The Capacity of String-Duplication Systems

Farzad Farnoud (Hassanzadeh)
Electrical Engineering
California Institute of Technology
Pasadena, CA 91125, U.S.A.
farnoud@caltech.edu

Moshe Schwartz
Electrical and Computer Engineering
Ben-Gurion University of the Negev
Beer Sheva 8410501, Israel
schwartz@ee.bgu.ac.il

Jehoshua Bruck
Electrical Engineering
California Institute of Technology
Pasadena, CA 91125, U.S.A.
bruck@paradise.caltech.edu

Abstract—It is known that the majority of the human genome consists of repeated sequences. Furthermore, it is believed that a significant part of the rest of the genome also originated from repeated sequences and has mutated to its current form. In this paper, we investigate the possibility of constructing an exponentially large number of sequences from a short initial sequence and simple duplication rules, including those resembling genomic duplication processes. In other words, our goal is to find out the capacity, or the expressive power, of these string-duplication systems. Our results include the exact capacities, and bounds on the capacities, of four fundamental string-duplication systems.

I. INTRODUCTION

More than 50% of the human genome consists of repeated sequences [6]. An important class of these repeated sequences are interspersed repeats, which are caused by transposons. A transposon, or a “jumping gene”, is a segment of DNA that can “copy and paste” or “cut and paste” itself into new positions of the genome. Currently, 45% of the human genome is known to consist of transposon-driven repeats [6].

A second type of repeats are tandem repeats, generally thought to be caused by slipped-strand mispairings [11]. A slipped-strand mispairing is said to occur when, during DNA synthesis, one strand in a DNA duplex becomes misaligned with the other. These mispairings may lead to deletions or insertion of a repeated sequence [9]. While tandem repeats are known to constitute only 3% of the human genome, they cause important phenomena such as chromosome fragility, expansion diseases, silencing genes [12], and rapid morphological variation [4].

While interspersed repeats and random repeats together account for a significant part of the human genome, it is likely that a substantial portion of the unique genome, the part that is not known to contain repeated sequences, also has its origins in ancient repeated sequences that are no longer recognizable due to change over time [6], [12].

Motivated by the prevalence and the significance of repeated sequences and the fact that much of our unique DNA was likely originally repeated sequences, in this paper we study the capacity of string-duplication systems with simple duplication rules including rules that resemble the repeat-producing genomic processes, namely duplication of transposons and duplication caused by slipped-strand mispairings. A string-duplication system, to be defined formally later, consists of a set of rewriting rules, an initial sequence, and all sequences that can be obtained by applying the rules to the initial sequence a finite number of times. The notion of capacity, defined later in the paper, represents the average number of bits per symbol that can asymptotically be encoded by the sequences in a string-duplication system, and thus illustrates the expressive power and the diversity of that system.

In this paper, we consider four duplication rules. The first is the end duplication rule, which allows substrings of a certain length k to be appended to the end of previous sequences. For example, if k = 3 we may construct the sequence TCATGC from TCATG. While this rule is not biologically motivated, we present it first because of the simplicity of proving the related results. In particular, we show that nearly all sequences with the same alphabet as the initial sequence can be generated with this rule.

The second rule is called tandem duplication and allows a substring of length k to be duplicated next to its original position. For example, for k = 3, from the sequence TCATG, one can generate TCATCATGC. We show that this rule has capacity zero regardless of the initial sequence. However, if one allows substrings of all length larger than a given value to be copied, the capacity becomes positive except in trivial cases.

The third rule is reversed tandem duplication, which is similar to tandem duplication except that the copy is reversed before insertion. For instance, in our previous example, the sequence TCATTAGC can be generated. Here, the capacity is zero only in the trivial case in which the initial sequence consists of only one unique symbol.

The last rule is duplication with a gap, where the copy of a substring of a given length k can be inserted after k′ symbols. This rule is motivated by the fact that transposons may insert themselves in places far from their original positions. As an example, for k = 3 and k′ = 1, from TCATG, one can obtain TCATGCATC. For this rule, we show that the capacity is zero if and only if the initial sequence is periodic with period equal to the greatest common divisor of k and k′.

We note that tandem duplication has been already studied in a series of papers [1], [2], [7], [8]. However, this was done in the context of the theory of formal languages, and the goal of these studies was mainly to determine their place in the
Chomsky hierarchy of formal languages.

In the next section, we present the preliminaries and in the following four sections, we present the results for each of the aforementioned duplication rules. Due to lack of space, we omit some proofs or only provide their outline. Complete proofs can be found in the full version of this paper, available on arXiv [3].

II. PRELIMINARIES

Let $\Sigma$ be some finite alphabet. We recall some useful notation commonly used in the theory of formal languages. An $n$-string $x = x_1x_2\ldots x_n \in \Sigma^n$ is a finite sequence of alphabet symbols, $x_i \in \Sigma$. We say $n$ is the length of $x$ and denote it by $|x| = n$. For two strings, $x \in \Sigma^n$ and $y \in \Sigma^m$, their concatenation is denoted by $xy \in \Sigma^{n+m}$. For a positive integer $m$ and a string $s, s^m$ denotes the concatenation of $m$ copies of $s$. The set of all finite strings over the alphabet $\Sigma$ is denoted by $\Sigma^*$. We say $v \in \Sigma^*$ is a substring of $x$ if $x = uvw$, where $u, w \in \Sigma^*$. The alpha-representation of a string $s$, denoted by $R(s)$, is the set of all letters from $\Sigma$ making up $s$. Thus, $R(s) \subseteq \Sigma$. The alpha-diversity of $s$ is the size of the alpha-representation of $s$, denoted by $\delta(s) = |R(s)|$. Furthermore, let the number of occurrences of a symbol $a \in \Sigma$ in a sequence $s \in \Sigma^*$ be denoted by $n_s(a)$.

For $S \subseteq \Sigma^*$, we let $S^* = \{w_1w_2\ldots w_m \mid w_i \in S, m \geq 0\}$, whereas $S^+ = \{w_1w_2\ldots w_m \mid w_i \in S, m \geq 1\}$. For any $x \in \Sigma^*$, $|x| = n \geq m$, the $m$-suffix of $x$ is $w \in \Sigma^m$, such that $x = uvw$ for some $v \in \Sigma^*$. Similarly, the $m$-prefix of $x$ is $u \in \Sigma^m$, where $x = uv$ for some $v \in \Sigma^*$. A string system $S$ is a subset $S \subseteq \Sigma^*$. For any integer $n$, we denote by $N_S(n)$ the number of length $n$ strings in $S$, i.e., $N_S(n) = |S \cap \Sigma^n|$. The capacity of a string system $S$ is defined by

$$\text{cap}(S) = \limsup_{n \to \infty} \frac{\log_2 N_S(n)}{n}.$$

A string-duplication system is a tuple $S = (\Sigma, s, T)$, where $\Sigma$ is a finite alphabet, $s \in \Sigma^*$ is a finite string (which we will use to start the duplication process), and $T$ is a set of functions such that each $T \in T$ is a mapping from $\Sigma^*$ to $\Sigma^*$ that defines a string-duplication rule. The resulting string system $S$, induced by $(\Sigma, s, T)$, is defined as the closure of the string-duplication functions $T$ on the initial string set $\{s\}$, i.e., $S$ is the minimal set for which $s \in S$, and for each $s' \in S$ and $T \in T$ we also have $T(s') \in S$.

III. END DUPLICATION

We define the end-duplication function, $T_{\text{end}} : \Sigma^* \to \Sigma^*$, as follows:

$$T_{\text{end}}(x) = \begin{cases} uvwvw & \text{if } x = uvw, |u| = i, |v| = k \\ x & \text{otherwise.} \end{cases}$$

We also define two sets of these functions which will be used later:

$$T_{\text{end}} = \{T_{\text{end}} \mid i \geq 0\}, \quad T_{\text{end}} = \{T_{\text{end}} \mid i \geq 0, k' \geq k\}.$$

Intuitively, in the end-duplication system, the transformations duplicate a substring of length $k$ and append the duplicate substring to the end of the original string.

**Theorem 1.** Let $\Sigma$ be any finite alphabet, $k \geq 1$ any integer, and $s \in \Sigma^*, |s| \geq k$. Then for $S_{\text{end}}^{\text{end}} = (\Sigma, s, T_{\text{end}})$, we have $\text{cap}(S_{\text{end}}^{\text{end}}) = \log_2 \delta(s)$.

**Proof:** (Sketch). First we note that by requiring $|s| \geq k$ we avoid the degenerate case of $S_{\text{end}}^{\text{end}}$ containing only $s$. We further note that, by the definition of the duplication functions, $R(x) = R(T_{\text{end}}^k(x))$ for all non-negative integers $i$ and $k$, and thus, all the strings in $S_{\text{end}}^{\text{end}}$ have the same alpha-representation. Thus, trivially, $\text{cap}(S_{\text{end}}^{\text{end}}) \leq \log_2 \delta(s)$.

We now turn to prove the inequality in the other direction. It can be shown that after $2k$ duplication steps we can obtain from any $x \in \Sigma^*, |x| \geq k$, a string $x'$ with any given $k$-suffix $w$, provided $R(w) \subseteq R(x)$. Thus, from the initial string $s$, we can obtain a string $s'$ with all of the strings of $R(s)^k$ appearing as $k$-substrings, using at most $2k\delta(s)^k$ duplication steps\(^1\), i.e.,

$$|s'| \leq |s| + 2k^2 \delta(s)^k.$$

After having obtained $s'$, each duplication may duplicate any of the $k$-strings in $R(s)^k$ in a single operation. Thus, for all $n = |s'| + tk$, $t$ a non-negative integer, the number of distinct strings in $S_{\text{end}}^{\text{end}}(n)$ is bounded from below by

$$N_{S_{\text{end}}^{\text{end}}}(n) \geq \delta(s)^n - |s'|.$$

Since $|s'|$ is a constant, we have $\text{cap}(S_{\text{end}}^{\text{end}}) \geq \log_2 \delta(s)$. The following is an obvious corollary.

**Theorem 2.** Let $\Sigma$ be any finite alphabet, $k \geq 1$ any integer, and $s \in \Sigma^*, |s| \geq k$. Then for $S_{\text{end}}^{\text{end}} = (\Sigma, s, T_{\text{end}})$, we have $\text{cap}(S_{\text{end}}^{\text{end}}) = \text{cap}(S_{\text{end}}^{\text{end}}) = \log_2 \delta(s)$.

**Proof:** Since for all $n \geq k$, we have $N_{S_{\text{end}}^{\text{end}}}(n) \leq N_{S_{\text{end}}^{\text{end}}}(n) \leq \delta(s)^n$, the claim follows.

IV. TANDEM DUPLICATION

We now consider different duplication rules, $T_{i,k}^{\text{tan}} : \Sigma^* \to \Sigma^*$, defined by

$$T_{i,k}^{\text{tan}}(x) = \begin{cases} uvwvw & \text{if } x = uvw, |u| = i, |v| = k \\ x & \text{otherwise.} \end{cases}$$

We also define the sets $T_{k}^{\text{tan}} = \{T_{i,k}^{\text{tan}} \mid i \geq 0\}, \quad T_{k}^{\text{tan}} = \{T_{i,k'}^{\text{tan}} \mid i \geq 0, k' \geq k\}$.

Unlike the end duplication discussed in the previous section, tandem duplication takes a $k$-substring and duplicates it adjacent to itself in the string. Also, the capacity of tandem-duplication systems is in complete contrast to end-duplication systems.\(^1\)This bound may be improved, but this will not affect the capacity calculation.
Theorem 3. Let $\Sigma$ be any finite alphabet, $k$ any positive integer, and $s < \Sigma^*$, with $|s| \geq k$. Then for $S_k^\lambda = (\Sigma, s, T_k^\lambda)$, we have $\operatorname{cap}(S_k^\lambda) = 0$.

Proof: Consider any $n$-string $x \in \Sigma^*$, $n \geq k$. Instead of viewing $x = x_1 x_2 \ldots x_n$ as a sequence of $n$ symbols from $\Sigma$, we can, by abuse of notation, view it as a sequence of $n - k + 1$ overlapping $k$-substrings $x = x'_1 x'_2 \ldots x'_{n-k+1}$, where

$$x'_i = x_{i+k-1} \ldots x_{i+1}.$$

For a $k$-string $y = y_1 y_2 \ldots y_k$, $y \in \Sigma$, its cyclic shift by one position is denoted by $E y = y_2 y_3 \ldots y_k y_1$. A cyclic shift by $j$ positions is denoted by $E^j y = y_{j+1} y_{j+2} \ldots y_k y_1 y_2 \ldots y_j$.

We say two $k$-strings, $y, z \in \Sigma^k$, are cyclically equivalent if $y = E^j z$, for some integer $j$. Clearly this is an equivalence relation. Let $\phi(y)$ denote the equivalence class of $y$. If $y$ and $z$ are cyclically equivalent, then $\phi(y) = \phi(z)$.

We now define

$$\Phi(x) = \phi(x'_1) \phi(x'_2) \ldots \phi(x'_{n-k+1}),$$

i.e., $\Phi(x)$ is the image of the overlapping $k$-substrings of $x$ under $\phi$. We also observe that knowing $x'_1$ and $\Phi(x)$ enables a full reconstruction of $x$.

At this point we turn to consider the effect of the duplication $T_k^\lambda$ on a string $x \in \Sigma^*$, $|x| \geq k$. When viewed as a sequence of overlapping $k$-substrings, as defined above,

$$T_k^\lambda(x) = x'_1 \ldots x'_{n-k+1} x'_1 E x'_2 \ldots E^{k-1} x'_1 x'_2 x'_{n-k+1} \ldots x'_{n-k+1}.$$

Since $\phi(x'_1) = \phi(E(x'_1))$ for all $j$, we have

$$\Phi(T_k^\lambda(x)) = \phi(x'_1) \ldots \phi(x'_{n-k+1}) \phi(x'_1) \ldots \phi(x'_{n-k+1}) \phi(x'_{n-k+1}) \ldots \phi(x'_{n-k+1}),$$

where $\phi(x'_1)$ appears $k + 1$ consecutive times.

Thus, we may think of $\phi(x'_1)$ as a bin, and the action of $T_k^\lambda$ as throwing $k$ balls into the bin $\phi(x'_1)$. The number of bins does not change throughout the process, and is equal to one more than the number of times $\phi(x'_1) \neq \phi(x'_{n-k+1})$, where $x = s$ is the original string. If $b$ is the number of bins defined by $s$, then the number of strings obtained by $m$ duplications is exactly $(b + m - 1)$. Since this number grows only polynomially in the length of the resulting string, we have $\operatorname{cap}(S_k^\lambda) = 0$.

When considering $S_k^\lambda = (\Sigma, s, T_k^\lambda)$ the situation appears to be harder to analyze.

Theorem 4. For any finite alphabet $\Sigma$, and any string $s \in \Sigma^*$ of nontrivial alpha-diversity, $\delta(s) \geq 2$, we have

$$\operatorname{cap}(S_k^\lambda) \geq \log_2 (r + 1),$$

where $r$ is the largest (real) root of the polynomial

$$f(x) = x^{\delta(s)} - \sum_{i=0}^{\delta(s)-2} x^i.$$

Proof: The proof strategy is the following: we shall show that $S_k^\lambda$ contains, among other things, a regular language. The capacity of that regular language will serve as the lower bound we claim.

For the first phase of the proof, assume $i_1 < i_2 < \ldots < i_{\delta(s)}$ are the indices of $\delta(s)$ distinct alphabet symbols in $s$. We produce a sequence of strings, $s_0, s_1, \ldots, s_{\delta(s)-1}$, where $s_0 = s$, defined iteratively by

$$s_j = T_k^\lambda_{i_{\delta(s)}-j} \ldots T_k^\lambda_{i_2-j} \ldots T_k^\lambda_{i_1-j}(s_{j-1}),$$

for $j = 1, 2, \ldots, \delta(s) - 1$. After this set of steps, the $\delta(s)$-substring starting at position $i_{\delta(s)}$ of $s_{\delta(s)-1}$ contains $\delta(s)$ distinct symbols. In what follows we will only use these symbols for duplication, and thus, the constant amount of other symbols in $s_{\delta(s)-1}$ does not affect the capacity calculation.

Thus, for ease of presentation we shall assume from now on that $|s| = \delta(s)$, i.e., the initial string contains no repeated symbol from the alphabet. Furthermore, with- out loss of generality, let us assume these symbols are $a_{\delta(s)}, a_{\delta(s)-1}, \ldots, a_1$, in this order.

We now perform the following iterations: In iteration $i$, where $i = \delta(s), \delta(s) - 1, \ldots, 2$, we duplicate $i$-substrings equal to $a_{i} a_{i-1} \ldots a_{1}$ $a_1$. As a final iteration, we may duplicate 1-substrings without constraining their content. It is easy to verify the resulting strings form the following regular language.

$$S = \left( a_1^{+} a_{\delta(s)}^{+} \left( \ldots \left( a_1^{+} a_1^{+} \right)^{+} \right)^{+} \right)^{+}. $$

The construction process implies $S \subseteq S_k^\lambda$.

![Figure 1](image-url)

Figure 1. The finite-state automaton accepting the regular language used in the proof of Theorem 4.

The finite-state automaton accepting $S$ is depicted in Figure 1. The graph is primitive and lossless, and thus, for the purpose of calculating the capacity, instead of counting the number of length $n$ words in $S$, we can count the number of length $n$ paths in the automaton graph $G$ (see [5], [10]). By Perron-Frobenius theory,

$$\operatorname{cap}(S_k^\lambda) \geq \operatorname{cap}(S) = \log_2 \lambda(A_G).$$
Lemma 7. We have the trivial upper bound provided in the following theorem, stated without proof.

Corollary 5. We have the trivial upper bound \( \text{cap}(S_{\text{ran}}^{\text{cap}}) \leq \log_2 |\Sigma| = 1 \).

Proof: By applying Theorem 4 we get \( \text{cap}(S_{\text{ran}}^{\text{cap}}) \geq 1 \). We also have the trivial upper bound \( \text{cap}(S_{\text{ran}}^{\text{cap}}) \leq \log_2 r > 0 \), where \( r \) is the largest root of the polynomial \( f(x) = x^k + x - 1 \).

V. REVERSED TANDEM DUPLICATION

Consider the reversed tandem duplication rule \( T_{i,k}^r : \Sigma^* \rightarrow \Sigma^* \) defined as

\[
T_{i,k}^r(x) = \begin{cases} 
uvwvw & \text{if } x = uvwx, \ |u| = i, \ |v| = k, \\
\text{otherwise}
\end{cases}
\]

where \( y^k \) is the reverse of \( y \), i.e., \( y^k = y_ny_{n-1} \ldots y_1 \) for a sequence \( y = y_1y_2 \ldots y_m \in \Sigma^* \). Furthermore, let

\[
T_{i,k}^r = \{ \ T_{i,k}^r \mid \ i \geq 0 \}
\]

and use \( S_{\text{ran}}^{\text{cap}} = (\Sigma, s, T_{i,k}^r) \). Since the starting string \( s \) will play a crucial role, we shall often use the notation \( S_{i,k}^r(s) \).

Lemma 7. Let \( s \in \Sigma^k \) with \( s \neq s^k \). Then \( \text{cap}(S_{i,k}^r(s)) \geq 1/k \).

Proof: By repeatedly applying duplication to the last block of \( k \) symbols, we can create any sequence of alternating blocks \( s \) and \( s^k \), starting with \( s \). To extend any run of \( s \), except the first one, (resp. any run of \( s^k \) we can apply duplication to the last block of the previous run, which is an \( s^k \) block (resp. \( s \)). Thus, the regular language \( S = s^k \{ s, s^k \}^* \), satisfies \( S \subseteq S_{i,k}^r(s) \) and thus \( \text{cap}(S_{i,k}^r(s)) \geq \text{cap}(S) \). Furthermore, since \( s \neq s^k \), we see that \( \text{cap}(S) = 1/k \).

Note that the requirement that \( s \neq s^k \) implies that \( k \geq 2 \).

The following theorem states that the capacity of reversed tandem duplication is positive except in trivial cases.

Theorem 8. For any \( s \in \Sigma^* \), \( |s| \geq k \), we have \( \text{cap}(S_{i,k}^r(s)) = 0 \) if and only if \( \text{cap}(s) = 1 \).

Proof: (Sketch). It is clear that if \( \text{cap}(s) = 1 \), then \( \text{cap}(S_{i,k}^r(s)) = 0 \). For the other direction, suppose that \( \text{cap}(S_{i,k}^r(s)) = 0 \). We show that \( \text{cap}(s) = 1 \). We only prove this for \( |s| = k \). Denote \( s = s_1s_2 \ldots s_k \), with \( s_i \in \Sigma \), and let \( t = s_2s_3 \ldots s_k \). It is not difficult to see that since \( \text{cap}(S_{i,k}^r(s)) = 0 \), we have \( \text{cap}(S_{i,k}^r(t)) = 0 \). Hence, applying Lemma 7 to \( s \) and \( t \) implies that \( s = s^k \) and \( t = t^k \). From these, it can be shown that \( \text{cap}(s) = 1 \).

Theorem 9. For all \( s \in \Sigma^* \), \( |s| \geq k \), \( \text{cap}(S_{i,k}^r(s)) \) depends on \( s \) only through \( \text{cap}(s) \).

Proof: Consider two sequences \( s, t \in \Sigma^* \), \( |s| = |t| \geq k \), such that \( \text{cap}(s) = \text{cap}(t) \). Since the identity of the symbols is irrelevant to the capacity, we may assume that \( R(s) = R(t) \). By appropriate duplications, it is easy to find a sequence \( t' \in S_{i,k}^r(t) \) such that for all \( a \in \Sigma \), we have \( n_a(t') \geq n_a(t) \). We then apply Lemma 9 and show that \( s \) is a substring of some sequence \( t'' \in S_{i,k}^r(t') \). Hence, \( \text{cap}(S_{i,k}^r(s)) \leq \text{cap}(S_{i,k}^r(t')) \leq \text{cap}(S_{i,k}^r(t)) \). Similarly, we can show that \( \text{cap}(S_{i,k}^r(t')) \leq \text{cap}(S_{i,k}^r(s)) \). Hence, \( \text{cap}(S_{i,k}^r(s)) = \text{cap}(S_{i,k}^r(t)) \).

VI. DUPLICATION WITH A GAP

Consider the duplication-with-a-gap rule \( T_{i,k,k'}^\text{gap} : \Sigma^* \rightarrow \Sigma^* \) defined as

\[
T_{i,k,k'}^\text{gap}(x) = \begin{cases} 
uvwvz & \text{if } x = uvwx, \ |u| = i, \ |v| = k, \\
\text{otherwise}
\end{cases}
\]

where \( y^k \) is the reverse of \( y \), i.e., \( y^k = y_ny_{n-1} \ldots y_1 \) for a sequence \( y = y_1y_2 \ldots y_m \in \Sigma^* \). Furthermore, let

\[
T_{i,k,k'}^\text{gap} = \{ \ T_{i,k,k'}^\text{gap} \mid \ i \geq 0 \}
\]

and use \( S_{i,k,k'}^\text{gap} = (\Sigma, s, T_{i,k,k'}^\text{gap}) \). The following lemma, presented without proof, is useful for characterizing the set of sequences \( s \) with \( \text{cap}(S_{i,k,k'}^\text{gap}(s)) > 0 \).
Lemma 11. For all $s \in \Sigma^*$ such that $|s| \geq k + k'$, we have
$$\text{cap}(S_{k,k'}^\text{gap}(s)) \geq \frac{1}{k} \log_2 \left(1 + d_H \left(s_{1,k}, (s^2)_{k+1,k}\right)\right).$$

The next corollary is an immediate result of the lemma.

Corollary 12. Assume $\text{cap}(S_{k,k'}^\text{gap}(s)) = 0$, where $s \in \Sigma^*$ and $|s| \geq k + k'$. For any $(k + k')$-substring of $s$, denoted $x_1 \ldots x_k y_1 \ldots y_{k'}$, with $x_i, y_i \in \Sigma$, we have
$$x_1 \ldots x_k = y_1 \ldots y_{k'}, \quad \text{if } k > k',$$
$$x_1 \ldots x_k = y_1 \ldots y_k, \quad \text{if } k \leq k'.$$

This corollary is used in the following theorem.

Theorem 13. For $s \in \Sigma^*$, $|s| \geq k + k'$, we have $\text{cap}(S_{k,k'}^\text{gap}(s)) = 0$ if and only if $s$ is periodic with period $\gcd(k, k')$.

Proof: (Sketch.) We start with the easy direction. Assume $s$ is periodic with period $\gcd(k, k')$. Note that in this case $S_{k,k'}^\text{gap}(s)$ contains only one sequence of length $ik + k'$ for each $i \geq 1$, which is itself a periodic extension of $s$. No other sequences appear in $S_{k,k'}^\text{gap}(s)$. Thus, the capacity is 0.

We now turn to the other direction. Assume the capacity is 0. Here we only consider the case of $k > k'$. First, assume $s = x_1 \ldots x_k y_1 \ldots y_{k'}$, with $x_i, y_i \in \Sigma$, has length $k + k'$. Denote $k'' = k - k'$. We show that $s$ is periodic with period $\gcd(k, k')$. From Corollary 12, it follows that $y_1 \ldots y_{k'} = x_1 \ldots x_k$, so we can write $s = x_1 \ldots x_k x_1 \ldots x_{k'}$. Furthermore, said corollary implies that $x_i = x_{k+i}$ for $i \in [k - k']$ and so $s = x_1 \ldots x_k x_1 \ldots x_{k'} x_1 \ldots x_k$. By once applying the rule of $T_{0,k''}$ to $s$ we obtain
$$t = x_1 \ldots x_{k'} x_1 \ldots x_{k'} x_1 \ldots x_{k'} x_1 \ldots x_{k''}.$$