Constant Multiplicative Sensitivity on the CDAWGs

Rikuya Hamai
Department of Information Science and Technology, Kyushu University, Fukuoka, Japan

Hiroto Fujimaru
Department of Information Science and Technology, Kyushu University, Fukuoka, Japan

Shunsuke Inenaga
Department of Informatics, Kyushu University, Fukuoka, Japan

Abstract

Compact directed acyclic word graphs (*CDAWGs*) [Blumer et al. 1987] are a fundamental data structure on strings with applications in text pattern searching, data compression, and pattern discovery. Intuitively, the CDAWG of a string T is obtained by merging isomorphic subtrees of the suffix tree [Weiner 1973] of the same string T, and thus CDAWGs are a compact indexing structure. Indeed, the CDAWG size e can be sublinear in n for some highly repetitive strings. Of its various applications, the CDAWG allows for computing pattern occurrences, maximal exact matches (MEMs), minimal absent words (MAWs), and minimal unique substrings (MUSs) in optimal time using $O(\mathsf{e})$ space. For designing space-efficient data storage, it is crucial that the underlying data structure is robust against data edits and errors. As a mathematical measure for this, the notion of *compression sensitivity* [Akagi et al. 2023] was introduced as the maximum of the size increase in the compressed data structures after edits operations. In this paper, we investigate the sensitivity of CDAWGs when a single character edit operation is performed at an arbitrary position in the input string T. We show that the size of the CDAWG after an edit operation on T is asymptotically at most 8 times larger than the original CDAWG before the edit. This $O(1)$ upper bound significantly improves on the only known upper bound $O(n/\log n)$ for the problem.

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms

Keywords and phrases string data structures, maximal repeats, data compression, compression sensitivity, CDAWGs

Digital Object Identifier 10.4230/LIPIcs.CPM.2026.8

Funding *Rikuya Hamai*: JST BOOST Grant Number JPMJBS2406.
Hiroto Fujimaru: JST BOOST Grant Number JPMJBS2406.
Shunsuke Inenaga: JSPS KAKENHI Grant Numbers JP23K24808 and JP23K18466.

1 Introduction

Compact directed acyclic word graphs (*CDAWGs*) [4] are a fundamental data structure on strings with applications in text pattern searching, data compression, and pattern discovery. Intuitively, the CDAWG of a string T (denoted $\mathsf{CDAWG}(T)$) is a minimal partial DFA that is obtained by merging isomorphic subtrees of the suffix tree [30] of the same string T. Let e denote the size (i.e. the number of edges) of the CDAWG for the input string. It is known that $\mathsf{e} \leq 2n-2$ holds [4] for any strings of length n, and e can be sublinear in n for some highly repetitive strings [27, 2, 25]. CDAWGs can thus be regarded as a compressed text indexing structure, which can be stored in $O(\mathsf{e})$ space without explicitly storing the string [3, 14]. A grammar-based string compression of size $O(\mathsf{e})$ is also known [3, 7].

Given a pattern P of length m, the CDAWG of text T permits exact pattern matching in optimal $O(m + occ)$ time with $O(\mathsf{e})$ space, where occ is the number of occurrences of P in T. By adapting the suffix-tree based approach (Algorithm 1 in [23]), inexact matches such as *exact maximal matches* (*MEMs*) and generalized MEMs (k-MEMs and k-rare MEMs) can also be computed in optimal $O(m)$ time and with $O(\mathsf{e})$ space by the CDAWG. Finding (generalized) MEMs has important applications in bioinformatics. CDAWGs are also used as a space-efficient data structure allowing for optimal-time detection of “unusual words” (such as minimal absent words (MAWs) [8], minimal unique substrings (MUSs) [13]), and string net occurrences [12] from the input string with $O(\mathsf{e})$ space each [3, 16, 20]. On a more practical side, CDAWGs enjoy applications in natural language processing [29] and in analysis of text generated by language models (LMs) [19].

In this paper, we consider the *multiplicative sensitivity* of the CDAWG size e which is defined by the worst-case value of $\mathsf{e}(T')/\mathsf{e}(T)$, where T is the input string and T' is a string obtained by performing a single-character edit operation on T. In case where the edit operation to the string T is performed at either end of T, then the multiplicative sensitivity of CDAWGs is known to be asymptotically at most 2, and it is tight [15, 9]. However, the general case with an arbitrary single-character-wise edit on T was not well understood for CDAWGs. In this paper, we prove that any edit operation at an arbitrary position on the string can increase the size of the CDAWG asymptotically *at most 8 times larger* than the original. We emphasize that the only known upper bound for the sensitivity of CDAWGs is $O(n/\log n)$, which trivially follows since $\mathsf{e} \in O(n)$ and $\mathsf{e} \in \Omega(\log n)$ for any string of length n [4, 3]. Our technique for proving the constant sensitivity of CDAWGs is purely combinatorial, which involves new and original ideas that were not present in the special case of left/right-end edits [15, 9].

Compression sensitivity. Following the earlier work of Lagarde and Perifel [18] and Akagi et al. [1], string compressors and repetitiveness measures can be categorized into three classes:

- **(A) Stable:** Those whose multiplicative sensitivity is $O(1)$;
- **(B) Changeable:** Those whose multiplicative sensitivity is $\mathrm{polylog}(n)$;
- **(C) Catastrophic:** Those whose multiplicative sensitivity is $O(n^c)$ with some constant $0 < c \leq 1$.

To mention a few of them, it is shown in [1] that Class (A) includes the substring complexity [17], the smallest macro scheme [28], the Lempel-Ziv 77 families [31, 28], and the smallest grammar [26, 6]. Our new result implies that the CDAWGs belong to Class (A). Class (B) includes run-length Burrows-Wheeler transform (RLBWT) as shown in [1, 10, 11], and lex-parse [24] as shown by Nakashima et al. [22]. It is shown by Lagarde and Perifel [18] that the Lempel-Ziv 78 [32] belongs to Class (C).

2 Preliminaries

Strings. Let Σ be an *alphabet* of size σ. An element of Σ^* is called a *string*. For a string $T \in \Sigma^*$, the length of T is denoted by $|T|$. The *empty string*, denoted by ε, is the string of length 0. For any non-negative integer $n \geq 0$, let Σ^n denote the set of strings of length n. For any two strings S and T, let $\mathsf{ed}(S,T)$ denote the edit distance between S and T. For any string T and a non-negative integer $\ell \geq 0$, let $\mathcal{K}(T,\ell) = \{S \mid \mathsf{ed}(S,T) = \ell\}$.

For string $T = uvw$, u, v, and w are called a *prefix*, *substring*, and *suffix* of T, respectively. The sets of prefixes, substrings, and suffixes of string T are denoted by $\mathsf{Prefix}(T)$, $\mathsf{Substr}(T)$, and $\mathsf{Suffix}(T)$, respectively. For a string T of length n, $T[i]$ denotes the ith character of T for $1 \leq i \leq n$, and $T[i..j] = T[i] \cdots T[j]$ denotes the substring of T that begins at position i and ends at position j on T for $1 \leq i \leq j \leq n$.

Maximal substrings and maximal repeats. A substring $w \in \mathsf{Substr}(T)$ of T is *left-maximal* if (1) $w \in \mathsf{Prefix}(T)$ or (2) there exist two distinct characters $a, b \in \Sigma$ such that $aw, bw \in \mathsf{Substr}(T)$, and it is *right-maximal* if (1) $w \in \mathsf{Suffix}(T)$ or (2) there exist two distinct characters $a, b \in \Sigma$ such that $wa, wb \in \mathsf{Substr}(T)$. These substrings w that occur at least twice in T are also called *left-maximal repeats* and *right-maximal repeats* in T, respectively.

Let $\mathsf{LeftM}(T)$ and $\mathsf{RightM}(T)$ denote the sets of left-maximal and right-maximal substrings in T. Let $\mathsf{M}(T) = \mathsf{LeftM}(T) \cap \mathsf{RightM}(T)$. The elements in $\mathsf{M}(T)$ are called *maximal substrings* in T, and the elements in $\mathsf{M}(T) \setminus \{T\}$ are called *maximal repeats* in T. A character $a \in \Sigma$ is said to be a *right-extension* of a maximal repeat w of T if $wa \in \mathsf{Substr}(T)$.

For any substring w of a string T, we define its *left-representation* and *right-representation* by $\mathsf{lexp}_T(w) = \alpha w$ and $\mathsf{rexp}_T(w) = w\beta$, where $\alpha, \beta \in \Sigma^*$ are the shortest strings such that αw is left-maximal in T and $w\beta$ is right-maximal in T, respectively.

CDAWG. The *compact directed acyclic word graph* (*CDAWG*) of a string T, denoted $\mathsf{CDAWG}(T)$, is the minimal DFA that recognizes all substrings of T, in which each transition (edge) is labeled by a non-empty substring of T. $\mathsf{CDAWG}(T)$ has a unique source that represents the empty string ε and a unique sink that represents T. All the other internal nodes represent the maximal repeats in T, namely, the set of the longest strings represented by the nodes of $\mathsf{CDAWG}(T)$ are equal to $\mathsf{M}(T)$. See Figure 1 for a concrete example of CDAWGs. The *size* of $\mathsf{CDAWG}(T)$ for a string T of length n is the number $\mathsf{e}(T)$ of edges in $\mathsf{CDAWG}(T)$, which is equal to the number of right-extensions of maximal repeats in T. For each $x \in \mathsf{M}(T)$, let $\mathsf{d}_T(x)$ denote the number of out-edges of node x in $\mathsf{CDAWG}(T)$. It is clear that $\mathsf{e}(T) = \sum_{x \in \mathsf{M}(T)} \mathsf{d}_T(x)$. In what follows, we will identify maximal substrings with CDAWG nodes, and right-extensions of maximal repeats with CDAWG edges, respectively.

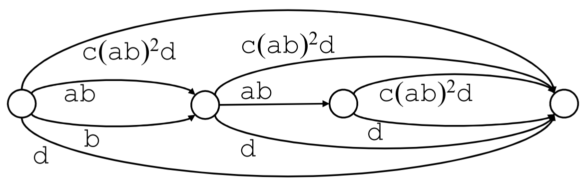

Figure 1 Illustration for $\mathsf{CDAWG}(T)$ of string $T = (\mathtt{ab})^2\mathtt{c}(\mathtt{ab})^2\mathtt{d}$. The longest strings represented by the nodes of $\mathsf{CDAWG}(T)$ are the maximal substrings in $\mathsf{M}(T) = \{\varepsilon, \mathtt{ab}, (\mathtt{ab})^2, (\mathtt{ab})^2\mathtt{c}(\mathtt{ab})^2\mathtt{d}\}$.

Sensitivity of CDAWG size and our results. Using the measure e, we define the worst-case multiplicative *sensitivity* of the CDAWG with an edit operation (insertion, deletion, or substitution) by $\mathsf{MS}(\mathsf{e}, n) = \max_{T \in \Sigma^n, T' \in \mathcal{K}(T,1)}\{\mathsf{e}(T')/\mathsf{e}(T)\}$.

In this paper, we prove the following:

▶ **Theorem 1.** *For any string T of length n, $\mathsf{MS}(\mathsf{e}, n) \le (8\mathsf{e} + 4)/\mathsf{e}$ holds. For any string T of length n ending with a unique character \$, $\mathsf{MS}(\mathsf{e}, n) \le (5\mathsf{e} + 4)/\mathsf{e}$ holds.*

3 Occurrences of maximal repeats crossing the edited position

To present an upper bound for the sensitivity of CDAWG size, it is essential to consider new occurrences of maximal repeats that contain or touch the edited position i. In this section, we introduce several new definitions regarding those new occurrences of maximal repeats.

▶ **Definition 2** (Crossing occurrences). *Let $x = T'[j..k]$ be a non-empty substring of T' that* touches or contains *the edited position i. That is, if the edit operation is insertion and substitution, (1) $k = i-1$ (touching i from left), (2) $j \le i \le k$ (containing i), or (3) $j = i+1$ (touching i from right). If the edit operation is deletion, (1) $k = i-1$ (touching i from left), (2) $j \le i-1 \wedge i \le k$ (containing i), or (3) $j = i$ (touching i from right). These occurrences of a substring x in T' are said to be* crossing occurrences *for the edited position i. We will call these occurrences simply as crossing occurrences of x.*

We denote the left most crossing occurrence $T'[j'..k']$ of x as x_L. For x_L, we consider the following substrings P_{x_L} and S_{x_L} of T' (see Figure 2 for illustration):

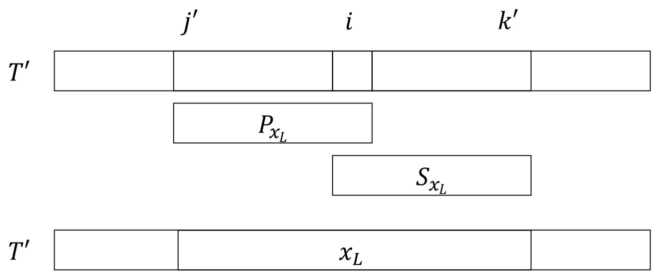

Figure 2 Illustration of x_L in T' for the case where x_L contains i, with insertion and substitution.

In the case that the edit operation is insertion or substitution, let

$$P_{x_L} = \begin{cases} T'[j'..i] & \text{if } x_L \text{ touches } i \text{ from left or contains } i, \\ \varepsilon & \text{if } x_L \text{ touches } i \text{ from right,} \end{cases}$$

$$S_{x_L} = \begin{cases} \varepsilon & \text{if } x_L \text{ touches } i \text{ from left,} \\ T'[i..k'] & \text{if } x_L \text{ contains } i \text{ or touches } i \text{ from right.} \end{cases}$$

In the case that the edit operation is deletion, let

$$P_{x_L} = \begin{cases} T'[j'..i-1] & \text{if } x_L \text{ touches } i \text{ from left or contains } i, \\ \varepsilon & \text{if } x_L \text{ touches } i \text{ from right,} \end{cases}$$

$$S_{x_L} = \begin{cases} \varepsilon & \text{if } x_L \text{ touches } i \text{ from left,} \\ T'[i..k'] & \text{if } x_L \text{ contains } i \text{ or touches } i \text{ from right.} \end{cases}$$

We define the rightmost crossing occurrence x_R, together with P_{x_R} and S_{x_R}, analogously.

▶ **Definition 3.** *We categorize strings x that have crossing occurrence(s) in the edited string in T' into the five following types, depending on the properties of x:*

Type (i): *x has only one crossing occurrence of x in T'.*

Type (ii):

1. *x has two or more crossing occurrences of x in T'.*
2. *If all occurrences of x in T' are crossing occurrences of x in T', then $x \notin \mathsf{LeftM}(T')$ and $x \notin \mathsf{RightM}(T')$.*

Type (iii):

1. *x has two or more crossing occurrences of x in T'.*
2. *If all occurrences of x in T' are crossing occurrences of x in T', then $x \notin \mathsf{LeftM}(T')$ and $x \in \mathsf{RightM}(T')$.*

Type (iv):
1. *x has two or more crossing occurrences of x in T'.*
2. *If all occurrences of x in T' are crossing occurrences of x in T', then $x \in \mathsf{LeftM}(T')$ and $x \notin \mathsf{RightM}(T')$.*

Type (v):
1. *x has two or more crossing occurrences of x in T'.*
2. *If all occurrences of x in T' are crossing occurrences of x in T', then $x \in \mathsf{M}(T')$.*

Figure 3 illustrates the five types of a string x when $x_L \notin \mathsf{Prefix}(T')$ and $x_R \notin \mathsf{Suffix}(T')$.

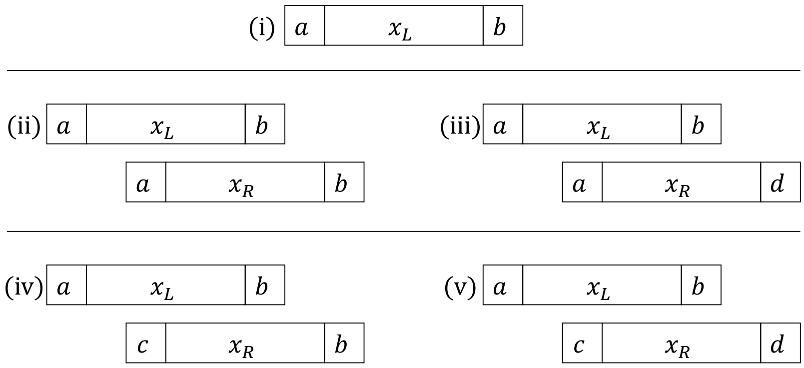

Figure 3 Illustration for the five types of a string x when $x_L \notin \mathsf{Prefix}(T')$ and $x_R \notin \mathsf{Suffix}(T')$, where $a \neq c$ and $b \neq d$ for characters $a, b, c, d \in \Sigma$.

The reason why we only consider x_L and x_R is due to the periodicity for x. We remark that when x has multiple crossing occurrences which contain or touch the edited position i in T', then the characters immediately before all crossing occurrences of x in T' except for x_L are the same and the characters immediately after all crossing occurrences of x in T' except for x_R are the same. This is because when x has three or more crossing occurrences containing i in T', then all these crossing occurrences of x are periodic (c.f. [5, 21]). Therefore, to examine whether x is maximal in $\mathsf{M}(T')$ or not, we do not need to consider all crossing occurrences of x in T' other than x_L and x_R.

In Definitions 4 and 5 below, we introduce N and Q which are respectively the sets of new maximal repeats and existing maximal repeats in T', such that $\mathsf{N} \cup \mathsf{Q} = \mathsf{M}(T') \setminus \{T'\}$. We then partition each of them into smaller subsets that are suitable for our needs.

▶ **Definition 4** (Partition of new maximal repeats). *Let $\mathsf{N} = (\mathsf{M}(T') \setminus \mathsf{M}(T)) \setminus \{T'\}$ denote the set of new maximal repeats in T'. We divide N into the three following disjoint subsets $\mathsf{N}_1 = \mathsf{N} \cap \mathsf{RightM}(T)$, $\mathsf{N}_2 = \mathsf{N} \cap \mathsf{LeftM}(T)$, and $\mathsf{N}_3 = \mathsf{N} \setminus (\mathsf{N}_1 \cup \mathsf{N}_2)$. Further, we divide N_3 into two disjoint subsets $\mathsf{N}_{3\mathrm{B}}$ and $\mathsf{N}_{3\mathrm{A}}$ as follows: $\mathsf{N}_{3\mathrm{B}}$ is the set of strings $x \in \mathsf{N}_3$ such that (1) x is of Type* (v)*, and (2) there is no other right-extension of x in T' than the right-extension(s) of the crossing occurrence(s) of x, and $\mathsf{N}_{3\mathrm{A}} = \mathsf{N}_3 \setminus \mathsf{N}_{3\mathrm{B}}$.*

We note that the following properties hold for $\mathsf{N}_1, \mathsf{N}_2$ and N_3 by definition:
- $x \in \mathsf{N}_1 \Rightarrow x \notin \mathsf{LeftM}(T)$ because $x \notin \mathsf{M}(T)$.
- $x \in \mathsf{N}_2 \Rightarrow x \notin \mathsf{RightM}(T)$ because $x \notin \mathsf{M}(T)$.
- $x \in \mathsf{N}_3 \Rightarrow x \notin \mathsf{RightM}(T) \wedge x \notin \mathsf{LeftM}(T)$.

▶ **Definition 5.** *Let $\mathsf{Q} = (\mathsf{M}(T') \cap \mathsf{M}(T)) \setminus \{T'\}$ denote the set of existing maximal repeats in T'. We divide Q into the two following subsets $\mathsf{Q}_1 = \{x \in \mathsf{Q} \mid \mathsf{d}_{T'}(x) > \mathsf{d}_T(x)\}$ and $\mathsf{Q}_2 = \{x \in \mathsf{Q} \mid \mathsf{d}_{T'}(x) \leq \mathsf{d}_T(x)\}$.*

Namely, Q_1 (resp. Q_2) is the set of existing nodes of the CDAWG for which the number of out-edges increases (resp. does not increase).

▶ **Example 6.** Consider the string $T = \mathrm{cabcabcdabca}\mathbf{d}\mathrm{bcabcdabcabdcabcabcabdabcab}$ and the string $T' = \mathrm{cabcabcdabca|bcabcdabcabdcabcabcabdabcab}$ obtained by deleting the character $T[13] = \mathbf{d}$ from T highlighted in bold. The edited position in T' is designated by a $|$. For instance, $\mathrm{dabcab} \in \mathsf{Q}_1, \mathrm{bcabc} \in \mathsf{Q}_2, \mathrm{abcabc} \in \mathsf{N}_1, \mathrm{abcabcab} \in \mathsf{N}_{3\mathrm{A}}, \mathrm{cabcabcdabcab} \in \mathsf{N}_{3\mathrm{B}}$ (also see Figure 4).

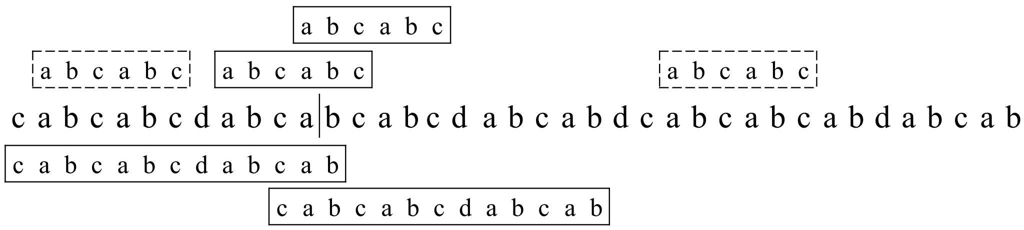

Figure 4 Illustration for $T' = \mathrm{cabcabcdabca|bcabcdabcabdcabcabcabdabcab}$ in Example 6 and the occurrences of $\mathrm{abcabc} \in \mathsf{N}_1$ and $\mathrm{cabcabcdabcab} \in \mathsf{N}_{3\mathrm{B}}$ in T'. The $|$ symbol in T' exhibits the edit position. The solid line boxes exhibit the crossing occurrences of abcabc and cabcabcdabcab in T', and the dashed line boxes exhibit the non-crossing occurrences of them in T'.

Recall that $\mathsf{e} = \sum_{x \in \mathsf{M}(T)} \mathsf{d}_T(x)$ denotes the number of edges in $\mathsf{CDAWG}(T)$ before the edit. In the subsequent sections, we work on the three disjoint subsets $\mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$, $\mathsf{N}_2 \cup \mathsf{Q}$, and $\mathsf{N}_{3\mathrm{B}}$ of $\mathsf{M}(T')$, and show that $\sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}} \mathsf{d}_{T'}(x) \leq 3\mathsf{e} + 2$ (Section 4), $\sum_{x \in \mathsf{N}_2 \cup \mathsf{Q}} \mathsf{d}_{T'}(x) \leq 3\mathsf{e} + 2$ (Section 5), and $\sum_{x \in \mathsf{N}_{3\mathrm{B}}} \mathsf{d}_{T'}(x) \leq 2\mathsf{e}$ (Section 6). All these immediately lead to Theorem 1 that upper bounds the number of edges in $\mathsf{CDAWG}(T')$ after the edit to $8\mathsf{e} + 4$.

The tighter upper bound $5\mathsf{e} + 4$ for strings ending with \$ is shown in Section 7.

4 Upper bound for total out-degrees of nodes w.r.t. $\mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$

In this section, we show an upper bound for the total out-degrees of the nodes corresponding to strings in $\mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}} \subseteq \mathsf{M}(T')$. Recall that $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ implies $x \notin \mathsf{LeftM}(T)$.

We first describe useful properties of strings $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$.

▶ **Lemma 7.** *Any $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ occurs in T.*

Proof. In the case $x \in \mathsf{N}_1$, since $x \in \mathsf{RightM}(T)$, x occurs in T.

Let us consider the case $x \in \mathsf{N}_{3\mathrm{A}}$ that is of Type (i), (ii), (iii) or (iv). Since x is not of Type (v), if all occurrences of x in T' are crossing occurrences of x in T', then $x \notin \mathsf{M}(T')$. Therefore, x occurs in T.

Let us consider the case $x \in \mathsf{N}_{3\mathrm{A}}$ that is of Type (v). Due to the definition of $\mathsf{N}_{3\mathrm{A}}$, there exists a distinct right-extension of x in T' other than the right-extension(s) of the crossing occurrence(s) of x. Therefore, there is a non-crossing occurrence of x in T', which implies that x occurs in T (see also Figure 5). ◀

▶ **Lemma 8.** *For any $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ that is of Type* (i), (ii) *or* (iii)*, there does not exist $y \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ such that $|y| > |x|$ and $S_{x_L} = S_{y_G}$, where $G \in \{L, R\}$.*

Proof. If x_L is a prefix of T', then clearly there is no y satisfying $|y| > |x|$ and $S_{x_L} = S_{y_G}$. In what follows, we consider the case that x_L is not a prefix of T'.

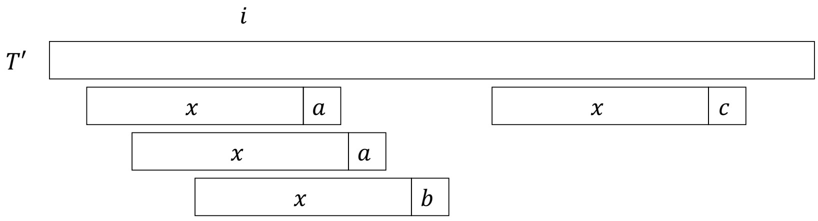

Figure 5 Illustration for Lemma 7 where i is the edited position and a, b, c differ from each other.

For a contrary, suppose that for $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ that is of Type (i), (ii) or (iii), there exists $y \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ such that $|y| > |x|$ and $S_{x_L} = S_{y_G}$, where $G \in \{L, R\}$. See also Figure 6. Let a be the character immediately before x_L. Since x is of Type (i), (ii) or (iii), every crossing occurrence of x in T' is immediately preceded by a. Because $x \in \mathsf{M}(T')$, it holds that $x \in \mathsf{Prefix}(T')$, or there is a distinct character $b \in \Sigma \setminus \{a\}$ such that bx occurs in T'. This implies that there is a non-crossing occurrence of x in T', which is as a prefix of T or is immediately preceded by b in T. By Lemma 7, y occurs in T, and thus ax that is a suffix of y also occurs in T. Hence $x \in \mathsf{LeftM}(T)$, however, this contradicts that $x \notin \mathsf{LeftM}(T)$. ◀

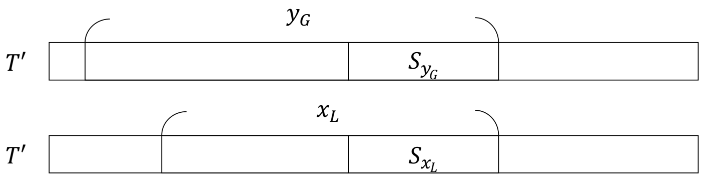

Figure 6 Illustration for Lemma 8: impossible occurrences of x and y with $S_{x_L} = S_{y_G}$.

▶ **Lemma 9.** *For any $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ that is of Type* (iv) *or* (v)*, there do not exist $y, z \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ with $|y| > |x|$ and $|z| > |x|$ satisfying $S_{x_L} = S_{y_G}$ and $S_{x_R} = S_{z_F}$ simultaneously, where $G, F \in \{L, R\}$.*

Proof. If x_L is a prefix of T', then clearly there is no y satisfying $|y| > |x|$ and $S_{x_L} = S_{y_G}$. In what follows, we consider the case that x_L is not a prefix of T'.

For a contrary, suppose that for $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ that is of Type (iv) or (v), there exist $y, z \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ with $|y| > |x|$ and $|z| > |x|$ such that $S_{x_L} = S_{y_G}$ and $S_{x_R} = S_{z_F}$ at the same time, where $G, F \in \{L, R\}$. Let the character immediately before x_L and the character immediately before x_R be a and c ($a \neq c$), respectively. By Lemma 7, y and z occur in T, and thus ax that is a suffix of y and cx that is a suffix of z both occur in T. Therefore, $x \in \mathsf{LeftM}(T)$, however, this contradicts that $x \notin \mathsf{LeftM}(T)$. ◀

4.1 Correspondence between $\mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ and $\mathsf{M}(T)$

For any $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ that is of Type (i), (ii) or (iii), we associate x with S_{x_L}. For any $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ that is of Type (iv) or (v), if there does not exist $y \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ such that $|y| > |x|$ and $S_{x_L} = S_{y_G}$ with $G \in \{L, R\}$, we associate x with S_{x_L}, and otherwise we associate x with S_{x_R}.

By Lemma 8 and Lemma 9, each $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ can be associated to a distinct string S_{x_G} with $G \in \{L, R\}$. Note however that S_{x_G} may not be maximal in T. Thus we introduce a function U that bridges each $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ to a distinct maximal substring in T.

▶ **Definition 10.** *For any* $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$*, let* $U(x) = \mathsf{lexp}_T(S_{x_G})$ *(see Figure 7).*

By Lemma 7, x occurs in T and thus its suffix S_{x_G} also occurs in T. Hence $U(x) = \mathsf{lexp}_T(S_{x_G})$ is well defined.

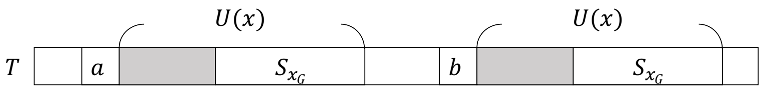

Figure 7 Illustration for $U(x)$ $(a \neq b)$.

▶ **Lemma 11.** *For any* $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$*,* $U(x) \in \mathsf{M}(T)$*.*

Proof. By Definition 10, $U(x) = \mathsf{lexp}_T(S_{x_G}) \in \mathsf{LeftM}(T)$. Therefore, it suffices for us to prove $U(x) \in \mathsf{RightM}(T)$. From now on, we consider the four following cases:

Case (a) $x \in \mathsf{N}_1$**.** In this case, $x \in \mathsf{RightM}(T)$, therefore S_{x_G} that is a suffix of x also satisfies $S_{x_G} \in \mathsf{RightM}(T)$. Hence, $U(x) = \mathsf{lexp}_T(S_{x_G}) \in \mathsf{RightM}(T)$.

Case (b) $x \in \mathsf{N}_{3\mathrm{A}}$ **and** x **is of Type (i), (ii) or (iv).** Let the character immediately after all crossing occurrences of x in T' be a. There exists xb $(b \neq a)$ in T' or $x \in \mathsf{Suffix}(T')$ because $x \in \mathsf{M}(T')$. Since the character immediately after all crossing occurrences of x in T' is a, then there exists xb $(b \neq a)$ or $x \in \mathsf{Suffix}(T)$ in T. Hence, $S_{x_G} \in \mathsf{RightM}(T)$ since the character immediately after S_{x_G} is a. Thus, $U(x) = \mathsf{lexp}_T(S_{x_G}) \in \mathsf{RightM}(T)$.

Case (c) $x \in \mathsf{N}_{3\mathrm{A}}$ **and** x **is of Type (iii).** Since x is of Type (iii), we associate x with S_{x_L}. Because x is of Type (iii) and S_{x_L} is a suffix of a S_{x_R}, $S_{x_G} \in \mathsf{RightM}(T)$ holds. Thus, $U(x) = \mathsf{lexp}_T(S_{x_G}) \in \mathsf{RightM}(T)$.

Case (d) $x \in \mathsf{N}_{3\mathrm{A}}$ **and** x **is of Type (v).** Let the character immediately after S_{x_G} be a. Since there exists a distinct right-extension of x in T' other than the right-extension(s) of the crossing occurrence(s) of x, there exists xb $(b \neq a)$ in T. Therefore, $S_{x_G} \in \mathsf{RightM}(T)$. Thus, $U(x) = \mathsf{lexp}_T(S_{x_G}) \in \mathsf{RightM}(T)$.

Consequently, we have $U(x) \in \mathsf{M}(T)$. ◀

The next lemma states the uniqueness of $U(x)$.

▶ **Lemma 12.** *For any* $x, y \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ *with* $x \neq y$*,* $U(x) \neq U(y)$*.*

Proof. Suppose that there exist $x, y \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ such that $x \neq y$ and $U(x) = U(y)$. Let x and y correspond to S_{x_G} and S_{y_F}, respectively, where $G, F \in \{L, R\}$. Let $U(x) = AS_{x_G}, U(y) = BS_{y_F}$ $(A, B \in \mathsf{Substr}(T))$, and assume without loss of generality that $|S_{x_G}| < |S_{y_F}|$ due to Lemma 8 and Lemma 9. Then $|A| > |B|$ because $U(x) = U(y)$. Since $U(y) = BS_{y_F} \in \mathsf{M}(T)$ by Lemma 11, and since BS_{x_G} is a prefix of BS_{y_F} (see Figure 8), we have $BS_{x_G} \in \mathsf{LeftM}(T)$. This contradicts $\mathsf{lexp}_T(S_{x_G}) = AS_{x_G}$. ◀

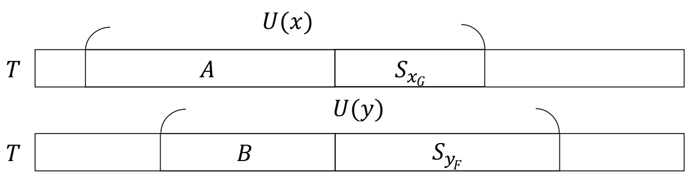

Figure 8 Illustration for the proof of Lemma 12, where $U(x) = U(y)$.

4.2 Upper bound w.r.t. $\mathsf{N}_1 \cup \mathsf{N}_{3A}$

▶ **Lemma 13.** $\sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3A}} \mathsf{d}_{T'}(x) \leq 3\mathsf{e} + 2$.

Proof. Let $U(x) = \mathsf{lexp}_T(S_{x_G})$, where $G \in \{L, R\}$. Since S_{x_G} is a suffix of x, $\mathsf{d}_T(x) \leq \mathsf{d}_T(U(x))$. Since there are at most two distinct characters immediately after the crossing occurrences of x, $\mathsf{d}_{T'}(x) \leq \mathsf{d}_T(U(x)) + 2$. For $U(x) \neq T$, we have $\mathsf{d}_T(U(x)) \geq 1$. Thus $\mathsf{d}_{T'}(x) \leq \mathsf{d}_T(U(x)) + 2 \leq 3\mathsf{d}_T(U(x))$. For $U(x) = T$, we have $\mathsf{d}_T(U(x)) = 0$. Thus $\mathsf{d}_{T'}(x) \leq 2$. By using Lemma 12 and summing up these, we get $\sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3A}} \mathsf{d}_{T'}(x) \leq \sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3A}} 3\mathsf{d}_T(U(x)) + 2 \leq 3\mathsf{e} + 2$. ◀

5 Upper bound for total out-degrees of nodes w.r.t. $\mathsf{N}_2 \cup \mathsf{Q}$

In this section, we show an upper bound for the total out-degrees of nodes corresponding to strings that are elements of $\mathsf{N}_2 \cup \mathsf{Q} \subseteq \mathsf{M}(T')$.

We first present properties of the strings in $\mathsf{N}_2 \cup \mathsf{Q}$. In particular, we focus on the strings in $\mathsf{N}_2 \cup \mathsf{Q}_1$, as the strings in Q_2 are less important and can be handled in a trivial manner.

▶ **Lemma 14.** *Any $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ occurs in T.*

Proof. Since $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$, $x \in \mathsf{LeftM}(T)$. Thus $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ occurs in T. ◀

▶ **Lemma 15.** *For any $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ that is of Type* (i), (ii) *or* (iv)*, there does not exist $y \in \mathsf{N}_2 \cup \mathsf{Q}_1$ such that $|y| > |x|$ and $P_{x_G} = P_{y_F}$, where $G, F \in \{L, R\}$.*

Proof. The case that $x \in \mathsf{N}_2$ follows from a symmetrical argument to Lemma 8, in which y may belong to N_2 or Q_1. Let us consider the case that $x \in \mathsf{Q}_1$. Suppose that for $x \in \mathsf{Q}_1$ which is of Type (i), (ii) or (iv), there is $y \in \mathsf{N}_2 \cup \mathsf{Q}_1$ such that $|y| > |x|$ and $P_{x_G} = P_{y_F}$, where $G, F \in \{L, R\}$. If x_R is a suffix of T', then there is no y such that $|y| > |x|$ and $P_{x_G} = P_{y_F}$. From now on consider the case that x_R is not a suffix of T'. Let b be the character immediately after x_G. Then, since x is of Type (i), (ii) or (iv), character b immediately follows every crossing occurrence of x in T'. Note that xb is a prefix of y. Due to Lemma 14, y occurs in T, implying xb also occurs in T. Thus the number of right-extensions of x in T' is no more than the number of right-extensions of x in T. However, this contradicts $x \in \mathsf{Q}_1$. ◀

▶ **Lemma 16.** *For any $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ that is of Type* (iii) *or* (v)*, there do not exist $y, z \in \mathsf{N}_2 \cup \mathsf{Q}_1$ with $|y| > |x|$ and $|z| > |x|$ satisfying $P_{x_L} = P_{y_G}$ and $P_{x_R} = P_{z_F}$ simultaneously, where $G, F \in \{L, R\}$.*

Proof. If x_R is a suffix of T', then clearly there is no z satisfying $|z| > |x|$ and $P_{x_R} = S_{z_G}$. In what follows, we consider the case that x_R is not a suffix of T'.

Suppose that for $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ which is of Type (iii) or (v), there exist $y, z \in \mathsf{N}_2 \cup \mathsf{Q}_1$ with $|y| > |x|$, $|z| > |x|$ that satisfy $P_{x_L} = P_{y_G}$ and $P_{x_R} = P_{z_F}$ at the same time, where $G, F \in \{L, R\}$. Let b and d ($b \neq d$) be the character immediately after x_L in T' and the character immediately after x_R in T', respectively. By Lemma 14, y and z occur in T, and hence xb that is a prefix of y and xd that is a prefix of z also occur in T. Therefore, $x \in \mathsf{RightM}(T)$. However, if $x \in \mathsf{N}_2(T)$, this contradicts $x \notin \mathsf{RightM}(T)$. Also, if $x \in \mathsf{Q}_1$, the number of right-extensions of x in T' does not increase from the number of right-extensions of x in T. However, this contradicts $x \in \mathsf{Q}_1$. ◀

5.1 Correspondence between $\mathsf{N}_2 \cup \mathsf{Q}_1$ and $\mathsf{M}(T)$

For any $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ that is of Type (i), (ii) or (iv), then we associate x with both P_{x_L} and P_{x_R}. For any $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ that is of Type (iii) or (v),

- if there exists $y \in \mathsf{N}_2 \cup \mathsf{Q}_1$ with $|y| > |x|$ such that $P_{x_R} = P_{y_G}$ where $G \in \{L, R\}$, then we associate x with P_{x_L} (see Figure 9);
- if there exists $y \in \mathsf{N}_2 \cup \mathsf{Q}_1$ with $|y| > |x|$ such that $P_{x_L} = P_{y_G}$ where $G \in \{L, R\}$, then we associate x with P_{x_R} (see Figure 10);
- otherwise, we associate x with both P_{x_L} and P_{x_R}.

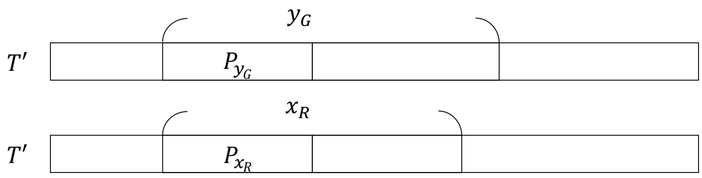

Figure 9 When there exists $y \in \mathsf{N}_2 \cup \mathsf{Q}_1$ with $|y| > |x|$ such that $P_{x_R} = P_{y_G}$, where $G \in \{L, R\}$.

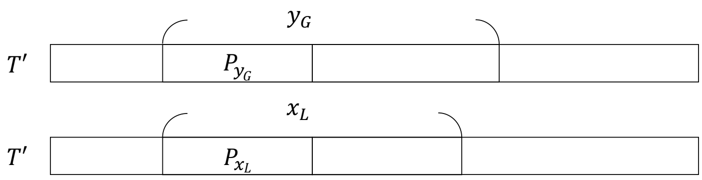

Figure 10 When there exists $y \in \mathsf{N}_2 \cup \mathsf{Q}_1$ with $|y| > |x|$ such that $P_{x_L} = P_{y_G}$, where $G \in \{L, R\}$.

By Lemmas 15 and 16, each $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ corresponds to a distinct string P_{x_G}, where $G \in \{L, R\}$. Below, for each $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$, we define $H(x)$ and $I(x)$ to which x corresponds:

▶ **Definition 17.** *For each $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ associated to P_{x_L}, let $H(x) = \mathsf{rexp}_T(P_{x_L})$. For each $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ associated to P_{x_R}, let $I(x) = \mathsf{rexp}_T(P_{x_R})$. See Figure 11. When there is only one crossing occurrence of x (i.e. $x_L = x_R$), only $H(x)$ is defined as above and $I(x)$ is undefined.*

$H(x)$ (resp. $I(x)$) is undefined for any $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ that is *not* associated to P_{x_L} (resp. P_{x_R}).

By Lemma 14 every $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$ occurs in T, and thus $H(x)$ and $I(x)$ are well defined when x is associated to P_{x_L} and P_{x_R}, respectively.

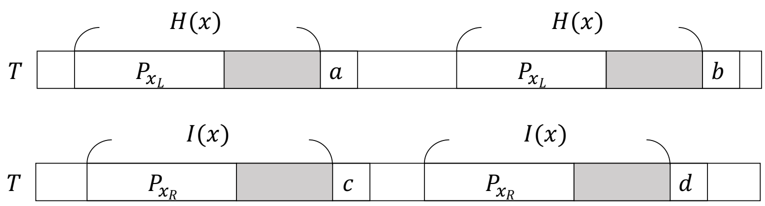

Figure 11 Illustration for $H(x)$ and $I(x)$ $(a \neq b, c \neq d)$.

▶ **Lemma 18.** *For any $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$, $H(x) \in \mathsf{M}(T)$ if $H(x)$ is defined, and $I(x) \in \mathsf{M}(T)$ if $I(x)$ is defined.*

Proof. By Definition 17, $H(x), I(x) \in \mathsf{RightM}(T)$. Therefore, it suffices for us to prove $H(x), I(x) \in \mathsf{LeftM}(T)$. For any $x \in \mathsf{N}_2 \cup \mathsf{Q}_1$, $x \in \mathsf{LeftM}(T)$. Since P_{x_G} ($G \in \{L, R\}$) is a prefix of x, we have $P_{x_G} \in \mathsf{LeftM}(T)$. Hence $H(x), I(x) \in \mathsf{LeftM}(T)$ holds. ◀

▶ **Lemma 19.** *For any $x, y \in \mathsf{N}_1 \cup \mathsf{Q}_1$ with $x \neq y$, let $\mathcal{L}$ be a list of $H(x)$, $I(x)$, $H(y)$, $I(y)$ which are defined. Then the elements in $\mathcal{L}$ differ from each other.*

Proof. By a symmetrical argument to Lemma 12. ◀

5.2 Upper bound w.r.t. $\mathsf{N}_2 \cup \mathsf{Q}$

▶ **Lemma 20.** $\sum_{x \in \mathsf{N}_2 \cup \mathsf{Q}} \mathsf{d}_{T'}(x) \leq 3\mathsf{e} + 2$.

Proof. Below, we consider all the four possible cases depending on whether $x \in \mathsf{N}_2$ or $x \in \mathsf{Q}_1$, and whether $H(x), I(x) \neq T$.

When $x \in \mathsf{N}_2$ and $H(x), I(x) \neq T$.

- First, we consider the case that x is associated with both $H(x)$ and $I(x)$. Since $x \in \mathsf{N}_2$, then $x \notin \mathsf{RightM}(T)$. Therefore, the number of characters that are immediately after x in T is at most one. Moreover, there are at most two distinct characters immediately after the crossing occurrences of x. Hence, there are at most three distinct characters immediately after x in T', namely we have

$$\mathsf{d}_{T'}(x) \leq 3. \tag{1}$$

  In addition, since $H(x), I(x) \neq T$, it holds that $\mathsf{d}_T(H(x)), \mathsf{d}_T(I(x)) \geq 1$. By Inequality 1, we get $\mathsf{d}_{T'}(x) \leq 3 \leq \mathsf{d}_T(H(x)) + \mathsf{d}_T(I(x)) + 1 \leq 2\mathsf{d}_T(H(x)) + 2\mathsf{d}_T(I(x))$.
- Second, we consider the case that x is associated with only one of $H(x)$ or $I(x)$.
  - Assume that we associate x with $H(x)$. Since $x \in \mathsf{N}_2$, then $x \notin \mathsf{RightM}(T)$. Therefore, the number of characters immediately after x in T is at most one. In this case, we do not associate x with $I(x)$, hence, x has only one crossing occurrence or there exists $y \in \mathsf{N}_2 \cup \mathsf{Q}_1$ such that $|y| > |x|$ and $P_{x_R} = P_{y_G}$ where $G \in \{L, R\}$. When x has only one crossing occurrence, there are at most one character immediately after the crossing occurrence of x. When there exists $y \in \mathsf{N}_2 \cup \mathsf{Q}_1$ such that $|y| > |x|$ and $P_{x_R} = P_{y_G}$ where $G \in \{L, R\}$, then such y occurs in T due to Lemma 14. Therefore, although there are at most two distinct characters immediately after the crossing occurrences of x, one of them is the character immediately after x in T as shown in Figure 12. Hence, we have

$$\mathsf{d}_{T'}(x) \leq 2. \tag{2}$$

    In addition, since $H(x) \neq T$, then $\mathsf{d}_T(H(x)) \geq 1$ holds. By Inequality 2, we get $\mathsf{d}_{T'}(x) \leq 2 \leq \mathsf{d}_T(H(x)) + 1 \leq 2\mathsf{d}_T(H(x))$.
  - Let us assume that we associate x with $I(x)$. In the same way as we associate x with $H(x)$, we get $\mathsf{d}_{T'}(x) \leq 2 \leq \mathsf{d}_T(I(x)) + 1 \leq 2\mathsf{d}_T(I(x))$.

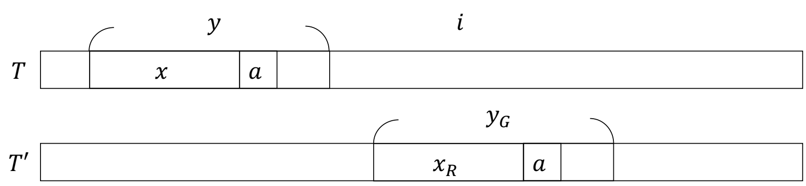

Figure 12 xa occurs in T, where a is the character immediately after the crossing occurrence x_R.

When $x \in \mathsf{N}_2$ and $H(x) = T$ or $I(x) = T$.

- Let $H(x) = T$. Now that $H(x)$ is defined, x is associated with $H(x)$.
  - First, we consider the case that we associate x with both $H(x)$ and $I(x)$. In the same way as in Inequality 1, we get $\mathsf{d}_{T'}(x) \leq 3$. In addition, since $H(x) = T$ and Lemma 19 holds, $I(x) \neq T$ and thus $\mathsf{d}_T(I(x)) \geq 1$ holds. Hence, we have $\mathsf{d}_{T'}(x) \leq 3 \leq \mathsf{d}_T(I(x)) + 2 \leq 2\mathsf{d}_T(H(x)) + 2\mathsf{d}_T(I(x)) + 2$.
  - Second, we consider the case that we only associate x with $H(x)$. Since $H(x) = T$, $\mathsf{d}_T(H(x)) = 0$ holds. In the same way as in Inequality 2, we get $\mathsf{d}_{T'}(x) \leq 2$. Thus, we have $\mathsf{d}_{T'}(x) \leq 2 \leq 2\mathsf{d}_T(H(x)) + 2$.
- Let $I(x) = T$. Now that $I(x)$ is defined, x is associated with $I(x)$. In the same way as in the case for $H(x) = T$, we get $\mathsf{d}_{T'}(x) \leq 3 \leq \mathsf{d}_T(H(x)) + 2 \leq 2\mathsf{d}_T(H(x)) + 2\mathsf{d}_T(I(x)) + 2$ in the case that we associate x with both $H(x)$ and $I(x)$, and we get $\mathsf{d}_{T'}(x) \leq 2 \leq 2\mathsf{d}_T(I(x)) + 2$ in the case that we only associate x with $I(x)$.

When $x \in \mathsf{Q}_1$ and $H(x), I(x) \neq T$. Here, we analyze $\mathsf{d}_{T'}(x) - \mathsf{d}_T(x)$ since $x \in \mathsf{Q}_1$.

- First, we consider the case that we associate x with both $H(x)$ and $I(x)$. There are at most two distinct characters immediately after the crossing occurrences of x. Hence,

  $$\mathsf{d}_{T'}(x) - \mathsf{d}_T(x) \leq 2. \tag{3}$$

  In addition, since $H(x), I(x) \neq T$, then $\mathsf{d}_T(H(x)), \mathsf{d}_T(I(x)) \geq 1$ holds. By Inequality 3, we get $\mathsf{d}_{T'}(x) - \mathsf{d}_T(x) \leq 2 \leq \mathsf{d}_T(H(x)) + \mathsf{d}_T(I(x)) \leq \mathsf{d}_T(H(x)) + \mathsf{d}_T(I(x))$.
- Second, we consider the case that we associate x with only one of $H(x)$ or $I(x)$. Here, let us assume that we associate x with $H(x)$. In this case, we do not associate x with $I(x)$, hence, x has only one crossing occurrence or there exists $y \in \mathsf{N}_2 \cup \mathsf{Q}_1$ such that $|y| > |x|$ and $P_{x_R} = P_{y_G}$ where $G \in \{L, R\}$. When x has only one crossing occurrence, there are at most one character immediately after the crossing occurrence of x. When there exists $y \in \mathsf{N}_2 \cup \mathsf{Q}_1$ such that $|y| > |x|$ and $P_{x_R} = P_{y_G}$ where $G \in \{L, R\}$, then such y occurs in T due to Lemma 14. Therefore, although there are at most two distinct characters immediately after the crossing occurrences of x, one of them is the character immediately after x in T as shown in Figure 12. Hence, we have

  $$\mathsf{d}_{T'}(x) - \mathsf{d}_T(x) \leq 1. \tag{4}$$

  In addition, since $H(x) \neq T$, then $\mathsf{d}_T(H(x)) \geq 1$ holds. By Inequality 4, we get $\mathsf{d}_{T'}(x) - \mathsf{d}_T(x) \leq 1 \leq \mathsf{d}_T(H(x))$. In the case that we associate x with $I(x)$, in the same way as we associate x with $H(x)$, we get $\mathsf{d}_{T'}(x) - \mathsf{d}_T(x) \leq 1 \leq \mathsf{d}_T(I(x))$.

When $x \in \mathsf{Q}_1$ and $H(x) = T$ or $I(x) = T$. Here, we analyze $\mathsf{d}_{T'}(x) - \mathsf{d}_T(x)$ since $x \in \mathsf{Q}_1$.

- Let $H(x) = T$. Since $H(x)$ is defined, x is associated with $H(x)$.
  - First, let us consider the case that we associate x with both $H(x)$ and $I(x)$. In the same way as in Inequality 3, we get $\mathsf{d}_{T'}(x) - \mathsf{d}_T(x) \leq 2$. In addition, since $H(x) = T$ and Lemma 19 holds, $I(x) \neq T$ and thus $\mathsf{d}_T(I(x)) \geq 1$ holds. Hence $\mathsf{d}_{T'}(x) - \mathsf{d}_T(x) \leq 2 \leq \mathsf{d}_T(I(x)) + 1 \leq \mathsf{d}_T(H(x)) + \mathsf{d}_T(I(x)) + 1$.
  - Second, let us consider the case that we only associate x with $H(x)$. Since $H(x) = T$, $\mathsf{d}_T(H(x)) = 0$ holds. In the same way as in Inequality 4, we get $\mathsf{d}_{T'}(x) - \mathsf{d}_T(x) \leq 1$. Thus, we have $\mathsf{d}_{T'}(x) - \mathsf{d}_T(x) \leq 1 \leq \mathsf{d}_T(H(x)) + 1$.

- Let $I(x) = T$. Since $I(x)$ is defined, x is associated with $I(x)$. In the same way as in the case for $H(x) = T$, we get $\mathsf{d}_{T'}(x) - \mathsf{d}_T(x) \leq 2 \leq \mathsf{d}_T(H(x)) + 1 \leq \mathsf{d}_T(H(x)) + \mathsf{d}_T(I(x)) + 1$ in the case that we associate x with both $H(x)$ and $I(x)$, and we get $\mathsf{d}_{T'}(x) - \mathsf{d}_T(x) \leq 1 \leq \mathsf{d}_T(I(x)) + 1$ in the case that we only associate x with $I(x)$.

Table 1 Upper bounds for each case of Lemma 20.

	When $H(x) \neq T \wedge I(x) \neq T$	When $H(x) = T \vee I(x) = T$
$\mathsf{d}_{T'}(x) - \mathsf{d}_T(x)$ $(x \in \mathsf{Q}_1)$	$\leq \mathsf{d}_T(H(x)) + \mathsf{d}_T(I(x))$	$\leq \mathsf{d}_T(H(x)) + \mathsf{d}_T(I(x)) + 1$
$\mathsf{d}_{T'}(x)$ $(x \in \mathsf{N}_2)$	$\leq 2(\mathsf{d}_T(H(x)) + \mathsf{d}_T(I(x)))$	$\leq 2(\mathsf{d}_T(H(x)) + \mathsf{d}_T(I(x))) + 2$

Wrapping up. Table 1 summarizes the bounds obtained above. For simplicity, let $\mathsf{d}_T(I(x)) = 0$ when $I(x)$ is undefined, and let $\mathsf{d}_T(H(x)) = 0$ when $H(x)$ is undefined. Note that this does not affect our upper bound analysis, since no maximal repeats in T' are associated to the undefined $H(x)$'s and $I(x)$'s. By Lemma 19, there is at most one string x such that $H(x) = T$ or $I(x) = T$. Thus, by using Lemma 19 and summing up the values in Table 1, we obtain $\sum_{x \in \mathsf{N}_2} \mathsf{d}_{T'}(x) + \sum_{x \in \mathsf{Q}_1} (\mathsf{d}_{T'}(x) - \mathsf{d}_T(x)) \leq 2\mathsf{e} + 2$. Also, since the number of out-edges of $x \in \mathsf{Q}_2$ does not increase, we get $\sum_{x \in \mathsf{Q}_2} \mathsf{d}_{T'}(x) + \sum_{x \in \mathsf{Q}_1} \mathsf{d}_T(x) \leq \sum_{x \in \mathsf{Q}_2} \mathsf{d}_T(x) + \sum_{x \in \mathsf{Q}_1} \mathsf{d}_T(x) \leq \sum_{x \in \mathsf{Q}} \mathsf{d}_T(x) \leq \sum_{x \in \mathsf{M}(T)} \mathsf{d}_T(x) = \mathsf{e}$. By adding $\sum_{x \in \mathsf{N}_2} \mathsf{d}_{T'}(x) + \sum_{x \in \mathsf{Q}_1} (\mathsf{d}_{T'}(x) - \mathsf{d}_T(x)) \leq 2\mathsf{e} + 2$, we get $\sum_{x \in \mathsf{N}_2 \cup \mathsf{Q}} \mathsf{d}_{T'}(x) \leq 3\mathsf{e} + 2$. ◀

6 Upper bound for total out-degrees of nodes w.r.t. $\mathsf{N}_{3\mathrm{B}}$

In this section, we show an upper bound for the total out-degrees of nodes corresponding to strings that are elements of $\mathsf{N}_{3\mathrm{B}} \subseteq \mathsf{M}(T')$. We first describe useful properties of strings $x \in \mathsf{N}_{3\mathrm{B}}$.

▶ **Definition 21.** *For any $x \in \mathsf{N}_{3\mathrm{B}}$, let J_x be the string that is obtained by removing P_{x_R} and S_{x_L} from x, namely $x = P_{x_R} J_x S_{x_L}$.*

Note that, by the definition of Type (v), each $x \in \mathsf{N}_{3\mathrm{B}}$ has two or more crossing occurrences in T'. Hence J_x always exists (possibly the empty string). See Figures 13 and 14.

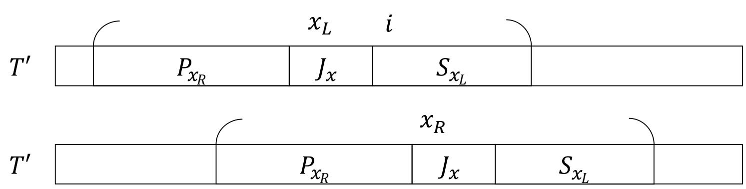

Figure 13 Illustration for J_x in case of insertions and substitutions.

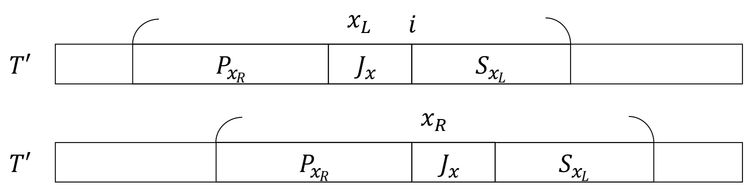

Figure 14 Illustration for J_x in case of deletions.

▶ **Lemma 22.** *For any $x, y \in \mathsf{N}_{3\mathrm{B}}$ with $x \neq y$, $J_x \neq J_y$.*

Proof. For a contrary, suppose that there exist $x, y \in \mathsf{N}_{3\mathrm{B}}$ such that $x \neq y$ and $J_x = J_y$. Since x is of Type (v), the characters immediately after x_L and x_R are different and let a, b ($a \neq b$) be these characters, respectively. If $|S_{x_L}| < |S_{y_L}|$, then both $S_{x_L}a$ and $S_{x_L}b$ must be prefixes of S_{y_L} (see Figures 15 and 16), which contradicts that $a \neq b$. The other case where $|S_{x_L}| > |S_{y_L}|$ also leads to a contradiction. Hence $|S_{x_L}| = |S_{y_L}|$, which implies $S_{x_L} = S_{y_L}$. Also, $P_{x_R} = P_{y_R}$ follows in a symmetric manner. These imply $x = y$, which is a contradiction. ◀

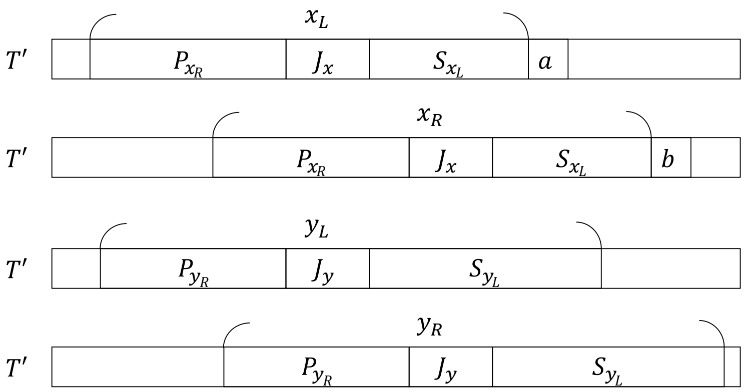

Figure 15 Illustration for Lemma 22 in case of insertions and substitutions, where $J_x = J_y$.

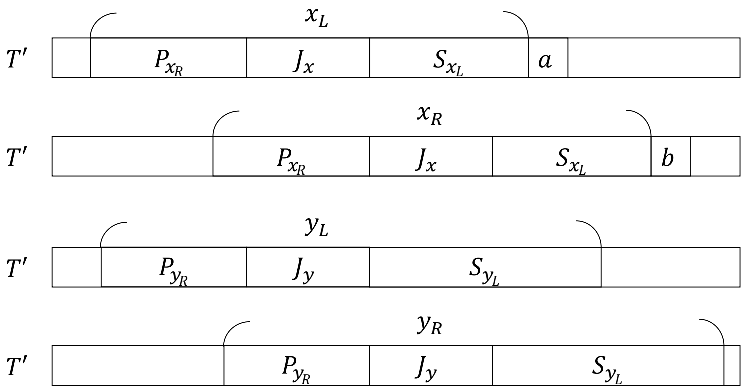

Figure 16 Illustration for Lemma 22 in case of deletions, where $J_x = J_y$.

6.1 Correspondence between $\mathsf{N}_{3\mathrm{B}}$ and $\mathsf{M}(T)$

For any $x \in \mathsf{N}_{3\mathrm{B}}$, we associate x with J_x. For any $x \in \mathsf{N}_{3\mathrm{B}}$ we define $K(x)$ to which x corresponds, by using J_x, as follows:

▶ **Definition 23.** *For any $x \in \mathsf{N}_{3\mathrm{B}}$, let $K(x) = \mathsf{lexp}_T(\mathsf{rexp}_T(J_x)) = \mathsf{rexp}_T(\mathsf{lexp}_T(J_x))$ (see also Figure 17).*

We note that $K(x)$ is well defined since J_x is a substring of T.

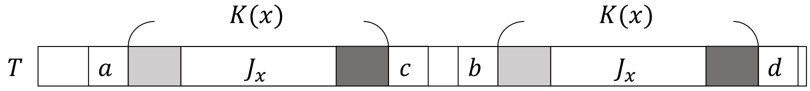

Figure 17 Illustration for Definition 23, where $a \neq b$ and $c \neq d$.

▶ **Lemma 24.** *For any $x, y \in \mathsf{N}_{3\mathrm{B}}$ with $x \neq y$, $K(x) \neq K(y)$.*

Proof. Suppose that there exist $x, y \in \mathsf{N}_{3\mathrm{B}}$ such that $x \neq y$ and $K(x) = K(y)$, and without loss of generality that $|J_x| \leq |J_y|$. Then, J_x occurs at least twice in J_y. Therefore, $K(x) \neq K(y)$, however, this is a contradiction. ◀

6.2 Upper bound w.r.t. $\mathsf{N}_{3\mathrm{B}}$

▶ **Lemma 25.** $\sum_{x \in \mathsf{N}_{3\mathrm{B}}} \mathsf{d}_{T'}(x) \leq 2\mathsf{e}$.

Proof. By the definition of $\mathsf{N}_{3\mathrm{B}}$, there is no other right-extension of x in T' than the right-extension(s) of the crossing occurrence(s) of x. Thus there are at most two distinct characters immediately after x in T'. Since $K(x)$ occurs at least twice in T, $\mathsf{d}_T(K(x)) \geq 1$. Hence, we get $\mathsf{d}_{T'}(x) \leq 2 \leq 2\mathsf{d}_T(K(x))$. Therefore, we have $\sum_{x \in \mathsf{N}_{3\mathrm{B}}} \mathsf{d}_{T'}(x) \leq 2\mathsf{e}$ by Lemma 24. ◀

7 Tighter upper bound for strings ending with \$

In this section, we consider any string ending with a unique end-marker \$, and present a tighter upper bound for the worst-case multiplicative sensitivity of e:

▶ **Lemma 26.** *For any string T of length n ending with a unique character \$, $\mathsf{MS}(\mathsf{e}, n) \leq (5\mathsf{e}+4)/\mathsf{e}$ holds.*

Proof. We first examine the upper bound in Lemma 13. When T ends with a unique character \$, the argument of Lemma 13 can be adapted as follows. Let $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$, and let $U(x)$ be defined in the same way as in Lemma 13. Since every maximal repeat in the CDAWG of any string ending with \$ has at least two outgoing edges, we have $\mathsf{d}_T(U(x)) \geq 2$. From the same reasoning as in Lemma 13, it follows that $\mathsf{d}_{T'}(x) \leq \mathsf{d}_T(U(x)) + 2$, which can be tightened to $\mathsf{d}_{T'}(x) \leq 2\mathsf{d}_T(U(x))$ when $\mathsf{d}_T(U(x)) \geq 2$.

By Lemma 12, the mapping $x \mapsto U(x)$ is injective. Therefore,

$$\sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}} \mathsf{d}_{T'}(x) \leq 2 \sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}} \mathsf{d}_T(U(x)) + 2 \leq 2\mathsf{e} + 2.$$

By applying similar arguments to the nodes considered in Lemma 20 and Lemma 25, we obtain upper bounds of $2\mathsf{e}+2$ and e, respectively.

Summing up these three bounds gives us

$$(2\mathsf{e}+2) + (2\mathsf{e}+2) + \mathsf{e} = 5\mathsf{e} + 4.$$

Hence, the total upper bound is $5\mathsf{e}+4$, as claimed. ◀

8 Conclusions

This paper proved that the worst-case multiplicative sensitivity of CDAWGs is asymptotically at most 8. Our analysis is built on new combinatorial properties of maximal repeats and their right-extensions, that are incidental to an edit operation on the strings. The only known lower bound for the multiplicative sensitivity of CDAWGs is asymptotically 2 [9]. It is intriguing future work to close the gap between these upper bound and lower bound.

References

1 Tooru Akagi, Mitsuru Funakoshi, and Shunsuke Inenaga. Sensitivity of string compressors and repetitiveness measures. *Information and Computation*, 291:104999, 2023. doi:10.1016/J.IC.2022.104999.

2 Pawel Baturo, Marcin Piatkowski, and Wojciech Rytter. Usefulness of directed acyclic subword graphs in problems related to Standard Sturmian Words. *Int. J. Found. Comput. Sci.*, 20(6):1005–1023, 2009. doi:10.1142/S0129054109007017.

3 Djamal Belazzougui and Fabio Cunial. Fast label extraction in the CDAWG. In *SPIRE 2017*, pages 161–175, 2017. doi:10.1007/978-3-319-67428-5_14.

4 Anselm Blumer, J. Blumer, David Haussler, Ross McConnell, and Andrzej Ehrenfeucht. Complete inverted files for efficient text retrieval and analysis. *J. ACM*, 34(3):578–595, 1987. doi:10.1145/28869.28873.

5 Dany Breslauer and Zvi Galil. An optimal $o(\log\log n)$ time parallel string matching algorithm. *SIAM J. Comput.*, 19(6):1051–1058, 1990. doi:10.1137/0219072.

6 Moses Charikar, Eric Lehman, Ding Liu, Rina Panigrahy, Manoj Prabhakaran, Amit Sahai, and Abhi Shelat. The smallest grammar problem. *IEEE Trans. Inf. Theory*, 51(7):2554–2576, 2005. doi:10.1109/TIT.2005.850116.

7 Alan M. Cleary and Jordan Dood. Constructing the CDAWG CFG using lcp-intervals. In *DCC 2023*, pages 178–187, 2023. doi:10.1109/DCC55655.2023.00026.

8 Maxime Crochemore, F. Mignosi, and A. Restivo. Automata and forbidden words. *Information Processing Letters*, 67(3):111–117, 1998. doi:10.1016/S0020-0190(98)00104-5.

9 Hiroto Fujimaru, Yuto Nakashima, and Shunsuke Inenaga. Tight bounds for the sensitivity of CDAWGs with left-end edits. *Acta Informatica*, 62(12), 2025. doi:10.1007/S00236-025-00478-Y.

10 Sara Giuliani, Shunsuke Inenaga, Zsuzsanna Lipták, Nicola Prezza, Marinella Sciortino, and Anna Toffanello. Novel results on the number of runs of the Burrows-Wheeler-transform. In *SOFSEM*, pages 249–262, 2021. doi:10.1007/978-3-030-67731-2_18.

11 Sara Giuliani, Shunsuke Inenaga, Zsuzsanna Lipták, Giuseppe Romana, Marinella Sciortino, and Cristian Urbina. Bit catastrophes for the Burrows-Wheeler transform. *Theory of Computing Systems*, 69(19), 2025. doi:10.1007/S00224-024-10212-9.

12 Peaker Guo, Patrick Eades, Anthony Wirth, and Justin Zobel. Exploiting new properties of string net frequency for efficient computation. In *CPM 2024*, volume 296 of *LIPIcs*, pages 16:1–16:16, 2024. doi:10.4230/LIPIcs.CPM.2024.16.

13 Lucian Ilie and William F Smyth. Minimum unique substrings and maximum repeats. *Fundamenta Informaticae*, 110(1-4):183–195, 2011. doi:10.3233/FI-2011-536.

14 Shunsuke Inenaga. Linear-size suffix tries and linear-size CDAWGs simplified and improved. *Acta Informatica*, 61(4):445–468, 2024. doi:10.1007/S00236-024-00465-9.

15 Shunsuke Inenaga, Hiromasa Hoshino, Ayumi Shinohara, Masayuki Takeda, Setsuo Arikawa, Giancarlo Mauri, and Giulio Pavesi. On-line construction of compact directed acyclic word graphs. *Discrete Applied Mathematics*, 146(2):156–179, 2005. doi:10.1016/J.DAM.2004.04.012.

16 Shunsuke Inenaga, Takuya Mieno, Hiroki Arimura, Mitsuru Funakoshi, and Yuta Fujishige. Computing minimal absent words and extended bispecial factors with CDAWG space. In *IWOCA 2024*, volume 14764 of *Lecture Notes in Computer Science*, pages 327–340, 2024. doi:10.1007/978-3-031-63021-7_25.

17 Tomasz Kociumaka, Gonzalo Navarro, and Nicola Prezza. Toward a definitive compressibility measure for repetitive sequences. *IEEE Trans. Inf. Theory*, 69(4):2074–2092, 2023. doi:10.1109/TIT.2022.3224382.

18 Guillaume Lagarde and Sylvain Perifel. Lempel-Ziv: a “one-bit catastrophe” but not a tragedy. In *SODA*, pages 1478–1495, 2018. doi:10.1137/1.9781611975031.97.

19 William Merrill, Noah A. Smith, and Yanai Elazar. Evaluating n-gram novelty of language models using Rusty-DAWG. In *EMNLP 2024*, pages 14459–14473, 2024. doi:10.18653/V1/2024.EMNLP-MAIN.800.

20 Takuya Mieno and Shunsuke Inenaga. Space-efficient online computation of string net occurrences. In *CPM 2025*, LIPIcs, pages 23:1–23:13, 2025. doi:10.4230/LIPIcs.CPM.2025.23.

21 Masamichi Miyazaki, Ayumi Shinohara, and Masayuki Takeda. An improved pattern matching algorithm for strings in terms of straight line programs. *Journal of Discrete Algorithms*, 1(1):187–204, 2000.

22 Yuto Nakashima, Dominik Köppl, Mitsuru Funakoshi, Shunsuke Inenaga, and Hideo Bannai. Edit and alphabet-ordering sensitivity of lex-parse. In *MFCS 2024*, volume 306 of *LIPIcs*, pages 75:1–75:15, 2024. doi:10.4230/LIPIcs.MFCS.2024.75.

23 Gonzalo Navarro. Computing mems and relatives on repetitive text collections. *ACM Trans. Algorithms*, 21(1):12:1–12:33, 2025. doi:10.1145/3701561.

24 Gonzalo Navarro, Carlos Ochoa, and Nicola Prezza. On the approximation ratio of ordered parsings. *IEEE Trans. Inf. Theory*, 67(2):1008–1026, 2021. doi:10.1109/TIT.2020.3042746.

25 Jakub Radoszewski and Wojciech Rytter. On the structure of compacted subword graphs of Thue-Morse words and their applications. *J. Discrete Algorithms*, 11:15–24, 2012. doi:10.1016/J.JDA.2011.01.001.

26 Wojciech Rytter. Application of Lempel-Ziv factorization to the approximation of grammar-based compression. *Theor. Comput. Sci.*, 302(1-3):211–222, 2003. doi:10.1016/S0304-3975(02)00777-6.

27 Wojciech Rytter. The structure of subword graphs and suffix trees of Fibonacci words. *Theor. Comput. Sci.*, 363(2):211–223, 2006. doi:10.1016/J.TCS.2006.07.025.

28 James A. Storer and Thomas G. Szymanski. Data compression via textual substitution. *J. ACM*, 29(4):928–951, 1982. doi:10.1145/322344.322346.

29 Masayuki Takeda, Tetsuya Matsumoto, Tomoko Fukuda, and Ichiro Nanri. Discovering characteristic expressions from literary works: a new text analysis method beyond n-gram statistics and KWIC. In *Discovery Science 2000*, pages 112–126, 2000. doi:10.1007/3-540-44418-1_10.

30 Peter Weiner. Linear pattern matching algorithms. In *Proceedings of the 14th Annual Symposium on Switching and Automata Theory*, pages 1–11, 1973. doi:10.1109/SWAT.1973.13.

31 J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977. doi:10.1109/TIT.1977.1055714.

32 Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *IEEE Trans. Inf. Theory*, 24(5):530–536, 1978. doi:10.1109/TIT.1978.1055934.