Tight Bounds on the Maximum Number of Shortest Unique Substrings*

Takuya Mieno[1], Shunsuke Inenaga[2], Hideo Bannai[3], and Masayuki Takeda[4]

1 **Department of Informatics, Kyushu University, Japan**
takuya.mieno@inf.kyushu-u.ac.jp
2 **Department of Informatics, Kyushu University, Japan**
inenaga@inf.kyushu-u.ac.jp
3 **Department of Informatics, Kyushu University, Japan**
bannai@inf.kyushu-u.ac.jp
4 **Department of Informatics, Kyushu University, Japan**
takeda@inf.kyushu-u.ac.jp

Abstract

A substring Q of a string S is called a shortest unique substring (SUS) for interval $[s,t]$ in S, if Q occurs exactly once in S, this occurrence of Q contains interval $[s,t]$, and every substring of S which contains interval $[s,t]$ and is shorter than Q occurs at least twice in S. The SUS problem is, given a string S, to preprocess S so that for any subsequent query interval $[s,t]$ all the SUSs for interval $[s,t]$ can be answered quickly. When $s = t$, we call the SUSs for $[s,t]$ as *point SUSs*, and when $s \leq t$, we call the SUSs for $[s,t]$ as *interval SUSs*. There exist optimal $O(n)$-time preprocessing scheme which answers queries in optimal $O(k)$ time for both point and interval SUSs, where n is the length of S and k is the number of outputs for a given query. In this paper, we reveal structural, combinatorial properties underlying the SUS problem: Namely, we show that the number of intervals in S that correspond to point SUSs for all query positions in S is less than $1.5n$, and show that this is a matching upper and lower bound. Also, we consider the maximum number of intervals in S that correspond to interval SUSs for all query intervals in S.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases shortest unique substrings, maximal unique substrings

Digital Object Identifier 10.4230/LIPIcs.CPM.2017.24

1 Introduction

1.1 Shortest unique substring (SUS) problems

A substring Q of a string S is called a *shortest unique substring* (*SUS*) for interval $[s,t]$ in S, if (1) Q occurs exactly once in S, (2) this occurrence of Q contains interval $[s,t]$, and (3) every substring of S which contains interval $[s,t]$ and is shorter than Q occurs at least twice in S. The *SUS problem* is to preprocess a given string S so that for any subsequent query interval $[s,t]$, SUSs for interval $[s,t]$ can be answered quickly. When $s = t$, a query $[s,t]$ refers to a single position in the string S, and the problem is specifically called the *point SUS problem*. For clarity, when $s \leq t$, the problem is called the *interval SUS problem*.

* This work was in part supported by JSPS KAKENHI Grant Numbers JP25240003, JP26280003, JP16H02783, JP17H01697.

28th Annual Symposium on Combinatorial Pattern Matching (CPM 2017).
Editors: Juha Kärkkäinen, Jakub Radoszewski, and Wojciech Rytter; Article No. 24; pp. 24:1–24:11
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Pei et al. [5] were the first to consider the point SUS problem, motivated by some applications in bioinformatics. They considered two versions of this problem, depending on whether a single point SUS has to be returned (the *single point SUS problem*) or all point SUSs have to be returned (the *all point SUSs problem*) for a query position.

There is a series of research for the single point SUS problem. Pei et al. [5] gave an $O(n^2)$-time preprocessing scheme which returns a single point SUS for a query position in $O(1)$ time, where n is the length of the input string. Tsuruta et al. [6] and Ileri et al. [3] independently showed optimal $O(n)$-time preprocessing schemes which return a single point SUS for a query position in $O(1)$ time. Hon et al. [1] proposed an *in-place* algorithm for the same version of the problem, achieving the same bounds as the above solutions.

For the all point SUS problem which is more difficult, Tsuruta et al. [6] and Ileri et al. [3] also showed optimal algorithms achieving $O(n)$ preprocessing time and $O(k)$ query time, where k is the number of all point SUSs for a query point.

Hu et al. [2] were the first to consider the interval SUS problem, and they proposed an optimal algorithm for the interval SUS problem, using $O(n)$ time for preprocessing and $O(k')$ time for queries, where k' is the number of interval SUSs for a query interval. Recently, Mieno et al. [4] proposed an algorithm which solves the interval SUS problem on strings represented by *run-length encoding* (RLE). If r is the size of the RLE of a given string of length n, then $r \leq n$ always holds. Mieno et al.'s algorithm uses $O(r)$ space, requires $O(r \log r)$ time to construct, and answers all SUSs for a query interval in $O(k' + \sqrt{\log r / \log\log r})$ time.

A substring X of a string S is said to be a *minimal unique substring* (*MUS*) of S, if (i) X occurs in S exactly once and (ii) every proper substring of X occurs at least twice in S. All the above algorithms for the SUS problems pre-compute all MUSs of the input string S (or some data structure which is essentially equivalent to MUSs), and extensively use MUSs to return the SUSs for a query position or interval.

Tsuruta et al. [6] showed that the maximum number of MUSs contained in a string of length n is at most n. This immediately follows from the fact that MUSs do not nest. Mieno et al. [4] proved that the maximum number of MUSs in a string is bounded by $2r - 1$, where r is the size of the RLE of the string. They also showed a series of strings which have $2r - 1$ MUSs, and hence this bound is tight. These properties played significant roles in designing efficient algorithms for the SUS problems.

On the other hand, structural properties of SUSs are not well understood. A trivial upperbound for the maximum number of intervals that correspond to point SUSs is $3n$, since every MUS can be a SUS for some position of the input string S, and for each query position p ($1 \leq p \leq n$), there can be at most 2 SUSs that are not MUSs (one that ends at position p and the other that begins at position p).

1.2 Our contribution

The main contribution of this paper is matching upper and lower bounds for the maximum number of SUSs for the point SUS problem, which translate to "less than $1.5n$ point SUSs". Namely, we prove that any string of length n contains at most $(3n - 1)/2$ SUSs for the point SUS problem. We give a series of strings which contains $(3n - 1)/2$ SUSs for any odd number $n \geq 5$. Therefore, our bound is tight, and to our knowledge, this is the first non-trivial result for structural properties of SUSs.

We also consider the maximum number of SUSs for the interval SUS problem. In so doing, we exclude a special case where a query interval $[s, t]$ itself is a unique substring that occurs exactly once in S. This is because we have $\Theta(n^2)$ bounds for such trivial SUSs. We then prove that any string of length n contains less than $2n$ *non-trivial* SUSs for the interval

SUS problem. We also prove that there exists a string of length n which contains $(2-\varepsilon)n$ non-trivial SUSs for any small number $\varepsilon > 0$.

1.3 Related work

Xu [7] introduced the *longest repeat* (*LR*) problem. An interval $[i, j]$ of a string S is said to be an LR for interval $[s, t]$ if (a) the substring $R = S[i..j]$ occurs at least twice in S, (b) the occurrence $[i, j]$ of R contains $[s, t]$ and (c) there does not exist an interval $[i', j']$ of S such that $j' - i' > j - i$, the substring $S[i'..j']$ occurs at least twice in S, and the interval $[i', j']$ contains interval $[s, t]$. The point and interval LR problems are defined analogously as the point and interval SUS problems, respectively.

Xu [7] presented an optimal algorithm which, after $O(n)$-time preprocessing, returns all LRs for a given interval in $O(k'')$ time, where k'' is the number of output LRs. He claimed that although the point/interval SUS problems and the point/interval LR problems look alike, these problems are actually quite different, with a support from an example where an SUS and LR for the same query point seem rather unrelated.

Our $(3n-1)/2$ bound for the maximum number of SUSs for the point SUS problem also supports his claim in the following sense: In the preprocessing, Xu's algorithm computes the set of *maximal repeats* (*MR*). An interval $[i, j]$ of a string S is said to be an MR if (A) the substring $W = S[i..j]$ occurs at least twice in S, and (B) for any $1 \leq i' \leq i \leq j \leq j' \leq n$ with $j' - i' > j - i$, the superstring $Y = S[i'..j']$ of W occurs once in S. It is easy to see that the maximum number of MRs is bounded by n, since for any position in S, there can be at most one MR that begins at that position. This bound is also tight: any even palindrome consisting of $n/2$ distinct characters contains n intervals for which the corresponding substrings are MRs (e.g., for even palindrome `abcdeedcba` of length 10, any interval $[i, i]$ for $1 \leq i \leq 10$ is an MR). By definition, any LR of string S is also an MR of S. Hence, the maximum number of LRs is also bounded by n. Since the above lower bound for MRs with palindromes also applies to LRs, this upper bound for LRs is also tight. Thus, there is a gap of $(n-1)/2$ between the maximum numbers of SUSs and LRs.

2 Preliminaries

2.1 Notations

Let Σ be the alphabet. An element of Σ^* is called a string. We denote the length of string S by $|S|$. The empty string is the string of length 0. For any string S of length n and integer $1 \leq i \leq n$, let $S[i]$ denote the ith character of S. For any $1 \leq i \leq j \leq n$, let $S[i..j]$ denote the substring of S that starts at position i and ends at position j in S. For convenience, $S[i..j]$ is the empty string if $i > j$. For any strings S and w, let $\#occ_S(w)$ denote the number of occurrences of w in S, namely, $\#occ_S(w) = |\{i : S[i..i + |w| - 1] = w\}|$.

2.2 MUSs and SUSs

Let S be any string of length n, and w be any non-empty substring of S. We say that w is a *repeating substring* of S iff $\#occ_S(w) \geq 2$, and that w is a *unique substring* of S iff $\#occ_S(w) = 1$. Since any unique substring w of S occurs exactly once in S, we will sometimes identify w with its corresponding interval $[i, j]$ such that $w = S[i..j]$. We also say that interval $[i, j]$ is unique iff the corresponding $S[i..j]$ is a unique substring of S.

A unique substring $w = S[i..j]$ of S is said to be a *minimal unique substring* (*MUS*) iff any proper substring of w is a repeating substring, namely, $\#occ_S(S[i'..j']) \geq 2$ for any i'

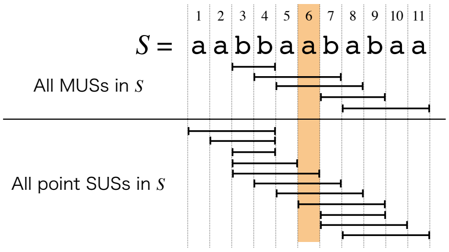

Figure 1 For string $S = \texttt{aabbaababaa}$, the set $\mathcal{M}_S = \{[3..4], [4..7], [5..8], [7..9], [8..11]\} = \{\texttt{bb}, \texttt{baab}, \texttt{aaba}, \texttt{bab}, \texttt{abaa}\}$ of all MUSs of S is shown in the upper part of the diagram. The set $\mathcal{PS}_S$ of all SUSs for all positions of string S is shown in the lower part of the diagram. For example, the intervals $[3..6] = \texttt{bbaa}$, $[4..7] = \texttt{baab}$, $[5..8] = \texttt{aaba}$, and $[6..9] = \texttt{abab}$ are SUSs for query position 6, where the first SUS $[3..6]$ is obtained by extending the right-end of MUS $[3..4]$ up to position 6, the second SUS $[4..7]$ and the third $[5..8]$ are MUSs of S, and the fourth SUS $[6..9]$ is obtained by extending the left-end of MUS $[8..11]$ up to position 6.

and j' with $i' \geq i$, $j' \leq j$, and $j' - i' < j - i$. Let $\mathcal{M}_S$ be the set of all MUSs in S, namely, $\mathcal{M}_S = \{[i, j] : S[i..j] \text{ is a MUS of } S\}$. The next lemma follows from the definition of MUSs.

▶ **Lemma 1** ([6]). *No element of $\mathcal{M}_S$ is nested in another element of $\mathcal{M}_S$, namely, any two MUSs $[i, j], [k, \ell] \in \mathcal{M}_S$ satisfy $[i, j] \not\subset [k, \ell]$ and $[k, \ell] \not\subset [i, j]$. Therefore, $0 < |\mathcal{M}_S| \leq n$.*

For any substring $S[i..j]$ and an interval $[s, t]$ in S, $S[i..j]$ is said to be a *shortest unique substring* (*SUS*) for interval $[s, t]$ iff

1. $S[i..j]$ is a unique substring of S,
2. $[s, t] \subset [i, j]$, and
3. $S[i'..j']$ is a repeating substring of S for any i', j' with $[s, t] \subset [i', j']$ and $j' - i' < j - i$.

In particular, a SUS for some interval $[p, p]$ of length 1 is said to be a SUS for position p and is sometimes referred to as a *point* SUS in S. Also, a SUS for some interval (including those of length 1) is sometimes referred to as an *interval* SUS in S.

Since any SUS $S[i..j]$ occurs in S exactly once, we will sometimes identify it with the interval $[i, j]$ which corresponds to its unique occurrence in S.

Clearly, if $[i, j]$ is unique, then $[i, j]$ is the only SUS for the interval $[i, j]$. For any interval $[i, j]$ with $i < j$, if $[i, j]$ is unique and there is no other interval $[s, t] \subset [i, j]$ for which $[i, j]$ is a SUS, then we say that $[i, j]$ is a *trivial* interval SUS. Also, we say that $[i, j]$ is a *non-trivial* interval SUS if $[i, j]$ is not a trivial SUS.

For any interval $[s, t] \subset [1, |S|]$, let $\mathsf{SUS}_S([s, t])$ denote the set of interval SUSs of S that contain query interval $[s, t]$, and $\mathcal{IS}_S$ the set of all non-trivial interval SUSs of S. Also, for any position $p \in [1, |S|]$, let $\mathsf{SUS}_S(p)$ denote the set of point SUSs of S that contain query position p, and $\mathcal{PS}_S$ the set of all point SUSs of S, namely, $\mathcal{PS}_S = \bigcup_{p=1}^{n} \mathsf{SUS}_S(p)$. Figure 1 shows examples of MUSs and SUSs.

Hu et al. [2] showed that it is possible to preprocess a given string S of length n in $O(n)$ time so that later, we can return all SUSs that contain a query interval $[s, t]$ in $O(k)$ time, where k is the number of such SUSs.

As is shown in Lemma 1, the number of MUSs in any string S of length n is bounded by n. In this paper, we show that the number of point SUSs in S is less than $1.5n$, more precisely, $|\mathcal{PS}_S| \leq (3n - 1)/2$. We will do so by first showing two different bounds on $|\mathcal{PS}_S|$ in terms of the number $|\mathcal{M}_S|$ of MUSs in the string S, and then merging these two results that lead to the claimed bound. Moreover, this bound is indeed tight, namely, we show a series of strings containing $(3n-1)/2$ SUSs. In addition, we show that the number of non-trivial SUSs in S is less than $2n$, namely, $|\mathcal{IS}_S| < 2n$. We also prove that there exists a string of length n which contains $(2-\varepsilon)n$ non-trivial SUSs for any small number $\varepsilon > 0$.

3 Bounds on the number of point SUSs

Here we show a tight bound for the maximum number of point SUSs in a string. In this section, whenever we speak of SUSs, we mean point SUSs (those for the point SUS problem).

3.1 Upperbound A

In this subsection, we show our first upperbound on the number of SUSs in a string S. In so doing, we define the subsets $\mathcal{LS}_S$, $\mathcal{MS}_S$, and $\mathcal{RS}_S$ of the set $\mathcal{PS}_S$ of all SUS of string S by

$$\begin{aligned}\mathcal{LS}_S &= \mathcal{PS}_S \cap \{[x,y] \notin \mathcal{M}_S : x < \exists i \leq y \ [i,y] \in \mathcal{M}_S\},\\ \mathcal{MS}_S &= \mathcal{PS}_S \cap \mathcal{M}_S, \text{ and}\\ \mathcal{RS}_S &= \mathcal{PS}_S \cap \{[x,y] \notin \mathcal{M}_S : x \leq \exists j < y \ [x,j] \in \mathcal{M}_S\}.\end{aligned}$$

Intuitively, $\mathcal{LS}_S$ is the set of SUSs of S which are *not* MUSs of S and can be obtained by extending the beginning positions of some MUSs to the left up to query positions, $\mathcal{MS}_S$ is the set of SUSs of S which are also MUSs of S, and $\mathcal{RS}_S$ is the set of SUSs of S which are *not* MUSs of S and can be obtained by extending the ending positions of some MUSs to the right up to query positions.

It follows from their definitions that $\mathcal{LS}_S \cap \mathcal{MS}_S = \phi$, $\mathcal{MS}_S \cap \mathcal{RS}_S = \phi$, $\mathcal{RS}_S \cap \mathcal{LS}_S = \phi$ and that $\mathcal{PS}_S = \mathcal{LS}_S \cup \mathcal{MS}_S \cup \mathcal{RS}_S$.

Figure 3 in the next subsection shows examples of $\mathcal{LS}_S$, $\mathcal{MS}_S$, and $\mathcal{RS}_S$ for string $S = \texttt{aabbaababaa}$. Also compare it with Figure 1 which shows $\mathcal{PS}_S$ for the same string S.

In the proof of the following theorem, we will evaluate the sizes of these three sets $\mathcal{LS}_S$, $\mathcal{MS}_S$, and $\mathcal{RS}_S$ separately.

▶ **Theorem 2.** *For any string S, $|\mathcal{PS}_S| \leq 2|S| - |\mathcal{M}_S|$.*

Proof. Let $n = |S|$ and $m = |\mathcal{M}_S|$. For any $1 \leq i \leq m$, let $[b_i, e_i]$ denote the MUS of S that has the ith smallest beginning position in $\mathcal{M}_S$.

It is clear that $|\mathcal{MS}_S| \leq m$. Note that the inequality is due to that fact that some MUS may not be a point SUS for any position in S (such a MUS is called *meaningless* in the literature [6]).

Next, we consider the size of $\mathcal{RS}_S$. By definition, for any $[x,y] \in \mathcal{RS}_S$, x is equal to the beginning position of a MUS of S. Therefore, we can bound $|\mathcal{RS}_S|$ by summing up the number of SUSs that begin with b_i for every $[b_i, e_i] \in \mathcal{M}_S$. For any $1 \leq i \leq m-1$, consider two adjacent MUSs $[b_i, e_i], [b_{i+1}, e_{i+1}] \in \mathcal{M}_S$. Recall that $b_i < b_{i+1}$. Then, for any $j \geq e_{i+1}$, the interval $[b_i, j]$ contains both MUSs $[b_i, e_i]$ and $[b_{i+1}, e_{i+1}]$. This implies that $[b_i, j] \notin \mathcal{PS}_S$ (see Figure 2), since otherwise both $[b_i, j]$ and $[b_{i+1}, j]$ are SUSs for position j, a contradiction. Thus, for any $[b_i, e_i] \in \mathcal{M}_S$ with $1 \leq i \leq m-1$, the number of SUSs that begin with b_i and belong to $\mathcal{RS}_S$ is at most $e_{i+1} - e_i - 1$. Also, the number of SUSs that begin with b_m and belong to $\mathcal{RS}_S$ is at most $n - e_m$. Consequently, we get $|\mathcal{RS}_S| = \sum_{i=1}^{m-1}(e_{i+1} - e_i - 1) + n - e_m = e_m - e_1 - (m-1) + n - e_m \leq n - m$.

A symmetric argument gives us the same bound for $|\mathcal{LS}_S|$, namely, $|\mathcal{LS}_S| \leq n - m$. Overall, we obtain $|\mathcal{PS}_S| = |\mathcal{LS}_S| + |\mathcal{MS}_S| + |\mathcal{RS}_S| \leq 2(n-m) + m = 2n - m$. ◀

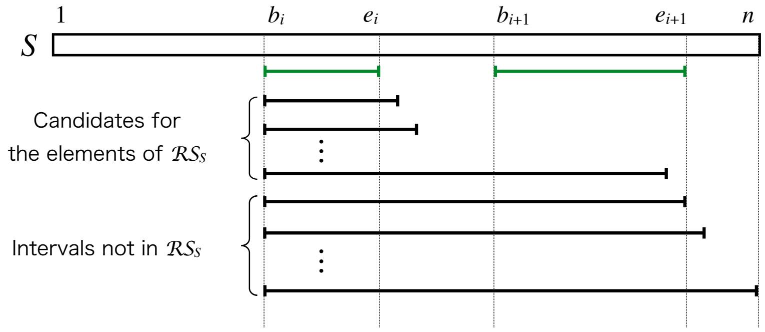
1
b_i
e_i
b_{i+1}
e_{i+1}
n
S
Candidates for the elements of $\mathcal{RS}_S$
Intervals not in $\mathcal{RS}_S$

Figure 2 Illustration for Theorem 2. Consider two adjacent MUSs $[b_i, e_i]$ and $[b_{i+1}, e_{i+1}]$ depicted as the two intervals on the top. For any $e_i < e < e_{i+1}$, $[b_i, e]$ can be an element of $\mathcal{RS}_S$. On the other hand, for any $e' \geq e_{i+1}$, $[b_i, e']$ can never be an element of $\mathcal{PS}_S$ since $[b_i, e']$ contains two distinct MUSs $[b_i, e_i]$ and $[b_i, e_{i+1}]$, and hence $[b_i, e']$ can never be an element of $\mathcal{RS}_S$ as well.

3.2 Upperbound B

In this subsection, we provide another upperbound on the size of $\mathcal{PS}_S$.

▶ **Theorem 3.** *For any string S, $|\mathcal{PS}_S| \leq |S| + |\mathcal{M}_S| - 1$.*

In order to show Theorem 3, we will use a function $f : \mathcal{PS}_S \to \{1, 2, \ldots, n\}$ and its inverse image $f^{-1} : \{1, 2, \ldots, n\} \to 2^{\mathcal{PS}_S}$. The next lemma is useful to define f and f^{-1}.

▶ **Lemma 4.** *For any string S and interval $[x, y]$ such that $1 \leq x \leq y \leq |S|$, if $[x, y] \in \mathcal{RS}_S$ then $[x, y] \in \mathsf{SUS}_S(y)$, and if $[x, y] \in \mathcal{LS}_S$ then $[x, y] \in \mathsf{SUS}_S(x)$.*

Proof. We first prove the former case. Assume on the contrary that some $[x, y] \in \mathcal{RS}_S$ satisfies $[x, y] \notin \mathsf{SUS}_S(y)$. This implies that there exists a position p in S such that $x \leq p < y$ and $[x, y] \in \mathsf{SUS}_S(p)$. In addition, since $[x, y] \in \mathcal{RS}_S$, there exists a position q such that $x \leq q < y$ and $[x, q] \in \mathcal{M}_S$. Let $z = \max\{p, q\}$. Then, $S[x..z]$ is a unique substring of S which is shorter than $S[x..y]$ and contains position p. However, this contradicts that $S[x..y]$ is a SUS for position p. Thus, if $[x, y] \in \mathcal{RS}_S$ then $[x, y] \in \mathsf{SUS}_S(y)$. The latter case is symmetric and thus can be shown similarly. ◀

We are now ready to define f:

$$f([x, y]) = \begin{cases} x & \text{if } [x, y] \in \mathcal{LS}_S \cup \mathcal{MS}_S, \\ y & \text{if } [x, y] \in \mathcal{RS}_S. \end{cases}$$

Intuitively, the function f charges a given interval $[x, y]$ to its beginning position x if $[x, y]$ is an element of $\mathcal{M}_S \cap \mathcal{PS}_S$ or if $[x, y]$ is an element of $\mathsf{SUS}_S(p)$ for some query position p which is obtained by extending the left-end of a MUS to the left up to p. On the other hand, it charges $[x, y]$ to its ending position y if the interval is an element of $\mathsf{SUS}_S(p)$ for some query position p which is obtained by extending the right-end of a MUS to the right up to p. Figure 3 shows examples for how the function f charges given interval $[x, y] \in \mathcal{PS}_S$.

We also define the inverse image f^{-1} of f as follows:

$$f^{-1}(u) = \{[x, y] \in \mathcal{PS}_S \ : \ f([x, y]) = u\}.$$

For positions u for which there is no element $[x, y]$ in $\mathcal{PS}_S$ satisfying $f([x, y]) = u$, let $f^{-1}(u) = \emptyset$. See also Figure 3 for examples of f^{-1}.

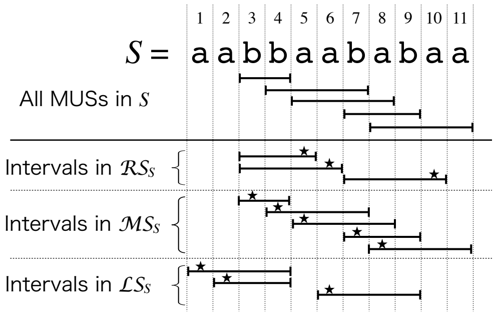
1 2 3 4 5 6 7 8 9 10 11
S = a a b b a a b a b a a
All MUSs in S
Intervals in $\mathcal{RS}_S$
Intervals in $\mathcal{MS}_S$
Intervals in $\mathcal{LS}_S$

Figure 3 Illustration for functions f and f^{-1} of string S = aabbaababaa. The upper part of this diagram shows all MUSs in S, and the lower part shows all SUSs for all positions in S. Each star shows the position to which the function f maps the corresponding interval. Here, $\mathcal{RS}_S = \{[3,5],[3,6],[7,10]\}$, $\mathcal{MS}_S = \{[3,4],[4,7],[5,8],[7,9],[8,11]\}$, and $\mathcal{LS}_S = \{[1,4],[2,4],[6,10]\}$. Hence, we have $f([3,5]) = 5$, $f([3,6]) = 6$, $f([7,10]) = 10$, $f([3,4]) = 3$, $f([4,7]) = 4$, $f([5,8]) = 5$, $f([7,9]) = 7$, $f([8,11]) = 8$, $f([1,4]) = 1$, $f([2,4]) = 2$, and $f([6,10]) = 6$. For the inverse image, f^{-1}, we have $f^{-1}(1) = \{[1,4]\}$, $f^{-1}(2) = \{[2,4]\}$, $f^{-1}(3) = \{[3,4]\}$, $f^{-1}(4) = \{[4,7]\}$, $f^{-1}(5) = \{[3,5],[5,8]\}$, $f^{-1}(6) = \{[3,6],[6,10]\}$, $f^{-1}(7) = \{[7,9]\}$, $f^{-1}(8) = \{[8,11]\}$, $f^{-1}(9) = f^{-1}(11) = \emptyset$, and $f^{-1}(10) = \{[7,10]\}$.

By the definition of f^{-1}, it is clear that $|\mathcal{PS}_S| = \sum_{u=1}^{|S|} |f^{-1}(u)|$. Hence, in what follows we analyze $|f^{-1}(u)|$ for all positions u in string S.

▶ **Lemma 5.** *For any string and position* $1 \le u \le |S|$, $|f^{-1}(u)| \le 2$.

Proof. Assume on the contrary that $|f^{-1}(u)| \ge 3$ for some position u in S. Let $[x_1, y_1]$, $[x_2, y_2]$ be any distinct elements of $f^{-1}(u)$. We firstly consider the following cases.

1. Case where $[x_1, y_1], [x_2, y_2] \in \mathcal{LS}_S$: It follows from the definition of f^{-1} that $f([x_1, y_1]) = f([x_2, y_2]) = u$, and it follows from the definition of f that $x_1 = x_2 = u$. Since $[x_1, y_1]$ and $[x_2, y_2]$ are distinct, $y_1 \neq y_2$. Assume w.l.o.g. that $y_1 < y_2$. Then, $[x_2, y_2] = [u, y_2]$ is a SUS for position u but it is longer than another SUS $[x_1, y_1] = [u, y_1]$ for position u, a contradiction.
2. Case where $[x_1, y_1], [x_2, y_2] \in \mathcal{MS}_S$: It follows from the definition of f^{-1} that $f([x_1, y_1]) = f([x_2, y_2]) = u$, and it follows from the definition of f that $x_1 = x_2 = u$. Since $[x_1, y_1]$ and $[x_2, y_2]$ are distinct, $y_1 \neq y_2$. Assume w.l.o.g. that $y_1 < y_2$. Then, $[x_2, y_2] = [u, y_2]$ is a MUS, but it contains another MUS $[x_1, y_1] = [u, y_1]$, a contradiction.
3. Case where $[x_1, y_1], [x_2, y_2] \in \mathcal{RS}_S$: This is symmetric to Case (1) and thus we can obtain a contradiction in a similar way.

Hence, none of the above three cases is possible, and thus the remaining possibility is the case where $|f^{-1}(u)| = 3$ and each element of $f^{-1}(u)$ belongs to a different subset of $\mathcal{PS}_S$, namely, $f^{-1}(u) = \{[x_1, y_1], [x_2, y_2], [x_3, y_3]\}$ for some $[x_1, y_1] \in \mathcal{LS}_S$, $[x_2, y_2] \in \mathcal{MS}_S$, and $[x_3, y_3] \in \mathcal{RS}_S$. It follows from the definition of f^{-1} that $f([x_1, y_1]) = f([x_2, y_2]) = u$, and it follows from the definition of f that $x_1 = x_2 = u$. Since $[x_1, y_1]$ and $[x_2, y_2]$ are distinct, $y_1 \neq y_2$. There are two sub-cases.

(i) If $y_1 < y_2$, then a MUS $[x_2, y_2] = [u, y_2]$ contains a shorter SUS $[x_1, y_1] = [u, y_1]$ for position u, a contradiction.

(ii) If $y_1 > y_2$, then a SUS $[x_1, y_1] = [u, y_1]$ for position u contains a shorter MUS $[x_2, y_2] = [u, y_2]$, a contradiction.

Hence, neither of the sub-cases is possible.

Overall, we conclude that $|f^{-1}(u)| \le 2$. ◀

By Lemma 5, for any position u in string S we have $|f^{-1}(u)| \leq 2$. Now let us consider any position u for which $|f^{-1}(u)| = 2$. We have the next lemma.

▶ **Lemma 6.** *For any position u in string S for which $|f^{-1}(u)| = 2$, let $f^{-1}(u) = \{[x_1, y_1], [x_2, y_2]\}$ and assume w.l.o.g. that $x_1 \leq x_2$. Then, $x_1 \neq x_2$, $[x_1, y_1] \in \mathcal{RS}_S$ and $[x_2, y_2] \in \mathcal{LS}_S \cup \mathcal{MS}_S$.*

Proof. Suppose $x_1 = x_2$ and assume w.l.o.g. that $y_1 < y_2$. Then, from the definition of f, we have that ($x_1 = u$ or $y_1 = u$) and ($x_2 = u$ or $y_2 = u$) and thus $x_1 = x_2 = u$. Since $[x_2, y_2] \in f^{-1}(u)$ is not a MUS since it includes $[x_1, y_1]$, it must be that $[x_2, y_2] \in \mathsf{SUS}_S(u)$. This is a contradiction, because there exists a shorter unique substring $[x_1, y_1]$ that contains u. Thus we have $x_1 \neq x_2$. Assume on the contrary that $[x_1, y_1] \in \mathcal{LS}_S \cup \mathcal{MS}_S$. Then, it follows from the definition of f that $f([x_1, y_1]) = x_1$. In addition, since $[x_1, y_1] \in f^{-1}(u)$, we have $u = x_1$. This implies that $u = x_1 < x_2$, but it contradicts that $[x_2, y_2] \in f^{-1}(u)$. Thus, $[x_1, y_1] \notin \mathcal{LS}_S \cup \mathcal{MS}_S$, namely, $[x_1, y_1] \in \mathcal{RS}_S$. Now, it follows from the arguments in the proof of Lemma 5 that $[x_2, y_2] \notin \mathcal{RS}_S$, and hence $[x_2, y_2] \in \mathcal{MS}_S \cup \mathcal{LS}_S$. ◀

Let $m = |\mathcal{M}_S|$, and $\mathcal{M}_S = \{[b_1, e_1], \ldots, [b_m, e_m]\}$. The next corollary immediately follows from Lemmas 4 and 6.

▶ **Corollary 7.** *For any position u in string S with $|f^{-1}(u)| = 2$, there exist two integers $1 \leq i < j \leq m$ such that $\mathsf{SUS}_S(u) = \{[b_i, u], [u, e_j]\}$.*

For any position u in string S before b_1 or after b_m, we have the next lemma.

▶ **Lemma 8.** *For any position u in string S s.t. $1 \leq u \leq b_1$ or $b_m < u \leq n$, $|f^{-1}(u)| \leq 1$.*

Proof. Assume on the contrary that $|f^{-1}(u)| = 2$ for some $1 \leq u \leq b_1$. By Lemma 6, there exists $[x, y] \in f^{-1}(u)$ such that $[x, y] \in \mathcal{RS}_S$. By the definitions of f and f^{-1}, we have $y = u$. Also, by the definition of $\mathcal{RS}_S$, there exists a position $e < y$ in S such that $[x, e] \in \mathcal{M}_S$. Now we have $x \leq e < y = u \leq b_1$, however, this contradicts that b_1 is the beginning position of the first (leftmost) MUS in $\mathcal{M}_S$. Thus $|f^{-1}(u)| \leq 1$ for any $1 \leq u \leq b_1$.

Assume on the contrary that $|f^{-1}(u)| = 2$ for some $b_m < u \leq n$. By Lemma 6, there exists $[x', y'] \in f^{-1}(u)$ such that $[x', y'] \in \mathcal{MS}_S \cup \mathcal{LS}_S$. By the definition of f and f^{-1}, we have $x' = u$. There are two cases to consider:

- If $[x', y'] \in \mathcal{MS}_S$, then $[x', y'] \in \mathcal{M}_S$. Thus $x' = u > b_m$ is the beginning position of a MUS in $\mathcal{M}_S$, however, this contradicts that b_m is the beginning position of the last (rightmost) MUS in $\mathcal{M}_S$.
- If $[x', y'] \in \mathcal{LS}_S$, then by the definition of $\mathcal{LS}_S$ there exists a position $b > x'$ such that $[b, y'] \in \mathcal{M}_S$. Now we have $b > x' = u > b_m$, however, this contradicts that b_m is the beginning position of the last (rightmost) MUS in $\mathcal{M}_S$.

Consequently, $|f^{-1}(u)| \leq 1$ for any $b_m < u \leq n$. ◀

▶ **Lemma 9.** *For any non-empty string S, let $U = \{u : |f^{-1}(u)| = 2\}$. Then, $|U| \leq |\mathcal{M}_S| - 1$.*

Proof. Let $n = |S|$ and $m = |\mathcal{M}_S|$. Recall that for any $1 \leq i \leq m$, $[b_i, e_i]$ denotes the ith element of $\mathcal{M}_S$.

Let $B = \{b_i \ : \ 1 \leq i \leq m-1\}$. We define function $g : U \to B$ as $g(u) = \max\{b < u \ : \ b \in B\}$. By the definition of U and Lemma 8, any position $u \in U$ satisfies $b_1 < u \leq b_m$. Therefore, $g(u)$ is well-defined for any position $u \in U$, and $g(u)$ returns the predecessor of u in the set B. It is clear that $|B| = m - 1$. Thus, if g is an injection, then we immediately obtain the claimed bound $|U| \leq |B| = m - 1$.

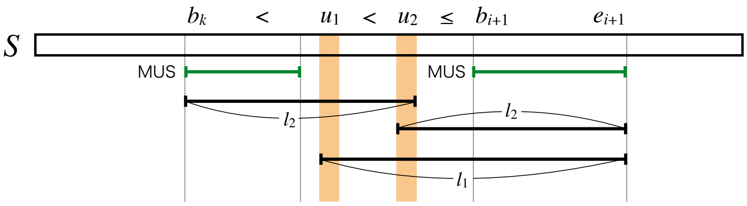

Figure 4 Illustration for Lemma 9. The two intervals show two MUSs $[b_k, e_k]$, $[b_{i+1}, e_{i+1}] \in \mathcal{M}_S$, where $b_k \leq b_i$. Both $[b_k, u_2]$ and $[u_2, b_{i+1}]$ are SUSs for position u_2, and $[u_1, e_{i+1}]$ is a SUS for position u_1. Since $u_1 < u_2$, it holds that $l_1 > l_2$, where l_1 and l_2 are the lengths of SUSs for positions u_1 and u_2, respectively. Then, the interval $[b_k, u_2]$ of length l_2 contains position u_1 and $S[b_k..u_2]$ is a unique substring of S. However, this contradicts that l_1 is the length of each SUS for position u_1.

In what follows, we show that g is indeed an injection. Assume on the contrary that g is not an injection. Let u_1 and u_2 be elements in U such that $u_1 < u_2$ and $g(u_1) = g(u_2)$. Let $b_i \in B$ such that $b_i = g(u_1) = g(u_2)$. Then, by the definition of g, we have $b_i < u_1 < u_2 \leq b_{i+1}$. See Figure 4 for illustration.

Let l_1 and l_2 be the lengths of the SUSs for positions u_1 and u_2, respectively. Since $|f^{-1}(u_2)| = 2$, it follows from Corollary 7 that there exists $b_k \in B$ such that $b_k \leq b_i$ and $\mathsf{SUS}_S(u_2) = \{[b_k, u_2], [u_2, e_{i+1}]\}$. This implies $l_2 = u_2 - b_k + 1 = e_{i+1} - u_2 + 1$. On the other hand, since $|f^{-1}(u_1)| = 2$, it follows from Corollary 7 that $[u_1, e_{i+1}] \in \mathsf{SUS}_S(u_1)$, which implies $l_1 = e_{i+1} - u_1 + 1$. Since $u_1 < u_2$, we have $l_1 > l_2$.

Now focus on a SUS $[b_k, u_2]$ for position u_2. Since $b_k \leq b_i < u_1 < u_2$, $[b_k, u_2]$ contains u_1. However, $[b_k, u_2]$ is a SUS for position u_2 and is of length $l_2 < l_1$. This contradicts that $[u_1, e_{i+1}]$ of length l_1 is each SUS for position u_1. Hence g is an injection. ◀

We are ready to prove the main result of this subsection, Theorem 3.

Proof. Let $n = |S|$, $m = |\mathcal{M}_S|$, $U = \{u : |f^{-1}(u)| = 2\}$, and $V = \{1, \cdots, n\} \setminus U$. It is clear that $|U| + |V| = n$. By Lemma 5, $V = \{u : |f^{-1}(u)| \leq 1\}$. Also, by Lemma 9, $|U| \leq m - 1$. Recall that $|\mathcal{PS}_S| = \sum_{u=1}^{n} |f^{-1}(u)|$. Putting all together, we obtain $|\mathcal{PS}_S| = \sum_{u=1}^{n} |f^{-1}(u)| \leq |V| + 2|U| = n + |U| \leq n + m - 1$. ◀

3.3 Matching upper and lower bounds

We are ready to show the main result of this paper.

▶ **Theorem 10.** *For any non-empty string S, $|\mathcal{PS}_S| \leq (3|S| - 1)/2$. This bound is tight, namely, for any odd $n \geq 5$ there exists a string T of length n s.t. $|\mathcal{PS}_T| = (3n - 1)/2$.*

Proof. By Theorem 2, we have $|\mathcal{M}_S| \leq 2|S| - |\mathcal{PS}_S|$. Also, by Theorem 3, we have $|\mathcal{PS}_S| - |S| + 1 \leq |\mathcal{M}_S|$. Thus $|\mathcal{PS}_S| - |S| + 1 \leq 2|S| - |\mathcal{PS}_S|$, which immediately leads to the claimed bound $|\mathcal{PS}_S| \leq (3|S| - 1)/2$.

We show that the above upperbound is indeed tight. For any odd number $n = 2k - 1 \geq 5$, consider string $T = a_1 x a_2 x \cdots a_{k-1} x a_k$, where $a_1, \ldots, a_k, x \in \Sigma$, $a_i \neq a_j$ for all $1 \leq i \neq j \leq k$, and $x \neq a_i$ for all $1 \leq i \leq k$. For any $1 \leq i \leq k$, $T[2i - 1] = a_i$ is a unique substring of T, and thus $[2i - 1, 2i - 1] \in \mathsf{SUS}_T(2i - 1)$. Also, for any $1 \leq i \leq k - 1$, $T[2i] = x$ is a repeating substring of T while $T[2i - 1..2i] = a_i x$ and $T[2i..2i + 1] = x a_{i+1}$ are unique substrings of T. This implies that $[2i - 1, 2i], [2i, 2i + 1] \in \mathsf{SUS}_T(2i)$. Hence, we have $|\mathcal{PS}_T| = k + 2(k - 1) = 3k - 2 = 3(n + 1)/2 - 2 = (3n - 1)/2$. ◀

3.4 Lower bound for fixed-size alphabet

The lowerbound of Theorem 10 is due to a series of strings over an alphabet of unbounded size. In this subsection, we fix the alphabet size σ and present a series of strings that contain many point SUSs.

▶ **Theorem 11.** *Let $n \geq 2$ and $2 \leq \sigma \leq (n+3)/2$. There exists a string T of length n over an alphabet of size σ such that $|\mathcal{PS}_T| = n + \sigma - 2$.*

Proof. Let $\Sigma = \{a_1, \cdots, a_{\sigma-1}, x\}$ and $T = a_1 x a_2 x \cdots a_{\sigma-1} x^{n-2\sigma+3}$. For any $1 \leq i \leq \sigma - 1$, $T[2i-1] = a_i$ is a unique substring of T, and thus $[2i-1, 2i-1] \in \mathsf{SUS}_T(2i-1)$. For any $1 \leq j \leq \sigma - 2$, $T[2j] = x$ is a repeating substring of T while $T[2j-1..2j] = a_j x$ and $T[2j..2j+1] = x a_{j+1}$ are unique substrings of T. This implies that $[2j-1, 2j], [2j, 2j+1] \in \mathsf{SUS}_T(2j)$. For any $2\sigma - 2 \leq k \leq n-1$, $T[2\sigma-2..k] = x^{k-2\sigma+3}$ is a repeating substring of T while $T[2\sigma-1..k] = a_{\sigma-1} x^{k-2\sigma+3}$ is a unique substrings of T. This implies that $[2\sigma-1, k] \in \mathsf{SUS}_T(k)$. Also, $T[2\sigma-1..n] = x^{n-2\sigma+2}$ is a repeating substring of T and $T[2\sigma-2..n] = x^{n-2\sigma+3}$ is a unique substring of T, and thus $[2\sigma-2..n] \in \mathsf{SUS}_T(n)$. Summing up all the point SUSs above, we obtain $|\mathcal{PS}_T| = \sigma - 1 + 2(\sigma-2) + n - 2\sigma + 2 + 1 = n + \sigma - 2$. ◀

4 Bounds on the number of interval SUSs

In this section, we show almost tight bounds for the maximum number of non-trivial interval SUSs $\mathcal{IS}_S$ of a string S. The following upper bound for $|\mathcal{IS}_S|$ can be obtained in an analogous way to Theorem 2.

▶ **Lemma 12.** *For any non-empty string S, $|\mathcal{IS}_S| \leq 2|S| - |\mathcal{M}_S|$.*

We also have the following lower bound for $|\mathcal{IS}_S|$.

▶ **Lemma 13.** *For any $\varepsilon > 0$, there exists a string T of length n such that $|\mathcal{IS}_T| > (2-\varepsilon)n$.*

Proof. Let $x = \lceil 3/(2\varepsilon) \rceil$, $T = c_1 a^x c_2 a^x c_3$ and $n = |T| = 2x + 3$. Clearly, c_1, c_2 and c_3 are MUSs of T and are in $\mathcal{IS}_T$. For all $2 \leq i \leq x+1$, $T[1..i]$ and $T[i..x+2]$ are unique substrings of T, and $T[2..i]$ and $T[i..x+1]$ are repeating substrings of T. This implies $T[1..i] \in \mathsf{SUS}_S([2, i])$ and $T[i..x+2] \in \mathsf{SUS}_S([i, x+1])$. Similarly, for all $x + 3 \leq j \leq 2x + 2$, $T[x+2..j] \in \mathsf{SUS}_S([x+3, j])$ and $T[j..2x+3] \in \mathsf{SUS}_S([j, 2x+2])$. Then, we have $|\mathcal{IS}_T| = 4x + 3$. Hence, $|\mathcal{IS}_T| - (2-\varepsilon)n = 4x + 3 - (2-\varepsilon)(2x+3) = 2\varepsilon x + 3\varepsilon - 3 = 2\varepsilon\lceil 3/(2\varepsilon) \rceil + 3\varepsilon - 3 \geq 3\varepsilon > 0$. ◀

As is shown in the following theorem, the number of non-trivial interval SUSs contained in the string T of Lemma 13 "almost coincides" with the upper bound of Lemma. Namely:

▶ **Theorem 14.** *For any $\varepsilon > 0$, there is a string T such that $(2|T| - |\mathcal{M}_T|) - (2-\varepsilon)|T| \leq 5\varepsilon$.*

Proof. For any $\varepsilon > 0$, consider the string T of Lemma 13. We remark that T contains 3 MUSs, namely, $|\mathcal{M}_T| = 3$. Hence, we obtain $(2|T| - |\mathcal{M}_T|) - (2-\varepsilon)|T| = \varepsilon|T| - |\mathcal{M}_T| = \varepsilon|T| - 3 = \varepsilon(2\lceil 3/(2\varepsilon) \rceil + 3) - 3 = 2\varepsilon\lceil 3/(2\varepsilon) \rceil + 3\varepsilon - 3 \leq 2\varepsilon(3/(2\varepsilon) + 1) + 3\varepsilon - 3 = 5\varepsilon \to 0 \, (\varepsilon \to 0)$. ◀

5 Conclusions and open questions

In this paper, we presented matching upper and lower bounds for the maximum number of SUSs for the point SUS problem. Namely, we proved that any string of length n can contain at most $(3n-1)/2$ SUSs for the point SUS problem, and showed that this bound is tight by giving a string of length n containing $(3n-1)/2$ SUSs. For a fixed alphabet size σ, we also presented a string of length n containing $n + \sigma - 2$ SUSs. Moreover, we showed that any string of length n which contains m MUSs can have at most $2n - m$ non-trivial interval SUSs, and that for any $\varepsilon > 0$ there is a string of length n which contains $(2 - \varepsilon)n$ non-trivial interval SUSs.

An interesting future work is to show a non-trivial upper bound of the maximum number of point SUSs for a fixed alphabet size σ. We conjecture that the tight upper bound matches our lower bound $n + \sigma - 2$. Another future work is to close the small gap between the upper and lower bounds on the maximum number of non-trivial interval SUSs shown in Theorem 14.

References

1 Wing-Kai Hon, Sharma V. Thankachan, and Bojian Xu. An in-place framework for exact and approximate shortest unique substring queries. In Khaled M. Elbassioni and Kazuhisa Makino, editors, *Proceedings of the 26th International Symposium on Algorithms and Computation (ISAAC 2015)*, volume 9472 of *LNCS*, pages 755–767. Springer, 2015. doi:10.1007/978-3-662-48971-0_63.

2 Xiaocheng Hu, Jian Pei, and Yufei Tao. Shortest unique queries on strings. In Edleno Silva de Moura and Maxime Crochemore, editors, *Proceedings of the 21st International Symposium on String Processing and Information Retrieval (SPIRE 2014)*, volume 8799 of *LNCS*, pages 161–172. Springer, 2014. doi:10.1007/978-3-319-11918-2_16.

3 Atalay Mert Ileri, M. Oguzhan Külekci, and Bojian Xu. Shortest unique substring query revisited. In Alexander S. Kulikov, Sergei O. Kuznetsov, and Pavel A. Pevzner, editors, *Proceedings of the 25th Annual Symposium on Combinatorial Pattern Matching (CPM 2014)*, volume 8486 of *LNCS*, pages 172–181. Springer, 2014. doi:10.1007/978-3-319-07566-2_18.

4 Takuya Mieno, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Shortest unique substring queries on run-length encoded strings. In Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier, editors, *Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, volume 58 of *LIPIcs*, pages 69:1–69:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.MFCS.2016.69.

5 Jian Pei, Wush Chi-Hsuan Wu, and Mi-Yen Yeh. On shortest unique substring queries. In Christian S. Jensen, Christopher M. Jermaine, and Xiaofang Zhou, editors, *Proceedings of the 29th IEEE International Conference on Data Engineering (ICDE 2013)*, pages 937–948. IEEE Computer Society, 2013. doi:10.1109/ICDE.2013.6544887.

6 Kazuya Tsuruta, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Shortest unique substrings queries in optimal time. In Viliam Geffert, Bart Preneel, Branislav Rovan, Julius Stuller, and A Min Tjoa, editors, *Proceedings of the 40th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2014)*, volume 8327 of *LNCS*, pages 503–513. Springer, 2014. doi:10.1007/978-3-319-04298-5_44.

7 Bojian Xu. On stabbing queries for generalized longest repeat. In Jun Huan, Satoru Miyano, Amarda Shehu, Xiaohua Tony Hu, Bin Ma, Sanguthevar Rajasekaran, Vijay K. Gombar, Matthieu-P. Schapranow, Illhoi Yoo, Jiayu Zhou, Brian Chen, Vinay Pai, and Brian G. Pierce, editors, *Proceedings of the 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2015)*, pages 523–530. IEEE Computer Society, 2015. doi:10.1109/BIBM.2015.7359738.