Discrete Applied Mathematics 22 (1988/89) 289-293
North-Holland

COMMUNICATION

CONSTRUCTION OF INFINITE DE BRUIJN ARRAYS

Antal IVÁNYI

Department of General Computer Science, Eotvös Loránd University, Bogdánfy Ödon u. 10/a, H-1117 Budapest, Hungary

Received 14 October 1988

Communicated by L. Lovász

We construct a periodic array containing every k-ary $m\times n$ array as a subarray exactly once. Using the algorithm SUPER (which for $k\geq 3$ generates an infinite k-ary sequence whose beginning parts of length k^m, $m=1,2,\ldots$, are de Bruijn sequences) we also construct infinite $k^m\times\infty$ k-ary arrays in which each beginning part of size $k^m\times k^{mn-m}$, $n=1,2,\ldots$, as a periodic array, contains every k-ary $m\times n$ array exactly once.

Keywords. k-ary perfect maps, k-ary infinite de Bruijn matrices

1. Introduction

This paper deals with the construction of finite and infinite de Bruijn arrays (perfect maps and supercomplex arrays). Such arrays are connected with frequency allocation for multibeam satellites [2], picture coding and processing [11] and complexity of nucleotid sequences [3]. Algorithms for constructing de Bruijn sequences are described in [1, 4, 10, 14].

Definition 1.1. Let $k\geq 2$, m, n, M, N be positive integers, $X=\{0,1,\ldots,k-1\}$. A (k,m,n,M,N) *array* (or de Bruijn array) is a periodic $M\times N$ array with elements from X and $m\leq M$, $n\leq N$, $M\times N=k^{mn}$ in which each of the different k-ary $m\times n$ arrays appears exactly once.

Definition 1.2. Let $k\geq 2$, m and M be positive integers, $X=\{0,1,\ldots,k-1\}$. A (k,m,M)*-array* (or infinite de Bruijn array) is a k-ary infinite $M\times\infty$ array with elements from X whose beginning parts of length k^{mn}/M as periodic arrays are $(k,m,n,M,k^{mn}/M)$-arrays for $n=1,2,\ldots$.

We remark that $(k,1,n,1,k^n)$-arrays are de Bruijn sequences and $(k,1,1)$-arrays are infinite de Bruijn sequences.

The following existence results are known. For any m and n there are M and

N such that a $(2, m, n, M, N)$-array exists [5, 6]. If k is odd, then for any m a $(k, m, 2, k^m, k^m)$-array exists [5]. No $(2, 1, 1)$-arrays exist [1, 7, 13], but for $k \geq 3$ $(k, 1, 1)$-arrays exist [1, 7, 13]. Constructions of (k, m, m, M, M)-arrays are presented in [6] for $k = 2$ and in [9] for $m = 2$.

2. Algorithms

The algorithm BRUIJN walks in a de Bruijn graph $B(k, n)$ defined as follows: the vertex set is X^n and the edge set is X^{n+1} in such a way that $\kappa_1 \kappa_2 \ldots \kappa_{n+1} \in X^{n+1}$ determines an edge going from the vertex $\kappa_1 \kappa_2 \cdots \kappa_n \in X^n$ to the vertex $\kappa_2 \kappa_3 \ldots \kappa_{n+1} \in X^n$.

If $m \geq n$, then any sequence $q = \gamma_1 \gamma_2 \ldots \gamma_m$ ($\gamma_i \in X$, $i = 1, \ldots, m$) determines a directed path in $B(k, n)$ beginning at the vertex $\gamma_1 \gamma_2 \ldots \gamma_n$, going through the vertices $\gamma_2 \gamma_3 \cdots \gamma_{n+1}, \ldots, \gamma_{m-n} \gamma_{m-n+1} \cdots \gamma_{m-1}$ and ending at the vertex $\gamma_{m-n+1} \gamma_{m-n+2} \cdots \gamma_m$. BRUIJN finds an Eulerian circuit p of $B(k, n)$ [12, p. 413].

Algorithm BRUIJN.

Input. The alphabet size t ($t \geq 2$) and the pattern size n ($n \geq 1$).

Output. A $(t, 1, n, 1, t^n)$-array p.

Step 1. If $n = 1$, then let $p := 01 \ldots (t-1)$ and Stop.

Step 2. Let $p := \kappa_1 \kappa_2 \ldots \kappa_n = 00 \ldots 0$.

Step 3. If $p = \kappa_1 \kappa_2 \ldots \kappa_s$ and $s = t^n + n + 1$, then go to Step 7.

Step 4. If $p = \kappa_1 \kappa_2 \ldots \kappa_s$, $\kappa_s = i$ and the last vertex $V = \kappa_{s-n+1} \kappa_{s-n+2} \cdots \kappa_s$ has at least one unused outgoing edge, then let κ_{s+1} be the first suitable element in the sequence $i, i+1, \ldots, t-1, 0, 1, \ldots, i-1$ and go to Step 3.

Step 5. (Now the last vertex V in p has no unused outgoing edges.) Let us find and insert into p a suitable circuit seeking its start vertex going back in p from V and constructing it using Step 4 [12].

Step 6. Go to Step 3.

Step 7. (Now $p = \kappa_1 \kappa_2 \ldots \kappa_s$ and $s = t^n + n + 1$.) Let $r := t^n$, $p := \kappa_1 \kappa_2 \ldots \kappa_r$ and Stop.

The algorithm SUPER generates infinite de Bruijn sequences using the following characteristics of $B(k, n)$:

(a) There is a ono-to-one mapping among the Euler circuits of $B(k, n)$ and the Hamiltonian circuits of $B(k, n+1)$ [10].

(b) If $n \geq 3$ and $k \geq 1$, then any Hamiltonian circuit p of $B(k, n)$ can be continued in order to get an Eulerian circuit q of $B(k, n)$ [7].

Algorithm SUPER.

Input. The alphabet size t ($t \geq 3$).

Output. A $(t, 1, 1)$-array p.

Step 1. Let $p := 01 \cdots (t-1)$ and $n := 1$.

Step 2. Continue p in order to get an Eulerian circuit of $B(t,n)$ using Steps 3-7 of Algorithm BRUIJN.

Step 3. Let $n := n+1$ and go to Step 2.

3. Construction results

Theorem 3.1. *For any $k \geq 2$, $m \geq 1$ and $n \geq 1$ there are M and N such that a (k,m,n,M,N)-array P exists.*

Proof (Sketch). (a) If $\min(m,n)=1$, then Algorithm BRUIJN generates the required array.

(b) If $n=m=2$, then see [9].

(c) If $n \geq 3$ and $m \geq 2$, then we construct P as follows.

(c.1) If the input parameters of BRUIJN are the alphabet size k and the pattern size m, then the output as a column will be the first column of P.

(c.2) The ith, $i=2,\ldots,k^{m(n-1)}$ column of P is generated shifting cyclically downwards its $(i-1)$th column by b_{i-1} where

$$b_1 b_2 \ldots b_s, \quad s = k^{m(n-1)} - 1$$

is the output of BRUIJN for $t = k^m$.

(d) The case $n=2$, $m \geq 3$ is similar to case (c).

(e) Since in the cases (c) and (d) the height (k^m) of the constructed array is a divisor of the sum of the shift sizes and any two $m \times n$ subarrays are different (either their first columns or at least one of their corresponding shift sizes are different), the construction is correct [9]. □

Theorem 3.2. *For any odd $k \geq 3$ and $m \geq 1$ and also for any even $k \geq 2$ and $m \geq 3$ there is an M such that a (k,m,M)-array S exists.*

Proof (Sketch). (a) The output of BRUIJN as a column for alphabet size k and pattern size m gives the first column of S.

(b) The ith, $i=2,3,\ldots$ column of S is generated by cyclically downward shifting of its $(i-1)$th column by b_{i-1}, where $b_1 b_2 \ldots$ is the output of SUPER for alphabet size k^m.

(c) To prove the correctness of this construction it is enough to show that k^m divides the sum of the relative shift sizes and any two $m \times n$ subarrays are different in the $k^m \times k^{mn-m}$ beginning parts for $n=1,2,\ldots$ [9]. □

We remark that if k is even and $m=2$, then the algorithm used in the proof of Theorem 3.2 generates a k-ary $k^m \times \infty$ array whose beginning parts of length k^{mn-m} as periodic arrays are (k,m,n,k^m,k^{mn-m})-arrays for $n=1,3,4,5,\ldots$ $(n \neq 2)$.

Theorem 3.2 does not cover the case when k is even and $m \leq 2$. No $(2, 1, M)$- and $(2, 2, M)$-arrays exist [8]. If $s \geq 2$, then $(2s, 1, 1)$-arrays [1, 7] and $(2s, 2, 2s^2)$-arrays [8] exist.

4. An example

If $t = 3$ and $m = 2$, then Algorithm BRUIJN gives $p = 001122021$. If $t = k^m = 9$, then the 81-length beginning part of the output of SUPER is:

$$q = b_1 b_2 \ldots b_{81}$$
$$= 01234\ 56788\ 00224\ 46681\ 13355\ 77036\ 04714\ 82583\ 72615\ 05162$$
$$73840\ 63074\ 17528\ 53186\ 42087\ 65432\ 1.$$

In this case the sequence of the absolute shift sizes $p = c_1 c_2 \cdots c_{81}$ is defined as

$$0 \leq c_j \leq 8, \quad c_j \equiv b_1 + \cdots + b_{81}, \qquad j = 1, 2, \ldots, 81.$$

Table 1 shows the first 81 column of a (3, 2, 9)-array and under the columns the corresponding relative (q) and absolute (p) shift sizes:

Table 1

00101	11010	00010	12022	21000	22000	00220	11020	12102	00210	20120	01001	20110	10111	01000	10001	0
00022	02200	11102	11120	22021	01002	00012	12102	20220	00222	22220	02001	12021	12212	21001	22122	1
11012	02101	11101	20102	02111	20111	11201	11201	22212	11021	01201	12112	01021	21222	12111	21112	1
11100	10011	22210	20221	20102	10110	11100	20210	01001	11200	20021	10112	00102	20020	02112	00200	2
22102	12012	22210	01210	12202	01220	22010	02200	20200	22120	10212	22220	10122	00202	00222	20202	2
22211	21122	00021	21000	01210	01221	22011	21001	10110	22011	01102	21222	11210	21121	12220	11011	0
00210	20120	22221	12201	00012	12001	00121	10211	01001	00001	01000	00001	21202	11010	11002	01210	2
22020	00202	11102	02111	10221	12222	22122	00112	01021	22102	12012	20220	22001	02000	20221	02120	1
11221	21221	00022	00012	11120	20112	11202	01022	12122	11112	12111	11110	02210	02101	20110	12021	0

$q =$
01234 56788 00224 46681 13355 77036 04714 82583 72615 05162 73840 63074 17528 53186 42087 65432 1
$p =$
01361 63108 88137 28545 60384 20030 04237 68436 46340 05635 36500 60072 31687 36763 70086 38368 0

References

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

[2] J Denes and A D Keedwell, Frequency allocation for a mobile radio telephone system, IEEE Trans Communications 36 (1988) 765–767

[3] W Ebeling and R Feistel, Physik der Selbstorganisation und Evolution (Akademie-Verlag, Berlin, 1982)

[4] T. Etzion, Algorithms to construct *m*-ary de Bruijn sequences, J. Algorithms 7 (1986) 331–340.
[5] T. Etzion, Constructions for perfect maps and pseudo-random arrays, IEEE Trans. Inform. Theory (to appear).
[6] C.T. Fan, S.M. Fan, S.L. Ma and M.K. Siu, On de Bruijn arrays, Ars Combin. 19A (1985) 205–213.
[7] A Iványi, On the *d*-complexity of words, Annnales Univ Sci. Budapest. Sect. Comput 8 (1987) 69–90
[8] A. Iványi, Construction of supercomplex matrices, in: A. Iványi, ed., Fourth Conference of Program Designers (Eotvos Loránd University, Budapest, 1988).
[9] A. Iványi and Z. Tòth, Existence of de Bruijn words, in: I. Peák, ed., Automata, Languages and Programming Systems (Karl Marx University of Economics, Budapest, 1988).
[10] L. Lovász, Combinatorial Problems and Exercises (Akadémiai Kiadò, Budapest, 1979).
[11] S.L. Ma, A note on binary arrays with a certain window property, IEEE Trans. Inform. Theory 30 (1984) 774–775.
[12] C H. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity (Prentice-Hall, Englewood Cliffs, NJ, 1982).
[13] N. Voròs, On the complexity of symbol-sequences, in: A. Iványi, ed., Conference of Young Programmers and Mathematicians (Eotvos Loránd University, Budapest, 1984), 43–50.
[14] S. Xie, Notes on de Bruijn sequences, Discrete Appl. Math. 16 (2) (1987) 157–177.