Small Space Encoding and Recognition of k-Palindromic Prefixes

Gabriel Bathie
DIENS, École normale supérieure de Paris, PSL Research University, France
LaBRI, Université de Bordeaux, France

Jonas Ellert
DIENS, École normale supérieure de Paris, PSL Research University, France

Tatiana Starikovskaya
DIENS, École normale supérieure de Paris, PSL Research University, France

Abstract

Palindromes are non-empty strings that read the same forward and backward. We study the problem of recognizing so-called *k-palindromic* strings, which can be represented as the concatenation of exactly k palindromes. [Rubinchik and Shur, MFCS 2020] showed that the problem is solvable in linear space and time. We present a read-only algorithm that recognizes all k-palindromic prefixes of a string T of length n in $\mathcal{O}(n \cdot 6^{k^2} \cdot \log^k n)$ time and $\mathcal{O}(6^{k^2} \cdot \log^k n)$ space. As a corollary, we also obtain a read-only algorithm for computing the palindromic length of T, i.e., the smallest k such that T is k-palindromic, in $\mathcal{O}(n \cdot 6^{k^2} \cdot \log^{\lceil k/2 \rceil} n)$ time and $\mathcal{O}(6^{k^2} \cdot \log^{\lceil k/2 \rceil} n)$ space.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases palindromic length, read-only algorithms, palindromes

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2025.9

Related Version *Full Version*: `https://doi.org/10.48550/arXiv.2410.03309` [5]

Funding *Gabriel Bathie, Jonas Ellert, and Tatiana Starikovskaya:* Partially funded by grant ANR-20-CE48-0001 from the French National Research Agency.

1 Introduction

A *palindrome* is a non-empty string that equals its reversed copy, i.e., a string that reads the same both forward and backward. Throughout, we denote the language of palindromes by PAL. Natural variants of PAL include the language of even-length palindromes PAL_{ev} and the language of palindromes of length greater than one $\mathsf{PAL}_{>1}$. Recognising $\mathsf{PAL}^*_{\text{ev}}$ (often referred to as "palstar"), $\mathsf{PAL}^*_{>1}$, and $\mathsf{PAL}^k = \{P_1 P_2 \cdots P_k : P_i \in \mathsf{PAL}, 1 \leq i \leq k\}$ is a classical problem of formal language theory, introduced by Knuth, Morris, and Pratt [27].[1] In this problem, one is given an input string and must decide whether it is in the language.

Languages $\mathsf{PAL}^*_{\text{ev}}$, $\mathsf{PAL}^*_{>1}$, and PAL^k are context-free, and Valiant's parser from 1975 recognizes them in $\mathcal{O}(n^\omega)$ time, where n is the length of the input and ω is the matrix multiplication exponent. Only in 2018, Abboud, Backurs, and Vassilevska Williams showed that Valiant's parser is optimal if the current clique algorithms are optimal [1], meaning that for general context-free languages, there is little hope of achieving a faster recognition algorithm. The origins of the study of derivatives of the PAL languages are in fact in line with the result of [1]: At one time, it was popularly believed that $\mathsf{PAL}^*_{\text{ev}}$ cannot be recognised in linear time, and it was considered as a candidate for a "hard" context-free language (see

[1] * is a Kleene star.

36th International Symposium on Algorithms and Computation (ISAAC 2025).
Editors: Ho-Lin Chen, Wing-Kai Hon, and Meng-Tsung Tsai; Article No. 9; pp. 9:1–9:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

[27, Section 6]). However, [27] refuted this hypothesis by showing an $\mathcal{O}(n)$-time recognition algorithm for $\mathsf{PAL}^*_{\mathrm{ev}}$. Manacher [32] found another way to recognize $\mathsf{PAL}^*_{\mathrm{ev}}$ in linear time, and Galil [16] derived a real-time recognition algorithm (see also Slisenko [35]). Later, Galil and Seiferas [17] showed a linear-time recognition algorithm for $\mathsf{PAL}^*_{>1}$.

Recognition of PAL^k appeared to be a much more intricate problem. Galil and Seiferas [17] succeeded to design linear-time recognition algorithms for the cases $k = 1, 2, 3, 4$, but the general question remained open for almost 40 years. Only in 2015, Kosolobov, Rubinchik, and Shur [29] showed an $\mathcal{O}(nk)$-time recognition algorithm for PAL^k for all $k \in \mathbb{N}^+$, which was finally improved to optimal $\mathcal{O}(n)$ time by Rubinchik and Shur in 2020 [34]. A related question is that of computing the *palindromic length* of a string T, which is defined to be the smallest integer k such that $T \in \mathsf{PAL}^k$. The first $\mathcal{O}(n \log n)$-time algorithms for computing the palindromic length were presented in [12, 25, 33]. Finally, Borozdin et al. [10] showed an optimal $\mathcal{O}(n)$-time algorithm for this problem.

Our contributions. In this work, we turn our attention to the *space complexity* of recognising PAL^k and computing the palindromic length. We start by presenting a characterization of prefixes of a given string that belong to PAL^k. For $k = 1$, we refer to these prefixes as *prefix-palindromes*, and otherwise as *k-palindromic prefixes*. A crucial component of the linear time algorithm by Borozdin et al. [10] is the following property: the prefix-palindromes of a length-n string can be expressed as $\mathcal{O}(\log n)$ sets of form $\{XQ^a : a \in \{1, \ldots, u\}\}$, where u is an integer. In order to encode k-palindromic prefixes, we introduce a notion of *affine prefix sets* of order k. Intuitively, such a set consists of prefixes of the form $XQ_1^{a_1}Q_2^{a_2}\cdots Q_k^{a_k}$ with $\forall i \in [1, k] : a_i \in \{1, \ldots, u_i\}$, where u_i are integers. That is, rather than a single repeating substring Q, we allow multiple different substrings Q_i of different lengths. An affine prefix set of order k can then be encoded in $\mathcal{O}(k)$ space. By carefully analyzing the rich structure of periodic substrings induced by k-palindromic prefixes, we show that they can be expressed by a small number of affine prefix sets.

▶ **Theorem 1.1.** *Let $0 < \epsilon < 1$ be constant, $T[1..n]$ a string, and $k \in \mathbb{N}^+$. The set of prefixes of T that belong to PAL^k is the union of $\mathcal{O}(6^{k^2/(2-\epsilon)} \cdot \log^k n)$ affine prefix sets of order $\leq k$.*

The remainder of the paper focuses on the main ideas behind Theorem 1.1. In the full version [5], apart from explaining all details, we show a lower bound for the size of the representation (Theorem 1.2), and provide read-only algorithms for computing affine prefix sets and palindromic length (Theorems 1.3 and 1.4). The lower bound shows that our representation is within a $\log n$ factor of being asymptotically optimal for constant k. It is obtained by constructing a large family of strings uniquely identifiable by their palindromic prefixes.

▶ **Theorem 1.2.** *Let $T[1..n]$ be a string and let $k \in \mathbb{N}^+$. Encoding the lengths of the prefixes of T that belong to PAL^i, for each $i \in [1, k]$, requires $\Omega(k^{-k} \cdot (\log_3 n)^k)$ bits of space.*

The read-only algorithms directly implement the techniques used to show Theorem 1.1.

▶ **Theorem 1.3.** *Let $0 < \epsilon < 1$ be constant. Given a string $T[1..n]$ and $k \in \mathbb{N}^+$, there is a read-only algorithm that returns a compressed representation of all prefixes of T that belong to PAL^i, for each $i \in [1, k]$, in $\mathcal{O}(n \cdot 6^{k^2/(2-\epsilon)} \cdot \log^k n)$ time and $\mathcal{O}(6^{k^2/(2-\epsilon)} \cdot \log^k n)$ space.*

▶ **Theorem 1.4.** *Given a string $T[1..n]$, there is a read-only algorithm that computes the palindromic length k of T in $\mathcal{O}(n \cdot 6^{k^2} \cdot \log^{\lceil k/2 \rceil} n)$ time and $\mathcal{O}(6^{k^2} \cdot \log^{\lceil k/2 \rceil} n)$ space.*

In particular, for $k = \mathcal{O}(\log\log n)$, the algorithm uses $n \log^{\mathcal{O}(k)} n$ time and $\log^{\mathcal{O}(k)} n$ space, and for $k = o(\sqrt{\log n})$, it uses $n^{1+o(1)}$ time and sublinear $n^{o(1)}$ space. In the regime of small palindromic length, this is an improvement over all previously-known algorithms [10, 34], which require $\Omega(n)$ space. It remains an intriguing open problem to achieve *both* linear time and sublinear space. On the other hand, Theorem 1.2 does not imply a lower bound for the algorithms of Theorems 1.3 and 1.4 because they have access to the input. This said, proving an $\Omega(\log^{f(k)} n)$ space lower bound for such algorithms appears out of reach with current techniques. The only lower bound method for read-only string processing the authors are aware of relies on deterministic branching programs [9], and shows that any sublinear-space algorithm for longest common substring requires slightly superlinear time [28].

Related work. The problem of computing palindromes in small space has received significant attention: the longest palindromic substring [8, 24], and all approximate prefix-palindromes [3, 6] have been studied. More broadly, small-space recognition of formal languages has been explored for regular [19, 20, 21, 22, 11], Dyck [26, 30, 31], visibly pushdown [2, 14, 18, 7], context-free [23], and $\mathsf{DLIN}/\mathsf{LL}(k)$ languages [4].

2 Preliminaries

Series, strings, and substrings. For $i, j \in \mathbb{Z}$, we write $[i, j] = [i, j+1) = (i-1, j] = (i-1, j+1)$ to denote $\{h \in \mathbb{Z} \mid i \le h \le j\}$. A series $a_1, b_1, c_1, a_2, b_2, c_2, \ldots, a_t, b_t, c_t$ is denoted by $(a_i, b_i, c_i)_{i=1}^{t}$. The empty series is denoted by ε. We use the dot-product to denote the concatenation of two series, e.g., for $t \ge 3$ one can represent $(a_i, b_i, c_i)_{i=1}^{t} = (a_i, b_i, c_i)_{i=1}^{t-3} \cdot (a_i, b_i, c_i)_{i=t-2}^{t}$. We may omit the subscript and superscript for series of length one, e.g., $(a_1, b_1, c_1) = (a_i, b_i, c_i)_{i=1}^{1}$.

A *string* T of length $|T| = n$ is a sequence of n *symbols* from a set Σ, which we call the *alphabet*. The input string is also called *the text*. We denote the set of all length-n strings by Σ^n, and we set $\Sigma^{\le n} = \bigcup_{m=0}^{n} \Sigma^m$ as well as $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$. The empty string is denoted by ε. For $i, j \in [1, n]$, the i-th symbol in T is denoted by $T[i]$. The *substring* $T[i..j] = T[i..j+1) = T(i-1..j] = T(i-1..j+1)$ is the empty string ε if $j < i$, and the string $T[i]T[i+1]\cdots T[j]$ otherwise. We may call a substring $T[i..j]$ a *fragment* of T to emphasize that we mean the specific occurrence of $T[i..j]$ that starts at a position i. For example, in the string $T = \texttt{abcabc}$, the substrings $T[1..3]$ and $T[4..6]$ are identical, but $T[1..3]$ and $T[4..6]$ are distinct fragments. A string S is a *prefix* of T if there is $i \in [1, n]$ such that $S = T[1..i]$, in which case we may simply write $T[..i]$. Similarly, S is a *suffix* of T if there is $i \in [1, n]$ such that $S = T[i..n]$, in which case we may simply write $T[i..]$. We extend this notion to the empty suffix $T[n+1..n] = T[n+1..]$ and the empty prefix $T[1..0] = T[..0]$. A substring (hence also a suffix or prefix) of T is *proper* if it is shorter than T. When introducing a string S, we may simply say that $S[1..m]$ is a string rather than saying that S is a string of length m. The concatenation of two strings $S[1..m]$ and $T[1..n]$ is the string $S[1]S[2]\cdots S[m]T[1]T[2]\cdots T[n]$, denoted by either $S \cdot T$ or simply ST. For non-negative integer a, we write T^a to denote the length-(an) string obtained by concatenating a copies of T. We extend this idea to non-negative rational exponents $\alpha \in \mathbb{Q}$, for which we write T^α to denote $T^{\lfloor \alpha \rfloor} \cdot T[1..(\alpha n \bmod n)]$. We only use this notation if $\alpha n \in \mathbb{N}$.

Palindromes and periodicities. For a string $T[1..n]$, we write $\mathsf{rev}(T)$ to denote its reverse, i.e., $\mathsf{rev}(T) = T[n]T[n-1]\cdots T[1]$. We then say that T is a palindrome if and only if T is non-empty and $T = \mathsf{rev}(T)$. The set of all palindromes is denoted by PAL. For a positive

integer k, the set PAL^k contains all the strings that can be written as the concatenation of exactly k palindromes. We refer to such strings as *k-palindromic*. If $k = 1$, and a string is a one-palindromic prefix of another string, we also refer to it as *prefix-palindrome*.

We define the *forward cyclic rotation* $\mathsf{rot}(T) = T[2..n]T[1]$. More generally, a cyclic rotation $\mathsf{rot}^s(T)$ with *shift* $s \in \mathbb{Z}$ is obtained by iterating rot (if s is positive) or the inverse operation rot^{-1} (if s is negative) exactly $|s|$ times. A non-empty string $T[1..n]$ is *primitive* if it is distinct from its non-trivial rotations, i.e., if $T = \mathsf{rot}^s(T)$ holds only when n divides s. If a string X can be represented as Y^a for some primitive string Y and integer a, then Y is called *the primitive root* of X.

A string $T[1..n]$ has *period* $p \in \mathbb{N}^+$ if $\forall i \in [1, n-p] : T[i] = T[i+p]$, or equivalently if $T[1..n-p] = T(p..n]$. The string $T[1..n-p] = T(p..n]$ is a *border* of T. If T has period $p \leq n/2$, then we say that T is p-periodic. If T has period $p \leq n$, then it can be written as $T = P^{\lfloor n/p \rfloor} P[1..n \bmod p]$, where $P = T[1..p]$. We may alternatively use a rational exponent and write $T = P^{n/p}$. Below, we provide some simple auxiliary lemmas regarding periodic strings and palindromes (with proofs in the full version).

▶ **Fact 2.1** (Periodicity Lemma [13]). *If p and q are distinct periods of a string of length at least $p + q - \gcd(p, q)$, then $\gcd(p, q)$ is a period of the string.*

▶ **Corollary 2.2** (Folklore). *For a primitive string Q, the minimal period of Q^2 is $|Q|$.*

▶ **Lemma 2.3.** *Assume that a palindrome P has a q-periodic prefix of length $m \geq 3q/2$. If $|P| \leq 2m - q$, then P is q-periodic.*

Model of computation. We assume the word-RAM model of computation [15], using words of size $\Theta(\log n)$ when processing an input string of length n. The presented algorithms are deterministic and read-only, i.e., they cannot write to the memory occupied by the input. Space complexities are stated in number of words, ignoring the space occupied by the input.

3 Combinatorial Properties of Affine Prefix Sets

In this section, we study the combinatorial structure of k-palindromic prefixes of T. We start with the definition of *affine sets*, which we will use as a scaffolding for our analysis.

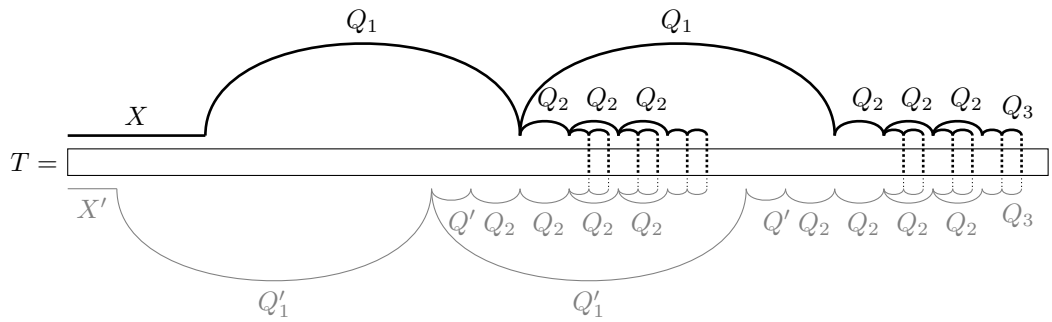
▶ **Definition 3.1** (Affine sets). *A set of strings $\mathcal{A}$ is* affine *if there exist $t \in \mathbb{N}_0$, a string X, primitive strings $Q_1, \ldots, Q_t$, and positive integers $\ell_1, \ldots, \ell_t$ and $u_1, \ldots, u_t$ such that*

$$\forall i \in [1, t] : \ell_i \leq u_i \qquad \text{and} \qquad \mathcal{A} = \{XQ_1^{a_1} \cdots Q_t^{a_t} \mid \forall r \in [1, t] : a_r \in [\ell_r, u_r]\}.$$

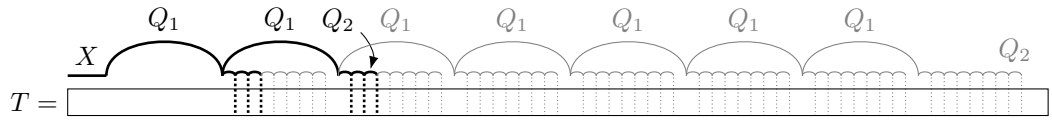
The tuple $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t \rangle$ is a representation *of $\mathcal{A}$, and t is the* order *of the representation. The order of $\mathcal{A}$ is the minimal order achieved by any of its representations. We call $\{Q_i\}$* the components *of a representation, and ℓ_i (resp., u_i) the exponent lower (resp., upper) bounds.*

A representation *generates* (the strings of) the corresponding affine string set. If $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t \rangle$ generates $\mathcal{A}$ and $\langle X', (Q'_i, \ell'_i, u'_i)_{i=1}^{t'} \rangle$ generates $\mathcal{B}$, then their concatenation is defined as $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t \cdot (Y, a, a) \cdot (Q'_i, \ell'_i, u'_i)_{i=1}^{t'} \rangle$, where Y is a primitive string and a is a positive integer such that $Y^a = X'$ (i.e., Y is the primitive root of X'). The concatenation generates $\mathcal{A} \cdot \mathcal{B} = \{A \cdot B : A \in \mathcal{A} \wedge B \in \mathcal{B}\}$. (If $X' = \varepsilon$, then the concatenation is $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t \cdot (Q'_i, \ell'_i, u'_i)_{i=1}^{t'} \rangle$.)

In what follows, we consider *affine prefix sets*, i.e., affine sets that contain only prefixes of the given input string T. We will show that a small number of affine prefix sets suffices to represent the k-palindromic prefixes of T. The case where $k = 1$, i.e., the structure of



(a) An affine prefix set $\mathcal{A}$ of a string T with representation $\langle X, (Q_1, ,) \cdot (Q_2, ,) \cdot (Q_3, ,)\rangle$ (drawn above T). This representation is irreducible. The set $\mathcal{A}$ contains all the prefixes of T that end at positions drawn in dotted lines. In this example, the set $\mathcal{A}$ has the alternative representation $\langle X', (Q'_1, 1, 2) \cdot (Q', 1, 1) \cdot (Q_2, 2, 4) \cdot (Q_3, 1, 2)\rangle$. This representation is reducible because Q' has the same exponent upper and lower bound, and because Q_2 has an exponent lower bound larger than 1.



(b) An affine prefix set $\mathcal{A}$ of a string T with representation $\langle X, (Q_1, 1, 2) \cdot (Q_2, 1, 3)\rangle$ (drawn in black). If this representation is strongly affine, then its expansion $\langle X, (Q_1, 1, 7) \cdot (Q_2, 1, 8)\rangle$ is also a representation of an affine prefix set of T (drawn in gray).

Figure 1 Affine prefix sets.

prefix-palindromes, is well-understood: there are $\mathcal{O}(\log n)$ groups of such palindromes, where each group can be expressed as an arithmetic progression and a corresponding periodic prefix of T (see [10, Lemma 5]). Below, we restate this result in the framework of affine prefix sets.

▶ **Lemma 3.2.** *The prefix-palindromes of a string $T[1..n]$ can be partitioned into $\mathcal{O}(\log n)$ affine sets of order at most 1. Each set of order 1 has representation $\langle U(VU)^\ell, (VU, 1, u)\rangle$ for some $U \in \mathsf{PAL} \cup \{\varepsilon\}$, $V \in \mathsf{PAL}$ and integers $\ell \geq 1$ and $u > 1$.*

3.1 Reducing affine prefix sets

A single affine set may have multiple equivalent representations. For example, the affine set $S = \{\mathtt{caba}, \mathtt{cababa}, \mathtt{cababababa}\}$ is represented by $\langle \mathtt{c}, (\mathtt{ab}, 1, 3)$ and $(\mathtt{a}, 1, 1)\rangle$, $\langle \mathtt{ca}, (\mathtt{ba}, 1, 3)\rangle$. Arguably, the latter representation is preferable, as it has a lower order and can thus be encoded more efficiently. Hence we propose a way of potentially decreasing the order of a representation by *reducing* it.

▶ **Definition 3.3** (Irreducible representation). *A representation $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t\rangle$ of an affine string set is* irreducible *if and only if $\forall r \in [1, t] : 1 = \ell_r < u_r$ and $\forall r \in [1, t) : |Q_r| > |Q_{r+1}|$.*

From now on, we say that Q_r with $r \in [1, t]$ is *fixed* if $\ell_r = u_r$, and *flexible* otherwise. If there is some $r \in [1, t)$ such that $|Q_r| \leq |Q_{r+1}|$, then we say that there is an *inversion* between Q_r and Q_{r+1}. Thus, a representation is irreducible if and only if all components are flexible and have unit lower bounds, and there are no inversions. As per this definition, $\langle \mathtt{ca}, (\mathtt{ba}, 1, 3)\rangle$ is the only irreducible representation of S from the previous example. (See also Figure 1a.)

Properties of flexible components. Now we show how to make an arbitrary representation irreducible, possibly decreasing (but never increasing) its order. The reduction exploits the structure of periodic substrings induced by flexible components.

▶ **Lemma 3.4.** *Let $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t \rangle$ be a representation of an affine prefix set, and consider any $r \in [1, t)$ such that Q_r is flexible. Then $|Q_r|$ is a period of every string $Q_r^{a_r} Q_{r+1}^{a_{r+1}} \cdots Q_t^{a_t}$ that satisfies $a_r \in \mathbb{N}_0$ and $\forall j \in (r, t] : a_j \in [\ell_j, u_j]$.*

Proof. Let $P = Q_1^{\ell_1} Q_2^{\ell_2} \cdots Q_r^{\ell_r}$ and $S = Q_{r+1}^{a_{r+1}} Q_{r+2}^{a_{r+2}} \cdots Q_t^{a_t}$. By the definition of an affine prefix set, XPS is a prefix of the underlying string T. Since Q_r is flexible, it holds $\ell_r < u_r$, and thus XPQ_rS is also a prefix of T. If both XPS and XPQ_rS are prefixes of T, then S is a prefix of Q_rS. Hence Q_rS and S have periods $|Q_r|$. Consequently, $Q_r^{a_r} S$, for all $a_r \in \mathbb{N}_0$, also has period $|Q_r|$. ◀

If two adjacent components Q_r and Q_{r+1} are flexible, then the lemma allows us to obtain the following lower bound on the length of Q_r.

▶ **Lemma 3.5.** *Let $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t \rangle$ be a representation of an affine prefix set, and let $r \in [1, t)$. If both Q_r and Q_{r+1} are flexible, then either $Q_r = Q_{r+1}$ or*

$$|Q_r| > |Q_{r+1}^{u_{r+1}-1}| + \left(\sum_{j=r+2}^{t} |Q_j^{u_j}| \right) + \gcd(|Q_r|, |Q_{r+1}|).$$

Proof. For flexible Q_r and Q_{r+1}, let $q_r = |Q_r|$, $q_{r+1} = |Q_{r+1}|$, and $p = \gcd(q_r, q_{r+1})$. Let $Q'_{r+1} = Q_{r+1}^{u_{r+1}} Q_{r+2}^{u_{r+2}} \cdots Q_t^{u_t}$. By Lemma 3.4, both q_r and q_{r+1} are periods of Q'_{r+1}, and q_r is a period of $Q_r Q'_{r+1}$. Since q_r is a period of $Q_r Q'_{r+1}$, it is also a period of $Q_r Q_{r+1}$. Hence $Q_r = Q_{r+1}$ if and only if $q_r = q_{r+1}$. For the sake of contradiction, assume that the lemma does not hold, i.e., $q_r \neq q_{r+1}$ and $q_r \leq |Q'_{r+1}| - q_{r+1} + p$. We make two observations.

First, Q'_{r+1} is of length $|Q'_{r+1}| \geq q_r + q_{r+1} - p$, and it has distinct periods q_r and q_{r+1}. The periodicity lemma (Fact 2.1) implies that p is a period of Q'_{r+1}, and thus also a period of its prefix Q_{r+1}. If $p < q_{r+1}$, then $Q_{r+1} = Q_{r+1}[1..p]^{q_{r+1}/p}$, which contradicts the primitivity of Q_{r+1}. Second, Q'_{r+1} is of length $|Q'_{r+1}| \geq q_r + q_{r+1} - p \geq q_r$. Since q_r is a period of $Q_r Q'_{r+1}$, we know that Q_r is a prefix of Q'_{r+1}. Hence p is also a period of Q_r. If $p < q_r$, then $Q_r = Q_r[1..p]^{q_r/p}$, which contradicts the primitivity of Q_r.

We have shown that $\gcd(q_r, q_{r+1}) \geq \max(q_r, q_{r+1})$. This is only possible if $\gcd(q_r, q_{r+1}) = q_r = q_{r+1}$, which contradicts the assumption that $q_r \neq q_{r+1}$. Therefore, the lemma holds. ◀

▶ **Lemma 3.6.** *Let $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t \rangle$ be an irreducible representation of an affine prefix set $\mathcal{A}$ of a string of length n. Then it holds $|\mathcal{A}| = \prod_{i=1}^t u_i$ and $t \leq \log_2 n$.*

Proof. Let $E = \{(a_i)_{i=1}^t \mid \forall i \in [1, t] : a_i \in [1, u_i]\}$ of cardinality $|E| = \prod_{i=1}^t u_i$ be the set of exponent configurations admitted by the representation (where $[1, u_i] = [\ell_i, u_i]$ because the representation is irreducible). Then $\mathcal{A} = \{XQ_1^{a_1} Q_2^{a_2} \cdots Q_t^{a_t} \mid (a_i)_{i=1}^t \in E\}$. In order to show $|\mathcal{A}| = |E|$, it suffices to show that no two elements in E generate the same string.

For the sake of contradiction, assume that there are distinct sequences $(a_i)_{i=1}^t, (b_i)_{i=1}^t \in E$ that generate the same string $S = XQ_1^{a_1} Q_2^{a_2} \cdots Q_t^{a_t} = XQ_1^{b_1} Q_2^{b_2} \cdots Q_t^{b_t}$. Let $r \in [1, t]$ be the minimal index such that $a_r \neq b_r$, and assume w.l.o.g. that $a_r > b_r$. Then S has the prefix $XQ_1^{a_1} \cdots Q_{r-1}^{a_{r-1}} Q_r^{b_r} = XQ_1^{b_1} \cdots Q_{r-1}^{b_{r-1}} Q_r^{b_r}$, and we can factorize the corresponding suffix in two different ways as $Q_r^{a_r - b_r} Q_{r+1}^{a_{r+1}} \cdots Q_t^{a_t} = Q_{r+1}^{b_{r+1}} \cdots Q_t^{b_t}$. However, the two factorizations have different lengths $|Q_r^{a_r - b_r} Q_{r+1}^{a_{r+1}} \cdots Q_t^{a_t}| > |Q_r Q_{r+1}| > |Q_{r+1}^{u_{r+1}} \cdots Q_t^{u_t}| \geq |Q_{r+1}^{b_{r+1}} \cdots Q_t^{b_t}|$, where the second inequality is due to Lemma 3.5. Because of this contradiction, there cannot be distinct sequences $(a_i)_{i=1}^t, (b_i)_{i=1}^t \in E$ that define the same string.

Finally, it holds $\forall i \in [1,t] : u_i \geq 2$ for any irreducible representation, and thus $|\mathcal{A}| = \prod_{i=1}^{t} u_i \geq 2^t$. Since trivially $|\mathcal{A}| \leq n$, it follows $2^t \leq n$ or equivalently $t \leq \log_2 n$. ◀

Transforming representations. Now we use the properties of flexible components to transform an arbitrary representation into an irreducible one. We use the following operations.

▶ **Lemma 3.7.** *Let $\rho = \langle X, (Q_i, \ell_i, u_i)_{i=1}^{t} \rangle$ be a representation of an affine prefix set $\mathcal{A}$.*

1. *If there is $r \in [1,t)$ such that Q_r is flexible and Q_{r+1} is fixed, then let $y = |Q_{r+1}^{\ell_{r+1}}| \bmod |Q_r|$. $\mathcal{A}$ has representation*

$$\mathsf{switch}_r(\rho) = \langle X, (Q_i, \ell_i, u_i)_{i=1}^{r-1} \cdot (Q_{r+1}, \ell_{r+1}, u_{r+1}) \cdot (\mathsf{rot}^y(Q_r), \ell_r, u_r) \cdot (Q_i, \ell_i, u_i)_{i=r+2}^{t} \rangle.$$

2. *If there is $r \in [1,t)$ such that both Q_r and Q_{r+1} are flexible and $|Q_r| \leq |Q_{r+1}|$, then $Q_r = Q_{r+1}$ and $\mathcal{A}$ has representation*

$$\mathsf{merge}_r(\rho) = \langle X, (Q_i, \ell_i, u_i)_{i=1}^{r-1} \cdot (Q_r, \ell_r + \ell_{r+1}, u_r + u_{r+1}) \cdot (Q_i, \ell_i, u_i)_{i=r+2}^{t} \rangle.$$

3. *If there is $r \in [1,t]$ such that $\ell_r > 1$, then $\mathcal{A}$ has representation*

$$\mathsf{split}_r(\rho) = \langle X, (Q_i, \ell_i, u_i)_{i=1}^{r-1} \cdot (Q_r, \ell_r - 1, \ell_r - 1) \cdot (Q_r, 1, u_r - \ell_r + 1) \cdot (Q_i, \ell_i, u_i)_{i=r+1}^{t} \rangle.$$

4. *If Q_1 is fixed, then $\mathcal{A}$ has representation $\mathsf{truncate}(\rho) = \langle X Q_1^{\ell_1}, (Q_{i+1}, \ell_{i+1}, u_{i+1})_{i=1}^{t-1} \rangle$.*

Proof. Statements (3) and (4) are trivial. For (2), if $|Q_r| \leq |Q_{r+1}|$ and both Q_r and Q_{r+1} are flexible, then Lemma 3.5 implies $Q_r = Q_{r+1}$. Hence the statement follows.

Finally, we show that statement (1) holds. Assume that Q_r is flexible and Q_{r+1} is fixed. Then Lemma 3.4 implies that $|Q_r|$ is a period of $Q_r Q_{r+1}^{\ell_{r+1}}$, and thus $Q_{r+1}^{\ell_{r+1}} = Q_r^x Q_r[1..y]$ with $x = \lfloor |Q_{r+1}^{\ell_{r+1}}| / |Q_r| \rfloor$ and $y = |Q_{r+1}^{\ell_{r+1}}| \bmod |Q_r|$. (Either x or y might be zero, but this is irrelevant for the proof.) Let $P = Q_r[1..y]$ and $S = Q_r(y..|Q_r|]$. Any rotation of a primitive string is primitive, and hence $\mathsf{rot}^y(Q_r) = SP$ is primitive. For any exponent $a \in [\ell_r, u_r]$, it holds $Q_r^a Q_{r+1}^{\ell_{r+1}} = (PS)^a (PS)^x P = (PS)^x P (SP)^a = Q_{r+1}^{\ell_{r+1}} (\mathsf{rot}^y(Q_r))^a$. Hence the stated transformation does not change the represented affine prefix set. ◀

The leftmost (i.e., lowest index) fixed component Q_r of a representation can either be removed with $\mathsf{truncate}$ (if $r = 1$), or it can be moved further to the left with switch_{r-1} (if $r > 1$). By repeatedly applying $\mathsf{truncate}$ and switch, we obtain the following lemma.

▶ **Lemma 3.8.** *Let $\rho = \langle X, (Q_i, \ell_i, u_i)_{i=1}^{t} \rangle$ be a representation of an affine prefix set, and let $F = \{j \in [1,t] \mid \ell_j < u_j\} = \{j_1, \ldots, j_{|F|}\}$ with $j_1 < j_2 < \cdots < j_{|F|}$ be the indices of the flexible components. Then the affine prefix set has a representation $\langle \hat{X}, (\hat{Q}_{j_i}, \ell_{j_i}, u_{j_i})_{i=1}^{|F|} \rangle$ such that $\hat{Q}_{j_i}$ is a rotation of Q_{j_i} for every $i \in [1, |F|]$. Both $\hat{X}$ and all the $\hat{Q}_{j_i}$ are functions of X, $Q_1, \ldots, Q_t$, and $\ell_1, \ldots, \ell_t$, i.e., they are independent of $u_1, \ldots, u_t$.*

After applying Lemma 3.8, we repeatedly apply merge to remove all inversions. Then, we apply split until all flexible components have exponent lower bound 1. This may result in new fixed components, which we remove with Lemma 3.8, resulting in the following lemma.

▶ **Lemma 3.9.** *An affine prefix set represented by $\langle X, (Q_i, \ell_i, u_i)_{i=1}^{t} \rangle$ has an irreducible representation of order $|L| \leq t$, where $L = \{|Q_r| \mid r \in [1,t] : \ell_r < u_r\}$.*

3.2 Strongly affine representations

To analyze the intricate structure of repetitive fragments induced by affine prefix sets, it is helpful to assume that periodicity extends slightly beyond the region in question. To this end, we define a *strongly affine representation* of an affine prefix set of T, in which the exponent upper bound of each (flexible) component can be increased by five and still yield an affine prefix set of T. A supplementary drawing is provided in Figure 1b.

▶ **Definition 3.10** (Strongly affine representations). *A representation $\rho = \langle X, (Q_i, \ell_i, u_i)_{i=1}^t \rangle$ of an affine prefix set of a string T is* strongly affine *if and only if its* periodic expansion $\mathsf{expand}(\rho) = \langle X, (Q_i, \ell_i, u'_i)_{i=1}^t \rangle$ *is also the representation of an affine prefix set of T, where u'_i, $i \in [1, t]$, is defined as follows: $u'_i = u_i$ if $u_i = \ell_i$ and $u'_i = u_i + 5$ otherwise.*

▶ **Definition 3.11** (Canonical representation). *A representation of an affine prefix set is* canonical *if and only if it is both strongly affine and irreducible.*

It can be readily verified that, if ρ is strongly affine, then also $\mathsf{truncate}(\rho)$, $\mathsf{split}_r(\rho)$, $\mathsf{merge}_r(\rho)$, and $\mathsf{switch}_r(\rho)$ are strongly affine (for any r, assuming that the respective operation is indeed applicable). We obtained Lemma 3.9 by applying a sequence of these operations, and hence we have the following immediate corollary.

▶ **Corollary 3.12.** *An affine prefix set with strongly affine representation $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t \rangle$ has a canonical representation of order $|L| \leq t$, where $L = \{|Q_r| \mid r \in [1, t] : \ell_r < u_r\}$.*

Whether a representation ρ of an affine prefix set $\mathcal{A}$ of T is strongly affine does not only depend on ρ, it also depends on what T looks like beyond the end of the longest prefix represented by ρ. Therefore, one cannot hope to transform an arbitrary representation into an equivalent strongly affine representation. However, by "removing" the last five copies of each component and treating them separately, we show that we can cover an affine prefix set of order t with at most 6^t canonical representations.

▶ **Lemma 3.13.** *An affine prefix set of order t can be partitioned into at most 6^t affine prefix sets, each of which has a canonical representation of order at most t.*

Proof. Let $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t \rangle$ be a representation of an affine prefix set. We produce a set of representations $R = \{\langle X, (Q_i, \ell'_i, u'_i)_{i=1}^t \rangle \mid \forall r \in [1, t] : (\ell'_r, u'_r) \in B_r\}$, where $\forall r \in [1, t]$, we define $B_r = \{(u, u) \mid u \in [\max(\ell_r, u_r - 4), u_r]\} \cup \{(\ell_r, \max(\ell_r, u_r - 5))\}$. It is easy to see that the affine sets generated by representations in R form a partition of the affine set generated by $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t \rangle$. By design, for any representation in R, and for any component Q_r, we know that Q_r is either fixed, or it has exponent lower bound ℓ_r and exponent upper bound $u_r - 5$. Hence the instances in R are strongly affine, and it follows from Corollary 3.12 that each of them has an equivalent canonical representation of order at most t. Finally, it holds $\forall i \in [1, t] : |B_i| \leq 6$ and thus $|R| = \prod_{i=1}^t |B_i| \leq 6^t$. ◀

By applying the technique from the proof above to the prefix-palindromes, i.e., to each of the representations of order 1 produced by Lemma 3.2, we obtain the following result.

▶ **Corollary 3.14.** *The set of prefix-palindromes of a string $T[1..n]$ can be partitioned into $\mathcal{O}(\log n)$ affine sets of order at most 1. Each set of order 1 has canonical representation $\langle U(VU)^\ell, (VU, 1, u) \rangle$ for some $U \in \mathsf{PAL} \cup \{\varepsilon\}$, $V \in \mathsf{PAL}$ and integers $\ell \geq 1$ and $u > 1$.*

▶ **Corollary 3.15.** *Let $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t \rangle$ be a canonical representation of an affine prefix set. Then it holds $\forall r \in [1, t] : |Q_r| > \sum_{j=r+1}^t |Q_j^{u_j+4}|$.*

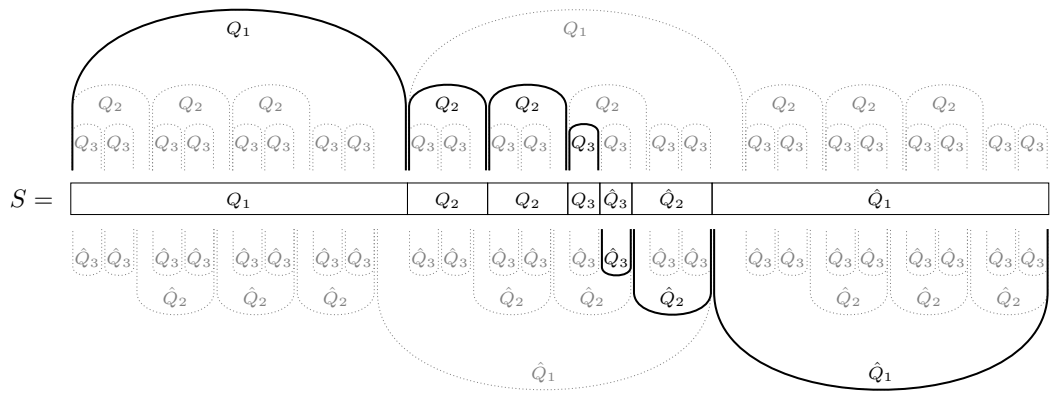


Figure 2 Lemma 3.18 applied to an irreducible representation $\langle X, (Q_1, 1, 2) \cdot (Q_2, 1, 3) \cdot (Q_3, 1, 2)\rangle$. The drawing shows the longest prefix $S = Q_1^2 Q_2^3 Q_3^2$ generated by the representation. By the lemma, for any $a_1 \in [0, 2]$, $a_2 \in [0, 3]$ and $a_3 \in [0, 2]$, it holds $S = Q_1^{2-a_1} Q_2^{3-a_2} Q_3^{2-a_3} \cdot \hat{Q}_3^{a_3} \hat{Q}_2^{a_2} \hat{Q}_1^{a_1}$, where each $\hat{Q}_j$ is the length-$|Q_j|$ suffix of S. The drawing highlights the case where $a_1 = a_2 = a_3 = 1$.

Proof. If $\rho = \langle X, (Q_i, \ell_i, u_i)_{i=1}^t\rangle$ is canonical, then clearly $\mathsf{expand}(\rho) = \langle X, (Q_i, \ell_i, u_i + 5)_{i=1}^t\rangle$ is irreducible. Thus, the statement follows from Lemma 3.5 applied to $\mathsf{expand}(\rho)$. ◀

▶ **Lemma 3.16.** *Let $\langle X, (Q_i, 1, u_i)_{i=1}^t\rangle$ be a canonical representation of an affine prefix set, and let $h \in [0, 5]$. Then $\langle \varepsilon, (Q_i, 1, u_i + h)_{i=2}^t\rangle$ is an irreducible representation of an affine prefix set of the string Q_1^2, and, if $h < 5$, also of the string Q_1.*

Proof. Consider any $h \in [0, 5]$. Due to the strong affinity, $\langle X, (Q_i, 1, u_i + h)_{i=1}^t\rangle$ represents an affine prefix set. Let $(a_i)_{i=2}^t$ be a sequence of exponents with $\forall j \in [2, t] : a_j \in [1, u_i + h]$. By Lemma 3.4, the string $Q_1 Q_2^{a_2} Q_3^{a_3} \cdots Q_t^{a_t}$ has period $|Q_1|$. Due to Lemma 3.5, it holds $|Q_2^{a_2} Q_3^{a_3} \cdots Q_t^{a_t}| < |Q_1 Q_2| < |Q_1^2|$. Hence we have shown that $Q_2^{a_2} Q_3^{a_3} \cdots Q_t^{a_t}$ is a prefix of Q_1^2, and $\langle \varepsilon, (Q_i, 1, u_i + h)_{i=2}^t\rangle$ is a representation of an affine prefix set of Q_1^2. Since $\langle X, (Q_i, 1, u_i)_{i=1}^t\rangle$ is irreducible, it is easy to see that also $\langle \varepsilon, (Q_i, 1, u_i + h)_{i=2}^t\rangle$ is irreducible.

If $h < 5$, then Lemma 3.5 invoked with $\langle X, (Q_i, 1, u_i + 5)_{i=1}^t\rangle$ implies $|Q_2^{a_2} Q_3^{a_3} \cdots Q_t^{a_t}| < |Q_1|$, and $\langle \varepsilon, (Q_i, 1, u_i + h)_{i=2}^t\rangle$ indeed only generates strings of length less than $|Q_1|$. ◀

▶ **Corollary 3.17.** *Let $\langle X, (Q_i, 1, u_i)_{i=1}^t\rangle$ be a canonical representation of an affine prefix set. Then $\langle \varepsilon, (Q_i, 1, u_i)_{i=2}^t\rangle$ is a canonical representation of an affine prefix set of the string Q_1^2.*

Proof. By Lemma 3.16, $\langle \varepsilon, (Q_i, 1, u_i + 5)_{i=2}^t\rangle$ is an irreducible representation of an affine prefix set of Q_1^2. Hence $\langle \varepsilon, (Q_i, 1, u_i)_{i=2}^t\rangle$ is a canonical representation for Q_1^2. ◀

3.3 Reversing the structure of affine prefix sets

We first show that a periodic fragment of T induced by an affine prefix set can be covered by a combination of a forward and a "backward" affine prefix set (see Figure 2):

▶ **Lemma 3.18.** *Let $\langle X, (Q_i, 1, u_i)_{i=1}^t\rangle$ be an irreducible representation of an affine prefix set, let $S = Q_1^{u_1} Q_2^{u_2} \cdots Q_t^{u_t}$, and for $j \in [1, t]$ let $\hat{Q}_j$ be the length-$|Q_j|$ suffix of S. For any sequence $(a_i)_{i=1}^t$ with $\forall j \in [1, t] : a_j \in [0, u_j]$, it holds*

$$S = Q_1^{u_1 - a_1} Q_2^{u_2 - a_2} \cdots Q_t^{u_t - a_t} \cdot \hat{Q}_t^{a_t} \hat{Q}_{t-1}^{a_{t-1}} \cdots \hat{Q}_1^{a_1}.$$

Proof. If $t = 1$, then $S = Q_1^{u_1} = \hat{Q}_1^{u_1} = Q_1^{u_1-a_1}\hat{Q}_1^{a_1}$. Inductively assume that the lemma holds for representations of order $t-1$. Now we show that it holds for representations of order t. If $\langle X, (Q_i, 1, u_i)_{i=1}^t\rangle$ is an irreducible representation of an affine prefix set, then clearly $\langle XQ_1^{u_1}, (Q_i, 1, u_i)_{i=2}^t\rangle$ is an irreducible representation of another affine prefix set. This representation is of order $t-1$, and hence the inductive assumption implies

$$S = Q_1^{u_1} \cdot Q_2^{u_2-a_2}Q_3^{u_3-a_3}\cdots Q_t^{u_t-a_t} \cdot \hat{Q}_t^{a_t}\hat{Q}_{t-1}^{a_{t-1}}\cdots\hat{Q}_2^{a_2}.$$

If $a_1 = 0$, then there is nothing left to do. Hence assume $a_1 > 0$. Since $\langle X, (Q_i, 1, u_i)_{i=1}^t\rangle$ is an irreducible representation, Lemma 3.4 implies that $|Q_1|$ and therefore also $q = a_1 \cdot |Q_1|$ is a period of S. Hence S has a border of length $s - q$, where $s = |S|$, and it holds

$$S[1..s-q] = S[1+q..s] = Q_1^{u_1-a_1} \cdot Q_2^{u_2-a_2}Q_3^{u_3-a_3}\cdots Q_t^{u_t-a_t} \cdot \hat{Q}_t^{a_t}\hat{Q}_{t-1}^{a_{t-1}}\cdots\hat{Q}_2^{a_2}.$$

Finally, as mentioned before, $S[s-q+1..s]$ of length $q = a_1 \cdot |Q_1|$ has period $|Q_1|$. Hence $S[s-q+1..s] = (S[s-|Q_1|+1..s])^{a_1} = \hat{Q}_1^{a_1}$, which concludes the proof. ◀

We now build on this characterization to convert irreducible representations of affine prefix sets of S into irreducible representations of affine prefix sets of $\mathsf{rev}(S)$.

▶ **Corollary 3.19.** *Let $\langle X, (Q_i, 1, u_i)_{i=1}^t\rangle$ be a canonical representation of an affine prefix set, let $s = \sum_{i=2}^t (u_i+1)\cdot|Q_i|$, and for $j \in [1,t]$ let $\hat{Q}_j$ be the length-$|Q_j|$ suffix of $\mathsf{rot}^s(Q_1)$. Then $\langle \varepsilon, (\mathsf{rev}(\hat{Q}_i), 1, u_i)_{i=2}^t\rangle$ represents an affine prefix set of $\mathsf{rev}(\mathsf{rot}^s(Q_1))$.*

Proof. Consider any sequence $(a_i)_{i=2}^t$ of exponents admitted by the representation, i.e., $\forall j \in [2,t] : a_j \in [1, u_j]$. By Lemma 3.16, $\langle \varepsilon, (Q_i, 1, u_i+1)_{i=2}^t\rangle$ is an irreducible representation of an affine prefix set of Q_1, which implies $Q_1[1..s] = Q_2^{u_2+1}Q_3^{u_3+1}\cdots Q_t^{u_t+1}$. For this representation, Lemma 3.18 implies that $\hat{Q}_t^{a_t}\hat{Q}_{t-1}^{a_{t-1}}\cdots\hat{Q}_2^{a_2}$ is a suffix of $Q_1[1..s]$. Thus, its reversal $\mathsf{rev}(\hat{Q}_t^{a_t}\hat{Q}_{t-1}^{a_{t-1}}\cdots\hat{Q}_2^{a_2}) = \mathsf{rev}(\hat{Q}_2^{a_2})\mathsf{rev}(\hat{Q}_3^{a_3})\cdots\mathsf{rev}(\hat{Q}_t^{a_t})$ is a prefix of $\mathsf{rev}(Q_1[1..s])$, which is a prefix of $\mathsf{rev}(\mathsf{rot}^s(Q_1))$. ◀

4 Appending a Palindrome to an Affine Prefix Set

In this section, we show how to extend an affine prefix set $\mathcal{A}$ with a palindrome. This amounts to computing the union of affine prefix sets where each new prefix is formed by concatenating a prefix from $\mathcal{A}$ with a palindrome. We distinguish two cases, depending on whether the appended palindrome lies within a periodic fragment of T. In either case, we may temporarily overextend $\mathcal{A}$, producing sets that are not affine prefix sets. We then restore validity by truncating the sets using the lemma below. For a set of strings $\mathcal{A}$, denote $\mathcal{A}|_m = \{S \in \mathcal{A} : |S| \le m\}$.

▶ **Lemma 4.1.** *Let $\langle X, (Q_i, \ell_i, u_i)_{i=1}^t\rangle$ be a representation of an affine prefix set $\mathcal{A}$. For $m \in \mathbb{N}$, we can express $\mathcal{A}' = \mathcal{A}|_m$ as a union of at most $t' \le t$ affine prefix sets $\mathcal{A}' = \bigcup_{j=1}^{t'} \mathcal{A}_j$, each with a representation of order at most t.*

Proof Sketch. The proof is by induction. For the base case $t = 1$, it is enough to reduce the upper bound on the exponent of Q_1. For $t > 1$, the proof proceeds by finding a minimal exponent $a_1 \ge \ell_1$ such that $|XQ_1^{a_1}Q_2^{u_2}\cdots Q_t^{u_t}| > m$. If $a_1 > u_1$, then $\mathcal{A}' = \mathcal{A}$. Otherwise, we assume w.l.o.g. that $\mathcal{A}$ is irreducible (see Lemma 3.9). It follows that we do not need to consider prefixes generated with an exponent larger than a_1 for Q_1. We partition the remaining prefixes into two sets, one of which can be further broken down using the inductive hypothesis, reducing the problem to problems of smaller sizes until the result follows. ◀

4.1 Appending a long palindrome

Assume that the affine prefix set to be extended is given in a canonical representation $\langle X, (Q_i, 1, u_i)_{i=1}^t \rangle$. We first focus on appending *long palindromes* of length at least $2|Q_1|$, and then we show that the shorter palindromes can be handled recursively. Note that, for a canonical representation, T has a prefix $XQ_1^{u_1+5}$. At the same time, the longest prefix in the affine set is of length less than $XQ_1^{u_1+1}$ by Corollary 3.15. This leads us to a case distinction based on the center of the palindrome to be appended. If the center is before position $|XQ_1^{u_1+3}|$, then we can show that the entire palindrome is within the $|Q_1|$-periodic prefix of $T[|X|+1..n]$. Otherwise, the left half of the palindrome contains position $|XQ_1^{u_1+2}|$, and we can use this position as an anchor point for the extension.

4.1.1 Appending a long palindrome within a run of Q_1

We now focus on the case where the long palindrome to be appended is entirely within the $|Q_1|$-periodic prefix of $T[|X|+1..n]$. We proceed in two steps. First (in Theorem 4.2), we show how to append a palindrome under the assumption that the entire string has the form XQ_1^x for some integer x. Secondly (Corollary 4.3), we truncate the result of the first step so that it corresponds to XQ_1^α, where $\alpha \in \mathbb{Q}$ is the largest value such that XQ_1^α is a prefix of T.

▶ **Theorem 4.2.** *Let $\langle X, (Q_i, 1, u_i)_{i=1}^t \rangle$ be a canonical representation of an affine prefix set $\mathcal{A}$. Let $s = \sum_{i=2}^t (u_i+1) \cdot |Q_i|$, and for $j \in [1,t]$ let $\hat{Q}_j$ be the length-$|Q_j|$ suffix of $\mathsf{rot}^s(Q_1)$. If $\mathsf{rot}^r(Q_1) = \mathsf{rev}(Q_1)$ for some $r \in [s, s+|Q_1|)$, then*

$$\langle X \cdot Q_1 \cdot Q_1[1..r-s], (\mathsf{rot}^{r-s}(Q_1), 1, x) \cdot (\mathsf{rev}(\hat{Q}_i), 1, u_i)_{i=2}^t \rangle \tag{1}$$

represents an affine prefix set $\mathcal{A}'$ of XQ_1^{x+3}, for any positive integer x. Furthermore:

1. *If $Y' \in \mathcal{A}'$, then there is a string $Y \in \mathcal{A}$ and a palindrome P such that $Y' = YP$.*
2. *For $Y \in \mathcal{A}$ and $P \in \mathsf{PAL}$, if $|P| \geq 2|Q_1|$ and YP is a prefix of XQ_1^{x+1}, then $YP \in \mathcal{A}'$.*

Proof Sketch. The keystone of the proof is Corollary 3.19 which implies that a string $Q' = \mathsf{rev}(\hat{Q}_2)^{a_2}\mathsf{rev}(\hat{Q}_3)^{a_3} \cdots \mathsf{rev}(\hat{Q}_t)^{a_t}$, where $\forall j \in [2,t] : a_j \in [1, u_j]$, is a prefix of

$$\mathsf{rev}(\mathsf{rot}^s(Q_1)) = \mathsf{rot}^{-s}(\mathsf{rev}(Q_1)) = \mathsf{rot}^{-s}(\mathsf{rot}^r(Q_1)) = \mathsf{rot}^{r-s}(Q_1).$$

Using this fact, we first establish that a string XS' generated by the canonical representation in Equation (1) is a prefix of XQ_1^{x+3}. Next, we show that S' can be decomposed as SP, where $XS \in \mathcal{A}$ and P is a palindrome. It follows that P is a substring of a power of Q_1, and the condition $\mathsf{rot}^r(Q_1) = \mathsf{rev}(Q_1)$ ensures P is a palindrome. Finally, we conclude by considering any string $S \in \mathcal{A}$ and a sufficiently long palindrome P such that SP is a prefix of Q_1^{x+1}. Due to $XS \in \mathcal{A}$, there is some sequence $\forall j \in [1,t] : a_j \in [1, u_j]$ of exponents such that $S = Q_1^{u_1-a_1+1}Q_2^{u_2-a_2+1} \cdots Q_t^{u_t-a_t+1}$. From this and the fact that P is a palindrome, we show that XSP fits the structure required for membership in $\mathcal{A}'$, completing the proof. ◀

By combining Theorem 4.2 and Lemma 4.1, we obtain:

▶ **Corollary 4.3.** *Let $\langle X, (Q_i, 1, u_i)_{i=1}^t \rangle$ be a canonical representation of an affine prefix set $\mathcal{A}$. Let $\alpha \in \mathbb{Q}$ be the largest possibly fractional exponent such that XQ_1^α is a prefix of T, and define $\mathcal{S} = \{S \cdot P : S \cdot P$ is a prefix of $XQ_1^\alpha, S \in \mathcal{A}, P \in \mathsf{PAL}, |P| \geq 2|Q_1|\}$. There are $t' \leq t$ affine prefix sets $\mathcal{B}_i$, $i \in [1,t']$, each of order $\leq t$, such that for $\mathcal{B} = \bigcup_{i=1}^{t'} \mathcal{B}_i$ we have $\mathcal{S} \subseteq \mathcal{B}$ and for every $Y' \in \mathcal{B}$, there is a string $Y \in \mathcal{A}$ and a palindrome P such that $Y' = YP$.*

4.1.2 Appending a long palindrome outside a run of Q_1

▶ **Theorem 4.4.** *Let $\langle X, (Q_i, 1, u_i)_{i=1}^t\rangle$ be a canonical representation of an affine prefix set $\mathcal{A}$ and $s = \sum_{i=2}^t (u_i + 1) \cdot |Q_i|$. For $j \in [1, t]$, let $\hat{Q}_j$ be the length-$|Q_j|$ suffix of $\mathsf{rot}^s(Q_1)$. For any string P, $\langle X \cdot Q_1^{u_1+2} \cdot P \cdot \mathsf{rev}(Q_1)[1..|Q_1| - s], (\mathsf{rev}(\hat{Q}_i), 1, u_i)_{i=1}^t\rangle$ represents an affine prefix set $\mathcal{A}'$ of the string $X \cdot Q_1^{u_1+2} \cdot P \cdot \mathsf{rev}(Q_1^{u_1+2})$, where $\mathcal{A}' = \{SWP \cdot \mathsf{rev}(W) \mid S \in \mathcal{A}$ and $SW = X \cdot Q_1^{u_1+2}\}$.*

Proof Sketch. Let $q = |Q_1|$. We can split the output representation into a concatenation

$$\langle X \cdot Q_1^{u_1+2} \cdot P \cdot \mathsf{rev}(Q_1)[1..q - s], (\mathsf{rev}(\hat{Q}_1), 1, u_1)\rangle \cdot \langle \varepsilon, (\mathsf{rev}(\hat{Q}_i), 1, u_i)_{i=2}^t\rangle. \tag{2}$$

We first apply Corollary 3.19 to deduce that Equation (2) represents an affine prefix set of the string $X \cdot Q_1^{u_1+2} \cdot P \cdot \mathsf{rev}(Q_1^{u_1+2})$. Secondly, we show that every element in $\mathcal{A}$ contributes exactly one element to $\mathcal{A}'$, and hence $|\mathcal{A}'| = |\mathcal{A}|$. It thus suffices to show that any string generated by Equation (2) is in $\mathcal{A}'$. It then readily follows that Equation (2) generates exactly $\mathcal{A}'$. To do so, we consider any string S' generated by Equation (2). Such a string must be of the form $S' = XQ_1^{u_1+2}P \cdot \mathsf{rev}(W)$, where $\mathsf{rev}(W) = \mathsf{rev}(Q_1)[1..q - s] \cdot \mathsf{rev}(\hat{Q}_1)^{a_1}\mathsf{rev}(\hat{Q}_2)^{a_2} \cdots \mathsf{rev}(\hat{Q}_t)^{a_t}$ for some exponents $\forall i \in [1, t] : a_i \in [1, u_i]$. By our previous observations, $\mathsf{rev}(W)$ is a prefix of $\mathsf{rev}(Q_1)^{u_1+2}$, and thus there is a unique string S such that $SW = XQ_1^{u_1+2}$ and $S' = SWP \cdot \mathsf{rev}(W)$. It remains to be shown that $S \in \mathcal{A}$, which then implies $S' \in \mathcal{A}'$. For this purpose, we carefully analyze the length of S and show that a prefix of length $|S|$ indeed belongs to $\mathcal{A}$, concluding the proof. ◀

For a fragment $P = T[x..y]$ of T, denote its center $(x + y)/2$ by $\mathrm{cen}(P)$.

▶ **Corollary 4.5.** *Let $\langle X, (Q_i, 1, u_i)_{i=1}^t\rangle$ be a canonical representation of an affine prefix set $\mathcal{A}$, and consider the set of strings $\mathcal{A}' = \{S \cdot P : S \cdot P$ is a prefix of $T, S \in \mathcal{A}, P \in \mathsf{PAL}, \mathrm{cen}(P) > |XQ_1^{u_1+3}|\}$. There are $t' = \mathcal{O}(t \log n)$ affine prefix sets $\mathcal{B}_i$, $i \in [1, t']$, each of order $\leq t + 1$, such that both of the following properties hold for $\mathcal{B} = \bigcup_{i=1}^{t'} \mathcal{B}_i$:*
1. *$\mathcal{A}' \subseteq \mathcal{B}$.*
2. *For every $Y' \in \mathcal{B}$, there is a string $Y \in \mathcal{A}$ and a palindrome P such that $Y' = YP$.*

Proof Sketch. Consider any $SP \in \mathcal{A}'$, where SP is a prefix of T, $S \in \mathcal{A}$, $P \in \mathsf{PAL}$, $\mathrm{cen}(P) > |XQ_1^{u_1+3}|$. Due to $S \in \mathcal{A}$, Corollary 3.15 implies $|S| < |XQ_1^{u_1+1}|$. Let $P' = T[x..y]$, where $x = 1 + |XQ_1^{u_1+2}|$ and $y = 2 \cdot \mathrm{cen}(P) - x$. We claim that $P' \in \mathsf{PAL}$. Indeed, the starting position $|S| + 1$ of P is less than the starting position x of P', and the centers of P and P' coincide with $\mathrm{cen}(P) - x = y - \mathrm{cen}(P)$. We call P' the *core palindrome* of SP. Note that every core palindrome is a prefix of $T[x..n]$ (which is independent of SP). Therefore, by Corollary 3.14, the set of core palindromes can be represented as the union of $\mathcal{O}(\log n)$ affine prefix sets. Let $\mathcal{C}$ be any of these sets. We now describe how to compute the part of $\mathcal{A}'$ that contains strings of the form $SP = SWP' \cdot \mathsf{rev}(W)$, where $S \in \mathcal{A}$, $P \in \mathsf{PAL}$, and the core palindrome of SP is some $P' \in \mathcal{C}$. The procedure depends on the representation of $\mathcal{C}$, which, by Corollary 3.14, is covered by one of the following cases. Let $q = |Q_1|$.

Case 1: $\mathcal{C}$ is given in strongly affine representation $\langle U \cdot (VU)^\ell, (VU, 1, u)\rangle$, where VU is primitive and $|VU| > q$. In this case, we consider one fixed core palindrome in $\mathcal{C}$ and apply Theorem 4.4 and Lemma 4.1 to obtain an affine prefix set generated by it. We then show that the sets generated by other core palindromes have a similar representation, which allows to union them and to obtain the final affine prefix set.

Case 2: $\mathcal{C}$ is given in representation $\langle P', \varepsilon\rangle$ of order 0, i.e., it contains a single core palindrome P'. We proceed exactly like in Case 1, but with a single palindrome.

Case 3: $\mathcal{C}$ has strongly affine representation $\langle U \cdot (VU)^\ell, (VU, 1, u)\rangle$, where VU is primitive and $|VU| = q$. For $i \in [1, u]$, let $P_i = U \cdot (VU)^{\ell+i}$. We show that, if $\mathcal{A}'$ contains some $SP = S \cdot W \cdot P_i \cdot \mathsf{rev}(W) = XQ_1^{u_1+2} P_i \cdot \mathsf{rev}(W)$ with $S \in \mathcal{A}$, then the entire SP can be written as XQ_1^α for some $\alpha \in \mathbb{Q}$. Hence we can simply apply Corollary 4.3 and obtain that the affine prefix set generated by $\mathcal{C}$ is the union of at most t affine prefix sets of order at most t.

Case 4: $\mathcal{C}$ has strongly affine representation $\langle U \cdot (VU)^\ell, (VU, 1, u)\rangle$, where VU is primitive and $|VU| < q$. We show that this case is impossible due to primitivity of Q_1.

We call the created affine prefix sets $\mathcal{B}_i$. There are $\mathcal{O}(\log n)$ core palindrome sets, each handled by a single case. Each case creates $\leq t$ representations of order $\leq t + 1$. Hence, there are $\mathcal{O}(t \log n)$ affine prefix sets $\mathcal{B}_i$ in total, each of order $\leq t + 1$ Furthermore, the four cases cover all possibilities, and hence $\mathcal{A}' \subseteq \mathcal{B} = \bigcup_{i=1}^{t'} \mathcal{B}_i$. The second property holds by construction of the sets $\mathcal{B}_i$. ◀

4.1.3 Appending all long palindromes and recursion

▶ **Lemma 4.6.** *Let $\langle X, (Q_i, 1, u_i)_{i=1}^t\rangle$ be a canonical representation of an affine prefix set $\mathcal{A}$ and $\mathcal{A}' = \{S \cdot P \mid S \in \mathcal{A}, P \in \mathsf{PAL}, |P| \geq 2|Q_1|$, and $S \cdot P$ is a prefix of $T\}$. There are $t' = \mathcal{O}(t \log n)$ affine prefix sets $\mathcal{B}_i$, $1 \leq i \leq t'$, each of order $\leq t + 1$, such that $\mathcal{A}' \subseteq \cup_{i=1}^{t'} \mathcal{B}_i$ and for each string $S' \in \cup_{i=1}^{t'} \mathcal{B}_i$, there is a string $S \in \mathcal{A}$ and $P \in \mathsf{PAL}$ such that $S' = S \cdot P$.*

Proof. We consider the sets from Corollaries 4.3 and 4.5, defined by

$$\mathcal{A}_1 = \{S \cdot P : S \cdot P \text{ is a prefix of } XQ_1^\alpha, S \in \mathcal{A}, P \in \mathsf{PAL}, |P| \geq 2|Q_1|\} \text{ and}$$

$$\mathcal{A}_2 = \{S \cdot P : S \cdot P \text{ is a prefix of } T, S \in \mathcal{A}, P \in \mathsf{PAL}, \mathrm{cen}(P) > |X| + (u_1 + 3) \cdot |Q_1|\},$$

where α is the largest (possibly fractional) exponent such that XQ_1^α is a prefix of T. Due to Corollaries 4.3 and 4.5, we can express (a superset of) $\mathcal{A}_1 \cup \mathcal{A}_2$ as the union of $\mathcal{O}(t \log n)$ affine prefix sets, each of order $\leq t + 1$, where every string in each of the prefix sets is the concatenation of a string from $\mathcal{A}$ and a palindrome. It remains to be shown that $\mathcal{A}' \subseteq \mathcal{A}_1 \cup \mathcal{A}_2$. For the sake of contradiction, assume that there is some string $SP \in \mathcal{A}' \setminus (\mathcal{A}_1 \cup \mathcal{A}_2)$, where $S \in \mathcal{A}$, $P \in \mathsf{PAL}$ and $|P| \geq 2|Q_1|$. Due to $SP \notin \mathcal{A}_1$, SP is not a prefix of XQ_1^α and hence it must be longer than XQ_1^α. Let $m = |XQ_1^\alpha| - |S|$. We show a lower bound on m. Since the given representation is strongly affine, it holds $\alpha \geq u_1 + 5$. It is also irreducible, and hence Corollary 3.15 implies $|S| < |XQ_1^{u_1+1}|$. Therefore, it holds $m > 4|Q_1|$. Note that P does not have period $|Q_1|$, but its length-m prefix, which is a suffix of Q_1^α, does. Hence, by Lemma 2.3, it follows that P is of length over $2m - |Q_1|$, and therefore

$$\mathrm{cen}(P) \geq |S| + |P|/2 > |S| + m - |Q_1|/2 = |XQ_1^\alpha| - |Q_1|/2 > |XQ_1^{u_1+4}|.$$

This implies $SP \in \mathcal{A}_2$, which contradicts the initial assumption. ◀

We have shown that appending palindromes of length at least $2|Q_1|$ results in $\mathcal{O}(t \log n)$ affine prefix sets of order $\leq t+1$. For appending shorter palindromes, we exploit properties of strongly affine prefix sets that allow us to apply the previously described approach recursively.

▶ **Lemma 4.7.** *Let $\langle X, (Q_i, 1, u_i)_{i=1}^t\rangle$ be a canonical representation of an affine prefix set $\mathcal{A}$ and $\mathcal{A}' = \{S \cdot P : S \cdot P$ is a prefix of $T, S \in \mathcal{A}, P \in \mathsf{PAL}\}$ a set of strings. Then $\mathcal{A}'$ is a union of $\mathcal{O}((t + 1)^2 \log n)$ affine prefix sets, each of order $\leq t + 1$.*

▶ **Theorem 1.1.** *Let $0 < \epsilon < 1$ be constant, $T[1..n]$ a string, and $k \in \mathbb{N}^+$. The set of prefixes of T that belong to PAL^k is the union of $\mathcal{O}(6^{k^2/(2-\epsilon)} \cdot \log^k n)$ affine prefix sets of order $\leq k$.*

Proof. We start with the empty affine prefix set representing PAL^0. We proceed in k levels $k' \in [0, k)$. The union of the affine prefix sets of level k' is exactly the set of all k'-palindromic prefixes of T. For each affine prefix set of the current level k', we first apply Lemmas 3.9 and 3.13 to obtain at most $6^{k'}$ canonical representations of order $\leq k'$. Then, for each of the representations, we append a palindrome using Lemma 4.7, resulting in at most $c \cdot (k'+1)^2 \log n$ affine prefix sets of order at most $k'+1$, which we move to level $k'+1$. Here, c is a positive constant that depends on the precise complexity analysis of Lemma 4.7. Hence, after processing level $k-1$, the total number of affine prefix sets is bounded by $\prod_{k'=0}^{k-1}(6^{k'} \cdot c \cdot (k'+1)^2 \log n) \leq (k!)^2 \cdot c^k \cdot 6^{(k^2/2)} \cdot \log^k n$. For all sufficiently large k (depending only on ϵ and c), the bound simplifies to $6^{k^2/(2-\epsilon)} \cdot \log^k n$. (And for the remaining values of k, we have $(k!)^2 \cdot c^k \cdot 6^{(k^2/2)} = \mathcal{O}(1)$.) ◀

▶ Remark 4.8. Lemmas 3.9 and 3.13 only work with the lengths of the components in the representations of affine prefix sets and the exponents bounds. Since all affine prefix sets are of small order, it is not difficult to see that these lemmas can be implemented efficiently. To implement Lemma 4.7, we make use of two procedures: The first one computes the longest prefix of a string of form XQ^α, where $\langle X, (Q_i, 1, u_i)_{i=1}^t \rangle$ is a representation of one of the sets, and the second computes a representation of the set of prefix-palindromes of a string. In the read-only model, both procedures can be implemented in $\mathcal{O}(n)$ time and $\mathcal{O}(\log n)$ space. Bounding the number of calls by the number of the generated affine prefix sets, we finally obtain Theorem 1.3. The algorithm of Theorem 1.3 can then be used to test if the palindromic length of T is at most k by checking whether T is a k-palindromic prefix, however, to achieve even better complexity, we run two copies of the algorithm, one from the left and the other from the right, and then combine their results to obtain Theorem 1.4.

References

1. Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. If the current clique algorithms are optimal, so is Valiant's parser. *SIAM J. Comput.*, 47(6):2527–2555, 2018. doi:10.1137/16M1061771.
2. Rajeev Alur and P. Madhusudan. Visibly pushdown languages. In *STOC*, pages 202–211, 2004. doi:10.1145/1007352.1007390.
3. Amihood Amir and Benny Porat. Approximate on-line palindrome recognition, and applications. In *CPM*, pages 21–29, 2014. doi:10.1007/978-3-319-07566-2_3.
4. Ajesh Babu, Nutan Limaye, Jaikumar Radhakrishnan, and Girish Varma. Streaming algorithms for language recognition problems. *Theor. Comput. Sci.*, 494:13–23, 2013. doi:10.1016/J.TCS.2012.12.028.
5. Gabriel Bathie, Jonas Ellert, and Tatiana Starikovskaya. Small space encoding and recognition of k-palindromic prefixes. *CoRR*, abs/2410.03309, 2024. doi:10.48550/arXiv.2410.03309.
6. Gabriel Bathie, Tomasz Kociumaka, and Tatiana Starikovskaya. Small-space algorithms for the online language distance problem for palindromes and squares. In *ISAAC*, pages 10:1–10:17, 2023. doi:10.4230/LIPICS.ISAAC.2023.10.
7. Gabriel Bathie and Tatiana Starikovskaya. Property testing of regular languages with applications to streaming property testing of visibly pushdown languages. In *ICALP*, pages 119:1–119:17, 2021. doi:10.4230/LIPICS.ICALP.2021.119.
8. Petra Berenbrink, Funda Ergün, Frederik Mallmann-Trenn, and Erfan Sadeqi Azer. Palindrome recognition in the streaming model. In *STACS*, pages 149–161, 2014. doi:10.4230/LIPICS.STACS.2014.149.

9 Allan Borodin and Stephen A. Cook. A time-space tradeoff for sorting on a general sequential model of computation. In *STOC*, pages 294–301, 1980. doi:10.1145/800141.804677.

10 Kirill Borozdin, Dmitry Kosolobov, Mikhail Rubinchik, and Arseny M. Shur. Palindromic length in linear time. In *CPM*, pages 23:1–23:12, 2017. doi:10.4230/LIPIcs.CPM.2017.23.

11 Bartlomiej Dudek, Pawel Gawrychowski, Garance Gourdel, and Tatiana Starikovskaya. Streaming regular expression membership and pattern matching. In *SODA*, pages 670–694, 2022. doi:10.1137/1.9781611977073.30.

12 Gabriele Fici, Travis Gagie, Juha Kärkkäinen, and Dominik Kempa. A subquadratic algorithm for minimum palindromic factorization. *J. Discrete Algorithms*, 28:41–48, 2014. doi:10.1016/J.JDA.2014.08.001.

13 Nathan Fine and Herbert Wilf. Uniqueness theorems for periodic functions. *Proceedings of the American Mathematical Society*, 16(1):109–114, 1965. doi:10.2307/2034009.

14 Nathanaël François, Frédéric Magniez, Michel de Rougemont, and Olivier Serre. Streaming property testing of visibly pushdown languages. In *ESA*, pages 43:1–43:17, 2016. doi:10.4230/LIPICS.ESA.2016.43.

15 Michael L. Fredman and Dan E. Willard. BLASTING through the information theoretic barrier with FUSION TREES. In *STOC*, pages 1–7, 1990. doi:10.1145/100216.100217.

16 Zvi Galil. On converting on-line algorithms into real-time and on real-time algorithms for string-matching and palindrome recognition. *SIGACT News*, 7(4):26–30, 1975. doi:10.1145/990502.990505.

17 Zvi Galil and Joel I. Seiferas. A linear-time on-line recognition algorithm for "palstar". *J. ACM*, 25(1):102–111, 1978. doi:10.1145/322047.322056.

18 Moses Ganardi. Visibly pushdown languages over sliding windows. In *STACS*, pages 29:1–29:17, 2019. doi:10.4230/LIPICS.STACS.2019.29.

19 Moses Ganardi, Danny Hucke, Daniel König, Markus Lohrey, and Konstantinos Mamouras. Automata theory on sliding windows. In *STACS*, pages 31:1–31:14, 2018. doi:10.4230/LIPICS.STACS.2018.31.

20 Moses Ganardi, Danny Hucke, and Markus Lohrey. Querying regular languages over sliding windows. In *FSTTCS*, pages 18:1–18:14, 2016. doi:10.4230/LIPICS.FSTTCS.2016.18.

21 Moses Ganardi, Danny Hucke, and Markus Lohrey. Randomized sliding window algorithms for regular languages. In *ICALP*, pages 127:1–127:13, 2018. doi:10.4230/LIPICS.ICALP.2018.127.

22 Moses Ganardi, Danny Hucke, Markus Lohrey, and Tatiana Starikovskaya. Sliding window property testing for regular languages. In *ISAAC*, pages 6:1–6:13, 2019. doi:10.4230/LIPICS.ISAAC.2019.6.

23 Moses Ganardi, Artur Jez, and Markus Lohrey. Sliding windows over context-free languages. In *MFCS*, pages 15:1–15:15, 2018. doi:10.4230/LIPICS.MFCS.2018.15.

24 Pawel Gawrychowski, Oleg Merkurev, Arseny M. Shur, and Przemyslaw Uznanski. Tight tradeoffs for real-time approximation of longest palindromes in streams. *Algorithmica*, 81(9):3630–3654, 2019. doi:10.1007/S00453-019-00591-8.

25 Tomohiro I, Shiho Sugimoto, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Computing palindromic factorizations and palindromic covers on-line. In *CPM*, pages 150–161, 2014. doi:10.1007/978-3-319-07566-2_16.

26 Rahul Jain and Ashwin Nayak. The space complexity of recognizing well-parenthesized expressions in the streaming model: The index function revisited. *IEEE Trans. Inf. Theory*, 60(10):6646–6668, 2014. doi:10.1109/TIT.2014.2339859.

27 Donald E. Knuth, James H. Morris Jr., and Vaughan R. Pratt. Fast pattern matching in strings. *SIAM J. Comput.*, 6(2):323–350, 1977. doi:10.1137/0206024.

28 Tomasz Kociumaka, Tatiana Starikovskaya, and Hjalte Wedel Vildhøj. Sublinear space algorithms for the longest common substring problem. In *ESA*, pages 605–617, 2014. doi:10.1007/978-3-662-44777-2_50.

29 Dmitry Kosolobov, Mikhail Rubinchik, and Arseny M. Shur. Pal^k is linear recognizable online. In *SOFSEM*, pages 289–301, 2015. doi:10.1007/978-3-662-46078-8_24.

30 Andreas Krebs, Nutan Limaye, and Srikanth Srinivasan. Streaming algorithms for recognizing nearly well-parenthesized expressions. In *MFCS*, pages 412–423, 2011. doi:10.1007/978-3-642-22993-0_38.

31 Frédéric Magniez, Claire Mathieu, and Ashwin Nayak. Recognizing well-parenthesized expressions in the streaming model. *SIAM J. Comput.*, 43(6):1880–1905, 2014. doi:10.1137/130926122.

32 Glenn K. Manacher. A new linear-time "on-line" algorithm for finding the smallest initial palindrome of a string. *J. ACM*, 22(3):346–351, 1975. doi:10.1145/321892.321896.

33 Mikhail Rubinchik and Arseny M. Shur. EERTREE: an efficient data structure for processing palindromes in strings. *Eur. J. Comb.*, 68:249–265, 2018. doi:10.1016/J.EJC.2017.07.021.

34 Mikhail Rubinchik and Arseny M. Shur. Palindromic k-factorization in pure linear time. In *MFCS*, pages 81:1–81:14, 2020. doi:10.4230/LIPICS.MFCS.2020.81.

35 Anatol O. Slisenko. A simplified proof of the real-time recognizability of palindromes on Turing machines. *J. Sov. Math.*, 15:68–77, 1981. doi:10.1007/BF01404109.