Journal of Discrete Algorithms 20 (2013) 43–50

Contents lists available at SciVerse ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

A computational framework for determining run-maximal strings

Andrew Baker, Antoine Deza*, Frantisek Franek

Department of Computing and Software, McMaster University, 280 Main St. West, Hamilton, Ontario, Canada

ARTICLE INFO

Article history:
Available online 28 December 2012

Keywords:
Strings
Runs
R-covers
Run-maximality

ABSTRACT

We investigate the function $\rho_d(n) = \max\{\mathbf{r}(x) \mid x$ is a (d,n)-string$\}$, where $\mathbf{r}(x)$ denotes the number of runs in a string x and (d,n)-string denotes a string of length n with exactly d distinct symbols. The notion of an r-cover is presented and discussed with emphasis on the recursive computational determination of $\rho_d(n)$. This notion is used as a key element of a computational framework for an efficient computation of the maximum number of runs. In particular, we were able to determine all previously known $\rho_2(n)$ values for $n \leqslant 60$ in a matter of hours, confirming the results reported by Kolpakov and Kucherov, and were able to extend the computations up to and including $n = 74$. Noticeably, these computations reveal the unexpected existence of a binary run-maximal string of length 66 containing *aaaa*.

1. Introduction

In [2] the notion of an r-cover was introduced as a means to represent the distribution of the runs in a string and thus describe the structure of run-maximal strings. The straightforward assertion from [2] that a run-maximal string has an r-cover – except possibly a single weak point – holds only when the size of the alphabet is not kept fixed. However, the approach can be adapted inductively to handle situations with fixed alphabets and can be used to speed up computations of the maximum number of runs.

We encode a square as a triple (s,e,p) where s is the starting position of the square, e is the ending position of the square, and p is its period. Note that $e = s + 2p - 1$. Similarly, we encode a run as a triple (s,e,p). It is clear from the context whether a triple (s,e,p) encodes a square or a run. Note that the exponent of such a run equals $\lfloor \frac{e-s+1}{p} \rfloor$ and the tail of the run equals the remainder of the division of $(e - s + 1)$ by p. The *leading square* of a run (s,e,p) refers to the square $(s, s + 2p - 1, p)$. The *trailing square* of a run (s,e,p) refers to the square $(e - 2p + 1, e, p)$.

The *join* $x[i_1 \mathinner{..} i_k] \cup x[j_1 \mathinner{..} j_m]$ of two substrings of a string $x = x[1 \mathinner{..} n]$ is defined if $i_1 \leqslant j_1 \leqslant i_k + 1$ and then $x[i_1 \mathinner{..} i_k] \cup x[j_1 \mathinner{..} j_m] = x[i_1 \mathinner{..} \max\{i_k, j_m\}]$, or if $j_1 \leqslant i_1 \leqslant j_m + 1$ and then $x[i_1 \mathinner{..} i_k] \cup x[j_1 \mathinner{..} j_m] = x[j_1 \mathinner{..} \max\{i_k, j_m\}]$. Simply put, the join is defined when the two substrings are either adjacent or overlap. For two encodings (s_1, e_1, p_1) and (s_2, e_2, p_2) of squares in a string x, the join $(s_1, e_1, p_1) \cup (s_2, e_2, p_2)$ represents the join of $x[s_1 \mathinner{..} e_1] \cup x[s_2 \mathinner{..} e_2]$. The alphabet of x is denoted by $\mathcal{A}(x)$, a (d,n)-string refers to a string of length n with exactly d distinct symbols, $\mathbf{r}(x)$ denotes the number of runs in a string x, and $\rho_d(n)$ refers to the maximum number of runs over all (d,n)-strings, i.e. $\rho_d(n) = \max\{\mathbf{r}(x) \mid x$ is a (d,n)-string$\}$. The number of distinct symbols of a string x is denoted as $d(x)$. A *singleton* is a symbol which occurs exactly once in the string under consideration, so a singleton-free string is a string in which each symbol occurs at least twice. A square (s,e,p) is *left-shiftable* if $x[s-1]$ is defined $(s > 1)$, and $x[s-1] = x[s+p-1]$.

* Corresponding author.
E-mail address: deza@mcmaster.ca (A. Deza).

1570-8667/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jda.2012.12.004

Similarly, a square is *right-shiftable* if $x[e+1]$ is defined $(e<n)$ and $x[e+1]=x[s]$. In other words, a square (s,e,p) is left-shiftable exactly when $(s-1,e-1,p)$ is also a square, and is right-shiftable exactly when $(s+1,e+1,p)$ is also a square. To simplify the notation, for the empty string ε we set $\mathbf{r}(\varepsilon)=0$ and $\rho_d(0)=0$.

Considering $\rho(n)$ the maximum number of runs over all strings of length n, i.e. $\rho(n)=\max\{\rho_d(n)\colon\ 1\leqslant d\leqslant n\}$, the investigation of the asymptotic behavior of $\rho(n)/n$ has provided a rich line of research. See [4,8] and references therein for more details and additional results and approaches.

2. Computational approach to runs

The computational framework for determining $\rho_d(n)$ presented in subsequent sections is based on the following approach: We first compute a lower bound of $\rho_d(n)$, denoted as $\rho_d^-(n)$. Then it is enough to restrict our search to the (d,n)-strings potentially satisfying $\mathbf{r}(x)>\rho_d^-(n)$, thus significantly reducing the search space. This section introduces necessary conditions guaranteeing that for a string x, $\mathbf{r}(x)>\rho_d^-(n)$. We show that for a string x to potentially satisfy $\mathbf{r}(x)>\rho_d^-(n)$, it must have an *r-cover* and be $\rho_d^-(n)$*-dense*. Only the r-covered strings are generated and the ones not satisfying the $\rho_d^-(n)$-density are eliminated at the earliest possible stage.

Definition 1. An *r-cover* of a string $x=x[1\mathinner{..}n]$ is a sequence of primitively rooted squares $\{S_i=(s_i,e_i,p_i)\mid 1\leqslant i\leqslant m\}$ so that

(1) none of the S_i's, $1\leqslant i\leqslant m$ is left-shiftable;
(2) $s_i<s_{i+1}\leqslant e_i+1<e_{i+1}+1$ for any $1\leqslant i<m$, i.e. two consecutive squares are either adjacent or overlap without one containing the other;
(3) $\bigcup_{1\leqslant i\leqslant m}S_i=x$;
(4) for any run $R=(s,e,p)$ of x there is an S_i with $1\leqslant i\leqslant n$ containing the leading square of the run R.

A string which has an r-cover is referred to as *r-covered*.

An r-cover with no adjacent squares is referred to as *overlapping r-cover*.

See Fig. 1 for an illustration of an overlapping r-cover.

Lemma 2. *The r-cover of an r-covered string is unique.*

Proof. Let us assume that we have two different r-covers of x, $\{S_i\mid 1\leqslant i\leqslant m\}$ and $\{S'_j\mid 1\leqslant j\leqslant k\}$. We shall prove by induction that they are identical. By Definition 1(4), S_1 is a substring of S'_1 and, by the same argument, S'_1 is a substring of S_1, and thus $S_1=S'_1$. Let the induction hypothesis be $S_i=S'_i$ for $1\leqslant i\leqslant t$. If $\bigcup_{1\leqslant i\leqslant t}S_i=x$, we have $t=m=k$ and we are done. Otherwise consider S_{t+1}. By Definition 1(4), there is S'_v so that S_{t+1} is a substring of S'_v and $v>t$. We need to show that $v=t+1$. If not, then S'_{t+1} is a substring of $\bigcup_{1\leqslant i\leqslant t+1}S_i$ as otherwise S'_{t+1} would contain S_{t+1}, contradicting $v\neq t+1$. Since S'_{t+1} is not a substring of $\bigcup_{1\leqslant i\leqslant t}S_i$, then S'_{t+1} is a substring of S_{t+1}, which in turn is a substring of S'_v, a contradiction. Therefore, S_{t+1} is a substring of S'_{t+1}. Similarly, S'_{t+1} i a substring of S_{t+1} and so $S_{t+1}=S'_{t+1}$, which completes the induction. □

Lemma 3. *Any r-covered string is singleton-free.*

Proof. Let $\{S_j\mid 1\leqslant j\leqslant m\}$ be the r-cover of $x=x[1\mathinner{..}n]$. For any $1\leqslant i\leqslant n$, $x[i]\in S_t$ for some t by Definition 1(3). Since S_t is a square, the symbol $x[i]$ occurs in x at least twice. □

Before defining a $\rho_d^-(n)$*-dense string*, we recall the notion of a *core* of a run introduced in [5]: for a run (s,e,p), its core is the intersection of the set of indices of its leading square $(s,s+2p-1,p)$ and the set of indices of its trailing square $(e-2p+1,e,p)$.

Definition 4.

(a) Let $k_i(x)$ be the number of cores in x containing the position i. Given a (d,n)-string x, the vector $k(x)=(k_1(x),\dots,k_n(x))$ is referred to as the *core vector* of x.
(b) A singleton-free (d,n)-string x is $\rho_d^-(n)$*-dense*, if its core vector $k(x)$ satisfies $k_i(x)>\rho_d^-(n)-\mathbf{r}(x[1\mathinner{..}i-1])-m_i$ for $i=1\mathinner{..}n$, where $m_i=\max\{\rho_{d_2}(n-i)\colon\ d-d_1\leqslant d_2\leqslant\min(n-i,d)\}$ and $d_1=d(x[1\mathinner{..}i-1])$.

Lemma 5. *If a (d,n)-string x is not $\rho_d^-(n)$-dense, then $\mathbf{r}(x)\leqslant\rho_d^-(n)$.*