

Article

Not peer-reviewed version

Assembly Theory - Formalizing Assembly Spaces, Discovering Patterns and Bounds

[Wawrzyniec Bieniawski](#) , [Andrzej Tomski](#) , [Szymon Łukaszyk](#) ^{*} , Piotr Masierak , Szymon Tworz

Posted Date: 29 December 2025

doi: 10.20944/preprints202409.1581.v12

Keywords: assembly theory; assembly index; information theory; graph theory; complexity measures; compression algorithms; information entropy; computational biology



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Assembly Theory - Formalizing Assembly Spaces, Discovering Patterns and Bounds

Wawrzyniec Bieniawski ¹, Piotr Masierak ¹, Szymon Tworz ¹, Andrzej Tomski ²
and Szymon Łukaszyk ^{1,*}

¹ Łukaszyk Patent Attorneys, ul. Głowackiego 8, 40-052 Katowice, Poland

² University of Silesia, Institute of Mathematics, Bankowa 14, 40-007 Katowice, Poland

* Correspondence: szymon@patent.pl

Abstract

Assembly theory defines structural complexity as the minimum number of steps required to construct an object in an assembly space. We formalize the assembly space as an acyclic digraph of strings. Key results include analytical bounds on the minimum and maximum assembly indices as functions of string length and alphabet size, and relations between the assembly index (ASI), assembly depth, depth index, Shannon entropy, and expected waiting times for strings drawn from uniform distributions. We identify patterns in minimum- and maximum-ASI strings and provide construction methods for the latter. While computing ASI is NP-complete, we develop efficient implementations that enable ASI computation of long strings. We establish a counterintuitive, inverse relationship between a string ASI and its expected waiting time. Geometric visualizations reveal that ordered decimal representations of low ASI bitstrings of even length N naturally cluster on diagonals and oblique lines of the squares with sides equal to $2^{N/2}$. Comparison with grammar-based compression (Re-Pair) shows that ASI provides superior compression by exploiting global combinatorial patterns. These findings advance complexity measures with applications in computational biology (where DNA sequences must violate Chargaff's rules to achieve minimum ASI), graph theory, and data compression.

Keywords: assembly theory; assembly index; information theory; graph theory; complexity measures; compression algorithms; information entropy; computational biology

1. Introduction

Assembly theory quantifies the minimum number of constructive steps required to generate an object, providing a principled measure of its structural complexity. It is built around two fundamental concepts: an assembly pool (a predefined set of basic items and a set of hitherto assembled subitems) and an assembly step (joining a pair of items taken from the assembly pool in a predefined order and cloning [1] such an assembled subitem to the assembly pool if it is not there), which together define the assembly index (ASI) as the minimal number of steps necessary to construct the object. Since its original formulation in 2017 [2], AT has been generalized to a broad range of theoretical and practical settings. A considerable body of work on AT [3–9,9,10,10–14] showcased the scale invariance of AT: its implementations, whether based on graphs, building blocks, DNA nucleobases, bounded structures, or more abstract units assembled in various spaces, share the same fundamental characteristics.

Intrinsic purpose-driven mechanics implementing global combinatorics (heuristics) of AT provides a genuinely different perspective from classical notions of algorithmic or statistical complexity, despite the fact that AT appears to share many methodological features with classical dictionary-based [15–17] and more particularly grammar-based [18–33] data compression schemes. This resemblance stems from the fact that both AT and compression algorithms seek to capture redundancy, identify reusable substructures, and derive concise generative descriptions of data. However, instead of quantifying randomness or compressibility, AT focuses on causal construction pathways and structural

regularities encoded in minimal assembly pathways (i.e., in the ASI). This perspective naturally raises the question of how such regularities can be efficiently detected, represented, and exploited by formal computational methods.

In this study, we focus on the general string version of AT. The paper is organized as follows. Section 2 introduces definitions and basic theorems used in the paper. Section 3 concerns the minimum complexity strings of AT, showing certain relations between the minimum ASI, assembly depth, depth index, and the Shannon entropy. Section 4 concerns the maximum complexity strings of AT, provides certain bounds of the maximum ASI as a function of a string length and the number of basic symbols, as well as certain methods of constructing high ASI strings. Section 5 concerns results common to the minimum and maximum complexity strings of AT, including a correlation between the expected waiting time for a string of outcomes obtainable from a uniform categorical distribution and the ASI of this string. Section 6 discusses the supremacy of ASI compression over all polynomial-time compression algorithms. Finally, Section 7 summarizes the findings of this study. In general, the proofs of theorems and lemmas, methods, and tables are given in Appendices.

2. General Framework

Consider a string $C_k^{(N,b)}$ (we often write it simply as C_k) of length N made of b distinct basic symbols (unit-length strings) c . We shall label the symbols c using decimal digits $\{0, 1, \dots, b-1\}$ and letters if $b > 10$, where $b_{-k} := b - k$. We can assume that all strings C_k belong to a set C and are vertices of a graph, where certain strings are connected via edges e from a set E . The Definition 2.1 and Lemma 2.1 were already stated in our previous studies [10,12]. We restate them here for clarity.

Definition 2.1 (Assembly Space). *An assembly space $\Omega = (C, E)$ is an acyclic digraph of strings $C = \{C_k\}, k \in \mathbb{N}$, where all $b \in \mathbb{N}$ unit length strings (basic symbol(s)) are source vertices and the remaining strings are 2-in-regular assembly steps vertices, and E is a set of edges.*

Definition 2.2 (Assembly Step). *An assembly step s is the formation of a new string C_z in an assembly space Ω from two, not necessarily different, strings C_x, C_y by concatenating $C_z = C_x \circ C_y = \text{strcat}(C_x, C_y)$ which establishes the edges $e = (C_x, C_z)$ and $e' = (C_y, C_z)$.*

Thus, any edge $e = (C_x, C_z)$ unambiguously resolves to either $e = (C_x, C_x \circ C_y)$ or $e = (C_x, C_y \circ C_x)$. For example, the edge $e = ([010], [0101])$ unambiguously resolves to $e = ([010], [010] \circ [1])$. Even though all the Ω vertices are strings, it is convenient to separate this set into a set $B := C \setminus \{C_k^{(N,b)} \in C : N \neq 1\}$ of source vertices, and a set $S := C \setminus \{C_k^{(N,b)} \in C : N = 1\}$ of 2-in-regular assembly steps vertices.

Definitions 2.1 and 2.2 are consistent: all vertices are unique (in any standard graph, all vertices should be unique) and all are strings. Since an assembly step always consists of joining two parts only [2], this can be thought of as the left and right fragments of the newly formed string, and those strings that can be the result of the concatenation of two shorter strings are assembly step 2-in-regular vertices. The uniqueness of each vertex is a sufficient criterion to establish the admissibility of an assembly step and to introduce the notion of an assembly pool: vertices (strings) present in the assembly space can not be *assembled again* as new vertices of Ω , as they would not be unique.

Definition 2.3 (String Assembly Space). *An assembly space Ω_{C_s} of a string C_s is the assembly space 2.1 containing the vertex C_s and all the vertices leading to the string C_s .*

There can be more than one assembly space for a target string, reflecting different assembly pathways leading to it. However, the ASI $a^{(N,b)}(C_s)$ of a string $C_s^{(N,b)}$ is the minimum cardinality $|S(\Omega_{C_s})|$ of the set of the assembly step vertices $S(\Omega_{C_s})$ of all assembly spaces Ω_{C_s} of the string C_s .

Strings in an assembly space 2.1 can form 2-grams, 3-grams, 4-grams, ..., n -grams, in general, wherein the following holds.

Lemma 2.1. For all b a 4-gram is the shortest string that allows for more than one ASI.

$N = 2$ provides b^2 available 2-grams with unit ASI. $N = 3$ provides b^3 available 3-grams with ASI equal to two. Only $N = 4$ provides b^4 4-grams that include b^2 4-grams with ASI equal to two, that is b 4-grams $C_{k,\min}^{(4,b)} = [***]$ and $b(b-1)$ 4-grams $C_{l,\min}^{(4,b)} = [**]**$, while the ASI of the remaining $b^4 - b^2$ 4-grams is three. For example, to assemble the 4-gram $C_{k,\min}^{(4,4)} = [0202]$, we need to assemble the 2-gram $[02]$ and reuse it, while there is nothing available to reuse, in the case of the 4-gram $C_l^{(4,4)} = [0123]$.

Definition 2.4 (Clear/Mixed String). We call a string (an n -gram) containing only one symbol a clear string (a clear n -gram). We call a string (an n -gram) which is not clear a mixed string (a mixed n -gram).

Theorem 2.1. Each k_r copies of an n_r -gram $C_r^{(n_r,b)}$ contained in a string $C_m^{(N,b)}$ decrease its ASI at least by $k_r(n_r - 1) - a^{(n_r,b)}(C_r)$. That is

$$a^{(N,b)}(C_m) \leq N - 1 - \sum_{r=1}^R \left[k_r(n_r - 1) - a^{(n_r,b)}(C_r) \right], \quad (1)$$

where R is the total number of repeated n_r -grams.

For example, due to the presence of three copies of a 5-gram $[01001]$, each with $a^{(5,6)}([01001]) = 3$, in a string

$$C_k^{(24,6)} = [12|01001|21|01001|235|01001|52], \quad (2)$$

its ASI amounts to $a^{(24,6)}(C_k) = 24 - 1 - (3 \cdot (5 - 1) - 3) = 14$. The relation (1) provides the upper bound on ASI as it does not describe a situation in which n -gram for $n > 2$ is assembled based on a $(n-1)$ -gram also present in the string $C_m^{(N,b)}$, outside of this n -gram. For example, the ASI $a^{(14,9)}([56|101|78|101|43|01]) = 10$, while $14 - 1 - (2(3 - 1) - 2) = 11$.

Consider the strings C_k and C_l , where C_l is C_k in reverse order. The string C_k can have a distribution of n_r -grams leading to its ASI decrease (1). But the string C_l will have the same distribution of n_r -grams in reverse order. Consequently, the strings C_k and C_l have the same ASI.

Definition 2.5 (AT String Equivalence). We consider two strings C_k and C_l to be AT-equivalent if they have the same ASI because C_l is C_k in reverse order and/or if C_l and C_k differ at most in symbol permutations.

If it is not otherwise clear from the context, in this study, we consider equivalent strings (e.g., $\{[001222], [111200], \dots\}$) as one string.

Another measure of a string's complexity is the assembly depth [34].

Definition 2.6 (Assembly Depth). The assembly depth (ASD)

$$d_s^{(N_k,b)}(C_k) := \max \left(d^{(N_l,b)}(C_l), d^{(N_m,b)}(C_m) \right) + 1, \quad (3)$$

where $d_0^{(1,b)}(c) := 0$, and $d^{(N_l,b)}(C_l)$ and $d^{(N_m,b)}(C_m)$ are the ASDs of two substrings C_l, C_m of the string C_k that were joined in step s . For $N > 3$, and if there are more assembly pathways with different depths w_j leading to a string, which happens if at least two independent assembly steps are possible, the minimum pathway depth is the ASD of this string. Hence, the ASD captures the notion of an independent assembly step.

The ASD naturally divides the lengths of strings into sections $2^{d-1} < N \leq 2^d$ (cf. Figures 4 and 5).

Theorem 2.2. If an assembly space Ω contains strings having the same (non-zero) ASD they were assembled in independent assembly steps.

In other words, if two strings C_l, C_m in Ω have the same ASD, their assembly pathways are unrelated to each other; by the defining equation (3), neither of them could have been used in the assembly pathway of the other.

Corollary 2.1. *If ASI and ASD of a string are equal to each other, an assembly space of this string cannot contain independent assembly steps.*

Lemma 2.2. *For all b the maximum length N of any string that can be assembled with the ASD $d_s^{(N)}$ (3) satisfies*

$$N \leq 2^{d_s^{(N)}}. \quad (4)$$

Theorem 2.3. *For all b the minimum ASD (minASD) as a function of a string length N , is given by*

$$d_{\min}^{(N)} = \lceil \log_2(N) \rceil, \quad (5)$$

where $\lceil x \rceil$ denotes the ceiling function.

The ASD need not be a monotonically nondecreasing function of the assembly step. For example

$$[11] \quad d_1 = 1; \quad [110] \quad d_2 = 2; \quad [01] \quad d_3 = 1; \quad [00] \quad d_4 = 1; \quad [0001] \quad d_5 = 2; \quad [0001110] \quad d_6 = 3. \quad (6)$$

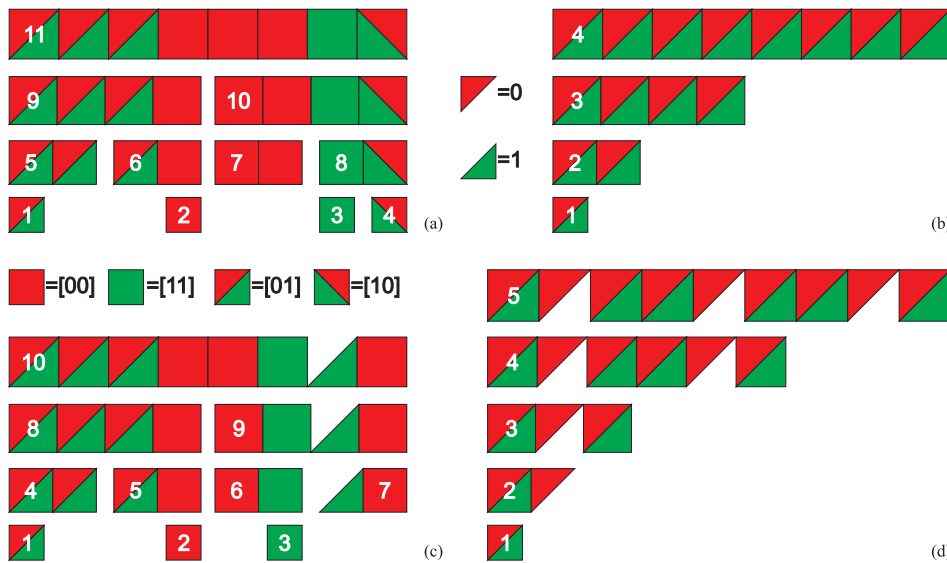


Figure 1. Assembly spaces of bitstrings $C_{\max}^{(N,2)}$ (a, c) and $C_{\min}^{(N,2)}$ (b, d) for $N = 2^s = 16$ (a, b) and $N = 15 \neq 2^s$ (c, d), where the assembly index is a number in a string (final string for (a, c)) and the assembly depth corresponds to the level. For $N = 2^s$, $d_{\max}^{(2^s,b)} = d_{\min}^{(2^s,b)} = s$. In general, for $N \neq 2^s$, the assembly depth $d_{\max}^{(N,b)} < d_{\min}^{(N,b)}$.

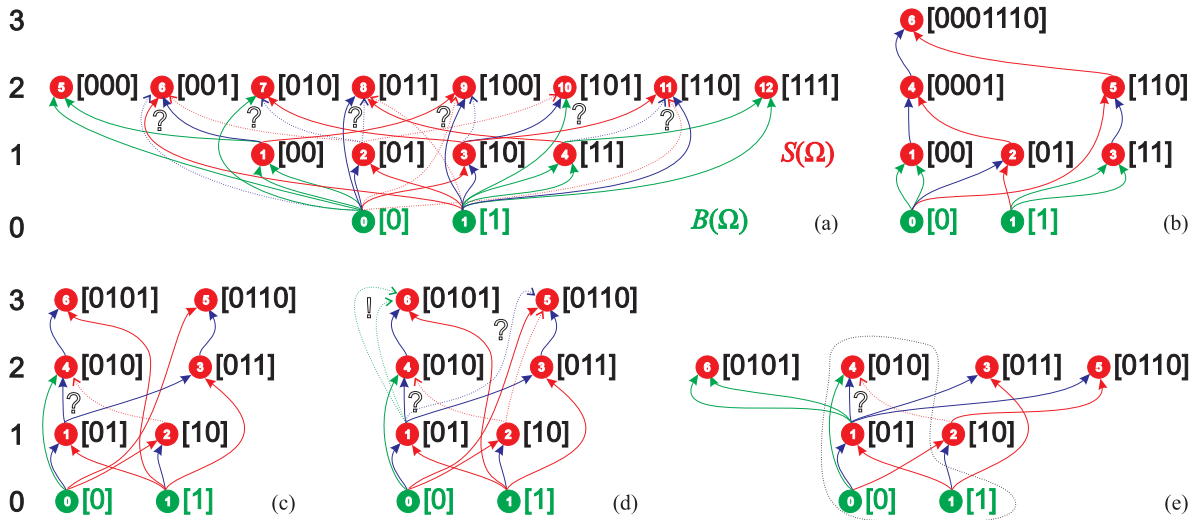


Figure 2. Assembly space Ω , assembly index, and assembly depth. The assembly space of all eight binary 3-grams with all pathways (a). Blue edge provides the 1st string, red edge provides the 2nd string in the assembly step, and the order is irrelevant for two green edges or green edge provides the 1st or 2nd string in dependence of the color of the complementary edge. Dotted edges and question marks indicate alternative pathways. The assembly space of the bitstring $C_6^{(7,2)} = [0001110]$ showing that its assembly index $a^{(7,2)}(C_6) = 6$ (b). The evolution of assembly spaces of strings [0101] and [0110] (c-e). Strings [0101] and [0110] are initially assembled from 3-grams and basic symbols, increasing the assembly depth (c). New pathways increasing the number of independent assembly steps are found (d), and the edges of Ω are reconfigured, decreasing the number of assembly steps of the string [0101] from three to two steps and the assembly depth of both 4-grams from three to two. Five assembly spaces of the bitstrings [0101], [010] (two alternatives, one encircled), [011], and [0110] (e)

We cannot consider the ASD apart from the ASI. For example, the ASD of a string $C_{\max}^{(7,2)} = [0001110]$ is $d_{a_{\max}}^{(7,2)} = \lceil \log_2(7) \rceil = 3$ even though this string can be assembled in six steps with three larger pathway depths $w_6 \in \{4, 5, 6\}$ as

$$\begin{array}{llll}
 [00] & d_1 = 1, & [00] & w_1 = 1, \\
 [01] & d_2 = 1, & [01] & w_2 = 1, \\
 [11] & d_3 = 1, & [11] & w_3 = 1, \\
 [110] & d_4 = 2, & [0001] & w_4 = 2, \\
 [0001] & d_5 = 2, & [000111] & w_5 = 3, \\
 [0001110] & d_6 = 3, & [0001110] & w_6 = 4,
 \end{array}
 \quad
 \begin{array}{llll}
 [00] & w_1 = 1, & [00] & w_1 = 1, \\
 [01] & w_2 = 1, & [01] & w_2 = 1, \\
 [11] & w_3 = 1, & [0001] & w_3 = 2, \\
 [0001] & w_4 = 2, & [000111] & w_4 = 3, \\
 [000111] & w_5 = 3, & [000111] & w_5 = 4, \\
 [0001110] & w_6 = 4, & [0001110] & w_6 = 5,
 \end{array}
 \quad
 \begin{array}{llll}
 [00] & w_1 = 1, & [00] & w_1 = 1, \\
 [01] & w_2 = 1, & [01] & w_2 = 1, \\
 [11] & w_3 = 1, & [0001] & w_3 = 2, \\
 [0001] & w_4 = 2, & [000111] & w_4 = 3, \\
 [000111] & w_5 = 3, & [000111] & w_5 = 4, \\
 [0001110] & w_6 = 4, & [0001110] & w_6 = 5,
 \end{array}
 \quad
 \begin{array}{llll}
 [00] & w_1 = 1, & [00] & w_1 = 1, \\
 [01] & w_2 = 2, & [01] & w_2 = 1, \\
 [11] & w_3 = 3, & [0001] & w_3 = 3, \\
 [0001] & w_4 = 4, & [000111] & w_4 = 4, \\
 [000111] & w_5 = 5, & [000111] & w_5 = 5, \\
 [0001110] & w_6 = 6, & [0001110] & w_6 = 6.
 \end{array}
 \quad (7)$$

Similarly, the ASD of a string $C_{\max}^{(8,2)} = [00011101]$ is $d_{a_{\max}}^{(8,2)} = \lceil \log_2(8) \rceil = 3$ as

$$\begin{array}{llll}
 [00] & d_1 = 1, & [00] & w_1 = 1, \\
 [01] & d_2 = 1, & [01] & w_2 = 1, \\
 [11] & d_3 = 1, & [11] & w_3 = 1, \\
 [0001] & d_4 = 2, & [0001] & w_4 = 2, \\
 [1101] & d_5 = 2, & [000111] & w_5 = 3, \\
 [00011101] & d_6 = 3, & [00011101] & w_6 = 4,
 \end{array}
 \quad
 \begin{array}{llll}
 [00] & w_1 = 1, & [00] & w_1 = 1, \\
 [01] & w_2 = 1, & [01] & w_2 = 1, \\
 [11] & w_3 = 1, & [0001] & w_3 = 2, \\
 [0001] & w_4 = 2, & [000111] & w_4 = 3, \\
 [000111] & w_5 = 3, & [000111] & w_5 = 4, \\
 [00011101] & w_6 = 4, & [00011101] & w_6 = 5,
 \end{array}
 \quad
 \begin{array}{llll}
 [00] & w_1 = 1, & [00] & w_1 = 1, \\
 [01] & w_2 = 1, & [01] & w_2 = 1, \\
 [11] & w_3 = 1, & [0001] & w_3 = 2, \\
 [0001] & w_4 = 2, & [000111] & w_4 = 3, \\
 [000111] & w_5 = 4, & [000111] & w_5 = 4, \\
 [00011101] & w_6 = 5, & [00011101] & w_6 = 5,
 \end{array}
 \quad
 \begin{array}{llll}
 [01] & w_1 = 1, & [01] & w_1 = 1, \\
 [001] & w_2 = 2, & [001] & w_2 = 2, \\
 [0001] & w_3 = 3, & [0001] & w_3 = 3, \\
 [000111] & w_4 = 4, & [000111] & w_4 = 4, \\
 [000111] & w_5 = 5, & [000111] & w_5 = 5, \\
 [00011101] & w_6 = 6, & [00011101] & w_6 = 6.
 \end{array}
 \quad (8)$$

However, the non-maximum and non-minimum ASI string $C_k^{(8,2)} = [01001011]$ has only two 2-grams that can be assembled in independent steps. Hence, its ASD cannot be decreased to $\lceil \log_2(8) \rceil = 3$

$$\begin{array}{ll} [01] & d_1 = 1, & [01] & w_1 = 1, \\ [11] & d_2 = 1, & [010] & w_2 = 2, \\ [010] & d_3 = 2, & [010010] & w_3 = 3, \\ [010010] & d_4 = 3, & [0100101] & w_4 = 4, \\ [01001011] & d_5 = 4, & [01001011] & w_5 = 5. \end{array} \quad (9)$$

In general, the Ω that contains a 2^d -gram having the ASD d can also contain $\{2^{d-1} + 1, 2^{d-1} + 2, \dots, 2^d - 1\}$ -grams having the ASD d and based on the shorter n -grams of length $n < 2^{d-1} + 1$.

Theorem 2.4. For all b the ASD of any maximum ASI string $C_{\max}^{(N,b)}$, corresponds to the minASD (5) of Theorem 2.3, that is

$$d_{a_{\max}}^{(N,b)} = \lceil \log_2(N) \rceil, \quad (10)$$

For example, as shown in Figure 1c,d, the string $C_{\max}^{(15,2)} = [010101000011100]$ has the ASI $a_{\max}^{(15,2)} = 10$ and the ASD $d_{a_{\max}}^{(15,2)} = 4$, while the string $C_{\min}^{(15,2)} = [010010100101001]$ has smaller ASI $a_{\min}^{(15)} = 5$ but larger ASD $d_{a_{\min}}^{(15,2)} = 5$. On the other hand, the ASD of the maxASI string $C_{(N-5)}^{(16,2)} = [0101010000001110]$ and the minASI string $C_{\min}^{(16,2)} = [0101010101010101]$, shown in Figure 1a,b, is the same.

Here, we introduce the following definition, which — as we shall see — is also related to the independent assembly step.

Definition 2.7 (Depth Index). We call the number of steps $\hat{a}_{\min}^{(N)}$ to reach 1 starting from $N := N_0$ and assigning

$$N_{s+1} = \begin{cases} N_s - 1 & \text{if } N_s \text{ is odd,} \\ N_s - 2 & \text{if } N_s = 2^s + 2, s \in \mathbb{N}, \\ N_s / 2 & \text{otherwise} \end{cases} \quad (11)$$

the depth index (DPI).

The relation (11) yields the same number of steps as the Chandah-sutra method (OEIS A014701) and, unlike the minimum ASI, is an analytical function of N . For example, $\hat{a}_{\min}^{(2^s)} = s$ and $\hat{a}_{\min}^{(2^s-1)} = 2(s-1)$.

Applying AT to evolutionary biology, we can assume that initially, a new string of length N is formed in an assembly space based on a basic symbol and a string of length $N-1$. Subsequently, this string assembly space evolves to reduce the cardinality $|S(\Omega_{C_s})|$ of the set of the assembly step vertices until it equals the ASI of this string, that is until $|S(\Omega_{C_s})| = a^{(N,b)}(C_s)$. Hence, the assembly spaces evolve by reconfiguring the network of edges to decrease the ASD of newly assembled strings, possibly finding shorter pathways for these strings, and if only such a decrease would not result in ASI increase (Length $N = 15$ shown in Figure 1d is the shortest length, where $5 = d_{a_{\min}}^{(15)} > \lceil \log_2(15) \rceil = 4$).

The concepts of assembly space, string assembly space, ASI, and ASD, as well as the evolution of assembly spaces, are illustrated in Figure 2. Although the alternative pathways shown in Figure 2a demonstrate that the mixed 2-gram $[01]$ is not required to construct all 3-grams in two steps, this 2-gram is necessary to achieve the ASI of longer strings (Cf. a constrained AT* version shown in list (38) discusses in Section 6.).

3. Minimum Complexity Strings of AT

In this section, we consider factors relevant to the minimum string complexity in the context of AT. The Theorem 3.1 was already stated in our previous study [10]; we restate it here for clarity.

Theorem 3.1. For all b the minimum ASI ($\min\text{ASI}$) $a^{(N)}(C_{\min})$ as a function of N corresponds to the shortest addition chain for N (OEIS A003313).

Any shortest addition chain for n starts with one, not zero, as zero is the neutral element of addition. For the same reason, two is considered the smallest prime, as one is the neutral element of multiplication. Hence, the fundamental theorem of arithmetic can be thought of as the shortest multiplication chain for N . It is conjectured that the largest prime factor must appear in some shortest addition chain for N [35].

Theorem 3.2. The strings $C_{\min}^{(2^s, b)}$ can contain at most two distinct symbols if $b > 1$. Other minASI strings of length $N \neq 2^s$ can contain at most three distinct symbols if $b > 2$.

Proof. Minimum ASI strings of length $N = 2^s$ are formed by joining the newly assembled string to itself, where a clear or mixed 2-gram is assembled in the first step. Minimum ASI strings of other lengths admit a 2-gram and a 3-gram containing this 2-gram and an additional basic symbol.

To formally prove the first part, we can also use mathematical induction on the assembly step s . If $s = 1$, then the minASI strings $C_{\min}^{(2, b)}$ are 2-grams of the form $[c_1 c_2]$, where $c_1, c_2 \in B(\Omega)$. If $c_1 = c_2$, the string contains one distinct symbol, and if $c_1 \neq c_2$, the string contains two distinct symbols. In both cases, the string has a form (A13) and the number of distinct symbols does not exceed two. Now assume that for some $k \in \mathbb{N}$, all minASI strings $C_{\min}^{(2^k, b)}$ contain at most two distinct symbols. We must show that $C_{\min}^{(2^{k+1}, b)}$ also contains at most two distinct symbols. We construct $C_{\min}^{(2^{k+1}, b)}$ by joining two identical minASI strings $C_{\min}^{(2^k, b)}$

$$C_{\min}^{(2^k, b)} \circ C_{\min}^{(2^k, b)} = C_{\min}^{(2^{k+1}, b)}, \quad (12)$$

with each other. By the inductive hypothesis, each $C_{\min}^{(2^k, b)}$ contains at most two distinct symbols. Therefore, their concatenation also contains at most two distinct symbols. By induction, for all $s \in \mathbb{N}$, the minASI string $C_{\min}^{(2^s, b)}$ contains at most two distinct symbols.

We will now show that other minASI strings of length $N \neq 2^s$ can contain at most three distinct symbols if $b > 2$. We provide the construction of minASI strings with three symbols. In the first step $s = 1$, we assemble a 2-gram $[c_1 c_2]$ where $c_1, c_2 \in B(\Omega)$ and $c_1 \neq c_2$. Next, we join the existing 2-gram $[c_1 c_2]$ with a new symbol $c_3 \in B(\Omega)$ where $c_3 \notin \{c_1, c_2\}$. This forms a 3-gram $[c_1 c_2 c_3]$, introducing a third distinct symbol and further increasing the ASI by 1. We continue assembling by joining the longest string formed so far with itself or with previously formed strings, maintaining the minimal ASI increase.

Assume *a contrario* that there exists a minASI string $C_{\min}^{(N, b)}$ of length $N \neq 2^s$ that contains four or more distinct symbols. But, incorporating such a fourth symbol is equivalent to assembling a maxASI 4-gram, which contradicts the minimality of $C_{\min}^{(N, b)}$ (only a 2-gram must be assembled from basic symbols and a 3-gram must be assembled from a basic symbol and a 2-gram). Thus, Theorem 3.2 is proven. $\square$

By Theorem 2.1 the maximum ASI decrease is provided by minASI 2^s -grams and amounts to $k(2^s - 1) - s$. On the other hand, 1-symbol n -grams are guaranteed to have the minimum ASI for all n . Therefore, the maxASI string will tend to contain only one such long n -gram for each b . We have found, for example, that the sum of the lengths of an n_0 -gram of zeros and an n_1 -gram of ones should preferably be greater than or equal to $N/2$ of a maxASI string.

The assembly spaces of strings $a_{\min}^{(N)}$ of length $N \neq 2^s$ are not unique [36]. For example, a string $C_{\min}^{(5, b)} = [01010]$ can be assembled in three steps from four assembly spaces with $S(\Omega) = \{[01], [010]\}$, $S(\Omega) = \{[01], [0101]\}$, $S(\Omega) = \{[10], [010]\}$, or $S(\Omega) = \{[10], [1010]\}$.

Strings with non-minASI can contain all symbols. For example, the string [37]

$$C_k = [01234012340123401234], \quad (13)$$

Lemma 3.1. *The minASI strings [10] (strings (15)) of lengths*

$$\tilde{N}_3 := 2^{d-1} + 3 \cdot 2^l, \quad l = 0, 1, \dots, d-3, \quad d \geq 3 \quad \Leftrightarrow \quad a_{\min}^{(\tilde{N}_3)} = d+1 = \lceil \log_2(\tilde{N}_3) \rceil + 1, \quad (16)$$

have only one independent assembly step in their assembly spaces, and excluding this step, they are assembled by joining the longest string assembled so far with itself. Therefore, their ASI is one greater than the minASD (5).

Lemma 3.2. *The minASI strings of lengths*

$$\tilde{N}_5 := 2^{d-1} + 5 \cdot 2^l, \quad l = 0, 1, \dots, d-4, \quad d \geq 4 \quad \Leftrightarrow \quad a_{\min}^{(\tilde{N}_5)} = d+1 = \lceil \log_2(\tilde{N}_5) \rceil + 1, \quad (17)$$

have only one independent assembly step in their assembly spaces, and excluding this step, they are assembled by joining the longest string assembled so far with itself. Therefore, their ASI is one greater than the minASD (5).

Lemma 3.3. *The minASI strings of lengths*

$$\tilde{N}_9 := 2^{d-1} + 9 \cdot 2^l, \quad l = 0, 1, \dots, d-5, \quad d \geq 5 \quad \Leftrightarrow \quad a_{\min}^{(\tilde{N}_9)} = d+1 = \lceil \log_2(\tilde{N}_9) \rceil + 1, \quad (18)$$

have only one independent assembly step in their assembly spaces, and excluding this step, they are assembled by joining the longest string assembled so far with itself. Therefore, their ASI is one greater than the minASD (5).

Lemmas 3.1–3.3 allow for the following generalization.

Theorem 3.5. *The minASI strings of lengths*

$$\tilde{N}_{2^n+1} := 2^{d-1} + (2^{k-4} + 1)2^l, \quad k \geq 5, d \geq k-2, l = 0, 1, \dots, d-(k-2), \quad \Leftrightarrow \quad a_{\min}^{(\tilde{N}_{2^n+1})} = d+1 = \lceil \log_2(\tilde{N}_{2^n+1}) \rceil + 1, \quad (19)$$

have only one independent assembly step in their assembly spaces, and excluding this step, they are assembled by joining the longest string assembled so far with itself. Therefore, their ASI is one greater than the minASD (5).

Proof. The lengths of the strings (19) are listed in rows in Table A1 starting after the length of the substring assembled in an independent assembly step marked green. Hence, the first row contains the lengths of strings of Theorem 3.1 shown on the diagonal of Figure 3b, and so on. $\square$

Theorem 3.6. *The minASI strings [10] of lengths*

$$\tilde{N}_7 := 2^{d-1} + 7 \cdot 2^{d-4} \in \{15, 30, 60, \dots\}, \quad d \geq 4 \quad \Leftrightarrow \quad a_{\min}^{(\tilde{N}_7)} = d_{\min}^{(\tilde{N}_7)} = d_{\min}^{(\tilde{N}_7)} + 1 = \hat{a}_{\min}^{(\tilde{N}_1)} - 1 = \lceil \log_2(\tilde{N}_7) \rceil + 1, \quad (20)$$

are assembled by joining the longest string assembled so far with itself. Their ASI and ASD are the same, one greater than the minASD (5) and one smaller than the DPI.

Theorem 3.6 seems to allow for the following generalization, which we have validated numerically based on the sequence OEIS A003313 for $N \leq 10^5$.

Conjecture 3.1. *For d , l , and $\tilde{N}_{2^n+1}$ defined by the relation (19), the following holds*

$$\tilde{N}_{2^n+1,a} := \tilde{N}_{2^n+1} + 2^d = 3 \cdot 2^{d-1} + (2^{k-4} + 1)2^l \wedge k = 5 \quad \Leftrightarrow \quad a_{\min}^{(\tilde{N}_{2^n+1,a})} = d+2 = \lceil \log_2(\tilde{N}_{2^n+1,a}) \rceil + 1, \quad (21a)$$

$$\tilde{N}_{2^n+1,b} := \tilde{N}_{2^n+1} + 2^{d+1} = 5 \cdot 2^{d-1} + (2^{k-4} + 1)2^l \wedge k \in \{5, 6\} \quad \Leftrightarrow \quad a_{\min}^{(\tilde{N}_{2^n+1,b})} = d+3 = \lceil \log_2(\tilde{N}_{2^n+1,b}) \rceil + 1. \quad (21b)$$

The lengths of the strings (21a) and (21b) are listed in rows in Table A1. Furthermore, we have numerically validated the following conjecture.

Conjecture 3.2. *The minASI strings of lengths*

$$\tilde{N}_{15} := 2^{d-1} + 15 \cdot 2^l, \quad l = 0, 1, \dots, d-5, \quad d \geq 5, \quad (22a)$$

$$\tilde{N}_{27} := 2^{d-1} + 27 \cdot 2^l, \quad l = 0, 1, \dots, d-6, \quad d \geq 6, \quad (22b)$$

$$\tilde{N}_{50.9} := 50 \cdot 2^{d-6} + 9 \cdot 2^l, \quad l = 0, 1, \dots, d-6, \quad d \geq 6, \quad (22c)$$

have the property of

$$a_{min}^{(\tilde{N}_*)} = d + 2 = \lceil \log_2(\tilde{N}_*) \rceil + 2. \quad (22d)$$

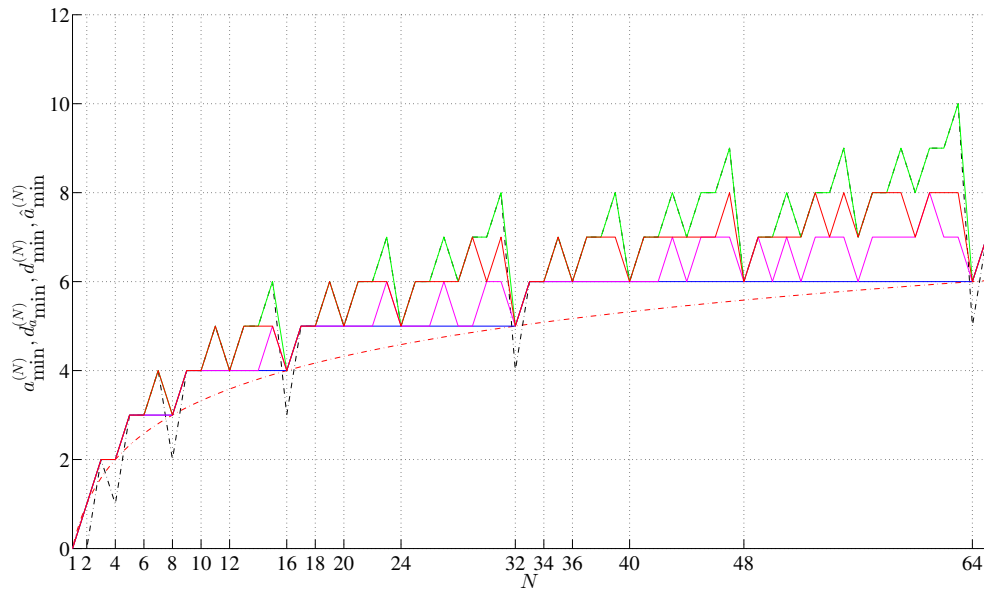


Figure 4. The minimum assembly depth ($\lfloor \log_2(N) \rfloor$, blue), the assembly depth of the minimum assembly index string (magenta), the minimum assembly index (OEIS A003313, red; $\log_2(N)$, red, dash-dot), depth index (OEIS A014701, green), and $\lfloor \log_2(N) \rfloor + N_1 - 2$, where N_1 is the Hamming weight of the binary representation of N (black) for $1 < N \leq 65$.

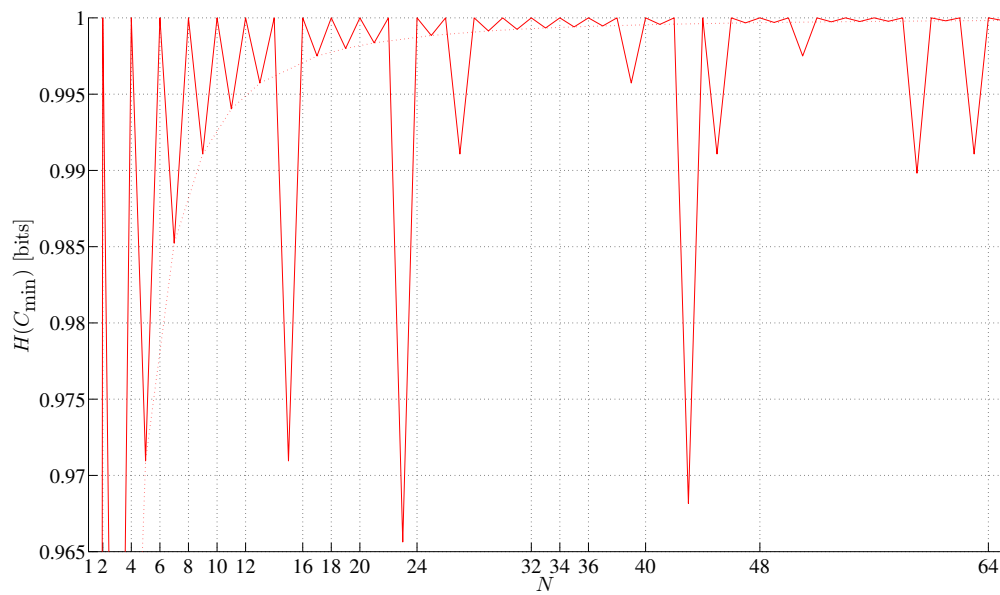


Figure 5. The Shannon entropy of the most balanced bitstrings having the minimum assembly index for $1 < N \leq 65$

Strings of lengths (14), (16), and (20), revealed in [10] based on the degree of causation, showed that there are certain regularities among the minASI strings. Here, we extended these results to strings of lengths (19), (21), and (22). In general, Theorems 3.4–3.6 (in particular Theorem 3.6) and Conjectures 3.2 and 3.1 show a peculiar interdependence among the minASD (5), minASI ASD, minASI, and DPI, as shown in Figure 4. In particular, they show that

- the Ω of minASI strings having ASI equal to DPI cannot contain strings assembled in independent assembly steps,
- the Ω s of other minASI strings can contain at least two such strings, and therefore
- the assembly space of a maxASI string will tend to maximize the number of strings assembled in independent assembly steps in the Ω , taking into account the saturation of the Ω as it cannot contain more than b^n distinct n -grams, and hence to minimize the possible ASD.

We note that, in general, the difference between the DPI and minASI is larger than 1.

Lemma 3.4. *The minASI bitstrings assembled along the pathway given by the DPI 2.7 and beginning with $C_{\min}^{(2)} = [* \star]$ are balanced bitstrings if N is even or nearly balanced bitstrings ($N_0 = N_1 + 1$) if N is odd.*

In other words, the Definition 2.7 removes the *imbalance propagation*. For example, an imbalanced pathway $\{2, 4, 5, 10, 20\}$ ($N_1 = 8$) becomes a balanced pathway $\{2, 4, 8, 10, 20\}$ ($N_1 = 10 = N/2$) and $2^s + 2 \rightarrow 2^{s-1} + 1 \rightarrow 2^{s-1} \Leftrightarrow 2^s + 2 \rightarrow 2^s \rightarrow 2^{s-1}$. We have also examined the Shannon entropy

$$H(C_{\min}^{(N)}) = -p_0 \log_2(p_0) - p_1 \log_2(p_1), \quad (23)$$

of the most balanced minASI bitstrings, where $p_0 = N_0/N$ and $p_1 = N_1/N$ are fractions of the respective symbols $\{0, 1\}$ within the string (N_1 is the Hamming weight). Although by Theorem 3.1, the minASI as a function of the length of a string does not depend on b , we have chosen the most balanced bitstrings, as they maximize the Shannon entropy (23). This choice is also supported by physics [38–40]. For the same reason, we preferred the pathway $\{2, 3, 5, 10, 15\}$ (cf. Figure 1d) over $\{2, 3, 6, 12, 15\}$, for example, as the string assembled using the former pathway is more balanced ($N_1 = 6$) than the one assembled using the latter one ($N_1 = 5$). Similarly, we preferred the pathway that provided a more balanced string over the one that provided independent assembly steps. $N = 14$ is the first exception. $C_{\min}^{(14)}$ assembled in five steps along the pathway $\{2, (3, 4), 7, 14\}$ with the independent assembly steps 3 and 4 has the hamming weight $N_1 = 6$ as compared to $C_{\min}^{(14)}$ assembled in five steps along the pathway $\{2, 4, 8, 12, 14\}$ with no independent assembly steps and the hamming weight $N_1 = N/2 = 7$. The resulting pathways of the minASI strings maximizing the number of independent assembly steps or the binary Shannon entropy (23) are listed in Table A2 for $N \leq 65$. As shown in Figure 5, the Shannon entropy (23) of the most balanced minASI bitstrings rapidly converges to one with exceptions for lengths $N \in \{15, 23, 27, 39, 43, 45, 51, 59, 63, \dots\}$ substantially corresponding to lengths at which DPI is larger than the minASI (cf. Figure 4), which highlights the interdependence among the minASI and DPI.

We demonstrated in this section that minimum complexity AT strings cannot be fully derived by any rule-based system. The same is true for the maxASI strings, as we shall show in a subsequent section.

4. Maximum Assembly Index Strings of AT

A seven-bit string is the longest string that can have the maxASI $a_{\max}^{(7,2)} = N - 1 = 6$. It contains two clear 3-grams

$$C_{(N-1)}^{(7,2)} = [0001110], \quad (24)$$

and its length cannot be increased without a repetition of a 2-gram, which keeps the ASI at the same level $a_{\max}^{(8,2)} = 8 - 2 = 6$. This led us to seek a general method to construct the longest possible string

having the maxASI $a_{\max}^{(N,b)}(C_{(N-1)}) = N - 1$, as a function of the radix b . We denote the length of this string by $N_{(N-1)}$ and we call this string a $C_{(N-1)}$ string. We developed two methods of constructing a string $C_{(N-1)}$ of the form

$$C_{(N-1)} = [0001112 \dots b_{-2}b_{-1}b_{-1}b_{-1} \dots 0]. \quad (25)$$

In both methods, we start with an initial balanced string of length $3b$ containing b clear 3-grams ordered as

$$[0001112 \dots b_{-2}b_{-1}b_{-1}b_{-1}]. \quad (26)$$

The 2-grams that can be inserted into the initial string (26) can be arranged in a $b \times b$ matrix

$$\begin{bmatrix} \cancel{00} & \cancel{01} & 02 & \dots & 0b_{-1} \\ 10 & \cancel{11} & \cancel{12} & \dots & 1b_{-1} \\ 20 & 21 & \cancel{22} & \dots & 2b_{-1} \\ \dots & \dots & \dots & \dots & \dots \\ b_{-2}0 & b_{-2}1 & b_{-2}2 & \dots & \cancel{b_{-2}b_{-1}} \\ b_{-1}0 & b_{-1}1 & b_{-1}2 & \dots & \cancel{b_{-1}b_{-1}} \end{bmatrix}, \quad (27)$$

where the crossed out entries on a diagonal cannot be reused, as they would form repetitions in this string. Due to the order of 3-grams in the string (26) we can also cross out the entries in the first superdiagonal of the matrix (27). By construction, the starting string (26) has length $3b$ and contains only $b + (b - 1) = 2b - 1$ 2-grams $[il]$ and $[m(m + 1)]$, where $0 \leq l \leq b_{-1}$ and $0 \leq m \leq b_{-2}$.

In the 1st step of the 1st method, we begin by forming a string containing 2-grams on the first subdiagonal of the matrix (27), starting with 10

$$[102132 \dots b_{-2}b_{-3}b_{-1}b_{-2}], \quad (28)$$

and we append it to the string (26). With this step, we also eliminate the 2-grams on the second superdiagonal starting with the 2-gram 02, as well as the 2-gram $b_{-1}1$. In the 2nd step, we form a string containing 2-grams on the third superdiagonal beginning with the 2-gram 03

$$[0314 \dots b_{-5}b_{-2}b_{-4}b_{-1}], \quad (29)$$

and append it to the string formed so far. With this step, we also remove the 2-gram $b_{-2}0$ and the middle part of the second subdiagonal containing $\{31, 42, \dots, b_{-2}b_{-4}\}$. And so on. Finally, we append 0 if b is even. This process is illustrated in Figure 6 and generates the following $C_{(N-1)}^{(N,b)}$ strings

$$\begin{aligned} C_{(N-1)}^{(3,1)} &= [000], \\ C_{(N-1)}^{(7,2)} &= [000111|0], \\ C_{(N-1)}^{(13,3)} &= [000111222|10|20], \\ C_{(N-1)}^{(21,4)} &= [000111222333|102132|03|0], \\ C_{(N-1)}^{(31,5)} &= [000111222333444|10213243|0314|20|40], \\ C_{(N-1)}^{(43,6)} &= [000111222333444555|1021324354|031425|0415|2053|0], \\ C_{(N-1)}^{(57,7)} &= [000111222333444555666|102132435465|03142536|041526|2064|0516|30], \\ C_{(N-1)}^{(73,8)} &= [000111222333444555666777|10213243546576|0314253647|04152637|2075|051627|306174|0], \\ C_{(N-1)}^{(91,9)} &= [\dots|1021324354657687|031425364758|0415263748|2086|05162738|30617285|0718|40], \\ &\dots \end{aligned} \quad (30)$$

We discuss the 2nd method in Appendix A.15.

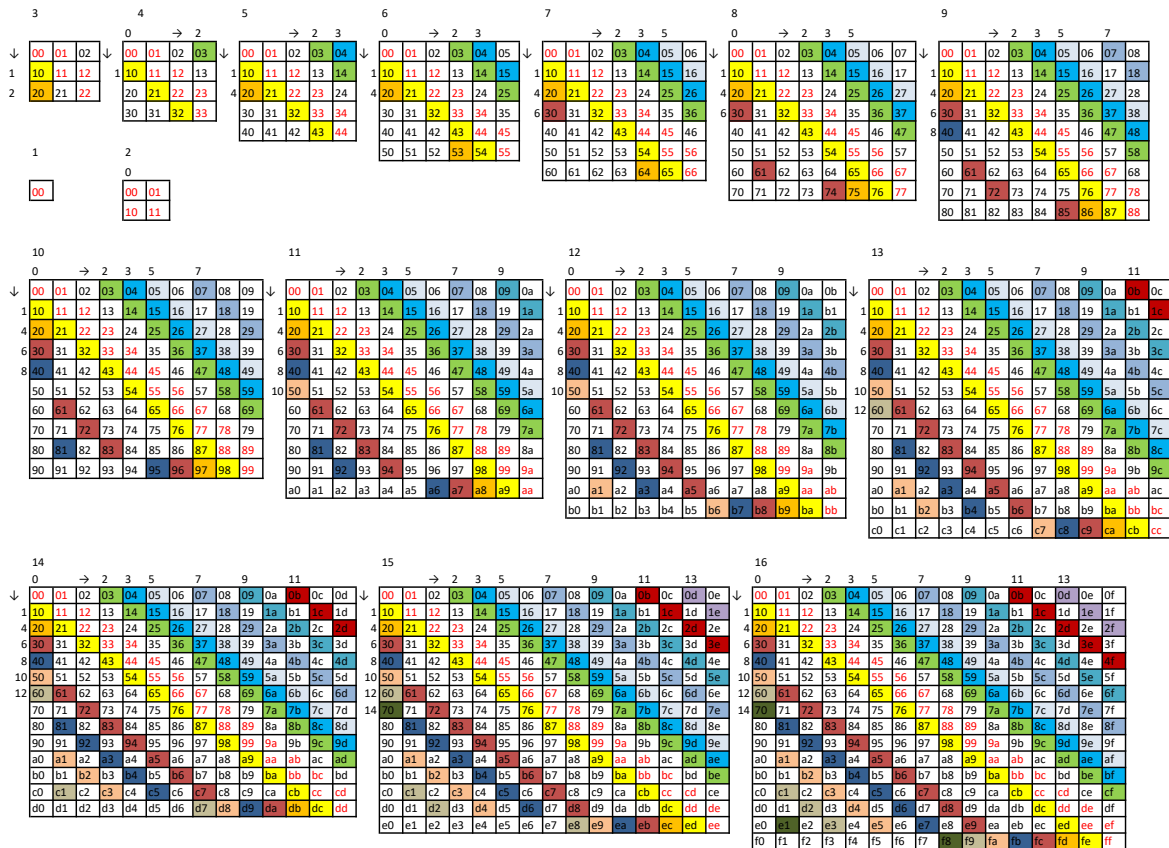


Figure 6. 2-gram matrices for $1 \leq b \leq 16$ that illustrate the generation of $C_{(N-1)}$ strings according to a method presented in this section. Coloured 2-grams are appended to the initial string of clear 3-grams in the order indicated by arrows starting from the 1st column or row. Finally, 0 is appended at the end if b is even.

The longest length of a string that has the ASI of $N - 1$ (a $C_{(N-1)}$ string) is given by

$$N_{(N-1)} = 3b + (b - 1)^2 = bN_c + 1 = b^2 + b + 1 \quad (31)$$

(OEIS A353887), where $N_c = b + 1$ is the number of occurrences of all but one symbol within the string, so this string has odd length and is nearly balanced. To be the longest, a $C_{(N-1)}$ string must contain all 2-grams from the matrix (27) and all clear 3-grams. A string that contains all k -grams is a linear, minimal de Bruijn string (e.g., $\{[00110], [01100], \dots\}$ for $b = k = 2$), which must start and end with the same symbol. Its length is given by $N_{\text{de Bruijn}} = b^k + k - 1$, and there are $(b!)^{b^{k-1}}$ such strings. We consider $k = 2$ and have to add b symbols to turn b clear 2-grams of a de Bruijn string into b clear 3-grams, which clearly does not introduce any disjoint occurrences [41]. This leads to the formula (31). For $b = 1$ the upper and the lower bound on the ASI are the same, $a_{\max}^{(N,1)} = a_{\min}^{(N)}$, and this is the only case where the maxASI is not a monotonically non-decreasing function of N . Otherwise, the maxASI must rise. If it were to become constant for $N > \hat{N}_{\max}$, then at some even larger N it would inevitably become lower than the minASI bound of Theorem 3.1, which also rises, and this would be a contradiction.

Subsequently, we considered other strings with the maxASI. A $C_{(N-1)}$ string contains all 2-grams. Hence, inserting any basic symbol into any position inevitably leads to a repetition of a 2-gram. Without loss of generality (w.l.o.g.) we append it at the start of the $C_{(N-1)}$ string, obtaining a string $[*000111\dots]$. Another symbol can be introduced to this string without an additional 2-gram repetition, for example, as another prefix $[\star * 000111\dots]$ leading to the repetition of the 2-gram $\star *$ or $*0$ but not both of them (here we allow $\star = *$). Hence, both the length and the ASI of this string increase by one.

Finally, 0 can be appended at the start of this string without an additional 2-gram repetition, provided that $\star = 1$ and $\star = 0$. For $b = 2$ there are only four¹ bitstrings containing only two copies of a 2-gram having the ASI $a_{\max}^{(10,2)} = 8$

$$[01|0000|1110], \quad [1|0000|11101], \quad [101|0000|111], \quad [1|0000|10111], \quad (32)$$

where the leftmost one is the only unbalanced string within this set ($N_0 \neq N_1$). Each 10-bit string (32) has one clear 4-gram surrounded by the same symbol ($[\dots 1|0000|1\dots]$) and one clear 3-gram ($[\dots 111\dots]$), which can be concatenated ($[\dots 1|0000|111\dots]$) in eight bits or separated ($[\dots 1|0000|1\dots 111\dots]$) in nine bits. The first three strings (32) correspond to the former case, while the last one to the latter. Each of the strings (32) contains three pairs of mutually exclusive 2-grams, so that only one pair can be reused to decrease the maximum $N - 1$ ASI by one.

We developed two methods of constructing generalized $C_{(N-b)}$ strings (32) for $b > 2$. They have $b - 1$ clear 4-grams ($c_l c_l c_l c_l$), one clear 3-gram and $(b - 1) \times 3$ of mutually exclusive 2-grams $\{c_k c_l, c_l c_l, c_l c_m\}$ overlapped in patterns $[\dots c_k | c_l c_l c_l c_l | c_k \dots]$ surrounding clear 4-gram, so that only one pair from each pattern can be reused in 3^{b-1} ways to reduce the maxASI to $N - b$. The 1st method is based on the balanced $C_{(N-2)}$ string (32) $[101|0000|111]$ and generates the most balanced $C_{(N-b)}$ strings. It starts with a string of length $6b - 2$ containing all $b - 1$ clear 4-grams in the form

$$[b_{-1}01|0000|12|1111|2\dots b_{-1}|b_{-2}b_{-2}b_{-2}b_{-2}|b_{-1}b_{-1}b_{-1}], \quad (33)$$

to which the missing $b^2 - 2b$ symbols are appended using the patterns shown in Figure 7 yielding the $C_{(N-b)}$ strings of the form

$$\begin{aligned} C_{(N-1)}^{(3,1)} &= [000], \\ C_{(N-2)}^{(10,2)} &= [101|0000|111], \\ C_{(N-3)}^{(19,3)} &= [201|0000|12|1111|222|102], \\ C_{(N-4)}^{(30,4)} &= [301|0000|12|1111|23|2222|333|103|202|13], \\ C_{(N-5)}^{(43,5)} &= [401|0000|12|1111|23|2222|34|3333|444|104|30313202|1424], \\ C_{(N-6)}^{(58,6)} &= [501|0000|12|1111|23|2222|34|3333|45|4444|555|105|404142430313202|152535], \\ C_{(N-7)}^{(75,7)} &= [601|0000|12|1111|23|2222|34|3333|45|4444|56|5555|666|106|505152535404142430313202|16263646], \\ C_{(N-8)}^{(94,8)} &= [701|0000|12|1111|23|2222|34|3333|45|4444|56|5555|67|6666|777| \\ &\quad 107|60616263646505152535404142430313202|1727374757], \\ &\quad \dots \\ C_{(N-b)}^{(b^2+4b-2,b)} &= [b_{-1}01|0000|12|1111|2\dots b_{-1}|b_{-2}b_{-2}b_{-2}b_{-2}|b_{-1}b_{-1}b_{-1}|10b_{-1}|b_{-2}0b_{-2}\dots 21\dots b_{-1}]. \end{aligned} \quad (34)$$

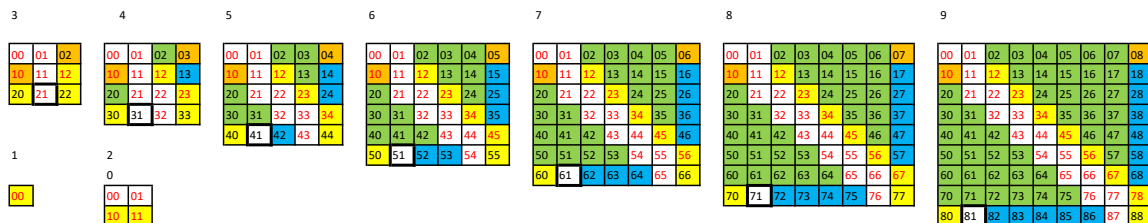


Figure 7. 2-gram matrices for $1 \leq b \leq 9$ that illustrate the generation of the most balanced $C_{(N-b)}$ strings (33). Coloured 2-grams are appended to the initial string of clear 4-grams in the order indicated by colours.

We discuss the 2nd method, generating non-balanced $C_{(N-b)}$ strings, in Appendix A.16.

¹ Sixteen if we relax the Definition 2.5 (cf. Figure 9b).

The mutually exclusive 2-grams of any $C_{(N-b)}$ string comprise $b - 1$ adjoining clear 2-grams on the diagonal of the 2-grams matrix (27) and $2(b - 1)$ 2-grams on its sub- and super-diagonals, as shown in Figure 7, so that each clear 4-gram is surrounded by the same symbol only one pair from each pattern can be reused. In particular, $C_{(N-b)}$ strings can have forms (A25) or (34) with all clear 4-grams separated by 2-grams and forming 7-gram patterns in the form $[\dots c_l c_k c_l c_l c_l c_l c_k \dots]$. However, this is not necessary. For example, the string $C_{(N-3)} = [1|0000|102|1111|21|222|01]$ does not follow this rule. $C_{(N-b)}$ strings are not the maximum length maxASI strings having the property of maxASI incremented by one with every three symbols. For example, the string (34) for $b = 3$ can be suffixed with 202, yielding a string with $a^{(22,3)} = 18$ which still is not the maxASI string with this property for $b = 3$ (cf. Table A5).

5. Results Common to the Minimum and Maximum Complexity Strings

The minASI and maxASI bounds are illustrated in Figure 8 for $b = \{2, 3, 4\}$. For each b , maxASI initially equals $N - 1$ up to the 1st threshold (31). Then it flattens and decreases by one every 3 symbols: once for $b = 2$ and $2b$ times for $b > 2$, as we conjecture based on the results for $b = 3$.

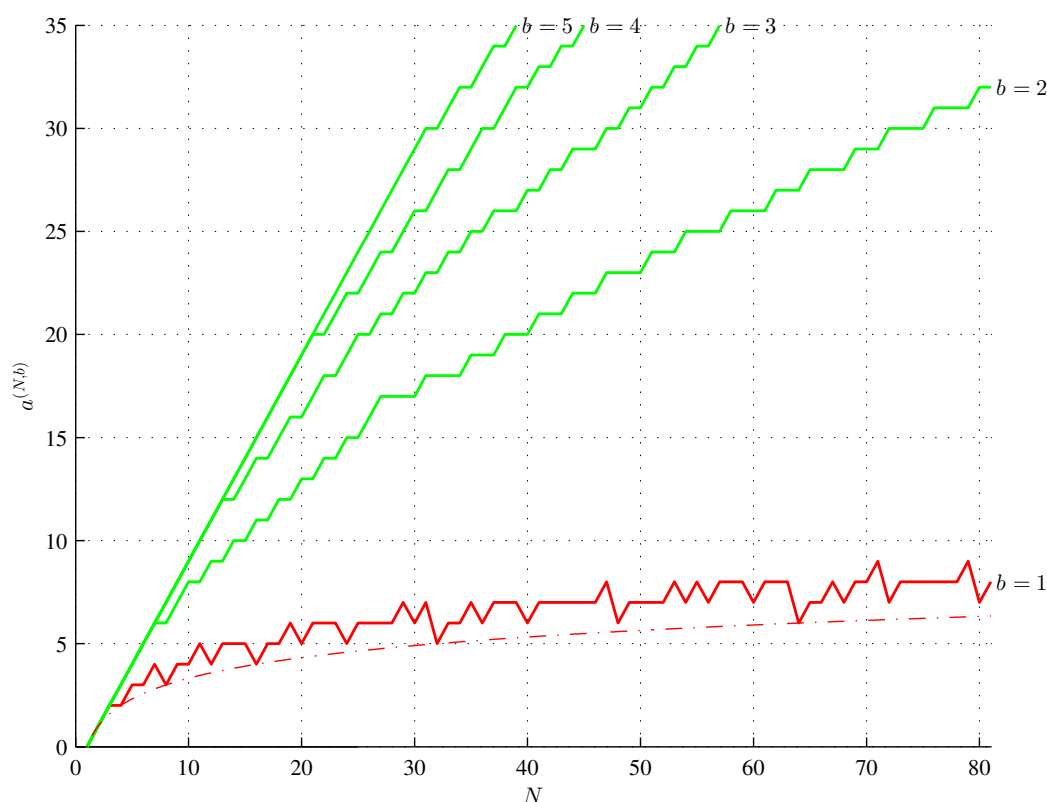


Figure 8. The minimum assembly index (red; $\log_2(N)$, red, dash-dot) and the maximum assembly index (green) for $b \in \{2, 3, 4, 5\}$ and $N \leq 81$.

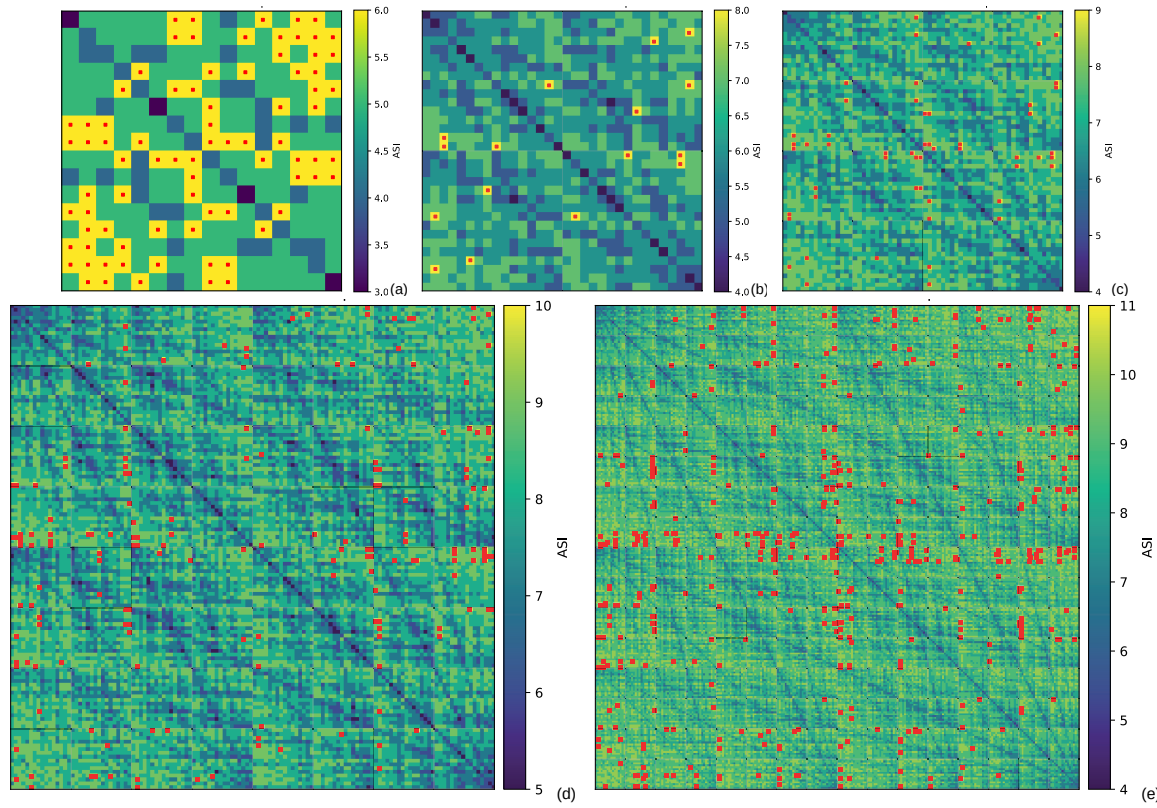


Figure 9. The ASI of all bitstrings of lengths $N = 8$ (a), $N = 10$ (b), $N = 12$ (c), $N = 14$ (d), and $N = 16$ (e), as a function of their decimal representations, in squares with sides equal to $\sqrt{2^N}$. Red dots indicate maxASI strings.

The ASI of a string is also related to the expected waiting time (EWT) of this string. The EWT is the expected number of trials until a specific string appears in a sequence of independent and identically distributed (i.i.d.) random variables taking values in a finite set of $b > 1$ equiprobable outcomes, called a uniform categorical distribution. The general formula for the EWT of a string, which can be determined using Markov chains, is [42]

$$E[C_l^{(N,b)}] = \sum_{k=1}^N b^k \cdot I_k, \quad (35)$$

where I_k is an indicator function that is 1 if the prefix of length k is also the suffix of the string $C_l^{(N,b)}$ and 0 otherwise. Thus, the EWT is bounded from below and from above as

$$b^N \leq E[C_l^{(N,b)}] \leq \frac{b}{b-1} (b^N - 1). \quad (36)$$

The upper EWT bound (36) is attained by clear strings, that is, also by the minASI strings, in which case $I_k = 1, \forall k$ and the formula (35) simplifies to a geometric series $E[C_l^{(N,b)}] = \sum_{k=1}^N b^k$ which resolves to the upper EWT bound (36). The minASI strings containing two or three symbols do not satisfy this relation and hence have shorter EWT. For example, the minASI alternating strings have the EWT given by $E[C_l^{(N,b)}] = \sum_{m=1}^{N/2} b^{2m}$ if N is even and $E[C_l^{(N,b)}] = \sum_{m=1}^{\lceil N/2 \rceil} b^{2m-1}$ if N is odd, that is $E[C_k^{(N,b)}] \in \{2, 4, 10, 20, 42, 84, \dots\}$ given by $\{\times 2, \times 2 + 2, \times 2, \dots\}$ (OEIS A026644). The non-alternating minASI bitstrings can have the EWT even closer to the lower bound (36). For example, $E[010010100101001] = 33828$ (cf. Figure 1d) lies firmly at the lower end of the spectrum, being positioned just 3.24% of the way through the total range. The lower EWT bound (36) is attained by the strings having no prefixes matching suffixes, in which case, for $k = N$ the whole string matches itself. The maxASI strings feature the EWT values equal to or close to this bound.

Applying AT to evolutionary biology, the inverse proportionality between the EWT of a string and its ASI is anticipated but also feels counterintuitive, as common sense would suggest that strings of low complexity should be *provided earlier* than more complex ones.

We found a correlation between the maxASI and the decimal representation of a string providing an intuitive geometric interpretation of the complexity landscape. For example, Figure 9 shows the ASI of all bitstrings of even lengths $N \in \{8, 10, 12, 14, 16\}$ distributed in squares of sides equal to $2^{N/2}$, starting from the top-left corner ($\text{val}_2[00\dots] = 0$) and ending at the bottom-right corner ($\text{val}_2[11\dots] = 2^N - 1$). The decimal value of each pixel is given by $r2^{N/2} + c$, where $0 \leq r, c \leq 2^{N/2} - 1$ are the row and column indices. The low ASI strings naturally occur on diagonals and oblique lines because of the internal symmetries they exhibit. For example, minASI strings naturally occur on the main diagonal, as $r(2^{N/2} + 1)$ for $r = c$ corresponds to a bitstring containing a prefix of length $N/2$ equal to its suffix. Thus, the EWT (35) of a main diagonal bitstring C_d is bounded from below by $E[C_d^{(N,2)}] \geq 2^N + 2^{N/2}$, and its ASI, taking into account the relation (1), is bounded by

$$a_{\min}^{(N,2)} \leq a^{(N,2)}(C_d) \leq N - 1 - 2(N/2 - 1) + a_{\max}^{(N/2,2)} = a_{\max}^{(N/2,2)} + 1. \quad (37)$$

For example, for $N = 8$, the ASI of all main diagonal strings C_d satisfy $3 \leq a^{(8,2)}(C_d) \leq 4$, and they include the four minASI strings $\{[00\dots], [01\dots]\} \times 2$ corresponding to equidistributed decimals 0, 85, 170, and 255. For $N = 10$, the depleted sixteen maxASI strings correspond to $\{94, 122, 270, 286, 376, 450, 482, 488, 535, 541, 573, 647, 737, 753, 901, 929\}$ decimal values. For $N = 12$, the ten minASI strings $\{[00\dots], [001\dots], [010\dots], [01\dots], [011\dots]\} \times 2$ are equidistributed on the main square diagonal every 585 and every 1365. Finally, for $N = 16$, the small ASI strings are aligned on diagonals and skewed lines with the four minASI strings again equidistributed on the main square diagonal, while the 506 maxASI strings tend to attain row-like and column-like arrangements.

6. Supremacy of the ASI Compression over Polynomial-Time Compression Algorithms

Calculating the ASI of a string can be contrasted with compression algorithms that run in polynomial time, such as basic dictionary (LZ-type) schemes [43–45] or grammar-based algorithms, such as Re-Pair. Indeed, Turing's universality and algorithmic complexity show that any computable transformation, including the ASI, can be simulated symbolically with no loss of information [43]. This naturally raised doubts regarding the trivial computability of the ASI and its potential equivalence to popular lossless compression algorithms [43].

Yet these doubts proved moot. As the length of a string increases, computing the ASI becomes a hyper-exponentially hard, exhaustive process over all possible combinations and repeated substrings due to the enormous combinatorial space of possible construction pathways and the need to globally evaluate which fragments are most efficient to create early versus later; such a pathway minimization cannot be resolved entirely through local heuristics; there is no local strategy of finding optimal substrings in AT. This ASI problem is widely known in AT literature (cf. [13,46–48]). It was conjectured [10] that the ASI problem is NP-complete. A certificate-dependent (assembly-step) proof of this conjecture within predefined assembly spaces [12,47] was found [48] by providing a Karp reduction from the vertex cover problem. A complete, self-contained proofs of NP-completeness of the ASI problem for both decision and optimization variants and for all assembly spaces were found [14] by establishing a correspondence between string assembly spaces and straight-line grammars. Since, for each ASI, there exists a naïve algorithm (exhaustive search of the entire assembly tree), the ASI is computable for all finite objects [7]. Thus, one can say that determining ASI is computable and algorithmically trivial, but it is also computationally non-trivial because the assembly tree grows exponentially with N . One can see this non-triviality by noting that integrating machine-learning-driven methods does not improve ASI computation, even for moderately sized molecules [49]. Interestingly, human intuition can often

identify structural patterns or regularities enabling rapid non-algorithmic qualitative estimation of the ASI that remains inaccessible to sequential algorithms.

The computational non-triviality of the ASI raised a second question: whether a worse and slower algorithm [43], such as NP-complete ASI exhaustive search, merits consideration if it provides the same explanatory power and predictive accuracy as known algorithms. Indeed, if additional constraints are imposed on AT, such as restricting assembly steps to substrings of predefined length, allowing/excluding certain assembly steps [12,48], etc., then certain variants of calculating the ASI may yield the same, or even worse results as compared to known compression algorithms, given the large number of the latter.

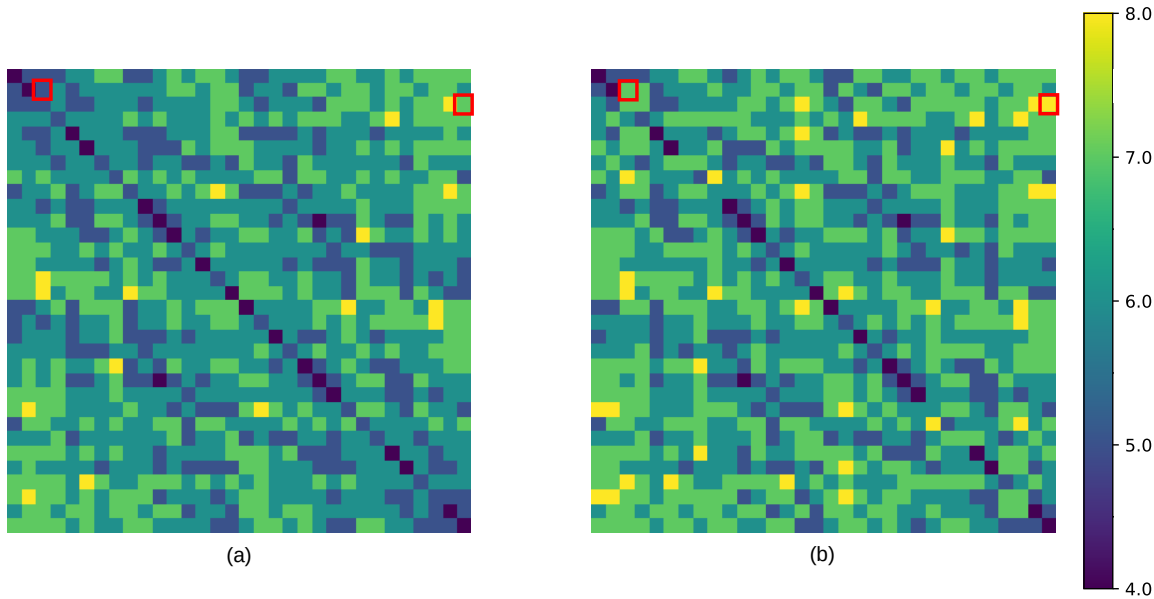


Figure 10. The ASI (a) and Re-Pair steps (b) of all bitstrings of length $N = 10$, as a function of their decimal representations ($\{0, 1, \dots, 1023\}$), in squares with sides equal to 32 pixels. Visible lack of perfect correlation between the ASI and Re-Pair: the number of Re-Pair steps is higher than ASI for 100 strings (e.g., $7 = \text{Re-Pair}(34) > \text{ASI}(34) = 5$ for $[0000100010]$, $8 = \text{Re-Pair}(95) > \text{ASI}(95) = 7$ for $[0001011111]$).

As an example, let us compare AT, a constrained AT* version that prohibits mixed 2-gram $[10]$ in the assembly pool (cf. Figure 2b), and the grammar-based Re-Pair method. The Re-Pair method involves counting all 2-grams in a string, selecting the most frequent 2-gram that occurs at least twice, replacing all its occurrences with a new symbol, and repeating these steps until no 2-gram occurs more than once. This is illustrated in the list (38) for an exemplary maxASI bitstring $[1111110000001001101010110]$ of length $N = 25$. As shown, the ASI of this string is fifteen, while the ASI* obtained by such a constrained AT* version is sixteen, which corresponds to the number of steps required to compress this string into a Re-Pair compressed version $[DAEB1BACC0A0]$.

step	ASI	ASI* (no 2-gram $[10]$)	Re-Pair
1.	$[00]$	$[00]$	$[11] \times 7 \rightarrow A, [AAA000000100A01010A0]$
2.	$[01]$	$[11]$	$[00] \times 6 \rightarrow B, [AAABBB1BA01010A0]$
3.	$[10]$	$[01]$	$[01] \times 2 \rightarrow C, [AAABBB1BACC0A0]$
4.	$[11]$	$[000]$	$[AA] \times 2 \rightarrow D, [DABBB1BACC0A0]$
5.	$[0110]$	$[111]$	$[BB] \times 2 \rightarrow E, [DAEB1BACC0A0]$
6.	$[1111]$	$[111111]$	$[DA]$
7.	$[111111]$	$[111111000]$	$[DAE]$
8.	$[11111100]$	$[111111000000]$	$[DAEB]$
9.	$[1111110000]$	$[1111110000001]$	$[DAEB1]$
10.	$[111111000000]$	$[111111000000100]$	$[DAEB1B]$
11.	$[11111100000010]$	$[11111100000010011]$	$[DAEB1BA]$
12.	$[111111000000100110]$	$[1111110000001001101]$	$[DAEB1BAC]$
13.	$[11111100000010011010]$	$[111111000000100110101]$	$[DAEB1BACC]$
14.	$[1111110000001001101010]$	$[11111100000010011010101]$	$[DAEB1BACC0]$
15.	$[1111110000001001101010110]$	$[1111110000001001101010111]$	$[DAEB1BACC0A]$
16.		$[11111100000010011010101110]$	$[DAEB1BACC0A0]$

However, an unconstrained AT provides different results. The distributions of ASI and Re-Pair steps for all the bitstrings of length $N = 10$ illustrated in Figure 10, show that ASI compression is superior to the one provided by the Re-Pair algorithm. We encourage the reader to conduct an independent comparison of the ASI (e.g., for the maxASI strings provided in Tables A3–A6) with the results produced by well-known compression algorithms.

7. Discussion

The present study provides a general mathematical framework for the general AT string version and demonstrates how this formalization enables efficient computation of assembly indices across various diverse structural sequences. Clarifying the theory's conceptual structure provides suggestions for its computational implementation. This allowed us to formalize bounding and pruning processes (an approach also proposed in [13,46]) and to develop software capable of effectively computing the ASI of a string, significantly reducing the number of potential configurations to explore. This, in turn, resulted in substantial efficiency gains, allowing us to find the exemplary long maxASI strings (cf. Tables A3–A6) and determine the maxASI bounds shown in Figure 8, overcoming important limitations exhibited by all the other known implementations of the algorithms to compute the ASI of *big objects* [50].

Fortunately, the non-triviality of computing the ASI does not close the field but rather delineates a structured landscape for algorithmic exploration. The framework developed here is equally applicable to symbolic representations of various real world patterns within the context of AT, such as detecting biosignatures [6], describing the evolution of music [51], and, as we conjecture, natural language processing (AI-generated text detection, authorship style analysis, plagiarism detection, time series analysis, etc.), financial data changes (distinction between random and predictable behaviour), sensor data anomaly analysis, medical signal analysis, etc.

The implications of our findings extend beyond the formal structure of AT itself. Traditional physics, while offering predictive power from past initial conditions to future states, lacks a functional perspective necessary to differentiate meaningful novelty from random fluctuations. AT may offer a unifying language for describing how information becomes structure. It can do so by showing how biological sequences are governed by foundational mathematical principles, for instance, for understanding the rules underlying the assembly of macromolecules such as proteins and DNA. Evolutionary biology explains the survival and prevalence of certain traits, but it does not address the mechanisms for generating novel phenotypic variants. We have shown, for example, that a DNA strand of length N containing four nucleobases cannot represent a minASI string without violating Chargaff's rules and Theorem 3.2. Theorem 3.2 establishes that a minASI string can contain at most three distinct symbols (if $N \neq 2^s$, as otherwise it can contain at most two), while the first Chargaff's rule (for double-stranded DNA) states that four nucleobases A, T, and G, C should be pairwise balanced. On the other hand, biological systems tend to deviate from the equilibrium of maximum entropy [10,52] corresponding to balanced strings,

For $b = 1$, the information entropy (23) vanishes, as the bit ($b = 2$) is the smallest amount and the quantum of information. Yet AT explains the assembly of such minASI strings, extending beyond the minimal two-valued system necessary to convey any information. Our results clarify that AT is a formal system for describing the generative and causal structure of objects, positioning it as a theoretical extension of information theory, emphasizing not the description of information but the effort required to construct it.

Author Contributions: W.B.: first concept of a general method for constructing the $C_{(N-1)}$ string and its implementation (the 1st method); the concept of the 2-gram matrix (27); proposition leading to Theorem 2.1; outline of the Sections 1 and 7; numerous clarity corrections and improvements; P.M.: outline of the 2nd method (A.15); observation of the relation between Theorems 2.3 and 2.4; crucial observations leading to the proofs of Lemmas 3.2 and 3.4; novel Strings (19); the concept of a Table A1; Conjecture 3.1; numerous clarity corrections and improvements; A.T.: formal proof of Theorem 3.2; proof of the Theorem 3.3; conceptualization of the proof of the

Theorem 2.4 and equation (4); numerous clarity corrections and improvements; S.T.: development of an efficient software for computing the ASI of a string generating the strings listed in Tables A3–A6; hint on the visualisation of the ASI distributions shown in Figure 8; numerous clarity corrections and improvements; S.L.: the remaining part of the study.

Funding: This research received no external funding.

Data Availability Statement: Public repository for the code and results of this study is given under the link <https://github.com/szluk/AssemblyTheory> (accessed on 28 December 2025).

Acknowledgments: Mariola Bala for her motivation, Rafał Bobrowski for assessing the time required to calculate a maxASI of a string as a function of its length, Robert Dobosz for providing valuable hints on earlier research [53] linking information and probability theories, Rafał Winiarski for noting that the relation (1) is inequality, anonymous referee for pointing out the coincidence of the DPI with $\lfloor \log_2(N) \rfloor + N_1 - 2$, shown in Figure 4, and SŁ's daughter Anna for her support in finding the pattern of strings (34). SŁ thanks his wife, Magdalena Bartocha, for her everlasting support, and his partner and friend, Renata Sobajda, for her prayers.

Conflicts of Interest: Authors Wawrzyniec Bieniawski and Piotr Masierak were employed by the company Łukaszcyk Patent Attorneys. The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AT	assembly theory;
N	length of a string;
b	number of basic symbols $c \in \{0, 1, \dots, b-1\}$;
$C_s, C_s^{(N,b)}$	a string;
$ASI, a^{(N,b)}(C_s)$	assembly index of a string C_s (minASI - minimum, maxASI - maximum);
$S(\Omega_{C_s})$	assembly space of a string C_s ;
$ASD, d_s^{(N_k,b)}(C_s)$	assembly depth of a string C_s (minASD - minimum, minASI ASD - the ASD of a minASI string);
DPI	depth index (OEIS A014701);
$EWT, E[C_k^{(N,b)}]$	expected waiting time;

OEIS Sequences

The following OEIS sequences are referred to in this manuscript:

A003313	Length of shortest addition chain for n (minASI);
A014701	Number of multiplications to compute n -th power by the Chandah-sutra method (DPI);
A026644	Number of moves to solve Chinese rings puzzle;
A048645	Integers with one or two 1-bits in their binary expansion;
A173786	Triangle read by rows: $T(n, k) = 2^n + 2^k, 0 \leq k \leq n$.

Appendix A

Table A1. Certain lengths of minASI strings, which are defined by the ASI and the minASI ASD for $2 \leq s \leq 7$.

s	$a_{\min}^{(N_*)} = 1$	$a_{\min}^{(N_*)} = 2$	$a_{\min}^{(N_*)} = 3$	$a_{\min}^{(N_*)} = 4$	$a_{\min}^{(N_*)} = 5$	$a_{\min}^{(N_*)} = 6$	$a_{\min}^{(N_*)} = 7$	$a_{\min}^{(N_*)} = 8$	$a_{\min}^{(N_*)} = 9$	...	$\tilde{N}_{2^n+1}$
2	2	4	3	7	14	28	56	112	224	...	$\tilde{N}_3$
					15	30	60	120	240	480	$\tilde{N}_{3,d}$
						23	46	92	184	368	$\tilde{N}_{3,b}$
3	2	4	8	3	11	22	44	88	176	...	$\tilde{N}_3$
						27	54	108	216	432	$\tilde{N}_{3,d}$
							43	86	172	344	$\tilde{N}_{3,b}$
	2	4	8	5	13	26	52	104	208	...	$\tilde{N}_5$
						45	90	180	360	...	$\tilde{N}_{5,b}$
4	2	4	8	16	3	19	38	76	152	...	$\tilde{N}_3$
							51	102	204	408	$\tilde{N}_{3,d}$
								83	166	332	$\tilde{N}_{3,b}$
	2	4	8	16	5	21	42	84	168	...	$\tilde{N}_5$
							85	170	340	...	$\tilde{N}_{5,b}$
	2	4	8	16	9	25	50	100	200	...	$\tilde{N}_9$
5	2	4	8	16	32	3	35	70	140	...	$\tilde{N}_3$
								99	198	396	$\tilde{N}_{3,d}$
									163	326	$\tilde{N}_{3,b}$
	2	4	8	16	32	5	37	74	148	...	$\tilde{N}_5$
								165	330	...	$\tilde{N}_{5,b}$
	2	4	8	16	32	9	41	82	164	...	$\tilde{N}_9$
6	2	4	8	16	32	17	49	98	196	...	$\tilde{N}_{17}$
	2	4	8	16	32	64	3	67	134	...	$\tilde{N}_3$
									195	390	$\tilde{N}_{3,d}$
										323	$\tilde{N}_{3,b}$
	2	4	8	16	32	64	5	69	138	276	$\tilde{N}_5$
7	2	4	8	16	32	64	9	73	146	...	$\tilde{N}_9$
									325	650	$\tilde{N}_{5,b}$
	2	4	8	16	32	64	17	81	162	...	$\tilde{N}_{17}$
	2	4	8	16	32	64	33	97	194	...	$\tilde{N}_{33}$
	2	4	8	16	32	64	128	3	131	...	$\tilde{N}_3$
8										387	$\tilde{N}_{3,d}$
	2	4	8	16	32	64	128	5	133	266	$\tilde{N}_5$
										645	$\tilde{N}_{5,b}$
	2	4	8	16	32	64	128	9	137	...	$\tilde{N}_9$
	2	4	8	16	32	64	128	17	145	...	$\tilde{N}_{17}$
	2	4	8	16	32	64	128	33	161	...	$\tilde{N}_{33}$
9	2	4	8	16	32	64	128	65	193	...	$\tilde{N}_{65}$

Table A2. Pathways leading to minASI strings (maximizing the number of independent assembly steps - MIA, maximizing the binary Shannon entropy - MBL). for $2 \leq N \leq 65$ (see Section 3 for details).

N	$d_{\min}^{(N)} = \lceil \log_2(N) \rceil$	$d_{\min}^{(N)}$	$a_{\min}^{(N)}$	$\hat{a}_{\min}^{(N)}$	MIA pathway	MBL pathway (Hamming weight N_1)	String
2	1	1	1	1		{2} (1)	$\hat{N}_1$
3	2	2	2	2		{2,3} (1)	$\hat{N}_1$
4	2	2	2	2		{2,4} (2)	$\hat{N}_1$
5	3	3	3	3		{2,4,5} (2)	$\hat{N}_1$
6	3	3	3	3		{2,4,6} (3)	$\hat{N}_1$
7	3	3	4	4	{2, (3,4), 7}	{2,4,6,7} (3)	$\hat{N}_3$
8	3	3	3	3		{2,4,8} (4)	$\hat{N}_1$
9	4	4	4	4		{2,4,8,9} (4)	$\hat{N}_1$
10	4	4	4	4		{2,4,8,10} (5)	$\hat{N}_1$
11	4	4	5	5	{2, (3,4), 7, 11}	{2,4,8,10,11} (5)	$\hat{N}_3$
12	4	4	4	4	{2,3,6,12}	{2,4,8,12} (6)	$\hat{N}_1$
13	4	4	5	5	{2,4, (5,8), 13}	{2,4,8,12,13} (6)	$\hat{N}_5$
14	4	4	5	5	{2, (3,4), 7, 14}	{2,4,8,12,14} (7)	$\hat{N}_3$
15	4	5	5	6		{2,3,5,10,15} (6)	$\hat{N}_7$
16	4	4	4	4		{2,4,8,16} (8)	$\hat{N}_1$
17	5	5	5	5		{2,4,8,16,17} (8)	$\hat{N}_1$
18	5	5	5	5		{2,4,8,16,18} (9)	$\hat{N}_1$
19	5	5	6	6	{2, (3,4), 8, 11, 19}	{2,4,8,10,18,19} (9)	$\hat{N}_3$
20	5	5	5	5	{2,3,5,10,20}	{2,4,8,16,20} (10)	$\hat{N}_1$
21	5	5	6	6	{2,4, (5,8), 16, 21}	{2,4,8,16,20,21} (10)	$\hat{N}_5$
22	5	5	6	6	{2, (3,4), 7, 11, 22}	{2,4,8,16,20,22} (11)	$\hat{N}_3$
23	5	6	6	7		{2,3,5,10,20,23} (9)	$\hat{N}_{2^{n+1},b}$
24	5	5	5	5		{2,4,8,12,24} (12)	$\hat{N}_1$
25	5	5	6	6	{2,4,8, (9,16), 25}	{2,4,8,16,24,25} (12)	$\hat{N}_9$
26	5	5	6	6	{2,4, (5,8), 13, 26}	{2,4,8,16,24,26} (13)	$\hat{N}_5$
27	5	6	6	7	{2,3,6,12,24,27}	{2,4,5,9,18,27} (12)	$\hat{N}_{2^{n+1},a}$
28	5	5	6	6	{2, (3,4), 7, 14, 28}	{2,4,8,16,24,28} (14)	$\hat{N}_3$
29	5	6	7	7	{2,4,8, (9,10), 20, 29}	{2,4,8,16,24,28,29} (14)	
30	5	6	6	7	{2,3,5,10,15,30}	{2,4,6,10,20,30} (15)	$\hat{N}_7$
31	5	6	7	8	{2,4, (5,8), 13, 26, 31}	{2,4,8,10,20,30,31} (15)	$\hat{N}_{15}$
32	5	5	5	5		{2,4,8,16,32} (16)	$\hat{N}_1$
33	6	6	6	6		{2,4,8,16,32,33} (16)	$\hat{N}_1$
34	6	6	6	6		{2,4,8,16,32,34} (17)	$\hat{N}_1$
35	6	6	7	7	{2, (3,4), 7, 14, 28, 35}	{2,4,8,16,32,34,35} (17)	$\hat{N}_3$
36	6	6	6	6		{2,4,8,16,32,36} (18)	$\hat{N}_1$
37	6	6	7	7	{2,4, (5,8), 16, 32, 37}	{2,4,8,16,32,36,37} (18)	$\hat{N}_5$
38	6	6	7	7	{2, (3,4), 8, 11, 19, 38}	{2,4,8,16,32,36,38} (19)	$\hat{N}_3$
39	6	6	7	8	{2,4, (5,8), 13, 26, 39}	{2,4, (5,8), 13, 26, 39} (18)	
40	6	6	6	6	{2,4,8,16,32,40}	{2,4,8,16,32,40} (20)	$\hat{N}_1$
41	6	6	7	7	{2,4,8, (9,16), 25, 41}	{2,4,8,16,32,40,41} (20)	$\hat{N}_9$
42	6	6	7	7	{2, (3,4), 7, 14, 28, 42}	{2,4,8,16,32,40,42} (21)	$\hat{N}_5$
43	6	7	7	8		{2,3,5,10,20,40,43} (17)	$\hat{N}_{2^{n+1},b}$
44	6	6	7	7	{2, (3,4), 7, 11, 22, 44}	{2,4,8,16,32,40,44} (22)	$\hat{N}_3$
45	6	7	7	8	{2,3,5,10,20,40,45}	{2,4,5,9,18,27,45} (20)	$\hat{N}_{2^{n+1},b}$
46	6	7	7	8	{2,3,5,10,20,23,46}	{2,4,6,10,20,40,46} (23)	$\hat{N}_{2^{n+1},b}$
47	6	7	8	9	{2, (3,4), 7, 11, 22, 44, 47}	{2,4,6,10,20,40,46,47} (23)	$\hat{N}_{15}$
48	6	6	6	6		{2,4,8,12,24,48} (24)	$\hat{N}_1$
49	6	7	7	7		{2,4,8,12,24,48,49} (24)	$\hat{N}_{17}$
50	6	6	7	7	{2,4,8, (9,16), 25, 50}	{2,4,8,16,32,40,48,50} (25)	$\hat{N}_9$
51	6	7	7	8		{2,4,8,16,17,34,51} (24)	$\hat{N}_{2^{n+1},a}$
52	6	6	7	7	{2,4, (5,8), 13, 26, 52}	{2,4,8,16,32,40,48,52} (26)	$\hat{N}_5$
53	6	7	8	8	{2,4, (5,8), 16, 32, 48, 53}	{2,4,8,16,32,40,48,52,53} (26)	
54	6	7	7	8	{2,3,6,12,24,27,54}	{2,4,6,12,24,48,54} (27)	$\hat{N}_{2^{n+1},a}$
55	6	7	8	9	{2, (3,4), 7, 11, 22, 44, 55}	{2,4,8,16,18,36,54,55} (27)	
56	6	6	7	7	{2, (3,4), 7, 14, 28, 56}	{2,4,8,16,32,48,56} (28)	$\hat{N}_3$
57	6	7	8	8	{2, (3,4), 7, 14, 28, 56, 57}	{2,4,8,16,32,48,56,57} (28)	
58	6	7	8	8	{2, (3,4), 7, 14, 28, 29, 58}	{2,4,8,16,32,48,56,58} (29)	
59	6	7	8	9	{2, (3,4), 7, 14, 28, 56, 59}	{2,4,5,9,18,27,54,59} (26)	$\hat{N}_{27}$
60	6	7	7	8	{2,4,8,12,24,48,60}	{2,4,6,10,20,30,60} (30)	$\hat{N}_7$
61	6	8	8	9	{2,4,8,12,24,48,60,61}	{2,4,8,16,20,40,60,61} (30)	
62	6	7	8	9	{2, (3,4), 7, 14, 28, 31, 62}	{2,4,8,16,20,40,60,62} (31)	$\hat{N}_{15}$
63	6	7	8	10	{2, (3,4), 7, 14, 21, 42, 63}	{2,4,5,9,18,27,45,63} (28)	
64	6	6	6	6		{2,4,8,16,32,64} (32)	$\hat{N}_1$
65	7	7	7	7		{2,4,8,16,32,64,65} (32)	$\hat{N}_1$

Table A3. Exemplary maxASI bitstrings for $10 \leq N \leq 39$.

N	$C_{\max}^{(N,2)}$	$a_{\max}^{(N,2)}$	N	$C_{\max}^{(N,2)}$	$a_{\max}^{(N,2)}$
10	[1010000111]	8	25	[0000000101010110011111110]	15
11	[00010111100]	8	26	[01001100000111111101010110]	16
12	[101010000111]	9	27	[000000011111110101011001000]	16
13	[1000001110101]	9	28	[0110101011111110000000110010]	17
14	[10011000010111]	10	29	[0110000000101010011101111110]	17
15	[000001010111110]	10	30	[1001000000001100101010111111]	17
16	[1001100001010111]	11	31	[010101001000000011111101101100]	18
17	[0000001010111110]	11	32	[010011000000001010101111111001]	18
18	[100110100001010111]	12	33	[10000000001001111111011101101010]	18
19	[0111110110000010100]	12	34	[100000000010011111110111011010101]	18
20	[10011010000101011111]	13	35	[1010100000001011001001111110001101]	19
21	[000000010101100111110]	13	36	[10101000000010110010011111100011101]	19
22	[001011111101001100000]	14	37	[1011010101000000010010001111111001101]	19
23	[00000001010101100111110]	14	38	[10111010101000000010010001111111001101]	20
24	[011001111111010100000001]	15	39	[1110011001000110100000010101011011111]	20

Table A4. Exemplary maxASI bitstrings for $40 \leq N \leq 85$.

N	$C_{\max}^{(N,2)}$	$a_{\max}^{(N,2)}$
40	[00110010111111110101000000011011000101101]	20
41	[000001111110011110101001011011011000100110]	21
42	[001101111110101010110000000111100100100101]	21
43	[0111100111110110010100000011100011000101101]	21
44	[11101010101011111100100100011000000010110110]	22
45	[11101010101011111100100100011000000011011010]	22
46	[01111001111110100011100000001100100101001101]	22
47	[011110011111101000111000000011001001010110110]	23
48	[0111100111111010001110000000110010010101101100]	23
49	[01111001111110100011100000001100100101011011000]	23
50	[10100111111100010001111010000001011011000011100110]	23
51	[101001111111000100011110100000010110110010011010101]	24
52	[1010011111110001000111101000000101101100100100110101]	24
53	[10100111111100010001111010000001011011001001001101010]	24
54	[1010011111110001000111101000000101101100100101010101]	25
55	[1010011111110001000111101000000101101100100100110101010]	25
56	[10100111111100010001111010000001011011001001001101010101]	25
57	[101001111111000100011110100000010110110010010011010101010]	25
58	[1010011111110001000111101000000101101100100100110101010101]	26
59	[100010111001110011111101101000000110000011001001001010101]	26
60	[101010111011100111111101001000011001000100010010110000000]	26
61	[1010101110111001111111010010000110010001000100101100000001]	26
62	[10101011101110011111110100100001100100010001001011011000001]	27
63	[101010111011100111111101001000011001000100010010110110000000]	27
64	[1010101110111001111111010010000110010001000100101101100000001]	27
65	[10101011101110011111110100100001100100010001001011011011000001]	28
66	[101010101011001000111110110111110001000000110011001110011010010010]	28
67	[1010101010110010001111101101111100010000001100110011100110100100101]	28
68	[10101010101100100011111011011111000100000011001100111001110001011110]	28
69	[101010101011001000111110110111110001000000110011001110011100010100101]	29
70	[1010101010110010001111101101111100010000001100110011100111000101001001]	29
71	[10101010101100100011111011011111000100000011001100111001110001010010010]	29
72	[101010101011001000111110110111110001000000110011001110011100010100100101]	30
73	[1010101010110010001111101101111100010000001100110011100111000101001001001]	30
74	[10101010101100100011111011011111000100000011001100111001110001010010000001]	30
75	[101010101011001000111110110111110001000000110011001110011100010100100000001]	30
76	[1010101010110010001111101101111100010000001100110011100111000101001000010000]	31
77	[1010101010110010001111101101111100010000001100110011100111000101001000000000]	31
78	[101010101011001000111110110111110001000000110011001110011100010100100000000001]	31
79	[1001011101101011111110100110110010011101010110100101000110011110111100000001011]	31
80	[10010111011010111111101001101100100111010101101001010001100111101111000000010101]	32
81	[100101110110101111111010011011001001110101011010010100011001111011110000000101010]	32
82	[1001011101101011111110100110110010011101010110100101000110011110111100000001010100]	32
83	[10010111011010111111101001101100100111010101101001010001100111101111000000010101000]	33
84	[100101110110101111111010011011001001110101011010010100011001111011110000000101011100]	33
85	[1001011101101011111110100110110010011101010110100101000110011110111100000001010111000]	33

Table A5. Exemplary maxASI strings for $b = 3$ and $13 \leq N \leq 54$.

N	$C_{\max}^{(N,3)}$	$a_{\max}^{(N,3)}$
13	[0002220111210]	12
14	[00022201112101]	12
15	[000222011121012]	13
16	[0002220111210120]	14
17	[20011121002201021]	14
18	[222111210100001202]	15
19	[0221110100122200021]	16
20	[02211101001222000211]	16
21	[022111010012220002112]	17
22	[022111010012220002021]	18
23	[02211101001222000211201]	18
24	[022111010012220002011210]	19
25	[0222212112002010001111021]	20
26	[02222121120020100011110210]	20
27	[012221211200201000111102202]	21
28	[0122212112002010001111022010]	21
29	[01222121120020100011110220210]	22
30	[012221211200201000111102202102]	22
31	[0122212112002010001111022021020]	23
32	[01222121120020100011110220210200]	23
33	[012221211200201000111102202102000]	24
34	[0122212112002010001111022021020001]	24
35	[01222121120020100011110220210200000]	25
36	[012221211200201000111102202102000001]	25
37	[0122212112002010001111022021020010101]	26
38	[01222121120020100011110220210200101101]	26
39	[012221211200201000111102202102001011012]	26
40	[0122212112002010001111022021020010110000]	27
41	[01222121120020100011110220210200101100002]	27
42	[012221211200201000111102202102001011000022]	28
43	[0122212112002010001111022021020010110000222]	28
44	[01222121120020100011110220210200101100000110]	29
45	[012221211200201000111102202102001011000001110]	29
46	[2111020110222211012201112212121010020000001202]	29
47	[21110201102222110122011122121210100200000010220]	30
48	[211102011022221101220111221212101002000000120210]	30
49	[2111020110222211012201112212121010020000001202112]	31
50	[21110201102222110122011122121210100200000012021120]	31
51	[211102011022221101220111221212101002000000120212210]	32
52	[2111020110222211012201112212121010020000001202122120]	32
53	[21110201102222110122011122121210100200000012021200220]	33
54	[211102011022221101220111221212101002000000120212002202]	33

Table A6. Exemplary maxASI strings for $b = 4$ and $21 \leq N \leq 49$.

N	$C_{\max}^{(N,4)}$	$a_{\max}^{(N,4)}$
21	[000111222333102132030]	20
22	[0001112223331021320302]	20
23	[00011122233310213203012]	21
24	[010000111222333102132030]	22
25	[0100001112223331021320302]	22
26	[01000011122233310213203023]	23
27	[010000111222333102132030221]	24
28	[000110201333112130122230323]	24
29	[00011320133311121022232302030]	25
30	[30100001211112322233310320213]	26
31	[301000012111123222333103202130]	26
32	[3010000121111232223331032021303]	27
33	[30100001211112322233310320213313]	28
34	[301000012111123222333203102133130]	28
35	[3010000121111232223332031021331300]	29
36	[30100001211112322233320310213313110]	30
37	[301000012111123222333203102133131101]	30
38	[3010000121111232223332031021331311011]	31
39	[30100001211112322233320310213313110221]	32
40	[301000012111123222333203102133131102210]	32
41	[3010000121111232223332031021331311022101]	33
42	[30100001211112322233320310213313110322011]	33
43	[301000012111123222333203102133131102210103]	34
44	[3010000121111232223332031021331311022101030]	34
45	[30100001211112322233320310213313110221201300]	35
46	[301000012111123222333203102133131102212013002]	36
47	[3010000121111232223332031021331311022120130023]	36
48	[30100001211112322233320310213313110221201300230]	37
49	[301000012111123222333203102133131102212013002303]	38

Appendix A.1. Proof of Theorem 2.1

Consider the following two strings of the same length $N + 8$ with $** \neq 01$ and the same distributions of other repetitions (if any)

$$C_k = [\dots 01 \dots 01 \dots 01 \dots * \star \dots], \quad C_l = [\dots 01 \dots 01 \dots 22 \dots 22 \dots]. \quad (A1)$$

Assembling a 2-gram takes one assembly step. Each appending of a 2-gram to an assembled string counts as another assembly step. Hence, in a general case (i.e., for strings C_k, C_l containing also other symbols), the string C_k requires six additional assembly steps, the same as the string C_l . Thus, a string containing the same three 2-grams has the same ASI as a string containing two pairs of the same 2-grams, provided that both strings have the same distributions of other repetitions and have the same lengths.

Consider the following two strings of the same length $N + 6$ with the same distributions of other repetitions

$$C_k = [\dots 01 \dots 01 \dots 01 \dots], \quad C_l = [\dots 010 \dots 010 \dots]. \quad (A2)$$

The assembly of a 3-gram takes two steps. Hence, in the general case, the string C_k requires four additional assembly steps, the same as the string C_l . Thus, a string containing the same three 2-grams has the same ASI as a string containing the same two 3-grams, provided that both strings have the same distributions of other repetitions.

Similarly, a string containing the same two 3-grams has the same ASI as a string containing two pairs of the same 2-grams, provided that both strings have the same distributions of other repetitions and have the same lengths.

Consider the following two strings of the same length $N + 9$ with the same distributions of other repetitions

$$C_k = [\dots 0101 \dots 0101 \dots * \star \dots], \quad C_l = [\dots 010 \dots 010 \dots 010 \dots]. \quad (A3)$$

The assembly of such a 4-gram takes two steps. Hence, in a general case, the string C_k requires five additional assembly steps, the same as the string C_l . Thus, a string containing the same two 4-grams of the minASI has the same ASI as a string containing the same three 3-grams, provided that both strings have the same distributions of other repetitions and have the same lengths.

Consider the following two strings of the same length $N + 8$ with the same distributions of other repetitions

$$C_k = [\dots 0001 \dots 0001 \dots], \quad C_l = [\dots 110 \dots 10 \dots 110 \dots]. \quad (A4)$$

The assembly of such a 4-gram takes three steps. Hence, in a general case, the string C_k requires five additional assembly steps, the same as the string C_l . Thus, a string containing the same two 4-grams of the maxASI has the same ASI as a string containing a 2-gram and the same two 3-grams based on this 2-gram, provided that both strings have the same distributions of other repetitions.

Consider the following two strings of the same length $N + 10$ with the same distributions of other repetitions

$$C_k = [\dots 110 \dots 00 \dots 110 \dots 00 \dots], \quad C_l = [\dots 110 \dots 10 \dots 110 \dots * \star \dots], \quad (A5)$$

where $** \notin \{11, 10\}$. In a general case, the string C_k requires seven additional assembly steps, the same as the string C_l . Thus, a string containing the same two 2-grams and the same two 3-grams not based on this 2-gram has the same ASI as a string containing a 2-gram and the same two 3-grams based on this 2-gram, provided that both strings have the same distributions of other repetitions and have the same lengths.

In general, the above considerations show that

- k copies of a 2-gram in a string decrease the ASI of this string at least by $k - 1$;
- k copies of a 3-gram in a string decrease the ASI of this string at least by $2k - 2$;

- k copies of a minASI 4-gram in a string decrease the ASI of this string at least by $3k - 2$;
- k copies of a maxASI 4-gram in a string decrease the ASI of this string at least by $3k - 3$;

where, the phrase "at least" is meant to indicate that other repetitions, such as e.g. 2-grams forming multiple 4-grams, etc. can further decrease the ASI of the string. W.l.o.g., consider the following string

$$C_m^{(N,b)} = [\dots [c_1 c_2 \dots c_n] \dots [c_1 c_2 \dots c_n] \dots], \quad (\text{A6})$$

containing two copies of an n -gram $C_l^{(n,b)} = [c_1 c_2 \dots c_n]$. The n -gram $C_l^{(n,b)}$ can be assembled in at least $a^{(n,b)}(C_l)$ steps and appended to the assembled string C_m in one step. Consider that the ASI of the n -gram $C_l^{(n,b)}$ is $a^{(n,b)}(C_l) = n - 1$, i.e. the n -gram does not have any repetitions that can be reused and $n < 8$. Then one copy of this n -gram - as expected - does not decrease the ASI of the string $C_m^{(N,b)}$, as $1(n - 1) - (n - 1) = 0$, while more copies k decrease it by $(n - 1)(k - 1)$. On the other hand, if $a^{(n,b)}(C_l) < n - 1$ then even a single copy of this n -gram will decrease the ASI of C_m .

Appendix A.2. Proof of Theorem 2.2

Without loss of generality (w.l.o.g.) assume *a contrario* that Ω contains two strings C_l, C_m having the same ASD, i.e., $d^{(N_l,b)}(C_l) = d^{(N_m,b)}(C_m) \neq 0$, that were not assembled in independent assembly steps, i.e., that C_m was used in the assembly of C_l along with a basic symbol c in some previous step s . Then

$$d_s^{(N_l,b)}(C_l) = \max(d^{(N_m,b)}(C_m), d^{(1,b)}(c)) + 1 = d^{(N_m,b)}(C_m) + 1 \neq d^{(N_m,b)}(C_m), \quad (\text{A7})$$

which contradicts our assumption and completes the proof.

Appendix A.3. Proof of Lemma 2.2

Assume *a contrario* that $N > 2^{d_s^{(N)}}$. Then for the ASD $d_s^{(N)} = 0$, we have $N > 2^0 = 1$ which is a contradiction as all basic symbols c are unit-length strings and $N = 1$. Similarly, for $d_s^{(N)} = 1$, $N > 2$ is also contradiction in the case of 2-grams, and so on. This is a consequence of the ASD Definition (3).

Appendix A.4. Proof of Theorem 2.3

$d_s^{(N)} \geq \log_2(N)$ follows from the relation (4). $d_{\min}^{(2)} = \lceil \log_2(2) \rceil = 1$ satisfies both the definition (3) and our hypothesis (5). Similarly $N = 3$. Using induction on length N , assume that for some $N > 3$, we can assemble a minASD string with ASD (5). We need to show that for $N + 1$, we can assemble a string with the ASD satisfying

$$d_{\min}^{(N+1)} = \lceil \log_2(N + 1) \rceil. \quad (\text{A8})$$

Since, by definition (3), the ASD as a function of N is monotonously nondecreasing and can increase at most by one between N and $N + 1$, we have

$$d_{\min}^{(N+1)} = d_{\min}^{(N)} = \lceil \log_2(N) \rceil \quad \text{or} \quad d_{\min}^{(N+1)} = d_{\min}^{(N)} + 1 = \lceil \log_2(N) \rceil + 1, \quad (\text{A9})$$

where we used relations (5) and (A8). Solving the relation (A9) for N yields

$$d_{\min}^{(N+1)} = \begin{cases} d_{\min}^{(N)} = s & \text{if } 2^{s-1} < N < 2^s, \\ d_{\min}^{(N)} + 1 = s + 1 & \text{if } N = 2^s, \end{cases} \quad (\text{A10})$$

and completes the proof.

Appendix A.5. Proof of Theorem 2.4

Using the property of the ceiling function $n = \lceil x \rceil \Leftrightarrow n - 1 < x \leq n$ valid for $n \in \mathbb{N}, x \in \mathbb{R}$, we have

$$d_{a_{\max}}^{(N,b)} = \lceil \log_2(N) \rceil \Leftrightarrow d_{a_{\max}}^{(N,b)} - 1 < \log_2(N) \leq d_{a_{\max}}^{(N,b)}, \quad (\text{A11})$$

The non-strict inequality (A11) corresponds to the non-strict inequality (4) valid for any N and any ASD. Therefore, we need to prove that the strict inequality $d_{a_{\max}}^{(N,b)} < \log_2(N) + 1$ holds for all $C_{\max}$ strings. Assume, for contradiction, that there exists a maxASI string $C_{\max}^{(N,b)}$ such that

$$d_{a_{\max}}^{(N,b)} \geq \log_2(N) + 1 = \log_2(2N) \implies 2^{d_{a_{\max}}^{(N,b)}} \geq 2N \implies N \leq 2^{d_{a_{\max}}^{(N,b)}-1}. \quad (\text{A12})$$

But this relation does not hold for the maxASI string $C_{\max}^{(N,b)}$.

Appendix A.6. Proof of Theorem 3.1

Strings $C_{\min}$ for which $a^{(N)}(C_{\min}) = \min_k \left(\{a^{(N,b)}(C_k)\} \right)$, $\forall k \in \{1, 2, \dots, b^N\}$ can be formed in subsequent steps s by joining the longest string assembled so far with itself until $N = 2^s$ is reached. Therefore, if $N = 2^s$, then $\min_k \left(\{a^{(2^s)}(C_k)\} \right) = s = \log_2(N)$. Only b^2 strings have such ASI if $N = 2^s$, including respectively b and $b(b-1)$ strings

$$C_k^{(2^s,b)} = [* * \dots], \quad C_l^{(2^s,b)} = [* * * \dots], \quad (\text{A13})$$

and the assembly space of each of the strings (A13) is unique. At each assembly step, its length doubles.

An addition chain for $N \in \mathbb{N}$ having the shortest length $s \in \mathbb{N}$ (commonly denoted as $l(N)$) is defined as a sequence $1 = a_0 < a_1 < \dots < a_s = N$ of integers such that $\forall j \geq 1, a_j = a_k + a_l$ for $k \leq l < j$. Hence, $j = 1 \implies k = l = 0$ and the first step in forming an addition chain for N is always $a_1 = a_0 + a_0 = 1 + 1 = 2$, which is equivalent to saying that the ASI of any 2-gram is one. The second step in forming an addition chain can be $a_2 = a_0 + a_1 = 1 + 2 = 3$ or $a_2 = a_1 + a_1 = 2 + 2 = 4$, which corresponds to assembling a 3-gram based on the previously assembled 2-gram or to assembling a minASI 4-gram (A13) from this 2-gram. The maxASI 4-gram can be assembled in a third step, $a_3 = a_2 + a_0 = 3 + 1 = 4$, which corresponds to joining a basic symbol to a 3-gram. Therefore, four is the smallest number achievable in two ways according to Theorem 2.1.

Thus, finding the shortest addition chain for N corresponds to finding the ASI of a string containing basic symbols and/or 2-grams and/or 3-grams containing these 2-grams if $N \neq 2^s$ since due to Theorem 2.1 only they provide the same assembly indices $\{0, 1, 2\}$ with no internal repetitions.

Appendix A.7. Proof of Theorem 3.3

To prove that the minASI ASD equals the minASI, we use mathematical induction on the length N of the string. For the base case ($N = 2^0 = 1$), the string consists of a single basic symbol $c \in P_0^{(b)}$, where $P_s^{(b)}$ denotes the assembly pool. Hence, its ASI is $a_{\min}^{(1)} := 0$ and its ASD $d_{a_{\min}}^{(1)} := 0$. Therefore, $d_{a_{\min}}^{(1)} = a_{\min}^{(1)} = 0$. Assume now that for all strings of length 2^s less than N , the ASD equals the minASI, that is

$$d_{a_{\min}}^{(2^s)} = a_{\min}^{(2^s)} \quad \forall 2^s < N. \quad (\text{A14})$$

For some integer s , we construct the minASI string as follows. First, we assemble a 2-gram from two basic symbols:

$$c_1 \circ c_2 = C^{(2,b)}, \quad c_1, c_2 \in P_0^{(b)}. \quad (\text{A15})$$

Its ASI is $a_{\min}^{(2)} = 1$ and its ASD is $d_{a_{\min}}^{(2)} = 1$. Then for each $s \geq 2$ we have $C^{(2^{s-1},b)}$ with the ASI $a_{\min}^{(2^{s-1})} = s - 1$ and the ASD $d_{a_{\min}}^{(2^{s-1})} = s - 1$ and we construct $C^{(2^s,b)}$ by joining two copies of $C^{(2^{s-1},b)}$

$$C^{(2^{s-1},b)} \circ C^{(2^{s-1},b)} = C^{(2^s,b)}. \quad (\text{A16})$$

The ASI of the string $C^{(2^s,b)}$ is equal to

$$a_{\min}^{(2^s)} = a_{\min}^{(2^{s-1})} + 1 = (s - 1) + 1 = s, \quad (\text{A17})$$

and, similarly, its ASD is equal to

$$d_{a_{\min}}^{(2^s)} := \max(d_{a_{\min}}^{(2^{s-1})}, d_{a_{\min}}^{(2^{s-1})}) + 1 = (s-1) + 1 = s. \quad (\text{A18})$$

Therefore, $a_{\min}^{(2^s)} = d_{a_{\min}}^{(2^s)} = s$. At any step, we assemble strings (A13), and no two assembly steps can be independent, which follows from Theorem 3.1. The equation (A10) establishes that $N = 2^s$ is the largest N for which $d_{\min}^{(N)} = s$. This proves $d_{\min}^{(2^s)} = d_{a_{\min}}^{(2^s)} = a_{\min}^{(2^s)} = s$. Finally, the even part of the definition of the DPI 2.7 is the only defining part of this definition iff $N = 2^s$. Hence, $a_{\min}^{(2^s)} = d_{a_{\min}}^{(2^s)} = a_{\min}^{(2^s)} = \hat{a}_{\min}^{(2^s)} = s$.

Appendix A.8. Proof of Theorem 3.4

The lengths (14) (OEIS A173786 or OEIS A048645) are the generalization of the strings of length $N = 2^{s-1} + 2^{s-1} = 2^s$ of the Theorem 3.3. For other lengths of the strings (14), the base case for $s = 2, l = 0$ describes the assembly of a 3-gram, by joining a symbol to a 2-gram made in the first step, so that both the ASI and the ASD of this 3-gram increase by one. And so on. For any s we can join a symbol to a string of length $N = 2^{s-1}$ assembled in $s - 1$ steps or join two such strings, as shown in Figure 3a.

To see that $\hat{a}_{\min}^{(\hat{N}_1)} = s$ (14) holds for $\hat{N}_1 \neq 2^s$ note that there is only one odd part of the definition of the DPI 2.7 that *restores* $N = 2^s$. For example, we reach one starting from $\hat{N}_1 = 20$ in five consecutive steps $20 \rightarrow 10 \rightarrow 5 \rightarrow 4 \rightarrow 2 \rightarrow 1$.

Appendix A.9. Proof of Lemma 3.1

We begin at $d = 3$ by assembling a $C_{\min}^{(7)}$ using a 4-gram and a 3-gram assembled independently (e.g., using an assembly space (15)) with $a_{\min}^{(7)} = 4$ and $d_{a_{\min}}^{(7)} = \lceil \log_2(7) \rceil = 3$. For $d = 4$, the string $C_{\min}^{(11)}$ can be assembled by joining the string $C_{\min}^{(8)}$ assembled in three steps and the 3-gram, while the string $C_{\min}^{(14)}$ by joining two strings $C_{\min}^{(7)}$ made in the previous step. For any d , the shortest string $C_{\min}^{(N_3)}$ can be assembled by joining the string $C_{\min}^{(2^{d-1})}$ (A13) assembled in $d - 1$ steps and the 3-gram, while the remaining strings $C_{\min}^{(N_3)}$ - by joining two strings made in a previous step $d - 1$, as shown in Figure 3b.

Appendix A.10. Proof of Lemma 3.2

We begin at $d = 4$ by assembling a $C_{\min}^{(13)}$ through $\{2, 4, (5, 8), 13\}$ with $a_{\min}^{(13)} = d_{\min}^{(13)} + 1 = 5$. For any d , the shortest string (17) $C_{\min}^{(N_5)}$ can be assembled by joining the string $C_{\min}^{(2^{d-1})}$ (A13) assembled in $d - 1$ steps with the 5-gram assembled in the independent assembly step, while the remaining strings $C_{\min}^{(N_5)}$ - by joining two strings made in a previous step $d - 1$, as shown in Figure 3c.

Appendix A.11. Proof of Lemma 3.3

We begin at $d = 5$ by assembling a $C_{\min}^{(25)}$ with $a_{\min}^{(25)} = d_{\min}^{(25)} + 1 = 6$. For any d , we assemble the shortest strings (18) as

$$\begin{aligned} &\{2, \quad 4, \quad 8, \quad (9, \quad 16), \quad 25\}, \\ &\{\dots \quad \quad \quad 32, \quad 41\}, \\ &\{\dots \quad \quad \quad 64, \quad 73\}, \\ &\{\dots \quad \quad \quad 128, \quad 137\}, \end{aligned} \tag{A19}$$

with one independent assembly step (9,16) to assemble the string of length $N = 2^{d-1}$ and joining 9-gram at the last step, while the remaining strings $C_{\min}^{(\tilde{N}_9)}$ - by joining two strings made in a previous step $d - 1$, as shown in Figure 3d.

Appendix A.12. Proof of Theorem 3.6

The equality of ASI and ASD of the strings (20) follows from the proof of Theorem 3.4. Furthermore $\forall d$,

$$\begin{aligned} 2^{d-1} &< 2^{d-1} + 7 \cdot 2^{d-4} < 2^d \quad / - 2^{d-1} \\ 0 &< 7 \cdot 2^{d-4} < 2^d - 2^{d-1} = 2^{d-1} \quad / \cdot 2^{4-d} \\ 0 &\leq 7 \leq 2^{d-1} 2^{4-d} = 8. \end{aligned} \tag{A20}$$

shows that $d_{a_{\min}}^{(\tilde{N}_7)} = \lceil \log_2(\tilde{N}_7) \rceil + 1$. Finally, $\hat{d}_{\min}^{(\hat{N}_1)} = \lceil \log_2(\tilde{N}_7) \rceil + 2$ follows from the DPI Definition 2.7: six steps are required to reach one starting from fifteen and additional steps for thirty, sixty, etc., which completes the proof.

Appendix A.13. Support for Conjecture 3.2

The shortest strings of length $\tilde{N}_{15}$ (22a) can be assembled with the pathways

$$\begin{aligned} & \{2, \quad 4, \quad (5, \quad 8), \quad 13, \quad 26, \quad 31\} \\ & \{\dots \qquad \qquad \qquad 39, \quad 47\}, \\ & \{\dots \qquad \qquad \qquad 78, \quad 79\}, \end{aligned} \tag{A21}$$

shown in Figure 3e; the shortest strings of length $\tilde{N}_{27}$ (22b) can be assembled with the pathways

$$\begin{aligned} &\{2, \quad (3, \ 4), \ 7, \ 14, \ 28, \ 31, \ 59\} \\ &\{\dots \qquad \qquad \qquad 14 \ 28, \ 56, \ 84, \ 91\} \\ &\{\dots \qquad \qquad \qquad 11 \ 18, \ 36, \ 72, \ 144, \ 155\} \\ &\dots \end{aligned} \tag{A22}$$

shown in Figure 3f; and for any d , the shortest strings of length $\tilde{N}_{50.9}$ (22c) can be assembled as

$$\begin{aligned} &\{2, \quad 4, \quad 8, \quad (9, \quad 16), \quad 25, \quad 50, \quad 59\}, \\ &\{\dots \qquad \qquad \qquad 100, \quad 109\}, \\ &\{\dots \qquad \qquad \qquad 200, \quad 209\} \end{aligned} \tag{A23}$$

The remaining strings of length $\tilde{N}_{15}$, $\tilde{N}_{27}$, and $\tilde{N}_{50.9}$ (22) can be assembled by joining two strings made in a previous step $d - 1$.

Appendix A.14. Proof of Lemma 3.4

By Theorems 3.1 and 3.2, a minASI string of length $N = 2^s$ assembled beginning with $C_{\min}^{(2)} = [**]$ is a balanced bitstring. To assemble a longer string of other lengths, we assign $N_{s+1} = N_s + 1$ or $N_{s+1} = N_s + 2$. However, the Definition 2.7 removes the longest string of an odd length $N = 2^s + 1$ from the sequence if it is not the first one in the sequence. Strings longer than this string of length $N = 2^s + 1$ are assembled by joining the longest string assembled so far with itself ($N_{s+1} = 2N_s$) or by joining a basic symbol chosen to preserve the balance of the string ($N_{s+1} = N_s + 1$).

Appendix A.15. The 2nd Method for Generating $C_{(N-1)}$ Strings

This method is similar to the 1st method discussed in Section 4. We also start with a string of clear 3-grams (26) and the matrix of 2-grams (27) with a crossed diagonal and the first superdiagonal. In the first step, we append the 2-gram $0b_{-1}$ (top right 2-gram of the matrix of 2-grams (27)) at the end of the string (26). Next, we generally perform the following pairs of iterations:

- 1. we check subsequent subdiagonals until we find one that does not contain a 2-gram present in the string formed so far, we append it at the end of this string and proceed to step 2;
- 2. we check subsequent superdiagonals until we find one that does not contain a 2-gram present in the string formed so far, we append it at the end of this string and proceed to step 1.

Finally, we append 0 if b is even. The method is illustrated in Figure A1 and generates the $C_{(N-1)}$ strings in the form

$$\begin{aligned} C_{(N-1)}^{(3,1)} &= [000], \\ C_{(N-1)}^{(7,2)} &= [000111|0], \\ C_{(N-1)}^{(13,3)} &= [000111222|02|10], \\ C_{(N-1)}^{(21,4)} &= [000111222333|03|102132|0], \\ C_{(N-1)}^{(31,5)} &= [000111222333444|04|10213243|0314|20], \\ C_{(N-1)}^{(43,6)} &= [000111222333444555|05|1021324354|031425|304152|0], \\ C_{(N-1)}^{(57,7)} &= [000111222333444555666|06|102132435465|03142536|405162|041526|30], \\ C_{(N-1)}^{(73,8)} &= [000111222333444555666777|07|10213243546576|0314253647|3041526374|051627|506172|0], \\ C_{(N-1)}^{(91,9)} &= [\dots|08|1021324354657687|031425364758|304152637485|05162738|607182|061728|40], \\ C_{(N-1)}^{(111,10)} &= [\dots|09|102132435465768798|03142536475869|30415263748596|0516273849|5061728394| \\ &\quad 071829|708192|0], \\ C_{(N-1)}^{(133,11)} &= [\dots|0a|102132435465768798a9|031425364758697a|30415263748596a7|05162738495a| \\ &\quad 60718293a4|061728394a|8091a2|08192a|50], \\ C_{(N-1)}^{(157,12)} &= [\dots|0b|102132435465768798a9ba|031425364758697a8b|30415263748596a7b8|05162738495a6b| \\ &\quad 5061728394a5b6|0718293a4b|708192a3b4|091a2b|90a1b2|0], \\ C_{(N-1)}^{(183,13)} &= [\dots|0c|102132435465768798a9bacb|031425364758697a8b9c|30415263748596a7b8c9|05162738495a6b7c| \\ &\quad 5061728394a5b6c7|0718293a4b5c|8091a2b3c4|08192a3b4c|a0b1c2|0a1b2c|60], \\ &\dots \end{aligned} \tag{A24}$$

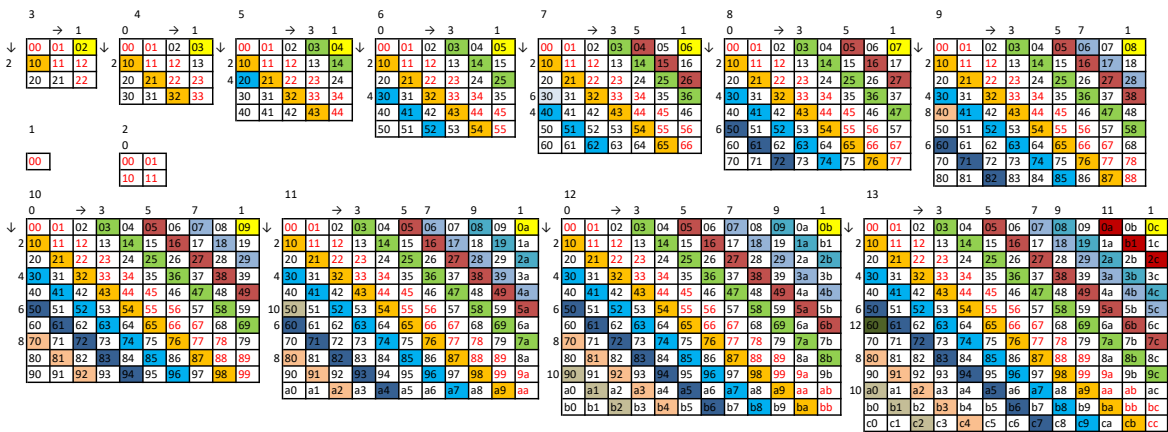


Figure A1. 2-gram matrices for $1 \leq b \leq 13$ that illustrate the generation of $N_{(N-1)}$ strings according to the 2nd method. Coloured 2-grams are appended to the initial string of clear 3-grams in the order indicated by arrows starting from the 1st column or row. Finally, 0 is appended at the end if b is even.

Appendix A.16. Method for Generating Non-Balanced $C_{(N-b)}$ Strings

The $C_{(N-b)}$ strings can also have the following non-balanced (e.g. $N_0 = 6, N_1 = 4$ for $C_{(N-2)}$) form

$$\begin{aligned} C_{(N-1)}^{(3,1)} &= [000], \\ C_{(N-2)}^{(10,2)} &= [01|0000|1110], \\ C_{(N-3)}^{(19,3)} &= [01|0000|12|1111|2221020], \\ C_{(N-4)}^{(30,4)} &= [01|0000|12|1111|23|2222|333102132030], \\ C_{(N-5)}^{(43,5)} &= [01|0000|12|1111|23|2222|34|3333|444|1021324303142040], \\ C_{(N-6)}^{(58,6)} &= [01|0000|12|1111|23|2222|34|3333|45|4444|555|1021324354031425041520530], \\ C_{(N-7)}^{(75,7)} &= [01|0000|12|1111|23|2222|34|3333|45|4444|56|5555|666102132435465031425360415262064051630], \\ C_{(N-8)}^{(94,8)} &= [01|0000|12|1111|23|2222|34|3333|45|4444|56|5555|67|6666| \\ &\quad 7771021324354657603142536470415263720750516273061740], \\ &\quad \dots \\ C_{(N-b)}^{(b^2+4b-2,b)} &= [01|0000|12|1111|2 \dots b_{-1}|b_{-2}b_{-2}b_{-2}b_{-2}|b_{-1}b_{-1}b_{-1}1021 \dots b_{-1}b_{-2} \dots 0]. \end{aligned} \tag{A25}$$

References

1. Wootters WK, Zurek WH. A single quantum cannot be cloned. *Nature*. 1982 Oct;299(5886):802-3. Available from: <http://www.nature.com/articles/299802a0>.

2. Marshall SM, Murray ARG, Cronin L. A probabilistic framework for identifying biosignatures using Pathway Complexity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2017 Dec;375(2109):20160342. Available from: <https://royalsocietypublishing.org/doi/10.1098/rsta.2016.0342>.

3. Imari Walker S, Cronin L, Drew A, Domagal-Goldman S, Fisher T, Line M. Probabilistic Biosignature Frameworks. In: Meadows V, Arney G, Schmidt B, Des Marais DJ, editors. *Planetary Astrobiology*. University of Arizona Press; 2019. p. 1-1. Available from: <https://uapress.arizona.edu/book/planetary-astrobiology>.

4. Meadows VS, Arney GN, Schmidt BE, Des Marais DJ, editors. *Planetary astrobiology*. University of Arizona space science series. Tucson: The University of Arizona Press ; Houston : Lunar and Planetary Institute; 2020. OCLC: 1151198948.

5. Liu Y, Mathis C, Bajczyk MD, Marshall SM, Wilbraham L, Cronin L. Exploring and mapping chemical space with molecular assembly trees. *Science Advances*. 2021 Sep;7(39):eabj2465. Available from: <https://www.science.org/doi/10.1126/sciadv.abj2465>.

6. Marshall SM, Mathis C, Carrick E, Keenan G, Cooper GJT, Graham H, et al. Identifying molecules as biosignatures with assembly theory and mass spectrometry. *Nature Communications*. 2021 May;12(1):3033. Available from: <https://www.nature.com/articles/s41467-021-23258-x>.

7. Marshall SM, Moore DG, Murray ARG, Walker SI, Cronin L. Formalising the Pathways to Life Using Assembly Spaces. *Entropy*. 2022 Jun;24(7):884. Available from: <https://www.mdpi.com/1099-4300/24/7/884>.

8. Sharma A, Czégel D, Lachmann M, Kempes CP, Walker SI, Cronin L. Assembly theory explains and quantifies selection and evolution. *Nature*. 2023 Oct;622(7982):321-8. Available from: <https://www.nature.com/articles/s41586-023-06600-9>.

9. Jirasek M, Sharma A, Bame JR, Mehr SHM, Bell N, Marshall SM, et al. Investigating and Quantifying Molecular Complexity Using Assembly Theory and Spectroscopy. *ACS Central Science*. 2024 May;10(5):1054-64. Available from: <https://pubs.acs.org/doi/10.1021/acscentsci.4c00120>.

10. Łukaszyk S, Bieniawski W. Assembly Theory of Binary Messages. *Mathematics*. 2024 May;12(10):1600. Available from: <https://www.mdpi.com/2227-7390/12/10/1600>.

11. Raubitzek S, Schatten A, König P, Marica E, Eresheim S, Mallinger K. Autocatalytic Sets and Assembly Theory: A Toy Model Perspective. *Entropy*. 2024 Sep;26(9):808. Available from: <https://www.mdpi.com/1099-4300/26/9/808>.

12. Łukaszyk S. On the "Assembly Theory and its Relationship with Computational Complexity"; 2024. Available from: <https://www.preprints.org/manuscript/202412.1492/v1>.

13. Patarroyo KY, Sharma A, Seet I, Packmore I, Walker SI, Cronin L. Quantifying the Complexity of Materials with Assembly Theory. *arXiv*; 2025. ArXiv:2502.09750. Available from: <http://arxiv.org/abs/2502.09750>.

14. Masierak P. Computational Complexity of Determining the Assembly Index; 2025. Available from: <https://www.researchgate.net/doi/10.13140/RG.2.2.19762.88005>.

15. Ziv J, Lempel A. Compression of individual sequences via variable-rate coding. IEEE Transactions on Information Theory. 1978 Sep;24(5):530-6. Available from: <https://ieeexplore.ieee.org/document/1055934>.
16. Storer JA, Szymanski TG. Data compression via textual substitution. Journal of the ACM. 1982 Oct;29(4):928-51. Available from: <https://dl.acm.org/doi/10.1145/322344.322346>.
17. Welch. A Technique for High-Performance Data Compression. Computer. 1984 Jun;17(6):8-19. Available from: <https://ieeexplore.ieee.org/document/1659158>.
18. Charikar M, Lehman E, Liu D, Panigrahy R, Prabhakaran M, Sahai A, et al. The Smallest Grammar Problem. IEEE Transactions on Information Theory. 2005 Jul;51(7):2554-76. Available from: <http://ieeexplore.ieee.org/document/1459058/>.
19. Kieffer JC, En-Hui Yang. Grammar-based codes: a new class of universal lossless source codes. IEEE Transactions on Information Theory. 2000 May;46(3):737-54. Available from: <http://ieeexplore.ieee.org/document/841160/>.
20. Kieffer J, En-hui Yang, Park T, Yakowitz S. Complexity of preprocessor in MPM data compression system. In: Proceedings DCC '98 Data Compression Conference (Cat. No.98TB100225). Snowbird, UT, USA: IEEE Comput. Soc; 1998. p. 554. Available from: <http://ieeexplore.ieee.org/document/672292/>.
21. Lehman E. Approximation Algorithms for Grammar-Based Data Compression [Ph.D. Thesis]. Massachusetts Institute of Technology (MIT); 2002. Available from: https://compression.ru/download/articles/grammar/lehman_phd_2002_approximation_algorithms.pdf.
22. Kieffer JC, Yang Eh. Compression and Explanation using Hierarchical Grammars. The Computer Journal. 2000;43(3):212-22. Available from: https://www.researchgate.net/publication/2826982_1_INTRODUCTION_Compression_and_Explanation_using_Hierarchical_Grammars.
23. Kieffer JC, En-Hui Yang, Nelson GJ, Cosman P. Universal lossless compression via multilevel pattern matching. IEEE Transactions on Information Theory. 2000 Jul;46(4):1227-45. Available from: <http://ieeexplore.ieee.org/document/850665/>.
24. Kieffer J, Flajolet P, Yang Eh. Universal Lossless Data Compression Via Binary Decision Diagrams. arXiv; 2011. ArXiv:1111.1432. Available from: <http://arxiv.org/abs/1111.1432>.
25. Nevill-Manning CG. Compression and Explanation using Hierarchical Grammars. The Computer Journal. 1997 Feb;40(2 and 3):103-16. Available from: https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/40.2_and_3.103.
26. Larsson NJ, Moffat A. Offline dictionary-based compression. In: Proceedings DCC'99 Data Compression Conference (Cat. No. PR00096); 1999. p. 296-305. ISSN: 1068-0314. Available from: <https://ieeexplore.ieee.org/document/755679>.
27. Larsson NJ, Moffat A. Off-line dictionary-based compression. Proceedings of the IEEE. 2000 Nov;88(11):1722-32. Available from: <http://ieeexplore.ieee.org/document/892708/>.
28. Nevill-Manning C, Witten I. Compression and Explanation using Hierarchical Grammars. The Computer Journal. 1999;40(2). Source: CiteSeer.
29. Nevill-Manning CG, Witten IH. Identifying Hierarchical Structure in Sequences: A linear-time algorithm. Journal of Artificial Intelligence Research. 1997 Sep;7:67-82. Available from: <https://jair.org/index.php/jair/article/view/10192>.
30. Apostolico A, Lonardi S. Off-line compression by greedy textual substitution. Proceedings of the IEEE. 2000 Nov;88(11):1733-44. Available from: <http://ieeexplore.ieee.org/document/892709/>.
31. Apostolico A, Lonardi S. Compression of biological sequences by greedy off-line textual substitution. In: Proceedings DCC 2000. Data Compression Conference. Snowbird, UT, USA: IEEE Comput. Soc; 2000. p. 143-52. Available from: <http://ieeexplore.ieee.org/document/838154/>.
32. Sakamoto H, Maruyama S, Kida T, Shimozono S. A Space-Saving Approximation Algorithm for Grammar-Based Compression. IEICE Transactions on Information and Systems. 2009;E92-D(2):158-65. Available from: http://www.jstage.jst.go.jp/article/transinf/E92.D/2/E92.D_2_158/_article.
33. Takabatake Y, I T, Sakamoto H. A Space-Optimal Grammar Compression. LIPIcs, Volume 87, ESA 2017. 2017;87:67:1-67:15. Available from: <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2017.67>.
34. Pagel S, Sharma A, Cronin L. Mapping Evolution of Molecules Across Biochemistry with Assembly Theory; 2024. Available from: <https://arxiv.org/abs/2409.05993>.
35. Knuth DE. The art of computer programming. Volume 2: Seminumerical algorithms / Donald E. Knuth (Stanford University). Third edition, forthy-first printing ed. Boston: Addison-Wesley; 2021. .

36. Clift NM. Calculating optimal addition chains. *Computing*. 2011 Mar;91(3):265-84. Available from: <http://link.springer.com/10.1007/s00607-010-0118-8>.
37. Cronin L. Exploring assembly index of strings is a good way to show why assembly & entropy are intrinsically different.; 2024. Accessed: 2025-12-28. <https://x.com/leecronin/status/1850289225935257665>.
38. Łukaszyk S. 15. In: *Black Hole Horizons as Patternless Binary Messages and Markers of Dimensionality*. Nova Science Publishers; 2023. p. 317-74. Available from: <https://novapublishers.com/shop/future-relativity-gravitation-cosmology/>.
39. Łukaszyk S. Life as the Explanation of the Measurement Problem. *Journal of Physics: Conference Series*. 2024 Feb;2701(1):012124. Available from: <https://dx.doi.org/10.1088/1742-6596/2701/1/012124>.
40. Łukaszyk S. Black hole merger as an event converting two qubits into one. *Frontiers in Quantum Science and Technology*. 2025 Oct;4:1656200. Available from: <https://www.frontiersin.org/articles/10.3389/frqst.2025.1656200/full>.
41. Gabric D, Shallit J, Zhong XF. Avoidance of split overlaps. *Discrete Mathematics*. 2021 Feb;344(2):112176. Available from: <https://linkinghub.elsevier.com/retrieve/pii/S0012365X20303629>.
42. Guibas LJ, Odlyzko AM. String overlaps, pattern matching, and nontransitive games. *Journal of Combinatorial Theory, Series A*. 1981 Mar;30(2):183-208. Available from: <https://linkinghub.elsevier.com/retrieve/pii/0097316581900054>.
43. Ozelim L, Uthamacumaran A, Abrahão FS, Hernández-Orozco S, Kiani NA, Tegnér J, et al.. Assembly Theory Reduced to Shannon Entropy and Rendered Redundant by Naive Statistical Algorithms. *arXiv*; 2025. Available from: <http://arxiv.org/abs/2408.15108>.
44. Abrahão FS, Hernández-Orozco S, Kiani NA, Tegnér J, Zenil H. Assembly Theory is an approximation to algorithmic complexity based on LZ compression that does not explain selection or evolution. *PLOS Complex Systems*. 2024;1(1):e0000014. Available from: <https://journals.plos.org/complexsystems/article?id=10.1371/journal.pcsy.0000014>.
45. Uthamacumaran A, Abrahão FS, Kiani NA, Zenil H. On the salient limitations of the methods of assembly theory and their classification of molecular biosignatures. *npj Systems Biology and Applications*. 2024 Aug;10(1):82. Available from: <https://www.nature.com/articles/s41540-024-00403-y>.
46. Vimal D, Parzych G, Smith OM, Parkar D, Bergen S, Daymude JJ, et al.. Open, Reproducible Calculation of Assembly Indices. *arXiv*; 2025. ArXiv:2507.08852 version: 1. Available from: <http://arxiv.org/abs/2507.08852>.
47. Flamm C, Merkle D, Stadler PF. Assembly in Directed Hypergraphs. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2025 Oct;481(2324):20250331. Available from: <https://royalsocietypublishing.org/doi/10.1098/rspa.2025.0331>.
48. Kempes CP, Lachmann M, Iannaccone A, G MF, M RC, Walker SI, et al. Assembly theory and its relationship with computational complexity. *npj Complexity*. 2025;2(1):27.
49. Gebhard TD, Bell A, Gong J, Hastings JJA, Fricke GM, Cabrol N, et al. Inferring molecular complexity from mass spectrometry data using machine learning. In: *Machine Learning and the Physical Sciences workshop, NeurIPS 2022*; 2022. .
50. Cronin L, Parra JCM, Patarroyo KY. Assembly Addition Chains. *arXiv*; 2025. Available from: <https://arxiv.org/abs/2512.18030>.
51. Krzyżanowski W. Procesy ewolucji kulturowej muzyki w środowisku technologii cyfrowych [Rozprawa doktorska]. Poznań: Uniwersytet im. Adama Mickiewicza w Poznaniu, Wydział Nauk o Sztuce; 2025. Praca doktorska napisana pod kierunkiem prof. UAM dr hab. Piotra Podlipniaka, złożona w 2025 r.
52. Vopson MM. The second law of infodynamics and its implications for the simulated universe hypothesis. *AIP Advances*. 2023 Oct;13(10):105308. Available from: <https://pubs.aip.org/adv/article/13/10/105308/2915332/The-second-law-of-infodynamics-and-its>.
53. Mugur-Schachter M. On a Crucial Problem in Probabilities and Solution. *arXiv*; 2008. Available from: <https://arxiv.org/abs/0801.2654>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.