The Universality and Linearity of Compression by Substring Enumeration

Danny Dubé
Université Laval, Canada
Email: Danny.Dube@ift.ulaval.ca

Hidetoshi Yokoo
Gunma University, Japan
Email: yokoo@cs.gunma-u.ac.jp

***Abstract*—A new lossless data compression technique called compression by substring enumeration (CSE) has recently been introduced. Two conjectures have been stated in the original paper and they have not been proved there nor in subsequent papers on CSE. The first conjecture says that CSE is universal for Markovian sources, provided an appropriate predictor is devised. The second one says that CSE has a linear complexity both in time and in space. In this paper, we present an appropriate predictor and demonstrate that CSE indeed becomes universal for any order-k Markovian source. Finally, we prove that the compacted substring tree on which CSE's linear complexity depends effectively has linear size.**

I. Background on CSE

A. Notation

Throughout the paper, we adopt the following conventions: $\mathbb{N}$ denotes the set of natural numbers; ϵ denotes the empty string; a and b denote bits; i, j, k, l, n, and p are in $\mathbb{N}$; u, v, and w are strings in $\{\mathtt{0},\mathtt{1}\}^*$; and $|\cdot|$ is used to obtain the length of a string, or the size of a set, depending on the context.

The data that is sent to the compression by substring enumeration (CSE) compressor is a binary string, denoted by $\mathbf{D}$, of length N bits [1].[1] CSE assumes $\mathbf{D}$ to be circular, and encodes it into the pair of its equivalence class of strings under rotation and its rank, or lexicographic order, in the class. In the literature, such an equivalence class of strings is called a *necklace* [4]. We identify each necklace with the lexicographically smallest string in its equivalence class. In this paper, we concentrate only on the encoding of necklaces, which is the core component of the CSE technique.

B. Occurrences

Of particular importance to CSE is the notion of *occurrences* of a substring in $\mathbf{D}$. It is not exactly the notion that is usually adopted since CSE considers $\mathbf{D}$ to be circular.

First, we define the notion of *occurrence* of a substring *at a given position*. We say that a substring $w \in \{\mathtt{0},\mathtt{1}\}^*$ *occurs at position* p in $\mathbf{D}$, denoted by $w \in_p \mathbf{D}$, if:

$$\exists u, v \in \{\mathtt{0},\mathtt{1}\}^*.\ \exists i \in \mathbb{N}.\ uwv = \mathbf{D}^i \text{ and } 0 \leq |u| = p < N,$$

where $\mathbf{D}^i$ denotes i copies of $\mathbf{D}$ concatenated together. Note that we restrict the positions to lie in the range 0 to $N-1$, inclusively. Without this restriction, for any substring w such that $w \in_p \mathbf{D}$, we would also have $w \in_{p+N} \mathbf{D}$, due to $\mathbf{D}$'s circularity. The restriction is necessary to make the definition of *number of occurrences* sensible, as seen below. Note also that we put *no* restriction of the length of the substrings themselves. In particular, w can be as short as ϵ. At the other extreme, w can be longer than $\mathbf{D}$.

Second, we define the notion of occurrence. We say that a substring w *occurs* in $\mathbf{D}$, denoted by $w \in \mathbf{D}$, if there exists a position p such that $w \in_p \mathbf{D}$.

Third, we define the notion of *number of occurrences*. The *number of occurrences* of a substring w in $\mathbf{D}$, denoted by C_w, is $|\{p \in \mathbb{N} \mid w \in_p \mathbf{D}\}|$. Obviously, we have that $C_w > 0$ if and only if $w \in \mathbf{D}$. This definition makes it clear that we need the restriction on the possible positions of occurrences. Otherwise, C_w could only be 0 or ∞. The following equations are direct consequences of the definition of C_w.

$$C_\epsilon = N \tag{1}$$

$$C_{\mathtt{0}w} + C_{\mathtt{1}w} = C_w = C_{w\mathtt{0}} + C_{w\mathtt{1}}, \quad \forall w \in \{\mathtt{0},\mathtt{1}\}^* \tag{2}$$

$$\sum_{w \in \{\mathtt{0},\mathtt{1}\}^n} C_w = N, \quad \forall n \in \mathbb{N} \tag{3}$$

C. Compression by Substring Enumeration (CSE)

From Eq. (2) above, we can derive

$$\begin{aligned} C_{\mathtt{0}w\mathtt{1}} &= C_{\mathtt{0}w} - C_{\mathtt{0}w\mathtt{0}}, \\ C_{\mathtt{1}w\mathtt{0}} &= C_{w\mathtt{0}} - C_{\mathtt{0}w\mathtt{0}}, \\ C_{\mathtt{1}w\mathtt{1}} &= C_{w\mathtt{1}} - C_{\mathtt{0}w\mathtt{1}}, \end{aligned}$$

which can be used to compute each quantity in the left-hand sides from $C_{\mathtt{0}w\mathtt{0}}$. We combine these equations with $C_{\mathtt{0}w\mathtt{0}} \geq 0$, $C_{\mathtt{0}w\mathtt{1}} \geq 0$, $C_{\mathtt{1}w\mathtt{0}} \geq 0$, and $C_{\mathtt{1}w\mathtt{1}} \geq 0$ to yield

$$\max(0, C_{\mathtt{0}w} - C_{w\mathtt{1}}) \leq C_{\mathtt{0}w\mathtt{0}} \leq \min(C_{\mathtt{0}w}, C_{w\mathtt{0}}). \tag{4}$$

When we already have $C_{\mathtt{0}w}$, $C_{w\mathtt{0}}$, and $C_{w\mathtt{1}}$, we can efficiently transmit $C_{\mathtt{0}w\mathtt{0}}$ using the above bounds. When predicting and encoding $C_{\mathtt{0}w\mathtt{0}}$, we refer to w as the *core* of $\mathtt{0}w\mathtt{0}$. The size of the set of possible values for $C_{\mathtt{0}w\mathtt{0}}$ is given by

$$\begin{aligned} &\min(C_{\mathtt{0}w}, C_{w\mathtt{0}}) - \max(0, C_{\mathtt{0}w} - C_{w\mathtt{1}}) + 1 \\ = \ &\min(C_{\mathtt{0}w}, C_{\mathtt{1}w}, C_{w\mathtt{0}}, C_{w\mathtt{1}}) + 1. \end{aligned} \tag{5}$$

In particular, this means that, if $\min(C_{\mathtt{0}w}, C_{\mathtt{1}w}, C_{w\mathtt{0}}, C_{w\mathtt{1}}) = 0$, then $C_{\mathtt{0}w\mathtt{0}}$ is forced to take a unique value. We say that

[1] Subsequent work on CSE appears in [2], [3] and [7].

1. **Send** N; **Send** C_0; **Send** $\mathrm{rank}(\mathbf{D})$;
2. **For** $l := 2$ **to** N **do**
3. **For** all $w \in I(\mathbf{D})$ such that $|w| = l - 2$ **do**
4. **Predict** and **Send** C_{0w0};

Fig. 1. Pseudo-code for CSE's compression algorithm

such a prediction is *trivial* and that core w is not interesting. We denote by $I(\mathbf{D})$ the set of *interesting* cores as follows.

$$U(\mathbf{D}) = \{w \in \{0,1\}^* \mid w0 \in \mathbf{D} \text{ and } w1 \in \mathbf{D}\} \quad (6)$$

$$V(\mathbf{D}) = \{w \in \{0,1\}^* \mid 0w \in \mathbf{D} \text{ and } 1w \in \mathbf{D}\} \quad (7)$$

$$I(\mathbf{D}) = U(\mathbf{D}) \cap V(\mathbf{D}) \quad (8)$$

Note that when $C_w \leq 1$, we necessarily have that $w \notin I(\mathbf{D})$.

We can summarize these observations into the CSE compression algorithm presented in Figure 1. The double **for** loops of the algorithm suggest that up to $\Theta(N^2)$ numbers of occurrences might have to be transmitted from the compressor to the decompressor. This is not the case. The number of the numbers that CSE transmits is no more than the string length. This is proved in Section IV.

II. Predictions in CSE

The most important operation in CSE is the prediction of the numbers C_{0w0} of occurrences. In earlier experiments, three predictors have been used: a uniform predictor [1], a predictor that learns how to efficiently predict C_{0w0} [1], and a combinatorial predictor. Here, we present the uniform and the combinatorial predictions.

A. Uniform Prediction

Uniform prediction simply consists in assigning the same probability to each possible value of C_{0w0} (or to each value of some other variable). This simple prediction can be used every time we have a lower and an upper bound on the possible values. Given the bounds established for C_{0w0} in Eq. (4), we can uniformly predict C_{0w0} and encode its actual value using $\lg\left(\min(C_{0w}, C_{1w}, C_{w0}, C_{w1}) + 1\right)$ bits.

B. Combinatorial Prediction

Uniform prediction has the advantage of being simple. However, it is too simplistic and, to the best of our knowledge, it cannot make CSE universal. For example, if we have $C_{0w} = C_{1w} = C_{w0} = C_{w1} = 1000$, then, intuitively, we expect $C_{0w0} \approx 500$ to be more probable than $C_{0w0} \approx 1000$.

Combinatorial prediction assigns, to some legal value C_{0w0}, the probability $p_c(C_{0w0} \mid C_{0w}, C_{w0}, C_{w1})$, which is

$$\frac{\binom{C_w}{C_{0w0},\, C_{0w1},\, C_{1w0},\, C_{1w1}}}{\sum_{C_{0w0}=\max(0,C_{0w}-C_{w1})}^{\min(C_{0w},C_{w0})} \binom{C_w}{C_{0w0},\, C_{0w1},\, C_{1w0},\, C_{1w1}}},$$

where C_{0w1}, C_{1w0}, and C_{1w1} should be seen as functions of C_{0w0}. This prediction is inspired by the following picture. Imagine that the C_w occurrences of the core are interspersed in $\mathbf{D}$, without overlaps. Then any C_{0w0} occurrences of w might be the ones that are preceded and followed by 0s; among the remaining occurrences of w, any C_{0w1} of them might be the ones that are preceded by a 0 and followed by a 1; and so on.

The expression for $p_c(C_{0w0} \mid C_{0w}, C_{w0}, C_{w1})$ can be simplified. The numerator and the denominator can be transformed into $\binom{C_w}{C_{0w}}\binom{C_{0w}}{C_{0w0}}\binom{C_{1w}}{C_{1w0}}$ and $\binom{C_w}{C_{0w}}\binom{C_w}{C_{w0}}$, respectively, leading to:

$$p_c(C_{0w0} \mid C_{0w}, C_{w0}, C_{w1}) = \frac{\binom{C_{0w}}{C_{0w0}}\binom{C_{1w}}{C_{1w0}}}{\binom{C_w}{C_{w0}}}. \quad (9)$$

A particularly interesting property of combinatorial prediction is that the probability that is assigned to the joint prediction of all the numbers C_{0vw0}, for a given w, has a very simple form. Indeed, we have the following:

$$\prod_{v\in\{0,1\}^*} p_c(C_{0vw0} \mid C_{0vw}, C_{vw0}, C_{vw1}) = \prod_{v\in\{0,1\}^*} \frac{\binom{C_{0vw}}{C_{0vw0}}\binom{C_{1vw}}{C_{1vw0}}}{\binom{C_{vw}}{C_{vw0}}} = \frac{1}{\binom{C_w}{C_{w0}}}. \quad (10)$$

The closed form of the joint probability is obtained thanks to the telescopic product. Only the denominator of the case $v = \epsilon$ remains. Since we cancel the two factors of the numerator of the probability for a particular v with the denominators of probabilities for other, longer v's, one might wonder if we really get an equality here. In fact, apart from a finite number of vs, all individual probabilities that are multiplied together are equal to 1. Indeed, first note that when v is long enough (e.g., when $|v| \geq N$), we have that $C_{vw} \leq 1$, which causes $vw \notin I(\mathbf{D})$.[2] Then note that, for $vw \notin I(\mathbf{D})$, combinatorial prediction assigns probability 1 to the single legal value of C_{0vw0}.

C. Optimal Switch from Uniform to Combinatorial

The prediction that we propose here combines uniform prediction and combinatorial prediction. Given some length l, CSE ought to use uniform prediction for C_{0w0} when $|w| < l$ and combinatorial prediction otherwise. In order to determine the optimal threshold where to switch from uniform to combinatorial prediction, universal CSE has to evaluate the cost of predicting and encoding each C_{0w0} using each prediction method. It then has to select the length l_{opt} that minimizes the overall cost of prediction and encoding. We define l_{opt} as $\arg\min_{l'}\left(\sum_{l<l'} \upsilon_l\right) + \left(\sum_{l\geq l'} \gamma_l\right)$, where υ_l and γ_l are the costs of predicting and encoding the l-bit cores using the uniform and combinatorial prediction methods, respectively:

$$\upsilon_l = \sum_{w\in I(\mathbf{D})\cap\{0,1\}^l} \mathcal{K}_{\mathrm{u}}(C_{0w0}); \quad \gamma_l = \sum_{w\in I(\mathbf{D})\cap\{0,1\}^l} \mathcal{K}_{\mathrm{c}}(0w0);$$

[2]This reasoning holds provided $\mathbf{D}$ is non-repetitive. If $\mathbf{D}$ happens to be repetitive, a similar but slightly more complicated reasoning gives us the guarantee that $vw \notin I(\mathbf{D})$ when v is long enough.

(see Section III-E for the definitions of $\mathcal{K}_{\mathrm{u}}$ and $\mathcal{K}_{\mathrm{c}}$). Consequently, we propose to replace line 4 in Figure 1 by:

If $|w| < l_{\mathrm{opt}}$ **then**
  Predict and **Send** $C_{\mathtt{0}w\mathtt{0}}$ **uniformly**
Else
  Predict and **Send** $C_{\mathtt{0}w\mathtt{0}}$ **combinatorially**;

III. Proof of Universality

A. Markovian Source

The universality of CSE is proved for an order-k Markovian source $\mathbf{X}$. Note that the proof does not require the implementation of CSE to depend on k or on its existence.

We denote by X_i, for $i \geq 0$, the ith random variable of $\mathbf{X}$ and the subsequence of the source random variables from X_i to X_j by X_i^j. We define the *source* probability distribution $p^{(i)}$ on the strings of length i as

$$p^{(i)}(w) = \Pr(X_0^{i-1} = w), \quad \text{for } w \in \{\mathtt{0},\mathtt{1}\}^i.$$

Since $\mathbf{X}$ is an order-k Markovian source, $p^{(k+1)}$ completely characterizes $\mathbf{X}$. Indeed, for $i < k+1$, $p^{(i)}$ can be recovered using the consistency rule:

$$p^{(i)}(w) = p^{(i+1)}(w\mathtt{0}) + p^{(i+1)}(w\mathtt{1});$$

and, for $i > k+1$, $p^{(i)}$ can be recovered thanks to the finiteness of the order of $\mathbf{X}$:

$$p^{(i)}(awb) = p^{(i-1)}(aw) * p^{(i-1)}(wb) / p^{(i-2)}(w).$$

Since $\mathbf{X}$ is an order-k Markovian source, its entropy can be expressed in various forms, in particular as its kth-order entropy:

$$\begin{aligned} H(\mathbf{X}) &= \lim_{n\to\infty} H(X_1^n)/n \\ &= H(X_{k+1} \mid X_1 \dots X_k) \\ &= -\sum_{wa\in\{\mathtt{0},\mathtt{1}\}^{k+1}} p^{(k+1)}(wa) \lg \frac{p^{(k+1)}(wa)}{p^{(k)}(w)}. \end{aligned}$$

B. Empirical Probability Distributions

Given $\mathbf{D}$, we define the *empirical* probability distributions $\tilde{p}^{(i)}_{\mathbf{D}}$ on the strings of length i as

$$\tilde{p}^{(i)}_{\mathbf{D}}(w) = C_{\mathbf{D},\,w} \,/\, N, \quad \text{for } w \in \{\mathtt{0},\mathtt{1}\}^i.$$

Note that we add $\mathbf{D}$ as a subscript to C_w to indicate that $C_{\mathbf{D},\,w}$ is obtained from $\mathbf{D}$. This is because, below, we need to obtain numbers of occurrences from strings other than $\mathbf{D}$. The $\tilde{p}^{(i)}_{\mathbf{D}}$ probability distributions obey the consistency rule.

Based on the empirical probability distributions, we define the *empirical kth-order entropy* of $\mathbf{D}$ as

$$\tilde{H}(\mathbf{D}) = -\sum_{wa\in\{\mathtt{0},\mathtt{1}\}^{k+1}} \tilde{p}^{(k+1)}_{\mathbf{D}}(wa) \lg \frac{\tilde{p}^{(k+1)}_{\mathbf{D}}(wa)}{\tilde{p}^{(k)}_{\mathbf{D}}(w)}.$$

C. Empirical Probability Distributions of Random Strings

Both the empirical probability distributions and the empirical kth-order entropy have been presented for $\mathbf{D}$, which is a specific N-bit string. However, we can also use them on an N-bit random string X_0^{N-1} and get the empirical probability distributions $\tilde{p}^{(i)}_{X_0^{N-1}}$ and the empirical kth-order entropy $\tilde{H}(X_0^{N-1})$. Note that these probability distributions and this entropy, respectively, are random variables. In other words, the outcomes of random variable $\tilde{H}(X_0^{N-1})$ are specific entropies and each specific entropy has some probability of occurrence.

Random probability distribution $\tilde{p}^{(k+1)}_{X_0^{N-1}}$, by its very nature, cannot be equal to the source probability distribution $p^{(k+1)}$, since the latter is a specific probability distribution while the former is a random variable whose outcomes are specific probability distributions. Unfortunately, $E\left[\tilde{p}^{(k+1)}_{X_0^{N-1}}\right]$, even if it is a specific probability distribution, is not necessarily equal to $p^{(k+1)}$ either. One of the reasons is that the empirical probability distributions are based on numbers of occurrences, and these include occurrences of substrings that *wrap around* $\mathbf{D}$. Consequently, these substrings are not generated by consecutive random variables of $\mathbf{X}$; e.g. there is a substring that is generated by $X_{N-4}^{N-1}X_0^8$.

However, the empirical probability distribution random variables have the tendency to become more similar to the source probability distribution when we let N grow. Let us be more precise. We are especially interested in the probability distributions of substrings of $k+1$ bits. So we have

$$\lim_{N\to\infty} \tilde{p}^{(k+1)}_{X_0^{N-1}} = p^{(k+1)} \quad \text{almost surely.}$$

We omit the proof. As a consequence, and because entropy is a continuous function, we also have

$$\lim_{N\to\infty} \tilde{H}(X_0^{N-1}) = H(\mathbf{X}) \quad \text{almost surely.}$$

D. Typical Set of Strings

The convergence of the empirical entropy of the random strings toward the entropy of the source allows us to define a typical set. Let us define $A_\delta^{(N)}$, the set of typical binary strings of length N:

$$A_\delta^{(N)} = \left\{ \mathbf{D} \in \{\mathtt{0},\mathtt{1}\}^N \;\middle|\; \left| \tilde{H}(\mathbf{D}) - H(\mathbf{X}) \right| < \delta \right\}.$$

$A_\delta^{(N)}$ is such that $\Pr\left(X_0^{N-1} \in A_\delta^{(N)}\right) > 1-\delta$, and we can choose any $\delta > 0$, provided we choose N large enough.

E. Various Cost Functions

We define a few cost functions, which we use to bound the size of the codewords for the strings compressed by CSE.

It is possible to encode a natural number n using $O(\lg n)$ bits, even if we have no *a priori* upper bound on n. For instance, we can do so using Elias gamma coding [5]. The size of the codeword for $n \in \mathbb{N}$ is $\mathcal{K}_{\mathbb{N}}(n)$.

Let $\mathcal{K}_{\mathrm{u}}(X)$ be the cost of encoding the outcome of a random variable X with n possible outcomes when assigning these a

$$
\begin{aligned}
& \mathcal{K}_{\mathrm{CSE}}(\mathbf{D}) \\
= \;& \mathcal{K}_{\mathrm{CSE}|_{l_{\mathrm{opt}}}}(\mathbf{D}) \\
\leq \;& \mathcal{K}_{\mathrm{CSE}|_{k}}(\mathbf{D}) \\
= \;& \mathcal{K}_{\mathbb{N}}(N) + \mathcal{K}_{\mathrm{u}}(C_{\mathtt{0}}) + \mathcal{K}_{\mathrm{u}}(\mathrm{rank}(\mathbf{D})) \\
& + \sum_{\substack{w\in\{\mathtt{0},\mathtt{1}\}^* \\ |w|<k}} \mathcal{K}_{\mathrm{u}}(C_{\mathtt{0}w\mathtt{0}}) + \sum_{\substack{w\in\{\mathtt{0},\mathtt{1}\}^* \\ |w|\geq k}} \mathcal{K}_{\mathrm{c}}(\mathtt{0}w\mathtt{0}) \qquad (11) \\
\leq \;& \sum_{\substack{w\in\{\mathtt{0},\mathtt{1}\}^* \\ |w|\geq k}} \mathcal{K}_{\mathrm{c}}(\mathtt{0}w\mathtt{0}) + \alpha_1 \lg N \qquad (12) \\
= \;& \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} \sum_{v\in\{\mathtt{0},\mathtt{1}\}^*} \mathcal{K}_{\mathrm{c}}(\mathtt{0}v w\mathtt{0}) + \alpha_1 \lg N \\
= \;& \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} \sum_{v\in\{\mathtt{0},\mathtt{1}\}^*} -\lg \frac{\binom{C_{\mathtt{0}vw}}{C_{\mathtt{0}vw\mathtt{0}}}\binom{C_{\mathtt{1}vw}}{C_{\mathtt{1}vw\mathtt{0}}}}{\binom{C_{vw}}{C_{vw\mathtt{0}}}} + \alpha_1 \lg N \\
= \;& \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} \lg \binom{C_w}{C_{w\mathtt{0}}} + \alpha_1 \lg N \qquad (13) \\
= \;& \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} \lg \frac{C_w!}{C_{w\mathtt{0}}!\, C_{w\mathtt{1}}!} + \alpha_1 \lg N \\
\leq \;& \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} \lg \frac{C_w^{C_w}}{C_{w\mathtt{0}}^{C_{w\mathtt{0}}}\, C_{w\mathtt{1}}^{C_{w\mathtt{1}}}} + \alpha_1 \lg N \qquad (14) \\
= \;& \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} \left[-C_{w\mathtt{0}} \lg \frac{C_{w\mathtt{0}}}{C_w} - C_{w\mathtt{1}} \lg \frac{C_{w\mathtt{1}}}{C_w} \right] + \alpha_1 \lg N
\end{aligned}
$$

Fig. 2. Upper bound for both typical and atypical cases

uniform probability distribution. Naturally, $\mathcal{K}_{\mathrm{u}}(X) \in O(\lg n)$. In the sequel, we keep n implicit since all such costs will be upper-bounded by $\lg N$.

Let $\mathcal{K}_{\mathrm{c}}(\mathtt{0}w\mathtt{0})$ be the cost of encoding the value $C_{\mathtt{0}w\mathtt{0}}$ combinatorially, knowing $C_{\mathtt{0}w}$, $C_{w\mathtt{0}}$, and $C_{w\mathtt{0}}$:

$$\mathcal{K}_{\mathrm{c}}(\mathtt{0}w\mathtt{0}) = -\lg\ p_c(C_{\mathtt{0}w\mathtt{0}} \mid C_{\mathtt{0}w},\, C_{w\mathtt{0}},\, C_{w\mathtt{1}}).$$

Note that we write $\mathcal{K}_{\mathrm{c}}(\mathtt{0}w\mathtt{0})$, not $\mathcal{K}_{\mathrm{c}}(C_{\mathtt{0}w\mathtt{0}})$, because $C_{\mathtt{0}w\mathtt{0}}$ is only a natural number (e.g., 5) which would not allow us to unambiguously identify the related strings $\mathtt{0}w$, $w\mathtt{0}$, and $\mathtt{1}w$ and their respective $C_{\mathtt{0}w}$, $C_{w\mathtt{0}}$, and $C_{\mathtt{1}w}$.

We denote by $\mathcal{K}_{\mathrm{CSE}|_n}(\mathbf{D})$ the cost of compressing $\mathbf{D}$ using CSE and by forcing the prediction to switch from uniform to combinatorial when cores are n bits long or more. Finally, we denote by $\mathcal{K}_{\mathrm{CSE}}(\mathbf{D})$ the cost $\mathcal{K}_{\mathrm{CSE}|_{l_{\mathrm{opt}}}}(\mathbf{D})$.

F. An Upper Bound on the Cost

Figure 2 presents the beginning of a derivation for an upper bound that suits both typical and atypical cases. Eq. (11) directly follows from CSE's algorithm in Figure 1, as modified in Section II-C. Eq. (12) gathers all the costs that are logarithmic. In Eq. (13), we use the telescopic product presented in Section II-B. Eq. (14) is simple to prove.

G. Cost in the Typical Case

Assuming $\mathbf{D} \in A_\delta^{(N)}$, we derive the following.

$$
\begin{aligned}
& \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} \left[-C_{w\mathtt{0}} \lg \frac{C_{w\mathtt{0}}}{C_w} - C_{w\mathtt{1}} \lg \frac{C_{w\mathtt{1}}}{C_w} \right] + \alpha_1 \lg N \\
= \;& N \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} \left[-\frac{C_{w\mathtt{0}}}{N} \lg \frac{C_{w\mathtt{0}}}{C_w} - \frac{C_{w\mathtt{1}}}{N} \lg \frac{C_{w\mathtt{1}}}{C_w} \right] + \alpha_1 \lg N \\
= \;& N\, \tilde{H}(\mathbf{D}) + \alpha_1 \lg N \\
\leq \;& N\,(H(\mathbf{X}) + \delta) + \alpha_1 \lg N
\end{aligned}
$$

H. Cost in the Atypical Case

Assuming $\mathbf{D} \notin A_\delta^{(N)}$, we derive the following.

$$
\begin{aligned}
& \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} \left[-C_{w\mathtt{0}} \lg \frac{C_{w\mathtt{0}}}{C_w} - C_{w\mathtt{1}} \lg \frac{C_{w\mathtt{1}}}{C_w} \right] + \alpha_1 \lg N \\
= \;& \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} C_w \left[-\frac{C_{w\mathtt{0}}}{C_w} \lg \frac{C_{w\mathtt{0}}}{C_w} - \frac{C_{w\mathtt{1}}}{C_w} \lg \frac{C_{w\mathtt{1}}}{C_w} \right] + \alpha_1 \lg N \\
= \;& \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} C_w\, h\left(\frac{C_{w\mathtt{0}}}{C_w}\right) + \alpha_1 \lg N \qquad (15) \\
\leq \;& \sum_{w\in\{\mathtt{0},\mathtt{1}\}^k} C_w + \alpha_1 \lg N \qquad (16) \\
= \;& N + \alpha_1 \lg N
\end{aligned}
$$

Eq. (15) uses the entropy $h(\cdot)$ of a binary random variable, which is then bounded above by 1 in Eq. (16).

I. Overall Cost of Encoding with CSE

Combining the costs in the typical and atypical cases, with their respective probabilities, we get the following average cost:

$$
\begin{aligned}
& E\left[\mathcal{K}_{\mathrm{CSE}}(X_0^{N-1})\right] \\
\leq \;& (N\,(H(\mathbf{X}) + \delta) + \alpha_1 \lg N) + \delta\,(N + \alpha_1 \lg N)
\end{aligned}
$$

By letting N grow as much as needed, we bound the cost per symbol of CSE arbitrarily close to $H(\mathbf{X})$.

$$\lim_{N\to\infty} \frac{E\left[\mathcal{K}_{\mathrm{CSE}}(X_0^{N-1})\right]}{N} = H(\mathbf{X})$$

IV. CSE's Linear Complexity

The original paper [1] mainly considers non-repetitive (aperiodic) strings. For a non-repetitive string $\mathbf{D}$, the size of the associated necklace is equal to the string length N. In other words, all the N rotations of non-repetitive $\mathbf{D}$ are different from each other. In the BWT-transformed matrix [6], in which all the rotations of $\mathbf{D}$ are lexicographically arranged as rows, the rows are all different, and every pair of adjacent rows share a prefix of length between 0 and $N-1$. If we represent such

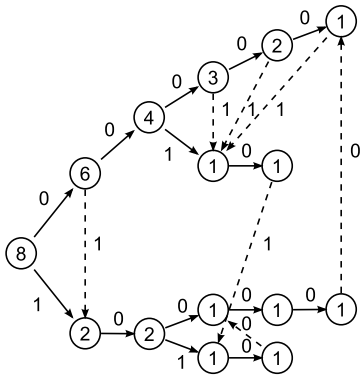

Fig. 3. The compacted substring tree (CST) for '01000001'

a prefix by w, we have both $w0$ and $w1$ in $\mathbf{D}$. This leads us to the following, which refers to Eq. (6).

Lemma 1: For any non-repetitive $\mathbf{D}$ of length N, we have $|U(\mathbf{D})| = N - 1$.

Proof: Let w_i denote the longest prefix that is shared by the ith and $(i+1)$st rows of the (sorted) BWT matrix of $\mathbf{D}$. There exist $N-1$ different w_i's since the matrix has N rows. The length of every w_i is shorter than N. This means that both w_i0 and w_i1 occur in $\mathbf{D}$, and therefore, the lemma is proved.

If we read $\mathbf{D}$ in the reverse direction and apply Lemma 1, we have the following.

Lemma 2: For any non-repetitive $\mathbf{D}$ of length N, we have $|V(\mathbf{D})| = N - 1$.

Note that the double **for** loops of the CSE algorithm transmit C_{0w0} only for $w \in I(\mathbf{D})$. Thus, from Lemmas 1 and 2, we confirm that the number of those numbers the loops emit is no more than $N-1$. The implementation of CSE uses a data structure called the *compacted substring tree* (CST) [1]. The authors of [1] conjectured that the CST for a non-repetitive $\mathbf{D}$ always has $2N-1$ nodes, and used it as a ground for linear implementation.

In the CST, the path from the root to each node represents a substring w, and the node stores the value of C_w. Since identical subtrees are linked by a *backward arc*, the CST is not strictly a tree but a graph with cycles. Figure 3 presents an example of CST, in which backward arcs are depicted with dashed lines. We here prove the conjecture [1] that the CST for a non-repetitive $\mathbf{D}$ always has $2N-1$ nodes.

We refer to normal arcs, depicted with solid lines in the figure, which go from a level-L node to a level-$(L+1)$ node, as *forward arcs*. We represent the node corresponding to a substring w by n_w. The labels only on the forward arcs of the path from the root to a node n_w constitute a substring w. When we are about to create a node n_w, if we already have a node n_x such that $C_x = C_w$ for $w = bx$ ($b \in \{0, 1\}$), then we must not build n_w and rather add a backward arc from its parent to n_x [1]. Using the way for the construction of CST, we can immediately prove the following.

Lemma 3: For a string w, if $0w$ does occur and $1w$ does not occur in $\mathbf{D}$, then a node n_{0w} does not exist in the CST.

Proof: Since $0w \in \mathbf{D}$ and $1w \notin \mathbf{D}$, we have $C_{0w} = C_w$ from Eq. (2). In this case, n_{0w} is not built and, instead, a backward arc is added to n_w.

Lemma 4: For a string w, if both $0w$ and $1w$ occur in $\mathbf{D}$, then a node n_{0w} exists in the CST for $\mathbf{D}$.

Proof: Since $C_{0w} > 0$ and $C_{1w} > 0$, we have $C_w = C_{0w} + C_{1w} > C_{0w}$. In this case, there exists a node n_{0w} in the CST.

We combine the above two lemmas into the following theorem for further reference.

Theorem 1: In the CST corresponding to $\mathbf{D}$, there exists a node n_{0w} if and only if both $0w$ and $1w$ occur in $\mathbf{D}$. Similarly, there exists a node n_{1w} if and only if $0w \in \mathbf{D}$ and $1w \in \mathbf{D}$.

Now, let us prune backward arcs from a CST to obtain a *real tree*. We still use the same term "CST" to refer to a CST without backward arcs. Let t_0 and t_1 be subtrees rooted at n_0 and n_1, respectively, in a CST without backward arcs. Note that only backward arcs are deleted from CST and no change has been made on the nodes of the CST. Thus, we now establish the following theorem.

Theorem 2: In the CST without backward arcs corresponding to a non-repetitive $\mathbf{D}$ of length N, the subtrees t_0 and t_1 are isomorphic, both of which have $N-1$ nodes.

Proof: It follows directly from Theorem 1 that the subtrees t_0 and t_1 are isomorphic. We use again the same theorem with Lemma 2 to conclude that each of t_0 and t_1 have $N-1$ nodes.

Corollary 1: The CST corresponding to a non-repetitive $\mathbf{D}$ of length N has $2N-1$ nodes.

Corollary 2: The numbers of nodes on the same level of subtrees t_0 and t_1 of the CST corresponding to a non-repetitive $\mathbf{D}$ are equal to one another.

V. Related Work

In this paper, we showed universality of CSE when the source is Markovian. A stronger result, which establishes the universality of CSE for stationary and ergodic sources is to appear about at the same time as ISIT 2011 [7].

Acknowledgments

We wish to thank the anonymous reviewers whose comments contributed to improve this paper.

References

[1] D. Dubé and V. Beaudoin, "Lossless data compression via substring enumeration," in *Proceedings of the Data Compression Conference*, Snowbird, Utah, USA, March 2010, pp. 229–238.

[2] D. Dubé, "Using synchronization bits to boost compression by substring enumeration," in *Proceedings of the International Symposium on Information Theory and its Applications*, Taichung, Taiwan, October 2010.

[3] ——, "On the use of stronger synchronization to boost compression by substring enumeration," in *Proceedings of the Data Compression Conference*, Snowbird, Utah, USA, March 2011.

[4] F. Ruskey and J. Sawada, "An efficient algorithm for generating necklaces with fixed density," *SIAM Journal of Computation*, vol. 29, no. 2, pp. 671–684, 1999.

[5] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 194–203, March 1975.

[6] M. Burrows and D. Wheeler, "A block sorting lossless data compression algorithm," Digital Equipment Corporation, Tech. Rep. 124, 1994.

[7] H. Yokoo, "Asymptotic optimal lossless compression via the CSE technique," in *Proceedings of the International Conference on Data Compression, Communication and Processing*, Palinuro, Italy, June 2011.