$C(x)$ given x. Applications to left c.e. reals: Computability, Turing-completeness and computational gaps

George Davie

July 2025

Most strings x readily give their own complexity: "my length minus something small". But some strings are *complexity withholding*—there is no short program that takes x and outputs $C(x)$.
These strings do *not* want to give their complexity; that is, there is *no* short program which takes x and outputs $C(x)$.

In other words, $C(C(x) \,|\, x)$ is large.

This paper reveals a surprising phenomenon: complexity withholding strings create *computational gaps.* Even with vastly more computational resources than should be needed, we cannot reach other strings of comparable complexity.

The key lies in the waiting times inherent in computations.

Before we continue, we note that there *are* strings x with large $C(C(x) \,|\, x)$. In fact, by Shen and Bauwens [3][1]: for each n there are strings of length n such that

$$C(C(x) \,|\, x) \geq \log n - O(1).$$

Recall that, since the length $C(C(x) \,|\, x))$ of such a program for x of length n is bounded from above by $\log n$, see Li and Vitányi [1], this is the best possible bound.

For such strings x, x itself does not help at all in finding $C(x)$.

Strings x with large $C(C(x) \,|\, x)$ are rare but play a central role in the theory. One example is in counterexamples to information symmetry of Kolmogorov complexity; see Gács [9]. Also see, for example, Section 2.8 in Li and Vitányi [1].

We will sometimes call strings for which $C(C(x) \,|\, x)$ is not small, *complexity withholding.*

[1] Also see this paper for references to previous work on the function $C(C(x) \,|\, x)$.

Now let α and β be left c.e. reals, (P_α, P_β) a pair of programs generating them from below, and α_n, β_n their first n digits. It is obvious that an α_n computes a β_n, if the *settling time* of α_n—the time it takes for the approximations to the c.e. real to settle on α_n—is longer than that of β_n.

That is, just wait for the growing approximation to α_n to settle and then read off the approximation to β_n—which will be the actual β_n. This will follow if the Kolmogorov complexity $C(\alpha_n)$ is sufficiently greater than $C(\beta_n)$.

In this sense, for any two left c.e. reals, either α_n computes β_n or conversely, since one must settle first.

How hard is it to compute the initial segment settling *last* from the one settling *first*?

Our basic observation is that there exists a small computable band d[2] such that, if α_n is outside the band, $C(\alpha_n) \geq C(\beta_n) + d$, then it is often very hard, requiring a lot of extra information. In fact, we require at least around $C(C(\beta_n) \mid \beta_n)$ extra bits. Since $C(C(\beta_n) \mid \beta_n)$ has no finite bound, and we only want to lift the complexity by d, this is surprising.

We will show that this fact implies sharp results around Chaitin's characterisation of computability in terms of initial segment complexity. For example, we will show that if $C(\alpha_n) \geq C(n) + d$ for all n, then α is Turing-complete in a very strong sense. In some sense, computable sequences are only some constant d away from being Turing-complete. We will also look at Frank Stephan's relativisation of Chaitin's result, in this light.

Further, and most surprisingly, we will show the general difficulty of computing an initial segment α_n of complexity at least d more *or less* than β_n. It will be comparably difficult even to compute *any* n-length string x[3] of complexity at least d less.

Since left c.e. reals are so-called *limit computable*, concepts of domination as in Section 3.5 of Soare [11] all apply. We have not yet explored the deeper relations with these concepts from pure computability theory.

References

[1] Li, Ming, and P. M. Vitányi. 2019. An introduction to Kolmogorov complexity and its applications. Cham Springer Series: Texts in Computer Science

[2] d not much longer than $l(P_\beta) + l(P_\alpha)$

[3] Not necessarily an initial segment of any c.e. real.

[2] A. Kučera and T. A. Slaman. Randomness and recursive enumerability. SIAM Journal on Computing, 31:199-211, 2001.

[3] B. Bauwens, A. Shen, Complexity Of Complexity And Strings With Maximal Plain And Prefix Kolmogorov Complexity, The Journal of Symbolic Logic. 79 (2014) 620632. doi:10.1017/jsl.2014.15.

[4] G.J. Chaitin. Information-theoretic characterizations of recursive infinite strings. Theor. Comput. Sci., 2:4548, 1976.

[5] R.P. Daley. Noncomplex sequences: characterizations and examples. Journal of Symbolic Logic, 41:626-638, 1976.

[6] R. G. Downey. D.R. Hirschfeldt. Algorithmic randomness and complexity, Springer Verlag 2010

[7] A. Nies. Computability and Randomness. Oxford, England: Oxford University Press UK. (2008)

[8] Csima, B. F., and Shore, R. A. (2007). The settling-time reducibility ordering. Journal of Symbolic Logic, 72(3), 1055-1071. doi:10.2178/jsl/1191333856

[9] P. Gács, On the symmetry of algorithmic information, Soviet Math. Dokl., vol. 15 (1974), pp. 1477-1480.

[10] W. Merkle and F. Stephan. On C-degrees, H-degrees and T-degrees. In Twenty-Second Annual IEEE Conference on Computational Complexity (CCC 2007). IEEE Computer Society Press, San Diego, CA, 2007.

[11] Soare, R. I. Turing computability theory and applications, Springer Berlin, 2018