Festschrift for

Bořivoj Melichar

August 30, 2012

Edited by Jan Holub, Bruce W. Watson and Jan Žďárek

Prague Stringology Club
http://www.stringology.org/

A Parameterized Formulation for the Maximum Number of Runs Problem*

Andrew Baker, Antoine Deza, and Frantisek Franek

Advanced Optimization Laboratory
Department of Computing and Software
McMaster University, Hamilton, Ontario, Canada
{bakerar2,deza,franek}@mcmaster.ca
http://optlab.cas.mcmaster.ca/

Dedicated to Professor Bořivoj Melichar on the occasion of his 70th birthday

Abstract. A parameterized approach to the problem of the maximum number of runs in a string was introduced by Deza and Franek. In the approach referred to as the *d-step approach*, in addition to the usual parameter the length of the string, the size of the string's alphabet is considered. The behaviour of the function $\rho_d(n)$, the maximum number of runs over all strings of length n with exactly d distinct symbols, can be handily expressed in the terms of properties of a table referred to as the $(d, n-d)$ *table* in which $\rho_d(n)$ is the entry at the dth row and $(n-d)$th column. The approach leads to a conjectured upper bound $\rho_d(n) \le n-d$ for $2 \le d \le n$. The parameterized formulation shows that the maximum within any column of the $(d, n-d)$ table is achieved on the main diagonal, i.e. for $n = 2d$, and motivates the investigation of the structural properties of the run-maximal strings of length n bounded by a constant times the size of the alphabet d. We show that $\rho_d(n) = \rho_{n-d}(2n-2d)$ for $2 \le d \le n \le 2d$, $\rho_d(2d) \le \rho_{d-1}(2d-1)+1$ for $d \ge 3$, $\rho_{d-1}(2d-1) = \rho_{d-2}(2d-2) = \rho_{d-3}(2d-3)$ for $d \ge 5$, and $\{\rho_d(n) \le n-d \text{ for } 2 \le d \le n\} \Leftrightarrow \{\rho_d(9d) \le 8d \text{ for } d \ge 2\}$. The results allow for an efficient computational verification of entries in the $(d, n-d)$ table for higher values of n and point to a plausible way of either proving the maximum number of runs conjecture by showing that possible counter-examples on the main diagonal would exhibit an impossible structure, or to discover an unexpected counter-example on the main diagonal of the $(d, n-d)$ table. This approach provides a purely analytical proof of $\rho_d(2d) = d$ for $d \le 15$ and, using the computational results of $\rho_2(d+2)$ for $d = 16, \ldots, 23$, a proof of $\rho_d(2d) = d$ for $d \le 23$.

Keywords: string, runs, maximum number of runs, parameterized approach, $(d, n-d)$ table

1 Foreword

The two first authors of this contribution have known of Bořislav Melichar's work since they ventured into the field of stringology a few years ago, while the third author has known him and his work in compilers for many years. Bořek's – as known to his friends – accomplishments include establishing a highly reputed research group, nurturing an impressive list of graduate students, and his pioneering and high-impact research work as an internationally recognized leader in the field. It is an equal honour and pleasure to dedicate to Bořek our work originally presented at the 2011 edition of the vibrant Prague Stringology Conference series.

* This work was supported by the *Natural Sciences and Engineering Research Council of Canada* and *MITACS*, and by the *Canada Research Chair program*, and made possible by the facilities of the *Shared Hierarchical Academic Research Computing Network* (http://www.sharcnet.ca/).

Festschrift for Bořivoj Melichar. J. Holub, B. W. Watson and J. Žďárek (Eds.) Published by Czech Technical University in Prague, Czech Republic

2 Introduction

The problem of determining the maximum number of runs in a string has a rich history and many researchers have contributed to the effort. The notion of a run is due to Main [17], the term itself was introduced in [13]. Kolpakov and Kucherov [14,15] showed that the function $\rho(n)$, the maximum number of runs over all strings of length n, is linear. Several papers dealt with lower and upper bounds or expected values for $\rho(n)$, see [3,4,5,9,10,11,12,18,19,20,21,22] and references therein.

The counting estimates leading to the best upper bounds [4,5] rely heavily on a computational approach and seem to reach a point where it gets highly challenging, bordering intractability, to verify the results or make further progress. A few researchers tried a structural approach, for instance [8,16].

A parameterized approach to the investigation of the structural aspects of run-maximal strings was introduced by Deza and Franek [6]. In addition to considering the length of the string they introduced the parameter d giving the function $\rho_d(n)$, the maximum number of runs over all strings of length n with exactly d distinct symbols. These values are presented in the so-called $(d, n-d)$ table, where the value of $\rho_d(n)$ is the entry at the row d and the column $n-d$. In Table 1, the entries for the first 10 rows and the first 10 columns are presented. Several properties of the table were presented in [6], the most important being the fact that $\rho_d(n) \leq n-d$ for $2 \leq d \leq n$ is equivalent with $\rho_d(2d) \leq d$ for $d \geq 2$. In other words, if the diagonal obeys the upper bound $n-d$, so do all the entries in the table everywhere. Though in the related literature, the *maximum number of runs conjecture* – or simply *runs conjecture* – refers to the hypothesis that $\rho(n) \leq n$, in this paper we will take it to be $\rho_d(n) \leq n-d$. Note that while the upper bound of n is not achieved for any known string, the $n-d$ bound is achieved for all pairs (d, n) satisfying $n-d \leq \min(23, d)$.

We discuss several additional properties of the $(d, n-d)$ table, the behaviour of the function $\rho_d(n)$ on or nearby the main diagonal, and discuss some structural properties of run-maximal strings on the main diagonal. The results allow for the extension of computational verification of the maximum number of runs conjecture to higher values of n and also indicate a viable approach to an analytical investigation of the conjecture by either showing a possible counter-example to the conjecture would have to exhibit an impossible structure, or exhibiting a counter-example on the main diagonal of the $(d, n-d)$ table and direct calculation of entries for smaller columns.

Let us remark, that although we believe with the majority of the researchers in the field that the conjecture is true and hence view the d-step approach as a possible tool to prove it, if a counter-example exists, there must be one on the main diagonal and we believe it will be easier to find there as the run-maximal strings of length being twice the size of the alphabet seem to exhibit a richer structure than general run-maximal strings. For example, all tractable run-maximal strings satisfying $n = 2d$ are, up to relabeling, unique. A counter-example would be in essence a quite striking result. The parameterized approach is inspired by a similar $(d, n-d)$ table used for investigating the Hirsch bound for the diameter of bounded polytopes. The associated Hirsch $(d, n-d)$ table exhibits similar property as the $(d, n-d)$ table considered in this paper. The Conjecture of Hirsch was recently disproved by Santos [23] by exhibiting a violation on the main diagonal with $d = 43$.

d \ $n-d$	1	2	3	4	5	6	7	8	9	10	11
1	**1**	*1*	1	1	1	1	1	1	1	1	.
2	1	**2**	*2*	3	4	5	5	6	7	8	.
3	1	2	**3**	*3*	4	5	6	6	7	8	.
4	1	2	3	**4**	*4*	5	6	7	7	8	.
5	1	2	3	4	**5**	*5*	6	7	8	8	.
6	1	2	3	4	5	**6**	*6*	7	8	9	.
7	1	2	3	4	5	6	**7**	*7*	8	9	.
8	1	2	3	4	5	6	7	**8**	*8*	9	.
9	1	2	3	4	5	6	7	8	**9**	*9*	.
10	1	2	3	4	5	6	7	8	9	**10**	.
11	.	.	.	.	.	.	.	.	.	.	.

Table 1. Values for $\rho_d(n)$ with $1 \le d \le 10$ and $1 \le n-d \le 10$. For more values, see [2]

3 Notation and Preliminaries

Throughout this paper, we refer to k-tuples: a symbol which occurs exactly k times in the string under consideration. Specially named k-tuples are the *singleton* (1-tuple), *pair* (2-tuple), *triple* (3-tuple), *quadruple* (4-tuple), and *quintuple* (5-tuple).

Definition 1. *A safe position in a string $\boldsymbol{x}$ is one which, when removed from $\boldsymbol{x}$, does not result in two runs being merged into one in the resulting new string.*

A safe position does not ensure that the number of runs will not change when that position is removed, only that no runs will be lost through being merged; runs may still be destroyed by having an essential symbol removed. Safe positions are important in that they may be removed from a string while only affecting the runs which contain them. When the position of a symbol is unambiguous, we may thus refer to a *safe symbol* rather than to its position – for instance we can talk about a safe singleton, or about the first member of a pair being safe, etc.

At various points we will need to relabel all occurrences of a symbol in a string or substring. Let $\boldsymbol{x}^a_b$ denote the string $\boldsymbol{x}$, in which all occurrences of a are replaced by b, and vice versa. $S_d(n)$ refers to the set of strings of length n with exactly d distinct symbols. For a string $\boldsymbol{x}$, $\mathcal{A}(\boldsymbol{x})$ denotes the alphabet of $\boldsymbol{x}$, while $r(\boldsymbol{x})$ denotes the number of runs of $\boldsymbol{x}$.

Lemma 2. *There exists a run-maximal string in $S_d(n)$ with no unsafe singletons for $2 \le d \le n$.*

Proof. Let $\boldsymbol{x}$ be a run-maximal string in $S_d(n)$. We will show that one of the following conditions must hold:

(*i*) $\boldsymbol{x}$ has no singletons, or
(*ii*) $\boldsymbol{x}$ has exactly one singleton which is safe, or
(*iii*) $\boldsymbol{x}$ has exactly one singleton which is unsafe, and there exists another run-maximal string $\boldsymbol{x'} \in S_d(n)$ where $\boldsymbol{x'}$ has no unsafe singletons, or
(*iv*) $\boldsymbol{x}$ has more than one singleton, all of which are safe.

Let $\boldsymbol{x}$ have some unsafe singletons.

First, consider the case that $\boldsymbol{x}$ has exactly one singleton, C, which is unsafe: $\boldsymbol{x} = \boldsymbol{uavav}C\boldsymbol{avavw}$, where $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$ are (possibly empty) strings, and $a \in \mathcal{A}(\boldsymbol{x}) - \{C\}$.

Let $\boldsymbol{x}' = \boldsymbol{uavav}(C\boldsymbol{avavw})^a_C = \boldsymbol{uavav}(aC\boldsymbol{v}^a_C C\boldsymbol{v}^a_C \boldsymbol{w}^a_C) = \boldsymbol{uavava}C\tilde{\boldsymbol{v}}C\tilde{\boldsymbol{v}}\tilde{\boldsymbol{w}}$. Clearly, $\boldsymbol{x'} \in S_d(n)$, $r(\boldsymbol{x}') \geq r(\boldsymbol{x})$, so $\boldsymbol{x}'$ is run-maximal and has no singletons.

Next, consider the case that $\boldsymbol{x}$ has at least 2 singletons C, D, of which one is unsafe, C. Without loss of generality, we can assume C occurs before $D : \boldsymbol{x} = \boldsymbol{uavav}C\boldsymbol{avavw}D\boldsymbol{z}$, where $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$, and $\boldsymbol{z}$ are (possibly empty) strings and $a \in \mathcal{A}(\boldsymbol{x}) - \{C, D\}$. Let $\boldsymbol{x_1} = \boldsymbol{uavav}(C\boldsymbol{avavw}D\boldsymbol{z})^a_C = \boldsymbol{uavava}C\tilde{\boldsymbol{v}}C\tilde{\boldsymbol{v}}\tilde{\boldsymbol{w}}D\tilde{\boldsymbol{z}}$. Clearly, $\boldsymbol{x_1} \in S_d(n)$ and $r(\boldsymbol{x_1}) \geq r(\boldsymbol{x})$. We then modify $\boldsymbol{x_1}$ by removing the safe symbol a immediately to the left of the first occurrence of C, yielding $\boldsymbol{x_2}$. Finally, we add a second copy of D adjacent to the original D, restoring the original length: $\boldsymbol{x_3} = \boldsymbol{uavav}C\tilde{\boldsymbol{v}}C\tilde{\boldsymbol{v}}\tilde{\boldsymbol{w}}DD\tilde{\boldsymbol{z}}$. $\boldsymbol{x_3} \in S_d(n)$ and $r(\boldsymbol{x_3}) > r(\boldsymbol{x_2}) \geq r(\boldsymbol{x_1}) \geq r(\boldsymbol{x})$, which contradicts the run-maximality of $\boldsymbol{x}$. ☐

Lemma 3 is a simple observation that for a position to be unsafe, a symbol must occur twice to the left and twice to the right of that position.

Lemma 3. *If a string $\boldsymbol{x}$ consists only of singletons, pairs, and triples, then every position is safe.*

A corollary of Lemma 3 is that the maximum number of runs in a string with only singletons, pairs, and triples is limited by the number of pairs and triples. Specifically, $r(\boldsymbol{x}) = \#pairs + \lfloor \frac{3}{2}\#triples \rfloor$. This is because a pair can only be involved in a single run, and a triple can be involved in at most 2 runs. The densest structure achievable is through overlapping triples in the pattern *aababb*, which has three runs for every two triples. The pairs, meanwhile, are maximized through adjacent copies.

4 Run-maximal strings below the main diagonal and in the immediate neighbourhood above

We first remark that every value below the main diagonal in the $(d, n-d)$ table is equal to the value on the main diagonal directly above it. In other words, the values on and below the main diagonal in a column are constant.

Proposition 4. *We have $\rho_d(n) = \rho_{n-d}(2n-2d)$ for $2 \leq d \leq n < 2d$.*

Proof. Consider a run-maximal string $\boldsymbol{x} \in S_d(n)$, where $2 \leq d \leq n < 2d$. By Lemma 2, we can assume $\boldsymbol{x}$ has no unsafe singletons. Since $n < 2d$, $\boldsymbol{x}$ must have a singleton, and hence it must be safe. We can remove this safe singleton, yielding a new string $\boldsymbol{y} \in S_{d-1}(n-1)$ and so $\rho_d(n) = r(\boldsymbol{x}) = r(\boldsymbol{y}) \leq \rho_{d-1}(n-1)$. Recall the following inequality noted in [6]:

$$\rho_d(n) \leq \rho_{d+1}(n+1) \text{ for } 2 \leq d \leq n \tag{1}$$

Thus, $\rho_{d-1}(n-1) = \rho_d(n)$. ☐

Proposition 4 together with inequality (1) gives the following equivalence noted in [6]: $\{\rho_d(n) \leq n-d \text{ for } 2 \leq d \leq n\} \Leftrightarrow \{\rho_d(2d) \leq d \text{ for } 2 \leq d\}$.

If there is a counter-example to the conjectured upper bound, then the main diagonal must contain a counter-example. If it falls under the main diagonal, then by Proposition 4 there must be a counter-example on the main diagonal – i.e. it can be *pushed up*, and if it falls above the main diagonal, by the inequality (1), there must be a counter-example on the main diagonal – i.e. the counter-example can be *pushed down.*

We extend Proposition 4 to bound the behaviour of the entries in the immediate neighbourhood above the main diagonal in the $(d, n-d)$ table. Proposition 5 establishes that the difference between the entry on the main diagonal and the entry immediately above it is at most 1. In addition, the difference is 1 if and only if every run-maximal string in $S_d(2d)$ consists entirely of pairs; otherwise, the difference is 0.

Proposition 5. *We have* $\rho_d(2d) \leq \rho_{d-1}(2d-1) + 1$ *for* $d \geq 3$.

Proof. Let $\boldsymbol{x} \in S_d(2d)$ be a run-maximal string with no unsafe singletons (by Lemma 2). If $\boldsymbol{x}$ does not have a singleton, then it consists entirely of pairs. It is clear that the pairs must be adjacent and that $r(\boldsymbol{x}) = d$ and so $\boldsymbol{x} = aabbcc\ldots$. Removing the first a and renaming the second to b, $\boldsymbol{y} = bbbcc\ldots \in S_{d-1}(2d-1)$ and $\rho_{d-1}(2d-1) \geq r(\boldsymbol{y}) = r(\boldsymbol{x}) - 1 = \rho_d(2d) - 1$. If $\boldsymbol{x}$ has a singleton, since it is safe we can remove it forming a string $\boldsymbol{y} \in S_{d-1}(2d-1)$ so that $\rho_{d-1}(2d-1) \geq r(\boldsymbol{y}) = r(\boldsymbol{x}) = \rho_d(2d)$, and so $\rho_{d-1}(2d-1) = \rho_d(2d)$. □

We have seen that the gap between the first entry above the diagonal and the diagonal entry is at most 1. Proposition 6 establishes that the three entries just above the diagonal are identical.

Proposition 6. *We have* $\rho_{d-1}(2d-1) = \rho_{d-2}(2d-2) = \rho_{d-3}(2d-3)$ *for* $d \geq 5$.

Proof. Let $\boldsymbol{x}$ be a run-maximal string in $S_{d-1}(2d-1)$. By Lemma 2 we can assume that either it has a safe singleton or no singletons at all. In the former case, we can remove the safe singleton obtaining $\boldsymbol{y} \in S_{d-2}(2d-2)$ so that $\rho_{d-2}(2d-2) \geq r(\boldsymbol{y}) \geq r(\boldsymbol{x}) = \rho_{d-1}(2d-1)$, and so $\rho_{d-1}(2d-1) = \rho_{d-2}(2d-2)$. In the latter case, $\boldsymbol{x}$ consists of pairs and one triple, and thus, by Lemma 3, all positions are safe. Therefore, we can move all the pairs to the end of the string, yielding $\boldsymbol{y} = aaabbcc\ldots \in S_{d-1}(2d-1)$ and by removing the first a and renaming the remaining a's to c's, $\boldsymbol{z} = ccbbcc\ldots \in S_{d-2}(2d-2)$. It follows that $\rho_{d-2}(2d-2) \geq r(\boldsymbol{z}) = r(\boldsymbol{y}) = r(\boldsymbol{x}) = \rho_{d-1}(2d-1)$, and so $\rho_{d-1}(2d-1) = \rho_{d-2}(2d-2)$.

Let $\boldsymbol{x}$ be now a run-maximal string in $S_{d-2}(2d-2)$. Again, if $\boldsymbol{x}$ has a singleton, we can assume by Lemma 2 it is safe and form $\boldsymbol{y}$ by removing the singleton. $\boldsymbol{y} \in S_{d-3}(2d-3)$ and $\rho_{d-3}(2d-3) \geq r(\boldsymbol{y}) \geq r(\boldsymbol{x}) = \rho_{d-2}(2d-2)$. If $\boldsymbol{x}$ does not have a singleton, then $r(\boldsymbol{x}) = d-1$. To see this, consider the two cases:

(*i*) $\boldsymbol{x}$ consists of two triples and several pairs. The most runs which may be obtained in such a string, after grouping the pairs at the end of the string, is through the arrangement $aababbccddee\ldots$. In this case, there are $d-4$ runs from the pairs, and 3 runs from the triples, giving a total of $d-1$ runs.

(*ii*) $\boldsymbol{x}$ consists of a quadruple and several pairs. The most runs which may be obtained in this case is from a string with either the structure $aabbaaccddee\ldots$, or $aabaabccddee\ldots$, where all the pairs have been grouped at the end, except for the pair of bs which is used to break up the quadruple. In both cases, there are $d-4$ runs involving characters c onward, and three runs involving the characters a and b, again giving a total of $d-1$ runs.

Now consider a string $\boldsymbol{y} = aabbaabbcdee\ldots \in S_{d-2}(2d-2)$, which has two quadruples (of a's and b's), two singletons (c and d), and several pairs ($e\ldots$). This string has

$d-6$ runs from the pairs ee onward, and 5 runs from the characters a and b, giving a total of $d-1$ runs, i.e. $r(\boldsymbol{x}) = r(\boldsymbol{y})$. The singleton c in $\boldsymbol{y}$ being clearly safe, we can remove it and continue as in the previous case. □

Remark 7 below providing a lower bound for the first 4 entries above the main diagonal of the $(d, n-d)$ table, is a corollary of the inequality $\rho_{d+s}(n+2s) \geq \rho_d(n)+s$, noted in [6], applied to $\rho_2(k) = k-3$ for $k = 5, 6, 7$ and 8.

Remark 7. We have $\rho_{d-k}(2d-k) \geq d-1$ for $k = 1, 2, 3$ and 4 and $d \geq 6$.

5 Structural properties of run-maximal strings on the main diagonal

We explore structural properties of the run-maximal strings on the main diagonal. These results yield properties for run-maximal strings that have their length bounded by nine times the number of distinct symbols they contain. We can thus shift the critical region of the $(d, n-d)$ table as summarized in Theorem 8, the proof for which can be found at the end of this section.

Theorem 8. *We have* $\{\rho_d(n) \leq n-d$ *for* $2 \leq d \leq n\} \Leftrightarrow \{\rho_d(9d) \leq 8d$ *for* $d \geq 2\}$.

Proposition 9 describes useful structural properties of run-maximal strings on the main diagonal. The proof of the proposition relies on a series of lemmas all of which are dealing with the same basic scenario: assuming we know that the table obeys the conjecture for all columns to the left of column d, which is the first *unknown* column, we investigate the run-maximal strings of $S_d(2d)$.

Proposition 9. *[Proposition] Let* $\rho_{d'}(2d') \leq d'$ *for* $2 \leq d' < d$. *Let* $\boldsymbol{x}$ *be a run-maximal string in* $S_d(2d)$. *Either* $r(\boldsymbol{x}) = \rho_d(2d) = d$ *or* $\boldsymbol{x}$ *has at least* $\lceil \frac{7d}{8} \rceil$ *singletons, and no symbol occurs exactly 2, 3, ..., 8 times in* $\boldsymbol{x}$.

Proof. The proof that each symbol must be a singleton or occur at least 9 times is a direct result of the lemmas which make up the remainder of this section. Then, let $\boldsymbol{x} \in S_d(2d)$ be run-maximal, m_1 denote the number of singletons, and m_2 the number of non-singleton symbols of $\boldsymbol{x}$. We have $m_1 + 9m_2 \leq 2d$ and $m_1 + m_2 = d$, which implies that $m_2 \leq d/8$ and hence $m_1 \geq \lceil 7d/8 \rceil$. □

Proposition 9 provides a purely structural proof that $\rho_d(2d) = d$ for $d \leq 15$, and using the computer generated results of $\rho_2(d+2)$ for $d = 16, \ldots, 23$, that $\rho_d(2d) = d$ for $d \leq 23$.

Corollary 10. *We have* $\rho_d(2d) = d$ *for* $d \leq 23$ *and* $\rho_d(n) \leq n-d$ *for* $n-d \leq 23$.

Proof. Assume that run-maximal $\boldsymbol{x} \in S_d(2d)$ satisfies $r(\boldsymbol{x}) = \rho_d(2d) > d$. By Proposition 9, $\boldsymbol{x}$ consists only of singletons for $2 \leq d \leq 6$, $r(\boldsymbol{x}) = \rho_1(d+1) = 1$ for $8 \leq d \leq 15$, and $d < r(\boldsymbol{x}) = \rho_2(d+2)$ for $16 \leq d \leq 23$, which are impossible. □

Before we begin the lemmas to support 9Structural properties of run-maximal strings on the main diagonalproposition.9, we first introduce a few concepts.

Definition 11 (Map). *A run (s, p, d) of period p, starting at position s and ending at position d of a string $\boldsymbol{x}$* maps *position i to position j if $s \le i < j \le d$ and $j - i = p$. We denote a mapping from i to j by $i \to j$ and call it a single-mapping. We extend the mapping notation to $(i_1, i_2) \to (j_1, j_2)$, denoting $s \le i_1 < i_2 < j_1 < j_2 < d$ and $j_1 - i_1 = j_2 - i_2 = p$ and call it a double-mapping. The triple- and higher order mappings are defined analogously.*

A *multi-mapping* will be any mapping which is not a single-mapping. The presence of a multi-mapping imposes equality on the substrings bounded on each side. For example, in the double-mapping $(i_j, i_{j+1}) \to (i_{j+2}, i_{j+3})$, $\boldsymbol{x}[i_j..i_{j+1}]$ the substring between i_j and i_{j+1} is the same as $\boldsymbol{x}[i_{j+2}..i_{j+3}]$ the substring between i_{j+2} and i_{j+3}.

In the following lemmas, we assume that for $2 \le d' < d$, the conjecture holds, i.e. $\rho_{d'}(2d') \le d'$. Note that it is equivalent to $\rho_{d'}(n') \le n' - d'$ for $2 \le d' \le n'$ when $n' - d' < d$. We consider a run-maximal string $\boldsymbol{x} \in S_d(2d)$ containing a k-tuple of c's such that $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \ldots \boldsymbol{u_{k-1}} c \boldsymbol{u_k}$. We show that either the string $\boldsymbol{x}$ obeys the conjectured upper bound, or can be manipulated to obtain a new string $\boldsymbol{y}$ with a larger alphabet of the same or shorter length. We ensure that the manipulation process does not destroy more runs than the the amount the alphabet is increased or the length decreased. This allows us to estimate the number of runs in $\boldsymbol{y}$ based on the values in the table for some $d' < d$. In essence, we manipulate a string from column d to a string from some column $d' < d$ while monitoring the number of runs. In the manipulation process, we put an upper limit on the number of runs which are destroyed (π), and a lower limit on how many additional symbols are introduced (δ).

In order to have more distinct symbols in $\boldsymbol{y}$ than in $\boldsymbol{x}$ we employ several strategies. We can change all but one of the c's to new characters c_2, c_3, $\ldots c_k$, thus introducing $k-1$ new characters. When multiple disjoint copies of a substring occur in $\boldsymbol{x}$, we can replace all copies of a symbol within one copy of the substring with a new symbol which does not occur elsewhere in $\boldsymbol{x}$. Given $\boldsymbol{x} = \boldsymbol{uvu}$, we can increase the number of distinct symbols with $\boldsymbol{y} = \boldsymbol{uv\widehat{u}}$. $\boldsymbol{uv\widehat{u}w\widehat{u}}$ has two distinct symbols more than $\boldsymbol{uvuwu}$ does, etc.

Since the length of the string remains constant while the number of distinct characters increases, $\boldsymbol{y} \in S_{d+\delta}(n)$. Since $n - (d + \delta) < n - d$, by the induction hypothesis we know that $r(\boldsymbol{y}) \le n - d - \delta$. Therefore, $r(\boldsymbol{x}) - \pi \le r(\boldsymbol{y}) \le n - d - \delta$, so $\rho_d(n) = r(\boldsymbol{x}) \le n - d - \delta + \pi$. Thus, whenever $\pi \le \delta$, $\rho_d(2d) \le d$.

Lemma 12. *[Lemma]*
Let $\rho_{d'}(2d') \le d'$ for $d' < d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. Either $r(\boldsymbol{x}) = \rho_d(2d) \le d$ or $\boldsymbol{x}$ does not contain a pair.

Proof. As shown in [6], a pair of c's can be involved in at most one run. We change the second c to a new symbol c_2 creating $\boldsymbol{y} = \boldsymbol{u_0} c \boldsymbol{u_1} c_2 \boldsymbol{u_2}$. We destroy at most the single run which contains the pair ($\pi \le 1$), and gain 1 symbol ($\delta = 1$). As $\pi \le \delta$, $\boldsymbol{x}$ satisfies the conjecture or $\boldsymbol{x}$ does not contain a pair. □

Lemma 13. *[Lemma] Let $\rho_{d'}(2d') \le d'$ for $d' < d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. Either $r(\boldsymbol{x}) = \rho_d(2d) \le d$ or $\boldsymbol{x}$ does not contain a triple.*

Proof. If a triple of c's is involved in less than two runs, we can proceed as in the proof of the previous lemma. Let us thus assume that the c's are involved in two runs.

The string has the form $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_3}$. In this case, we replace two of the c's with new symbols c_2 and c_3 creating $\boldsymbol{y} = \boldsymbol{u_0} c \boldsymbol{u_1} c_2 \boldsymbol{u_2} c_3 \boldsymbol{u_3}$. This destroys at most only the two possible runs, while we gain two symbols ($\delta = 2$). δ is again sufficiently large, so either $\boldsymbol{x}$ satisfies the conjecture or $\boldsymbol{x}$ does not have a triple. $\square$

For the above two lemmas, we did not need to use the notion of mappings. But it can be seen that the runs involved only corresponded to single-mappings. If only single-mappings are involved, then it is straight-forward to obtain a new string with more distinct symbols while limiting the number of runs destroyed. In the following cases, we must always deal with a multi-mapping.

Lemma 14. *[Lemma] Let $\rho_{d'}(2d') \leq d'$ for $d' < d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. Either $r(\boldsymbol{x}) = \rho_d(2d) \leq d$ or $\boldsymbol{x}$ does not contain a quadruple.*

Proof. A quadruple of c's at positions $i_1 < i_2 < i_3 < i_4$ can be involved in at most four runs, corresponding to a double-mapping $(i_1, i_2) \to (i_3, i_4)$ and single-mappings $i_1 \to i_2$, $i_2 \to i_3$, and $i_3 \to i_4$. If there are only three or fewer runs the c's are involved in, replacing three occurrences of c's by three new symbols will give $\delta = 3$ and $\pi \leq 3$, hence $\pi \leq \delta$, giving the result of this lemma.

Hence we assume that the c's are involved in exactly four runs. In this case replacing three of the c's by new symbols is no longer enough, as π would be greater than δ. However, from the double-mapping $(i_1, i_2) \to (i_3, i_4)$, we know that $\boldsymbol{x}[i_1..i_2] = \boldsymbol{x}[i_3..i_4]$. Thus if $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_3} c \boldsymbol{u_4}$, then $\boldsymbol{u_1} = \boldsymbol{u_3}$, hence $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_1} c \boldsymbol{u_4}$.

If $\boldsymbol{u_1}$ is non-empty, let $a \in \boldsymbol{u_1}$. We replace the last three copies of c by new symbols c_2, c_3, and c_4, and all instances of a in the second occurrence of $\boldsymbol{u_1}$ by a new symbol a_1 producing $\widehat{\boldsymbol{u_1}}$: $\boldsymbol{y} = \boldsymbol{u_0} c \boldsymbol{u_1} c_2 \boldsymbol{u_2} c_3 \widehat{\boldsymbol{u_1}} c_4 \boldsymbol{u_4}$. This gives $\pi \leq 4$, but now $\delta = 4$, satisfying the lemma.

If $\boldsymbol{u_1}$ were empty, either $\boldsymbol{u_2}$ is non-empty, giving $\boldsymbol{x}$ the form: $\boldsymbol{x} = \boldsymbol{u_0}$ c c $\boldsymbol{u_2}$ c c $\boldsymbol{u_4}$. Since $\boldsymbol{u_2}$ is non-empty, in order for the single mapping $i_2 \to i_3$ to exist, there is a symbol in $\boldsymbol{u_2}$ which must occur between the first and second c's, or the third and fourth c's. However, this requires $\boldsymbol{u_1}$ to be non-empty, a contradiction. Therefore, the mapping $i_2 \to i_3$ cannot refer to a run in the string, a contradiction with our assumption of the c's being involved in four different runs.

Therefore, $\boldsymbol{u_2}$ must be empty as well and so we have $\boldsymbol{x} = \boldsymbol{u_0} cccc \boldsymbol{u_4}$, merging the 4 possible runs containing the quadruple into a single run, a contradiction. $\square$

Lemma 15. *[Lemma] Let $\rho_{d'}(2d') \leq d'$ for $d' < d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. Either $r(\boldsymbol{x}) = \rho_d(2d) \leq d$ or $\boldsymbol{x}$ does not contain a quintuple.*

Proof. A quintuple of c's at positions $i_1 < i_2 < i_3 < i_4 < i_5$ can be involved in at most 5 runs despite there being 6 possible mappings: double-mappings $(i_1, i_2) \to (i_3, i_4)$ and $(i_2 \to i_3) \to (i_4, i_5)$, and single-mappings $i_1 \to i_2$, $i_2 \to i_3$, $i_3 \to i_4$, and $i_4 \to i_5$. If both double-mappings exist, they correspond to the same run, as they have the same period p and overlap by at least p.

Again, if the quintuple is involved in fewer than 5 runs, we can just replace 4 of the c's with new symbols as in the previous lemmas. Thus we are assuming that the quintuple is involved in exactly 5 runs. However, in this case we do not need to introduce 5 new symbols, as we can always introduce 1 new symbol while only destroying a single run. There are 3 cases to discuss:

1. All mappings exist. Then $\boldsymbol{x}[i_5]$ is involved in two runs, one corresponding to $(i_1, i_2) \to (i_3, i_4)$ and $(i_2, i_3) \to (i_4, i_5)$, and one corresponding to $i_4 \to i_5$. If we replace $\boldsymbol{x}[i_5]$ by a new symbol c_5, we destroy the run corresponding to $i_4 \to i_5$, but only a part of the run corresponding to $(i_1, i_2) \to (i_3, i_4)$ and $(i_2 \to i_3) \to (i_4, i_5)$. We thus obtain $\pi \leq 1 = delta$.

2. The mapping $(i_1, i_2) \to (i_3, i_4)$ exists, but $(i_2, i_3) \to (i_4, i_5)$ does not, while all single-mappings exist. We can proceed as in the previous case.
3. The mapping $(i_1, i_2) \to (i_3, i_4)$ does not exist, but $(i_2, i_3) \to (i_4, i_5)$ does, while all possible single-mappings exist. We proceed as in the first case, but with $\boldsymbol{x}[\boldsymbol{i_1}]$ instead of $\boldsymbol{x}[\boldsymbol{i_5}]$. $\square$

Lemma 16. *[Lemma] Let $\rho_{d'}(2d') \leq d'$ for $d' < d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. Either $r(\boldsymbol{x}) = \rho_d(2d) \leq d$ or $\boldsymbol{x}$ does not contain a 6-tuple.*

Proof. A 6-tuple at positions $i_1 < \ldots < i_6$ can be involved in at most 8 runs, despite there being 9 available mappings:

– triple-mapping: $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$
– double-mappings: $(i_1, i_2) \to (i_3, i_4)$, $(i_2 \to i_3) \to (i_4, i_5)$, and $(i_3, i_4) \to (i_5, i_6)$
– single-mappings: $i_1 \to i_2$, $i_2 \to i_3$, $i_3 \to i_4$, $i_4 \to i_5$, and $i_5 \to i_6$

As in Lemma 15, if either both $(i_1, i_2) \to (i_3, i_4)$ and $(i_2 \to i_3) \to (i_4, i_5)$, or $(i_2 \to i_3) \to (i_4, i_5)$ and $(i_3, i_4) \to (i_5, i_6)$ exist, the two runs they correspond to are actually the same run.

Let $\boldsymbol{x} = \boldsymbol{u_0}\mathrm{c}\boldsymbol{u_1}\mathrm{c}\boldsymbol{u_2}\mathrm{c}\boldsymbol{u_3}\mathrm{c}\boldsymbol{u_4}\mathrm{c}\boldsymbol{u_5}\mathrm{c}\boldsymbol{u_6}$. We consider each configuration of of multi-mappings separately:

1. $(i_1, i_2) \to (i_3, i_4)$, $(i_3, i_4) \to (i_5, i_6)$, and all single-mappings: By the double-mappings, $\boldsymbol{u_1} = \boldsymbol{u_3} = \boldsymbol{u_6}$, and therefore the string $\boldsymbol{x}$ has the form: $\boldsymbol{x} = \boldsymbol{u_0}c\boldsymbol{u_1}c\boldsymbol{u_2}c\boldsymbol{u_1}c\boldsymbol{u_4}c\boldsymbol{u_1}c\boldsymbol{u_6}$. We consider the different cases of empty and non-empty substrings separately:
   (a) If $\boldsymbol{u_1}$ is non-empty, we replace 5 of the c's with new symbols, and all instances of some symbol in 2 of the 3 copies of $\boldsymbol{u_1}$, giving $\widehat{\boldsymbol{u_1}}$. So, $\boldsymbol{y} = \boldsymbol{u_0}c\boldsymbol{u_1}c_2\boldsymbol{u_2}c_3\widehat{\boldsymbol{u_1}}c_4\boldsymbol{u_4}c_5\widehat{\boldsymbol{u_1}}c_6\boldsymbol{u_6}$. This gives $\pi \leq 7 = \delta$.
   (b) Otherwise, $\boldsymbol{u_1}$ is empty. Assume that both $\boldsymbol{u_2}$ and $\boldsymbol{u_4}$ are non-empty. The string then has the form: $\boldsymbol{x} = \boldsymbol{u_0}cc\boldsymbol{u_2}cc\boldsymbol{u_4}cc\boldsymbol{u_6}$. This eliminates the possibility of runs from the single-mappings $i_2 \to i_3$ and $i_4 \to i_5$. By replacing 5 of the c's with new symbols, we have $\pi \leq 5 = \delta$.
2. $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$ and all single-mappings: By the triple-mapping, $\boldsymbol{u_1} = \boldsymbol{u_4}$ and $\boldsymbol{u_2} = \boldsymbol{u_5}$. If $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ are both be empty, then the possible run from the single mapping $i_1 \to i_2$ is merged with the one from $i_2 \to i_3$, and $i_4 \to i_5$ is merged with $i_5 \to i_6$. By replacing 5 of the c's with new symbols, we have $\pi \leq 4 < \delta = 5$. Therefore, assume at least one of $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ are non-empty. In this case, we can also replace all instances of some symbol in one of them (whichever is non-empty), giving $\boldsymbol{y} = \boldsymbol{u_0}c_1\boldsymbol{u_1}c_2\boldsymbol{u_2}c_3\boldsymbol{u_3}c_4\widehat{\boldsymbol{u_1}}c_5\widehat{\boldsymbol{u_2}}c_6\boldsymbol{u_6}$. This transformation destroys at most 6 runs and introduces 6 or 7 new symbols ($\pi \leq 6 \leq \delta \leq 7$).
3. $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, one of $(i_1, i_2) \to (i_3, i_4)$ or $(i_3, i_4) \to (i_5, i_6)$ (but not both), and all the single-mappings: Having one or the other of the double-mappings are clearly mirror cases of each other, so we will assume without loss of generality that $(i_1, i_2) \to (i_3, i_4)$ exists. By the double- and triple-mappings, $\boldsymbol{u_1} = \boldsymbol{u_3} = \boldsymbol{u_4}$ and $\boldsymbol{u_2} = \boldsymbol{u_5}$.
   (a) If $\boldsymbol{u_1}$ is non-empty, replace each instance of a symbol in 2 copies of it, along with 5 of the c's: with new symbols, $\boldsymbol{y} = \boldsymbol{u_0}c\boldsymbol{u_1}c_2\boldsymbol{u_2}c_3\widehat{\boldsymbol{u_1}}c_4\widehat{\boldsymbol{u_1}}c_5\boldsymbol{u_2}c_6\boldsymbol{u_6}$. This increases the number of distinct symbols by 7 while destroying at most 7 runs from the mappings ($\pi \leq 7 = \delta$).

(b) If $\boldsymbol{u_1}$ is empty, we have $\boldsymbol{x} = \boldsymbol{u_0}cc\boldsymbol{u_2}ccc\boldsymbol{u_2}c\boldsymbol{u_6}$. This arrangement loses 1 possible run due to merging $i_3 \to i_4$ and $i_4 \to i_5$, and eliminates the possible run from the mapping $i_2 \to i_3$ when $\boldsymbol{u_2}$ is non-empty, since if $\boldsymbol{u_2}$ is empty, all the runs merge down to a single run. This reduces π from 7 down to 5, so by replacing 5 of the c's with new symbols, we achieve $\pi \leq 5 = \delta$.

4. $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, $(i_2, i_3) \to (i_4, i_5)$ exist, and so do all the single-mappings. By the double- and triple-mappings, $\boldsymbol{u_1} = \boldsymbol{u_2} = \boldsymbol{u_4} = \boldsymbol{u_5}$.
   (a) If $\boldsymbol{u_1}$ is non-empty, we relabel each instance of a symbol in 3 copies of $\boldsymbol{u_1}$: $\boldsymbol{y} = \boldsymbol{u_0}c_1\boldsymbol{u_1}c_2\widehat{\boldsymbol{u_1}}c_3\boldsymbol{u_3}c_4\widehat{\boldsymbol{u_1}}c_5\widehat{\boldsymbol{u_1}}c_6\boldsymbol{u_6}$. This increases the number of distinct symbols by 8 while destroying at most 7 runs ($\pi \leq 7 < \delta = 8$).
   (b) If $\boldsymbol{u_1}$ is empty, $\boldsymbol{x} = \boldsymbol{u_0}ccc\boldsymbol{u_3}ccc\boldsymbol{u_6}$, and 2 single-mappings are lost through merging $i_1 \to i_2$ with $i_2 \to i_3$ and $i_4 \to i_5$ with $i_5 \to i_6$. Replacing 5 of the c's with new symbols is sufficient to give $\pi \leq 5 = \delta$.
5. $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, $(i_1, i_2) \to (i_3, i_4)$, $(i_3, i_4) \to (i_5, i_6)$ exist, and so do all the single-mappings. From the double- and triple-mappings, $\boldsymbol{u_1} = \boldsymbol{u_2} = \boldsymbol{u_3} = \boldsymbol{u_4} = \boldsymbol{u_5}$. All the possible runs are actually one long run, so the last c may be replaced with a new symbol without destroying any runs. This gives $\pi = 0 < \delta = 1$. □

Lemma 17. *[Lemma] Let $\rho_{d'}(2d') \leq d'$ for $d' < d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. Either $r(\boldsymbol{x}) = \rho_d(2d) \leq d$ or $\boldsymbol{x}$ does not contain a 7-tuple.*

Proof. A 7-tuple of c's at positions $i_1 < \ldots < i_7$ can be involved in 9 runs, despite there being 12 possible mappings:

- triple-mappings: $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, and$(i_2, i_3, i_4) \to (i_5, i_6, i_7)$
- double-mappings: $(i_1, i_2) \to (i_3, i_4)$, $(i_2, i_3) \to (i_4, i_5)$, $(i_3, i_4) \to (i_5, i_6)$, and $(i_4, i_5) \to (i_6, i_7)$
- single-mappings: $i_1 \to i_2$, $i_2 \to i_3$, $i_3 \to i_4$, $i_4 \to i_5$, $i_5 \to i_6$, and $i_6 toi_7$

As with the overlapping double-mappings, if both triple-mappings are present, they correspond to the same run. As having both triple-mappings cannot increase the possible number of runs, we assume without loss of generality that if a triple-mapping is present, it is $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$.

As with 15Structural properties of run-maximal strings on the main diagonallemma.15, we need every element of the 7-tuple to be covered by a multi-mapping.

There are 5 cases to consider:

1. $(i_1, i_2) \to (i_3, i_4)$, $(i_4, i_5) \to (i_6, i_7)$, and all single-mappings (a total of 8 mappings). Due to the double-mappings, $\boldsymbol{u_1} = \boldsymbol{u_3}$ and $\boldsymbol{u_4} = \boldsymbol{u_6}$, the string $\boldsymbol{x}$ must have the form $\boldsymbol{x} = \boldsymbol{u_0}c\boldsymbol{u_1}c\boldsymbol{u_2}c\boldsymbol{u_1}c\boldsymbol{u_4}c\boldsymbol{u_5}c\boldsymbol{u_4}c\boldsymbol{u_7}$.
   (a) $\boldsymbol{u_1}$ non-empty, $\boldsymbol{u_4}$ non-empty: replace all instances of a symbol in 1 copy of each of $\boldsymbol{u_1}$ and $\boldsymbol{u_4}$, along with 6 of the c's with new symbols:
   $\boldsymbol{y} = \boldsymbol{u_0}c\boldsymbol{u_1}c_2\boldsymbol{u_2}c_3\widehat{\boldsymbol{u_1}}c_4\boldsymbol{u_4}c_5\boldsymbol{u_5}c_6\boldsymbol{u_4}c_7\boldsymbol{u_7}$. This destroys at most 8 runs and introduces 8 new symbols ($\pi \leq 8 = \delta$).
   (b) $\boldsymbol{u_1}$ non-empty, $\boldsymbol{u_4}$ empty. The string $\boldsymbol{x}$ then has the form
   $\boldsymbol{x} = \boldsymbol{u_0}c\boldsymbol{u_1}c\boldsymbol{u_2}c\boldsymbol{u_1}cc\boldsymbol{u_5}cc\boldsymbol{u_7}$. This eliminates the possibility of a run corresponding to the mapping $i_5 \to i_6$, unless $\boldsymbol{u_5}$ is empty, in which case 2 possible runs are lost to merging into one. Replacing all instances of a symbol in 1 copy of $\boldsymbol{u_1}$ along with 6 of the c's by new symbols gives $\pi \leq 7 = \delta$.
   (c) $\boldsymbol{u_1}$ empty, $\boldsymbol{u_4}$ non-empty. This is a reversal of the previous case, and is satisfied accordingly.

(d) $\boldsymbol{u_1}$ and $\boldsymbol{u_4}$ empty. The possibility of runs corresponding to the mappings $i_2 \to i_3$ and $i_5 \to i_6$ are eliminated, so relabeling 6 of the c's gives $\pi \leq 6 = \delta$.

2. $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, $(i_4, i_5) \to (i_6, i_7)$, and all single-mappings (a total of 8 mappings). From the multi-mappings, $\boldsymbol{u_1} = \boldsymbol{u_4} = \boldsymbol{u_6}$ and $\boldsymbol{u_2} = \boldsymbol{u_5}$, so the string $\boldsymbol{x}$ must have the form $\boldsymbol{x} = \boldsymbol{u_0}c\boldsymbol{u_1}c\boldsymbol{u_2}c\boldsymbol{u_3}c\boldsymbol{u_1}c\boldsymbol{u_2}c\boldsymbol{u_1}c\boldsymbol{u_7}$.
   (a) If $\boldsymbol{u_1}$ is non-empty, we replace all instances of a symbol in 2 copies of $\boldsymbol{u_1}$, along with 6 of the c's with new symbols
   $\boldsymbol{y} = \boldsymbol{u_0}c\boldsymbol{u_1}c_2\boldsymbol{u_2}c_3\boldsymbol{u_3}c_4\widehat{\boldsymbol{u_1}}c_5\boldsymbol{u_2}c_6\widehat{\boldsymbol{u_1}}c_7\boldsymbol{u_7}$. This gives $\pi \leq 8 = \delta$.
   (b) Otherwise, $\boldsymbol{u_1}$ is empty, so the string $\boldsymbol{x} = \boldsymbol{u_0}cc\boldsymbol{u_2}c\boldsymbol{u_3}cc\boldsymbol{u_2}cc\boldsymbol{u_7}$. When $\boldsymbol{u_2}$ is non-empty, this eliminates the possibility of a run corresponding to the mapping $i_5 \to i_6$, so by replacing all instances of a symbol in a $\boldsymbol{u_2}$ along with 6 of the c's with new symbols, we achieve $\pi \leq 7 = \delta$.
   (c) If $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ are both empty, 3 possible runs are lost through merging, so relabeling 6 of the c's gives $\pi \leq 5 < \delta = 6$.

3. $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, $(i_1, i_2) \to (i_3, i_4)$, $(i_4, i_5) \to (i_6, i_7)$, and all the single-mappings (a total of 9 mappings). From the multi-mappings, $\boldsymbol{u_1} = \boldsymbol{u_4} = \boldsymbol{u_6}$ and $\boldsymbol{u_2} = \boldsymbol{u_3} = \boldsymbol{u_5}$.
   (a) If $\boldsymbol{u_1}$ are $\boldsymbol{u_2}$ are both non-empty, replacing all instances of a symbol in 2 copies of each of $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ along with 6 of the c's with new symbols, gives us $\pi \leq 9 < \delta = 10$.
   (b) If $\boldsymbol{u_1}$ is empty, the string has the form $\boldsymbol{x} = \boldsymbol{u_0}cc\boldsymbol{u_2}c\boldsymbol{u_2}cc\boldsymbol{u_2}cc\boldsymbol{u_7}$. The possible run corresponding to the mapping $i_5 \to i_6$ is eliminated, so replacing all instances of a symbol in 2 copies of $\boldsymbol{u_2}$ along with 6 of the c's with new symbols is sufficient to give $\pi \leq 8 = \delta$.
   (c) If $\boldsymbol{u_2}$ is empty, the string has the form $\boldsymbol{x} = \boldsymbol{u_0}c\boldsymbol{u_1}ccc\boldsymbol{u_1}cc\boldsymbol{u_1}c\boldsymbol{u_7}$. The runs corresponding to the mappings $i_2 \to i_3$ and $i_3 \to i_4$ are merged, and the possible run corresponding to the mapping $i_4 \to i_5$ is eliminated, so replacing all instances of a symbol in 2 copies of $\boldsymbol{u_1}$ along with 6 of the c's with new symbols is sufficient to give $\pi \leq 7 < \delta = 8$. □

Lemma 18. *[Lemma] Let $\rho_{d'}(2d') \leq d'$ for $d' < d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. Either $r(\boldsymbol{x}) = \rho_d(2d) \leq d$ or $\boldsymbol{x}$ does not contain an 8-tuple.*

Proof. 1. $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, $(i_5, i_6) \to (i_7, i_8)$, and all single-mappings (a total of 9 mappings). By the multi-mappings, the string has the form
$\boldsymbol{x} = \boldsymbol{u_0}c\boldsymbol{u_1}c\boldsymbol{u_2}c\boldsymbol{u_3}c\boldsymbol{u_1}c\boldsymbol{u_2}c\boldsymbol{u_6}c\boldsymbol{u_2}c\boldsymbol{u_8}$.
   (a) If $\boldsymbol{u_2}$ is non-empty, we can replace all instances of a symbol in 2 copies of $\boldsymbol{u_2}$:
   $\boldsymbol{y} = \boldsymbol{u_0}c\boldsymbol{u_1}c_2\boldsymbol{u_2}c_3\boldsymbol{u_3}c_4\boldsymbol{u_1}c_5\widehat{\boldsymbol{u_2}}c_6\boldsymbol{u_6}c_7\widehat{\boldsymbol{u_2}}c_8\boldsymbol{u_8}$. This gives $\pi \leq 9 = \delta$.
   (b) Otherwise, $\boldsymbol{u_2}$ is empty, giving $\boldsymbol{x} = \boldsymbol{u_0}c\boldsymbol{u_1}cc\boldsymbol{u_3}c\boldsymbol{u_1}cc\boldsymbol{u_6}cc\boldsymbol{u_8}$. This eliminates the possibility of a run from the mapping $i_6 \to i_7$. This means $\pi \leq 8$.
      i. If $\boldsymbol{u_1}$ is non-empty, we can replace all instances of a symbol in 1 of the copies of $\boldsymbol{u_1}$ along with 7 of the c's, giving
      $\boldsymbol{y} = \boldsymbol{u_0}c\boldsymbol{u_1}c_2c_3\boldsymbol{u_3}c_4\widehat{\boldsymbol{u_1}}c_5c_6\boldsymbol{u_6}c_7c_8\boldsymbol{u_8}$. This results in $\pi \leq 8 = \delta$.
      ii. If $\boldsymbol{u_1}$ is also empty, the string is structured as $\boldsymbol{x} = \boldsymbol{u_0}ccc\boldsymbol{u_3}ccc\boldsymbol{u_6}cc\boldsymbol{u_8}$. In addition to the elimination of the mapping $i_6 \to i_7$, the runs corresponding to the single mappings $i_1 \to i_2$ and $i_2 \to i_3$ are merged, along with the runs corresponding to the mappings $i_4 \to i_5$ and $i_5 \to i_6$. This reduces the maximum number of runs to $\pi \leq 6$. By relabeling 7 of the c's, we obtain $\pi \leq 6 < \delta = 7$.

2. $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, $(i_1, i_2) \to (i_3, i_4)$, $(i_5, i_6) \to (i_7, i_8)$, and all single-mappings (a total of 10 mappings). By the multi-mappings, the string has the form $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_1} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_6} c \boldsymbol{u_2} c \boldsymbol{u_8}$.
   (a) If $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ are both non-empty, we can replace all instances of a symbol in 2 copies of each, along with 7 of the c's:
   $\boldsymbol{y} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \widehat{\boldsymbol{u_1}} c \widehat{\boldsymbol{u_1}} c \widehat{\boldsymbol{u_2}} c \boldsymbol{u_6} c \widehat{\boldsymbol{u_2}} c \boldsymbol{u_8}$. This results in $\pi \le 10 < \delta = 11$.
   (b) If $\boldsymbol{u_1}$ is empty and $\boldsymbol{u_2}$ is non-empty, we have $\boldsymbol{x} = \boldsymbol{u_0} cc \boldsymbol{u_2} ccc \boldsymbol{u_2} c \boldsymbol{u_6} c \boldsymbol{u_2} c \boldsymbol{u_8}$. This eliminates the possibility of a run corresponding to the mapping $i_2 \to i_3$, and merges the runs corresponding to $i_3 \to i_4$ and $i_5 \to i_6$, so $\pi \le 8$. We replace all instances of a symbol in 2 of the copies of $\boldsymbol{u_2}$, giving $\boldsymbol{y} = \boldsymbol{u_0} c c_2 \boldsymbol{u_2} c_3 c_4 c_5 \widehat{\boldsymbol{u_2}} c_6 \boldsymbol{u_6} c_7 \widehat{\boldsymbol{u_2}} c_8 \boldsymbol{u_8}$. This results in $\pi \le 8 < \delta = 9$.
   (c) If $\boldsymbol{u_1}$ is non-empty, and $\boldsymbol{u_2}$ is empty, we have $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} cc \boldsymbol{u_1} c \boldsymbol{u_1} cc \boldsymbol{u_6} cc \boldsymbol{u_8}$. This eliminates the possibility of a run corresponding to the mapping $i_6 \to i_7$ (unless $\boldsymbol{u_6}$ is empty, which results in 3 possible runs being merged). We replace all instances of a symbol in 2 copies of $\boldsymbol{u_1}$, along with 7 of the c's with new symbols, giving $\boldsymbol{y} = \boldsymbol{u_0} c \boldsymbol{u_1} c_2 c_3 \widehat{\boldsymbol{u_1}} c_4 \widehat{\boldsymbol{u_1}} c_5 c_6 \boldsymbol{u_6} c_7 c_8 \boldsymbol{u_8}$. This results in $\pi \le 9 = \delta$.
   (d) If $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ are both empty, we have $\boldsymbol{x} = \boldsymbol{u_0} cccccc \boldsymbol{u_6} cc \boldsymbol{u_8}$, merging 5 runs corresponding to the single mappings, and preventing the possible run corresponding to $(i_1, i_2) \to (i_3, i_4)$ because because its generator would be non-primitive. Therefore, by replacing 7 of the c's with new symbols, we obtain$\pi \le 5 < \delta = 7$.
3. $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, $(i_2, i_3) \to (i_4, i_5)$, $(i_5, i_6) \to (i_7, i_8)$, and all single-mappings (a total of 10 mappings). By the multi-mappings, the string has the form $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_1} c \boldsymbol{u_3} c \boldsymbol{u_1} c \boldsymbol{u_1} c \boldsymbol{u_6} c \boldsymbol{u_1} c \boldsymbol{u_8}$.
   (a) If $\boldsymbol{u_1}$ is non-empty, we can replace all instances of a symbol in 4 copies of $\boldsymbol{u_1}$, along with 7 of the c's ith new symbols, yielding
   $\boldsymbol{y} = \boldsymbol{u_0} c \boldsymbol{u_1} c_2 \widehat{\boldsymbol{u_1}} c_3 \boldsymbol{u_3} c_4 \widehat{\boldsymbol{u_1}} c_5 \widehat{\boldsymbol{u_1}} c_6 \boldsymbol{u_6} c_7 \widehat{\boldsymbol{u_1}} c_8 \boldsymbol{u_8}$. This results in $\pi \le 10 = \delta$.
   (b) If $\boldsymbol{u_1}$ is empty, the string has the form $\boldsymbol{x} = \boldsymbol{u_0} ccc \boldsymbol{u_3} ccc \boldsymbol{u_6} cc \boldsymbol{u_8}$. This merges the runs corresponding to the mappings $i_1 \to i_2$ with $i_2 \to i_3$, and $i_4 \to i_5$ with $i_5 \to i_6$, and eliminates the possible run corresponding to the mapping $(i_2, i_3) \to (i_4, i_5)$ (unless $\boldsymbol{u_3}$ is empty, in which case the 2 more runs are lost through merging). This gives $\pi \le 7 = \delta$ by just replacing 7 of the c's with new symbols.
4. $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, $(i_3, i_4) \to (i_5, i_6)$, $(i_5, i_6) \to (i_7, i_8)$, and all single-mappings (a total of 10 mappings). By the multi-mappings, the string has the form $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_2} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_6} c \boldsymbol{u_2} c \boldsymbol{u_8}$.
   (a) If $\boldsymbol{u_2}$ is non-empty, replace all instances of a symbol in 3 copies of $\boldsymbol{u_2}$ with new symbols, giving $\pi \le 10 = \delta$.
   (b) If $\boldsymbol{u_2}$ is empty, the runs corresponding to the single mappings $i_2 \to i_3$ and $i_3 \to i_4$ are merged, giving 9 possible runs. If $\boldsymbol{u_1}$ is non-empty, the mapping corresponding to the $i_4 \to i_5$ is also prevented, giving 8 possible runs. (If $\boldsymbol{u_1}$ is empty, 5 possible runs are lost through merging, making the process trivial.) By replacing all instances of some symbol in 1 copy of $\boldsymbol{u_1}$ along with 7 of the c's gives $\pi \le 8 = \delta$.
5. $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, $(i_1, i_2) \to (i_3 \to i_4)$, $(i_3, i_4) \to (i_5 \to i_5)$, $(i_5, i_6) \to (i_7, i_8)$, and all single-mappings (a total of 11 mappings). By the multi-mappings, the string has the form $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_1} c \boldsymbol{u_1} c \boldsymbol{u_1} c \boldsymbol{u_1} c \boldsymbol{u_6} c \boldsymbol{u_1} c \boldsymbol{u_8}$. If $\boldsymbol{u_1}$ is non-empty, replace all instances of a symbol in 5 copies of $\boldsymbol{u_1}$, along with 7 of the c's with new symbols, giving $\pi \le 11 < \delta = 12$. Otherwise, $\boldsymbol{u_1}$ is empty, and 4 single runs are lost through being merged, giving $\pi \le 7 = \delta$.

6. $(i_1, i_2, i_3, i_4) \to (i_5, i_6, i_7, i_8)$ and all single-mappings (a total of 8 mappings), By the quadruple-mapping, the string has the form
$\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_3} c \boldsymbol{u_4} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_3} c \boldsymbol{u_8}$. $\boldsymbol{u_1}$, $\boldsymbol{u_2}$ and $\boldsymbol{u_3}$ cannot all be empty (or several runs are merged), so we replace all instances of a symbol in at least 1 of them, along with 7 of the c's with new symbols. This gives $\pi \leq 8 \leq \delta \leq 10$.
7. $(i_1, i_2, i_3, i_4) \to (i_5, i_6, i_7, i_8)$, $(i_1, i_2) \to (i_3, i_4)$, and all single-mappings (initially 9 runs). Having the double-mapping completely enclosed within one side of the quadruple-mapping, means it exists on the other side of the quadruple-mapping too, so $(i_5, i_6) \to (i_7, i_8)$ also exists. By the multi-mappings, the string has the form $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_1} c \boldsymbol{u_4} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_1} c \boldsymbol{u_8}$. This gives a total of 10 runs.
   (a) If $\boldsymbol{u_1}$ is non-empty, replaces all instances of a symbol in 3 of the copies of it, along with 7 of the c's, giving $\pi \leq 10 = \delta$.
   (b) Otherwise, $\boldsymbol{u_1}$ is empty, giving the structure $\boldsymbol{x} = \boldsymbol{u_0} cc \boldsymbol{u_2} cc \boldsymbol{u_4} cc \boldsymbol{u_2} cc \boldsymbol{u_8}$. However, this eliminates the possibility of the single-mappings $i_2 \to i_3$, $i_4 \to i_5$, and $i_6 \to i_7$ (unless $\boldsymbol{u_2}$ or $\boldsymbol{u_4}$ are empty, in which case 4 or 2 possible runs are lost through merging, respectively). This reduces the number of possible runs to at most 7, and we can achieve $\pi \leq 7 = \delta$ by simply replacing 7 of the c's with new symbols.
8. $(i_1, i_2, i_3, i_4) \to (i_5, i_6, i_7, i_8)$, $(i_2, i_3) \to (i_4, i_5)$, and all single-mappings (a total of 9 mappings): By the multi-mappings, the string has the form
$\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_3} c \boldsymbol{u_2} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_3} c \boldsymbol{u_8}$.
   (a) If $\boldsymbol{u_2}$ is non-empty, replace all instances of a symbol in 2 copies of it, along with 7 of the c's with new symbols, giving $\pi \leq 9 = \delta$.
   (b) Otherwise, $\boldsymbol{u_2}$ is empty, giving $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} cc \boldsymbol{u_3} cc \boldsymbol{u_1} cc \boldsymbol{u_3} c \boldsymbol{u_8}$. This eliminates the possibility of a run corresponding to the single mappings $i_3 \to i_4$ and $i_5 \to i_6$ (unless $\boldsymbol{u_1}$ or $\boldsymbol{u_3}$ are empty; in either case, 2 possible runs are lost through merging), giving $\pi \leq 7$, which is achievable by replacing 7 of the c's.
9. $(i_1, i_2, i_3, i_4) \to (i_5, i_6, i_7, i_8)$, $(i_3, i_4) \to (i_5, i_6)$, and all single-mappings (a total of 9 mappings). By the multi-mappings, the string has the form
$\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_1} c \boldsymbol{u_4} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_1} c \boldsymbol{u_8}$. This same configuration was previously discussed when we assumed it had 10 mappings, so it can be satisfied again in this case.
10. $(i_1, i_2, i_3, i_4) \to (i_5, i_6, i_7, i_8)$, $(i_1, i_2) \to (i_3, i_4)$, $(i_3, i_4) \to (i_5, i_6)$, and all single-mappings (a total of 10 mappings). By the multi-mappings, the string has the form $\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_1} c \boldsymbol{u_4} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_1} c \boldsymbol{u_8}$. This same configuration was previously discussed when we assumed it had 10 mappings, so it can be satisfied again in this case.
11. $(i_1, i_2, i_3, i_4) \to (i_5, i_6, i_7, i_8)$, $(i_2, i_3) \to (i_4, i_5)$, $(i_4, i_5) \to (i_6, i_7)$, and all single-mappings (a total of 10 mappings). By the multi-mappings, $\boldsymbol{u_1} = \boldsymbol{u_5}$, $\boldsymbol{u_2} = \boldsymbol{u_4} = \boldsymbol{u_6}$, and $\boldsymbol{u_3} = \boldsymbol{u_7}$, so the string has the form
$\boldsymbol{x} = \boldsymbol{u_0} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_3} c \boldsymbol{u_2} c \boldsymbol{u_1} c \boldsymbol{u_2} c \boldsymbol{u_3} c \boldsymbol{u_8}$.
   (a) If $\boldsymbol{u_2}$ and one of $\boldsymbol{u_1}$ or $\boldsymbol{u_3}$ is non-empty, replace all instances of a symbol in 1 copy of $\boldsymbol{u_1}$ or $\boldsymbol{u_3}$ and 2 copies of $\boldsymbol{u_2}$, along with 7 of the c's with new symbols, giving $\pi \leq 10 \leq \delta = 10$ or 11.
   (b) If $\boldsymbol{u_2}$ is non-empty, but both $\boldsymbol{u_1}$ and $\boldsymbol{u_3}$ are empty, the string has the form $\boldsymbol{x} = \boldsymbol{u_0} cc \boldsymbol{u_2} cc \boldsymbol{u_2} cc \boldsymbol{u_2} cc \boldsymbol{u_8}$. The possibility of runs corresponding to the mappings $i_2 \to i_3$, $i_4 \to i_5$, and $i_6 \to i_7$ is eliminated, so by replacing 7 of the c's we achieve $\pi \leq 7 = \delta$.

(c) If $\boldsymbol{u_2}$ is empty, the string has the form $\boldsymbol{x} = \boldsymbol{u_0}c\boldsymbol{u_1}cc\boldsymbol{u_3}cc\boldsymbol{u_1}cc\boldsymbol{u_3}c\boldsymbol{u_8}$. The possibility of runs corresponding to the mappings $i_3 \to i_4$ and $i_5 \to i_6$ is eliminated. Since neither $\boldsymbol{u_1}$ nor $\boldsymbol{u_3}$ are empty (or many more possible runs are lost through merging), raising 1 copy of each of these gives $\pi \leq 8 < \delta = 9$.

12. $(i_1, i_2, i_3, i_4) \to (i_5, i_6, i_7, i_8)$, $(i_1, i_2) \to (i_3, i_4)$, $(i_4, i_5) \to (i_6, i_7)$, and all single-mappings (a total of 10 mappings). By the multi-mappings, the string has the form $\boldsymbol{x} = \boldsymbol{u_0}c\boldsymbol{u_1}c\boldsymbol{u_2}c\boldsymbol{u_1}c\boldsymbol{u_2}c\boldsymbol{u_1}c\boldsymbol{u_2}c\boldsymbol{u_1}c\boldsymbol{u_8}$. Therefore, $\boldsymbol{x} = \boldsymbol{u_0}(c\boldsymbol{u_1}c\boldsymbol{u_2})^3c\boldsymbol{u_1}c\boldsymbol{u_8}$, so we can replace the first c only destroying at most a single run ($\pi \leq \delta = 1$).
13. $(i_1, i_2, i_3, i_4) \to (i_5, i_6, i_7, i_8)$, $(i_1, i_2, i_3) \to (i_4, i_5, i_6)$, and all single-mappings (a total of 9 mappings). By the multi-mappings, the string has the form $\boldsymbol{x} = \boldsymbol{u_0}c\boldsymbol{u_1}c\boldsymbol{u_1}c\boldsymbol{u_3}c\boldsymbol{u_1}c\boldsymbol{u_1}c\boldsymbol{u_1}c\boldsymbol{u_3}c\boldsymbol{u_8}$. This merges the possible runs from $i_1 \to i_2$ and $i_2 \to i_3$, as well as $i_4 \to i_5$, $i_5 \to i_6$, and $i_6 \to i_7$, leaving 6 possible runs. Replacing 7 of the c's with new symbols is sufficient to give $\pi \leq 6 < \delta = 7$.
In addition, we can layer up to 2 double-mappings on top of the triple and quadruple mappings, giving a total of 11 mappings. Again, there are at least 3 possible runs lost through merging, giving at most 8 runs. Since $\boldsymbol{u_1}$ and $\boldsymbol{u_3}$ cannot both be empty, we can replace all instances of a symbol in 1 of the copies of $\boldsymbol{u_1}$ or $\boldsymbol{u_3}$. Therefore, $\pi \leq 8 \leq \delta$.
14. $(i_1, i_2, i_3, i_4) \to (i_5, i_6, i_7, i_8)$, $(i_2, i_3, i_4) \to (i_5, i_6, i_7)$, and all single-mappings (a total of 9 mappings). By the multi-mappings, the string has the form $\boldsymbol{x} = \boldsymbol{u_0}c\boldsymbol{u_1}c\boldsymbol{u_1}c\boldsymbol{u_1}c\boldsymbol{u_4}c\boldsymbol{u_1}c\boldsymbol{u_1}c\boldsymbol{u_1}c\boldsymbol{u_8}$. This merges the possible runs corresponding to $i_1 \to i_2$, $i_2 \to i_3$, and $i_3 \to i_4$, along with $i_5 \to i_6$, $i_6 \to i_7$, and $i_7 \to i_8$, decreasing the maximum number of runs by 4. By replacing 7 of the c's with new symbols, we get $\pi \leq 5 < 7 = \delta$.
Once again, we can also layer on up to 2 additional double-mappings on top of the triple- and quadruple-mappings. However, we are still limited to 11 possible runs. Less the 4 possible runs lost to merging gives us $\pi \leq 7 = \delta$ from replacing 7 of the c's with new symbols. □

Remark 19. While the previous lemmas were provided for entries on the main diagonal, the result can be generalized to any entry in column $n-d$ where $\rho_{d'}(n') \leq n'-d'$ for $n'-d' < n-d$. Either $\rho_d(n) \leq n-d$, or no run-maximal $\boldsymbol{x} \in S_d(n)$ has a pair, triple, ..., 8-tuple. The induction hypothesis only requires that all entries to the left of the *unknown* column satisfy the conjecture; there is no restriction within the *unknown* column.

Having proven Proposition 9, we can present the proof of Theorem 8:

Proof. The proof follows directly from Proposition 9. If the conjecture does not hold, let d be the first column for which $\rho_d(2d) > d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. By Proposition 9, $\boldsymbol{x}$ has at least $k = \lceil \frac{7d}{8} \rceil$ singletons, and by Lemma 2 they must all be safe. Let us form $\boldsymbol{y}$ by removing all these safe singletons. This gives a string $\boldsymbol{y} \in S_{d-k}(2d-k)$ violating the conjecture, i.e. $r(\boldsymbol{y}) > d$. $d' = d-k = \frac{d}{8}$ and $d = 8d'$ and $2d-k = 9d'$. Thus we found a $\boldsymbol{y} \in S_{d'}(9d')$ such that $r(\boldsymbol{y}) > 8d'$. □

When investigating a single column, the first counter-example in the column cannot have a singleton, as otherwise the counter-example could be *pushed up*. Nor, by Proposition 9, can it contain a k-tuple for $2 \leq k \leq 8$. Theorem 8 together with these facts give a simplified way to computationally *verify* that the whole column d satisfies the conjecture: *show that there are no counter-examples for* $2 \leq d' \leq \frac{d}{8}$*, and* *only strings with no k-tuples, $1 \leq k \leq 8$, need to be considered when looking for the counter-examples.*

6 Conclusion

The properties presented in this paper constrain the behaviour of the entries in the $(d, n-d)$ table below the main diagonal and in an immediate neighbourhood above the main diagonal. One of the the main contributions lies in the characterization of structural properties of the run-maximal strings on the main diagonal, giving yet another property equivalent with the maximum number of runs conjecture. Not only do these results provide a faster way to computationally check the validity of the conjecture for greater lengths, they indicate a possible way to prove the conjecture along the ideas presented in Proposition 9 and its proof: a first counter-example on the main diagonal could not possibly have a k-tuple for any conceivable k. We were able to carry the reasoning up to $k = 8$, but these proofs are not easy to scale up as the combinatorial complexity increases. The hope and motivation for further research along these lines is that there is a common thread among all these various proofs that may lead to a uniform method ruling out all the k-tuples and thus proving the conjecture, or to exhibit an unexpected counter-example on the main diagonal of the $(d, n-d)$ table. Recent extensions of the parameterized approach shows the unexpected existence of a binary run-maximal string of length 66 containing a substring of four identical symbols *aaaa*, [1]. Similarly, considering squares instead of runs, the approach shows that, among all strings of length 33, no binary string achieves the maximum number of distinct primitively rooted squares [7] .

References

1. A. Baker, A. Deza, and F. Franek: *Computational framework for determining run-maximal strings*, AdvOL-Report 2011/06, McMaster University, 2011.
2. A. Baker, A. Deza, and F. Franek: *Run-maximal strings.* website, 2011, `http://optlab.mcmaster.ca/~bakerar2/research/runmax/`.
3. M. Crochemore and L. Ilie: *Maximal repetitions in strings.* Journal of Computer and System Sciences, 74(5) 2008, pp. 796–807.
4. M. Crochemore, L. Ilie, and L. Tinta: *Towards a solution to the "runs" conjecture.* Lecture Notes in Computer Science, 5029 2008, pp. 290–302.
5. M. Crochemore, L. Ilie, and L. Tinta: *The "runs" conjecture.* website, 2011, `http://www.csd.uwo.ca/faculty/ilie/runs.html`.
6. A. Deza and F. Franek: *A d-step analogue for runs on strings*, AdvOL-Report 2010/02, McMaster University, 2010.
7. A. Deza, F. Franek, and M. Jiang: *Computational framework for determining square-maximal strings*, in Proceedings of Prague Stringology Conference 2012, Prague, Czech Republic, 2012.
8. F. Franek, C. Fuller, J. Simpson, and W. Smyth: *More results on overlapping squares.* to appear.
9. F. Franek and J. Holub: *A different proof of Crochemore-Ilie lemma concerning microruns*, in London Algorithmics 2008: Theory and Practice, College Publications, London, UK, 2009, pp. 1–9.
10. F. Franek, R. Simpson, and W. Smyth: *The maximum number of runs in a string*, in Proceedings of 14th Australasian Workshop on Combinatorial Algorithms AWOCA 2003, Seoul National University, Seoul, Korea, 2008.
11. F. Franek and Q. Yang: *An asymptotic lower bound for the maximal number of runs in a string.* International Journal of Foundations of Computer Science, 19(1) 2008, pp. 195–203.

12. M. Giraud: *Not so many runs in strings*, in LATA 2008, Tarragona, Spain, 2008.
13. C. Iliopoulos, D. Moore, and W. Smyth: *A characterization of the squares in a Fibonacci string*. Theoretical Computer Science, 172 1997, pp. 281–291.
14. R. Kolpakov and G. Kucherov: *Finding maximal repetitions in a word in linear time*, in 40th Annual Symposium on Foundations of Computer Science, 1999, pp. 596–604.
15. R. Kolpakov and G. Kucherov: *On maximal repetitions in words*, in Proc. 12th Intl. Symp. on Fund. of Comp. Sci. 1999, vol. 1684, 1999, pp. 374–385.
16. E. Kopylov and W. Smyth: *The three squares lemma revisited*. Journal of Discrete Algorithms, 11 2012, pp. 3–14.
17. M. Main: *Detecting leftmost maximal periodicities*. Discrete Applied Mathematics, 25 1989, pp. 145–153.
18. W. Matsubara, K. Kusano, H. Bannai, and A. Shinohara: *A series of run-rich strings*. Lecture Notes in Computer Science, 5457 2009, pp. 578–587.
19. W. Matsubara, K. Kusano, A. Ishino, H. Bannai, and A. Shinohara: *New lower bounds for the maximum number of runs in a string*, in Proceedings of Prague Stringology Conference 2008, Prague, Czech Republic, 2008, pp. 140–145.
20. W. Matsubara, K. Kusano, A. Ishino, H. Bannai, and A. Shinohara: *Lower bounds for the maximum number of runs in a string*. website, 2011, `http://www.shino.ecei.tohoku.ac.jp/runs/`.
21. S. Puglisi, R. Simpson, and W. Smyth: *How many runs can a string contain?* Theoretical Computer Science, 401 2008, pp. 165–171.
22. W. Rytter: *The number of runs in a string: Improved analysis of the linear upper bound*. Lecture Notes in Computer Science, 3884 2006, pp. 184–195.
23. F. Santos: *A counterexample to the Hirsch conjecture*. Annals of Mathematics, 176 2012, pp. 383–412.

Festschrift for Bořivoj Melichar
Edited by Jan Holub, Bruce W. Watson and Jan Žďárek
Published by: Prague Stringology Club
Department of Theoretical Computer Science
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9, Praha 6, 160 00, Czech Republic.
URL: http://www.stringology.org/
E-mail: psc@stringology.org Phone: +420-2-2435-9811
Printed by Kresta & Honsnejman, Bělohorská 21, Praha 6 – Břevnov, 169 00, Czech Republic