Greedy Conjecture for Strings of Length 4

Alexander S. Kulikov[1(✉)], Sergey Savinov[2], and Evgeniy Sluzhaev[1,2]

[1] St. Petersburg Department of Steklov Institute of Mathematics, Saint Petersburg, Russia
kulikov@logic.pdmi.ras.ru
[2] St. Petersburg Academic University, Saint Petersburg, Russia

Abstract. In this short note, we prove that the greedy conjecture for the shortest common superstring problem is true for strings of length 4.

1 Introduction

In the shortest common superstring (SCS) problem one is given a set $\mathcal{S} = \{s_1, \ldots, s_n\}$ of n strings and the goal is to find a shortest string s such that each s_i is a substring of s. This is a well-known problem having applications in such areas as genome assembly and data compression.

The problem is known to be NP-hard [10] (even if the input strings have length 3 or if the alphabet is binary [2]) and APX-hard [12]. The fastest known exact solutions just reduce the problem to the Travelling salesman problem and have running time $(\sum_{i=1}^{n} |s_i|)^{O(1)} 2^n$ [1,6–8]. The currently best known approximation ratio is $2\frac{11}{23}$ [11]. Better upper bounds are known for special cases when input strings have bounded length [4,5]. A recent survey of known results (both practical and theoretical) is given in [3].

The well known greedy conjecture states that the following extremely simple greedy algorithm has approximation ratio 2 [14]: find two strings with longest mutual overlap and merge them into one string, repeat the process till only one string is left. This intriguing conjecture is open for more than 25 years already. There is a partial progress however: it is known that the conjecture is true for some orders in which the input strings are merged by the greedy algorithm [9,15].

In this short note, we consider another special case. We prove that the greedy conjecture is true if the input strings have length 4. (While for strings of length 3 the conjecture follows from the fact that the greedy algorithm achieves 2-approximation of the compression measure [13].) We do this by a careful analysis of possible overlaps produced by the greedy algorithm.

2 Preliminaries

An *overlap* $\mathrm{ov}(a, b)$ of two strings a and b is defined as the longest suffix of a which is also a prefix of b.

Let $\mathcal{S} = \{s_1, \ldots, s_n\}$ be a set of pairwise different 4-strings where by an r-string we denote just a string of length exactly r. Denote by s^{opt} and s^{gr} an

F. Cicalese et al. (Eds.): CPM 2015, LNCS 9133, pp. 307–315, 2015.
DOI: 10.1007/978-3-319-19929-0_26

optimal solution and a greedy solution for $\mathcal{S}$, respectively. Our goal is thus to show that

$$|s^{\mathrm{gr}}| \leq 2 \cdot |s^{\mathrm{opt}}| \,. \tag{1}$$

For technical reasons, we assume in this paper that in case of ties the greedy algorithm prefers strings of the form $\mathtt{aaaa}$ for $\mathtt{a} \in \Sigma$.

Let $\pi = (\pi_1, \ldots, \pi_n)$ be a permutation of $\{1, \ldots, n\}$. By overlapping n input strings in this particular order one gets a superstring of length

$$\sum_{i=1}^{n} |s_i| - \sum_{i=1}^{n-1} |\operatorname{ov}(s_{\pi_i}, s_{\pi_{i+1}})| \,. \tag{2}$$

The second term in the expression above is called a *compression* of $\mathcal{S}$ with respect to π. Thus, an equivalent reformulation of SCS is the following: find an order of n input strings that maximizes the compression. By c^{opt} and c^{gr} we denote the compression of the optimal solution s^{opt} and the greedy solution s^{gr}, respectively. By combining (1) with (2) we get an equivalent reformulation of what we need to prove:

$$4n - c^{\mathrm{gr}} \leq 2 \cdot (4n - c^{\mathrm{opt}}) \,. \tag{3}$$

For a string t of length at most 3, let $\#^{\mathrm{opt}}(t)$ and $\#^{\mathrm{gr}}(t)$ be the number of overlaps that are equal to t in s^{opt} and s^{gr}, respectively. Similarly, let $\#_i^{\mathrm{opt}}$ and $\#_i^{\mathrm{gr}}$ be the number of overlaps of length exactly i. Then (3) is equivalent to

$$4n - \#_1^{\mathrm{gr}} - 2\#_2^{\mathrm{gr}} - 3\#_3^{\mathrm{gr}} \leq 2 \cdot (4n - \#_1^{\mathrm{opt}} - 2\#_2^{\mathrm{opt}} - 3\#_3^{\mathrm{opt}}) \tag{4}$$

or

$$2\#_1^{\mathrm{opt}} + 4\#_2^{\mathrm{opt}} + 6\#_3^{\mathrm{opt}} \leq 4n + \#_1^{\mathrm{gr}} + 2\#_2^{\mathrm{gr}} + 3\#_3^{\mathrm{gr}} \,. \tag{5}$$

Since $\#_1^{\mathrm{opt}} + \#_2^{\mathrm{opt}} + \#_3^{\mathrm{opt}} \leq n$ it suffices to prove that

$$2\#_3^{\mathrm{opt}} \leq 3\#_3^{\mathrm{gr}} + 2\#_2^{\mathrm{gr}} + \#_1^{\mathrm{gr}} \,. \tag{6}$$

Let $\mathcal{S}_3^{\mathrm{gr}}$ be the set of strings at the point of time when the greedy algorithm already merged all pairs of strings whose overlap is 3 and there is no more overlaps of length 3 left. In the following lemma we show that the number of overlaps equal to a 3-string t in the greedy solution cannot be much smaller than that of the optimal solution.

Lemma 1. *For any 3-string t, $\#^{\mathrm{gr}}(t) \geq \#^{\mathrm{opt}}(t) - 1$. Moreover, if $\#^{\mathrm{gr}}(t) = \#^{\mathrm{opt}}(t) - 1$ then $\mathcal{S}_3^{\mathrm{gr}}$ contains a string with prefix t and suffix t.*

Proof. Assume, for the sake of contradiction, that $\#^{\mathrm{gr}}(t) \leq \#^{\mathrm{opt}}(t) - 2$. The optimal solution contains $\#^{\mathrm{opt}}(t)$ overlaps equal to t and hence among the input n strings there are at least $\#^{\mathrm{opt}}(t)$ strings whose prefix is t and at least $\#^{\mathrm{opt}}(t)$ strings whose suffix is t. Now consider the set $\mathcal{S}_3^{\mathrm{gr}}$. Since $\#^{\mathrm{gr}}(t) \leq \#^{\mathrm{opt}}(t) - 2$, we conclude that $\mathcal{S}_3^{\mathrm{gr}}$ contains at least two strings whose suffix is t and at least two strings whose prefix is t. Hence there are two *different* strings in this set whose overlap is t which contradicts to the fact that there are no more 3-overlaps. □

In the following the strings from $\mathcal{S}_3^{\mathrm{gr}}$ are called *blocks*. For a 3-string t, we say that a block is t-*bad* if its suffix and its prefix are equal to t and moreover $\#^{\mathrm{gr}}(t) = \#^{\mathrm{opt}}(t) - 1$. We call a block *bad* if it is t-bad for a 3-string t and *good* otherwise. Let $\#_{\mathrm{bad}}$ and $\#_{\mathrm{good}}$ be the number of overlaps in all bad and good blocks, respectively. Then clearly $\#_{\mathrm{bad}} + \#_{\mathrm{good}} = \#_3^{\mathrm{gr}}$ (recall that all the overlaps inside the blocks have length 3).

Note that if there are no bad blocks then already Lemma 1 is sufficient to prove (6): in this case, $\#^{\mathrm{gr}}(t) \geq \#^{\mathrm{opt}}(t)$ and therefore $\#_3^{\mathrm{gr}} \geq \#_3^{\mathrm{opt}}$.

Next, we consider bad blocks of fixed length: for a 3-string t, let

$$\chi_{=i}(t) = [S_3^{gr} \text{contains a } t\text{−}\textit{bad block of length exactly } i].$$

(throughout the paper, we use the standard Iverson brackets: $[P]$ is equal to 1 if P is true and is equal to 0 otherwise). Further, let

$$\chi_{=i} = \sum_{|t|=3} \chi_{=i}(t)\,.$$

Functions $\chi_{>i}(t)$, $\chi_{\geq i}(t)$, $\chi_{>i}$, and $\chi_{\geq i}$ are defined in a similar fashion.

Note that $\chi_{=4} = 0$. Indeed a bad block of length 4 must have a form `aaaa`. Also, $\#^{\mathrm{opt}}(\texttt{aaaa}) > 0$ and hence $\mathcal{S}$ contains another string starting or ending with `aaa`. But then the greedy algorithm must merge these two strings (as it prefers strings of the form `aaaa`). Hence for any 3-string t, $\chi_{\geq 5}(t)$ is exactly the number of t-bad blocks.

Lemma 2. *For any 3-string t,*

$$\min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\} + \chi_{\geq 5}(t) = \#^{\mathrm{opt}}(t)\,.$$

Proof. Consider the following two cases:

1. $\#^{\mathrm{gr}}(t) \geq \#^{\mathrm{opt}}(t)$, then $\min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\} = \#^{\mathrm{opt}}(t)$ and $\chi_{\geq 5}(t) = 0$.
2. $\#^{\mathrm{gr}}(t) < \#^{\mathrm{opt}}(t)$, then by Lemma 1, $\min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\} = \#^{\mathrm{opt}}(t) - 1$. There is at least one block starting with t and ending with t. Moreover there cannot be two different such blocks as otherwise the greedy algorithm would merge them. Therefore, there is exactly one t-bad block, i.e., $\chi_{\geq 5}(t) = 1$. □

By summing up the equality from Lemma 2 over all strings t of length 3 we get the following corollary.

Corollary 1.

$$\sum_{|t|=3} \min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\} + \chi_{\geq 5} = \#_3^{\mathrm{opt}}.$$

Assume now that $\chi_{=5} = 0$. Then due to the fact that a bad block of length exactly i contains $i - 4$ overlaps we have that $2\chi_{>5} \leq \#_3^{\mathrm{gr}}$. By adding twice the $\sum_{|t|=3} \min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\}$ to both sides of this inequality and applying Corollary 1 we get

$$2\#_3^{\text{opt}} \leq \#_3^{\text{gr}} + 2\sum_{|t|=3} \min\{\#^{\text{gr}}(t), \#^{\text{opt}}(t)\} \leq 3\#_3^{\text{gr}},$$

which implies (6).

Hence the most tricky case is when there are bad blocks of length 5. The rest of the paper is devoted to the analysis of this case. Note that such blocks have the form ababa (for different letters $\mathtt{a}, \mathtt{b} \in \Sigma$) and therefore these are aba-bad blocks. To analyze such blocks carefully we introduce the following definitions. For a 3-string t and $1 \leq i \leq 5$, $B_i(t) = 0$ if either t is not of the form aba, or t is of the form aba and there is no block ababa. In the remaining case (i.e., t is of the form aba and there is a block ababa) B_i's are defined as follows:

$B_1(\mathtt{aba}) = [\#^{\text{gr}}(\mathtt{bab}) > \#^{\text{opt}}(\mathtt{bab})]$,
$B_2(\mathtt{aba}) = [$there exists a block with prefix ba or suffix ab$]$,
$B_3(\mathtt{aba}) = [$there exists a block except ababa with prefix ab or suffix ba$]$,
$B_4(\mathtt{aba}) = [$there exists a good block of length at least 5 containing aba or bab as a proper substring$]$,
$B_5(\mathtt{aba}) = [B_2(\mathtt{aba}) = 0$ and $B_3(\mathtt{aba}) = 0$ and there exists a bad block of length at least 7 containing aba or bab as substring$]$.

Further, let for $1 \leq i \leq 5$, $B_i = \sum_{|t|=3} B_i(t)$.

Now we show B_i's provide an upper bound for the number of bad blocks of length exactly 5.

Lemma 3. $\chi_{=5} \leq \sum_{i=1}^{5} B_i$.

Proof. Note that if 3-string t is not of the form aba then $\chi_{=5}(t) = 0$ so the string t contributes nothing to the left-hand side of the inequality. We now focus on 3-strings t of the form aba. It is sufficient to prove the following inequality:

$$\chi_{=5}(\mathtt{aba}) \leq \sum_{i=1}^{5} B_i(\mathtt{aba}) \tag{7}$$

Assume that $\chi_{=5}(\mathtt{aba}) = 1$ and $B_1(\mathtt{aba}) = 0$ as otherwise the inequality holds for trivial reasons. From $B_1(\mathtt{aba}) = 0$ and Lemma 1 we have that $\#^{\text{opt}}(\mathtt{bab}) - 1 \leq \#^{\text{gr}}(\mathtt{bab}) \leq \#^{\text{opt}}(\mathtt{bab})$. Since $\#^{\text{gr}}(\mathtt{bab}) > 0$ (because $\mathcal{S}_3^{\text{gr}}$ contains the block ababa by definition of $\chi_{=5}(\mathtt{aba})$) we have that $\#^{\text{opt}}(\mathtt{bab}) > 0$, i.e. the optimal solution has at least one overlap of the form bab. Depending of the location of this overlap in the optimal string we consider the following cases:

1. The overlap bab in the optimal solution is contained as a substring of ababa. Since $\#^{\text{opt}}(\mathtt{aba}) > 0$, $\mathcal{S}$ contains at least one string except abab and baba containing aba as substring.
2. The overlap bab in the optimal solution is not in ababa. Hence $\mathcal{S}$ contains at least one string except abab and baba containing bab.

So in both cases there exists a string in $\mathcal{S}$ except abab and baba that contains $t' = \mathtt{aba}$ or $t' = \mathtt{bab}$. This string is contained by some block $r \in \mathcal{S}_3^{\mathrm{gr}}$ and besides $r \neq \mathtt{ababa}$ and $r \neq \mathtt{babab}$. Consider the following cases:

1. r is a good block. Then $B_4(\mathtt{aba}) > 0$ if t' is a proper substring of r and $B_2(\mathtt{aba}) + B_3(\mathtt{aba}) > 0$ otherwise. Therefore (7) holds.
2. r is a bad block of length 5. Then this block has a form ababa or babab, a contradiction.
3. r is a bad block of length 6. If t' is a prefix or a suffix of r then $B_2(\mathtt{aba}) + B_3(\mathtt{aba}) > 0$. Otherwise either $r = \mathtt{r}_1\mathtt{t}'_1\mathtt{t}'_2\mathtt{t}'_1\mathtt{r}_5\mathtt{r}_6$ or $r = \mathtt{r}_1\mathtt{r}_2\mathtt{t}'_1\mathtt{t}'_2\mathtt{t}'_1\mathtt{r}_6$ where $t' = \mathtt{t}'_1\mathtt{t}'_2\mathtt{t}'_3$. Since r is a bad block either $\mathtt{t}'_1\mathtt{t}'_1\mathtt{t}'_2\mathtt{t}'_1\mathtt{t}'_1\mathtt{t}'_2$ or $\mathtt{t}'_2\mathtt{t}'_1\mathtt{t}'_1\mathtt{t}'_2\mathtt{t}'_1\mathtt{t}'_1$. Finally, since either $t' = \mathtt{aba}$ or $t' = \mathtt{bab}$ in both these cases r has a prefix or a suffix ab or ba. Then $B_2(\mathtt{aba}) + B_3(\mathtt{aba}) > 0$ and (7) holds.
4. r is a bad block of length at least 7. Then $B_5(\mathtt{aba}) > 0$ and (7) holds. □

3 The Proof of the Main Theorem

In this section we prove the main result of this note: we first state auxiliary lemmas providing upper bounds on B_i's, then show how these lemmas imply the main result of the paper, and then provide the proofs of all the lemmas.

Lemma 4. $B_1 + \sum\limits_{|t|=3} \min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\} \leq \#_3^{\mathrm{gr}}$.

Lemma 5. $B_2 \leq \#_2^{\mathrm{gr}}$.

Lemma 6. $B_3 \leq \#_1^{\mathrm{gr}} + \#_2^{\mathrm{gr}}$.

Lemma 7. $B_4 \leq \#_{\mathrm{good}}$.

Lemma 8. $B_5 + 2\chi_{>5} + \chi_{=5} \leq \#_{\mathrm{bad}}$.

Theorem 1. *The greedy algorithm for strings of length* 4 *that prefers strings of the form* aaaa *in case of ties is* 2*-approximate.*

Proof. By adding the inequalities from Lemmas 5–8 to twice the inequality from Lemma 4 and applying equality $\#_{\mathrm{bad}} + \#_{\mathrm{good}} = \#_3^{\mathrm{gr}}$ one gets

$$2B_1 + B_2 + B_3 + B_4 + B_5 + 2\chi_{>5} + \chi_{=5} + 2\sum_{|t|=3} \min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\} \leq 3\#_3^{\mathrm{gr}} + 2\#_2^{\mathrm{gr}} + \#_1^{\mathrm{gr}}.$$

By further adding the inequality from Lemma 3 we get

$$2\sum_{|t|=3} \min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\} + 2\chi_{>5} + 2\chi_{=5} + B_1 \leq 3\#_3^{\mathrm{gr}} + 2\#_2^{\mathrm{gr}} + \#_1^{\mathrm{gr}}.$$

Finally, applying Corollary 1 we get

$$2\#_3^{\mathrm{opt}} + B_1 \leq 3\#_3^{\mathrm{gr}} + 2\#_2^{\mathrm{gr}} + \#_1^{\mathrm{gr}}$$

which implies (6). □

Proof (of Lemma 4). We have

$$B_1 + \sum_{|t|=3} \min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\} = \sum_{|t|=3} (B_1(t) + \min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\})$$
$$= \sum_{t \neq \mathtt{aba}} (B_1(t) + \min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\}) + \sum_{\mathtt{a},\mathtt{b} \in \Sigma} (B_1(\mathtt{aba}) + B_1(\mathtt{bab})$$
$$+ \min\{\#^{\mathrm{gr}}(\mathtt{aba}), \#^{\mathrm{opt}}(\mathtt{aba})\} + \min\{\#^{\mathrm{gr}}(\mathtt{bab}), \#^{\mathrm{opt}}(\mathtt{bab})\})$$

To prove this lemma, we consider the following cases:

Case 1. If $t \neq \mathtt{aba}$ then $B_1(t) = 0$ and hence

$$B_1(t) + \min\{\#^{\mathrm{gr}}(t), \#^{\mathrm{opt}}(t)\} \leq \#^{\mathrm{gr}}(t)\,.$$

Case 2. If $t = \mathtt{aba}$ and $B_1(\mathtt{aba}) + B_1(\mathtt{bab}) = 0$ then

$$B_1(\mathtt{aba}) + B_1(\mathtt{bab}) + \min\{\#^{\mathrm{gr}}(\mathtt{aba}), \#^{\mathrm{opt}}(\mathtt{aba})\}$$
$$+ \min\{\#^{\mathrm{gr}}(\mathtt{bab}), \#^{\mathrm{opt}}(\mathtt{bab}))\} \leq \#^{\mathrm{gr}}(\mathtt{aba}) + \#^{\mathrm{gr}}(\mathtt{bab})$$

Case 3. If $t = \mathtt{aba}$ and $B_1(\mathtt{aba}) = 1$ then $B_1(\mathtt{bab}) = 0$ and, by definition of B_1, $\#^{\mathrm{gr}}(\mathtt{bab}) > \#^{\mathrm{opt}}(\mathtt{bab})$. Hence

$$B_1(\mathtt{aba}) + B_1(\mathtt{bab}) + \min\{\#^{\mathrm{gr}}(\mathtt{aba}), \#^{\mathrm{opt}}(\mathtt{aba})\} + \min\{\#^{\mathrm{gr}}(\mathtt{bab}), \#^{\mathrm{opt}}(\mathtt{bab})\}$$
$$= 1 + \min\{\#^{\mathrm{gr}}(\mathtt{aba}), \#^{\mathrm{opt}}(\mathtt{aba})\} + \min\{\#^{\mathrm{gr}}(\mathtt{bab}), \#^{\mathrm{opt}}(\mathtt{bab})\}$$
$$\leq 1 + \#^{\mathrm{gr}}(\mathtt{aba}) + \#^{\mathrm{opt}}(\mathtt{bab})$$
$$\leq 1 + \#^{\mathrm{gr}}(\mathtt{aba}) + \#^{\mathrm{gr}}(\mathtt{bab}) - 1 = \#^{\mathrm{gr}}(\mathtt{aba}) + \#^{\mathrm{gr}}(\mathtt{bab})$$

Case 4. If $t = \mathtt{aba}$ and $B_1(\mathtt{bab}) = 1$. This case is similar to Case 3. □

Proof (of Lemma 5). We show that $B_2 \leq \#^{\mathrm{gr}}_2$. If $B_2(t) > 0$ then t is of the form $\mathtt{aba}$ and there exists a block with prefix $\mathtt{ba}$ or suffix $\mathtt{ab}$. Since $B_2(t) > 0$ there exists a pair of blocks: $\mathtt{ababa}$ and a block with a 2-prefix $\mathtt{ba}$ or a 2-suffix $\mathtt{ab}$. Note that for different strings t these pairs of blocks do not intersect and cannot be merged with 2-overlaps because the sets $\{\mathtt{a}, \mathtt{b}\}$ are different. Note that at least one block in this pair must be merged with 2-overlap with some block otherwise this pair of blocks must be merged by the greedy algorithm. Thus $\sum_t B_2(t) < \#^{\mathrm{gr}}_2$

□

For Lemma 6 we need the following auxiliary definitions. Let $\mathrm{Pref}(\mathtt{a}, \mathtt{b}) = \emptyset$ if there is no block $\mathtt{ababa}$ and the set of blocks with prefix $\mathtt{ab}$ otherwise. Similarly, let $\mathrm{Suff}(\mathtt{a}, \mathtt{b}) = \emptyset$ if there is no block $\mathtt{ababa}$ and the set of blocks with suffix $\mathtt{ba}$ otherwise. Then it is easy to see that:

$$(\mathtt{a} \neq \mathtt{a}' \vee \mathtt{b} \neq \mathtt{b}') \Rightarrow (\mathrm{Pref}(\mathtt{a}, \mathtt{b}) \cap \mathrm{Pref}(\mathtt{a}', \mathtt{b}') = \emptyset \wedge \mathrm{Suff}(\mathtt{a}, \mathtt{b}) \cap \mathrm{Suff}(\mathtt{a}', \mathtt{b}') = \emptyset)$$

Let

$$\mathrm{Pref}(\mathtt{a}) = \bigcup_{\mathtt{b} \in \Sigma} \mathrm{Pref}(\mathtt{a}, \mathtt{b}) \text{ and } \mathrm{Suff}(\mathtt{a}) = \bigcup_{\mathtt{b} \in \Sigma} \mathrm{Suff}(\mathtt{a}, \mathtt{b})\,.$$

Lemma 9. *If* $\mathtt{a} \neq \mathtt{c}$ *then the set of* 1- *and* 2-*suffixes of strings from* $\mathrm{Suff}(\mathtt{a})$ *does not intersect the set of* 1- *and* 2-*prefixes of strings from* $\mathrm{Pref}(\mathtt{c})$.

Proof. All 1-suffixes of strings from $\mathrm{Suff}(\mathtt{a})$ are equal to $\mathtt{a}$ while all 1-prefixes of strings from $\mathrm{Pref}(\mathtt{c})$ are equal to $\mathtt{c}$, hence they do not intersect.

Assume that 2-suffix of $b_1 \in \mathrm{Suff}(a)$ equals to 2-prefix of block $b_2 \in \mathrm{Pref}(c)$. 2-suffix of block b_1 has the form $\mathtt{xa}$ and 2-prefix of b_2 has the form $\mathtt{cy}$ so $x = \mathtt{c}, y = \mathtt{a}$. Hence b_1 has form $\mathtt{acaca}$ and b_2 has form $\mathtt{cacac}$, a contradiction. □

Proof (of Lemma 6). $B_3(t) > 0$ only for $t = \mathtt{aba}$: $B_3 = \sum\limits_t B_3(t) = \sum\limits_{\mathtt{a}} \sum\limits_{\mathtt{b}} B_3(\mathtt{aba})$.

By Lemma 9 one can form sets X_1^a of 1-overlaps of strings from $\mathrm{Suff}(a)$ and $\mathrm{Pref}(a)$ counted in $\#_1^{\mathrm{gr}}$. The lemma guarantees that these sets are disjoint. Similarly we can form sets $X_2^{\mathtt{a}}$ from 2-overlaps of strings from $\mathrm{Suff}(\mathtt{a})$ and $\mathrm{Pref}(\mathtt{a})$. Hence

$$\sum_{\mathtt{a}} |X_1^{\mathtt{a}}| \le \#_1^{\mathrm{gr}} \text{ and } \sum_{\mathtt{a}} |X_2^{\mathtt{a}}| \le \#_2^{\mathrm{gr}}\,. \tag{8}$$

Since for each nonzero $\chi_{=5}(\mathtt{aba})$ there exists a block $\mathtt{ababa}$ we have, for each $\mathtt{a}$,

$$\sum_{\mathtt{b}} B_3(\mathtt{aba}) \le \min\{|\,\mathrm{Pref}(\mathtt{a})|, |\,\mathrm{Suff}(\mathtt{a})|\}\,. \tag{9}$$

Since for each block $\mathtt{ababa}$ with $B_3(\mathtt{aba}) > 0$ there exists by definition a string with prefix $\mathtt{ab}$ or suffix $\mathtt{ba}$, we have:

$$\sum_{\mathtt{b}} B_3(\mathtt{aba}) < \max\{|\,\mathrm{Pref}(\mathtt{a})|, |\,\mathrm{Suff}(\mathtt{a})|\}\,. \tag{10}$$

Assume that $|\,\mathrm{Pref}(\mathtt{a})| \le |\,\mathrm{Suff}(\mathtt{a})|$ (the opposite case is symmetric). Let us show that

$$|X_1^a| + |X_2^a| \ge \sum_b B_3(\mathtt{aba})\,. \tag{11}$$

For this, assume the contrary. It follows from (9) and (10) that

$$|X_1^{\mathtt{a}}| + |X_2^{\mathtt{a}}| \le |\,\mathrm{Pref}(\mathtt{a})| - 1 \text{ and } |X_1^{\mathtt{a}}| + |X_2^{\mathtt{a}}| \le |\,\mathrm{Suff}(\mathtt{a})| - 2\,.$$

Hence there exists at least one block from $\mathrm{Pref}(\mathtt{a})$ whose prefix is not used in overlaps and there exist at least two blocks from $\mathrm{Suff}(\mathtt{a})$ whose suffixes are not used in overlaps. But this prefix can be merged with one of these suffixes, a contradiction establishing (11).

Finally, by summing (11) for all $\mathtt{a}$ and applying (8) we get the required inequality:

$$\sum_{\mathtt{a}\in\Sigma} \sum_{\mathtt{b}\in\Sigma} B_3(\mathtt{aba}) \le \sum_{\mathtt{a}} (|X_1^{\mathtt{a}}| + |X_2^{\mathtt{a}}|) \le \#_1^{\mathrm{gr}} + \#_2^{\mathrm{gr}}\,.$$ □

Proof (of Lemma 7). If for some $\mathtt{a}, \mathtt{b} \in \Sigma$, $B_4(\mathtt{aba}) + B_4(\mathtt{bab}) = 1$, then either aba or bab is contained by a good block as a proper substring, so there exists at least one overlap by t in a good block. Hence

$$B_4 = \sum_{|t|=3} B_4(t) \leq \#_{\text{good}} .$$

□

Proof (of Lemma 8). Let $\#^i_{\text{bad}}$ be the number of overlaps in bad blocks of length i.

Let $B^i_5(t) = [i \geq 7 \wedge t = \mathtt{aba} \wedge B_2(t) = B_3(t) = 0 \wedge$ there exists a block ababa and a bad-block of length i which contains aba or bab as a proper substring]

By definition,

$$B_5(t) \leq \sum_{i \geq 7} B^i_5(t)$$

Since there are two 3-overlaps in bad blocks of length 6, $2\chi_{=6} = \#^6_{\text{bad}}$.

Consider bad blocks of length $i \geq 7$. Each such block contains $i - 4$ 3-overlaps. Note that overlaps aba or bab that are counted in $B^i_5(\mathtt{aba})$ cannot be neighbouring as otherwise B^i_5 would contain blocks ababa and babab (while this is only possible if the initial set $\mathcal{S}$ contains equal strings).

Let aba be the first overlap in a block. Then this block has prefix cab for $\mathtt{c} \in \Sigma$. Its suffix also equals cab since this is a bad block. But in this case $B_2(\mathtt{aba}) > 0$ and then $B_5(\mathtt{aba}) = 0$, a contradiction. A similar contradiction arises if aba is the last overlap in a block. Thus, for $i \geq 7$ we have:

$$B^i_5 = \sum_s B^i_5(s) \leq \chi_{>5} \cdot \left\lceil \frac{i-6}{2} \right\rceil \leq \chi_{>5} \cdot (i-6) .$$

Then

$$2\chi^i_{>5} + B^i_5 \leq 2\chi_{>5} + \chi_{>5} \cdot (i-6) = \chi_{>5} \cdot (i-4) \leq \#^i_{\text{bad}} .$$

Finally, we have:

$$2\chi_{>5} + \chi_{=5} + B_5 \leq \chi_{=5} + 2\chi_{=6} + \sum_{i \geq 7} (2\chi_{=i} + B^i_5)$$
$$\leq \#^5_{\text{bad}} + \#^6_{\text{bad}} + \sum_{i \geq 7} \#^i_{\text{bad}} = \#_{\text{bad}} .$$

□

4 Conclusion

We have proved that the greedy conjecture for the shortest common superstring problem is true for strings of length 4. Extending the proof to the case of 5-strings seems to be even more tedious. At the same time resolving such special cases does not seem to help to resolve the general case.

Acknowledgments. Research is partially supported by the Government of the Russian Federation (grant 14.Z50.31.0030) and Grant of the President of the Russian Federation (MK-6550.2015.1).

References

1. Bellman, R.: Dynamic programming treatment of the travelling salesman problem. J. ACM (J.ACM) **9**(1), 61–63 (1962)
2. Gallant, J., Maier, D., Storer, J.: On finding minimal length superstrings. J. Comput. Syst. Sci. **20**(1), 50–58 (1980)
3. Gevezes, T., Pitsoulis, L.: The shortest superstring problem. In: Rassias, T.M., Floudas, C.A., Butenko, S. (eds.) Optimization in Science and Engineering– In Honor of the 60th Birthday of Panos M. Pardalos, pp. 189–227. Springer, New York (2014)
4. Golovnev, A., Kulikov, A.S., Mihajlin, I.: Approximating shortest superstring problem using de bruijn graphs. In: Fischer, J., Sanders, P. (eds.) CPM 2013. LNCS, vol. 7922, pp. 120–129. Springer, Heidelberg (2013)
5. Golovnev, A., Kulikov, A.S., Mihajlin, I.: Solving SCS for bounded length strings in fewer than 2^n steps. Inf. Process. Lett. **114**(8), 421–425 (2014)
6. Held, M., Karp, R.M.: A dynamic programming approach to sequencing problems. J. Soc. Ind. Appl. Math. **10**(1), 196–210 (1962)
7. Karp, R.M.: Dynamic programming meets the principle of inclusion and exclusion. Oper. Res. Lett. **1**(2), 49–51 (1982)
8. Kohn, S., Gottlieb, A., Kohn, M.: A generating function approach to the traveling salesman problem. In: Proceedings of the 1977 annual conference. pp. 294–300. ACM (1977)
9. Laube, U., Weinard, M.: Conditional inequalities and the shortest common superstring problem. Int. J. Found. Comput. Sci. **17**(1), 247–247 (2006)
10. Maier, D., Storer, J.A.: A note on the complexity of the superstring problem. Princeton University Technical report 233 (1977)
11. Mucha, M.: Lyndon words and short superstrings. In: Proceedings of the TwentyFourth Annual ACM-SIAM Symposium on Discrete Algorithms. SODA 2013, Society for Industrial and Applied Mathematics (2013)
12. Ott, S.: Lower bounds for approximating shortest superstrings over an alphabet of size 2. In: Widmayer, P., Neyer, G., Eidenbenz, S. (eds.) WG 1999. LNCS, vol. 1665, pp. 55–64. Springer, Heidelberg (1999)
13. Tarhio, J., Ukkonen, E.: A greedy algorithm for constructing shortest common superstrings. In: Gruska, J., Rovan, B., Wiedermann, J. (eds.) MFCS 1986. LNCS, pp. 602–610. Springer, Heidelberg (1986)
14. Turner, J.: Approximation algorithms for the shortest common superstring problem. Inf. Comput. **83**(1), 1–20 (1989)
15. Weinard, M., Schnitger, G.: On the greedy superstring conjecture. In: Pandya, P.K., Radhakrishnan, J. (eds.) FSTTCS 2003. LNCS, vol. 2914, pp. 387–398. Springer, Heidelberg (2003)