Annales Univ. Sci. Budapest., Sect. Comp. 39 (2013) 181–202

PRISM COMPLEXITY OF MATRICES

Antal Iványi (Budapest, Hungary)
Zoltán Kása (Târgu Mureş, Romania)

Dedicated to Professor K.-H. Indlekofer
on occasion of his 70th birthday

Communicated by Imre Kátai
(Received December 30, 2012; accepted January 17, 2013)

Abstract. Let d, m, and q be positive integers and $A(q) = \{0, \ldots, q-1\}$ be an alphabet. We investigate a generalization of the well-known subword complexity of d-dimensional matrices containing the elements of $A(q)$. Let $\mathcal{L} = (\mathbf{L}_1, \ldots, \mathbf{L}_m)$ be a list of distinct d-dimensional vectors, where $\mathbf{L}_i = (a_{i1}, \ldots, a_{id})$. The *prism complexity* of a d-dimensional q-ary matrix $\mathcal{M}$ is denoted by $C(d, q, \mathcal{L}, \mathcal{M})$ and is defined as the number of distinct d-dimensional q-ary submatrices, whose permitted sizes are listed in $\mathcal{L}$. We review and extend the earlier results, first of all results concerning maximum complexity of matrices and performance parameters of the construction algorithms.

1. Introduction

Let d, m, n, and q be positive integers, $A(q) = \{0, \ldots, q-1\}$ be an alphabet, ε be the empty matrix, $\mathcal{A}(q,d)^*$ be the set of d-dimensional q-ary matrices, $\mathcal{A}(q,d)^+$ be the set of nonempty d-dimensional q-ary matrices. Let $\mathcal{L} = (\mathbf{L}_1, \ldots, \mathbf{L}_m)$ be a list of d-dimensional vectors, where $\mathbf{L}_i$ gives the size of an $a_{i1} \times \cdots \times a_{id}$ sized submatrix of $\mathcal{M}$.

The $(q, d, \mathcal{L})$-*complexity* (or shortly prism complexity) $C(q, d, \mathcal{L}, \mathcal{M})$ of a matrix $\mathcal{M}$ is defined as follows.

Key words and phrases: Subword complexity, d-complexity, scattered complexity, prism complexity, complexity of arrays, De Bruijn graphs.
2010 Mathematics Subject Classification: 68R15.

Definition 1.1. Let q, d, and $\mathcal{L}$ be given. The $(q, d, \mathcal{L})$-*complexity* of a given d-dimensional q-ary matrix $\mathcal{M}$ is denoted by $C(q, d, \mathcal{L}, \mathcal{M})$ and is defined as the number of distinct submatrices (containing neighboring rows and neighboring columns) of $\mathcal{M}$ whose permitted sizes are given by $\mathcal{L}$, that is

$$C(q, d, \mathcal{L}, \mathcal{M}) = \sum_{i=1}^{m} f(q, d, \boldsymbol{L}_i, \mathcal{M}), \tag{1.1}$$

where $f(q, d, \mathbf{L}_i, \mathcal{M}) = |S(q, d, \mathbf{L}_i, \mathcal{M})|$ and $S(q, d, \mathbf{L}_i, \mathcal{M})$ is the set of the distinct $a_{i1} \times a_{i2} \times \cdots \times a_{id}$ sized submatrices of $\mathcal{M}$.

For example if $q = 3$, $d = 2$, $\mathcal{L} = \{(2, 2), (2, 3)\}$, and

$$\mathcal{M} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}, \tag{1.2}$$

then $C(q, d, \mathcal{L}, \mathcal{M}) = 5$, since $\mathcal{M}$ contains the following five submatrices having permitted size:

$$\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}. \tag{1.3}$$

We remark, that sometimes $\mathcal{M}$ is considered as a periodic matrix. In this case $C(q, d, \mathcal{L}, \mathcal{M}) = 12$, since there are further submatrices: four 2×2 sized matrices

$$\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 2 & 0 \end{pmatrix}, \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}, \tag{1.4}$$

and three 3×3 sized submatrices

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \tag{1.5}$$

If a matrix consists of distinct elements of $A(q)$, then it is called a *rainbow matrix.*

In this paper we deal with some special cases of this new combinatorial complexity measure of q-ary matrices—first of all with the characterization of extremal values and construction of extremal matrices.

The structure of the paper is the following. After this introductory Section 1 one-dimensional matrices (words) are considered in Section 2. Then, in Section 3 some construction algorithms of words having maximum subword complexity are presented. In Section 4 a further one-dimensional complexity measure (d-complexity) is analyzed. In Section 5 we deal with prism complexity of two-dimensional rectangular matrices.

2. Subword complexity of words

If $n \geq 1$, then we consider the $1 \times n$ sized matrices as one-dimensional matrices and call *words* or *sequences.* In this and the further three sections we deal with the complexity of words.

Wackerbauer, Witt, Atmanspacher, Kurths and Scheingraber [118] in 1994, De Luca [26] in 1999, Ilie [65] in 2004, Allouche [4] in 2012 published a survey on the complexity measures of symbol sequences. Allouche [4] classified the complexity measures of words as algorithmic, combinatorial, number-theoretic and inconstancy ones.

Many papers deal with special combinatorial complexity measures of matrices (especially with $1 \times n$ sized matrices). For example arithmetical (Frid [42, 43, 44]), d (Iványi [69], Kása [80, 81]), factor (Ilie [60]), I (Becher and Heiber [13], Kreinovich and Nava [87]), inconstancy (Allouche and Maillard-Teyssier [6]), Lempel–Ziv (Constantinescu and Ilie [21]), joint subword (Jacquet and Szpankowski [75]), linguistic (Popov, Segal, and Trifonov [102], Troyanskaya, Arbell, Koren, Landau, and Bolshoy [114]), palindrome (Allouche, Baake, Cassaigne, and Damanik [5], Anisiu, Anisiu and Kása [7]), pattern (Kamae and Salimov [78], Qu, Rao, Wen, and Xie [103]), repetition (Crochemore and Iliev [23], Ilie, Yu, and Zhang [64]), scattered substring (Okhotin [98]), scattered subword (Fazekas and Nagy [36], Gruber, Holzer and Kutrib [48], Kása [82, 85], Okhotin [98]), square (Ilie [61]), subarray (Iványi [70, 71, 72], Iványi and Tóth [74, 86], Ma [94], MacWilliams and Sloane [95], Siu [109], van Lint, MacWilliams and Sloane [116]), subsequence (Apostolico and Cunial [11], Kuliamin [88]), substring (Elzinga [31], Ilie and Smith [63]), subword (Ehrenfeucht and Rozenberg [28, 29, 30], Heinz [49], Ilie [59], Ilie and Smyth [63], Ivanko [67, 68], Lempel and Ziv [89], Rote [106]), and word complexity (Ilie, Yu and Zhang [65]).

In this paper we deal first of all with the above defined *prism complexity* and its special cases.

In the construction of the matrices having maximum complexity the De Bruijn graphs and their generalizations play important role (these graphs are intensively studied also as possible models of networks, therefore there are many connected papers, e.g. [1, 2, 4, 8, 12, 13, 14, 17, 19, 26, 32, 34, 35, 37, 39, 40, 41, 46, 49, 50, 54, 55, 56, 57, 59, 69, 70, 71, 72, 73, 74, 80, 81, 84, 86, 90, 94, 107, 108, 111, 113, 114, 118].

Let $\mathcal{A}(q,1)^n = A(q)^n$ be the language (set) of all n-length words (sequences) $w = w_1 \dots w_n$ over $A(q)$, $\mathcal{A}(q,1)^+ = A(q)^+$ be the language of nonempty words over $A(q)$.

A *subword* of a q-ary word $w = w_1 \dots w_n \in A(q)^n$ is defined as a contiguous part $w_i \dots w_j$ $(1 \leq i \leq j \leq n)$ of w. We remark that this definition corresponds to $\mathcal{L} = \{(1,1), \dots, (1,n)\}$.

A k-length *subsequence* of w is defined as $w_{j_1} \dots w_{j_k}$, where $1 \leq j_1 < \dots < j_k \leq n$. According to these definitions the empty matrix (word) ε is neither subword not subsequence.

Hunyadvári and Iványi proposed the following join generalization of the subword and subsequence complexity.

Definition 2.1. (Hunyadvári, Iványi 1984 [54, 55, 69]) Let d, r and s be positive integers, $u = u_1 \dots u_r$ and $v = v_1 \dots v_s$ be elements of $A(q)^+$. u is a *d-subword* of $(u \subseteq_d v)$ if there exists a sequence $j_1, \dots, j_r$ with $1 \leq j_1 < \dots < j_r \leq s$ and $j_{i+1} - j_i \leq d$ for $i = 1, \dots, s-1$ such that $u_i = v_{j_i}$ for $i = 1, \dots, r$. If for given d, r and s there exist several such sequences then the lexicographically smallest one belongs to the given d, r and s.

The d-complexity of a word $w \in A(q)^n$ is the number of its distinct d-subwords which can be computed as the sum of the multiplicities of the distinct 1-, ..., n-length d-subwords of w. We remark that in this section d means the *distance parameter* of the complexity, while in the other sections d means the number of dimensions of the considered matrices.

Definition 2.2. (Hunyadvári, Iványi 1984 [54, 55, 69]) For given $w \in A(q)^n$ the *d-complexity* $C(q, d, w)$ of w is

$$C(q, d, w) = \sum_{i=1}^{n} f(q, d, w, i), \tag{2.1}$$

where $f(q, d, w, i) = |S(q, d, w, i)|$, $S(q, d, w, i) = S(u | u \subseteq_d v) \cap A(q)^i$ for $i = 1, \dots, n$.

If in Definition 2.1 and Definition 2.2 $d = 1$, then we get the usual definitions of *subword*, resp. *subword complexity* introduced by Morse and Hendlund in 1938 [97] and redefined later by others [27, 49]. It is worth to remark that some authors (as Shallit [108], and Flaxman, Harrow, Sorkin [38]) count the empty word too as a subword.

In some cases we will suppose that the investigated matrices are *periodic*.

Let $m(q, n)$ denote the maximum number of 1-subwords of $w \in A(q)^n$. w is called *d-complex* if $C(q, 1, w) = m(q, n)$. An infinite word $w = w_1 w_2 \dots$ is called 1-*supercomplex* if $C(q, 1, w^{(k)}) = m(q, n)$ for all prefixes $w^{(k)} = w_1, \dots, w_k$ $(k = 1, 2, \dots)$ of w.

In 1984 Nóra Vörös [117] gave the following bounds.

Theorem 2.1. (Vörös [117]) *If n and q are positive integers and $w \in A(q)^n$, then*

$$n \le C(q,1,w) \le \sum_{i=1}^{n} \min(q^i, n-i+1) \tag{2.2}$$

and the bounds are sharp.

Proof. See [9, 13, 37, 117]. ■

(2.2) was reproved by Ferenczi and Kása [37] in 1999, by Anisiu and Cassaigne in 2004 [9], by Becher and Heiber in 2012 [13]. Theorem 2.1 was reproved in the case $q = 2$ by Shallit in 1993 [108].

In 1984 Nóra Vörös proved the following sufficient and necessary condition for the existence of supercomplex words.

Theorem 2.2. (Vörös [117]) *If $n \ge 1$ then there exists a 1-supercomplex word if and only if $q = 1$ or $q \ge 3$.*

Proof. See [12, 24, 38, 69, 117]. ■

This assertion in 1986 was reproved by Cummings and Wiedemann [24], in 1987 by Iványi [69], in 2004 by Flaxman, Harrow and Sorkin [38], and in 2011 by Becher and Heiber [12].

In 1993 Shallit proved the following closed form for the maximum 1-subword complexity of binary sequences.

Theorem 2.3. (Shallit [108]) *If n is a positive integer then*

$$m(2,n) = \binom{n-k+1}{2} + 2^{k+1} - 2, \tag{2.3}$$

where k is the unique integer such that $2^k + k - 1 \le n < 2^{k+1} + k$.

Proof. See [37, 108]. ■

In 1999 Ferenczi and Kása gave the following closed upper bound for $m(q,n)$.

Theorem 2.4. (Ferenczi, Kása [37]) *If $q \ge 3$ then*

$$m(q,n) \le \binom{n-k+1}{2} + q^{k+1} - 1, \tag{2.4}$$

where k is the unique integer such that $q^k + k - 1 \le n < q^{k+1} + k$, further

$$m(q,n) = \frac{n^2}{2} - \Omega(n \log n). \tag{2.5}$$

Proof. See [37]. ■

In 2004 Anisiu and Cassaigne analyzed *the* 1*-complexity function*

$$h(n,q,i) = \min(q^i, n-i-1) \quad \text{for } i = 1,\ldots,n \tag{2.6}$$

and proved the following theorem.

Theorem 2.5. (Anisiu, Cassaigne [9]) *If n and q are positive integers then there exists a word $w = w_1 \ldots w_n \in A(q)^n$ such that*

$$C(q,n,w) = m(q,n) = \sum_{i=1}^{n} h(n,q,i). \tag{2.7}$$

Proof. See [9]. ■

In 2004 Flaxman, Harrow and Sorkin proved the following upper bound of $m(q,n)$.

Theorem 2.6. (Flaxman, Harrow, Sorkin [38]) *If $q \geq 3$ and $n \geq 1$ are positive integers, then*

$$m(q,n) \leq \binom{n-k+1}{2} + \frac{q^{k+1}-1}{q-1} - 1, \tag{2.8}$$

where $k = \lfloor \log_q n \rfloor$.

Proof. See [38]. ■

Higgins in 2012 published the following upper bound for $m(q,n)$.

Theorem 2.7. (Higgins, [50]) *If q and n are positive integers, then*

$$m(q,n) = \frac{n^2}{2} - O(n \log n). \tag{2.9}$$

Proof. See [50]. ■

The previous seven theorems (Theorem 2.1, 2.2, ..., 2.7) are consequences of the following new theorem.

Theorem 2.8. *If q and n are positive numbers, then*

$$m(q,n) \leq \sum_{i=1}^{n} \min(q^i, n-i+1) \tag{2.10}$$

and

$$\sum_{i=1}^{n} \min(q^i, n-i+1) = \binom{n-k+1}{2} + \frac{q^{k+1}-1}{q-1} - 1, \tag{2.11}$$

where k is the unique integer such that $q^k + k - 1 \leq n < q^{k+1} + k$ and

$$m(q,n) = \frac{n^2}{2} - n\log_q n + n + \Theta(\log^2 n), \tag{2.12}$$

further if $q \neq 2$ then there exists an infinite word $w = w_1 w_2 \ldots$ whose prefixes $w^{(k)} = w_1 \ldots w_k$ have the 1-subword complexity $C(q, 1, w^{(k)}) = m(q,k)$.

Proof. The proof of (2.10) can be found e.g. in [54, 117].

$q^k \leq n - k + 1$ if and only if k is the unique integer such that $q^k + k - 1 \leq \leq n < q^{k+1} + k$. Therefore (2.10) implies

$$m(q,n) = q + \cdots + q^k + (n-k) + \cdots + 1. \tag{2.13}$$

Since

$$q + \cdots + q^k = \frac{q^k - 1}{q - 1}, \tag{2.14}$$

and

$$1 + \cdots + n - k = \binom{n-k}{2}, \tag{2.15}$$

(2.13), (2.14) and (2.15) imply (2.11). ■

In 1999 in the paper [37] appeared the following assertion: if n and q are positive integers, then

$$m(q,n) \leq \frac{(n-k)(n-k+I)}{2} + q^{k+1} - 1 \tag{2.16}$$

where k is the unique integer such that $2^k + k - 1 \leq n < 2^{k+1} + k$.

In 2004 in [38] the following assertion was published: if n and q are positive integers, then

$$m(q,n) = \frac{(n-k)(n-k+1)}{2} + \frac{q^{k+1} - 1}{q - 1} - 1. \tag{2.17}$$

where $k = \lfloor \log n \rfloor$.

The following example shows that (2.16) and (2.17) are not exact. Let $q = 3$ and $n = 2$. The word $w = (01)$ has maximal complexity $|\{0, 1, 01\}| = 3$. In this case (2.16) results 9 and (2.17) gives 4 while the word $w = (0, 1)$ has maximal complexity and so $|\{0, 1, 01\}| = 3$ is the correct value.

In the following we describe further interesting results.

In 1988 Sridhar [111] proved that De Bruijn graphs $B(q, n)$ are q-connected.

In 1991 Li and Zhang [91] counted the number of spanning trees and Eulerian tours. Blażewicz, Formanowicz Kasprzak and Kobler [15] in 2002 proposed a polynomial algorithm to decide whether a directed graph is a De Bruijn graph or the subgraph of a De Bruijn graph with given a (length of vertex names).

Flaxman, Harrow and Sorkin in 2004 [38] proved bounds for average 1-complexity and subsequence complexity. Szpankowski gave a more detailed analysis of average 1-complexity in 2011 [113]. Ivanko also investigated the average 1-complexity [67, 68].

3. Construction of De Bruijn words

In this section several construction algorithms of De Bruijn words are presented.

Definition 3.1. If $q \geq 1$ and $n \geq 1$ then the (q, a)-type De Bruijn word is defined so that it contains all possible q-ary words $w \in A(q)^n$ exactly once as a 1-subword.

This definition implies the length of the (q, a)-type nonperiodic and periodic De Bruijn words.

Corollary 3.1. *The length of a (q, n)-type nonperiodic De Bruijn word is $q^a + a - 1$ and the length of a (q, n)-type periodic De Bruijn word is q^a.*

The first known proof of the existence of $(2, a)$-type De Bruijn words appeared in 1894 and was published by T. Flye-Sante Marie [39]. This assertion was proved again many times afterwards, e.g. in 1946 independently by Good [47] and by De Bruijn [25]. For example Fredricksen, Kessler and Maiorana, Etzion proposed construction algorithms for binary and later q-ary De Bruijn words [34, 40, 41, 104].

3.1. Algorithm Martin

Generating De Bruijn words is a common task with respectable number of algorithms. Let $q \geq 2$ and $A(q) = \{0, \ldots, q-1\}$ be an alphabet. Our goals are to generate from one side a (q, k)-type De Bruijn word, and also to generate all (q, k)-type De Bruijn words for given q and k.

We present here a natural version Natural-Martin [84] of the classical Martin algorithm [96].

We begin with the word 0^a, and add at its right end the greatest possible letter, such that the suffix of length a of the obtained word does not duplicate

a previously occurring subword of length a. The algorithm repeats this until such a prolongation is impossible.

When we cannot continue, a nonperiodic De Bruijn word is obtained, with the length $q^a + a - 1$. In the following algorithm the input is q, the size of the alphabet and a, the pattern size. The *output* is $w = w_1 \ldots, w_{q^a+a-1}$, a (q, a)-type nonperiodic De Bruijn word. The *working variables* are l, which is a logical variable, signalizing whether the last suffix is a new subword or not, and the cycle variables i and k.

The pseudocodes are written according to the conventions described in [22].

NATURAL-MARTIN(q, a)

```
01 for i ← 1 to a                          // line 01–03: initialization of w and i
02     w_i ← 0
03 i ← a
04 repeat                                  // line 04–12: generation of the next word
05     l ← TRUE
06     j ← q
07     while j ≥ 1
08         if b_{i-j+2} w_{i-j+3} ... w_i (j-1) ⊄ w_1 w_2 ... w_i      // not a subword
09             i ← i + 1
10             w_i ← j - 1
11             l ← FALSE
12         else j ← j - 1
13 until l = TRUE          // line 13: if l = TRUE, none of letter can be added
14 return w                                // line 14: return of the result
```

Because this algorithm generates all elements of a De Bruijn sequence of length $q^a + a - 1$, further q and a are independent, the time complexity NATURAL-MARTIN is $\Omega(q^a)$. The more precise characterization of the running time depends on the implementation of line 08. The **repeat** statement is executed $q^a - 1$ times. The **while** statement is executed at most q times for each step of the **repeat.** The test $w_{i-a+2}w_{i-a+3} \ldots w_i a_k \not\subset w_1 w_2 \ldots w_i$ can be made in the worst case in aq^a steps. So, the total number of steps is not greater than aq^{2a+1}, resulting the worst case bound $O(q^{a+1})$. If we use the Knuth-Morris-Pratt string matching algorithm [22], then the worst case running time is $O(q^{2a})$.

3.2. Algorithm QUICK-MARTIN

Algorithm QUICK-MARTIN also generates one-dimensional perfect arrays (De Bruijn words). Its inputs are the alphabet size q and the window size a. Its

output is a q-ary perfect sequence of length q^a. The output begins with a zeros and always continues with the maximal permitted element of the alphabet.

The following effective implementation of Martin algorithm is due to Horváth and Iványi [53].

QUICK-MARTIN(q, a)

```
01 for i = 0 to q^{a-1} - 1                    // line 01–02: initialization of C
02     C[i] = q - 1
03 for i = 1 to a                              // line 03–04: initialization of w and k
04     w[i] = 0
05 for i = a + 1 to q^a                        // line 05–11: generation of w
06     k = w[i - a + 1]
07     for j = 1 to a - 1
08         k = kq + w[i - a + j]
09     w[i] = C[k]
10     C[k] = C[k] - 1
11 return w
```

This algorithm runs in $\Theta(aq^a)$ time.

3.3. Algorithm OPTIMAL-MARTIN

The following implementation [72] of Martin algorithm requires even smaller running time than QUICK-MARTIN.

The *input* of OPT-MARTIN is q: the size of the alphabet; and a: the length of the pattern. The *output* is $w = w_1 \dots w_{q^a}$. The *working variables* are $C = C[0], \dots, C[q^{a-1}]$: the counters belonging to the vertices of the De Bruijn graph; k: the decimal value of the label of the current vertex of the De Bruijn graph. i cycle variable.

OPTIMAL-MARTIN(q, a)

```
01 for i = 0 to q^{a-1} - 1
02     C[i] = q - 1
03 for i = 1 to a
04     w[i] = 0
05 k = 0
06 for i = a + 1 to q^a
07     k = q(k - w[i - a]q^{a-2}) + w[i - 1]
08     w[i] = C[k]
09     C[k] = C[k] - 1
10 return w
```

The running time of any algorithm which constructs a one-dimensional cyclical perfect array is $\Omega(q^a)$, since the sequence contains q^a elements. The running time of Optimal-Martin is $\Theta(q^a)$.

4. Scattered complexity of words

Scattered subwords of a word were defined by Kása in [82] as follows. This definition is *not* a special case of the general definition of prism complexity, since here the list J contains the permitted differences of the indices of the choosed letters, that is $J \subseteq \{1, \cdots, n-1\}$.

Definition 4.1. Let n and q be positive integers, $J \subseteq \{1, \ldots, n-1\}$ and $u = x_1 \ldots x_n \in A(q)^n$. A *$J$-subword* of length s of u is defined as $v = x_{i_1} \ldots x_{i_s}$, where

$i_1 \geq 1$,
$i_{j+1} - i_j \in J$ for $j = 1, \ldots, s-1$,
$i_s \leq n$.

Using Definition 4.1 we can formalize the concept of the scattered subword complexity.

Definition 4.2. For given J the scattered subword complexity (shortly J-complexity) of a word $u \in A(q)^n$ is the number of J-subwords of u.

In the case $1 \leq d \leq n-1$ and of $J = \{1, \ldots, d\}$ the J-subword is the *d-subword* defined in [69], while in the case $J = \{d, \ldots, n-1\}$ is the *super-d-subword* defined in [83]. The corresponding *d-complexity* and *super-d-complexity* are similarly defined.

The scattered subword complexity for rainbow words can be easily computed by a graph method [82]. The letters x_i of the rainbow word are the vertices of the graph, and two vertices, x_i and x_j, are joined by an arc from x_i to a_j if these letters can be neighbors in this order in a scattered subword. The scattered subword complexity is equal to the number of directed paths in this attached graph (here each vertex is considered as a null length path). The number of directed paths in this graph with n vertices can be computed by a Floyd-Warshall type algorithm with worst case comnplexity $\Theta(n^3)$ [82]. If $1 \leq d_1 \leq d_2 \leq n-1$, and the list J is $\{d_1, d_1+1, \ldots, d_2\}$, then the scattered subword complexity (the so called (d_1, d_2)-complexity [84]) can be computed by a linear algorithm [85].

This method combined with the classical Latin square method yields an algorithm by which even the scattered subwords can be obtained [82].

(d_1, d_2)-subwords are related to compositions of integers. Compositions are partitions in which the order of the components does matter. A (d_1, d_2)-*composition* is a restricted composition in which the components are natural numbers from the interval $[d_1, d_2]$.

For example, for the word *abcdefg* the (2,4)-subwords, which begin in a and end in g are: aeg, $aceg$, adg, acg, which correspond to the following compositions in which the components are the distances between the letters in the original word:

$$6 = 4 + 2 = 2 + 2 + 2 = 3 + 3 = 2 + 4.$$

In general, if $a_1 a_{i_1} \dots a_{i_s} a_{n+1}$ is a (d_1, d_2)-subword of the rainbow word $a_1 a_2 \cdots a_{n+1}$, then this subword corresponds to a composition:

$$n = (i_1 - 1) + (i_2 - i_1) + \cdots + (i_s - i_{s-1}) + (n + 1 - i_s).$$

Definition of the (d_1, d_2)-subword can be generalized for rainbow words as we choose letters not only going ahead in the word, but back too, at every step [85].

Definition 4.3. Let n, $d_1 \leq d_2$, q, and s be positive integer numbers, and let $u = x_1 \dots x_n \in A(q)^n$ be a rainbow word over the alphabet $A(q)$. A rainbow word $v = x_{i_1} \dots x_{i_s}$, where

$i_1 \geq 1$,

$d_1 \leq |i_{j+1} - i_j| \leq d_2, \quad$ for $j = 1, \dots, s-1$,

$i_s \leq n$, is an s-length *duplex* (d_1, d_2)-subword of u.

For example *acfbe* and *beadfc* both are duplex (2,4)-subwords, and the duplex subwords are rainbow words too. of the word *abcdef*.

The number of all duplex (d_1, d_2)-subwords of a word is the *duplex* (d_1, d_2)-*complexity* of that rainbow word.

We remark that the term *scattered subword complexity* was used earlier for example by Fazekas and Nagy in 2008, further by Gruber, Holzer and Kutrib in 2009 [48], but they defined the scattered complexity of languages, while Kása [83] defined the scattered complexity of words.

Fazekas and Nagy in 2008 [36], Gruber, Holzer and Kutrib in 2007 and in 2009 [48], Kása in 2011 and 2012 [82, 85], Okhotin in 2010 [98] investigated the scattered subsequence complexity of words.

Flaxman, Harrow and Sorkin in 2004 [38], Szpankowski in 2001 [113] characterized the average subsequence complexity.

5. Prism complexity of two-dimensional rectangular matrices

In this section we consider the complexity of the usual q-ary matrices, that is the $d = 2$ case of the prism complexity.

Definition 5.1. Let $q,\ a,\ b,\ A,\ B$ be positive integers., $\mathcal{M}$ an $A \times B$ sized, periodic q-ary matrix, $A \geq a$, $B \geq b$, $a \leq b$, $\mathcal{L} = \{(a, b)\}$ and $q^{ab} = AB$. $\mathcal{M}$ is called (q, a, b, M, N)-type De Bruijn matrix, if it contains every possible $a \times b$ sized q-ary submatrix exactly once.

We remark, that De Bruijn matrices are called also perfect maps [105, 99, 100] or De Bruijn tori [57].

The first result belongs to Reed and Stewart [105] proving the existence of a 4×4 sized periodic binary matrix containing all possible 2×2 sized binary submatrix exactly once. A connected empirical result is due to Péter Selmeczi, whose program gives that there are 800 5×5 sized non periodic binary array containing the 2×2 sized submatrices exactly once, 256 such matrices having identical first and fifth columns and 32 such matrices having additionally identical first and fifth rows too, so the problem solved by Reed and Stewart has 32 solutions.

Ma [94] in 1984, Fan Fan, Ma and Siu in 1985 proposed an algorithm which constructs a binary matrix containing every $a_1 \times a_2$ sized binary matrix as submatrix exactly once.

Iványi in 1989 proved the following theorem.

Theorem 5.1. (Iványi, 1989 [70]) *If $q \geq 1$, $a \geq 1$, $b \geq 1$, then there exist matrices A and B such that there exists a (q, a, b, A, B)-type de Bruijn matrix.*

Proof. (Sketch) (a) If $q = 1$ then the assertion is straightforward.

(b) If $a = 1$ then algorithm Bruijn produces the required word.

(c) If $a = b = 2$ then see [74] or the monograph of Knuth [86].

(d) If $q \geq a \geq 3$ and $b \geq 2$ then we construct $\mathcal{M}$ as follows.

(d1) We use Optimal-Martin with input data q and a and the w output will the first column of $\mathcal{M}$.

(d2) The ith ($i = 2, \ldots, q^{a(b-1)}$ column of $\mathcal{M}$ is generated shifting cyclically downwards its $(i-1)$th column by w_{i-1}, where $w_1 \ldots w_s$ ($s = q^{a(b-1)} - 1$) is the output of Optimal-Martin for alphabet size q^b.

(e) The case $b = 2$ and $a \geq 3$ is similar to (c).

(f) Since in the cases (d) and (e) the height q^a of the constructed matrix is a divisor of the sum of the shift sizes and any two $a \times b$ submatrices are distinct

(either their first columns or at least one of their corresponding shift sizes are different), the construction is correct (see also [74]). ■

It is worth to remark that the size of the constructed De Bruijn matrix is $q^a \times q^b$. The papers [99, 100] contain detailed analysis of the possible sizes of De Bruijn matrices as the function of the sizes of the window. sizes a and b.

The analog of (2.2), containing the bounds for a one-dimensional word w is the following assertion.

Lemma 5.1. *Let q, a, b, A, B be positive integers. If $\mathcal{M}$ is an $A \times B$ sized q-ary matrix and $\mathcal{L} = \{(a, b)\}$ then we get the following bounds:*

$$1 \leq C(q, 2, \mathcal{L}, \mathcal{M}) \leq \min(q^{ab}, AB). \tag{5.1}$$

The lower bound is sharp. In some cases the upper bound is also sharp.

Proof. If $\mathcal{M}$ is a homogeneous word (that is $q = 1$ or $\mathcal{M}$ contains only one element of a larger alphabet) then the $a \times b$ sized submatrices are identical so the complexity equals to 1. Since $A \geq a$ and $B \geq b$ therefore the complexity is always positive, so the lower bound is sharp.

The complexity of $\mathcal{M}$ is not larger than the number of possible dstinct $a \times b$ sized submatrices, and also is not larger than the number of elements of $\mathcal{M}$ (since each element can be the left upper element of one $a \times b$-sized submatrix), therefore the upper bound is correct. According to Theorem 5.1 the constructed $q^a \times q^b$ sized $\mathcal{M}$ contains dstinct submatrices, therefore in e.g. in this case the upper bound also is sharp. ■

For q large enough we can form a crossbow matrix [69] in which the elements are dstinct. For such matrices we get the following simple consequence of Lemma 5.1 and Theorem 5.1.

Corollary 5.1. *If $q \geq 1$, $A \geq a$, $B \geq b$,, $\mathcal{M}$ is an $A \times B$ sized q-ary matrix, $\mathcal{L} = \{(a, b)\}$ and $q^{ab} \geq AB$ then*

$$C(q, 2, \mathcal{L}, \mathcal{M}) = AB. \tag{5.2}$$

Proof. At the given conditions the minimum in (5.1) equals to AB. ■

Acknowledgement. The authors thank the referee for the proposed useful corrections.

References

[1] **Alekseyev, M.A. and P.A. Pevzner,** Colored de Bruijn graphs and the genome halving problem. *IEEE/ACM Trans Comput. Biology Bioinf.,* **4(1)** (2007), 98–107.

[2] **Alhakim, A.,** Spans of preference functions for de Bruijn sequences, *Discrete Appl. Math.,* **160(7-8)** (2012), 992–998.

[3] **Alhakim, A. and M. Akinwande,** A recursive construction of nonbinary de Bruijn sequences, *Des. Codes Cryptogr.,* **60(2)** (2011), 155–169.

[4] **Allouche, J.-P.,** Surveying some notions of complexity for finite and infinite sequences, in (eds. K. Matsumoto, S. Akiyama, K. Fukuyama, H. Nakada, H. Sugita, A. Tamagawa) *Functions in Number Theory and Their Probabilistic Aspects* (Kyoto, 2010, Kôkyrôku Bessatsu Series), 2012.

[5] **Allouche, J.-P., M. Baake, J. Cassaigne and D. Damanik,** Palindrome complexity, *Theoret. Comput. Sci.* (Selected papers in honor of Jean Berstel), **292(1)** (2003), 9–31.

[6] **Allouche, J.-P. and L. Maillard-Teyssier,** Inconstancy of finite and infinite sequences, *Theoret. Comput. Sci.,* **412(22)** (2011), 2268-2281.

[7] **Anisiu, M.-C., V. Anisiu and Z. Kása,** Total palindrome complexity of finite words *Discrete Math.*, **310(1)** (2010), 109–114.

[8] **Anisiu, M.-C, Z. Blázsik and Z. Kása,** Maximal complexity of finite words, *Pure Math. Appl. (Pu.M.A.),* **13(1-2)** (2002), 39–48. See also as *arXiv* arXiv:1002.2724, 2010, 8 pages.

[9] **Anisiu, M.-C and J. Cassaigne,** Properties of the complexity function for finite words, *Revue d'analyse numérique et de théorie de l'approximation,* **33(2)** (2004), 123–139.

[10] **Anisiu, M.-C. and A. Iványi** Two-dimensional arrays with maximal complexity, *Pure Math. Appl. (Pu.M.A.),* **17(3-4)** (2006), 197–204.

[11] **Apostolico, A., F. Cunial,** The subsequence composition of a string *Theoret. Comput. Sci.* **410(43)** (2009), 4360–4371.

[12] **Becher, V. and P.A. Heiber,** On extending de Bruijn sequences, *Inform. Proc. Letters,* **111(18)** (2011), 930–932.

[13] **Becher, V. and P.A. Heiber,** A linearly computable measure of string complexity, *Theoret. Comp. Sci.,* **438** (2012), 62–73.

[14] **Berstel, J. and D. Perrin,** The origins of combinatorics on words, *European J. Combin.,* **28(3)** (2007), 966–1022.

[15] **Błażewicz, J., F. Formanowicz, M. Kasprzak and D. Kobler,** On the recognition of de Bruijn graphs and their induced subgraphs, *Discrete Math.* **245(1-3)** (2002), 81–92.

[16] **Bond, J. and A. Iványi,** Modelling of interconnection networks using De Bruijn graphs, in: (ed. A. Iványi) *Third Conference of Program Designers* (Budapest, July 1–3, 1987), 75–87.

[17] **Çalik, C., M.S. Turan and F. Özbudak,** On feedback functions of maximum length nonlinear feedback shift registers, *IEICE Trans. Fundamental,* **E-93A(6)** (2010), 1226-1231.

[18] **Chan, Y.K., M.K. Siu and P. Ton,** Two-dimensional binary arrays with good autocorrelation, *Inform. and Control,* **42(2)** (1979), 125–130.

[19] **Chung, F., P. Diaconis and R. Graham,** Universal cycles for combinatorial structures, *Discrete Math.,* **110(1-3)** (1992), 43–59.

[20] **Compeau, Ph.E.C., P.A. Pevzner and G. Tesler,** How to apply de Bruijn graphs to genome assembly, *Nature Biotechnology* **29** (2011), 987–991.

[21] **Constantinescu, S. and L. Ilie,** The Lempel–Ziv complexity of fixed points of morphisms, *SIAM J. Discrete Math.,* **21(2)** (2007), 466–481.

[22] **Cormen, T.H., Ch.E. Leiserson, R.L. Rivest and C. Stein,** *Introduction to Algorithms* Third edition, The MIT Press/McGraw Hill, Cambridge/New York, 2009.

[23] **Crochemore, M. and L. Ilie,** Maximal repetitions in strings, *J. Comput. System Sci.,* **74(5)** (2008), 796–807.

[24] **Cummings, L.J. and D. Wiedemann,** Embedded de Bruijn sequences *Congr. Numer.,* **53** (1986), 155–160.

[25] **De Bruijn, N.G.,** A combinatorial problem, *Nederl. Akad. Wetensch. Proc.,* **49** (1946), 758–764. See also *Indagationes Math.* **8,** (1946) 46–467.

[26] **De Luca, A.** On the combinatorics of finite words, *Theoret. Comput. Sci.* (WORDS Conference, Rouen, 1997), **218(1)** (1999), 13–39.

[27] **Ehrenfeucht, A., K.P. Lee and G. Rozenberg,** Subword complexities of various classes of deterministic developmental languages without interactions, *Theoret. Comput. Sci.,* **1(1)** (1975), 59–75.

[28] **Ehrenfeucht, A. and G. Rozenberg,** On the subword complexities of homomorphic images of languages, *RAIRO Theor. Inform. Appl.,* **16** (1982), 303–316.

[29] **Ehrenfeucht, A. and G. Rozenberg,** On the subword complexities of locally catenative D0L-languages, *Inform. Process. Letters,* **16(1)** (1983), 7–9.

[30] **Ehrenfeucht, A. and G. Rozenberg,** On the subword complexities of m-free D0L-languages, *Inf. Process. Letters,* **17(3)** (1983), 121–124.

[31] **Elzinga, C.H.,** Complexity of categorical time series, *Sociological Methods & Research* **38(3)** (2010), 463–481.

[32] **Elzinga, C.H., S. Rahmann and H. Wang,** Algorithms for subsequence combinatorics, *Theoret. Comput. Sci.,* **409(3)** (2008), 394–404.

[33] **Etzion, T.,** Constructions for perfect maps and pseudorandom arrays. Coding techniques and coding theory, *IEEE Trans. Inform. Theory,* **34(5, part 2)** (1988), 1308–1316.

[34] **Etzion, T. and Z. Lempel,** Construction of De Bruijn sequences of minimal complexity, *IEEE Trans. Inform. Theory,* **30(5)** (1984), 705–709.

[35] **Fan, C.T., S.M. Fan, S.L. Ma and M.K. Siu,** On the Bruijn arrays, *Ars. Combin.,* **19A** (1985), 205–213.

[36] **Fazekas, Sz. Zs. and B. Nagy,** Scattered subword complexity of non-primitive words, *J. Autom., Lang. Comb.* **13(3-4)** (2008), 233–247.

[37] **Ferenczi, S. and Z. Kása,** Complexity for finite factors of infinite sequences, *Theoret. Comput. Sci.,* **21(1)** (1999), 177–195.

[38] **Flaxman, A., A.W. Harrow and G.B. Sorkin,** Strings with maximally many distinct subsequences and substrings, *Electron. J. Combin.,* (2004(1)), Research Paper 8, 10 pages.

[39] **Flye, Sainte-Marie, C.,** Question 48, *L'intermédiaire des mathématiciens,* **1** (1894), 107–110.

[40] **Fredricksen, H.,** A survey of full length nonlinear shift register cycle algorithms, *SIAM Rev.,* **24(2)** (1982), 195–221.

[41] **Fredricksen, H. and J. Maiorana,** Necklaces of beads in k colors and k-ary de Bruijn sequences, *Discrete Math.,* **23(3)** (1978), 207–210.

[42] **Frid, A.,** Sequences of linear arithmetical complexity, *Theoret. Inform. Sci.* (Combinatorics on Words), **339(1)** (2005), 68–87.

[43] **Frid, A.,** *Combinatorial Complexity Characteristics of Infinite Words, Languages and Permutations* (Russian), Dissertation, Novosibirsk, 2005.

[44] **Frid, A.,** On possible growths of arithmetical complexity, *RAIRO Inf. Theoret. Appl.,* **40(3)** (2006), 443–458.

[45] **Frid, A. and L. Zamboni,** On automatic infinite permutations, *arXiv,* arXiv:1109:6131v1 [csDM], 2011, 11 pages.

[46] **Games, R.A.,** A generalized recursive construction for de Bruijn sequences, *IEEE Trans. Inform. Theory,* **29(6)** (1983), 843–850.

[47] **Good, I.J.,** Normal recurring decimals. *J. London Math. Soc.* **21** (1946), 167–169.

[48] **Gruber, H., M. Holzer and M. Kutrib,** More on the size of Higman–Haines sets: effective constructions, *Fund. Inf.,* **91(1)** (2009), 105–121.

[49] **Heinz, M.,** Zur Teilwortcomplexität für Wörter und Folgen über einem endlichen Alphabet, *EIK*, **13(1-2)** (1977), 27–38.

[50] **Higgins, P.M.,** Burrows-Wheeler transformations and de Bruijn worwds, *Theoret. Comput. Sci.,* **457** (2012), 128–136.

[51] **Higman, G.,** Ordering by divisibility in abstract algebras, *Proc. London Math. Soc.,* **3(2)** (1952), 326–336.

[52] **Horton, J.D. and G.M. Nonay,** Self-orthogonal Hamilton path decompositions, *Discrete Math.,* **97(1-3)** (1991), 251–264.

[53] **Horváth, M. and A. Iványi** Growing perfect cubes. *Discrete Math.* **308(19)** (2008), 4378–4388.

[54] **Hunyadvári, L. and A. Iványi,** On some complexity measures of words, in: (ed I. Peák) *Automata, Languages and Mathematical Systems* Karl Marx University of Budapest, 1984, 7–16.

[55] **Hunyadvári, L. and A. Iványi,** On the subsequence conplexity of words, in: (ed. A. Iványi) *Conference on Young Programmers and Mathematicians* (Budapest, 23–27 May 1984), ELTE, Budapest, 1984, 8–16.

[56] **Hurlbert, G. and G. Isaak,** On the de Bruijn torus problem, *J. Combin. Theory, Ser. A,* **64(1)** (1993), 50–62.

[57] **Hurlbert, G. and G. Isaak,** A meshing technique for de Bruijn tori, *Contemp. Math.* (Jerusalem Combinatorics'93 Conference), **178** (1994), 153–160.

[58] **Hurlbert, G.H., C.J. Mitchell and K.G. Paterson,** On the existence of de Bruijn tori with two by two windows, *J. Comb.Theory Ser. A.,*, **76** (1996), 213–230.

[59] **Ilie, L.,** A simple proof that a word of length n has at most $2n$ distinct squares, *J. Combin. Theory Ser. A,* **112(1)** (2005), 163–164.

[60] **Ilie, L.,** Combinatorial complexity measures for strings, in: *Recent Advances in Formal Languages and Applications* (Studies in Computational Intelligence (SCI)), **25**, Springer-Verlag, Berlin, Heidelberg, 2006, 149–170.

[61] **Ilie, L.,** A note on the number of squares of a word, *Theoret. Comput. Sci.,* **380(3)** (2007), 373–376.

[62] **Ilie, L., P. Ochem and J. Shallit,** A generalization of repetition threshold, *Theoret. Comput. Sci.,* **345(2-3)** (2005), 359–369.

[63] **Ilie, L. and W.F. Smyth,** Minimum unique substrings and maximal repeats, *Fund. Inform.,* **110(1-4)** (2011), 183–195.

[64] **Ilie, L., S. Yu and K. Zhang,** Repetition complexity of words, in (ed. O.H. Ibarra and L. Zhang) *Computing and Combinatorics,* LNCS **2387** (2002), 320–329.

[65] **Ilie, L., S. Yu and K. Zhang,** Word complexity and repetitions in words, *Internat. J. Found. Comput. Sci.,* **15** (2004), 41–55.

[66] **Isaak, G.,** Constructions for higher dimensional perfect multifactors, *Aequationes Math.,* (2002), **64(1-2)** (2002), 70–88.

[67] **Ivanko, E.E.,** Exact approximation of average subword complexity of finite random words over finite alphabet, *Trudy Inst. of Mathematiki i Mehaniki of UO of RAN*, **14(4)** (2008), 185–189.

[68] **Ivanko, E.E.,** An improvement of subword complexity, *Random Operators Stoch. Equ.,* **19(2)** (2011), 187–196.

[69] **Iványi, A.,** On the d-complexity of words, *Annales Univ. Sci. Budapest., Sect. Comput.,* **8** (1987), 69–90.

[70] **Iványi, A.,** Construction of infinite de Bruijn arrays, *Discrete Appl. Math.,* **22(3)** (1989), 289–293.

[71] **Iványi, A.,** Construction of three-dimensional perfect matrices, *Ars Combinatoria,* (Twelfth British Combinatorial Conference, Norwich, 1989), **29C** (1990), 33–40.

[72] **Iványi, A.,** Perfect arrays, in (ed. A. Iványi) *Algorithms of Informatics, Vol. 3,* AnTonCom, Budapest, 2011, 1311–1349.

[73] **Iványi, A. and J. Madarász,** Perfect hypercubes. *Electron. Notes Discrete Math.* **38(1)** (2011), 475–480.

[74] **Iványi, A. and Z. Tóth,** Existence of de Bruijn words, in (ed. I. Peák) *Second Conference on Automata, Languages and Programming Systems* (Salgótarján, 1988), **DM, 88-4,** Karl Marx Univ. Econom., Budapest, 1988, 165–172.

[75] **Jacquet, P. and W. Szpankowski,** Joint string complexity for Markov sources, in: *23rd Intern. Meeting on Probabilistic, Combinatorial, and Asymptotic Methods for the Analysis of Algorithms* (AofA'12 , DMTCS Proc. AQ), 2012, 303–322.

[76] **Janson, S., S. Lonardi and S. Szpankowski,** On the average sequence complexity, in: (ed. S. C. Sahinalp et al.) *Combinatorial Pattern Matching*, LNCS **3109,** Springer Verlag, Berlin, Heidelberg, 2004, 74–88.

[77] **Janson, S., S. Lonardi and S. Szpankowski,** On the average sequence complexity, *Theor. Comp. Sci.,* **326** (2004), 213–227.

[78] **Kamae, T. and P.V. Salimov,** On maximal pattern complexity of some automatic words, *Ergod. Th. & Dynam. Sys.,* **359** (2010), 15–27.

[79] **Kása, Z.,** Computing the d-complexity of words by Fibonacci-like sequences, *Studia Univ. Babeş-Bolyai, Math.,* **35(3)** (1990), 49–53.

[80] **Kása, Z.,** On two conjectures on word complexity in: *Second Joint Conference on Modern Mathematics* (Ilieni, Romania, June 3–7, 1997), 44–44.

[81] **Kása, Z.,** On the d-complexity of strings, *Pure Math. Appl.* (Modern Applied Analysis, Ilieny, 1997), **9(1-2)** (1998), 119–128.

[82] **Kása, Z.,** On scattered subword complexity, *Acta Univ. Sapientiae, Inform.,* **3(1)** (2011), 127–136.

[83] **Kása, Z.,** Super-d-complexity of finite words, in (eds. H. F. Pop and A. Bege): *MACS 2010: 8th Joint Conference on Mathematics and Computer Science, Selected Papers,* Novadat, Győr, 2011, 257–266.

[84] **Kása, Z. and M.-C. Anisiu,** Complexity of words, in: (ed. A. Iványi) *Algorithms of Informatics, Vol. 3*, AnTonCom, Budapest, 2011, 1259–1334.

[85] **Kása, Z. and Z. Kátai,** Scattered subwords and compositions of integers, *Acta Univ. Sapientiae, Inform.,* **4(2)** (2012), 225–236.

[86] **Knuth, D.E.,** *The Art of Computer Programming, Vol. 4A,* Addison-Wesley, Upper-Saddle River, NJ, 2011.

[87] **Kreinovich, V. and J. Nava,** I-complexity and discrete derivative of logarithms: a symmetry-based explanation, *J. Uncertain Systems,* **6(2)** (2012), 118–121.

[88] **Kuliamin, V.V.,** Test sequence construction using minimum information on the tested system, *Programming Comp. Software,* **31(6)** (2005), 301-309.

[89] **Lempel, A. and J. Ziv,** On the complexity of finite sequences, *IEEE Trans. Inform. Theory,* **22(1)** (1976), 75–81.

[90] **Levé, F. and P. Séébold,** Proof of a conjecture on word complexity, *Bull. Belg. Math. Soc. Simon Levin,* **8(2)** (2001), 277–291.

[91] **Li, X. and F. Zhang,** On the numbers of spanning trees and Eulerian tours in generalized de Bruijn graphs, *Discrete Math.,* **94(3)** (1991), 189–197.

[92] **Liu, M.** Homomorphisms and automorphisms of 2-D de Bruijn-Good graphs, *Discrete Math.,* **85(1)** (1990), 105–109.

[93] **Lothaire, T.,** *Encyclopedia of Mathematics and its Applications*, (ed. P. Flajolet, M. Ismail, E. Lutwak) **104** *Applied Combinatorics on Words,* Cambridge University Press, Cambridge, 2005.

[94] **Ma, S.L.,** A note on binary arrays with a certain window property, *IEEE Trans. Inform. Theory,* **30(5)** (1984), 774–775.

[95] **MacWilliams, S.J. and N.J.A. Sloane,** Pseudo-random sequences and arrays, *Proc. IEEE.,* **64(12)** (1976), 1715–1729.

[96] **Martin, M.H.,** A problem in arrangements, *Bull. AMS,* **40** (1934), 859–864.

[97] **Morse, M. and G.A. Hedlund,** Symbolic dynamics, *Amer. J. Math.,* **60** (1938), 815–866.

[98] **Okhotin, A.,** On the state complexity of scattered substrings and superstrings, *Fundamenta Inf.,* **XX** (2010), 1–14.

[99] **Paterson, K.G.,** New classes of perfect maps I , *J. Comb. Theory, Series A,* **73(2)** (1996), 302–334.

[100] **Paterson, K.G.,** New classes of perfect maps II, *J. Comb. Theory, Series A,* **73(2)** (1996), 335–345.

[101] **Pemantle, R. and M. Wilson,** Twenty combinatorial examples of asymptotics derived from multivariate generating functions, *arXiv,* arXiv:math/0512548v2 [math.CO], 2007.

[102] **Popov, O., D.M. Segal and E.N. Trifonov,** Linguistic complexity of protein sequences as compared to texts of human languages, *Biosystems,* **38(1)** (1996), 65–74.

[103] **Qu, Y.-H., H. Rao, Z.-Y. Wen and Y.-M. Xue,** Maximal pattern complexity of higher dimensional words, *J. Combin. Theory Ser. A* **117(5)** (2010), 489–506.

[104] **Ralston, A.,** A new memoryless algorithm for de Bruijn sequences, *J. Algorithms,* **2(1)** (1981), 50–62.

[105] **Reed I.S. and R.M. Stewart,** Note on the existence of perfect maps, *IRE Trans. Inf. Theory,* **8(1)** (1962), 10–12.

[106] **Rote, G.,** Sequences with subword complexity $2n$, *J. Number Theory,* **46(2)** (1994), 196–213.

[107] **Ruskey, F., J. Savada and A. Williams,** De Bruijn sequences for fixed-weight binary strings, *SIAM Journal Discrete Math.,* **26** (2012), 605–617.

[108] **Shallit, J.O.,** On the maximum number of distinct factors in a binary string, *Graphs Comb.,* **9(2-4)** (1993) 197–200.

[109] **Siu, M. K.,** The combinatorics of binary arrays, *J. Stat. Planning Inference,* **62(1)** (1997), 103–113.

[110] **Smyth, W.F.,** Computing regularities in strings: A survey, *European J. Comb.,* **34(1)** (2013), 3–14.

[111] **Sridhar, M.A.,** On the connectivity of the De Bruijn graph, *Inf. Proc. Letters,* **27(6)** (1988), 315–318.

[112] **Stong, R.,** Hamilton decompositions of Cartesian products of graphs, *Discrete Math.,* **90** (1991), 169–190.

[113] **Szpankowski, W.,** *Average Case Analysis of Algorithms on Sequences,* John Wiley, New York, NY, 2001.

[114] **Troyanskaya, O.G., O. Arbell, Y. Koren, G.M. Landau and A. Bolshoy,** Sequence complexity profiles of prokaryotic genomic sequences: a fast algorithm for calculating linguistic complexity. *Bioinformatics,* **18** (2002), 679–688.

[115] **Van Aardenne-Ehrenfest, T. and N.G. de Bruijn,** Circuits and trees in oriented linear graphs. *Simon Stevin,* **28** (1951), 203–217.

[116] **van Lint, J.H., F.J. MacWilliams, N.J.A. Sloane,** On pseudo-random arrays. *SIAM J. Appl. Math.* **36(1)** (1979), 62–72.

[117] **Vörös, N.,** On the complexity measures of symbol-sequences, in (ed. A. Iványi): *Conference of Young Programmers and Mathematicians* (Budapest, 23–27 May 1984), Eötvös Loránd University, Faculty of Sciences, Budapest, 1984, 43–50.

[118] **Wackerbauer, R., A. Witt, H. Atmanspacher, I. Kurths and H. Scheingraber,** A comparative classification of complexity measures, *Chaos, Solitons & Fractals,* **4(1)** (1994), 133–173.

[119] **Wagner, D.G.,** On the perfect one-factorization conjecture, *Discrete Math.,* **104** (1992), 211–215.

[120] **Weisstein, E.W.,** De Bruijn sequences, 2012.
http://mathworld.wolfram.com/deBruijnSequence.html
[121] **Wikipedia,** De Bruijn sequences, 2012.
http://en.wikipedia.org/wiki/De_Bruijn_sequence

Antal Iványi
Faculty of Informatics
Eötvös Loránd University
H-1117 Budapest, Pázmány P. sétány 1/C
Hungary
tony@inf.elte.hu

Zoltán Kása
Department of Mathematics and Informatics
Sapientia Hungarian University of Transylvania
RO 540485 Târgu Mureş, Şoseaua Sighişoarei, 1C.
Romania
kasa@ms.sapientia.ro