C. Calude, C. Câmpeanu. Are binary codings universal?,
Complexity 1, 15 (1996). (in press)

Are Binary Codings Universal?*

Cristian Calude† **Cezar Câmpeanu**‡

Abstract

It is common sense to notice that one needs fewer digits to code numbers in ternary than in binary; new names are about $\log_3 2$ times shorter. Is this trade-off a consequence of the special coding scheme? The answer is negative. More generally, we argue that the answer to the question, stated in the title and formulated to the first author by C. Rackhoff, is in fact *negative*. The conclusion will be achieved by studying the role of the size of the alphabet in constructing instantaneous codes for all natural numbers, and defining random strings and sequences. We show that there is no optimal instantaneous code for all positive integers, and the binary is the worst possible. Codes over a fixed alphabet can be indefinitely improved themselves, but only "slightly"; in contrast, changing the size of the alphabet determines a significant, not linear, improvement. The key relation describing the above phenomenon can be expressed in terms of Chaitin complexity: changing the size of the coding alphabet from q to Q, $2 \leq q < Q$, results in an improvement of the complexity by a factor of $\log_Q q$.

As a consequence, a string avoiding consistently a fixed letter is not random. In binary, this corresponds to a trivial situation. In the non-binary case the distinction is relevant: more than 3.2^n ternary strings of length n are not random (many of these strings are binary random). This phenomenon is even sharper for infinite sequences.

Keywords: Instantaneous codes, Chaitin complexity, random strings, random sequences.

*This work has been supported by Auckland University Research Grant A18/XXXXX/62090/3414030.

†Computer Science Department, The University of Auckland, Private Bag 92019, Auckland, New Zealand; email: cristian@cs.auckland.ac.nz.

‡Faculty of Mathematics, Bucharest University, Str. Academiei 14, R-70109 Bucharest, Romania; email: campeanu@roimar.imar.ro.

1 Introduction

This note answers in the *negative* the question formulated by C. Rackhoff [17] and stated in the title. Apparently, the general feeling is that the binary case encompasses the whole strength and generality of coding phenomena, at least from an algorithmic point of view. This idea appears in the strongest form in Li and Vitányi [16], p. 147:

> *[the] measure treated in the main text is universal in the sense that neither the restriction to binary objects to be described, nor the restriction to binary descriptions (programs) results in any loss of generality.*

In contrast, we think that *descriptional complexity in the binary and non-binary cases has crucial differences;* some of them have been studied in Calude [1], Calude and Chiţescu [3], and particularly, Calude, Jürgensen and Salomaa [5] (see also [2]). These results are compatible with facts from classical information theory; see, for example, Rissanen ([19], p 27)[1] or the analysis in Fenwick [11].

2 Prerequisites

Denote by $\mathbf{N}$ and $\mathbf{R}$, respectively, the sets of natural numbers and real numbers. If X is a finite set, then $\#X$ denotes the cardinality of X. We shall use the following functions: i) $\lfloor \alpha \rfloor$, the integral part of the real $\alpha > 0$, ii) $\binom{n}{k}$, the binomial coefficient, and iii) $\log_Q$, the base Q logarithm. For a function $f : \mathbf{N} \to \mathbf{R}$ we write $f = \mathrm{O}(1)$ in case $|f(n)| \leq c$, for some constant $c > 0$ and all natural n. For $f, g : \mathbf{N} \to \mathbf{R}$ we write $f \asymp g$ in case there exists two positive reals α, β such that $f(x) \leq \alpha g(x)$ and $g(x) \leq \beta f(x)$, for almost all $x \in \mathbf{N}$.

For $Q \geq 2$ put $A_Q = \{0, 1, \ldots, Q-1\}$. By A_Q^* we denote the free monoid generated by A_Q (under concatenation). The elements of A_Q^* are called *strings;* λ is the empty string. For x in $A_Q^*, |x|$ is the length of x; $|\lambda| = 0$. For m in $\mathbf{N}$, $A_Q^m = \{x \in A_Q^* \mid \ |x| = m\}$. We denote by $string_Q(n)$ the nth string according to the quasi-lexicographical order: $string_Q(0) = \lambda, string_Q(1) = 0, \ldots, string_Q(Q+1) = 00$, etc.

Working with *partial recursive (p.r.) functions* $\varphi : A_Q^* \stackrel{o}{\to} A_Q^*$ we adopt the notations from Calude [2]. A *Chaitin computer* is a p.r. function $C : A_Q^* \stackrel{o}{\to} A_Q^*$ with a *prefix-free domain* (i.e. there is no pair of distinct strings x, y in the domain of C such that x is a prefix of y).[2] To a Chaitin computer C one

[1] *... the mean code length of an optimal code length does not differ from entropy by more than one bit. For large alphabet sizes this excess bit is tolerable, but certainly not for small one such as the binary alphabet, for which we would not get any compression at all, because each of the two codewords are one symbol long, no matter how skewed the distribution is.*

[2] Prefix-free sets are instantaneous codes.

associates the *self-delimiting complexity* or *Chaitin complexity* $H_{Q,C} : A_Q^* \stackrel{o}{\to} \mathbf{N}, H_{Q,C}(x) = \min\{|y| \mid y \in A_Q^*, C(y) = x\}$, with the convention $\min \emptyset = \infty$. The basic result obtained by Chaitin [7, 8, 9]—called the *Invariance Theorem*, states the existence of a Chaitin computer U (called *universal Chaitin computer*) such that for every Chaitin computer C there exists a constant c (depending upon U and C) such that $H_{Q,U}(x) \leq H_{Q,C}(x)+c$, for all $x \in A_Q^*$. For this paper we fix a universal Chaitin computer and denote by H_Q the induced complexity.

A function $f : X \to \mathbf{N}$ (where X is A_Q^* or $\mathbf{N}_+$) is semi-computable from above in case the set $\{(x,m) \mid x \in X, m \in \mathbf{N}, f(n) \leq m\}$ is recursively enumerable. For example, Chaitin complexity is semi-computable from above (but not computable).

Let $\Sigma_Q : \mathbf{N} \to \mathbf{N}$ be the function defined by

$$\Sigma_Q(n) = \max_{x \in A_Q^n} H_Q(x).$$

In view of a result in Chaitin [9] (see also Calude [2])

$$\Sigma_Q(n) = n + H_Q(string_Q(n)) + \mathrm{O}(1).$$

The random strings of length n are defined to be the strings with maximal Chaitin complexity among the strings of length n, i.e. the strings $x \in A_Q^n$ having $H_Q(x) \approx \Sigma_Q(n)$. A string $x \in A_Q^*$ is **Chaitin** t-**random** (t is a natural number) if $H_Q(x) \geq \Sigma_Q(|x|) - t$. A sequence $\mathbf{x}$ is random if all its prefixes $\mathbf{x}(n) = x_1 x_2 \cdots x_n$ are "almost random" (as strings). Formally, $\mathbf{x} = x_1 x_2 \cdots x_n \cdots \in A_Q^\omega$ is **random** in case $\lim_{n\to\infty}(H_Q(\mathbf{x}(n)) - n) = \infty$.

Various results show that the above definition is adequate, i.e. random strings and sequences share almost all properties associated with randomness; see Calude [2] for a detailed discussion.

3 Complexity of Natural Numbers

Suppose we want to communicate fast and unambiguously a sequence of numbers to a computer (e.g. instructions telling the computer what to do). To guaranty fast communication one has to use names with very low complexity, i.e. short code-strings. To avoid ambiguity we have to ensure that the computer *knows* where one number stops and the next starts. We are naturally lead to consider prefix-free codes for *all natural numbers.*

The domain of a universal binary Chaitin computer is a binary optimal, but not computable, coding for all natural numbers. Computability excludes optimality. However, the infinite improvement which can be achieved is not dramatic, in the sense, described by Knuth [14], that there is an "almost optimum" coding and no other computable scheme can be "much better" than it.

A big difference appears when we compare codings having different sized alphabets. In the usual (positional) system one needs fewer digits to code numbers in ternary than in binary. The improvement is not linear,[3] it is about $\log_3 2$ times better. Is this trade-off a consequence of our special coding? The aim of this section is to prove that the above trade-off is actually general.

We start with a slight improvement of some results due to Chaitin [7].

Lemma 3.1. *Let $f : \mathbf{N}_+ \to A_Q^*$ be an injective (not necessarily recursive) function having a prefix-free range. Then for every natural m the following inequality*

$$|f(n)| > \log_Q n + m,$$

holds true for infinitely many n.

Proof. Assume by absurdity that for some natural m the inequality $|f(n)| \leq \log_Q n + m$ holds true for almost all n. Then, the divergent series $\sum_{n\geq 1} Q^{-\log_Q n - m}$ appears to converge:

$$\sum_{n\geq 1} Q^{-\log_Q n-m} \leq \sum_{n\geq 1} Q^{-|f(n)|} + \mathrm{O}(1) < \infty.$$

$\square$

Corollary 3.2. *For every natural m,*

$$H_Q(string_Q(n)) > \log_Q n + m,$$

for infinitely many n.

It follows that for all m, no prefix-free coding of all natural numbers—over the alphabet A_Q—can be decreased below $\log_Q n + m$, for almost all n. Working with a larger alphabet this improvement can be achieved, even with a trivial, recursive code. For instance, take the code $F : \mathbf{N}_+ \to A_{Q+1}^*$ given by $F(n) = string_Q(n)Q$, the concatenation between $string_Q(n)$ and the new letter Q. Clearly,

$$|F(n)| = \lfloor \log_Q(n(Q-1)+1)\rfloor + 1 \leq \log_Q n + 2.$$

Lemma 3.3. *Let $F : \mathbf{N}_+ \to A_Q^*$ be recursive and injective, and let $g : \mathbf{N}_+ \to \mathbf{N}_+$ be a function, semi-computable from above, such that*

$$\sum_{n\geq 1} Q^{-g(n)} < \infty.$$

[3] Actually, for a fixed alphabet, the Invariance Theorem cannot be improved; see Calude [2], Calude, Jürgensen, Salomaa [5].

Then

$$H_Q(F(n)) \leq g(n) + O(1).$$

Proof. First let us notice that

$$\begin{aligned}
\sum_{n\geq N} \sum_{m\geq g(n)} Q^{-m} &= \sum_{n\geq N} \sum_{m=0}^{\infty} Q^{-(g(n)+m)} \\
&= \sum_{n\geq N} Q^{-g(n)} \sum_{m=0}^{\infty} Q^{-m} \\
&= \frac{Q}{Q-1} \sum_{n\geq N} Q^{-g(n)} \leq 1,
\end{aligned}$$

in case $\sum_{n\geq N} Q^{-g(n)} < Q/(Q-1)$. In view of Kraft-Chaitin inequality, there exists a Chaitin computer $C : A_Q^* \stackrel{o}{\to} A_Q^*$ such that for all $n \geq N$ and $m \geq g(n)$ there exists a string x of length m such that $C(x) = F(n)$. In particular, there exists a string x_n of length $g(n)$ such that $C(x_n) = F(n)$, for $n \geq N$; using the Invariance Theorem we deduce the required formula:

$$H_Q(F(n)) \leq g(n) + \mathrm{O}(1).$$

□

Example 3.4. *For every natural $Q \geq 2$, and every real α, one has*

$$H_Q(string_Q(n)) \leq \alpha \log_Q(n) + O(1),$$

if and only if $\alpha > 1$.

The next result will shed more light on the analysis described at the beginning of this section. We shall prove that *there is no optimal instantaneous code for all natural numbers*, and, the *binary coding is the worst possible.* For larger and larger alphabets we get better and better codings; in contrast with the case of codes over a fixed alphabet, which themselves can be indefinitely, but only "slightly" (see Knuth [14]), improved, changing the alphabet results in a significant improvement.

Theorem 3.5. *Let $q, Q \geq 2$ be naturals and consider two recursive one-one functions $F : \mathbf{N}_+ \to A_Q^*$, $f : \mathbf{N}_+ \to A_q^*$. Then,*

$$H_Q(F(n)) < (\log_Q q) H_q(f(n)) + O(1).$$

Proof. Take the function $g(n) = \lfloor (\log_Q q) H_q(f(n)) \rfloor$. One has:

$$\begin{aligned}\sum_{n\geq 1} Q^{-g(n)} &\leq \sum_{n\geq 1} Q^{1-(\log_Q q)H_q(f(n))} \\ &= Q\sum_{n\geq 1} q^{-H_q(f(n))} \\ &\leq Q < \infty.\end{aligned}$$

As g is semi-computable from above one can use Lemma 3.3 to get the required inequality. □

Corollary 3.6. *Let $2 \leq q < Q$ be naturals. Then, for every string $x \in A_q^*$ one has*

$$H_Q(x) \leq (\log_Q q) H_q(x) + O(1).$$

Proof. Consider the recursive one-one functions $F : \mathbf{N}_+ \to A_Q^*, f : \mathbf{N}_+ \to A_q^*$ given by: $F(n) = f(n) = string_q(n)$, and use Lemma 3.3. □

Corollary 3.7. *For all $2 \leq q \leq Q$,*

$$H_Q(string_Q(n)) \leq (\log_Q q) H_q(string_q(n)) + O(1).$$

4 Measures of Randomness

This section is devoted to an analysis of the role of the size of the alphabet for defining random strings and sequences.[4]

First let us notice the following two relations:

Theorem 4.1. *Let $2 \leq q < Q$. Then, there exists a constant α (which depends upon q, Q) such that for all $x \in A_q^*$ we have:*

$$|H_q(x) - (\log_q Q) H_Q(x)| \leq \alpha.$$

[4]Algorithmic information theory has been developed mainly in the binary case, that is in case the underlying alphabet is binary. We only know a few exceptions, namely Knuth [13], Calude [1, 2], Calude and Chiţescu [3], Kramosil [15], and Calude and Jürgensen [4].

Proof. Use Corollary 3.6 and Lemma 3.3 (take $F : \mathbf{N}_+ \rightarrow A_q^*, F(n) = string_q(n), g(n) = \lfloor (\log_q Q) H_Q(F(n)) \rfloor$.). $\square$

What is the complexity of strings $x \in A_q^*$ regarded as strings in A_Q^*?

Proposition 4.2. *For every $2 \leq q < Q$ and all $x \in A_q^*$,*

$$H_Q(x) < |x| + O(1).$$

Proof. Indeed, by Corollary 3.6 and the relation $H_Q(string_q(n)) = \mathrm{O}(\log_q n)$ one has:

$$\begin{aligned} H_Q(x) &\leq (\log_Q q) H_q(x) + \mathrm{O}(1) \\ &\leq (\log_Q q) \Sigma_q(|x|) + \mathrm{O}(1) \\ &\leq (\log_Q q)(|x| + \mathrm{O}(\log_q |x|)) \\ &= (\log_Q q)|x| + \mathrm{O}(\log_q |x|) \\ &< |x| + \mathrm{O}(1). \end{aligned}$$

$\square$

Corollary 4.3. *No string $x \in A_q^*$ is random over A_Q^*.*

In the binary case we have only two such strings, namely

$$00 \ldots 0 \text{ and } 11 \ldots 1,$$

which are obviously non-random. In the non-binary case we have

$$\sum_{i=2}^{Q-1} i^n \binom{Q}{i}$$

strings of length n over the alphabet A_Q which are non-binary because they do not contain all Q letters. For instance, for $Q = 3$ one has 3×2^n such strings, many of them being random as *binary* strings (in fact, according to a result of Chaitin [10], more then $3 \times 2^{n-c_2}$, where c_2 is a constant which depends on the size of the alphabet but not on the length n).

The above distinction is sharper for random sequences: we pass from two sequences in the binary case to an infinity (of the power of the continuum) in the non-binary cases.

Corollary 4.4. *No sequence $\mathbf{x} \in A_q^\omega$ is random over the alphabet A_Q in case $Q > q \geq 2$.*

Acknowledgment

We express our gratitude to C. Rackhoff, for formulating the problem of this paper, and to G. J. Chaitin, J. Gibbons, H. Jürgensen, A. Salomaa, B. Solovay, and L. Staiger, for useful discussions.

References

[1] C. Calude. *Theories of Computational Complexity*, North-Holland, Amsterdam, 1988.

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

[3] C. Calude, I. Chiţescu. Random strings according to A. N. Kolmogorov and P. Martin-Löf. Classical approach, *Found. Control Engrg.* 7 (1982), 73-85.

[4] C. Calude, H. Jürgensen. Randomness as an invariant for number representations, in H. Maurer, J. Karhumäki, G. Rozenberg (eds.). *New Results and Trends in Theoretical Computer Science*, Springer-Verlag, Berlin, 1994, 44-66.

[5] C. Calude, H. Jürgensen, A. Salomaa. *Coding without Tears*, manuscript, February 1994, 15 pp.

[6] G. J. Chaitin. On the length of programs for computing finite binary sequences, *J. Assoc. Comput. Mach.* 13(1966),547-569. (Reprinted in: [8], 219-244.)

[7] G. J. Chaitin. A theory of program size formally identical to information theory, *J. Assoc. Comput. Mach.* 22(1975), 329-340. (Reprinted in: [8], 197-223.)

[8] G. J. Chaitin. *Information, Randomness and Incompleteness, Papers on Algorithmic Information Theory*, World Scientific, Singapore, 1987. (2nd ed., 1990)

[9] G. J. Chaitin. *Algorithmic Information Theory*, Cambridge University Press, Cambridge, 1987. (third printing 1990)

[10] G. J. Chaitin. On the number of N-bit strings with maximum complexity, *Applied Mathematics and Computation* 59(1993), 97-100.

[11] P. Fenwick. Huffman code efficiency for extensions of sources, *IEEE Trans. Comm.* (in press)

[12] P. Gács. *Lecture Notes on Descriptional Complexity and Randomness*, Boston University, 1988, 62 pp.

[13] D. E. Knuth. *The Art of Computer Programming*, Vol. 2, *Seminumerical Algorithms*, Addison-Wesley, Reading, MA, 2nd ed., 1981.

[14] D. E. Knuth. Supernatural numbers, in D. A. Klarner (ed.). *The Mathematical Gardner,* Prindle, Weber & Schmidt, Wadsworth, Boston, MA, 1981, 310-325.

[15] I. Kramosil. Recursive classification of pseudo-random sequences, *Kybernetika (Prague)* 20(1984), 1-34. (supplement)

[16] M. Li, P. M. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*, Springer-Verlag, Berlin, 1993.

[17] C. Rackhoff. *Personal Communication to C. Calude*, November 1993.

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

[19] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*, World Scientific, Singapore, 1989.