Journal of Universal Computer Science, vol. 3, no. 11 (1997), 1226-1240
submitted: 8/8/97, accepted: 23/10/97, appeared: 28/11/97 © Springer Pub. Co.

Surjective Functions on Computably Growing Cantor Sets[1]

Peter Hertling
(Department of Computer Science, The University of Auckland,
Private Bag 92019, Auckland, New Zealand
Email: hertling@cs.auckland.ac.nz)

Abstract: Every infinite binary sequence is Turing reducible to a random one. This is a corollary of a result of Péter Gács stating that for every co-r.e. closed set with positive measure of infinite sequences there exists a computable mapping which maps a subset of the set onto the whole space of infinite sequences. Cristian Calude asked whether in this result one can replace the positive measure condition by a weaker condition not involving the measure. We show that this is indeed possible: it is sufficient to demand that the co-r.e. closed set contains a *computably growing Cantor set*. Furthermore, in the case of a set with positive measure we construct a surjective computable map which is more effective than the map constructed by Gács.

Key Words: Computable maps on infinite sequences, co-r.e. closed sets, Cantor sets, computability and measure

Category: F.1

1 Introduction and Notation

Every infinite binary random sequence is Turing reducible to random one. This was proved independently by Gács [Gács 1986] and Kučera [Kučera 1985] by different methods. Gács proved the following stronger result: for any set of infinite binary sequences which is co-r.e. closed (Π_1^0) and has positive measure there exists a computable mapping which maps a subset of the set onto the whole space of infinite binary sequences. Măndoiu [Măndoiu 1993] extended this result to arbitrary alphabets instead of the binary alphabet. Since there are co-r.e. closed sets with measure zero which can nonetheless be mapped computably onto the whole space of infinite sequences, Măndoiu [Măndoiu 1993] and Calude [Calude 1994], Problem 10, asked whether it is possible to replace the positive measure condition in Gács's result by a more general condition not involving the measure. We show that this is indeed possible: for any co-r.e. closed set which contains a *computably growing Cantor set* there is a computable mapping which maps a a subset of the set onto the space of all sequences. Computably growing Cantor sets will be defined in the following section.

In the case of binary sequences and a co-r.e. closed set of positive measure, Gács constructed a surjective map which is effective in the following sense: after reading n digits of a sequence on which the map is defined the algorithm for the map writes at least $n - 3 \cdot \sqrt{n} \cdot \log_2 n - c$ digits of the output sequence, where c is a constant. We improve this by constructing for any $\varepsilon > 0$ an algorithm which

[1] Proceedings of the *First Japan-New Zealand Workshop on Logic in Computer Science*, special issue editors D.S. Bridges, C.S. Calude, M.J. Dinneen and B. Khoussainov.

under the same conditions writes at least $n-(2+\varepsilon)\cdot\sqrt{n\cdot\log_2 n}-c$ digits, c a constant. For an input alphabet with p symbols and an output alphabet with q symbols we obtain for each $\varepsilon>0$ an algorithm which after reading n digits of a sequence on which the map is defined writes at least $\log_q p\cdot n-(2+\varepsilon)\log_q p\cdot\sqrt{n\log_p n}-c$ digits of the output sequence.

In the following section we introduce the necessary notions, state the main result, and show that it covers the known examples [Măndoiu 1993, Calude 1994] of co-r.e. closed sets with measure zero. In Section 3 we prove the main theorem. In the last section we show that our main theorem covers also the result by Gács and its strengthening by Măndoiu. Furthermore we construct the algorithm mentioned above which is more effective than the algorithm of Gács. This is done for arbitrary input and output alphabets. We close this section by introducing some notation.

$\mathbb{N}$ is the set of natural numbers, $\mathbb{R}$ is the set of real numbers. An *alphabet* is a finite set with at least two elements. Alphabets are denoted by Σ and Γ. The following notation is introduced for objects with respect to some fixed alphabet Σ, but applies as well to any other alphabet. Σ^* is the set of strings with digits from Σ, Σ^n is the set of strings of length n, $\Sigma^\omega=\{\alpha\mid\alpha:\mathbb{N}\to\Sigma\}$ is the set of infinite sequences. The length of a string x is denoted by $|x|$. By λ we denote the empty string. For a string x and a string or infinite sequence $y\in\Sigma^*\cup\Sigma^\omega$, xy is the concatenation of x and y. For a string or an infinite sequence $x\in\Sigma^*\cup\Sigma^\omega$ and an integer number $n\geq-1$, $x[0..n]$ denotes the initial segment of length $n+1$ of x (where $x[0..n]=x$ if $|x|\leq n+1$). A string $x\in\Sigma^*$ is a prefix of a string or an infinite sequence $y\in\Sigma^*\cup\Sigma^\omega$ if $y[0..|x|-1]=x$. This is denoted by $x\sqsubseteq y$. For $x\in\Sigma^*$ the set $[x]:=\{y\in\Sigma^*\cup\Sigma^\omega\mid x\sqsubseteq y\}$ is the set of all strings and infinite sequences y such that x is a prefix of y. We extend this to sets $A\subseteq\Sigma^*$ by $[A]:=\bigcup_{x\in A}[x]$. For $n\in\mathbb{N}$ and a set $A\subseteq\Sigma^*\cup\Sigma^\omega$ we define $A^{[n]}:=\{x\in\Sigma^n\mid A\cap[x]\neq\emptyset\}$.

The lower case letters k,l,m,n,p,q,r,s,t denote numbers, letters v,w,x,y,z from the end of the alphabet denote finite strings, and greek letters α,β,γ denote infinite sequences, respectively. A subset of Σ^* is called a language or simply a set. The capital letters A,B,C are used to denote subsets of Σ^* and boldface capital letters $\mathbf{A},\mathbf{B},\mathbf{C}$ are used to denote subsets of Σ^ω. Functions on strings or natural numbers are denoted by the letters f,g,h,F,G and once also by l. For sets X and Y, $X\subseteq Y$ denotes that X is a subset of Y and $f:\subseteq X\to Y$ denotes a possibly partial function with values in Y whose domain $\mathrm{dom}\,f$ is a subset of X. If $\mathrm{dom}\,f=X$ we also write $f:X\to Y$. By μ we denote the usual product measure on Σ^ω given by $\mu([x]\cap\Sigma^\omega)=|\Sigma|^{-|x|}$ for all $x\in\Sigma^*$.

2 The Main Result

Let Σ and Γ be two fixed alphabets. Let $f:\subseteq\Sigma^*\to\Gamma^*$ be a *monotonic* function, that is, a function with $x\sqsubseteq y\Rightarrow f(x)\sqsubseteq f(y)$ for all $x,y\in\mathrm{dom}\,f$. A string $x\in\mathrm{dom}\,f$ is called *f-nonterminating* iff there is a $y\in[x]\cap\mathrm{dom}\,f$ with $f(y)\neq f(x)$. The function $\overline{f}:\subseteq\Sigma^\omega\to\Gamma^\omega$ *induced by* f is defined by

1. $\mathrm{dom}\,\overline{f}:=\bigcap_n[f^{-1}([\Gamma^n])]=\{\alpha\in\Sigma^\omega\mid$ infinitely many prefixes $v\in\Sigma^*$ of α are f-nonterminating$\}$ and by
2. $\overline{f}(\alpha)\in[f(x)]$ for all $\alpha\in\mathrm{dom}\,f$ and all prefixes $v\in\mathrm{dom}\,f$ with $\alpha\in[v]$.

The (monotonic) function $f :\subseteq \Sigma^* \to \Gamma^*$ is called *approximable* iff the set $\{(x,y) \in \Sigma^* \times \Gamma^* \mid x \in \mathrm{dom}\, f \text{ and } y \sqsubseteq f(x)\}$ is recursively enumerable. A function $f :\subseteq \Sigma^* \to \Gamma^*$ is called a *process* iff it is total, monotonic, approximable, and the set $\{x \in \mathrm{dom}\, f \mid x \text{ is } f\text{-nonterminating}\}$ is recursively enumerable. A process can be realized by a Turing machine with a one-way read-only input tape, a one-way write-only output tape and one or more work tapes which, given an infinite input string, for any input tape head position writes as many digits as possible on the output tape and moves the input tape head only when the string read so far turns out to be nonterminal.

Lemma 1. *For a function $F :\subseteq \Sigma^\omega \to \Gamma^\omega$ the following conditions are equivalent:*

1. *There is a total, monotonic, computable $f : \Sigma^* \to \Gamma^*$ with $F = \overline{f}$.*
2. *There is a monotonic, computable $f :\subseteq \Sigma^* \to \Gamma^*$ with $F = \overline{f}$.*
3. *There is a process $f : \Sigma^* \to \Gamma^*$ with $F = \overline{f}$.*
4. *There is a monotonic, approximable $f :\subseteq \Sigma^* \to \Gamma^*$ with $F = \overline{f}$.*

Proof. Every total, monotonic, computable function $f : \Sigma^* \to \Gamma^*$ is a process. All processes and all monotonic computable functions are approximable. Therefore it is sufficient to prove the implication "4. $\Rightarrow$ 1.". Let $f :\subseteq \Sigma^* \to \Gamma^*$ be a monotonic approximable function and $h : \mathbb{N} \to \Sigma^* \times \Gamma^*$ be a computable function with $h(\mathbb{N}) = \{(x,y) \in \Sigma^* \times \Gamma^* \mid x \in \mathrm{dom}\, f \text{ and } y \sqsubseteq f(x)\}$. Define a function $g : \Sigma^* \to \Gamma^*$ by

$$\begin{aligned} g(x) := {} & \text{the longest string in} \\ & \{\lambda\} \cup \{y \in \Sigma^* \mid \exists z \in \Sigma^*.\ z \sqsubseteq x \text{ and } (z,y) \in h(\{0,\ldots,|x|\}\}\,. \end{aligned}$$

The function g is total, monotonic, computable and satisfies $\overline{g} = \overline{f}$. $\square$

If a function $F :\subseteq \Sigma^\omega \to \Gamma^\omega$ satisfies one — and then all — of the conditions in Lemma 1, then it is itself called *computable*. A subset $\mathbf{C} \subseteq \Sigma^\omega$ is called *co-r.e. closed* (or Π^0_1) iff there is an r.e. set $B \subseteq \Sigma^*$ with $\mathbf{C} = \Sigma^\omega \setminus [B]$.

Definition 2. 1. Let $g : \mathbb{N} \to \mathbb{N}$ be an increasing function and $h : \mathbb{N} \to \mathbb{N}$ be a function with $h(n) \geq 2$ for all n. A set $\mathbf{A} \subseteq \Sigma^\omega$ is called a (g,h)-*Cantor set* iff it is nonempty and for each $n \in \mathbb{N}$ and each $x \in \mathbf{A}^{[g(n)]}$ we have $|[x] \cap \mathbf{A}^{[g(n+1)]}| \geq h(n+1)$.
2. A set $\mathbf{A} \subseteq \Sigma^\omega$ is called a *computably growing Cantor set* iff there is a computable increasing function $g : \mathbb{N} \to \mathbb{N}$ such that $\mathbf{A}$ is a $(g,2)$-Cantor set.

Our main technical result is the following

Theorem 3. *Let $g : \mathbb{N} \to \mathbb{N}$ and $h : \mathbb{N} \to \mathbb{N}$ be two increasing computable functions with $g(0) = h(0) = 0$. Let Σ and Γ be two alphabets, and $\mathbf{C} \subseteq \Sigma^\omega$ be a co-r.e. closed set which contains a $(g, n \mapsto |\Gamma|^{h(n+1)-h(n)})$-Cantor set. Then there is a process $f : \Sigma^* \to \Gamma^*$ with*

1. *$\overline{f}(\mathbf{C} \cap \mathrm{dom}\, \overline{f}) = \Gamma^\omega$,*
2. *for all $n \in \mathbb{N}$ and all f-nonterminal strings $x \in \Sigma^*$ with $|x| \geq g(n)$ we have $|f(x)| \geq h(n)$.*

This theorem will be proved in the following section.

Corollary 4. *Let Σ and Γ be two alphabets, and $\mathbf{C} \subseteq \Sigma^\omega$ be a co-r.e. closed set which contains a computably growing Cantor set. Then there is a computable function $F :\subseteq \Sigma^\omega \to \Gamma^\omega$ with $F(\mathbf{C} \cap \operatorname{dom} F) = \Gamma^\omega$.*

Proof. Assume that $\tilde{g} : \mathbb{N} \to \mathbb{N}$ is a computable increasing function and $\mathbf{A} \subseteq \mathbf{C}$ is a $(\tilde{g}, 2)$-Cantor set. Let $c \in \mathbb{N}$ be a number with $2^c \geq |\Gamma|$. We define two functions $g, h : \mathbb{N} \to \mathbb{N}$ by $g(0) := 0$, $g(n) := \tilde{g}(c \cdot n)$ for $n > 0$, and $h(n) := n$ for all n. The functions g and h are computable, increasing and satisfy $g(0) = h(0) = 0$. The set $\mathbf{A}$ is a $(g, 2^c)$-Cantor set, hence a $(g, n \mapsto |\Gamma|^{h(n+1)-h(n)})$-Cantor set. Thus, Theorem 3 gives the assertion. □

In Section 4 we shall show that any co-r.e. set of infinite sequences with positive measure contains a computably growing Cantor set. There we shall also be concerned with the effectivity of the surjective mapping as it is expressed by the second part of the assertion in Theorem 3. If for an increasing function $g : \mathbb{N} \to \mathbb{N}$ we define the function $\tilde{g} : \mathbb{N} \to \mathbb{N}$ by $\tilde{g}(n) := \max\{k \mid g(k) \leq n\}$, then the second part of the assertion in Theorem 3 can be formulated as

$$|f(x)| \geq h\tilde{g}(|x|)$$

for all f-nonterminating strings $x \in \Sigma^*$.

Măndoiu [Măndoiu 1993] and Calude [Calude 1994] pointed out that there are very simple co-r.e. closed sets of sequences with measure zero which can be mapped onto the whole set of sequences. We close this section by showing that these cases are also covered by Theorem 3. Assume that $\Sigma = \{a_1, \ldots, a_p\}$ contains at least $p \geq 3$ elements. The set $\mathbf{C} := \{a_1, a_2\}^\omega \subseteq \Sigma^\omega$ is co-r.e. closed (because of $\mathbf{C} = \Sigma^\omega \setminus [\Sigma^* \setminus \{a_1, a_2\}^*]$) and it has measure zero (with respect to the product measure on Σ^ω). But it is obviously an $(\mathrm{id}_{\mathbb{N}}, 2)$-Cantor set. Hence, Corollary 4 can be applied to it. Of course, a computable map $F :\subseteq \Sigma^\omega \to \Gamma^\omega$ with $F(\mathbf{C}) = \Gamma^\omega$ (where Γ is any alphabet) can also easily be constructed directly.

3 Proof of the Main Theorem

This section contains the proof of Theorem 3. Let $w_0, w_1, w_2, \ldots$ be a recursive sequence of strings in Σ^* with

$$\mathbf{C} = \Sigma^\omega \setminus \bigcup_k [w_k]\,.$$

For $t \in \mathbb{N}$ we define

$$\mathbf{C}_t := \Sigma^\omega \setminus \bigcup_{k<t} [w_k]\,.$$

We shall construct a computable sequence $(f_t)_{t\in\mathbb{N}}$ of computable functions $f_t :\subseteq \Sigma^* \to \Gamma^*$ with $\overline{f_t}(\mathbf{C}_t \cap \operatorname{dom} \overline{f_t}) = \Sigma^\omega$ and show that the function $f : \Sigma^* \to \Gamma^*$ defined by $f(x) :=$ the longest string in $\{f_t(y) \mid y \sqsubseteq x,\ t \in \mathbb{N}\}$ is well-defined and satisfies the conditions stated in the theorem. For each t, the function $\overline{f_t}$ will map a subset of $\mathbf{C}_t$ bijectively onto Γ^ω. With growing t cylinders $[x]$ may be taken out of $\mathbf{C}_t$, but for each length $|x|$ of strings x this can happen only finitely many times. Therefore the limit f exists and $\overline{f}$ maps a subset of $\mathbf{C}$ bijectively onto Γ^ω.

We use $p := |\Sigma|$ and $q := |\Gamma|$. Let $g : \mathbb{N} \to \mathbb{N}$ and $h : \mathbb{N} \to \mathbb{N}$ be two computable increasing functions with $g(0) = h(0) = 0$ and let $\mathbf{A} \subseteq \mathbf{C}$ be a $(g, n \mapsto q^{h(n+1)-h(n)})$-Cantor set. The existence of $\mathbf{A}$ implies

$$p^{g(n+1)-g(n)} \geq q^{h(n+1)-h(n)} \quad \text{for all } n. \tag{1}$$

Everything else that we need about $\mathbf{A}$ is captured by the sets D_t^n defined for $t, n \in \mathbb{N}$ by

$$D_t^n := \{x \in \Sigma^{g(n)} \mid [x] \cap \Sigma^\omega \subseteq \mathbf{C}_t \text{ or } \\ ([x] \cap \mathbf{C}_t \neq \emptyset \text{ and } |[x] \cap D_t^{n+1}| \geq q^{h(n+1)-h(n)})\}\,.$$

Lemma 5. *1. The sets D_t^n are well-defined for $t, n \in \mathbb{N}$ and the set $\{(t, n, x) \in \mathbb{N}^2 \times \Sigma^* \mid x \in D_t^n\}$ is recursive.*
2. $D_0^n = \Sigma^{g(n)}$ for all n.
3. $D_{t+1}^n \subseteq D_t^n$ for all $t, n \in \mathbb{N}$.
4. If $x \in D_t^n$, then $|[x] \cap D_t^{n+1}| \geq q^{h(n+1)-h(n)}$ for all $t, n \in \mathbb{N}$.
5. $\lambda \in D_t^0$ for all $t \in \mathbb{N}$.

Proof. 1. For $|x| \geq \max\{|w_i| \mid i < t\}$ we have either $[x] \cap \Sigma^\omega \subseteq \mathbf{C}_t$ or $[x] \cap \mathbf{C}_t = \emptyset$. Hence, the sets D_t^n are welldefined for $g(n) \geq \max\{|w_i| \mid i < t\}$. By the recursive construction of D_t^n, all sets D_t^n are well-defined. For the same reason and because the set $\{(t, x) \in \mathbb{N} \times \Sigma^* \mid [x] \cap \Sigma^\omega \subseteq \mathbf{C}_t\}$ is recursive, also the set $\{(t, n, x) \in \mathbb{N}^2 \times \Sigma^* \mid x \in D_t^n\}$ is recursive.

2. This follows immediately from $\mathbf{C}_0 = \Sigma^\omega$

3. This follows from $\mathbf{C}_{t+1} \subseteq \mathbf{C}_t$ for all t.

4. This follows from (1) and the definition of D_t^n.

5. We shall show that $\mathbf{A}^{[g(n)]} \subseteq D_t^n$ for all $t, n \in \mathbb{N}$. This gives the assertion because $\mathbf{A} \neq \emptyset$ and $g(0) = 0$ imply $\lambda \in \mathbf{A}^{[g(0)]}$. For the proof of $\mathbf{A}^{[g(n)]} \subseteq D_t^n$ fix a number t. We distinguish two cases.

First case: $g(n) \geq \max\{|w_i| \mid i < t\}$. Then $x \in \mathbf{A}^{[g(n)]}$ and $\mathbf{A} \subseteq \mathbf{C} \subseteq \mathbf{C}_t$ imply $[x] \cap \mathbf{C}_t \neq \emptyset$. The assumption (first case) gives $[x] \cap \Sigma^\omega \subseteq \mathbf{C}_t$. By the definition of D_t^n we conclude $x \in D_t^n$.

Second case: $g(n) < \max\{|w_i| \mid i < t\}$. Again $x \in \mathbf{A}^{[g(n)]}$ and $\mathbf{A} \subseteq \mathbf{C} \subseteq \mathbf{C}_t$ imply $[x] \cap \mathbf{C}_t \neq \emptyset$. Furthermore, $x \in \mathbf{A}^{[g(n)]}$ implies by the definition of $\mathbf{A}$ that there are at least $q^{h(n+1)-h(n)}$ strings y in $[x] \cap \mathbf{A}^{[g(n+1)]}$. By induction hypothesis (for $n+1$) all of these strings lie in D_t^{n+1}. By definition of D_t^n we conclude $x \in D_t^n$. □

Now we shall construct a computable function

$$F : \mathbb{N} \times \bigcup_m \Sigma^{g(m)} \to \bigcup_m \Sigma^{h(m)}$$

with the following properties where we use $f_t(x) := F(t, x)$ and

$$L_t^n := \{x \in \Sigma^{g(n)} \mid |f_t(x)| = h(n)\}$$

for all $t, n \in \mathbb{N}$ and $x, y \in \bigcup_m \Gamma^{g(m)}$:

(I) $f_t(x) \sqsubseteq f_{t+1}(x)$,
(II) $x \sqsubseteq y \Rightarrow f_t(x) \sqsubseteq f_t(y)$,
(III) $|x| = g(n) \Rightarrow |f_t(x)| \leq h(n)$,
(IV) $L_t^n = \{x \in \Sigma^{g(n)} \mid x \text{ is } f_t\text{-nonterminal}\}$,
(V) if $x \in D_t^n \cap L_t^n$, then f_t maps $[x] \cap D_t^{n+1} \cap L_t^{n+1}$ bijectively onto $[f_t(x)] \cap \Gamma^{h(n+1)}$.

It will be clear from the construction that F is computable. First we define $f_0(x)$ for all $x \in \bigcup_m \Sigma^{g(m)}$. We set

$$f_0(\lambda) := \lambda$$

and for $y \in \Sigma^{g(n+1)}$ we set $x := y[0..g(n)-1]$ and

$$f_0(y) := \begin{cases} \text{the } j\text{-th string in } [f_0(x)] \cap \Gamma^{h(n+1)} \\ \qquad \text{if } x \in L_0^n \text{ and } y \text{ is the } j\text{-th string in } [x] \cap \Sigma^{g(n+1)}, \\ \qquad \text{for some } j \text{ with } 1 \leq j \leq q^{h(n+1)-h(n)}, \\ f_0(x) \quad \text{otherwise.} \end{cases}$$

The function f_0 is well-defined by (1). Now we fix a number $t \in \mathbb{N}$ and construct f_{t+1}. We set

$$f_{t+1}(\lambda) := \lambda\,.$$

For $y \in \Sigma^{g(n+1)}$ we set $x := y[0..g(n)-1]$ and distinguish three cases. We can assume that f_t has been constructed and also that $f_{t+1}(x)$ is defined already. We may use the induction hypotheses (I) to (V).

1st case: $y \in L_t^{n+1}$. Then we set

$$f_{t+1}(y) := f_t(y)\,.$$

2nd case: $y \notin L_t^{n+1}$ and $x \notin D_{t+1}^n \cap L_{t+1}^n$. Then we set

$$f_{t+1}(y) := f_{t+1}(x)\,.$$

3rd case: $y \notin L_t^{n+1}$ and $x \in D_{t+1}^n \cap L_{t+1}^n$. Before we can define $f_{t+1}(y)$ we need to statements. Let $k := |[x] \cap D_{t+1}^{n+1} \cap L_t^{n+1}|$ and $l := q^{h(n+1)-h(n)} - k$. We claim that

$$|([x] \cap D_{t+1}^{n+1}) \setminus ([x] \cap D_{t+1}^{n+1} \cap L_t^{n+1})| \geq l \tag{2}$$

and

$$|([f_{t+1}(x)] \cap \Gamma^{h(n+1)}) \setminus f_t([x] \cap D_{t+1}^{n+1} \cap L_t^{n+1})| = l\,. \tag{3}$$

Our assumption $x \in D_{t+1}^n$ and Lemma 5.4 imply that there are at least $q^{h(n+1)-h(n)}$ elements in $[x] \cap D_{t+1}^{n+1}$. This proves (2). Claim (3) follows for $k = 0$ immediately from $|f_{t+1}(x)| = h(n)$. Assume $k \neq 0$. This (to be more precise: $[x] \cap L_t^{n+1} \neq \emptyset$) and $|f_t(x)| \leq h(n)$ (induction hypothesis (III)) imply that x is f_t-nonterminal. By induction hypothesis (IV) we obtain $|f_t(x)| = h(n)$. Using $f_t(x) \sqsubseteq f_{t+1}(x)$ (induction hypothesis (I)) and the assumption $x \in L_{t+1}^n$ we conclude $f_{t+1}(x) = f_t(x)$. Hence, using induction hypothesis (II) we obtain

$$f_t([x] \cap D_{t+1}^{n+1} \cap L_t^{n+1}) \subseteq [f_t(x)] \cap \Gamma^{h(n+1)} = [f_{t+1}(x)] \cap \Gamma^{h(n+1)}\,.$$

Lemma 5.3 says $D^{n+1}_{t+1} \subseteq D^{n+1}_t$. Hence, by induction hypothesis (V) the function f_t is injective on the set $[x] \cap D^{n+1}_{t+1} \cap L^{n+1}_t$. Thus $|f([x] \cap D^{n+1}_{t+1} \cap L^{n+1}_t)| = k$. This implies the claim (3). We have proved (2) and (3).
Now we can define $f_{t+1}(y)$. Let $z_1, \ldots, z_l$ be the lexicographically ordered list of strings in $([f_{t+1}(x)] \cap \Gamma^{h(n+1)}) \setminus f_t([x] \cap D^{n+1}_{t+1} \cap L^{n+1}_t)$ and let $y_1, \ldots, y_l$ be the first l strings (according to the lexicographical ordering) in $([x] \cap D^{n+1}_{t+1}) \setminus ([x] \cap D^{n+1}_{t+1} \cap L^{n+1}_t)$. We define

$$f_{t+1}(y) := \begin{cases} z_j & \text{if } y = y_j \text{ for some } j \in \{1, \ldots, l\}, \\ f_{t+1}(x) & \text{otherwise.} \end{cases}$$

This ends the definition of F. It is clear that $\operatorname{dom} F = \mathbb{N} \times \bigcup_m \Sigma^{g(m)}$, and that $F(\mathbb{N} \times \bigcup_m \Sigma^{g(m)}) \subseteq \bigcup_m \Gamma^{h(m)}$. We have to check that F has the properties (I) to (V). In the following proofs we may always assume by induction hypothesis that any of these conditions is true for smaller values of t or for shorter strings x, y.

Proof of (I). We have $f_t(\lambda) = \lambda$ by construction for all t. Hence, (I) is true for $x = \lambda$ and all $t \in \mathbb{N}$. Fix numbers $t, n \in \mathbb{N}$ and a string $y \in \Sigma^{g(n+1)}$. It is sufficient to prove

$$f_t(y) \sqsubseteq f_{t+1}(y).$$

This is clear by construction of f_{t+1} in the 1st case. Set $x := y[0..g(n)-1]$. First, we claim that our assumption $y \notin L^{n+1}_t$ in the 2nd and 3rd case implies $f_t(y) = f_t(x)$. The assumption $y \notin L^{n+1}_t$ implies $|f_t(y)| \leq h(n)$. We have either $|f_t(x)| = h(n)$, in which case the induction hypothesis (II) implies $f_t(y) = f_t(x)$, or we have $|f_t(x)| < h(n)$, in which case the induction hypothesis (IV) implies that x is not f_t-nonterminal, hence $f_t(y) = f_t(x)$. We have proved the first claim: in the 2nd and 3rd case we have $f_t(y) = f_t(x)$. By induction hypothesis (I) we have $f_t(x) \sqsubseteq f_{t+1}(x)$. Finally, in the 2nd and 3rd case we have $f_{t+1}(x) \sqsubseteq f_{t+1}(y)$ by construction of f_{t+1}. Summarizing the last three statements gives

$$f_t(y) = f_t(x) \sqsubseteq f_{t+1}(x) \sqsubseteq f_{t+1}(y)$$

in the 2nd and 3rd case. □

Proof of (II). For $t = 0$, Property (II) follows immediately from the definition of f_0 and by induction. For general t fix numbers $t, n \in \mathbb{N}$ and a string $y \in \Sigma^{g(n+1)}$ and set $x := y[0..g(n)-1]$. It is sufficient to prove

$$f_{t+1}(x) \sqsubseteq f_{t+1}(y).$$

In the 2nd and 3rd case this follows immediately from the definition of $f_{t+1}(y)$. In the 1st case we have

$$f_t(x) \sqsubseteq f_t(y) = f_{t+1}(y)$$

by induction hypothesis (II) and by the construction of f_{t+1}. Hence it is sufficient to prove $f_{t+1}(x) = f_t(x)$. Indeed, the 1st case assumption $y \in L^{n+1}_t$ and $|f_t(x)| \leq h(n)$ (induction hypothesis (III)) imply that x is f_t-nonterminal. Using the induction hypothesis (IV) we conclude $|f_t(x)| = h(n)$. With $f_t(x) \sqsubseteq f_{t+1}(x)$ (induction hypothesis (I)) and $|f_{t+1}(x)| \leq h(n)$ (induction hypothesis (III)) we obtain $f_{t+1}(x) = f_t(x)$. □

Proof of (III). For $t = 0$, Property (III) follows immediately from the definition of f_0 and by induction. For general t observe first that $f_t(\lambda) = \lambda$ for all t. Fix numbers $t, n \in \mathbb{N}$ and a string $y \in \Sigma^{g(n+1)}$ and set $x := y[0..g(n)-1]$. It is sufficient to prove

$$|f_{t+1}(y)| \leq h(n+1).$$

In the 1st case in the definition of $f_{t+1}(y)$ this follows from $f_{t+1}(y) = f_t(y)$ and from $|f_t(y)| \leq h(n)$ (induction hypothesis (III)), and in the 2nd and 3rd case this follows from the definition of $f_{t+1}(y)$ and from $|f_{t+1}(x)| \leq h(n)$ (induction hypothesis (III)). □

For the proof of (IV) we need an additional property:

(VI) $L^n_{t+1} \subseteq L^n_t \cup D^n_{t+1}$ for all $t, n \in \mathbb{N}$.

We shall prove it immediately after (IV).

Proof of (IV). For $t = 0$, Property (IV) follows immediately from the definition of f_0 and from $|f_0(x)| \leq h(n)$ for $x \in \Sigma^{g(n)}$ (induction hypothesis (III)). For the case of general t, fix numbers $t, n \in \mathbb{N}$ and a string $x \in \Sigma^{g(n)}$. We wish to show

$$x \in L^n_{t+1} \iff x \text{ is } f_{t+1}\text{-nonterminal.}$$

First we assume $x \notin L^n_{t+1}$. Let y be an arbitrary string in $[x] \cap \Sigma^{g(n+1)}$. Then in the definition of $f_{t+1}(y)$ we are not in the 3rd case. Can the 1st case be valid? No, because the 1st case condition $y \in L^{n+1}_t$ together with $|f_t(x)| \leq h(n)$ (induction hypothesis (III)) and the induction hypothesis (IV) would imply $x \in L^n_t$. This and $f_t(x) \sqsubseteq f_{t+1}(x)$ (induction hypothesis (I)) and $|f_{t+1}(x)| \leq h(n)$ (induction hypothesis (III)) would imply $x \in L^n_{t+1}$ in contradiction to out assumption. Thus, in the definition of $f_{t+1}(y)$ the 2nd case is valid. But then $f_{t+1}(y) = f_{t+1}(x)$. So far we have shown: if $x \notin L^n_{t+1}$, then for all $y \in [x] \cap \Sigma^{g(n+1)}$, $f_{t+1}(y) = f_{t+1}(x)$. This implies $y \notin L^{n+1}_{t+1}$. By induction we obtain $f_{t+1}(z) = f_{t+1}(x)$ for all $z \in [x] \cap \bigcup_{m \geq n} \Sigma^{g(m)}$. This means that x is not f_{t+1}-nonterminal.

Secondly, we assume $x \in L^n_{t+1}$. If also $x \in L^n_t$, then by induction hypothesis (IV) there is a string $y \in [x] \cap L^{n+1}_t$. Then $f_{t+1}(y) = f_t(y) \in \Gamma^{h(n+1)}$ by definition of $f_{t+1}(y)$ (1st case) and hence, x is f_{t+1}-nonterminal. If $x \notin L^n_t$, then by induction hypothesis (VI) $x \in D^n_{t+1}$. That is, we have $x \in L^n_{t+1} \cap D^n_{t+1}$. If $[x] \cap D^{n+1}_{t+1} \cap L^{n+1}_t \neq \emptyset$, then for strings y in this set we obtain $f_{t+1}(y) = f_t(y) \in \Gamma^{h(n+1)}$ by the 1st case. If this set is empty, then according to the 3rd case there are strings $y \in [x] \cap \Sigma^{g(n+1)}$ with $f_{t+1}(y) \in \Gamma^{h(n+1)}$. In any case x is f_{t+1}-nonterminal. □

Proof of (VI). The statement $L^0_{t+1} \subseteq L^0_t \cup D^0_{t+1}$ is obviously true for any $t \in \mathbb{N}$ because $L^0_t = \{\lambda\} = D^0_t$ for all t. Now fix $t, n \in \mathbb{N}$ and a string $y \in L^{n+1}_{t+1}$. We have to show that $y \in L^{n+1}_t$ or $y \in D^{n+1}_{t+1}$. Assume $y \notin L^{n+1}_t$. Set $x := y[0..g(n)-1]$. In the construction of $f_{t+1}(y)$ either the 2nd or the 3rd case must be valid. But we cannot have $f_{t+1}(y) = f_{t+1}(x)$ because together with $|f_{t+1}(x)| \leq h(n)$ (induction hypothesis (III)) this would contradict $y \in L^{n+1}_{t+1}$. Hence, $f_{t+1}(y)$ must be defined according to the second subcase of the 3rd case, that is, we have $y = y_j$ for some

$y_j \in [x] \cap D^{n+1}_{t+1}$. We have shown: if $y \in L^{n+1}_{t+1} \setminus L^{n+1}_t$, then $y \in D^{n+1}_{t+1}$. That was to be shown. $\square$

Proof of (V). For $t = 0$, Property (V) follows immediately from Lemma 5.2, from the definition of f_0, and from $|f_0(x)| \leq h(n)$ for $x \in \Sigma^{g(n)}$ (induction hypothesis (III)). We fix numbers $t, n \in \mathbb{N}$ and fix a string $x \in D^n_{t+1} \cap L^n_{t+1}$. We have to show that f_{t+1} maps $[x] \cap D^{n+1}_{t+1} \cap L^{n+1}_{t+1}$ bijectively onto $[f_{t+1}(x)] \cap \Gamma^{h(n+1)}$. For elements $y \in [x] \cap D^{n+1}_{t+1} \cap L^{n+1}_{t+1}$ the value $f_{t+1}(y)$ must be defined according to the 1st case or the first subcase of the 3rd case (because of $|f_{t+1}(x)| \leq h(n)$ by induction hypothesis (III)). Hence, the set $[x] \cap D^{n+1}_{t+1} \cap L^{n+1}_{t+1}$ splits into the set $[x] \cap D^{n+1}_{t+1} \cap L^{n+1}_t$, on which f_{t+1} is defined according to the 1st case, and into the set $[x] \cap D^{n+1}_{t+1} \cap L^{n+1}_{t+1} \setminus [x] \cap D^{n+1}_{t+1} \cap L^{n+1}_t$, on which f_{t+1} is defined according to the first subcase of the 3rd case. In the discussion in the 3rd case we have seen that f_t (and hence f_{t+1}) maps the set $[x] \cap D^{n+1}_{t+1} \cap L^{n+1}_t$ injectively into the set $[f_{t+1}(x)] \cap \Gamma^{h(n+1)}$. The definition of f_{t+1} in the 3rd case ensures that indeed f_{t+1} maps the set $[x] \cap D^{n+1}_{t+1} \cap L^{n+1}_{t+1}$ bijectively onto the set $[f_{t+1}(x)] \cap \Gamma^{h(n+1)}$. $\square$

We have proved that F has all the properties (I) to (V). Properties (I), (II), and (III) tell us that by

$$f(x) := \text{the longest string in } \{f_t(y) \mid y \sqsubseteq x \text{ and } t \in \mathbb{N}\}$$

a total function $f : \Sigma^* \to \Gamma^*$ is welldefined. We claim that it proves the theorem.

By properties (I) and (II) f is monotonic. The function f is approximable because F is computable. We claim that for any $n \in \mathbb{N}$ and any string $x \in \Sigma^*$ with $g(n) \leq |x| < g(n+1)$

$$|f(x)| \geq h(n) \iff x \text{ is } f\text{-nonterminal}. \tag{4}$$

Fix a string $x \in \Sigma^*$ with length $g(n) \leq |x| < g(n+1)$. For the only-if part assume that $|f(x)| \geq h(n)$. By definition of f we have $f(x) = f(x[0..g(n)-1])$ and there must be a number t with $f_t(x[0..g(n)-1]) = f(x[0..g(n)-1])$. By property (III) we have $|f_t(x[0..g(n)-1])| = h(n)$ and by property (IV) the string $x[0..g(n)-1]$ must be f_t-nonterminal. Hence, also x is f-nonterminal. For the if part in claim (4) assume that x is f-nonterminal. Then there must be a string $y \in [x] \cap \Sigma^*$ with $f(y) \neq f(x)$. By the definition of f we can assume that $y \in \bigcup_{m \geq n+1} \Sigma^{g(m)}$. For large enough t we have $f(x[0..g(n)-1]) = f_t(x)$ and $f(y) = f_t(y)$. Hence, $x[0..g(n)-1]$ is f_t-nonterminal. By property (IV) we conclude $|f_t(x[0..g(n)-1])| = h(n)$, and hence also $|f(x)| \geq |f(x[0..g(n)-1])| = h(n)$. This ends the proof of the claim (4). This claim implies that f is a process and that f satisfies the second assertion in Theorem 3.

Finally we have to show that $\overline{f}(\mathbf{C} \cap \mathrm{dom}\, \overline{f}) = \Gamma^\omega$. Note that $\overline{f}$ is well-defined because f is monotonic. We need a lemma about the sets

$$M^n_t := L^n_t \cap D^n_t$$

for $t, n \in \mathbb{N}$.

Lemma 6. *Fix $t, n \in \mathbb{N}$. If $x \in M^n_t \setminus M^n_{t+1}$, then $x \notin M^n_s$ for all $s > t$.*

Proof. If $x \in M^n_t$, then $x \in L^n_t$. By (I) and (III) $x \in L^n_{t+1}$. With $x \notin M^n_{t+1}$ we conclude $x \notin D^n_{t+1}$. Lemma 5.3 implies $x \notin D^n_s$ for any $s > t$. $\square$

Corollary 7. *For each $n \in \mathbb{N}$ there is a $t \in \mathbb{N}$ with $M^m_s = M^m_t$ for all $s \geq t$ and $m \leq n$.*

Proof. The assertion follows from the last lemma and from the fact that each set M^m_s is a subset of the finite set $\Sigma^{g(m)}$. $\square$

We define the function $s : \mathbb{N} \to \mathbb{N}$ by

$$s(n) := \min\{t \in \mathbb{N} \mid M^m_r = M^m_t \text{ for all } r \geq t \text{ and } m \leq n\}\,.$$

Property (III) implies that $|f(x)| \leq h(m)$ for all $x \in \Sigma^{g(m)}$, $m \in \mathbb{N}$. Hence, the function f coincides with $f_{s(n+1)}$ on the sets $M^n_{s(n+1)}$ and $M^{n+1}_{s(n+1)}$ and that $|f(x)| = h(n)$ for $x \in M^n_{s(n+1)}$, for any n. By property (V), applied to $s(n+1)$, for each $x \in M^n_{s(n+1)}$, the function f maps the set $[x] \cap M^{n+1}_{s(n+1)}$ bijectively onto the set $[f(x)] \cap \Gamma^{h(n+1)}$. Note that $M^n_{s(n+1)} = M^n_{s(n)}$. We claim that for each n the function

$$f \text{ maps the set } \Sigma^{g(n)} \cap \bigcap_{m \leq n} [M^m_{s(m)}] \text{ bijectively onto } \Gamma^{h(n)}\,. \tag{5}$$

This is clear for $n = 0$ because $g(0) = h(0) = 0$ and $M^0_t = \{\lambda\}$ for all t, especially $M^0_{s(0)} = \{\lambda\}$. Assume that it is true for n. We have just seen that for each $x \in \Sigma^{g(n)} \cap \bigcap_{m \leq n}[M^m_{s(m)}]$ the function f maps $[x] \cap M^{n+1}_{s(n+1)}$ bijectively onto the set $[f(x)] \cap \Gamma^{h(n+1)}$. This gives the claim (5) for $n+1$.

We define $\mathbf{B} \subseteq \Sigma^\omega$ by $\mathbf{B} := \bigcap_n [M^n_{s(n)}]$. By (5), $\overline{f}$ maps $\mathbf{B}$ bijectively onto Γ^ω. We claim that $\mathbf{B} \subseteq \mathbf{C}$. Let $\alpha \in \mathbf{B}$. Then for every n, $\alpha[0..g(n)-1] \in M^n_{s(n)} \subseteq D^n_{s(n)}$. Hence, $[\alpha[0..g(n)-1]] \cap \mathbf{C}_{s(n)} \neq \emptyset$, hence $[\alpha[0..g(n)-1]] \cap \mathbf{C} \neq \emptyset$. Since $\mathbf{C}$ is closed we obtain $\alpha \in \mathbf{C}$ and thus $\mathbf{B} \subseteq \mathbf{C}$. This ends the proof of $\overline{f}(\mathbf{C} \cap \operatorname{dom} \overline{f}) = \Gamma^\omega$. We have proved Theorem 3.

4 Surjective Mappings on Co-r.e. Closed Sets with Positive Measure

In this section we prove the following two results.

Proposition 8. *Let Σ be a finite alphabet. Every closed subset of Σ^ω with positive measure contains a computably growing Cantor set.*

Hence, we can apply Corollary 4 in order to obtain for any co-r.e. closed set $\mathbf{C} \subseteq \Sigma^\omega$ a computable map F with $F(\mathbf{C} \cap \operatorname{dom} F) = \Gamma^\omega$. The following theorem gives an effective version of such a map.

Theorem 9. *Let Σ and Γ be two alphabets and set $p := |\Sigma|$ and $q := |\Gamma|$. Let $\mathbf{C} \subseteq \Sigma^\omega$ be a co-r.e. closed set with positive measure. For every $\varepsilon > 0$ there exist a constant c and a process $f : \Sigma^* \to \Gamma^*$ with $\overline{f}(\mathbf{C} \cap \operatorname{dom} \overline{f}) = \Gamma^\omega$ and*

$$|f(x)| \geq \log_q p \cdot |x| - (2 + \varepsilon) \cdot \log_q p \cdot \sqrt{|x| \cdot \log_p |x|} - c$$

for all f-nonterminating $x \in \Sigma^* \setminus \{\varepsilon\}$.

Note that in the binary case $|\Sigma| = |\Gamma| = 2$ the loss of digits $(2+\varepsilon) \cdot \sqrt{|x| \cdot \log_2 |x|} - c$ is asymptotically smaller than the loss of digits $3 \cdot \sqrt{|x|} \cdot \log_2(x) + c$ in the process constructed by Gács. By introducing lower order terms one can certainly get rid of the ε in the highest order error term.

Remark. Before we start with the proof we give a very informal explanation for the order $\Theta(\sqrt{|x| \cdot \log_2 |x|})$ of the error term. We consider the binary case. Later we shall see that in order to obtain a $(g, 2^{h(n+1)-h(n)})$-Cantor set we need a function g which grows slightly faster than the function h. In fact, formula (6) expresses roughly that $2^{g'(n)}$ grows at least as fast as $\frac{-1}{l'(n)} \cdot 2^{h'(n)}$ where l is a positive decreasing function. This implies that $l'(n)$ is at least of the order $1/n$. Hence g' grows at least as fast as $h' + c \cdot \log_2 n$, or, after integrating, $g(n)$ grows at least as fast as $h(n) + c \cdot n \log_2 n$. But $h(n)$ should be as large as possible compared with $g(n+1)$ because every non-terminating string x with length $g(n) \leq |x| < g(n+1)$ is mapped to a string of length $h(n)$. The difference between $g(n+1)$ and $g(n)$ is roughly $g'(n)$. Hence, we have to make $g'(n) + c \cdot n \log_2 n$ as small as possible compared with $g(n+1)$. This is the case if $g'(n)$ is of the same order as $n \log_2 n$, hence if $g(n)$ is of the order $n^2 \log_2 n$. Then the error term is just $\Theta(n \log_2 n) = \Theta(\sqrt{g(n+1) \log_2 g(n+1)})$.

Let Σ be an alphabet, $p := |\Sigma|$, $\mathbf{C} \subseteq \Sigma^\omega$ a closed set with positive measure, $l : \mathbb{N} \to \{x \in \mathbb{R} \mid x \geq 0\}$ a non-increasing function, and $g : \mathbb{N} \to \mathbb{N}$ an increasing function. We define

$$C^{\{n\}} := \{x \in \Sigma^{g(n)} \mid \mu([x] \cap \mathbf{C}) \geq l(n) \cdot p^{-g(n)}\}$$

for each $n \in \mathbb{N}$ and

$$\mathbf{A} := \bigcap_n [C^{\{n\}}] .$$

Lemma 10. *1. The set* $\mathbf{A}$ *is a subset of* $\mathbf{C}$.
2. For each n *we have* $\mathbf{A}^{[g(n)]} = \Sigma^{g(n)} \cap \bigcap_{m \leq n} [C^{\{m\}}]$.

Proof. 1. The set $\mathbf{A}$ is a subset of Σ^ω. If $\alpha \in \mathbf{A}$, then for each n the set $[\alpha[0..g(n)-1]]$ contains elements from $\mathbf{C}$. We conclude $\alpha \in \mathbf{C}$ because $\mathbf{C}$ is closed. Thus, $\mathbf{A} \subseteq \mathbf{C}$.

2. For the proof of the inclusion "$\subseteq$" fix a number n, assume that $\mathbf{A}^{[g(n)]}$ is nonempty, fix a string $x \in \mathbf{A}^{[g(n)]}$, and fix a sequence $\alpha \in [x] \cap \mathbf{A}$. For all $m \in \mathbb{N}$ we see $\alpha \in [C^{\{m\}}]$, hence $\alpha[0..g(m)-1] \in C^{\{m\}}$. This implies for all $m \leq n$

$$x = \alpha[0..g(n)-1] \in \Sigma^{g(n)} \cap [C^{\{m\}}] .$$

For the inclusion "$\supseteq$" we start with noticing that the assumption that the function l is non-increasing implies

$$x \in C^{\{n\}} \Rightarrow \exists y \in [x] \cap C^{\{n+1\}}$$

for all n. By taking the limit we obtain

$$x \in C^{\{n\}} \Rightarrow \exists \alpha \in [x] \cap \bigcap_{m \geq n} [C^{\{m\}}]$$

Assume that the set $\Sigma^{g(n)} \cap \bigcap_{m\le n}[C^{\{m\}}]$ is nonempty. Fix a string $x \in \Sigma^{g(n)} \cap \bigcap_{m\le n}[C^{\{m\}}]$. We have just seen that there is a sequence $\alpha \in [x] \cap \bigcap_{m\ge n}[C^{\{m\}}]$. By definition of x it is in $[C^{\{m\}}]$ for all $m \in \mathbb{N}$, hence in $\mathbf{A}$. This implies $x \in \mathbf{A}^{[g(n)]}$. This ends the proof of the second assertion of the lemma. $\square$

Let additionally $h : \mathbb{N} \to \mathbb{N}$ be an increasing function and $q \ge 2$ an integer.

Lemma 11. *If $l(0) \le \mu(\mathbf{C})$ and for each $n \in \mathbb{N}$ and $x \in C^{\{n\}}$ the set $[x] \cap C^{\{n+1\}}$ contains at least $q^{h(n+1)-h(n)}$ strings, then $\mathbf{A}$ is a $(g, n \mapsto q^{h(n+1)-h(n)})$-Cantor set.*

Proof. By the last lemma $\mathbf{A}^{[g(0)]} = \Sigma^{g(0)} \cap [C^{\{0\}}] = C^{\{0\}}$. The condition $l(0) \le \mu(\mathbf{C})$ implies $C^{\{0\}} \ne \emptyset$, hence $\mathbf{A} \ne \emptyset$. If $x \in \mathbf{A}^{[g(n)]}$, then by the last lemma $x \in C^{\{n\}}$. By assumption there are at least $q^{h(n+1)-h(n)}$ strings in $[x] \cap C^{\{n+1\}}$. By the last lemma all of them lie in $\mathbf{A}^{[g(n+1)]}$. This shows that $\mathbf{A}$ is a $(g, n \mapsto q^{h(n+1)-h(n)})$-Cantor set. $\square$

¿From now on we assume that $l(0) < 1$.

Lemma 12. *If $l(0) \le \mu(\mathbf{C})$ and for each $n \in \mathbb{N}$*

$$q^{h(n+1)-h(n)} \le \frac{l(n) - l(n+1)}{1 - l(n+1)} \cdot p^{g(n+1)-g(n)}\,, \tag{6}$$

then $\mathbf{A}$ is a $(g, n \mapsto q^{h(n+1)-h(n)})$-Cantor set.

Proof. In the last lemma we have seen that $l(0) \le \mu(\mathbf{C})$ implies that $\mathbf{A}$ is nonempty. Fix a number n and a string $x \in C^{\{n\}}$. Set $k := |[x] \cap C^{\{n+1\}}|$. We estimate

$$\begin{aligned} l(n)p^{-g(n)} &\le \mu([x] \cap \mathbf{C}) \\ &< k \cdot p^{-g(n+1)} + (p^{g(n+1)-g(n)} - k) \cdot l(n+1) \cdot p^{-g(n+1)} \\ &= k(1 - l(n+1))p^{-g(n+1)} + l(n+1)p^{-g(n)}\,. \end{aligned}$$

Hence,

$$k > \frac{l(n) - l(n+1)}{1 - l(n+1)} \cdot p^{g(n+1)-g(n)}\,,$$

Thus, by our assumption there are at least $q^{h(n+1)-h(n)}$ strings in $[x] \cap C^{\{n+1\}}$. The last lemma gives the assertion. $\square$

In order to prove Proposition 8 and Theorem 9 it remains to choose appropriate functions l, g, and h. We shall choose functions $l : \mathbb{N} \to \{x \in \mathbb{R} \mid x > 0\}$, $g : \mathbb{N} \to \mathbb{N}$ and $h : \mathbb{N} \to \mathbb{N}$ with the following properties:

(I) The function l is non-increasing and $l(0) < \mu(\mathbf{C})$,
(II) g and h are increasing and computable and satisfy $g(0) = h(0) = 0$,
(III) the inequality (6) is true for all $n \in \mathbb{N}$,

(IV) for every $\varepsilon > 0$ there is a constant c such that

$$h(n) \geq \log_q(p) \cdot (g(n+1) - (2+\varepsilon) \cdot \sqrt{g(n+1)\log_p(g(n+1))} - c) \qquad (7)$$

for all $n \in \mathbb{N}$.

Properties (I), (II), (III), Lemma 10.1, and Lemma 12 prove Proposition 8. Theorem 9 follows from all four properties, Lemma 10.1, Lemma 12, and Theorem 3, because the process f constructed in Theorem 3 maps an f-nonterminating string x with length $g(n) \leq |x| < g(n+1)$ to a string $f(x)$ with length $|f(x)| = h(n)$.

We choose a constant $c_1 \in \mathbb{N}$ so large such that $0 < 1/\mathrm{LogLog}\, c_1 < \mu(\mathbf{C})$ and define l by

$$l(n) := \frac{1}{\mathrm{LogLog}(n + c_1)}$$

where Log denotes the natural logarithm. We define two functions $G, H : \{n \in \mathbb{N} \mid n \geq \max\{p,q\}\}$ by

$$G(n) := \frac{1}{2}n^2 \log_p n + n\,(\log_p n + 2\log_p \mathrm{Log}\, n + c_2)$$
$$H(n) := \frac{1}{2}n^2 \log_q n$$

where $c_2 \in \mathbb{N}$ is a sufficiently large constant (this will be explained more precisely below) and we choose two computable functions $g, h : \mathbb{N} \to \mathbb{N}$ with $g(0) := h(0) := 0$ and for $n \geq 1$

$$|-g(n) + G(n + \max\{p,q\})| < 1 \quad \text{and} \quad |-h(n) + H(n + \max\{p,q\})| < 1\,. \quad (8)$$

It is clear that there exist computable functions g and h with these properties.

We claim that these functions l, g, and h have all the desired properties. By definition l is decreasing, $l(0) < \mu(\mathbf{C})$, $g(0) = h(0) = 0$, and g and h are computable. If follows also immediately from the definition that g and h are increasing. We show that (6) is true for all $n \in \mathbb{N}$. For $n = 0$ (6) means

$$q^{h(1)} \leq \frac{l(0) - l(1)}{1 - l(1)} p^{g(1)}\,.$$

This is obviously true if the constant c_2 is sufficiently large. Now fix an integer $n \geq 1$ and set $m := n + \max\{p,q\} + 1$. In view of (8) the inequality (6) is true if

$$q^{H(m)-H(m-1)} \cdot q^2 \leq \frac{l(n) - l(n+1)}{1 - l(n+1)} p^{G(m)-G(m-1)} \cdot p^{-2}\,.$$

This is equivalent to

$$q^2 p^2 \cdot \frac{1 - l(n+1)}{l(n) - l(n+1)}$$
$$\leq p^{m\cdot(\log_p m + 2\log_p \mathrm{Log}\, m + c_2) - (m-1)\cdot(\log_p(m-1) + 2\log_p \mathrm{Log}\,(m-1) + c_2)}\,.$$

Indeed, we obtain

$$
\begin{aligned}
& q^2p^2 \cdot \frac{1-l(n+1)}{l(n)-l(n+1)} \\
& \leq q^2p^2 \cdot \frac{1}{l(n)-l(n+1)} \\
& \leq q^2p^2 \cdot \frac{-1}{l'(n+1)} \\
& = q^2p^2 \cdot (n+1+c_1) \cdot \mathrm{Log}\,(n+1+c_1) \cdot (\mathrm{LogLog}\,(n+1+c_1))^2 \\
& \leq m \cdot (\mathrm{Log}\, m)^2 \cdot p^{c_2} \\
& = p^{\log_p m + 2\log_p \mathrm{Log}\, m + c_2} \\
& \leq p^{m\cdot(\log_p m + 2\log_p \mathrm{Log}\, m + c_2) - (m-1)\cdot(\log_p (m-1) + 2\log_p \mathrm{Log}(m-1) + c_2)}\,.
\end{aligned}
$$

The first estimation is obvious and the second follows because the function l' is increasing. The next estimate is true for small n if c_2 is large enough, and true for large n because the term $(n+1+c_1)\cdot\mathrm{Log}\,(n+1+c_1)\cdot(\mathrm{LogLog}\,(n+1+c_1))^2$ increases slower than the term $m\cdot(\mathrm{Log}\, m)^2 = (n+1+\max\{p,q\})\cdot(\mathrm{Log}\,(n+1+\max\{p,q\}))^2$. The last estimate is again obvious. We have shown that (6) is true for all n if c_2 is chosen sufficiently large.

Finally we have to show that also (IV) is true, i.e. that for given $\varepsilon > 0$ there is a constant c such that (7) is true for all n. The inequality (7) is equivalent to

$$(2+\varepsilon) \cdot \sqrt{g(n+1) \cdot \log_p(g(n+1))} + c \geq g(n+1) - \log_p(q)h(n)\,.$$

Since we can choose c as large as we wish it is sufficient to prove this for large n, especially for $n \geq 1$. Since the function $x \mapsto \sqrt{x \log_p x}$ does not grow faster than $x \mapsto x$ replacing $g(n+1)$ by $G(n+1+\max\{p,q\})$ and $h(n)$ by $H(n+1+\max\{p,q\})$ changes the terms on both sides at most by a constant. Hence, it is sufficient to show that there is a constant c such that for large n

$$(2+\varepsilon) \cdot \sqrt{G(n) \cdot \log_p(G(n))} + c \geq G(n) - \log_p(q)H(n-1)\,. \tag{9}$$

Indeed, we obtain

$$
\begin{aligned}
(2+\varepsilon) \cdot \sqrt{G(n) \cdot \log_p(G(n))} & \geq (2+\varepsilon) \cdot \sqrt{1/2} \cdot n \cdot \sqrt{\log_p n} \cdot \sqrt{2 \cdot \log_p n} \\
& = (2+\varepsilon) \cdot n \cdot \log_p n
\end{aligned}
$$

and on the other hand

$$
\begin{aligned}
& G(n) - \log_p(q)H(n-1) \\
& = \frac{1}{2} \cdot (n^2 \log_p n - (n-1)^2 \log_p(n-1)) + n(\log_p n + 2\log_p \mathrm{Log}\, n + c_2) \\
& \leq 2 \cdot n \cdot \log_p n + n(2\log_p \mathrm{Log}\, n + c_2)\,.
\end{aligned}
$$

The last summand $n(2\log_p \mathrm{Log}\, n + c_2)$ is of lower order than $n \cdot \log_p n$. Hence, because of $\varepsilon > 0$ we conclude that (9) is true for large n. This ends the proof of (7), i.e. of property (IV). We have proved Proposition 8 and Theorem 9.

Acknowledgements

The author thanks Cristian Calude for interesting discussions on the subject of this paper. He is supported by the DFG Research Grant No. HE 2489/2-1.

References

[Calude 1994] C. S. Calude. *Information and Randomness, an Algorithmic Perspective.* Springer-Verlag, Berlin, 1994.

[Gács 1986] P. Gács. Every sequence is reducible to a random one. *Information and Control*, 70:186–192, 1986.

[Kučera 1985] A. Kučera. Measure, Π^0_1-classes and complete extensions of PA. In H.-D. Ebbinghaus et al., editor, *Recursion Theory Week, Proceedings, Oberwolfach 1984*, volume 1141 of *Lecture Notes in Math.*, pages 245–259, Berlin, 1985. Springer-Verlag.

[Măndoiu 1993] I. Măndoiu. On a theorem of Gács. *Internat. J. Computer Math.*, 48:157–169, 1993.