On the Additive Sensitivity of CDAWG-grammars

Hiroto Fujimaru[1] * and Shunsuke Inenaga[2] **

[1] Department of Information Science and Technology, Kyushu University, Japan
fujimaru.hiroto.134@s.kyushu-u.ac.jp
[2] Department of Informatics, Kyushu University, Japan
inenaga.shunsuke.380@m.kyushu-u.ac.jp

Abstract. The *compact directed acyclic word graph* (*CDAWG*) [Blumer et al. 1987] of a string is the minimal compact automaton that recognizes all the suffixes of the string. CDAWGs can be used for various string tasks including text pattern searching, data compression, and pattern discovery. The *CDAWG-grammar* [Belazzougui & Cunial 2017] is a grammar-based text compression based on the CDAWG, which allows for representing the CDAWG in $O(\mathsf{e})$ space without storing the string, where e denotes the number of CDAWG edges. Let g be the size of the CDAWG-grammar for the input string T. We exhibit an infinite family of strings for which the CDAWG-grammar size increases by $3g-21$ after one edit, and we show that the worst-case additive sensitivity is upper bounded by $8g+4$.

1 Introduction

The *compact directed acyclic word graph* (*CDAWG*) [5] of a string is the minimal compact automaton that recognizes all the suffixes of the string. CDAWGs are known to be useful for various string tasks including text pattern searching [7,10], data compression [4,3,13], and pattern discovery [14].

CDAWGs can be obtained by merging isomorphic subtrees of suffix trees [5,4,3]. Due to this nature, as in the case of suffix trees of which each edge string label x is represented by a pair (i, j) of positions satisfying $T[i..j] = x$, classical implementations of CDAWGs also require $\Theta(n)$ space to explicitly store a copy of the string T being indexed, where $n = |T|$. This significantly limits the size of strings on which the CDAWGs are built. Still, the number e of the edges in the CDAWG tends to be small, for some highly repetitive strings, in particular, $\mathsf{e} = O(\log n)$ holds for the standard Sturmian words and Thue-Morse words [12,2,11].

The task of representing the CDAWG with $O(\mathsf{e})$ space was first achieved by Belazzougui and Cunial [4,3], via a connection from CDAWGs to *grammar-based compressions*. They observed that the reversed CDAWG induces a context-free grammar (CFG) that only generates the input string T, which is hereafter named as the *CDAWG-grammar*. By augmenting the CDAWG-grammar with a constant-time level ancestor data structure, the CDAWG can be stored in $O(\mathsf{e})$ space while retaining the ability to perform optimal $O(m + occ)$-time pattern matching, where m is the pattern length and occ is the number of occurrences to report [4,3]. While the CDAWG-grammar can easily be obtained via the graph topology of the corresponding CDAWG, it is also possible to build the CDAWG-grammar *directly* from suffix-array-based structures [6], without the need of explicitly building the CDAWG.

* Supported by JST BOOST Grant Number JPMJBS2406.
** Supported by JSPS KAKENHI Grant Numbers JP23K24808 and JP23K18466.

Hiroto Fujimaru, Shunsuke Inenaga: On the Additive Sensitivity of CDAWG-grammars, pp. 43–57.
Proceedings of PSC 2026, Jan Holub and Jan Žďárek (Eds.), ISBN 978-80-01-07576-0 © Czech Technical University in Prague, Czech Republic

The focus of this paper is to analyze the advantage of CDAWG-grammar in terms of *sensitivity*: The sensitivity of a repetitiveness measure c asks how much the measure size increases when a single-character edit operation is performed on the input string [1], and thus the sensitivity allows one to evaluate the robustness of the measure/compressor against errors/edits. Recent works have revealed the sensitivity of CDAWG size e in the case of left-end edit operations [8] as well as in the general case [9]. As for the CDAWG-grammar size g, it is known that $\mathsf{g} = \mathsf{e} - \mathrm{v}^{(1)}$ holds [4,3], where $\mathrm{v}^{(1)}$ denotes the number of CDAWG nodes of in-degree one. To obtain non-trivial bounds for the worst-case additive sensitivity of the CDAWG-grammar size g, we present new properties of $\mathrm{v}^{(1)}$ which were not previously well understood.

In this paper, we first provide an infinite family of strings such that $\mathsf{g}(T') - \mathsf{g}(T) = 3\mathsf{g} - 21$ for the one-character edit $T \mapsto T'$. After presenting a straightforward upper bound of $15\mathsf{g} + 4$, we give a non-trivial tighter upper bound of $8\mathsf{g} + 4$ based on combinatorial properties of CDAWGs and CDAWG-grammars. Thus, on the infinite sequence of input lengths realized by our construction, the lower and upper bounds are tight up to a constant factor.

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. Let $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$. If $T = uvw$, then 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 $1 \le i \le n$, 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 $1 \le i \le j \le n$. For $0 \le k \le |T|$, let $\varepsilon = T[k+1..k]$.

Maximal substrings. For two strings u and T, let $\mathsf{BegPos}(u, T) = \{i \mid T[i..i+|u|-1] = u\}$ and $\mathsf{EndPos}(u, T) = \{i \mid T[i-|u|+1..i] = u\}$ denote the sets of beginning and ending positions of u in T, respectively. For any substrings $u, v \in \mathsf{Substr}(T)$ of a string T, define $u \equiv^{\mathrm{L}}_T v \Leftrightarrow \mathsf{EndPos}(u, T) = \mathsf{EndPos}(v, T)$, and $u \equiv^{\mathrm{R}}_T v \Leftrightarrow \mathsf{BegPos}(u, T) = \mathsf{BegPos}(v, T)$. A substring $x \in \mathsf{Substr}(T)$ of T is said to be *left-maximal* in T if $ax, bx \in \mathsf{Substr}(T)$ for some two distinct characters $a, b \in \Sigma$, or $x \in \mathsf{Prefix}(T)$. Similarly, x is said to be *right-maximal* in T if $xa, xb \in \mathsf{Substr}(T)$ for some two distinct characters $a, b \in \Sigma$, or $x \in \mathsf{Suffix}(T)$. Let $\mathsf{LeftM}(T)$ and $\mathsf{RightM}(T)$ denote the sets of left-maximal and right-maximal substrings in T, respectively. It is known [5] that $x \in \mathsf{LeftM}(T)$ (resp. $x \in \mathsf{RightM}(T)$) iff x is the longest member of the equivalence class under $\equiv^{\mathrm{L}}_T$ (resp. under $\equiv^{\mathrm{R}}_T$).

Let $\mathsf{M}(T) = \mathsf{LeftM}(T) \cap \mathsf{RightM}(T)$. Any element of $\mathsf{M}(T)$ is said to be *maximal* in T.

A string x is said to be a repeat in a string T if $|\mathsf{BegPos}(x, T)| = |\mathsf{EndPos}(x, T)| \ge 2$. Let $\mathsf{MR}(T) = \mathsf{M}(T) \setminus \{T\}$ denote the set of *maximal repeats* in T.

For any substring S of a string T, we define its *right-representative* as $\mathsf{rexp}_T(S) = S\beta$, where $\beta \in \Sigma^*$ is the shortest string such that $S\beta$ is right-maximal in T.

CDAWGs. The *suffix tree* [15] of a string T is a rooted tree such that each edge is labeled by a non-empty substring of T and the path from the root to each node v represents a right-maximal substring in $\mathsf{RightM}(T)$. By assuming a unique end-marker \$ at the right-end of T, every internal node of the suffix tree of T has at least two children, and there is a one-to-one correspondence between the leaves and the

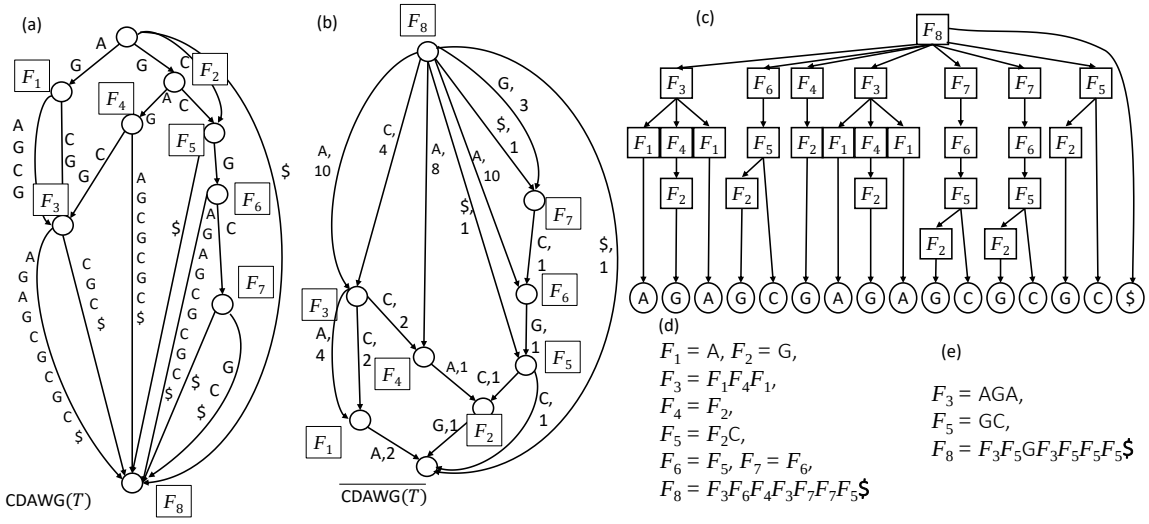


Figure 1. (a) $\mathsf{CDAWG}(T)$ of string T = AGAGCGAGAGCGCGC\$ for which $\mathsf{M}(T) = \{\varepsilon, \mathtt{G}, \mathtt{GC}, \mathtt{AG}, \mathtt{GAG}, \mathtt{GCG}, \mathtt{GCGC}, \mathtt{AGAGCG}, T\}$. The number of right-extensions of $\mathsf{CDAWG}(T)$ is the number $\mathsf{e}(T)$ of edges, which is 18 in this example. (b) The reversed DAG $\overline{\mathsf{CDAWG}(T)}$, where each edge is labeled by the initial character and the length of the edge's original label. (c) The derivation tree $\mathcal{T}(T)$ obtained by unfolding $\overline{\mathsf{CDAWG}(T)}$. (d) The grammar rules obtained from $\mathcal{T}(T)$. (e) The resulting grammar $\mathsf{G}_{\mathsf{CDAWG}}(T)$ without redundant rules. This grammar size is 13.

suffixes of T. The *compact directed acyclic word graph* (*CDAWG*) [5] of a string T, denoted $\mathsf{CDAWG}(T) = (\mathsf{V}_T, \mathsf{E}_T)$, is an edge-labeled smallest DAG that is obtained by merging isomorphic subtrees of the suffix tree for T. See the left diagram of Fig. 1 for a concrete example of CDAWGs. It is known that there is a one-to-one correspondence between the nodes of $\mathsf{CDAWG}(T)$ and the maximal substrings in $\mathsf{M}(T)$, such that the longest string represented by each node of $\mathsf{CDAWG}(T)$ is a maximal substring. In what follows, we identify the nodes of $\mathsf{CDAWG}(T)$ with the maximal substrings in $\mathsf{M}(T)$. In particular, we identify the non-sink nodes of $\mathsf{CDAWG}(T)$ with the maximal repeats in $\mathsf{MR}(T)$. For any $x \in \mathsf{MR}(T)$, $\mathsf{d}_T(x)$ denotes the out-degree of the node x in $\mathsf{CDAWG}(T)$. Similar to suffix trees, every internal node x of the $\mathsf{CDAWG}(T)$ has at least two children if T terminates with \$. Hence, the following fact holds.

Fact 1 *If T terminates with* \$, $\mathsf{d}_T(x) \geq 2$ *for any* $x \in \mathsf{MR}(T)$.

The *size* of $\mathsf{CDAWG}(T) = (\mathsf{V}_T, \mathsf{E}_T)$ for a string T of length n is the number $\mathsf{e}(T) = |\mathsf{E}_T|$ of edges in $\mathsf{CDAWG}(T)$, which is also referred to as the number of right-extensions of maximal repeats in T. Formally, a character $a \in \Sigma$ is a right-extension of a substring x in T if $xa \in \mathsf{Substr}(T)$; the distinct right-extensions of a CDAWG node are represented by its outgoing edges.

CDAWG-grammars. A *context-free grammar* (CFG) $\mathcal{G}$ which generates only a string T is called a grammar compression for T. The size of $\mathcal{G}$ is the total length of the right-hand sides of all the production rules in $\mathcal{G}$.

Suppose that the input string T terminates with a unique end-marker \$. The *CDAWG grammar* [4,3] of T, denoted $\mathsf{G}_{\mathsf{CDAWG}}(T)$, is a grammar compression for T that is built on $\mathsf{CDAWG}(T)$ as follows (see also Fig. 1): Let $\overline{\mathsf{CDAWG}(T)}$ denote the DAG obtained by reversing all edges of $\mathsf{CDAWG}(T)$. A critical observation is that, since $\mathsf{CDAWG}(T)$ is a DAG obtained by merging subtrees of the suffix tree for T, the source-to-sink paths in $\mathsf{CDAWG}(T)$ have a one-to-one correspondence with the suffixes of T, and so do the corresponding paths in $\overline{\mathsf{CDAWG}(T)}$. We label each

non-source node v in $\mathsf{CDAWG}(T)$ by a non-terminal F_i for $1 \leq i \leq |\mathsf{V}_T| - 1$ in an arbitrary topological order, and let p_k denote the path of $\mathsf{CDAWG}(T)$ that represents the suffix $T[k..|T|]$ starting at position k in T. Then, we build a tree $\mathcal{T}(T)$ by inserting, for each $1 \leq k \leq |T|$, the reversed path $\overline{p_k}$ of $\overline{\mathsf{CDAWG}(T)}$ as the path of $\mathcal{T}(T)$ from the root to the kth leaf that corresponds to the kth character $T[k]$. In other words, $\mathcal{T}(T)$ is the tree obtained by unfolding $\overline{\mathsf{CDAWG}(T)}$ into a tree. Observe that $\mathcal{T}(T)$ is the derivation tree of a grammar that generates (only) T, and its DAG representation is $\overline{\mathsf{CDAWG}(T)}$. This initial grammar may contain redundant unary rules whose right-hand side consists of a single symbol, either a terminal or a non-terminal (e.g., $F_6 \to F_5$ in Diagram (c) of Fig. 1). We remove all such redundant rules, and the resulting grammar is the CDAWG grammar $\mathsf{G}_{\mathsf{CDAWG}}(T)$ for T.

Let $\mathsf{g}(T)$ denote the size of $\mathsf{G}_{\mathsf{CDAWG}}(T)$. Let $F_i \to w \in (\Sigma \cup \{F_1, \ldots, F_{i-1}\})^+$ be a production in $\mathsf{G}_{\mathsf{CDAWG}}(T)$. The length of the right-hand side of the production in $\mathsf{G}_{\mathsf{CDAWG}}(T)$ is equal to the number of outgoing edges of the node of $\overline{\mathsf{CDAWG}(T)}$ that corresponds to F_i. This is equal to the number of in-edges of the node of $\mathsf{CDAWG}(T)$. Thus, we have $\mathsf{g}(T) \leq \mathsf{e}(T)$. Recalling that the redundant rules have been removed in $\mathsf{G}_{\mathsf{CDAWG}}(T)$, we obtain the following equation with the number $\mathrm{v}^{(1)}(T)$ of nodes of in-degree 1 in $\mathsf{CDAWG}(T)$.

Fact 2 $\mathsf{g}(T) = \mathsf{e}(T) - \mathrm{v}^{(1)}(T)$.

3 Sensitivity of CDAWG-grammar g and naive bound

We define the worst-case *additive sensitivity* of the CDAWG-grammar for any string of length n that terminates with \$ as follows:

$$\begin{aligned}\mathsf{AS}(\mathsf{g}, n) &= \max_{T \in \Sigma^n, T' \in \mathcal{K}(T,1)} \{\mathsf{g}(T') - \mathsf{g}(T) \mid T[|T|] = T'[|T'|] = \$\} \\ &= \max_{T \in \Sigma^n, T' \in \mathcal{K}(T,1)} \{\mathsf{e}(T') - \mathsf{e}(T) + \mathrm{v}^{(1)}(T) - \mathrm{v}^{(1)}(T') \mid T[|T|] = T'[|T'|] = \$\} \quad (1)\end{aligned}$$

where $\mathcal{K}(T, 1)$ denotes the set of strings of edit distance 1 from T. The second equation is due to Fact 2.

It follows from Fact 1 that $2\mathrm{v}^{(1)}(T) \leq 2|\mathsf{MR}(T)| \leq \mathsf{e}(T)$. Thus, by Fact 2 we get $\mathsf{e}(T)/2 \leq \mathsf{g}(T) \leq \mathsf{e}(T)$. Hamai et al. [9] showed that $\mathsf{e}(T') \leq 8\mathsf{e}(T) + 4$ holds for any strings T and $T' \in \mathcal{K}(T, 1)$. Using these inequalities, it immediately follows that $\mathsf{g}(T') \leq \mathsf{e}(T') \leq 8\mathsf{e}(T) + 4 \leq 16\mathsf{g}(T) + 4$, which implies $\mathsf{AS}(\mathsf{g}, n) \leq 15\mathsf{g}(T) + 4$. In the following sections, we improve the constant factor of $\mathsf{AS}(\mathsf{g}, n)$ from 15 to 8.

4 Lower bound for the sensitivity of CDAWG-grammar

Below, we present a lower bound for $\mathsf{AS}(\mathsf{g}, n)$:

Theorem 1. *There exists an infinite family of strings T such that $\mathsf{g}(T') - \mathsf{g}(T) = 3\mathsf{g}(T) - 21$ with $T' \in \mathcal{K}(T, 1)$. Thus, the worst-case additive sensitivity is lower bounded by $3\mathsf{g} - 21$ for infinitely many input lengths.*

Proof. Let $T = (ab)^m c(ab)^m c\$$ for an integer $m \geq 2$, where $a, b, c, \$ \in \Sigma$ are pairwise distinct characters. Let T' be obtained from T by substituting the first c with b, namely $T' = (ab)^m b(ab)^m c\$$. Clearly, $T' \in K(T, 1)$, and both T and T' terminate with the same end-marker \$.

We first compute $\mathsf{g}(T)$. The maximal substrings of T, except for the sink T, are precisely

$$\varepsilon, \quad (ab), (ab)^2, \ldots, (ab)^{m-1}, \quad (ab)^m c.$$

Indeed, for $1 \leq r \leq m-1$, the string $(ab)^r$ is both left- and right-maximal: it has occurrences preceded by different left contexts (the beginning of T, b, or c) and occurrences followed by different right contexts a and c. Also, $(ab)^m c$ is both left- and right-maximal, since its two occurrences are preceded by the beginning of T and by c, and are followed by a and \$, respectively. No other substring is both left- and right-maximal: a substring not containing c is a factor of the alternating blocks $(ab)^m$, and only the even aligned factors $(ab)^r$ can have two right contexts; among them, $(ab)^m$ has only the right context c. A substring containing c can be left-maximal only when it is $(ab)^m c$ or T.

The right-extensions of these maximal repeats are as follows: $\mathsf{d}_T(\varepsilon) = 4$, $\mathsf{d}_T((ab)^r) = 2$ for $1 \leq r \leq m-1$, and $\mathsf{d}_T((ab)^m c) = 2$. Hence $\mathsf{e}(T) = 4 + 2(m-1) + 2 = 2m + 4$.

Moreover, the nodes of in-degree one are exactly

$$(ab)^2, (ab)^3, \ldots, (ab)^{m-1},$$

and therefore $\mathrm{v}^{(1)}(T) = m-2$. By Fact 2, $\mathsf{g}(T) = \mathsf{e}(T) - \mathrm{v}^{(1)}(T) = (2m+4) - (m-2) = m + 6$.

Equivalently, the corresponding reduced CDAWG grammar can be described as follows. Let $X_{\mathtt{ab}}$ be the nonterminal corresponding to the node (ab), let X_D be the nonterminal corresponding to the node $D = (ab)^m c$, and let X_T be the start nonterminal. After removing the unary rules induced by the in-degree-one nodes $(ab)^2, \ldots, (ab)^{m-1}$, the grammar has the productions

$$X_{ab} \to ab, \qquad X_D \to \underbrace{X_{ab} \cdots X_{ab}}_{m \text{ copies}} c, \qquad X_T \to X_D \, X_D \, \$.$$

The total length of the right-hand sides is $2 + (m+1) + 3 = m + 6$, in agreement with the formula above.

We next compute $\mathsf{g}(T')$. The maximal substrings of T', except for the sink T', are precisely $\varepsilon, (ab), (ab)^2, \ldots, (ab)^m, b, b(ab), b(ab)^2, \ldots, b(ab)^{m-1}$.

The substitution of the first c by b creates the unique factor bb between the two alternating blocks. This makes the strings $b(ab)^r$ left- and right-maximal for $0 \leq r \leq m-1$, while the strings $(ab)^r$ remain left- and right-maximal for $1 \leq r \leq m$. All other factors either have a unique left context or a unique right context, or contain the unique end-marker and hence do not give additional maximal repeats.

Their right-degrees are $\mathsf{d}_{T'}(\varepsilon) = 4$, $\mathsf{d}_{T'}((ab)^m) = 2$, and every other maximal repeat among the above $2m-1$ strings has three right-extensions. Thus $\mathsf{e}(T') = 4 + 3(2m-1) + 2 = 6m + 3$. Furthermore, all the following $2m$ nodes have in-degree one: $(ab), (ab)^2, \ldots, (ab)^m, b, b(ab), b(ab)^2, \ldots, b(ab)^{m-1}$. Hence $\mathrm{v}^{(1)}(T') = 2m$. Again by Fact 2, $\mathsf{g}(T') = \mathsf{e}(T') - \mathrm{v}^{(1)}(T') = (6m+3) - 2m = 4m+3$. In terms of the reduced CDAWG grammar, the situation is particularly simple: all internal maximal-repeat nodes of $\mathsf{CDAWG}(T')$ have in-degree one, and hence the unary rules corresponding to them are removed. Thus the reduced grammar has only the start production

$$X_{T'} \to \underbrace{a\,b \cdots a\,b}_{m \text{ copies of } ab} \; b \; \underbrace{a\,b \cdots a\,b}_{m \text{ copies of } ab} \; c\,\$.$$

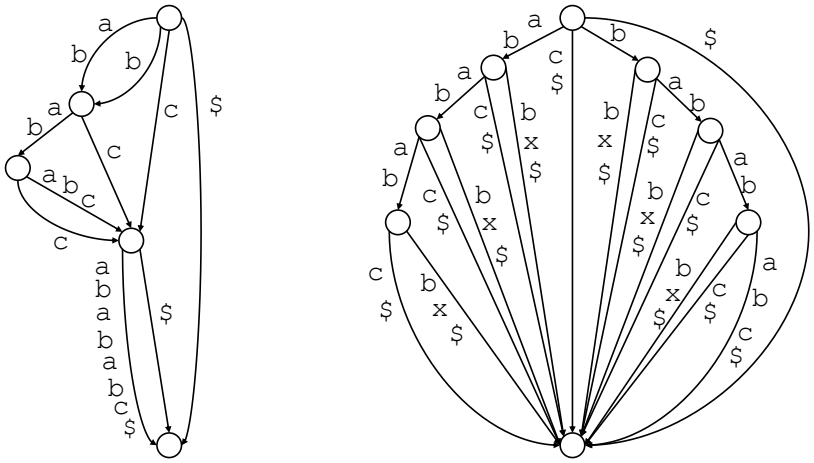


Figure 2. CDAWGs for the lower-bound example with $m = 3$. Here $T = (\mathtt{ab})^3\mathtt{c}(\mathtt{ab})^3\mathtt{c}\$ = \mathtt{abababcababababc}\$$ and $T' = (\mathtt{ab})^3\mathtt{b}(\mathtt{ab})^3\mathtt{c}\$ = \mathtt{abababbababababc}\$ = \mathtt{abababbx}\$$ with $\mathtt{x} = \mathtt{abababc}$. For T, the reduced CDAWG grammar has productions $X_{\mathtt{ab}} \to \mathtt{ab}$, $X_{\mathtt{x}} \to X_{\mathtt{ab}}X_{\mathtt{ab}}X_{\mathtt{ab}}\mathtt{c}$, and $X_T \to X_{\mathtt{x}}X_{\mathtt{x}}\$$. For T', all internal maximal-repeat nodes have in-degree one, and the reduced CDAWG grammar consists of $X_{T'} \to \mathtt{abababbababababc}\$$. We have $\mathsf{e}(T) = 10$, $\mathrm{v}^{(1)}(T) = 1$, and $\mathsf{g}(T) = 9$, while $\mathsf{e}(T') = 21$, $\mathrm{v}^{(1)}(T') = 6$, and $\mathsf{g}(T') = 15$. Hence $\mathsf{g}(T') - \mathsf{g}(T) = 6 = 3\mathsf{g}(T) - 21$.

The length of this right-hand side is $2m + 1 + 2m + 2 = 4m + 3$, again agreeing with $\mathsf{g}(T')$.

Consequently, $\mathsf{g}(T') - \mathsf{g}(T) = (4m + 3) - (m + 6) = 3m - 3$. Since $\mathsf{g}(T) = m + 6$, we obtain $\mathsf{g}(T') - \mathsf{g}(T) = 3\mathsf{g}(T) - 21$. This proves the theorem.

Fig. 2 shows the CDAWGs of the strings T and T' with $m = 3$.

5 Edge-sensitivity ingredients for CDAWG-grammars

In this section, we revisit the CDAWG sensitivity analysis of Hamai et al. [9] and derive an upper bound on $\mathsf{e}(T') - \mathsf{e}(T)$ tailored to our analysis of CDAWG-grammars.

Throughout this paper, T denotes an arbitrarily fixed string and T' denotes the string obtained by performing a single-character edit operation on T. For an insertion or substitution, i denotes the position in T' occupied by the inserted or substituted character. For a deletion of $T[i]$, i denotes the position immediately to the right of the resulting gap in T'.

Definition 2. *Let $x = T'[j..k]$ be a non-empty substring occurrence of T', where $1 \le j \le k \le |T'|$. This occurrence $[j..k]$ of x in T' is called* crossing *if it contains or touches the edit position i. More precisely, it is crossing if $k = i - 1$ or $j \le i \le k$ or $j = i + 1$ for an insertion or a substitution, and if $j \le i \le k + 1$ for a deletion.*

Whenever x has a crossing occurrence, let x_L (resp. x_R) denote its leftmost (resp. rightmost) crossing occurrence in T'. Write $[j_L..k_L]$(resp. $[j_R..k_R]$) for their intervals, where $k_L = j_L + |x| - 1$ and $k_R = j_R + |x| - 1$. We use $S_{x_L} = T'[i..k_L]$ and $S_{x_R} = T'[i..k_R]$ (Fig. 3); with the convention $T'[p..p-1] = \varepsilon$, either string may be empty. Thus these symbols are used only for strings having a crossing occurrence, and their index i is the edit position defined above.

We divide $\mathsf{MR}(T')$ into the following four subsets:

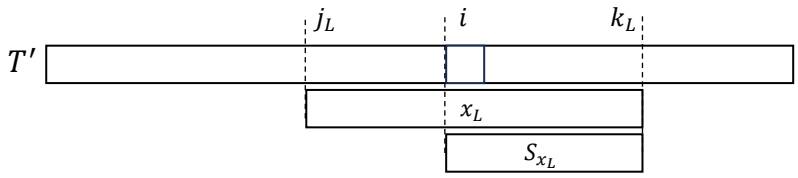


Figure 3. Illustration of x_L in T' for the case where x_L contains i.

Definition 3. *Let* $\mathsf{N} = \mathsf{MR}(T') \setminus \mathsf{MR}(T)$ *and* $\mathsf{Q} = \mathsf{MR}(T') \setminus \mathsf{N}$ *(or equivalently* $\mathsf{Q} = \mathsf{MR}(T') \cap \mathsf{MR}(T)$*). We divide* N *into* N_1*,* N_2*, and* N_3 *as* $\mathsf{N}_1 = \mathsf{N} \cap \mathsf{RightM}(T)$*,* $\mathsf{N}_2 = \mathsf{N} \cap \mathsf{LeftM}(T)$*, and* $\mathsf{N}_3 = \mathsf{N} \backslash (\mathsf{N}_1 \cup \mathsf{N}_2)$*. Further, let* $\mathsf{N}_{3\mathrm{B}}$ *denote the set of strings* $x \in \mathsf{N}_3$ *satisfying the following two conditions: (1) The crossing occurrences of* x *in* T' *have at least two distinct left contexts and at least two distinct right contexts, and (2) There is no distinct right-extension of* x *in* T' *other than the right-extension(s) of the crossing occurrence(s) of* x*. Let* $\mathsf{N}_{3\mathrm{A}} = \mathsf{N}_3 \setminus \mathsf{N}_{3\mathrm{B}}$*.*

Also, we define the number of remaining edges after edits by $\mathsf{rem_e}(T, T') = |\mathsf{E}_T \cap \mathsf{E}_{T'}|$. Thus, $\mathsf{rem_e}(T, T')$ is the number of CDAWG edges of T that remain unchanged after the edit. We write it simply as $\mathsf{rem_e}$ when it is clear from the context.

Upper bound for total out-degrees of nodes w.r.t. $\mathbf{N_1 \cup N_{3A}}$. Here, we show an upper bound for the total out-degrees of nodes corresponding to the strings that are elements of $\mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$.

Lemma 4 (Lemma 2 and 3 of [9]). *If* $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$*, then there do not exist distinct strings* $y, z \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ *with* $|x| < |y|$ *and* $|x| < |z|$ *such that both* $S_{x_L} = S_{y_F}$ *and* $S_{x_R} = S_{z_G}$ *hold at the same time, where* $F, G \in \{L, R\}$*.*

We next define, through the map U, the correspondence from $\mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ to $\mathsf{MR}(T)$. For any $x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$, if there is no $y \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ ($|y| > |x|, G \in \{L, R\}$) s.t. $S_{x_L} = S_{y_G}$, then we associate x with S_{x_L}, and otherwise we associate x with S_{x_R}.

Let $U(x)$ be the string that is obtained by extending S_{x_G}, to which x corresponds, to the left in T until becoming left-maximal in T. The following holds:

Lemma 5 (Lemmas 4 and 5 of [9]). *If* $x, y \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}$ *and* $x \neq y$*, then* $U(x), U(y) \in \mathsf{M}(T)$ *and* $U(x) \neq U(y)$*. Also,* $\mathsf{d}_{T'}(x) \leq \mathsf{d}_T(U(x)) + 2$*.*

Also, for each $x \in N_1 \cup N_{3A}$, we define $\delta_{\mathsf{N}}(x) = 2 - \max(d_{T'}(x) - d_T(U(x)), 0)$. Since the map U is injective by Lemma 5, $\sum_x \mathsf{d}_T(U(x)) \leq \mathsf{e}(T)$. Then, we have the following: $\sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}} \mathsf{d}_{T'}(x) \leq \sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}} (\mathsf{d}_T(U(x)) + 2 - \delta_{\mathsf{N}}(x)) \leq \mathsf{e}(T) + 2|\mathsf{M}(T)| - \sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}} (\delta_{\mathsf{N}}(x))$.

Lemma 6. $\sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}} \mathsf{d}_{T'}(x) \leq \mathsf{e}(T) + 2|\mathsf{M}(T)| - \sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}} \delta_{\mathsf{N}}(x)$.

Upper bound for total out-degrees of nodes w.r.t. $\mathbf{N_2 \cup Q}$. Here, we consider an upper bound for the total out-degrees of nodes corresponding to the strings that are elements of $\mathsf{N}_2 \cup \mathsf{Q}$. We divide the total out-edges of the nodes into two: the number $\mathcal{J}(T)$ of *the right-extensions of the non-crossing occurrences* and the number $\mathcal{L}(T)$ of *the right-extensions only in the crossing occurrences*, such that $\sum_{x \in \mathsf{N}_2 \cup \mathsf{Q}} \mathsf{d}_{T'}(x) = \mathcal{J}(T) + \mathcal{L}(T)$.

For $\mathcal{J}(T)$, the right-extensions of x with non-crossing occurrences in T' does not exceed the right-extensions of x in T because the non-crossing occurrences are not affected by the edit. If $x \in \mathsf{N}_2$, then x is not right-maximal in T. Thus, each node has at most one right-extension with non-crossing occurrences. On the other hand, each remaining right-extension of $x \in \mathsf{Q}$ corresponds by definition to an edge in $\mathsf{E}_T \cap \mathsf{E}_{T'}$. By summing up these, we get:

Lemma 7. $\mathcal{J}(T) \leq |\mathsf{N}_2| + \mathsf{rem_e}$.

From now on we consider the second term $\mathcal{L}(T)$. A right-extension $a \in \Sigma$ of $x \in \mathsf{N}_2 \cup \mathsf{Q}$ is said to be a *new* right-extension of x in T', if xa does not occur in T and x has a crossing occurrence that immediately precedes a in T'.

Lemma 8 (Adapted from [9]). *There is an injection from the new right-extensions of all strings in* $\mathsf{N}_2 \cup \mathsf{Q}$ *to the maximal repeats in* $\mathsf{M}(T)$ *(namely to the existing nodes in* $\mathsf{CDAWG}(T)$*), implying* $\mathcal{L}(T) \leq |\mathsf{M}(T)|$.

Putting these results together, the following lemma holds:

Lemma 9. $\sum_{x \in \mathsf{N}_2 \cup \mathsf{Q}} \mathsf{d}_{T'}(x) \leq \mathsf{rem_e} + |\mathsf{M}(T)| + |\mathsf{N}_2|$.

The following upper bound is known for $\mathsf{N}_{3\mathrm{B}}$:

Lemma 10 (Reformulation of Lemma 15 of [9]). $\sum_{x \in \mathsf{N}_{3\mathrm{B}}} \mathsf{d}_{T'}(x) \leq 2|\mathsf{MR}(T)|$.

Recall $|\mathsf{MR}(T)| = |\mathsf{M}(T) \setminus \{T\}| = |\mathsf{M}(T)| - 1$. By Lemmas 6, 9, and 10 we get:

Theorem 11. $\mathsf{e}(T') - \mathsf{e}(T) \leq \mathsf{rem_e} + 5|\mathsf{MR}(T)| - \sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}} (\delta_{\mathsf{N}}(x)) + |\mathsf{N}_2| + 3$.

6 Tighter bound for $\mathsf{AS}(\mathsf{g}, n)$

This section analyzes the second half $\mathrm{v}^{(1)}(T) - \mathrm{v}^{(1)}(T')$ of Equation 1 between the original string T and the edited string T'. By combining it with the result from Section 5, we will obtain a tighter bound $\mathsf{AS}(\mathsf{g}, n) \leq 8\mathsf{g} + 4$ (Corollary 27).

Difference in the number of nodes of in-degree one. We consider the number of $\mathsf{C}(T)$ *existing nodes whose in-degrees are changed from one after the edit.* Clearly $\mathrm{v}^{(1)}(T) - \mathrm{v}^{(1)}(T') \leq \mathsf{C}(T)$ holds.

Let $x \in \mathsf{MR}(T)$ be a node of in-degree one in the CDAWG for the original string T before the edit. We consider the following two cases:

Case 1: x is not a maximal repeat in T' after the edit (the node corresponding to x is deleted in $\mathsf{CDAWG}(T')$).
Case 2: x has two or more in-edges in $\mathsf{CDAWG}(T')$ after the edit.

Let V_1 and V_2 denote the sets of strings x in Cases 1 and 2, respectively.

In Case 2, we further define y as the (only) parent of a node $x \in \mathsf{V}_2$ in the CDAWG of the original string T. Also, let q be the longest proper suffix of x that is a maximal repeat in T. See Fig. 4. We categorize strings $x \in \mathsf{V}_2$ using the unique parent y and its suffix q, as follows:

Case 2(a): $y \notin \mathsf{MR}(T')$ (the node for y is deleted in $\mathsf{CDAWG}(T')$).
Case 2(b): $y \in \mathsf{MR}(T')$ and $q \notin \mathsf{MR}(T')$ (the node for q is deleted in $\mathsf{CDAWG}(T')$).

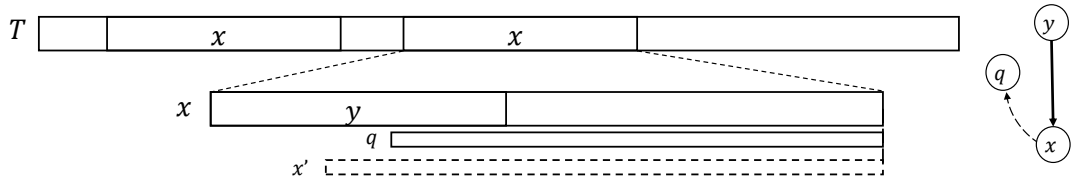


Figure 4. Illustration for the strings y and q for a string $x \in \mathsf{V}_2(T)$ and the relation of their nodes, where the dashed arc represents the suffix link. Any proper suffix x' of x that is longer than q is not a maximal repeat of T.

Case 2(c): $y, q \in \mathsf{MR}(T')$ (both of the nodes for y and q exist in $\mathsf{CDAWG}(T')$).

Let $\mathsf{V}_{2(\mathrm{a})}$, $\mathsf{V}_{2(\mathrm{b})}$, and $\mathsf{V}_{2(\mathrm{c})}$ denote the sets of strings $x \in \mathsf{MR}(T)$ which are in Case 2(a), Case 2(b), and Case 2(c), respectively.

Cases 1, 2(a), 2(b): For each $x \in \mathsf{V}_1 \cup \mathsf{V}_{2(\mathrm{a})} \cup \mathsf{V}_{2(\mathrm{b})}$, we define $f(x) \in \mathsf{MR}(T)$ as follows: (1) $f(x) = x$ if $x \in \mathsf{V}_1$, (2) $f(x) = y$ if $x \in \mathsf{V}_{2(\mathrm{a})}$, and (3) $f(x) = q$ if $x \in \mathsf{V}_{2(\mathrm{b})}$. Also, let $f(V) = \{f(x) \mid x \in V\}$ for each $V \in \{\mathsf{V}_1, \mathsf{V}_{2(\mathrm{a})}, \mathsf{V}_{2(\mathrm{b})}\}$.

By definition, the node for $f(x)$ and its out-edges are deleted in $\mathsf{CDAWG}(T')$. Since we have assumed that T terminates with a unique end-marker \$, $\mathsf{d}_T(f(x)) \geq 2$. Thus, at least $2|f(\mathsf{V}_1) \cup f(\mathsf{V}_{2(\mathrm{a})}) \cup f(\mathsf{V}_{2(\mathrm{b})})|$ edges are removed.

By the definition of $\mathsf{rem_e}$, we have that:

Lemma 12. *At least* $|\mathsf{V}_1| + |\mathsf{V}_{2(\mathrm{a})}| + |\mathsf{V}_{2(\mathrm{b})}|$ *existing edges are removed after edits. Then,* $\mathsf{e}(T) \geq \mathsf{rem_e} + (|\mathsf{V}_1| + |\mathsf{V}_{2(\mathrm{a})}| + |\mathsf{V}_{2(\mathrm{b})}|)$.

For $f(x)$, we have the following lemmas.

Lemma 13. *If* $x_1, x_2 \in \mathsf{V}_{2(\mathrm{a})}$ *and* $x_1 \neq x_2$*, then* $y_1 \neq y_2$.

Proof. Suppose that $y_1 = y_2$. Then y_1 is the longest proper prefix of both x_1 and x_2 that is a maximal repeat in T, which implies that $a \neq b$ where $a, b \in \Sigma$, $y_1 a$ is a prefix of x_1, and $y_1 b$ is a prefix of x_2. Since x_1, x_2 are substrings of T' after the edit, $y_1 \in \mathsf{M}(T')$, which is a contradiction. □

By symmetry, the following lemma also holds.

Lemma 14. *If* $x_1, x_2 \in \mathsf{V}_{2(\mathrm{b})}$ *and* $x_1 \neq x_2$*, then* $q_1 \neq q_2$.

Lemma 15. *If* $x_1 \in \mathsf{V}_{2(\mathrm{a})}$ *and* $x_2 \in \mathsf{V}_{2(\mathrm{b})}$*, then* $f(x_1) \neq f(x_2)$.

Proof. Suppose that $f(x_1) = f(x_2)$. By definition, $f(x_1)$ is a prefix of x_1 and a suffix of x_2. Since $x_1, x_2 \in \mathsf{M}(T')$, $f(x_1) \in \mathsf{M}(T')$, which is a contradiction. □

By Lemmas 13 and 14, there is an injective mapping between V and $f(V)$ for each $V \in \{\mathsf{V}_1, \mathsf{V}_{2(\mathrm{a})}, \mathsf{V}_{2(\mathrm{b})}\}$. Thus, $|V| = |f(V)|$ holds for each V. Also, for each element x, x can be counted twice in $f(\mathsf{V}_1)$ and $f(\mathsf{V}_{2(\mathrm{a})})$ or $f(\mathsf{V}_{2(\mathrm{b})})$, while there is no $x \in f(\mathsf{V}_{2(\mathrm{a})}) \cap f(\mathsf{V}_{2(\mathrm{b})})$ by Lemma 15.

As a result, $|\mathsf{V}_1| + |\mathsf{V}_{2(\mathrm{a})}| + |\mathsf{V}_{2(\mathrm{b})}| \leq 2|f(\mathsf{V}_1) \cup f(\mathsf{V}_{2(\mathrm{a})}) \cup f(\mathsf{V}_{2(\mathrm{b})})|$ holds. The number of removed edges is at least the quantity on the right-hand side, which implies that at least $|\mathsf{V}_1| + |\mathsf{V}_{2(\mathrm{a})}| + |\mathsf{V}_{2(\mathrm{b})}|$ edges are removed.

Case 2(c): We now consider the case $x \in \mathsf{V}_{2(\mathrm{c})}$. If the in-degree of x is at least 2 in $\mathsf{CDAWG}(T')$, the following lemma holds: For accessibility, recall that $\mathsf{I}_T(x) = \{y \mid (y, x) \in \mathsf{E}_T\}$ is the set of in-neighbors of x in $\mathsf{CDAWG}(T)$.

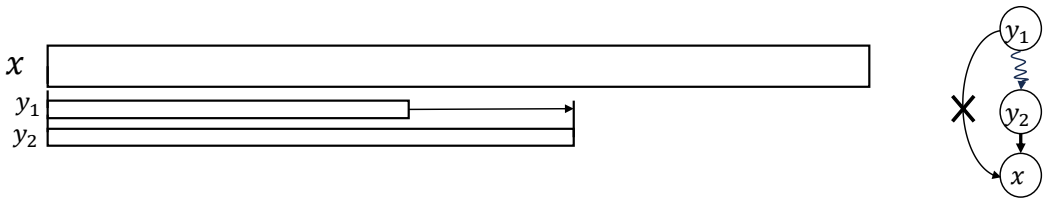


Figure 5. Illustration for the proof of Lemma 16. The case does not occur.

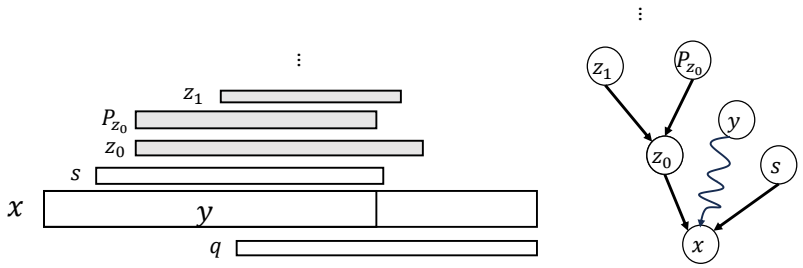


Figure 6. Illustration for z_k and P_{z_k} in x and $\mathsf{CDAWG}(T')$. Both s and z_0 are elements of $I_{T'}(x)$. Among them, z_0 is chosen so that its occurrence ends at the rightmost position.

Lemma 16. *If $y_1, y_2 \in \mathsf{I}_{T'}(x)$, at least one of y_1, y_2 has a non-prefix occurrence in x.*

Proof. Assume, to the contrary, that each of y_1 and y_2 occurs exactly once in x, as a prefix of x (see Fig. 5). Without loss of generality, let $|y_1| < |y_2|$. Let a be the character such that $y_1 a$ is a prefix of x. Since y_1 and y_2 are prefixes of x, the right-representative $\mathsf{rexp}_{T'}(y_1 a)$ of $y_1 a$ is not longer than y_2 and thus it is not x. If there is no other occurrence of y_1 in x, then $(y_1, x) \notin \mathsf{E}_{T'}$, but this is a contradiction. Thus, at least one of y_1, y_2 has a non-prefix occurrence in x. □

By Lemma 16, if $\mathsf{I}_{T'}(x)$ has at least two elements, then at least one in-neighbor in $\mathsf{I}_{T'}(x)$ has a non-prefix occurrence in x. Now, for $k \geq 0$, we define z_k and P_{z_k} as follows:

- z_0 is the maximal repeat in $\mathsf{I}_{T'}(x)$ whose ending position within x, denoted $j^{(x)}_{z_0}$, is the rightmost among those in $\mathsf{I}_{T'}(x)$.
- P_{z_k} is the maximal repeat which is the longest prefix of z_k where $P_{z_k} \in \mathsf{I}_{T'}(z_k)$.
- z_{k+1} is the maximal repeat in $\mathsf{I}_{T'}(z_k)$ whose ending position within z_k in x, denoted $j^{(x)}_{z_{k+1}}$, is the rightmost among those in $\mathsf{I}_{T'}(z_k)$.

If there exist two or more maximal repeats with the same rightmost ending position, then the shortest one is selected as z_k. P_{z_k} and z_{k+1} are undefined when z_k is a suffix of y or z_k is of in-degree one. See also Fig. 6.

For any k, let $i^{(x)}_{z_k}$ be the beginning position of z_k in x. Since $z_{k+1} \in \mathsf{I}_{T'}(z_k)$, we have $(z_{k+1}, z_k) \in \mathsf{E}_{T'}$, $|z_k| > |z_{k+1}|$ holds. Therefore, the recursive construction of P_{z_k} and z_{k+1} stops at some z_k.

Let $i^{(x)}_y$ and $j^{(x)}_y$ (resp. $i^{(x)}_q$ and $j^{(x)}_q$) denote the beginning and ending positions of y (resp. q) as prefix (resp. suffix) of x. For q and z_k, we have:

Lemma 17. *For $k \geq 0$, $i^{(x)}_{z_k} < i^{(x)}_q < j^{(x)}_y \leq j^{(x)}_{z_k}$ holds.*

Proof. We prove the lemma by induction (see also Fig. 7).

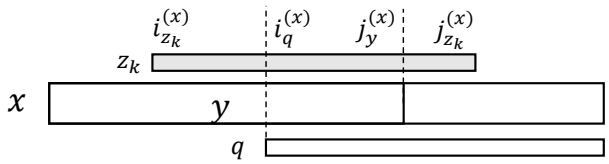


Figure 7. Illustration for Lemma 17.

Let $z'_0 = x[j^{(x)}_{z_0} + 1..|x|]$. Then $z_0 z'_0$ is a suffix of x, and z'_0 is the label of the edge (z_0, x) in $\mathsf{CDAWG}(T')$. Assume on the contrary that $i^{(x)}_{z_0} \geq i^{(x)}_q$. Then $z_0 z'_0$ is also a suffix of q and z'_0 is the label of the edge (z_0, q), which is a contradiction. Assume on the contrary that $j^{(x)}_y > j^{(x)}_{z_0}$. Then z_0 is a substring of y. Since $y \in \mathsf{MR}(T')$, every path from z_0 to x goes through the node y. This implies that z_0 does not have an out-edge to x, a contradiction. By the definition of q, any suffix of x that is longer than q is represented by the same node as x. Since x has in-degree one in $\mathsf{CDAWG}(T)$ and its unique in-neighbor is y, y and q must overlap in x. Hence $i^{(x)}_q < j^{(x)}_y$. Thus, $i^{(x)}_{z_0} < i^{(x)}_q < j^{(x)}_y \leq j^{(x)}_{z_0}$.

If $i^{(x)}_{z_k} < i^{(x)}_q < j^{(x)}_y \leq j^{(x)}_{z_k}$ holds and z_{k+1} is generated, then $j^{(x)}_y < j^{(x)}_{z_k}$ holds by definition. Assume on the contrary that $i^{(x)}_{z_{k+1}} \geq i^{(x)}_q$. Let s_{z_k} denote the overlap between z_k and q. Then z_{k+1} is a substring of s_{z_k}. Since $z_k, q \in \mathsf{MR}(T')$, $s_{z_k} \in \mathsf{MR}(T')$. Thus, every path from z_{k+1} to z_k goes through the node s_{z_k} (and thus z_{k+1} does not have an out-edge to z_k), a contradiction. Assume on the contrary that $j^{(x)}_y > j^{(x)}_{z_{k+1}}$. Then z_{k+1} is a substring of y. Since $y \in \mathsf{MR}(T')$, every path from z_{k+1} to x goes through the node y. This implies that z_{k+1} does not have an out-edge to z_k, which is a contradiction. Thus, $i^{(x)}_{z_{k+1}} < i^{(x)}_q < j^{(x)}_y \leq j^{(x)}_{z_{k+1}}$. □

Consider the string z_k such that z_{k+1} is not generated. Let $z_l(x) = z_k$. In the sequel, we consider two sub-cases of Case 2(c):

Case 2(c)-1: $z_l(x)$ has only a single in-edge and it is not a suffix of y. Let $\mathsf{V}_{2(c)-1}(T)$ denote the set of strings x in Case 2(c)-1. We remark that $z_l(x)$ has only a single in-edge, and $z_l(x)$ is a new node whose in-degree is one in $\mathsf{CDAWG}(T')$, instead of x. Also, we have the following lemma:

Lemma 18. *If x_1 and x_2 are distinct strings in Case 2(c)-1, then $z_l(x_1) \neq z_l(x_2)$.*

Proof. Suppose that $z_l(x_1) = z_l(x_2)$. Let x' be the maximal common substring of x_1 and x_2 which contains $z_l(x_1) = z_l(x_2)$. By the maximality of x' as a common substring of the two occurrences in T, x' is a maximal repeat in T. Also, let y_1 (resp. q_1) denote the longest proper prefix (resp. suffix) of x_1 that is in $\mathsf{MR}(T)$. Since $i^{(x_1)}_{z_l(x_1)} < i^{(x_1)}_{q_1} < j^{(x_1)}_{y_1} \leq j^{(x_1)}_{z_l(x_1)}$ by Lemma 17 and $z_l(x_1)$ is not a suffix of y_1, $i^{(x_1)}_{x'} < i^{(x_1)}_{q_1} < j^{(x_1)}_{y_1} < j^{(x_1)}_{x'}$ also holds. This also implies that $x' \neq y_1$. Since y_1 is the only node that has an out-edge to x_1 in $\mathsf{CDAWG}(T)$, and q_1 is the longest maximal repeat of T that is a proper suffix of x_1, we derive a contradiction as follows:

If the considered occurrence of x' is a suffix of x_1, then x' itself is a suffix of x_1 longer than q_1, contradicting the definition of q_1. Otherwise, let $a \in \Sigma$ be the character immediately following this occurrence of x' in x_1, so that $x'a \in \mathsf{Substr}(x_1)$. Then $\mathsf{rexp}_T(x'a)$ is a suffix of x_1 that is longer than q_1. If $\mathsf{rexp}_T(x'a) \neq x_1$, then this contradicts the definition of q_1. Otherwise, $\mathsf{rexp}_T(x'a) = x_1$, and hence x' has an outgoing edge to x_1 in $\mathsf{CDAWG}(T)$, contradicting that y_1 is the unique in-neighbor of x_1 in $\mathsf{CDAWG}(T)$. □

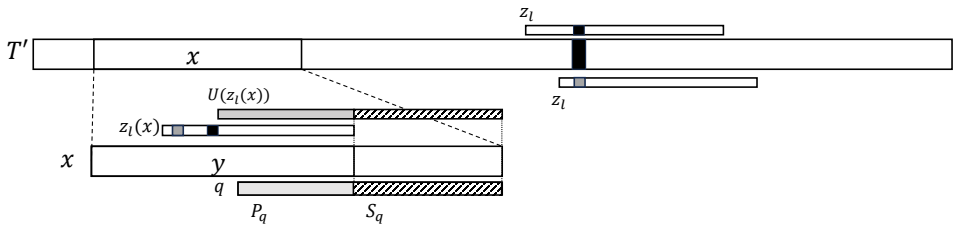


Figure 8. Illustration for the proof of Lemma 19. The black position in T' is the edited position. P_q has a non-crossing occurrence which does not contain nor touch the edited position.

By Lemma 18, there is an injection that maps a new node $z_l(x)$ in $\mathsf{CDAWG}(T')$ to an existing node x in $\mathsf{CDAWG}(T)$, and these nodes have only a single in-edge. Thus, the number of nodes whose in-degree is one does not change in Case 2(c)-1.

Case 2(c)-2: $z_l(x)$ is a suffix of y. Let $\mathsf{V}_{2(\mathrm{c})-2}$ be the set of strings x in Case 2(c)-2, and $\mathsf{V}^{(y)}_{2(\mathrm{c})-2}$ be the set of strings y corresponding to $x \in \mathsf{V}_{2(\mathrm{c})-2}$. If $x \in \mathsf{V}_{2(\mathrm{c})-2}$, $z_l(x)$ is right-maximal and is not left-maximal in T since $z_l(x)$ is a suffix of y and $z_l(x) \notin \mathsf{MR}(T)$. Then, $z_l(x) \in \mathsf{N}_1$. Let $P_q(x)$ denote the overlap of y and q. Since $y, q \in \mathsf{MR}(T)$, $P_q \in \mathsf{MR}(T)$. For $P_q(x)$, we have:

Lemma 19. *For every $x \in \mathsf{V}_{2(\mathrm{c})-2}$, $P_q = P_q(x)$ also has a crossing-occurrence in T'. Furthermore, $U(z_l(x))$ (the existing maximal repeat corresponding to $z_l(x)$) is a suffix of P_q.*

Proof. Suppose P_q has only non-crossing occurrence(s) in T'. See Fig. 8. $z_l(x)$ has a crossing occurrence in T' since $z_l(x) \in \mathsf{N}_1$. Since $z_l(x)$ has a crossing occurrence whereas P_q is assumed to have none and both $U(z_l(x))$ and P_q are suffixes of $z_l(x)$, $|U(z_l(x))| > |P_q|$ (also see Fig. 8). Thus $U(z_l(x))$ is a maximal repeat of T having P_q as a proper suffix. Let S_q be the string such that $q = P_q S_q$. Since $U(z_l(x))S_q$ is a suffix of x longer than q, $U(z_l(x))S_q \equiv^{\mathrm{L}}_T x$ holds. Since $U(z_l(x))$ is also a suffix of y, there exists a path from $U(z_l(x))$ to x which does not contain y in $\mathsf{CDAWG}(T)$. This implies that x has another in-edge from some node other than y in $\mathsf{CDAWG}(T)$ before edit, which is a contradiction. By the above arguments, if $U(z_l(x))$ is longer than P_q, then it also leads to a contradiction. Since $U(z_l(x))$ and P_q are suffixes of y, $U(z_l(x))$ is also a suffix of P_q. □

For the correspondence between y and $z_l(x), P_q$, we have the following lemma:

Lemma 20. *Let $y_1, y_2 \in V^{(y)}_{2(c)-2}$ be distinct. For any $x_1, x_2 \in V_{2(c)-2}$ corresponding to y_1, y_2, respectively, we have $z_l(x_1) \neq z_l(x_2)$ and $P_q(x_1) \neq P_q(x_2)$.*

Proof. Assume $z_l(x_1) = z_l(x_2)$. Since $z_l(x_1)$ occurs both in y_1 and in y_2 as their suffix, there exists a maximal repeat of T in $z_l(x_1)$ or in a suffix of y_1 that is longer than $z_l(x_1)$. This implies that there exists a maximal repeat of T that is longer than $P_q(x_1)$ (similar to Lemma 19), which is a contradiction. Thus, $z_l(x_1) \neq z_l(x_2)$. Assume $P_q(x_1) = P_q(x_2)$. Since this string occurs as a suffix of both y_1 and y_2, the same argument as above implies that there is a maximal repeat of T contained in $P_q(x_1)$ or in a suffix of y_1 that is longer than $P_q(x_1)$, which is a contradiction. □

By Lemma 20, there exists an injective mapping from y to $z_l(x)$ and P_q.

We further show the value of $|\mathsf{V}^{(y)}_{2(\mathrm{c})-2}|$ by using $z_l(x)$. Since $z_l(x)$ is a suffix of $y \in \mathsf{MR}(T)$, $z_l(x) \in \mathsf{N}_1$, which implies $\mathsf{d}_{T'}(z_l(x)) \leq \mathsf{d}_T(U(z_l(x))) + 2$ due to Lemma 5.

Let $\mathsf{V}^{(z)}_{2(\mathrm{c})-2}$ denote the set of $z_l(x)$ where $x \in \mathsf{V}_{2(\mathrm{c})-2}$. Let $\delta_{\mathsf{V}}(z_l(x))$ denote the value such that $\delta_{\mathsf{V}}(z_l(x)) = \max(\mathsf{d}_{T'}(z_l(x)) - \mathsf{d}_T(U(z_l(x))), 0)$, and x_y denote the arbitrarily chosen x corresponding to each $y \in \mathsf{V}^{(y)}_{2(\mathrm{c})-2}$. Also, the following lemma holds:

Lemma 21. $|\mathsf{N}_2| \leq |\mathsf{M}(T)| - \sum_{y \in \mathsf{V}^{(y)}_{2(\mathrm{c})-2}} \delta_{\mathsf{V}}(z_l(x_y))$.

Proof. For each $y \in \mathsf{V}^{(y)}_{2(\mathrm{c})-2}$, let $q = q(x_y)$ and $P_q = P_q(x_y)$. Since $q, y \in Q$ and P_q is their overlap in the occurrence inside x_y, we have that P_q is left-maximal and right-maximal in both T and T'. Hence, $P_q \in \mathsf{Q}$. Also, by Lemma 19, $U(z_l(x_y))$ is a suffix of $P_q(x_y)$, and by Lemma 17 and the definition of Case 2(c)-2, $P_q(x_y)$ is a suffix of $z_l(x_y)$.

Suppose $\delta_{\mathsf{V}}(z_l(x_y)) = 2$. Since $U(z_l(x_y))$ is a suffix of $P_q(x_y)$ and $P_q(x_y)$ is a suffix of $z_l(x_y)$, $\mathsf{d}_{T'}(P_q(x_y)) \geq \mathsf{d}_T(U(z_l(x_y))) + 2 \geq \mathsf{d}_T(P_q(x_y)) + 2$ holds. This implies that $P_q(x_y) \in Q$ has at least two new right-extensions in T'. By Lemma 8, these new right-extensions are injectively mapped to two distinct maximal repeats in $\mathsf{M}(T)$. Similarly, for each $z_l(x_y)$ where $\delta_{\mathsf{V}}(z_l(x_y)) = 1$, $\mathsf{d}_{T'}(P_q(x_y)) \geq \mathsf{d}_T(P_q(x_y)) + 1$ holds, and $P_q(x_y) \in Q$ has at least one new right-extension in T', which is injectively mapped to a maximal repeat in $\mathsf{M}(T)$ by Lemma 8. By Lemma 20, the strings $P_q(x_y)$ for distinct $y \in \mathsf{V}^{(y)}_{2(\mathrm{c})-2}$ are pairwise distinct. Then, at least $\sum_{y \in \mathsf{V}^{(y)}_{2(\mathrm{c})-2}} \delta_{\mathsf{V}}(z_l(x_y))$ new right-extensions of these strings $P_q(x_y) \in \mathsf{Q}$ are injectively mapped to distinct maximal repeats in $\mathsf{M}(T)$. On the other hand, each string $s \in \mathsf{N}_2$ has at least one new right-extension in T', since $s \in \mathsf{M}(T')$ but $s \notin \mathsf{RightM}(T)$. By Lemma 8, these new right-extensions are also injectively mapped to distinct maximal repeats in $\mathsf{M}(T)$. Applying the injection of Lemma 8 to the union of the above new right-extensions of strings in $\mathsf{N}_2 \cup \mathsf{Q}$, we obtain $|\mathsf{N}_2| + \sum_{y \in \mathsf{V}^{(y)}_{2(\mathrm{c})-2}} \delta_{\mathsf{V}}(z_l(x_y)) \leq |\mathsf{M}(T)|$. □

Since $z_l(x) \in \mathsf{N}_1$ and $\delta_{\mathsf{N}}(z_l(x)) = 2 - \max(\mathsf{d}_{T'}(z_l(x)) - \mathsf{d}_T(U(z_l(x))), 0)$, we obtain:

Corollary 22. *For every* $x \in \mathsf{V}_{2(\mathrm{c})-2}$, $\delta_{\mathsf{N}}(z_l(x)) + \delta_{\mathsf{V}}(z_l(x)) = 2$.

By Lemma 20, the chosen strings $z_l(x_y)$ are pairwise distinct elements of $\mathsf{V}^{(z)}_{2(\mathrm{c})-2}$. Hence:

Corollary 23. $|\mathsf{N}_1| \geq |\mathsf{V}^{(z)}_{2(\mathrm{c})-2}| \geq |\mathsf{V}^{(y)}_{2(\mathrm{c})-2}|$ *holds.*

To consider the size of $|\mathsf{V}_{2(\mathrm{c})-2}|$, let $D = \sum_{y \in \mathsf{V}^{(y)}_{2(\mathrm{c})-2}} (\mathsf{d}_T(y))$, and X be the total number of in-edges of $x \in \mathsf{V}_{2(\mathrm{c})-2}$ in $\mathsf{CDAWG}(T)$. See the upper diagram (a) of Fig. 9. Since $x \in \mathsf{V}_{2(\mathrm{c})-2}$ has only one in-edge (y, x) where $y \in \mathsf{V}^{(y)}_{2(\mathrm{c})-2}$ in $\mathsf{CDAWG}(T)$, $|\mathsf{V}_{2(\mathrm{c})-2}| = X \leq D$ holds.

We now leave the analysis of Case 2(c)-2 and combine all cases globally. Recall that $\mathsf{C}(T)$ counts the existing CDAWG nodes whose in-degree changes from one after the edit. Putting Cases 1, 2(a), 2(b), 2(c)-1, and 2(c)-2 together gives the following bound. Since $\mathsf{C}(T) = |\mathsf{V}_1| + |\mathsf{V}_{2(\mathrm{a})}| + |\mathsf{V}_{2(\mathrm{b})}| + |\mathsf{V}_{2(\mathrm{c})-1}(T)| + |\mathsf{V}_{2(\mathrm{c})-2}|$ and at least $|\mathsf{V}_{2(\mathrm{c})-1}(T)|$ new nodes have only one in-edge in Case 2(c)-1, we have the following:

Lemma 24. $\mathrm{v}^{(1)}(T) - \mathrm{v}^{(1)}(T') \leq |\mathsf{V}_1| + |\mathsf{V}_{2(\mathrm{a})}| + |\mathsf{V}_{2(\mathrm{b})}| + X$ *holds.*

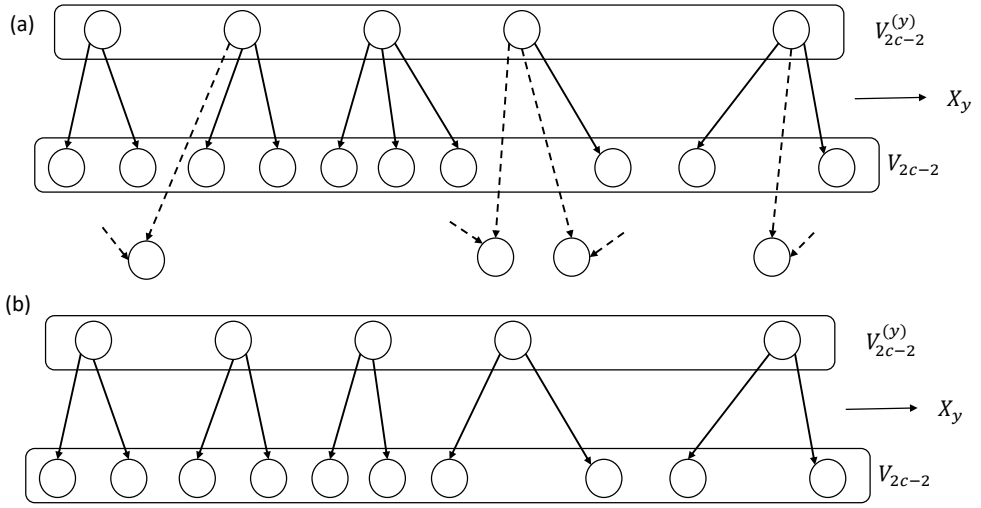


Figure 9. (a) Illustration for the definitions of $\mathsf{V}^{(y)}_{2(\mathrm{c})-2}$, X and D. D counts all of the out-edges from $y \in \mathsf{V}^{(y)}_{2(\mathrm{c})-2}$, although X only counts the edges going to $x \in \mathsf{V}_{2(\mathrm{c})-2}$ (only solid-line edges are counted). (b) The situation of (a) in which $\mathsf{g}(T') - \mathsf{g}(T)$ is maximized.

Finally, we combine the results of Theorem 11, Lemmas 12, 21 and 24, and Corollaries 22 and 23. Then, we have:

$$\begin{aligned}
&\mathsf{g}(T') - \mathsf{g}(T) \\
&= (\mathsf{e}(T') - \mathsf{e}(T)) + (\mathrm{v}^{(1)}(T) - \mathrm{v}^{(1)}(T')) \\
&\leq \mathsf{rem_e} + 5|\mathsf{MR}(T)| - \textstyle\sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}} \delta_{\mathsf{N}}(x) + |\mathsf{N}_2| + 3 + |\mathsf{V}_1| + |\mathsf{V}_{2(\mathrm{a})}| + |\mathsf{V}_{2(\mathrm{b})}| + X \\
&\leq \mathsf{e}(T) + 6|\mathsf{MR}(T)| - \textstyle\sum_{x \in \mathsf{N}_1 \cup \mathsf{N}_{3\mathrm{A}}} \delta_{\mathsf{N}}(x) - \sum_{y \in \mathsf{V}^{(y)}_{2(\mathrm{c})-2}} \delta_{\mathsf{V}}(z_l(x_y)) + X + 4 \\
&\leq \mathsf{e}(T) + 6|\mathsf{MR}(T)| - \textstyle\sum_{y \in \mathsf{V}^{(y)}_{2(\mathrm{c})-2}} (\delta_{\mathsf{N}}(z_l(x_y)) + \delta_{\mathsf{V}}(z_l(x_y))) + X + 4 \\
&\leq \mathsf{e}(T) + 6|\mathsf{MR}(T)| + X - 2|\mathsf{V}^{(y)}_{2(\mathrm{c})-2}| + 4.
\end{aligned}$$

We call this Inequality (α).

Now we consider the value of X. Let $\hat{D} = \sum_{y \in \mathsf{MR}(T) \setminus \mathsf{V}^{(y)}_{2(\mathrm{c})-2}} (\mathsf{d}_T(y))$. For D, $|\mathsf{MR}(T)|$ and $\mathsf{e}(T)$, the following holds:

Lemma 25. *Let* $\omega = X - 2|\mathsf{V}^{(y)}_{2(\mathrm{c})-2}|$, *then* $|\mathsf{MR}(T)| \leq \min(\mathsf{e}(T), \mathsf{e}(T) - \omega)/2$.

Proof. If $\omega \leq 0$, then $\mathsf{e}(T) \geq 2|\mathsf{MR}(T)|$ due to Fact 1. If $\omega > 0$, then $D \geq X = 2|\mathsf{V}^{(y)}_{2(\mathrm{c})-2}| + \omega$ holds. Since $D + \hat{D} = \mathsf{e}(T)$ and $\hat{D} \geq 2|\mathsf{MR}(T) \setminus \mathsf{V}^{(y)}_{2(\mathrm{c})-2}|$, $\mathsf{e}(T) \geq 2|\mathsf{MR}(T)| + \omega$. Then $|\mathsf{MR}(T)| \leq (\mathsf{e}(T) - \omega)/2$. □

By substituting the value of Lemma 25 to Inequality (α) above, we get $\mathsf{g}(T') - \mathsf{g}(T) \leq 4\mathsf{e}(T) + \omega + 4$ if $\omega \leq 0$, or $\mathsf{g}(T') - \mathsf{g}(T) \leq 4\mathsf{e}(T) - 2\omega + 4$ if $\omega > 0$. Therefore, $\mathsf{g}(T') - \mathsf{g}(T) \leq 4\mathsf{e}(T) + 4$. Fig. 9(b) illustrates the situation suggested by the above bound. As a result, we have the following theorem:

Theorem 26. *The additive sensitivity of the CDAWG-grammar is at most* $4\mathsf{e} + 4$.

Since $\mathsf{g}(T) \geq \mathsf{e}(T)/2$, Theorem 26 immediately leads to the following:

Corollary 27. *The additive sensitivity of the CDAWG-grammar is at most* $8\mathsf{g} + 4$.

References

1. T. Akagi, M. Funakoshi, and S. Inenaga: *Sensitivity of string compressors and repetitiveness measures.* Information and Computation, 291 2023, p. 104999.
2. P. Baturo, M. Piatkowski, and W. Rytter: *Usefulness of directed acyclic subword graphs in problems related to Standard Sturmian Words.* Int. J. Found. Comput. Sci., 20(6) 2009, pp. 1005–1023.
3. D. Belazzougui and F. Cunial: *Fast label extraction in the CDAWG*, in SPIRE 2017, 2017, pp. 161–175.
4. D. Belazzougui and F. Cunial: *Representing the Suffix Tree with the CDAWG*, in 28th Annual Symposium on Combinatorial Pattern Matching (CPM 2017), vol. 78, 2017, pp. 7:1–7:13.
5. A. Blumer, J. Blumer, D. Haussler, R. McConnell, and A. Ehrenfeucht: *Complete inverted files for efficient text retrieval and analysis.* Journal of the ACM, 34(3) 1987, pp. 578–595.
6. A. Cleary and J. Dood: *Constructing the CDAWG CFG using LCP-intervals*, in 2023 Data Compression Conference (DCC), 2023, pp. 178–187.
7. M. Crochemore and R. Vérin: *On compact directed acyclic word graphs*, in Structures in Logic and Computer Science: A Selection of Essays in Honor of A. Ehrenfeucht, Springer, 1997, pp. 192–211.
8. H. Fujimaru, Y. Nakashima, and S. Inenaga: *Tight bounds for the sensitivity of CDAWGs with left-end edits.* Acta Informatica, 62(12) 2025.
9. R. Hamai, H. Fujimaru, and S. Inenaga: *Constant sensitivity on the CDAWGs*, in Proc. the 37th Annual Symposium on Combinatorial Pattern Matching (CPM 2026), 2026, To appear.
10. S. Inenaga, H. Hoshino, A. Shinohara, M. Takeda, S. Arikawa, G. Mauri, and G. Pavesi: *On-line construction of compact directed acyclic word graphs.* Discrete Applied Mathematics, 146(2) 2005, pp. 156–179.
11. J. Radoszewski and W. Rytter: *On the structure of compacted subword graphs of Thue-Morse words and their applications.* J. Discrete Algorithms, 11 2012, pp. 15–24.
12. W. Rytter: *The structure of subword graphs and suffix trees of Fibonacci words.* Theor. Comput. Sci., 363(2) 2006, pp. 211–223.
13. T. Takagi, K. Goto, Y. Fujishige, S. Inenaga, and H. Arimura: *Linear-size CDAWG: New repetition-aware indexing and grammar compression*, in SPIRE 2017, 2017, pp. 304–316.
14. M. Takeda, T. Matsumoto, T. Fukuda, and I. Nanri: *Discovering characteristic expressions from literary works: a new text analysis method beyond n-gram statistics and KWIC*, in Discovery Science 2000, 2000, pp. 112–126.
15. P. Weiner: *Linear pattern matching algorithms*, in Proceedings of the 14th Annual Symposium on Switching and Automata Theory, 1973, pp. 1–11.