Constructing de Bruijn Sequences with Co-Lexicographic Order: The k-ary Grandmama Sequence

Patrick Baxter Dragon Oscar I. Hernandez Joe Sawada Aaron Williams Dennis Wong

April 3, 2017

Abstract

A k-ary de Bruijn sequence of order n is a circular k-ary string of length k^n which contains every k-ary string of length n exactly once as a substring. It is well-known that a k-ary de Bruijn sequence of order n can be constructed by concatenating the aperiodic prefixes of the k-ary necklaces of length n in lexicographic order. In this article we prove that an alternate de Bruijn sequence is created by replacing lexicographic order with co-lexicographic order. We also provide a simple successor rule for generating each successive symbol in $O(\mathrm{n})$-time.

1 Introduction

Let $\Sigma_k = \{0, 1, 2, \ldots, k-1\}$ be an alphabet of k symbols for a positive integer k. Let Σ_k^n be the set of k-ary strings of length $n > 0$. A *de Bruijn sequence of order* n is a circular k-ary string $\mathcal{D} = d_1 d_2 \cdots d_{k^n}$ which contains each element of Σ_k^n as substring exactly once. For example, $\mathcal{D} = 001021122$ is a de Bruijn sequence for $k = 3$ and $n = 2$ since its substrings of length n are successively $00, 01, 10, 02, 21, 11, 12, 22, 20$ (where the last substring "wraps around") and this list contains each string in Σ_k^n exactly once.

De Bruijn sequences are often used in education of discrete mathematics and theoretical computer science, including *Concrete Mathematics* textbook by Graham, Knuth, and Patashnik [10]) and *The Art of Computer Programming* by Knuth [12]. Historically, they have been referred to by many different names and have had many interesting applications. For example, Stein offered an interesting survey in the 'Memory Wheels' chapter in *Mathematics: The Man-Made Universe* [23].

The most well-known and well-studied de Bruijn sequence for all values of k and n is the lexicographically least de Bruijn sequence. By *lexicographically least* we mean the first in lexicographic order when the de Bruijn sequence is viewed linearly as a k-ary string of length k^n starting from 0^n. For example, the lexicographically least de Bruijn sequence for $k = 3$ and $n = 2$ is 001021122, which was seen earlier. Natural subsequences of the lexicographically least sequence have been shown to be de Bruijn sequences for interesting subsets of strings (see Moreno [15], Au [1], and Sawada, Williams, and Wong [19, 20]). Cooper and Heitsch determined its 'discrepancy' for $k = 2$ [3] which is important for pseudorandom bit generation.

The lexicographically-least de Bruijn sequence was first constructed (up to equivalence) by Martin in 1934 [13] for all n and k using a greedy algorithm. Knuth refers to Martin's construction as the "Granddaddy of all de Bruijn sequence constructions" [12]. We follow this playful nomenclature by using *Granddaddy* to refer to the de Bruijn sequence itself. The term *Ford sequence* is also used to describe the lexicographically-least de Bruijn sequence due to Ford's independent work [6]. Alternatively, the sequence can be constructed

directly and efficiently using the *FKM construction*[1], which is named after the work of Friedricksen, Kessler, and Maiorana [9, 8]. Informally, the FKM construction concatenates the aperiodic prefixes of necklaces in lexicographic order. An example of the construction appears in Figure 1. The existence of this elegant construction is made more interesting by the fact that the decision problem associated with finding lexicographically least Eulerian circuit is NP-hard by Matamala and Moreno [14].

The main result of this article is that the FKM construction provides a de Bruijn sequence when co-lexicographic order is used instead of lexicographic order, and we refer to the resulting sequence as the *Grandmama* de Bruijn sequence. This result is quite surprising due to its close relationship with the prominent Granddaddy de Bruijn sequence, and a comment immediately following Definition 4 helps explain why it was not previously discovered. The result also suggests that a larger family of de Bruijn sequences can be generated using generalizations of the FKM construction. Prior to this article the only other member of this family was the 'cool-daddy' de Bruijn sequence for fixed-weight binary strings [18].

This article also provides a *successor rule* that generates the Grandmama sequence one symbol at a time. More specifically, the successor rule maps each k-ary string of length n to the symbol that follows it in the sequence, and it runs in $O(n)$-time. This represents only the second successor rule that works for all k and n with the first being published recently [22]. Two other successor rules have been published for $k = 2$, including the Granddaddy successor by Fredricksen [7] and another by Huang [11].

A preliminary version of this article was presented at the 12th Latin American Theoretical Informatics Symposium (LATIN 2016) conference in Ensenada, México [4]. The preliminary version focused solely on the correctness of the necklace concatenation construction for $k = 2$. This article generalizes those reulsts to k-ary strings and also contributes directly to the binary result with a successor rule for that case.

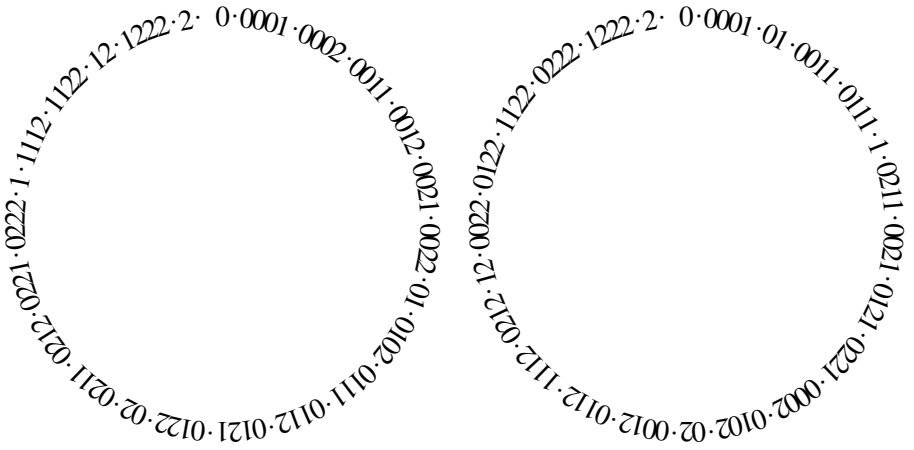
Before completing this introductory section we point out that the Granddaddy and Grandmama sequences are equivalent for some small values of n and k. Formally, a de Bruijn sequence $d_1 d_2 \cdots d_{k^n}$ is *equivalent* to any de Bruijn sequence that can be obtained by applying a finite sequence of rotations, reversals, and symbol remappings. In [4] it was shown that the Granddaddy and Grandmama sequences are equivalent for $k = 2$ and $n \leq 5$ but are not equivalent for $k = 2$ and $n > 5$. Similarly, when $k = 3$ and $n = 2$ the Granddaddy de Bruijn sequence $= 001021122$ and the Grandmama de Bruijn sequence $= 001102122$ are equivalent by reversal and substituting $0 \leftrightarrow 2$. However, the two constructions are not equivalent for $n \geq 3$ and $k \geq 3$ with $000100201101202102211121222$ and $000101110210020121120221222$ being the non-equivalent results for $n = 3$ and $k = 3$. In the article we further differentiate the two sequences by clarifying the difference between the lexicographic and co-lexicographic orders of necklaces.

2 Background Concepts

This section defines lexicographic and co-lexicographic order and discusses the concept of lex and co-lex successors. Necklaces, aperiodic prefixes, and Lyndon words are then formally introduced.

Throughout this article we use Greek letters for strings and substrings and Roman letters with optional subscripts for individual symbols. For example, $\alpha = a_1 a_2 a_3 a_4 a_5 = \beta \cdot x \cdot \gamma$ would denote a string of length five with a prefix β and a suffix γ that is separated by an individual symbol x (where $\cdot$ denotes concatenation).

[1] It had long been observed that the FKM construction produced the lexicographically-least de Bruijn sequence, although this observation was not formally proven until recently with [1], Moreno and Perrin [16], and [19, 20] appearing around the same time.



Necklace α	Aperiodic Prefix $\mathsf{ap}(\alpha)$
0000	0
0001	0001
0002	0002
0011	0011
0012	0012
0021	0021
0022	0022
0101	01
0102	0102
0111	0111
0112	0112
0121	0121
0122	0122
0202	02
0211	0211
0212	0212
0221	0221
0222	0222
1111	1
1112	1112
1122	1122
1212	12
1222	1222
2222	2

← lexicographic order

Necklace α	Aperiodic Prefix $\mathsf{ap}(\alpha)$
0000	0
0001	0001
0101	01
0011	0011
0111	0111
1111	1
0211	0211
0021	0021
0121	0121
0221	0221
0002	0002
0102	0102
0202	02
0012	0012
0112	0112
1112	1112
0212	0212
1212	12
0022	0022
0122	0122
1122	1122
0222	0222
1222	1222
2222	2

← co-lexicographic order

Figure 1: Constructing the Granddaddy de Bruijn sequence (left) and the Grandmama de Bruijn sequence (right) for $n = 4$ and $k = 3$.

2.1 Lexicographic and Co-lexicographic Order

Lexicographic and co-lexicographic order are total orders applied from left-to-right and right-to-left, respectively. The two concepts are formally defined below.

Definition 1. *The string $\alpha = a_1a_2\cdots a_n$ comes before $\beta = b_1b_2\cdots b_m$ in* lexicographic order *if one of the following two conditions hold:*

- *α is a strict prefix of β (i.e. $\alpha = b_1b_2\cdots b_n$ and $m > n$), or*
- *there exists an i such that $a_1a_2\cdots a_{i-1} = b_1b_2\cdots b_{i-1}$ and $a_i < b_i$.*

Definition 2. *The string $\alpha = a_1a_2\cdots a_n$ comes before $\beta = b_1b_2\cdots b_m$ in* co-lexicographic order *if one of the following two conditions hold:*

- *α is a strict suffix of β (i.e. $\alpha = b_{m-n+1}b_{m-n+2}\cdots b_m$ and $m > n$), or*
- *there exists an i such that $a_{n-i+1}a_{n-i+2}\cdots a_n = b_{m-i+1}b_{m-i+2}\cdots b_m$ and $a_{n-i} < b_{m-i}$.*

For example, 001122 and 112211 are ordered oppositely with respect to lexicographic and co-lexicographic order. Similarly, 012 and 0012 are also ordered oppositely by the two orders. For convenience, we use the shorthand *lex* and *co-lex* for lexicographic and co-lexicographic, respectively. We also note that lex and co-lex order of any $\mathbf{S} \subseteq \Sigma_k^n$ depends only on the second bullets of Definitions 1 and 2[2].

2.2 Lex and Co-lex Successors

If $\alpha \in \mathbf{S}$, then the *lex successor* of α with respect to $\mathbf{S}$ is the string that immediately follows α in the lexicographic order of $\mathbf{S}$; the lex successor is undefined if α is the last string in lex order in $\mathbf{S}$. The *co-lex successor* is defined analogously.

If $\mathbf{S} \subseteq \Sigma_k^n$ and β is the lex successor of α with respect to $\mathbf{S}$, then Definition 1 implies that we can write the two strings as follows:

$$\alpha = a_1a_2\cdots a_{i-1}\cdot x\cdot a_{i+1}a_{i+2}\cdots a_n$$
$$\beta = a_1a_2\cdots a_{i-1}\cdot y\cdot b_{i+1}b_{i+2}\cdots b_n$$

where $x, y \in \Sigma_k$ with $x < y$. More specifically, α is the last string in lex order with prefix $a_1a_2\cdots a_{i-1}x$, and β is the first string in lex order with prefix $a_1a_2\cdots a_{i-1}y$. We refer to $a_1a_2\cdots a_{i-1}$ as the *common prefix* of α and β, i as the *increasing index* and x as the *increasing symbol* of α and y as the *increased symbol* of β, and $b_{i+1}b_{i+2}\cdots b_n$ as the *minimum suffix* of β.

Similarly, if $\mathbf{S} \subseteq \Sigma_k^n$ and β is the co-lex successor of α with respect to $\mathbf{S}$, then by Definition 2 we can write the two strings as follows:

$$\alpha = a_1a_2\cdots a_{i-1}\cdot x\cdot a_{i+1}a_{i+2}\cdots a_n$$
$$\beta = b_1b_2\cdots b_{i-1}\ \cdot y\cdot\ a_{i+1}a_{i+2}\cdots a_n$$

where $x, y \in \Sigma_k$ with $x < y$. More specifically, α is the last string in co-lex order with suffix $xa_{i+1}a_{i+2}\cdots a_n$, and β is the first string in co-lex order with suffix $ya_{i+1}a_{i+2}\cdots a_n$, and i is chosen to be as small as possible. We refer to $a_{i+1}a_{i+2}\cdots a_n$ as the *common suffix* of α and β, i as the *increasing index* and x as the *increasing symbol* of α and y as the *increased symbol* of β, and $b_1b_2\cdots b_{i-1}$ as the *minimum prefix* of β.

The terms presented in this subsection will be used again in Section 3.1.

[2]The remark after Definition 4 does require the full generality of Definitions 1 and 2.

2.3 Necklaces

We define an equivalence relation on the set of k-ary strings of length n as follows: two strings are equivalent if one is a rotation of the other. The equivalence classes with respect to this relation are called *necklace classes*. For example, the necklace class containing the string 20102010 is $\{20102010, 01020102, 10201020, 02010201\}$. The preferred representatives for necklace classes are the lexicographicaly least strings in each necklace class, and are called *necklace representatives*. For convenience, we will simply call them *necklaces*. For example, the necklace for the class given above is 01020102. Let $\mathbf{N}_k(n)$ be the set of k-ary necklaces of length n. Two different orders of $\mathbf{N}_3(4)$ can be seen in Figure 1. To simplify discussion in later sections, let $\mathbf{N'}_k(n) = \mathbf{N}_k(n) - \{(k-1)^n\}$ and $\mathbf{N''}_k(n) = \mathbf{N}_k(n) - \{0^n, (k-1)^n\}$. Now we prove three lemmas about the prefixes and suffixes of necklaces in $\mathbf{N}_k(n)$.

Lemma 1. *There is a necklace in $\mathbf{N}_k(n)$ with suffix $\gamma \in \Sigma_k^m$ if and only if $0^{n-m}\gamma \in \mathbf{N}_k(n)$.*

Proof. If $0^{n-m}\gamma \in \mathbf{N}_k(n)$ then $\mathbf{N}_k(n)$ contains a necklace with suffix γ. For the other direction suppose that γ is the suffix of a necklace in $\mathbf{N}_k(n)$ and $\alpha\gamma \in \mathbf{N}_k(n)$. If $\alpha \neq 0^{n-m}$, then γ must not include a substring 0^{n-m} and it must not end in 0. Therefore, $0^{n-m}\gamma \in \mathbf{N}_k(n)$ since its prefix of length 0^{n-m} is strictly smaller in lex order than its other (circular) substrings of length $n - m$. □

Lemma 2. *Suppose $x\gamma$ and $z\gamma$ are both suffixes of necklaces in $\mathbf{N}_k(n)$ for $\gamma \in \Sigma_k^m$ and $x, z \in \Sigma_k$ with $x < z$. Then $y\gamma$ is also the suffix of some necklace in $\mathbf{N}_k(n)$ for all $x \leq y \leq z$.*

Proof. By Lemma 1 it is sufficient to consider prefixes of 0s. With this in mind we prove a slightly stronger result: If $0^{n-m-1}z\gamma \in \mathbf{N}_k(n)$ for $z > 0$, then $0^{n-m-1}y\gamma \in \mathbf{N}_k(n)$ where $y = z-1$. This is true since $0^{n-m-1}y$ must be strictly smaller in lex order than any (circular) substring of length $n - m$ in $0^{n-m-1}y\gamma$. □

Lemma 3. *If $0^i x\gamma \in \mathbf{N}_k(n)$ where $x > 0$, then $0^i(x-1)\gamma \in \mathbf{N}_k(n)$.*

Proof. If $0^i x\gamma \in \mathbf{N}_k(n)$ where $x > 0$, then by Lemma 1 we have $0^{i+1}\gamma \in \mathbf{N}_k(n)$. Thus by Lemma 2 we have $0^i(x-1)\gamma \in \mathbf{N}_k(n)$.

□

2.4 Periodic, Aperiodic Prefixes, and Lyndon Words

A string α is *periodic* if $\alpha = \beta^r$ for a positive integer $r > 1$, where exponentiation denotes repetition. Otherwise, the string is *aperiodic*. For example, $20102010 = (2010)^2$, and so it is periodic. In contrast, the string 2010201 is aperiodic.

Given a string α, its *aperiodic prefix*, denoted $\mathsf{ap}(\alpha)$, is its shortest prefix β such that $\alpha = \beta^r$ for some positive integer r. We define the *period* of α to be the length of its aperiodic prefix, denoted $|\mathsf{ap}(\alpha)|$. For example, if $\alpha = 20102010$ then $\mathsf{ap}(\alpha) = 2010$ and so α has period four. Note, if α is aperiodic, then $\mathsf{ap}(\alpha) = \alpha$, and so it has a period of $|\mathsf{ap}(\alpha)| = |\alpha|$.

A *Lyndon word* is an aperiodic necklace. We will discuss Lyndon words again in Section 4.1.

$\alpha \in \mathbf{N'}_3(4)$	0000	0001	0101	0011	0111	1111	0211	0021	0121	0221	0002	0102	0202	0012	0112	1112	0212	1212	0022	0122	1122	0222	1222
common suffix		01	1	11	111	111	1	21	21		02	02	2	12	112	12	212	2	22	122	22	222	222
increasing symbol	0	0	0	0	0	1	1	0	1	1	0	1	0	0	0	1	0	1	0	0	1	0	1
increased symbol	1	1	1	1	1	2	2	1	2	2	1	2	1	1	1	2	1	2	1	1	2	1	2
minimum prefix	000	0	00	0		0	00	0	0	000	0	0	00	0		0		00	0		0		
$\beta = \mathsf{succ}(\alpha)$	0001	0101	0011	0111	1111	0211	0021	0121	0221	0002	0102	0202	0012	0112	1112	0212	1212	0022	0122	1122	0222	1222	2222

Figure 2: Illustrating co-lex successors for $\mathbf{N}_3(4)$. Each column shows a necklace $\alpha \in \mathbf{N'}_3(4)$ being converted into its co-lex successor $\beta = \mathsf{succ}(\alpha)$. The common suffix of α and β, the increasing symbol in α, the increased symbol in β, and the minimum prefix in β are all shown. Definition 3 and Theorem 1 assert that the increasing and increased symbol always differs by 1 and the minimum prefix is always 0s.

3 Necklaces in Co-Lexicographic Order

In this section we consider the co-lex order of k-ary necklaces of length n. We begin by defining the co-lex successor and comparing it to the lex successor. We also prove a property of periodic strings in co-lex order.

3.1 Co-Lex Successor

Recall that $\mathbf{N'}_k(n) = \mathbf{N}_k(n) - \{(k-1)^n\}$. We define a function $\mathsf{succ} : \mathbf{N'}_k(n) \to \mathbf{N}_k(n)$ below, and will then prove that it maps necklaces in $\mathbf{N'}_k(n)$ to their co-lex successor in $\mathbf{N}_k(n)$.

Definition 3. *If $\alpha = a_1 a_2 \cdots a_n \in \mathbf{N'}_k(n)$, then $\mathsf{succ}(\alpha) = 0^{i-1}(a_i + 1)a_{i+1}a_{i+2}\cdots a_n$ where i is the minimum value with $0^{i-1}(a_i + 1)a_{i+1}a_{i+2}\cdots a_n \in \mathbf{N}_k(n)$.*

Successive applications of the function succ for necklaces in $\mathbf{N}_3(4)$ are illustrated in Figure 2. The terminology used in the proof was introduced in Section 2.2.

Theorem 1. *If $\alpha \in \mathbf{N'}_k(n)$, then $\mathsf{succ}(\alpha)$ is the co-lex successor of α with respect to $\mathbf{N}_k(n)$.*

Proof. Let $\alpha = a_1 a_2 \cdots a_n$ be a necklace in $\mathbf{N'}_k(n)$ and let $\beta = b_1 b_2 \cdots b_n = succ(\alpha)$. Let i be the increasing index and $b_{i+1}b_{i+2}\cdots b_n = a_{i+1}a_{i+2}\cdots a_n$ be the common suffix of α and β. Also let $x = a_i$ be the increasing symbol of α and $y = b_i$ be the increased symbol of β. By Lemma 2 the increased symbol of β is $y = x + 1$. By Lemma 1 the minimum prefix of β is 0^{i-1}. Thus $\beta = 0^{i-1}yb_{i+1}b_{i+2}\cdots b_n$. By Definition 3 the value i is the smallest possible value such that $0^{i-1}yb_{i+1}b_{i+2}\cdots b_n \in \mathbf{N}_k(n)$. Therefore, $\beta = \mathsf{succ}(\alpha)$. □

Theorem 1 implies that the co-lex successor of $\alpha \in \mathbf{N}_k(n)$ is fully determined by its increasing index. More specifically, the successor is obtained by incrementing the increasing symbol and filling the successor's prefix with 0s. On the other hand, it is interesting to note that the lex successor of $\alpha \in \mathbf{N}_k(n)$ presents a completely different challenge. In that case the increasing index is always the index of the rightmost symbol that is less than k, however, the minimum suffix depends on the prefix and is non-trivial to compute. For a better understanding of necklaces in lexicographic order refer to Ruskey, Savage, and Wang [17].

3.2 Periodic Strings

When proving our main result in Section 4 we will need to pay special attention to periodic necklaces. The following lemma ensures that periodic necklaces share certain suffixes and prefixes with the necklace before and after in co-lex order, respectively. To complement our $\mathsf{succ}(\alpha)$ notation, we let $\mathsf{pred}(\alpha)$ be the *co-lex*

predecessor of any $\alpha \in \mathbf{N}_k(n)$ that is not equal to the first necklace in co-lex order 0^n. In the statement of the lemma also recall that $\mathbf{N''}_k(n) = \mathbf{N}_k(n) - \{0^n, (k-1)^n\}$.

Lemma 4. *If $\beta = b_1 b_2 \cdots b_n \in \mathbf{N''}_k(n)$ is periodic, then $\mathsf{pred}(\beta)$ and $\mathsf{succ}(\beta)$ are aperiodic. Furthermore, $\mathsf{pred}(\beta)$ shares β's suffix of length $n-i-1$, and $\mathsf{succ}(\beta)$ shares β's prefix of length i, where i is the index such that $b_1 b_2 \cdots b_i = 0^i$ and $b_{i+1} > 0$.*

Proof. Let $\beta = b_1 b_2 \cdots b_n$ be a periodic necklace in $\mathbf{N''}_k(n)$. Let $\mathsf{ap}(\beta) = b_1 b_2 \cdots b_j$ and $u = \frac{n}{j}$ be the number of times $\mathsf{ap}(\beta)$ is repeated in β. Note that $\mathsf{ap}(\beta)$ must include $0^i b_{i+1}$ as a prefix where $b_{i+1} > 0$.

Let $\alpha = \mathsf{pred}(\beta)$. By Theorem 1 and the prefix $0^i b_{i+1}$ in β, it must be that the increasing index of α is $i+1$. Therefore, the common suffix of α and β is $b_{i+2} b_{i+3} \cdots b_n$ which has length $n-(i+1) = n-i-1$ as claimed. Furthermore, α is aperiodic due to its suffix $\mathsf{ap}(\beta)^{u-1}$ and prefix not equal to $\mathsf{ap}(\beta)$.

Let $\gamma = \mathsf{succ}(\beta)$ and t be the increment index of β. Since β contains more than one copy of $0^i b_{i+1}$, $t \geq i+2$. Therefore, by Theorem 1, the necklace γ that immediately follows β in co-lex order must have 0^{i+1} as a prefix. Therefore, β and γ share a prefix of length i. Also observe that by Theorem 1, the string $\gamma \neq 0^n$ has 0^{t-1} as a prefix, while its suffix $b_t b_{t+1} \cdots b_n$ does not contain the substring 0^{t-1} since $t \geq i+2$ and β is a periodic necklace with the prefix $0^i b_{i+1}$. Therefore γ is aperiodic. □

The following simplification of Lemma 4 will be helpful in Section 4.

Corollary 1. *If $\beta \in \mathbf{N''}_k(n)$ is periodic and $\alpha = \mathsf{pred}(\beta)$ is its co-lex predecessor, then $\mathsf{ap}(\alpha) \cdot \mathsf{ap}(\beta)$ has β as a suffix.*

Proof. Suppose $\beta = \mathsf{ap}(\beta)^u$ is a necklace in $\mathbf{N''}_k(n)$ where $u > 1$; β is periodic. By Lemma 4, α is aperiodic and has suffix $\mathsf{ap}(\beta)^{u-1}$. Hence, $\mathsf{ap}(\alpha) \cdot \mathsf{ap}(\beta) = \alpha \cdot \mathsf{ap}(\beta)$ has suffix $\mathsf{ap}(\beta)^{u-1} \cdot \mathsf{ap}(\beta) = \mathsf{ap}(\beta)^u = \beta$. □

4 The Grandmama de Bruijn Sequence

In this section we define the Grandmama de Bruijn sequence, provide a successor rule for generating it, and then prove that it is correct. We note that the de Bruijn sequence successor rule gives the next symbol in the de Bruijn sequence, whereas the co-lex successor rule from Section 3 gives each successive necklace. We conclude the section by discussing efficiency and implementations of the successor rule.

4.1 Definition of the Grandmama de Bruijn Sequence

The *k-ary Grandmama de Bruijn of order n* is constructed by listing the k-ary necklaces of length n in co-lexicographic order and then by concatenating their aperiodic prefixes. We denote this sequence by $\mathcal{M}_k(n)$ and it is defined formally below. An example appears in Figure 1.

Definition 4. $\mathcal{M}_k(n) = \mathsf{ap}(\alpha_1) \cdot \mathsf{ap}(\alpha_2) \cdots \mathsf{ap}(\alpha_{|\mathbf{N}_k(n)|})$*, where $\alpha_1, \alpha_2, \cdots \alpha_{|\mathbf{N}_k(n)|}$ is the co-lex order of $\mathbf{N}_k(n)$.*

The Granddaddy de Bruijn sequence can be constructed by replacing co-lex order by lex order in Definition 4. However, it can also be constructed by concatenating the k-ary Lyndon words whose length divides n in lex order [9, 8]. In fact, the Lyndon word concatenation and the necklace-prefix concatenation of Definition 4 produce identical sequences when lex order is used. For example, notice that the aperiodic prefixes on

the lefthand side of Figure 1 consist of the Lyndon words of length 1, 2, and 4 in lexicographic order. On the other hand, the same is not true when co-lex order is used. For example, when $k = 3$ and $n = 2$ the co-lex Lyndon word concatenation yields $0 \cdot 1 \cdot 01 \cdot 2 \cdot 02 \cdot 12 = 010120212$ while the necklace-prefix concatenation yields $0 \cdot 01 \cdot 1 \cdot 02 \cdot 12 \cdot 2 = 001102122$. Notice that the former is not a de Bruijn sequence while the latter is the Grandmama de Bruijn sequence. The difference between these two definitions was previously discussed in [18], and this subtlety helps explain why the otherwise natural construction of $\mathcal{M}_k(n)$ was not previously observed.

4.2 Successor Rule

We now define a function f which maps every k-ary string of length n to a single symbol of the alphabet. Let $zeroSuffix(\alpha)$ denote the length of the longest suffix of α that has the form 0^*. Thus $zeroSuffix(012000) = 3$. The function f is based on an index ℓ that is derived from $zeroSuffix(\alpha)$ and two test strings τ and τ' described at the bottom of Definition 5. Later in this section we will prove that f is the successor rule for the Grandmama sequence. In other words, f maps each substring of the Grandmama sequence to the next symbol in the sequence.

Definition 5. *The function $f : \Sigma_k^n \to \Sigma_k$ is defined as follows:*

$$f(b_1 b_2 \cdots b_n) = \begin{cases} b_1 + 1 & \text{if } b_1 b_2 \cdots b_n = 0^n \text{ or } \tau' \in \mathbf{N}_k(n) \quad (1a) \\ 0 & \text{if } \tau \in \mathbf{N}_k(n) \text{ and (1a) does not apply} \quad (1b) \\ b_1 & \text{otherwise} \quad (1c) \end{cases}$$

where $\ell = n - zeroSuffix(b_1 b_2 \cdots b_n)$ and $\tau = 0^{n-\ell} b_1 b_2 \cdots b_\ell$ and $\tau' = 0^{n-\ell}(b_1{+}1) b_2 \cdots b_\ell$.

Note that when $b_1 = k{-}1$, $\tau' \notin \Sigma_k^n$ and hence $\tau' \notin \mathbf{N}_k(n)$. To illustrate the definition consider $b_1 b_2 \cdots b_9 = 011200120 \in \Sigma_3^9$. In this case the index is $\ell = 8$, $\tau = 001120012$ and $\tau' = 011120012$. Notice that $\tau' \notin \mathbf{N}_3(9)$ and $\tau \in \mathbf{N}_3(9)$. Hence, $f(011200120) = 0$ by (1b).

Figure 3 further illustrates f applied to each string in Σ_3^3.

4.2.1 Binary Successor Rule

In the binary case the function f can be simplified to the following function g. In this definition $\overline{x}$ denotes the bitwise complement of x.

Definition 6. *The function $g : \Sigma_2^n \to \Sigma_2$ is defined as follows:*

$$g(b_1 b_2 \cdots b_n) = \begin{cases} \overline{b_1} & \text{if } b_1 b_2 \cdots b_n = 0^n \text{ or } \tau \in \mathbf{N}_2(n) \\ b_1 & \text{otherwise} \end{cases}$$

where $\ell = n - zeroSuffix(b_1 b_2 \cdots b_n)$ and $\tau = 0^{n-\ell} 1 b_2 b_3 \cdots b_\ell$.

$\mathcal{M}_3(3)$	0	0	0	1	0	1	1	1	0	2	1	0	0	2	0	1	2	1	1	2	0	2	2	1	2	2	2
$b_1b_2b_3 \in \Sigma_3^3$	000	001	010	101	011	111	110	102	021	210	100	002	020	201	012	121	211	112	120	202	022	221	212	122	222	220	200
ℓ		3	2	3	3	3	2	3	3	2	1	3	2	3	3	3	3	3	2	3	3	3	3	3	3	2	1
τ		001	001	101	011	111	011	102	021	021	001	002	002	201	012	121	211	112	012	202	022	221	212	122	222	022	002
τ'		101	011	201	111	211	021	202	121	031	002	102	012	301	112	221	311	212	022	302	122	321	312	222	322	032	003
equation (1)	(a)	(b)	(a)	(c)	(a)	(b)	(a)	(c)	(b)	(b)	(a)	(b)	(a)	(c)	(a)	(c)	(c)	(b)	(a)	(c)	(a)	(c)	(c)	(a)	(b)	(b)	(b)
$f(b_1b_2b_3)$	1	0	1	1	1	0	2	1	0	0	2	0	1	2	1	1	2	0	2	2	1	2	2	2	0	0	0
rotated $\mathcal{M}_3(3)$	1	0	1	1	1	0	2	1	0	0	2	0	1	2	1	1	2	0	2	2	1	2	2	2	0	0	0

Figure 3: The cyclic order of Σ_3^3 within the Grandmama de Bruijn sequence $\mathcal{M}_3(3)$ as generated by f. Each column gives a string $b_1b_2b_3 \in \Sigma_3^3$ and the corresponding values of ℓ, τ, and τ' from Definition 5. These values determine the case of equation (1) and the result of f. A rotated version of $\mathcal{M}_3(3)$ helps verify f.

4.3 Verification of Successor Rule

Now we prove our main result.

Theorem 2. *The Grandmama sequence $\mathcal{M}_k(n)$ is a de Bruijn sequence for all $k \geq 1$ and $n \geq 0$. Moreover, its substring $b_1b_2\cdots b_n \in \Sigma_k^n$ is followed by the symbol $f(b_1b_2\cdots b_n)$.*

Proof. Since $\mathcal{M}_k(n)$ is composed using the same strings as the Granddaddy de Bruijn sequence, $|\mathcal{M}_k(n)| = k^n$. Thus, to prove this theorem it suffices to show that each string $b_1b_2\cdots b_n \in \Sigma_k^n$ appears as a substring of $\mathcal{M}_k(n)$, and moreover, the bit following $b_1b_2\cdots b_n$ is $f(b_1b_2\cdots b_n)$. Given $b_1b_2\cdots b_n \in \Sigma_k^n$, we consider five cases based on Definition 5. Let $\ell = n - \mathit{zeroSuffix}(b_1b_2\cdots b_n)$ and let

$$\tau = 0^{n-\ell}\, b_1b_2\cdots b_\ell \quad \text{and} \quad \tau' = 0^{n-\ell}\, (b_1{+}1)\, b_2\cdots b_\ell.$$

Case 1. $b_1b_2\cdots b_n = x^n$ for some symbol $x \in \Sigma_k$.

We consider three different cases depending on x.

- $x = 0$. Since $\mathsf{succ}(0^n) = 0^{n-1}1$,

  $$\mathsf{ap}(0^n) \cdot \mathsf{ap}(0^{n-1}1) = 0^n1$$

  is a substring in $\mathcal{M}_k(n)$. The symbol following 0^n is 1, which matches $f(0^n) = b_1{+}1 = 1$ by (1a).
- $x \in \{1, 2, \ldots, k{-}2\}$. Since $\mathsf{succ}((x{-}1)x^{n-1}) = x^n$ and $\mathsf{succ}(x^n) = 0(x{+}1)x^{n-2}$,

  $$\mathsf{ap}((x{-}1)\ x^{n-1}) \cdot \mathsf{ap}(x^n) \cdot \mathsf{ap}(0\ (x{+}1)\ x^{n-2}) = (x{-}1)\ x^n\ 0\ (x{+}1)\ x^{n-2}$$

  is a substring in $\mathcal{M}_k(n)$. The symbol following x^n is 0, which matches $f(x^n) = 0$ by (1b) because $\tau' = (x{+}1)x^{n-1} \notin \mathbf{N}_k(n)$ and $\tau = x^n \in \mathbf{N}_k(n)$.
- $x = k{-}1$. Since $\mathsf{succ}((k{-}2)(k{-}1)^{n-1}) = (k{-}1)^n$ and because the $(k{-}1)^n$ is the last necklace on colex order and 0^n is the first necklace in colex order,

  $$\mathsf{ap}((k{-}2)\ (k{-}1)^{n-1}) \cdot \mathsf{ap}((k{-}1)^n)) \cdot \mathsf{ap}(0^n) = (k{-}2)(k{-}1)^n0$$

  is a substring in $\mathcal{M}_k(n)$ when considering the wrap-around. The symbol following $(k{-}1)^n$ is 0, which matches $f((k{-}1)^n) = 0$ by (1b) because $\tau' = k(k{-}1)^{n-1} \notin \mathbf{N}_k(n)$ and $\tau = (k{-}1)^n \in \mathbf{N}_k(n)$.

Case 2. $\tau' \in \mathbf{N}_k(n)$ and Case 1 does not apply.

Notice that τ' is the first necklace with suffix $(b_1{+}1)b_2 \cdots b_\ell$ in co-lex order. Let β be the last necklace in $\mathbf{N}_k(n)$ in co-lex order with suffix $b_1 b_2 \cdots b_\ell$ and let $\alpha = \mathsf{pred}(\beta)$. This is well-defined because τ has this suffix and $\tau \in \mathbf{N}_k(n)$ due to Lemma 3 and $\tau' \in \mathbf{N}_k(n)$. By Definition 3 and Theorem 1 it must be that $\tau' = \mathsf{succ}(\beta)$. In particular, τ' has prefix $0^{n-\ell}(b_1{+}1)$ and this prefix must also be in $\mathsf{ap}(\tau')$. Therefore, within $\mathcal{M}_k(n)$ the substring $b_1 b_2 \cdots b_n$ appears in

$$\mathsf{ap}(\alpha) \cdot \mathsf{ap}(\beta) \cdot \mathsf{ap}(\tau') = \ldots b_1 b_2 \cdots b_\ell \cdot 0^{n-\ell}(b_1{+}1) \ldots$$

by using Corollary 1 on α and β. The symbol following $b_1 b_2 \cdots b_n$ is $b_1{+}1$, which matches $f(b_1 b_2 \cdots b_n) = b_1{+}1$ by (1a) because $\tau' \in \mathbf{N}_k(n)$.

Case 3. $\tau \in \mathbf{N}_k(n)$ and Cases 1–2 do not apply.

Let β be the last necklace in $\mathbf{N}_k(n)$ in co-lex order with suffix $b_1 b_2 \cdots b_\ell$. Let $\alpha = \mathsf{pred}(\beta)$ and $\gamma = \mathsf{succ}(\beta)$. Since $\tau' \notin \mathbf{N}_k(n)$ Lemmas 1 and 2 ensure that there are no necklaces in $\mathbf{N}_k(n)$ with suffix $y b_2 \cdots b_\ell$ for any $y > b_1$. Therefore, β is also the last necklace with suffix $b_2 b_3 \cdots b_\ell$ in co-lex order. Therefore, β and γ have a common suffix of length at most $n - (\ell - 2) = n - \ell + 2$. Therefore, γ has prefix $0^{n-\ell+1}$ by Definition 3 and Theorem 1. Therefore, within $\mathcal{M}_k(n)$ the substring $b_1 b_2 \cdots b_n$ appears in

$$\mathsf{ap}(\alpha) \cdot \mathsf{ap}(\beta) \cdot \mathsf{ap}(\gamma) = \ldots b_1 b_2 \cdots b_\ell \cdot 0^{n-\ell+1} \ldots$$

by using Corollary 1 on α and β. The symbol following $b_1 b_2 \cdots b_n$ is 0, which matches $f(b_1 b_2 \cdots b_n) = 0$ by (1b) because $\tau' \notin \mathbf{N}_k(n)$ and $\tau \in \mathbf{N}_k(n)$.

Case 4. $b_1 b_2 \cdots b_n$ is periodic and Cases 1–3 do not apply.

Let β be the necklace representative of $b_1 b_2 \cdots b_n$. Since $b_1 b_2 \cdots b_n$ is periodic so is β. Our strategy is to find all of the rotations of β in $\mathcal{M}_k(n)$ and then identify $b_1 b_2 \cdots b_n$.

Let $\beta = a_1 a_2 \cdots a_n$ and i be the index such that $a_1 a_2 \cdots a_i = 0^i$ and $a_{i+1} > 0$; we know this index exists since $b_1 b_2 \cdots b_n = 0^n$ was covered in Case 1. Let j be the length of $\mathsf{ap}(\beta)$ and observe that $i < j$.

Let $\alpha = \mathsf{pred}(\beta)$ and $\gamma = \mathsf{succ}(\beta)$. By Lemma 4, both α and γ are aperiodic. Furthermore, $\mathsf{ap}(\alpha)$ has the same suffix of length $n - i - 1$ as β, and $\mathsf{ap}(\gamma)$ has the same prefix of length i as β. Therefore, $\mathcal{M}_k(n)$ contains $\mathsf{ap}(\alpha) \cdot \mathsf{ap}(\beta) \cdot \mathsf{ap}(\gamma)$ which itself must contain

$$a_{i+2} a_{i+3} \cdots a_n \;\cdot\; a_1 a_2 \cdots a_j \;\cdot\; a_1 a_2 \cdots a_i.$$

The identified subsequence has length $(n - i - 1) + j + i = n + j - 1$ and thus j (non-circular) substrings of length n. Furthermore, these substrings are precisely the j distinct rotations of β. Hence, $b_1 b_2 \cdots b_n$ is one of these substrings.

To complete the proof we must verify the successor rule. The last of the j substrings of length n given above is $a_{i+1} a_{i+2} \cdots a_n a_1 a_2 \cdots a_i$. Also recall that $a_1 a_2 \cdots a_i = 0^i$. Therefore, $b_1 b_2 \cdots b_n$ is not this substring since otherwise $\tau = a_1 a_2 \cdots a_n = \beta \in \mathbf{N}_k(n)$ and this was covered by previous cases. Since $b_1 b_2 \cdots b_n$ is not the last of the j substrings it must be followed by b_1. This matches $f(b_1 b_2 \cdots b_n) = b_1$ by (1c) because $\tau \notin \mathbf{N}_k(n)$ and $\tau' \notin \mathbf{N}_k(n)$ due to the previous cases.

Case 5. $b_1 b_2 \cdots b_n$ is aperiodic and Cases 1–4 do not apply.

Let the necklace representative of $b_1 b_2 \cdots b_n$ be $\gamma \in \mathbf{N}_k(n)$. Recall that $b_1 b_2 \cdots b_n = b_1 b_2 \cdots b_\ell 0^{n-\ell}$ where $b_\ell > 0$. Therefore, $\gamma = b_m b_{m+1} \cdots b_\ell 0^{n-\ell} b_1 b_2 \cdots b_{m-1}$ for some choice of m. In particular, $m \geq 2$

due to the fact that 0^n is the only necklace whose last symbol is 0 and $\beta \neq 0^n$. Thus, γ must have prefix $b_m b_{m+1} \cdots b_\ell 0^{n-\ell} b_1$.

Let $\beta = \mathsf{pred}(\gamma)$ and $\alpha = \mathsf{pred}(\beta)$. Note that α is well-defined since γ is not the first or second necklaces in co-lex order, namely 0^n or $0^{n-1}1$, due to previous cases. In particular, $\gamma \neq 0^{n-1}1$ since otherwise $b_1 b_2 \cdots b_n$ would be a rotation of $0^{n-1}1$ and hence $\tau = 0^{n-1}1 \in \mathbf{N}_k(n)$.

By Theorem 1 and $b_\ell > 0$ it must be that β has suffix $0^{n-\ell} b_1 b_2 \cdots b_{m-1}$. More specifically, β has suffix $b_1 b_2 \cdots b_{m-1}$. By Corollary 1, $\mathsf{ap}(\alpha) \cdot \mathsf{ap}(\beta)$ has suffix β and more specifically $b_1 b_2 \cdots b_{m-1}$. Also, γ is aperiodic due to the fact that $b_1 b_2 \cdots b_n$ is aperiodic. Therefore, $\mathcal{M}_k(n)$ contains $\mathsf{ap}(\alpha) \cdot \mathsf{ap}(\beta) \cdot \mathsf{ap}(\gamma)$ which itself contains

$$b_1 b_2 \cdots b_{m-1} \ \cdot \ b_m b_{m+1} \cdots b_\ell \ 0^{n-\ell} \ b_1.$$

Hence, $b_1 b_2 \cdots b_n$ appears is a substring in $\mathcal{M}_k(n)$. Furthermore, the symbol following $b_1 b_2 \cdots b_n$ is b_1. This matches $f(b_1 b_2 \cdots b_n) = b_1$ by (1c) because $\tau \notin \mathbf{N}_k(n)$ and $\tau' \in \mathbf{N}_k(n)$ due to the previous cases. □

4.4 Efficiency and Implementation

When analyzing the efficiency of the k-ary successor function f, the main challenge is testing if τ is a necklace in (1a) and testing if τ' is a necklace in (1b). These tests can be implemented in $O(n)$-time [2, 5].

Theorem 3. *The function f can be computed in $O(n)$-time.*

This theorem immediately implies that the Grandmama de Bruijn sequence can be generated in $O(n)$-time per bit starting from any arbitrary string in Σ_k^n. This is a nice advantage over the necklace concatenation approach which could be adapted to start from an arbitrary necklace. In the binary case, there is an efficient algorithm to list binary necklaces in co-lex order which allows $\mathcal{M}_2(n)$ to be generated in $O(1)$-time per bit [21]. An implementation in C that generates $\mathcal{M}_k(n)$ using the successor rule is provided in the Appendix. It is also available at the following git repository: `https://src-code.simons-rock.edu/git/awilliams/GrandmamaDeBruijn`.

5 Acknowledgement

The research of Joe Sawada is supported by the *Natural Sciences and Engineering Research Council of Canada* (NSERC) grant RGPIN 400673-2012.

References

[1] Y. H. Au. Generalized de Bruijn words for primitive words and powers. *Discrete Mathematics*, 338(12):2320–2331, 2015.

[2] K. S. Booth. Lexicographically least circular substrings. *Inform. Process. Lett.*, 10(4/5):240–242, 1980.

[3] J. Cooper and C. Heitsch. The discrepancy of the lex-least de Bruijn sequence. *Discrete Mathematics*, 310(6-7):1152–1159, 2010.

[4] P. Dragon, O. Hernandez, and A. Williams. The grandmama de Bruijn sequence. In E. C. E. Kranakis, G. Navarro, editor, *LATIN 2016: Theoretical Informatics*, pages 347–361. Springer, 2016.

[5] J. P. Duval. Factorizing words over an ordered alphabet. *Journal of Algorithms*, 4(4):363–381, 1983.

[6] L. R. Ford. A cyclic arrangement of m-tuples. *Report No. P-1071, RAND Corp.*, 1957.

[7] H. Fredricksen. Generation of the Ford sequence of length 2^n, n large. *Journal of Combinatorial Theory (A)*, 12:153–154, 1972.

[8] H. Fredricksen and I. J. Kessler. An algorithm for generating necklaces of beads in two colors. *Discrete Mathematics*, 61:181–188, 1986.

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

[10] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Professional, 2nd edition, 1994.

[11] Y. Huang. A new algorithm for the generation of binary de Bruijn sequences. *Journal of Algorithms*, 11:44–51, 1990.

[12] D. E. Knuth. *The Art of Computer Programming, Volume 4A, Combinatorial Algorithms*. Addison-Wesley Professional, 2011.

[13] M. H. Martin. A problem in arrangements. *Bulletin of the American Mathematical Society*, 40:859–864, 1934.

[14] M. Matamala and E. Moreno. Minimum Eulerian circuits and minimum de Bruijn sequences. *Discrete Mathematics*, 309(17):5298–5304, 2009.

[15] E. Moreno. On the theorem of Fredricksen and Maiorana about de Bruijn sequences. *Advances in Applied Mathematics*, 33(2):413–415, 2004.

[16] E. Moreno and D. Perrin. Corrigendum to "On the theorem of Fredricksen and Maiorana about de Bruijn sequences". *Advances in Applied Mathematics*, 62(1):184–187, 2015.

[17] F. Ruskey, C. Savage, and T. Wang. Generating necklaces. *Journal of Algorithms*, 13:414–413, 1992.

[18] F. Ruskey, J. Sawada, and A. Williams. De Bruijn sequences for fixed-weight binary strings. *SIAM Journal on Discrete Mathematics*, 26(2):605–617, 2012.

[19] J. Sawada, A. Williams, and D. Wong. The lexicographically smallest universal cycle for binary strings with minimum specified weight. *Journal of Discrete Algorithms*, 28(0):31 – 40, 2014. StringMasters 2012 & 2013 Special Issue.

[20] J. Sawada, A. Williams, and D. Wong. Generalizing the classic greedy and necklace constructions of de Bruijn sequences and universal cycles. *Electronic Journal of Combinatorics*, 23(1):#P1.24, 2016.

[21] J. Sawada, A. Williams, and D. Wong. Necklaces and Lyndon words in colexicographic and reflected Gray code order. *(submitted)*, 2017.

[22] J. Sawada, A. Williams, and D. Wong. A simple shift rule for k-ary de Bruijn sequences. *Discrete Mathematics*, 340(3):524–531, 2017.

[23] S. K. Stein. *Mathematics: The Man-Made Universe*. W. H. Freeman and Company, 3rd edition, 1994.

Appendix - C code to construct the Grandmama de Bruijn sequence

```
#include<stdio.h>
#define MAX_N 100

int Zeros(int a[], int n) {
   for (int i=1; i<=n; i++) if (a[i] != 0) return 0;
   return 1;
}
//-----------------------------------------
int IsNecklace(int b[], int n) {
   int i, p=1;

   for (i=2; i<=n; i++) {
      if (b[i-p] > b[i]) return 0;
      if (b[i-p] < b[i]) p = i;
   }
   if (n % p != 0) return 0;
   return p;
}
//-----------------------------------------
int Grandmama_successor(int a[], int n, int k) {
   int i,j=1,t,b[MAX_N];

   while (j<n && a[n-j+1] == 0) b[j++] = 0;
   t = 1;
   for (i=j; i<=n; i++) b[i] = a[t++];

   b[j]++;
   if (Zeros(a,n) || (b[j] < k && IsNecklace(b,n))) return a[1]+1;
   b[j]--;
   if (IsNecklace(b,n)) return 0;
   return a[1];
}
//-----------------------------------------
int main() {
   int i,n,k,new_bit,a[MAX_N];

   printf("Enter n: "); scanf("%d", &n);
   printf("Enter k: "); scanf("%d", &k);

   for (i=1; i<=n; i++) a[i] = 0;
   do {
      printf("%d", a[1]);
      new_bit = Grandmama_successor(a,n,k);
      for (i=1; i<=n; i++) a[i] = a[i+1];
      a[n] = new_bit;

   } while (!Zeros(a,n));
   printf("\n\n");
}
```