, simulates every Turing machine with n states and for each one that halts updates, if necessary, the output with more 1's.

2.2 Program size complexities on monotone machines Let $\mathcal{M}$ be a monotone machine, and M, M^∞ the respective maps for the input/output behavior of $\mathcal{M}$ for halting computations and possibly infinite computations (Definition 2.3). We denote the usual prefix-free complexity [2; 10; 7] for M by $H_\mathcal{M} : \{0,1\}^* \to \mathbb{N}$

$$H_\mathcal{M}(x) = \begin{cases} \min\{|p| : M(p) = x\} & \text{if } x \text{ is in the range of } M \\ \infty & \text{otherwise} \end{cases}$$

Definition 2.6 $H^\infty_\mathcal{M} : \{0,1\}^{\le\omega} \to \mathbb{N}$ is the program size complexity for functions M^∞.

$$H^\infty_\mathcal{M}(x) = \begin{cases} \min\{|p| : M^\infty(p) = x\} & \text{if } x \text{ is in the range of } M^\infty \\ \infty & \text{otherwise} \end{cases}$$

For $\mathcal{U}$ we drop subindexes and we simply write H and H^∞. The Invariance Theorem holds for H^∞:

$$\forall \text{ monotone machine } \mathcal{M}\ \exists c\ \forall s \in \{0,1\}^{\le\omega}\ H^\infty(s) \le H^\infty_\mathcal{M}(s) + c.$$

The complexity function H^∞ was first introduced in [1] without a detailed study of its properties. Notice that if we take monotone machines $\mathcal{M}$ according to Remark 2.5 instead of Definition 2.3, we obtain *the same* complexity functions $H_\mathcal{M}$ and $H^\infty_\mathcal{M}$.

In this work we only consider the H^∞ complexity of finite strings, that is, we restrict our attention to $H^\infty : \{0,1\}^* \to \mathbb{N}$. We will compare H^∞ with these other complexity functions:

H^A : $\{0,1\}^* \to \mathbb{N}$ is the program size complexity function for $\mathcal{U}^A$, a monotone universal machine with oracle A. We pay special attention to $A = \varnothing'$.

Hm : $\{0,1\}^{\leq\omega} \to \mathbb{N}$ (see [8]), where $Hm_{\mathcal{M}}(x) = \min\{|p| : M^\infty(p) \succeq x\}$ is the *monotone complexity function* for a monotone machine $\mathcal{M}$ and, as usual, for $\mathcal{U}$ we simply write Hm.

We mention some known results that will be used later.

Proposition 2.7 (For items 1. and 2. see [2], for item 3. see [1])

1. $\forall s \in \{0,1\}^*\ H(s) \leq |s| + H(|s|) + \mathcal{O}(1)$
2. $\forall n\ \exists s \in \{0,1\}^*$ *of length* n *such that:*
   (a) $H(s) \geq n$
   (b) $H^{\varnothing'}(s) \geq n$
3. $\forall s \in \{0,1\}^*\ H^{\varnothing'}(s) < H^\infty(s) + \mathcal{O}(1)$ *and* $H^\infty(s) < H(s) + \mathcal{O}(1)$

3 H^∞ is different from H

The following properties of H^∞ are in the spirit of those of H.

Proposition 3.1 *For all strings* s *and* t

1. $H(s) \leq H^\infty(s) + H(|s|) + \mathcal{O}(1)$
2. $\#\{s \in \{0,1\}^* : H^\infty(s) \leq n\} < 2^{n+1}$
3. $H^\infty(ts) \leq H^\infty(s) + H(t) + \mathcal{O}(1)$
4. $H^\infty(s) \leq H^\infty(st) + H(|t|) + \mathcal{O}(1)$
5. $H^\infty(s) \leq H^\infty(st) + H^\infty(|s|) + \mathcal{O}(1)$

Proof

1. Let $p, q \in \{0,1\}^*$ such that $U^\infty(p) = s$ and $U(q) = |s|$. Then there is a machine that first simulates $U(q)$ to obtain $|s|$, then starts a simulation of $U^\infty(p)$ writing its output on the output tape, until it has written $|s|$ symbols, and then halts.
2. There are at most $2^{n+1} - 1$ strings of length $\leq n$.
3. Let $p, q \in \{0,1\}^*$ such that $U^\infty(p) = s$ and $U(q) = t$. Then there is a machine that first simulates $U(q)$ until it halts and prints $U(q)$ on the output tape. Then, it starts a simulation of $U^\infty(p)$ writing its output on the output tape.
4. Let $p, q \in \{0,1\}^*$ such that $U^\infty(p) = st$ and $U(q) = |t|$. Then there is a machine that first simulates $U(q)$ until it halts to obtain $|t|$. Then it starts a simulation of $U^\infty(p)$ such that at each stage n of the simulation it writes the symbols needed to leave $U(p)[n]\upharpoonright(|U(p)[n]| - |t|)$ on the output tape.
5. Consider the following monotone machine:

   $t := 1$; $v := \lambda$; $w := \lambda$
   repeat
     if $U(v)[t]$ asks for reading then append to v the next bit in the input

if $U(w)[t]$ asks for reading then append to w the next bit in the input
extend the actual output to $U(w)[t]\upharpoonright(U(v)[t])$
$t := t + 1$

If p and q are shortest programs such that $U^\infty(p) = |s|$ and $U^\infty(q) = st$ respectively, then we can interleave p and q in a way such that at each stage t, $v \preceq p$ and $w \preceq q$ (notice that eventually $v = p$ and $w = q$). Thus, this machine will compute s and will never read more than $H^\infty(st)+H^\infty(|s|)$ bits.

□

H is recursively approximable from above, but H^∞ is not.

Proposition 3.2 *There is no effective decreasing approximation of H^∞.*

Proof Suppose there is a recursive function $h : \{0, 1\}^* \times \mathbb{N} \to \mathbb{N}$ such that for every string s, $\lim_{t\to\infty} h(s, t) = H^\infty(s)$ and for all $t \in \mathbb{N}$, $h(s, t) \geq h(s, t + 1)$. We write $h_t(s)$ for $h(s, t)$. Consider the monotone machine $\mathcal{M}$ with coding constant d given by the Recursion Theorem, which on input p does the following:

$t := 1$; print 0
repeat forever
  $n :=$ number of bits read by $U(p)[t]$
  for each string s not yet printed, $|s| \leq t$ and $h_t(s) \leq n + d$
    print s
  $t := t + 1$

Let p be a program such that $U^\infty(p) = k$ and $|p| = H^\infty(k)$. Notice that, as $t \to \infty$, the number of bits read by $U(p)[t]$ goes to $|p| = H^\infty(k)$. Let t_0 be such that for all $t \geq t_0$, $U(p)[t]$ reads no more from the input. Since there are only finitely many strings s such that $H^\infty(s) \leq H^\infty(k) + d$, there is a $t_1 \geq t_0$ such that for all $t \geq t_1$ and for all those strings s, $h_t(s) = H^\infty(s)$. Hence, every string s with $H^\infty(s) \leq H^\infty(k) + d$ will be printed.

Let $z = M^\infty(p)$. On one hand, we have $H^\infty(z) \leq |p| + d = H^\infty(k) + d$. On the other hand, by the construction of $\mathcal{M}$, z cannot be the output of a program of length $\leq H^\infty(k) + d$ (because z is different from each string s such that $H^\infty(s) \leq H^\infty(k) + d$). So it must be that $H^\infty(z) > H^\infty(k) + d$, a contradiction. □

The following lemma states a critical property that distinguishes H^∞ from H. It implies that H^∞ is not subadditive, i.e., it is not the case that $H^\infty(st) \leq H^\infty(s) + H^\infty(t) + \mathcal{O}(1)$. It also implies that H^∞ is not invariant under recursive permutations $\{0, 1\}^* \to \{0, 1\}^*$.

Lemma 3.3 *For every total recursive function f there is a natural k such that*

$$H^\infty(0^k1) > f(H^\infty(0^k)).$$

Proof Let f be any recursive function and $\mathcal{M}$ the following monotone machine with coding constant d given by the Recursion Theorem:

```
t := 1
do forever
   for each p such that |p| ≤ max{f(i) : 0 ≤ i ≤ d}
      if U(p)[t] = 0^j 1 then
         print enough 0's to leave at least 0^{j+1} on the output tape
   t := t + 1
```

Let $N = \max\{f(i) : 0 \leq i \leq d\}$. We claim there is a k such that $M^\infty(\lambda) = 0^k$. Since there are only finitely many programs of length less than or equal to N which output a string of the form 0^j1 for some j, then there is some stage at which $\mathcal{M}$ has written 0^k, with k greater than all such j's, and then it prints nothing else. Therefore, there is no program p with $|p| \leq N$ such that $U^\infty(p) = 0^k1$.

If $M^\infty(\lambda) = 0^k$ then $H^\infty(0^k) \leq d$. So, $f(H^\infty(0^k)) \leq N$. Also, for this k, there is no program of length $\leq N$ that outputs 0^k1 and thus $H^\infty(0^k1) > N$. Hence, $H^\infty(0^k1) > f(H^\infty(0^k))$. □

Note that $H(0^k) = H(0^k1) = H^\infty(0^k1)$ up to additive constants, so the above lemma gives an example where H^∞ is much smaller that H.

Proposition 3.4

1. *H^∞ is not subadditive*
2. *It is not the case that for every recursive one-one $g : \{0,1\}^* \to \{0,1\}^*$ $\exists c\ \forall s\ |H^\infty(g(s)) - H^\infty(s)| \leq c$*

Proof

1. Let f be the recursive injection $f(n) = n + c$. By Lemma 3.3 there is k such that $H^\infty(0^k1) > H^\infty(0^k) + c$. Since the last inequality holds for every c, it is not true that $H^\infty(0^k1) \leq H^\infty(0^k) + \mathcal{O}(1)$.

2. It is immediate from Lemma 3.3.

□

It is known that the complexity H is smooth in the length and lexicographic order over $\{0,1\}^*$ in the sense that $|H(\mathit{string}(n)) - H(\mathit{string}(n+1))| = \mathcal{O}(1)$. However, this is not the case for H^∞.

Proposition 3.5

1. *H^∞ is not smooth in the length and lexicographical order over $\{0,1\}^*$*
2. $\forall n\ |H^\infty(\mathit{string}(n)) - H^\infty(\mathit{string}(n+1))| \leq H(|\mathit{string}(n)|) + \mathcal{O}(1)$

Proof

1. Notice that $\forall n > 1$, $H^\infty(0^n1) \leq H^\infty(0^{n-1}1) + \mathcal{O}(1)$, because if $U^\infty(p) = 0^{n-1}1$ then there is a machine that first writes a 0 on the output tape and then simulates $U^\infty(p)$. By Lemma 3.3, for each c there is a n such that $H^\infty(0^n1) > H^\infty(0^n) + c$. Joining the two inequalities, we obtain $\forall c\ \exists n\ H^\infty(0^{n-1}1) > H^\infty(0^n) + c$. Since $\mathit{string}^{-1}(0^{n-1}1) = \mathit{string}^{-1}(0^n) + 1$, H^∞ is not smooth.

2. Consider the following monotone machine $\mathcal{M}$ with input pq:

   obtain $y = U(p)$
   simulate $z = U^\infty(q)$ till it outputs y bits
   write $string(string^{-1}(z) + 1)$

   Let $p, q \in \{0,1\}^*$ such that $U(p) = |string(n)|$ and $U^\infty(q) = string(n)$. Then, $M^\infty(pq) = string(n+1)$ and
   $$H^\infty(string(n+1)) \leq H^\infty(string(n)) + H(|string(n)|) + \mathcal{O}(1).$$
   Similarly, if $\mathcal{M}$, instead of writing $string(string^{-1}(z) + 1)$, writes $string(string^{-1}(z) - 1)$, we conclude
   $$H^\infty(string(n)) \leq H^\infty(string(n+1)) + H(|string(n+1)|) + \mathcal{O}(1).$$
   Since $|H(|string(n)|) - H(|string(n+1)|)| = \mathcal{O}(1)$, it follows that
   $$|H^\infty(string(n)) - H^\infty(string(n+1))| \leq H(|string(n)|) + \mathcal{O}(1).$$

□

4 H^∞ is different from H^A for every oracle A

Item 3 of Proposition 2.7 states that H^∞ is between H and $H^{\varnothing'}$. The following result shows that H^∞ is really strictly in between them.

Proposition 4.1 *For every c there is a string $s \in \{0,1\}^*$ such that*
$$H^{\varnothing'}(s) + c < H^\infty(s) < H(s) - c.$$

Proof Let $u_n = \min\{s \in \{0,1\}^n : H(s) \geq n\}$ and let $A = \{a_0, a_1, \ldots\}$ be any infinite r.e. set and consider a machine $\mathcal{M}$ which on input i does the following:

$j := 0$
repeat
  write a_j
  find a program p, $|p| \leq 3i$, such that $U(p) = a_j$
  $j := j + 1$

$M^\infty(i)$ outputs the string $v_i = a_0 a_1 \ldots a_{k_i}$, where $H(a_{k_i}) > 3i$ and for all z, $0 \leq z < k_i$ we have $H(a_z) \leq 3i$. We define $w_i = u_i v_i$. Let's see that both $H^\infty(w_i) - H^{\varnothing'}(w_i)$ and $H(w_i) - H^\infty(w_i)$ grow arbitrarily.

On one hand, we can construct a machine which on input i and p executes $U^\infty(p)$ till it outputs i bits and then halts. Since the first i bits of w_i are u_i and $H(i) \leq 2|i| + \mathcal{O}(1)$, we have $i \leq H(u_i) \leq H^\infty(w_i) + 2|i| + \mathcal{O}(1)$. But with the help of the $\varnothing'$-oracle we can compute w_i from i, so $H^{\varnothing'}(w_i) \leq 2|i| + \mathcal{O}(1)$. Thus we have $H^\infty(w_i) - H^{\varnothing'}(w_i) \geq i - 4|i| - \mathcal{O}(1)$.

On the other hand, given i and w_i, we can effectively compute a_{k_i}. Hence, $\forall i$ we have $3i < H(a_{k_i}) \leq H(w_i) + 2|i| + \mathcal{O}(1)$. Also, given u_i, we can compute w_i in the limit using the idea of machine $\mathcal{M}$, and hence $H^\infty(w_i) \leq 2|u_i| + \mathcal{O}(1) = 2i + \mathcal{O}(1)$. Then, for all i
$$H(w_i) - H^\infty(w_i) > i - 2|i| - \mathcal{O}(1).$$

□

Not only H^∞ is different from $H^{\varnothing'}$ but it differs from H^A (the prefix-free complexity of a universal monotone machine with oracle A), for every A.

Theorem 4.2 *There is no oracle A such that $\left|H^\infty - H^A\right| \leq \mathcal{O}(1)$.*

Proof Immediate from Lemma 3.3 and from the standard result that for all A, H^A is subadditive, so in particular, for every k, $H^A(0^k1) \leq H^A(0^k)+\mathcal{O}(1)$. □

5 H^∞ and the Cantor space

The advantage of H^∞ over H can be seen along the initial segments of every recursive sequence: if $A \in \{0,1\}^\omega$ is recursive then there are infinitely many n's such that $H(A\restriction n) - H^\infty(A\restriction n) > c$, for an arbitrary c.

Proposition 5.1 *Let $A \in \{0,1\}^\omega$ be a recursive sequence. Then*

1. $\limsup_{n\to\infty} H(A\restriction n) - H^\infty(A\restriction n) = \infty$
2. $\limsup_{n\to\infty} H^\infty(A\restriction n) - Hm(A\restriction n) = \infty$

Proof

1. Let $A(n)$ be the n-th bit of A. Let's consider the following monotone machine $\mathcal{M}$ with input p:

   obtain $n := U(p)$
   write $A\restriction(string^{-1}(0^n) - 1)$
   for $s := 0^n$ to 1^n in lexicographic order
       write $A(string^{-1}(s))$
       search for a program p such that $|p| < n$ and $U(p) = s$

   If $U(p) = n$, then $M^\infty(p)$ outputs $A\restriction k_n$ for some k_n such that $2^n \leq k_n < 2^{n+1}$, since for all n there is a string of length n with H-complexity greater than or equal to n. Let us fix n. On one hand, $H^\infty(A\restriction k_n) \leq H(n) + \mathcal{O}(1)$. On the other, $H(A\restriction k_n) \geq n + \mathcal{O}(1)$, because we can compute the first string in the lexicographic order with H-complexity $\geq n$ from a program for $A\restriction k_n$. Hence, for each n, $H(A\restriction k_n) - H^\infty(A\restriction k_n) \geq n - H(n) + \mathcal{O}(1)$.
2. Trivial because for each recursive sequence A there is a constant c such that $Hm(A\restriction n) \leq c$ and $\lim_{n\to\infty} H^\infty(B\restriction n) = \infty$ for every $B \in \{0,1\}^\omega$.

□

5.1 H-triviality and H^∞-triviality There is a standard convention to use H with arguments in $\mathbb{N}$. I.e., for any $n \in \mathbb{N}$, $H(n)$ is written instead of $H(f(n))$ where f is some particular representation of natural numbers on $\{0,1\}^*$. This convention makes sense because H is invariant (up to a constant) for any recursive representation of natural numbers.

H-triviality has been defined as follows (see [5]): $A \in \{0,1\}^\omega$ is *H-trivial* iff there is a constant c such that for all n, $H(A\restriction n) \leq H(n) + c$. The idea is that H-trivial sequences are exactly those whose initial segments have minimal H-complexity. Considering the above convention, A is H-trivial iff $\exists c\ \forall n\ H(A\restriction n) \leq H(0^n) + c$.

In general H^∞ is not invariant for recursive representations of $\mathbb{N}$. We propose the following definition that insures that recursive sequences are H^∞-trivial.

Definition 5.2 $A \in \{0,1\}^\omega$ is *H^∞-trivial* iff $\exists c\ \forall n\ H^\infty(A\restriction n) \le H^\infty(0^n) + c$.

Our choice of the right hand side of the above definition is supported by the following proposition.

Proposition 5.3 *Let $f : \mathbb{N} \to \{0,1\}^*$ be recursive and strictly increasing with respect to the length and lexicographical order over $\{0,1\}^*$. Then*

$$\forall n\ H^\infty(0^n) \le H^\infty(f(n)) + \mathcal{O}(1).$$

Proof Notice that, since f is strictly increasing, f has recursive range. We construct a monotone machine $\mathcal{M}$ with input p:

```
t := 0
repeat
    if U(p)[t]↓ is in the range of f then n := f^{-1}(U(p)[t])
    print the needed 0's to leave 0^n on the output tape
    t := t + 1
```

Since f is increasing in the length and lexicographic order over $\{0,1\}^*$, if p is a program for $\mathcal{U}$ such that $U^\infty(p) = f(n)$, then $M^\infty(p) = 0^n$. □

Chaitin observed that every recursive $A \in \{0,1\}^\omega$ is H-trivial [4] and that H-trivial sequences are Δ^0_2. However, H-triviality does not characterize the class Δ^0_1 of recursive sequences: Solovay [14] constructed a Δ^0_2 sequence which is H-trivial but not recursive (see also [5] for the construction of a strongly computably enumerable real with the same properties). Our next result implies that H^∞-trivial sequences are Δ^0_2, and Theorem 5.6 characterizes Δ^0_1 as the class of H^∞-trivial sequences.

Theorem 5.4 *Suppose that A is a sequence such that, for some $b \in \mathbb{N}$, $\forall n\ H^\infty(A\restriction n) \le H(n) + b$. Then A is H-trivial.*

Proof An r.e. set $W \subseteq \mathbb{N} \times 2^{<\omega}$ is a *Kraft-Chaitin set* (KC-set) if

$$\textstyle\sum_{\langle r,y\rangle \in W} 2^{-r} \le 1.$$

For any $E \subseteq W$, let the *weight* of E be $wt(E) = \sum\{2^{-r} :\ \langle r,n\rangle \in E\}$. The pairs enumerated into such a set W are called *axioms*. Chaitin proved that from a Kraft-Chaitin set W one may obtain a prefix machine M_d such that $\forall \langle r,y\rangle \in W\ \exists w\ (|w| = r\ \wedge\ M_d(w) = y)$.

The idea is to define a Δ^0_2 tree T such that $A \in [T]$, and a KC-set W showing that each path of T is H-trivial. For $x \in \{0,1\}^*$ and $t \in \mathbb{N}$, let

$$\begin{aligned} H^\infty(x)[t] &= \min\{|p| : U(p)[t] = x\} \text{ and} \\ H(x)[t] &= \min\{|p| : U(p)[t] = x \text{ and } U(p) \text{ halts in at most } t \text{ steps}\} \end{aligned}$$

be effective approximations of H^∞ and H. Notice that for all $x \in \{0,1\}^*$, $\lim_{t\to\infty} H^\infty(x)[t] = H^\infty(x)$ and $\lim_{t\to\infty} H(x)[t] = H(x)$.

Given s, let

$$T_s = \{\gamma : |\gamma| < s \ \wedge \ \forall m \leq |\gamma| \ H^\infty(\gamma\restriction m)[s] \leq H(m)[s] + b\}$$

then $(T_s)_{s\in\mathbb{N}}$ is an effective approximation of a Δ^0_2 tree T, and $[T]$ is the class of sequences A satisfying $\forall n \ H^\infty(A\restriction n) \leq H(n) + b$. Let $r = H(|\gamma|)[s]$. We define a KC-set W as follows: if $\gamma \in T_s$ and either there is $u < s$ greatest such that $\gamma \in T_u$ and $r < H(|\gamma|)[u]$, or $\gamma \notin T_u$ for all $u < s$, then put an axiom $\langle r + b + 1, \gamma\rangle$ into W.

Once we show that W is indeed a KC-set, we are done: by Chaitin's result, there is d such that $\langle k, \gamma\rangle \in W$ implies $H(\gamma) \leq k + d$. Thus, if $A \in [T]$, then $H(\gamma) \leq H(|\gamma|) + b + d + 1$ for each initial segment γ of A.

To show that W is a KC-set, define strings $D_s(\gamma)$ as follows. When we put an axiom $\langle r + b + 1, \gamma\rangle$ into W at stage s,

- let $D_s(\gamma)$ be a shortest p such that $U(p)[s] = \gamma$ (recall from Definition 2.1 that it is not required that U halts at stage s)
- if $\beta \prec \gamma$, we haven't defined $D_s(\beta)$ yet and $D_{s-1}(\beta)$ is defined as a prefix of p, then let $D_s(\beta)$ be a shortest q such that $U(q)[s] = \beta$

In all other cases, if $D_{s-1}(\beta)$ is defined then we let $D_s(\beta) = D_{s-1}(\beta)$. We claim that, *for each s, all the strings $D_s(\beta)$ are pairwise incompatible* (i.e., they form a prefix-free set). For suppose that $p \prec q$, where $p = D_s(\beta)$ was defined at stage $u \leq s$, and $q = D_s(\gamma)$ was defined at stage $t \leq s$. Thus, $\beta = U(p)[u]$ and $\gamma = U(q)[t]$. By the definition of monotone machines and the minimality of q, $u < t$ and $\beta \prec \gamma$. But then, at stage t we would redefine $D_u(\beta)$, a contradiction. This shows the claim.

If we put an axiom $\langle r+b+1, \gamma\rangle$ into W at stage t, then for all $s \geq t$, $D_s(\gamma)$ is defined and has length at most $H(|\gamma|)[t] + b$ (by the definition of the trees T_s). Thus, if $\widetilde{W_s}$ is the set of axioms $\langle k, \gamma\rangle$ in W_s where k is minimal for γ, then $wt(\widetilde{W_s}) \leq \sum_\gamma 2^{-|D_s(\gamma)|-1} \leq 1/2$ by the claim above. Hence $wt(W_s) \leq 1$ as all axioms weigh at most twice as much as the minimal ones, and W_s is a KC-set for each s. Hence W is a KC-set. □

Corollary 5.5 *If $A \in \{0,1\}^\omega$ is H^∞-trivial then A is H-trivial, hence in Δ^0_2.*

Theorem 5.6 *Let $A \in \{0,1\}^\omega$. A is H^∞-trivial iff A is recursive.*

Proof From right to left, it is easy to see that if A is a recursive sequence then A is H^∞-trivial.

For the converse, let A be H^∞-trivial via some constant b. By Corollary 5.5 A is Δ^0_2, hence, there is a recursive approximation $(A_s)_{s\in\mathbb{N}}$ such that $\lim_{s\to\infty} A_s = A$.

Recall that $H^\infty(x)[t] = \min\{|p| : U(p)[t] = x\}$. Consider the following program with coding constant c given by the Recursion Theorem:

```
k := 1; s_0 := 0; print 0
while ∃s_k > s_{k-1} such that H^∞(A_{s_k}↾k)[s_k] ≤ c + b do
     print 0
     k := k + 1
```

Let us see that the above program prints out infinitely many 0's. Suppose it writes 0^k for some k. Then, on one hand, $H^\infty(0^k) \leq c$, and on the other,

$\forall s > s_{k-1}$, we have $H^\infty(A_s\restriction k)[s] > c+b$. Also, $H^\infty(A_s\restriction k)[s] = H^\infty(A\restriction k)$ for s large enough. Hence, $H^\infty(A\restriction k) > H^\infty(0^k) + b$, which contradicts that A is H^∞-trivial via b.

So, for each k, there is some $q \in \{0,1\}^*$ with $|q| \le c+b$ such that $U(q)[s_k] = A_{s_k}\restriction k$. Since there are only $2^{c+b+1} - 1$ strings of length at most $c+b$, there must be at least one q such that, for *infinitely many* k, $U(q)[s_k] = A_{s_k}\restriction k$. Let's call I the set of all these k's. We will show that such a q necessarily computes A. Suppose not. Then, there is a t such that for all $s \ge t$, $U(q)[s]$ is not an initial segment of A. Thus, noticing that $(s_k)_{k\in\mathbb{N}}$ is increasing and I is infinite, there are infinitely many $s_k \ge t$ such that $k \in I$ and $U(q)[s_k] = A_{s_k}\restriction k \ne A\restriction k$. This contradicts that $A_{s_k}\restriction k \to A$ when $k \to \infty$. □

Corollary 5.7 *The class of H^∞-trivial sequences is strictly included in the class of H-trivial sequences.*

Proof By Corollary 5.5, any H^∞-trivial sequence is also H-trivial. Solovay [14] built an H-trivial sequence in Δ^0_2 which is not recursive. By Theorem 5.6 this sequence cannot be H^∞-trivial. □

5.2 H^∞-randomness

Definition 5.8

1. (Chaitin [2]) $A \in \{0,1\}^\omega$ is *H-random* iff $\exists c\ \forall n\ H(A\restriction n) > n - c$. Chaitin and Schnorr [2] showed that H-randomness coincides with Martin-Löf randomness [12].
2. (Levin [9]) $A \in \{0,1\}^\omega$ is *Hm-random* iff $\exists c\ \forall n\ Hm(A\restriction n) > n - c$.
3. $A \in \{0,1\}^\omega$ is *H^∞-random* iff $\exists c\ \forall n\ H^\infty(A\restriction n) > n - c$.

Using Levin's result [9] that *Hm*-randomness coincides with Martin-Löf randomness, and the fact that *Hm* gives a lower bound of H^∞, it follows immediately that the classes of H-random, H^∞-random and *Hm*-random sequences coincide. For the sake of completeness we give an alternative proof.

Proposition 5.9 (with D. Hirschfeldt) *There is a b_0 such that for all $b \ge b_0$ and z, if $Hm(z) \le |z| - b$, then there is $y \preceq z$ such that $H(y) \le |y| - b/2$.*

Proof Consider the following machine $\mathcal{M}$ with coding constant c. On input qp, first it simulates $U(q)$ until it halts. Let's call b the output of this simulation. Then it simulates $U^\infty(p)$ till it outputs a string y of length $b + l$ where l is the length of the prefix of p read by U^∞. Then it writes this string y on the output and stop.

Let b_0 be the first number such that $2\,|b_0| + c \le b_0/2$ and take $b \ge b_0$. Suppose $Hm(z) \le |z| - b$. Let p be a shortest program such that $U^\infty(p) \succeq z$ and let q be a shortest program such that $U(q) = b$. This means that $|p| = Hm(z)$ and $|q| = H(b)$. On input qp, the machine $\mathcal{M}$ will compute b and then it will start simulating $U^\infty(p)$. Since $|z| \ge Hm(z) + b = |p| + b$, the machine will eventually read l bits from p in a way that the simulation of $U^\infty(p\restriction l) = y$ and $|y| = l + b$. When this happens, the machine $\mathcal{M}$ writes y and stops. Then for $p' = p\restriction l$, we have $M(qp')\downarrow = y$ and $|y| = |p'| + b$. Hence

$$H(y) \le |q| + |p'| + c \le H(b) + |y| - b + c \le 2\,|b| - b + |y| + c \le |y| - b/2.$$

□

Corollary 5.10 *$A \in \{0,1\}^\omega$ is Martin-Löf random iff A is Hm-random iff A is H^∞-random.*

Proof Since $Hm \leq H + \mathcal{O}(1)$ it is clear that if a sequence is *Hm*-random then it is Martin-Löf random. For the opposite, suppose A is Martin-Löf random but not *Hm*-random. Let b_0 be as in Proposition 5.9 and let $2c \geq b_0$ be such that $\forall n\ H(A\restriction n) > n - c$. Since A is not *Hm*-random, $\forall d\ \exists n\ Hm(A\restriction n) \leq n - d$. In particular for $d = 2c$ there is an n such that $Hm(A\restriction n) \leq n - 2c$. On one hand, by Proposition 5.9, there is a $y \preceq A\restriction n$ such that $H(y) \leq |y| - c$. On the other, since y is a prefix of A and A is Martin-Löf random, we have $H(y) > |y| - c$. This is a contradiction.

Since *Hm* is a lower bound of H^∞, the above equivalence implies A is Martin-Löf random iff A is H^∞-random. □

References

[1] V. Becher, S. Daicz and G. Chaitin. A highly random number. In C. S. Calude, M. J. Dineen, and S. Sburlan, editors, *Combinatorics, Computability and Logic: Proceedings of the Third Discrete Mathematics and Theoretical Computer Science Conference (DMTCS'01)*, 55–68. Springer-Verlag London, 2001.

[2] G. J. Chaitin. A theory of program size formally identical to information theory, *J. ACM*, vol.22, 329–340, 1975.

[3] G. J. Chaitin. Algorithmic entropy of sets, *Computers & Mathematics with Applications*, vol.2, 233–245, 1976.

[4] G. J. Chaitin. Information-theoretical characterizations of recursive infinite strings. *Theoretical Computer Science*, 2:45–48,1976.

[5] R. Downey, D. Hirschfeldt, A. Nies and F. Stephan. Trivial reals, *Electronic Notes in Theoretical Computer Science (ENTCS)*, vol. 66:1, 2002. Also in *Proceedings of the 7th and 8th Asian Logic Conferences*, 103–131. World Scientific, Singapore, 2003. Eds. R. Downey, D. Decheng, T. Shih Ping, Q. Yu Hui, M. Yasugi.

[6] M. Ferbus-Zanda and S. Grigorieff. Church, cardinal and ordinal representations of integers and Kolmogorov complexity. Manuscript, 2003.

[7] P. Gács. On the symmetry of algorithmic information. *Soviet Math. Dodkl.*, 15, pp. 1477–1480, 1974.

[8] A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russ. Math. Surveys*, Vol. 25, pp. 83–124, 1970.

[9] L. A. Levin. On the Concept of a Random Sequence. *Doklady Akad. Nauk SSSR*, 14(5), 1413–1416, 1973.

[10] L. A. Levin. Laws of Information Conservation (Non-growth) and Aspects of the Foundations of Probability Theory. *Problems of Information Transmission*, 10:3, pp. 206–210, 1974.

[11] M. Li and P. Vitanyi. *An introduction to Kolmogorov complexity and its applications*, Springer, Amsterdam, 1997 (2d edition).

[12] P. Martin-Löf. The definition of random sequences. *Information and Control*, vol.9, 602–619, 1966.

[13] C. P. Schnorr. Process complexity and effective random tests. *Journal of Computer Systems Science*, Vol. 7, pp. 376–388, 1973.

[14] R. M. Solovay. *Draft of a paper (or series of papers) on Chaitin's work done for the most part during the period Sept. to Dec. 1974*, unpublished manuscript, IBM Thomas J. Watson Research Center, Yorktown Heights, New York. 215 pp., May 1975.

[15] V. A. Uspensky and A. Kh. Shen'. Relations between varieties of Kolmogorov complexities. *Math. Systems Theory*, Vol. 29, pp. 271–292, 1996.

Depto. Computación, Facultad Cs. Exactas y Naturales, Universidad de Buenos Aires
Argentina
Supported by Agencia Nacional de Promoción Científica y Tecnológica
vbecher@dc.uba.ar

Depto. Computación, Facultad Cs. Exactas y Naturales, Universidad de Buenos Aires
Argentina
Supported by a grant of Fundación Antorchas
sfigueir@dc.uba.ar

Department of Computer Science, University of Auckland
New Zealand
Supported by the Marsden grant of New Zealand
andre@cs.auckland.ac.nz

Depto. Computación, Facultad Cs. Exactas y Naturales, Universidad de Buenos Aires
Argentina
spicchi@dc.uba.ar