Generalized de Bruijn words for Primitive words and Powers

Yu Hin Au
Department of Mathematics
Milwaukee School of Engineering
au@msoe.edu

May 24, 2015

Abstract

We show that for every $n \geq 1$ and over any finite alphabet, there is a word whose circular factors of length n have a one-to-one correspondence with the set of primitive words. In particular, we prove that such a word can be obtained by a greedy algorithm, or by concatenating all Lyndon words of length n in increasing lexicographic order. We also look into connections between de Bruijn graphs of primitive words and Lyndon graphs.

Finally, we also show that the shortest word that contains every p-power of length pn over a k-letter alphabet has length between pk^n and roughly $(p + \frac{1}{k})k^n$, for all integers $p \geq 1$. An algorithm that generates a word which achieves the upper bound is provided.

1 Introduction

In this paper, we study generalizations of de Bruijn words, and provide a few results related to some well-studied collection of words. We first establish some notation. Given an integer $k \geq 2$, we define $\Sigma_k := \{0, 1, \ldots, k-1\}$, and let $|w|$ denote the length of any finite word $w \in \Sigma_k^*$. Also, we define $w[i]$ to be the i^{th} symbol in w, and $w[i\,..\,j]$ to be the word $w[i]w[i+1]\cdots w[j-1]w[j]$, for any indices i, j such that $1 \leq i \leq j \leq |w|$. If $i > j$, then we define $w[i\,..\,j]$ to be the empty word. Also, given any word $x \in \Sigma_k^n$ and an integer $p \geq 1$, we define x^p to be the word obtained from concatenating p copies of x. For example, $(01)^3 = 010101$. A word w is *p-power* if $w = x^p$ for some word x and some integer p. Conventionally, 2-powers are usually called squares, and 3-powers are called cubes.

We say that a word x is a *factor* (also sometimes called a subword) of another word w if $x = w[i\,..\,j]$ for some indices i, j, and we say that x is a *circular factor* of w if x is a factor of w^p for some integer p. Given integers n and k, a sequence in which every word in Σ_k^n appears as a circular factor exactly once is called a *de Bruijn word*, named after Nicolaas Govert de

Bruijn for his work on these sequences in [dB46]. For example, 00011101 is a de Bruijn word for $\{0,1\}^3$. It has long been known that such a sequence exists for Σ_k^n, for every $n, k \geq 1$. In fact, there are exponentially many such sequences [Mar94].

There are many ways to generate a de Bruijn word for Σ_k^n. First, one can be obtained by a greedy algorithm:

Algorithm A. *Generating a de Bruijn word w for Σ_k^n*

Input: *Integers $n, k \geq 1$*
Set $w[1\,..\,n] = 0^n$
Set $i = n + 1$
while *$\exists \alpha \in \Sigma_k$ such that $w[i-n+1\,..\,i-1]\alpha$ is not a factor of $w[1\,..\,i-1]$* **do**
    Set $w[i]$ to be the largest such symbol α
    Increment i
end
Discard last $n-1$ symbols in w
return w

In other words, we start with 0^n, and then successively append the largest symbol in the alphabet that does not create a factor of length n that had appeared earlier in our sequence, and stop if there is no such symbol. Then the resulting word, with the last $n-1$ symbols removed, is a de Bruijn word for Σ_k^n. This simple algorithm was discovered independently by several mathematicians [Fre82], first by [Mar34].

Alternatively, one can also construct a de Bruijn word for Σ_k^n by doing the following. Given a word $w \in \Sigma_k^n$, define

$$w^{(i)} := w[i+1\,..\,n]x[1\,..\,i]$$

for all $i = 1, \ldots, n$. We say that $w^{(1)}, \ldots, w^{(n)}$ are the *conjugates* of w, and define a word $w \in \Sigma_k^n$ to be *primitive* if $w \neq w^{(i)}$ for all $i \in \{1, 2, \ldots, n-1\}$. Next, a word $w \in \Sigma_k^n$ is *Lyndon* if w is primitive, and is the lexicographically smallest among its conjugates. The following result, due to Fredricksen and Maiorana [FM78], establishes a remarkable connection between de Bruijn words and Lyndon words.

Theorem 1. *Let w be the concatenation of all Lyndon words in Σ_k^* of length dividing n, in increasing lexicographic order. Then w is a de Bruijn word for Σ_k^n.*

For instance, the six binary Lyndon words with length dividing four are, in increasing lexicographic order, $0, 0001, 0011, 01, 0111$ and 1. Thus, by Theorem 1,

$$w := 0000100110101111$$

is a de Bruijn word for $\{0,1\}^4$. An advantage of this approach is that, unlike the greedy algorithm that requires exponential storage space during its execution, generating a de Bruijn word by concatenating Lyndon words can be done in constant time and space per bit [RSW92].

More recently, Moreno [Mor05] extended the notion of de Bruijn words to an arbitrary dictionary $\mathcal{D} \subseteq \Sigma_k^n$, and defined a de Bruijn word for $\mathcal{D}$ to be a sequence in which every word

in $\mathcal{D}$ (and no other words in Σ_k^n) appears as a circular factor exactly once. For instance, if we let $\mathcal{D}$ be the set of words in $\{0,1\}^4$ with at least two 1s, then the word 11101011001 is a de Bruijn word for $\mathcal{D}$. Yet further generalizations of de Bruijn words, such as *universal cycles*, have also been studied in the literature (see, for instance, [CDG92] and [Joh09]).

This paper will be organized as follows: In the next section, we first work with Moreno's generalization, and show that de Bruijn words of the set of primitive words in Σ_k^n exist, for all integers $n, k \geq 2$. Among other results, we prove that a de Bruijn word for the set of primitive words in Σ_k^n can be generated by either of the following procedures:

- Start with $w = 0^{n-1}$, and iteratively append the largest symbol in Σ_k that does not create a factor of length n that is not primitive or has already appeared in w. Stop when the word cannot be further extended, and discard the last $n-1$ symbols of w.
- Concatenate all Lyndon words of length n, in increasing lexicographic order.

Some of the tools we use, such as presenting greedy algorithms under the framework for preference functions and making connections between de Bruijn and Lyndon graphs of dictionaries, could help with the analysis and construction of de Bruijn words of other dictionaries. In Section 3, we look into a different generalization of de Bruijn words, and show that the shortest sequence that contains all p-powers of length pn as factors has length between pk^n and roughly $(p + \frac{1}{k})k^n$, for all integers $p \geq 1$. We provide an algorithmic proof for the upper bound, and discuss some computational results.

2 de Bruijn Words for Primitive Words

First of all, it is apparent de Bruijn words do not exist for some dictionaries $\mathcal{D} \subseteq \Sigma_k^n$. For instance, consider the dictionary $\mathcal{D} := \{0000, 0001, 0011, 0111\}$. There is clearly no binary word of length 4 that contains all four words in $\mathcal{D}$ as circular factors. Moreno [Mor05] observed that the dictionaries for which de Bruijn words exist can be characterized by looking at their corresponding de Bruijn graphs. Given $\mathcal{D} \subseteq \Sigma_k^n$, its *de Bruijn graph* $G^{\mathcal{D}}$ is defined as follows:

- Its vertices $V(G^{\mathcal{D}})$ is the set of words in Σ_k^{n-1} that are factors of some word in $\mathcal{D}$;
- Its arcs $E(G^{\mathcal{D}})$ is the set of ordered pairs $\{u, v\}$ where $u, v \in \Sigma_k^{n-1}$ and there exists a word in $\mathcal{D}$ whose prefix is u and suffix is v.

For example, Figure 1 illustrates $G^{\mathcal{D}}$ where $\mathcal{D}$ is the set of words in $\{0,1\}^4$ with at least two 1s. Each arc $\{u, v\}$ (which will sometimes be abbreviated as uv from here on to reduce cluttering) is labelled by the unique word in $\mathcal{D}$ of which u is a prefix and v is a suffix. Alternatively, $G^{\mathcal{D}}$ can be defined as the de Bruijn graph of Σ_k^n, with arcs corresponding to words in $\Sigma_k^n \setminus \mathcal{D}$ removed, and then isolated vertices deleted.

Given a directed graph G, an *Eulerian cycle* in G is a closed walk that uses every arc in G exactly once. An important property of de Bruijn graphs is that, for any dictionary $\mathcal{D} \subseteq \Sigma_k^n$, there is a one-to-one correspondence between de Bruijn words of $\mathcal{D}$ and the Eulerian cycles of $G^{\mathcal{D}}$ [Mor05]. For instance, an Eulerian cycle in the graph in Figure 1 can be obtained

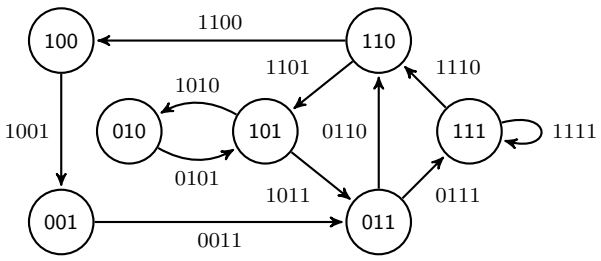

Figure 1: The de Bruijn graph for the set of words in $\{0,1\}^4$ with at least two 1s.

from starting at the vertex 001, and going through arcs 0011, 0111, 1111,1110, 1101, 1010, 0101, 1011, 0110, 1100, and 1001 in that order. Then by concatenating the last symbol in each of these arcs, we obtain 11101011001, the aforementioned de Bruijn word for this dictionary. Likewise, given any de Bruijn word, one can construct from its circular factors a corresponding Eulerian cycle in the de Bruijn graph.

Next, we show that there is a de Bruijn word for the set of primitive words in Σ_k^n, for every $n, k \geq 2$. In fact, we will provide three rather different proofs, as they each make use of different tools and connects with different existing results.

2.1 Using Greedy Algorithms

Before we focus on the set of primitive words, we look into a general framework that will allow us to analyze the viability of generating de Bruijn words using greedy algorithms for arbitrary dictionaries. Given a dictionary $\mathcal{D} \subseteq \Sigma_k^n$, Moreno [Mor05] showed that a necessary condition for $\mathcal{D}$ to have a de Bruijn word is the following:

$$|\{\alpha \in \Sigma_k : \alpha u \in \mathcal{D}\}| = |\{\alpha \in \Sigma_k : u\alpha \in \mathcal{D}\}|, \quad \forall u \in \Sigma_k^{n-1}. \tag{1}$$

That is, for any word u of length $n-1$, the number of symbols that can left-extend u to a word in $\mathcal{D}$ is equal to the number of symbols that can right-extend u to a word in $\mathcal{D}$. This is equivalent to the condition that the in-degree is equal to the out-degree for every vertex in the graph $G^{\mathcal{D}}$.

Next, given a dictionary $\mathcal{D}$, we say that a word $u \in \Sigma_k^*$ is *$\mathcal{D}$-nonrepeating* if it satisfies all of the following conditions:

1. $|u| \geq n-1$, and $u[1\,..\,n-1]$ is a factor of some word in $\mathcal{D}$;
2. u does not contain any word in $\Sigma_k^n \setminus \mathcal{D}$ as a factor;
3. u does not contain any word in $\mathcal{D}$ as a factor more than once.

Note that if $x \in \mathcal{D}$, then x and $x[1\,..\,n-1]$ are both $\mathcal{D}$-nonrepeating. Also, using the same correspondence between de Bruijn words of $\mathcal{D}$ and Eulerian cycles in $G^{\mathcal{D}}$ described previously, a $\mathcal{D}$-nonrepeating word translates to a walk in $G^{\mathcal{D}}$ in which no arc is used more than once. As we will see subsequently, these $\mathcal{D}$-nonrepeating words will serve as eligible starting points of constructing de Bruijn words for $\mathcal{D}$.

Next, let $\mathcal{P}$ be a *preference function* that maps each word in Σ_k^{n-1} to an ordered set that contains each symbol in Σ_k exactly once. We then define $f_{\max}(u)$ to be the word generated by the following algorithm

Algorithm B. *Generating* $f_{\max}(u)$
Input: *Dictionary* $\mathcal{D} \subseteq \Sigma_k^n$, *preference function* $\mathcal{P}$, $\mathcal{D}$*-nonrepeating word* u
Set $f_{\max}(u)[1\,..\,|u|] = u$
Set $i = |u| + 1$
while $\exists \alpha \in \Sigma_k$ *such that* $f_{\max}(u)[i-n+1\,..\,i-1]\alpha \in \mathcal{D}$ *and is not a factor of* $f_{\max}(u)[1\,..\,i-1]$ **do**
  Set $f_{\max}(u)[i]$ *to be the first such symbol in the set* $\mathcal{P}(f_{\max}(u)[i-n+1\,..\,i-1])$
  Increment i
end
return $f_{\max}(u)$

For example, let $\mathcal{D} = \{0,1\}^4$, $u = 0000$, and $\mathcal{P}$ be the preference function where

$$\mathcal{P}(w) = \{1, 0\}, \quad \forall w \in \{0,1\}^3.$$

In other words, when choosing a symbol to append to $f_{\max}(u)$, we always try the symbol 1 before 0. In this case, $f_{\max}(u) = 0000111101100101000$, and removing the last 3 symbols result in a de Bruijn word for $\mathcal{D}$. More generally, when $\mathcal{D} = \Sigma_k^n$, $u = 0^n$ and

$$\mathcal{P}(w) = \{k-1, k-2, \ldots, 1, 0\}, \quad \forall w \in \Sigma_k^{n-1},$$

the construction of $f_{\max}(u)$ (with the last $n-1$ symbols removed) coincides with Algorithm A, the aforementioned greedy algorithm that generates a de Bruijn word for Σ_k^n. Here, the preference function $\mathcal{P}$ can be interpreted as always attempting to pick the largest eligible symbol to extend $f_{\max}(u)$. While the framework with preference functions may seem a little clumsy at this point, it allows the possibility of having the preference of symbols vary upon the current suffix of $f_{\max}(u)$, which we shall explore later in this section.

We now characterize situations where, given dictionary $\mathcal{D}$, $\mathcal{D}$-nonrepeating word u, and preference function $\mathcal{P}$, $f_{\max}(u)$ is in fact a de Bruijn word for $\mathcal{D}$ (after having its last $n-1$ symbols removed). Consider the following closely related sequence:

Algorithm C. *Generating* $f_{\min}(u)$
Input: *Dictionary* $\mathcal{D} \subseteq \Sigma_k^n$, *preference function* $\mathcal{P}$, $\mathcal{D}$*-nonrepeating word* u
Set $f_{\min}(u)[1\,..\,|u|] = u$
Set $i = |u| + 1$
while $\exists \alpha \in \Sigma_k$ *such that* $f_{\min}(u)[i-n+1\,..\,i-1]\alpha \in \mathcal{D}$ *and is not a factor of* $f_{\min}(u)[1\,..\,i-1]$ **do**
  Set $f_{\min}(u)[i]$ *to be the last such symbol in the set* $\mathcal{P}(f_{\min}(u)[i-n+1\,..\,i-1])$
  Increment i
end
return $f_{\min}(u)$

That is, $f_{\min}(u)$ is constructed in a similar fashion as $f_{\max}(u)$, except that we iteratively append the least preferred symbol among all eligible ones, instead of the most preferred. Somewhat surprisingly, the words obtained from being greedy and "anti-greedy" can be related as follows.

Theorem 2. *Suppose we are given a dictionary $\mathcal{D} \subseteq \Sigma_k^n$ that satisfies (1), $u \in \Sigma_k^*$ that is $\mathcal{D}$-nonrepeating, and preference function $\mathcal{P}$. If $u[1\,..\,n-1]$ is a factor of $f_{\min}(w)$ for all $w \in \Sigma_k^{n-1}$ that is a factor of some word in $\mathcal{D}$, then $f_{\max}(u)$ contains every word in $\mathcal{D}$ as factor exactly once. Moreover, the word obtained from $f_{\max}(u)$ by discarding the last $n-1$ symbols is a de Bruijn word for $\mathcal{D}$.*

Proof. By construction (and the fact that u is $\mathcal{D}$-nonrepeating), every factor of $f_{\max}(u)$ of length n is in $\mathcal{D}$, and no such factors appear twice. Therefore, it suffices to show that every word in $\mathcal{D}$ does appear as a factor in $f_{\max}(u)$.

First, observe that $f_{\max}(u)$ must end with $u[1\,..\,n-1]$. Otherwise, let x be the suffix of $f_{\max}(u)$ of length $n-1$, and suppose x appears q times in $f_{\max}(u)$ as a factor. The construction of $f_{\max}(u)$ terminates at x implies that $|\,\{\beta : x\beta \in \mathcal{D}\}\,| = q-1$. However, since $f_{\max}(u)$ starts with $u[1\,..\,n-1]$ which by assumption is not equal to x, we have $|\,\{\beta : \beta x \in \mathcal{D}\}\,| \geq q$, contradicting the assumption that $\mathcal{D}$ satisfies (1).

Next, suppose for a contradiction that there exists $\alpha_1 \in \Sigma_k, y \in \Sigma_k^{n-1}$ such that $\alpha_1 y \in \mathcal{D}$ but is not a factor of $f_{\max}(u)$. Since $|\,\{\beta : \beta y \in \mathcal{D}\}\,| = |\,\{\beta : y\beta \in \mathcal{D}\}\,|$ and

$$|\,\{\beta : \beta y \text{ is a factor of } f_{\max}(u)\}\,| = |\,\{\beta : y\beta \text{ is a factor of } f_{\max}(u)\}\,|,$$

there exists $\alpha_2 \in \Sigma_k$ such that $y\alpha_2 \in \mathcal{D}$ but is not a factor of $f_{\max}(u)$. In particular, since the algorithm always chooses the most preferred symbol to extend $f_{\max}(u)$, we may assume that α_2 is the last symbol in the ordered set $\mathcal{P}(y)$ where $y\alpha_2$ is in $\mathcal{D}$.

Applying the same reasoning on $y[2\,..\,n-1]\alpha_2$, we conclude that if we let α_3 be the least preferred symbol in $\mathcal{P}(y[2\,..\,n-1]\alpha_2)$ such that $y[2\,..\,n-1]\alpha_2\alpha_3$ is in $\mathcal{D}$, then $y[2\,..\,n-1]\alpha_2\alpha_3$ does not appear in $f_{\max}(u)$.

Keep proceeding in this manner, and we conclude that any factor of length n in $f_{\min}(\alpha_1 y)$ does not appear in $f_{\max}(u)$. By the same argument we used above to show that $f_{\max}(u)$ must have $u[1\,..\,n-1]$ as its prefix and suffix, we may conclude that $f_{\min}(\alpha_1 y)$ has both $\alpha_1 y[1\,..\,n-1]$ as prefix and suffix. Since $f_{\min}(\alpha_1 y)$ contains $u[1\,..\,n-1]$ as a factor by assumption, this implies that there exists symbol β where $u[1\,..\,n-1]\beta$ is both in $\mathcal{D}$ and a factor of $f_{\min}(\alpha_1 y)$, and thus $u[1\,..\,n-1]\beta$ does not appear in $f_{\max}(u)$. However, since we have shown above that $f_{\max}(u)$ must end with $u[1\,..\,n-1]$, it then must contain all words in $\mathcal{D}$ with prefix $u[1\,..\,n-1]$, and thus we obtain a contradiction. Therefore, $f_{\max}(u)$ must contain every word in $\mathcal{D}$ as a factor exactly once. Finally, since $f_{\max}(u)$ both starts and ends with $u[1\,..\,n-1]$, a de Bruijn word for $\mathcal{D}$ can be obtained by discarding the last $n-1$ symbols of $f_{\max}(u)$. □

We remark that the converse of Theorem 2 is not true. For an example, let $\mathcal{D} = \{0,1\}^4$ and $\mathcal{P}(w) = \{1,0\}$ for all $w \in \{0,1\}^3$, then

$$f_{\max}(0011) = 0011110110010100001,$$

and removing the last 3 symbols result in a de Bruijn word for $\{0,1\}^4$. However, we see that

$$f_{\min}(000) = 00001000,$$

which does not contain 0011. Hence, while $f_{\min}(w)$ contains u for every $w \in \Sigma_k^{n-1}$ is a sufficient condition for $f_{\max}(u)$ to contain a de Bruijn word for $\mathcal{D}$, it is not necessary.

Next, we apply Theorem 2 to show that the simple greedy algorithm that generates a de Bruijn word for Σ_k^n can be adapted to generate a de Bruijn word for the set of primitive words. We first need the following.

Lemma 3. *Let $\mathcal{D}$ be the set of primitive words in Σ_k^n. Then $\mathcal{D}$ satisfies* (1).

Proof. For any $u \in \Sigma_k^{n-1}, \alpha \in \Sigma_k$, if αu is not primitive, then it can be written as $(\alpha x)^p$ for some word x and integer $p \geq 2$. But then $u\alpha = (x\alpha)^p$ is not primitive either. Thus, we see that for every $u \in \Sigma_k^{n-1}$, αu is primitive if and only if $u\alpha$ is primitive.

Therefore, the sets on either side of the equality in (1) are identical for every $u \in \Sigma_k^{n-1}$, so it is apparent that they have the same size. □

We will also need the following property of primitive words:

Lemma 4. *For every $u \in \Sigma_k^{n-1}$ and distinct symbols $\alpha, \beta \in \Sigma_k$, if $u\alpha$ is not primitive, then every factor of $u\beta^{n-1}$ of length n is primitive.*

Proof. To obtain a contradiction, suppose that $u\alpha$ is not primitive, and that there exists integer $\ell \leq n-1$ such that $u[\ell\,..\,n-1]\beta^\ell$ is also not primitive. Then there exist words x, y and integers $p, q \geq 2$ such that $u\alpha = x^p$ and $\beta^{\ell-1}u[\ell\,..\,n-1]\beta = y^q$ (the latter is due to $\beta^{\ell-1}u[\ell\,..\,n-1]\beta$ being a conjugate of $u[\ell\,..\,n-1]\beta^\ell$). Notice that $|y| > \ell$, or otherwise $\beta^{\ell-1}u[\ell\,..\,n-1]\beta = y^q$ implies $y = \beta^{|y|}$, and consequently $u = \beta^{n-1}$, which would imply that $u\alpha$ is primitive. Thus, we obtain that

$$u[s|x|] = \alpha, \quad \forall s \in \{1, \ldots, p-1\}, \tag{2}$$

$$u[t|y| + r] = \beta, \quad \forall t \in \{1, \ldots, q-1\}, r \in \{0, \ldots, \ell-1\}. \tag{3}$$

Define m to be the least common multiple of $|x|$ and $|y|$. If $m < n$, then $u[m] = \alpha$ by (2) and $u[m] = \beta$ by (3), a contradiction. Thus, $|x|$ and $|y|$ are coprime, and so for any fixed $r \in \{1, \ldots, |y|-1\}$, there exists $s \in \{1, \ldots, |y|-1\}$ such that $s|x| \equiv r \pmod{|y|}$. Since $u[s|x|] = \alpha$ for all $s \in \{1, \ldots, |y|-1\}$, this implies that $y = \alpha^{|y|-1}\beta$. But then $u\alpha = \left(\alpha^{|y|-1}\beta\right)^{q-1}\alpha^{|y|}$ would be primitive, which is a contradiction. □

We are finally ready to prove the following:

Theorem 5. *Let $\mathcal{D}$ be the set of primitive words in Σ_k^n where $n, k \geq 2$, and let $\mathcal{P}$ be the preference function where*

$$\mathcal{P}(w) = \{k-1, k-2, \ldots, 1, 0\}, \quad \forall w \in \Sigma_k^n.$$

Then $f_{\max}(0^{n-1})$ (minus the last $n-1$ symbols) is a de Bruijn word for $\mathcal{D}$.

Proof. First, 0^{n-1} is obviously $\mathcal{D}$-nonrepeating. Also, we have shown that the set of primitive words satisfies (1). Thus, by Theorem 2, it suffices to show that $f_{\min}(w)$ contains 0^{n-1} for all $w \in \Sigma_k^{n-1}$. By Lemma 4, we see that $f_{\min}(w)$ either has prefix $w0^\ell$ that contains a factor of 0^{n-1}, or $w0^\ell 1^{n-1}0^{n-1}$ for some $\ell \geq 0$. In either case, $f_{\min}(w)$ contains 0^{n-1}, and our claim follows. □

Thus, we have shown that starting with 0^{n-1} and iteratively appending the largest possible symbol that does not create a factor of length n that has already appeared or is not primitive will result in a de Bruijn word for the set of primitive words. it is not hard to see that the ingredients in the above arguments can be extended to show the following slightly stronger result:

Theorem 6. *let $\mathcal{D}$ be the set of primitive words in Σ_k^n, where $n, k \geq 2$. Let $\mathcal{P}$ be the preference function such that*

$$\mathcal{P}(w) = \{\alpha_1, \alpha_2, \ldots, \alpha_{k-1}\}, \quad \forall w \in \Sigma_k^n,$$

where $\{\alpha_1, \ldots, \alpha_{k-1}\}$ is any fixed ordering of the alphabet Σ_k. Then $f_{\max}((\alpha_{k-1})^{n-1})$ (minus the last $n-1$ symbols) is a de Bruijn word for $\mathcal{D}$.

In particular, this implies that the "prefer minimum" algorithm (start with $n-1$ copies of the largest symbol, iteratively extend sequence by writing down the smallest symbol that does not create a repeat or non-primitive factor of length n) also generates a de Bruijn word.

We next look into a case where the preference function $\mathcal{P}$ varies upon $w \in \Sigma_k^{n-1}$. First, Alhakim [Alh10] showed the following interesting result for binary sequences, which we paraphrase here using preference functions:

Theorem 7. *Let $\mathcal{D} = \{0,1\}^n$, and $\mathcal{P}$ be the preference function such that*

$$\mathcal{P}(w) = \begin{cases} \{1,0\} & \text{if } w \in \{0,1\}^{n-1} \text{ ends with a 0;} \\ \{0,1\} & \text{if } w \in \{0,1\}^{n-1} \text{ ends with a 1.} \end{cases}$$

Then $f_{\max}(0^n)$, with the last $n-1$ symbols removed and then the symbol 1 appended, is a de Bruijn word for $\{0,1\}^n$.

Alhakim named the construction of this sequence the "prefer opposite algorithm" — at each iteration, it prefers to extend the sequence by adding the symbol that is different from the current last symbol in the sequence. For an example, when $n = 4$, we obtain

$$f_{\max}(0000) = 0000101001101110 00.$$

Then we remove the last three 0's and add a 1, and obtain 0000101001101111, which is a de Bruijn word for $\{0,1\}^4$.

We now apply Theorem 2 again to show that a de Bruijn word for the set of primitive words can be obtained in this "prefer opposite" manner as well.

Theorem 8. *Let $\mathcal{D}$ be the set of primitive words in $\{0,1\}^n$, and define the preference function $\mathcal{P}$ such that*

$$\mathcal{P}(w) = \begin{cases} \{1,0\} & \text{if } w \in \{0,1\}^{n-1} \text{ ends with a 0;} \\ \{0,1\} & \text{if } w \in \{0,1\}^{n-1} \text{ ends with a 1.} \end{cases}$$

Then $f_{\max}(0^{n-1})$, with the last $n-1$ symbols removed, is a de Bruijn word for $\mathcal{D}$.

Proof. Again, 0^{n-1} is $\mathcal{D}$-nonrepeating, and the set of primitive words satisfies (1). Next, consider $f_{\min}(w)$, which intuitively is the word obtained from iteratively extending w with primitive factors in a "prefer same" manner. It only remains to show that $f_{\min}(w)$ contains 0^{n-1} for all $w \in \{0,1\}^{n-1}$. Let $\ell \geq n$ be the smallest integer such that $f_{\min}(w)[\ell] \neq f_{\min}(w)[\ell+1]$. Such an ℓ must exist, as the algorithm would not produce a non-primitive factor of length n, and thus would not append the same symbol n consecutive times.

Next, $f_{\min}(w)[\ell] \neq f_{\min}(w)[\ell+1]$ means that setting $f_{\min}(w)[\ell+1] = f_{\min}(w)[\ell]$ would have created a non-primitive factor (as the construction of $f_{\min}(w)$ "prefers same"). Thus, by Lemma 4, $f_{\min}(w)[\ell+1\,..\,\ell+n] = (f_{\min}(w)[\ell+1])^{n-1}$. Now if $f_{\min}(w)[\ell] = 1$, then we have our factor of 0^{n-1} in $f_{\min}(w)$. Otherwise, if 0^{n-1} had not shown up earlier in $f_{\min}(w)$ already, $f_{\min}(w)[\ell+n]$ would be followed by a string of $n-1$ 0's (by Lemma 4 again). Thus, we see that $f_{\min}(w)$ contains 0^{n-1} in any case, and the result follows from Theorem 2. □

Thus, we obtain another way of generating a de Bruijn word for the set of primitive words in $\{0,1\}^n$ using a greedy algorithm. Furthermore, we see that the use of preference functions and Theorem 2 give us a template to streamline the analysis of the feasibility of using greedy algorithms to generate de Bruijn words for arbitrary dictionaries.

2.2 Concatenation of Lyndon words

Recall that a de Bruijn word for Σ_k^n can also be obtained from concatenating all Lyndon words of length dividing n in increasing lexicographic order. Next, we show that a de Bruijn word for the primitive words can be produced by a similar concatenation.

Theorem 9. *Let w be the concatenation of all Lyndon words in Σ_k^n in increasing lexicographic order. Then w is a de Bruijn word for the set of primitive words in Σ_k^n.*

Theorem 9 was first conjectured by Michael Domaratzki, who has a proof for the case $k = 2$ (personal communication, July 2013). Also, throughout this section, we will let k' denote the symbol $k-1$ to reduce cluttering.

Before we prove Theorem 9, we need the following result due to Cummings, who previously published a proof for the case $k = 2$ in [Cum88]. It is also implied by Duval's [Duv88] algorithm of generating Lyndon words.

Lemma 10. *Let $x \in \Sigma_k^n$ be a Lyndon word. Define $\ell := \max\{i : x[i] \neq k'\}$. If $\ell \geq 2$, then $y := x[1\,..\,\ell-1](k')^{n-\ell+1}$ is also a Lyndon word.*

That is, if we replace the last non-k' letter in a Lyndon word by k', the resulting word is also Lyndon (unless it is $(k')^n$). We are now ready to prove Theorem 9.

Proof of Theorem 9. If w is the concatenation of all Lyndon words of length n, then w has length n times the number of Lyndon words in Σ_k^n. Thus, the number of circular factors of w of length n is equal to the number of primitive words in Σ_k^n, and it suffices to show that each primitive word appears at least once in w (as that would imply that each primitive word appears exactly once). We do so by showing that given any Lyndon word x, its conjugate $x^{(i)} = x[i+1\,..\,n]x[1\,..\,i]$ appears in w as a circular factor, for all $i \in \{1,\ldots,n\}$.

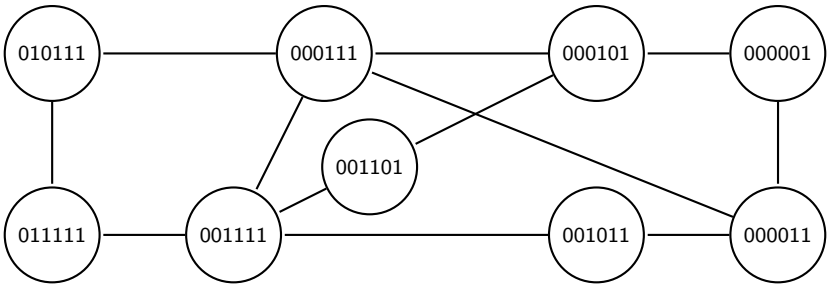

Figure 2: The Lyndon graph $\mathcal{L}_{6,2}$

First, obviously $x^{(n)} = x$ appears in w. Next, we write x as $x[1\,..\,\ell](k')^{n-\ell}$ such that $x[\ell] \neq k'$. If $\ell \geq 2$, then $y := x[1\,..\,\ell-1](k')^{n-\ell+1}$ is also Lyndon by Lemma 10. Thus, the Lyndon word that immediately follows x in w is sandwiched between x and y, and has prefix $x[1\,..\,\ell-1]$. Therefore, w contains the factor $x \cdot x[1\,..\,\ell-1]$, which contains the conjugates $x^{(1)}, x^{(2)}, \ldots, x^{(\ell-1)}$.

Next, we locate the factor $x^{(i)}$ in w, for all $i \in \{\ell, \ldots, n-1\}$. Note that $x^{(i)} = (k')^{i-\ell+1}x[1\,..\,\ell](k')^{n-i-1}$. Let y be the smallest Lyndon word that has prefix $x[1\,..\,\ell](k')^{n-i-1}$ (one must exist — x is one), and z be the Lyndon word that immediately precedes y in w. By the choice of y, $z[1\,..\,n+\ell-i-1] < y[1\,..\,n+\ell-i-1]$. Then by Lemma 10, the last $i-\ell+1$ symbols of z must all be k', and zy contains the factor $(k')^{i-\ell+1}x[1\,..\,\ell](k')^{n-i-1} = x^{(i)}$.

The remaining case when there is no Lyndon word preceding y in w implies $x[1\,..\,\ell](k')^{n-i-1}$ is the word of all 0s, and so $i = n-1$, and $x^{(i)} = (k')^{n-\ell}0^{\ell}$. Since the first and last Lyndon words in w are $0^{n-1}1$ and $(k'-1)k'^{n-1}$ respectively, w contains the circular factor $(k')^{n-1}0^{n-1}$, which must contain $x^{(i)}$. Hence, we are finished. □

As with the case of generating a de Bruijn word for Σ_k^n, concatenating Lyndon words is much more computationally efficient in generating a de Bruijn word for primitive words than using greedy algorithms, whose execution require exponential storage space.

2.3 Relating de Bruijn Graphs and Lyndon Graphs

Next, we detail yet another argument that shows the existence of de Bruijn words for primitive words. Unlike the two algorithmic proofs provided above, this argument is non-constructive, and makes use of connections between de Bruijn graphs and Lyndon graphs.

Given integers $n, k \geq 2$, we let $\mathcal{P}_{n,k}$ denote the de Bruijn graph of the set of primitive words in Σ_k^n. Also, let $\mathcal{L}_{n,k}$ denote the *Lyndon graph* of Σ_k^n, which has a vertex for each Lyndon word in Σ_k^n, and joins two Lyndon words by an edge if they differ in exactly one position. For example, Figure 2 illustrates the graph $\mathcal{L}_{6,2}$.

Notice that $\mathcal{L}_{6,2}$ only has one component. In fact, this is shown by Cummings to be true in general [Cum88].

Lemma 11. *$\mathcal{L}_{n,k}$ is connected for all $n, k \geq 2$.*

Proof. Given any pair of Lyndon words $x, y \in \Sigma_k^n$, Lemma 10 shows that there is a path from x to $x[1](k')^{n-1}$ in $\mathcal{L}_{n,k}$. Similarly, there is also a path between y and $y[1](k')^{n-1}$. Since

$x[1](k')^{n-1}$ is adjacent to $y[1](k')^{n-1}$, we see that there is a path between x and y in $\mathcal{L}_{n,k}$. Thus, $\mathcal{L}_{n,k}$ is connected. □

On the surface, $\mathcal{P}_{n,k}$ and $\mathcal{L}_{n,k}$ appear to have very little in common. First of all, the former is directed and the latter is not. Also, their vertices are represented by words of different lengths, with adjacency rules that are quite different. However, it turns out that they can be related through a series of basic graph operations.

Given a directed graph G, its *line graph* $L(G)$ is obtained by defining a vertex for each arc in G, and joining u and v in $L(G)$ if there is a vertex in G that is incident with their corresponding arcs. Note that while G is directed, $L(G)$ is undirected. Next, let G be an undirected graph and $S \subseteq V(G)$. Then *contracting* S in G yields the graph obtained from replacing the vertices in S by a single vertex v_S, and joining it to vertices in $V(G) \setminus S$ that was adjacent to some vertex in S.

Then we have the following:

Proposition 12. *Let $\mathcal{H}_{n,k}$ be the graph obtained from starting with $L(\mathcal{P}_{n,k})$, and successively contracting $\left\{x^{(i)} : i \in \{1, \ldots, n\}\right\}$ for all Lyndon words $x \in \Sigma_k^n$. Then $\mathcal{L}_{n,k}$ is a subgraph of $\mathcal{H}_{n,k}$.*

Proof. First, if during the contraction process, we label the vertex obtained from contracting $\left\{x^{(i)} : i \in \{1, \ldots, n\}\right\}$ by x for all Lyndon word $x \in \Sigma_k^n$, then it is easy to see that $\mathcal{H}_{n,k}$ and $\mathcal{L}_{n,k}$ have the same vertex set. Thus, it suffices to show that two Lyndon words are joined by an edge in $\mathcal{H}_{n,k}$ if they differ by exactly one position.

Let $u\alpha v$ and $u\beta v$ be two Lyndon words in Σ_k^n, where $u, v \in \Sigma_k^*$ and $\alpha, \beta \in \Sigma_k$. Observe that αvu and $vu\beta$ are both arcs in $\mathcal{P}_{n,k}$ (since they are both primitive), and share the vertex vu. Hence, αvu and $vu\beta$ are joined by an edge in $L(\mathcal{P}_{n,k})$. Since αvu and $vu\beta$ are conjugates of $u\alpha v$ and $u\beta v$ respectively, we see that $u\alpha v$ and $u\beta v$ are joined by an edge in $\mathcal{H}_{n,k}$. □

Figure 3 illustrates the transformation from $\mathcal{P}_{4,2}$ to $\mathcal{H}_{4,2}$, which turns out to be exactly the graph $\mathcal{L}_{4,2}$. In general, while $\mathcal{H}_{n,k}$ and $\mathcal{L}_{n,k}$ have the same vertex set, the former can have more edges. For instance, while 000011 and 001101 differ by three positions, they are adjacent in $\mathcal{H}_{6,2}$, since the arcs 000110 and 001101 share the vertex 00110 in $\mathcal{P}_{6,2}$.

Now we assemble the results in this section to provide yet another proof that a de Bruijn word for the primitive words exists, and we do that by showing that $\mathcal{P}_{n,k}$ has an Eulerian cycle.

First, Lemma 3 implies that every vertex in $\mathcal{P}_{n,k}$ has the same in-degree and out-degree. Thus, it suffices to show that the underlying undirected graph of $\mathcal{P}_{n,k}$ is connected.

To obtain a contradiction, suppose there are vertices u, v that belong to different components in $\mathcal{P}_{n,k}$. If we let x and y be arcs that are incident with u, v respectively, then x and y are in different components in $L(\mathcal{P}_{n,k})$. Next, observe that the n conjugates of any primitive word form a directed cycle of length n in $\mathcal{P}_{n,k}$. Thus, the n corresponding vertices cannot be spread across multiple components in $L(\mathcal{P}_{n,k})$, and hence $\mathcal{H}_{n,k}$ cannot have fewer components than $L(\mathcal{P}_{n,k})$.

However, $\mathcal{L}_{n,k}$ is shown to be connected, is contained in $\mathcal{H}_{n,k}$, and they have the same vertex set. Therefore, $\mathcal{H}_{n,k}$ only has one component, which implies that $L(\mathcal{P}_{n,k})$ is connected, a contradiction. Hence, we conclude that $\mathcal{P}_{n,k}$ has an Eulerian cycle, and there is a de Bruijn word for the set of primitive words in Σ_k^n.

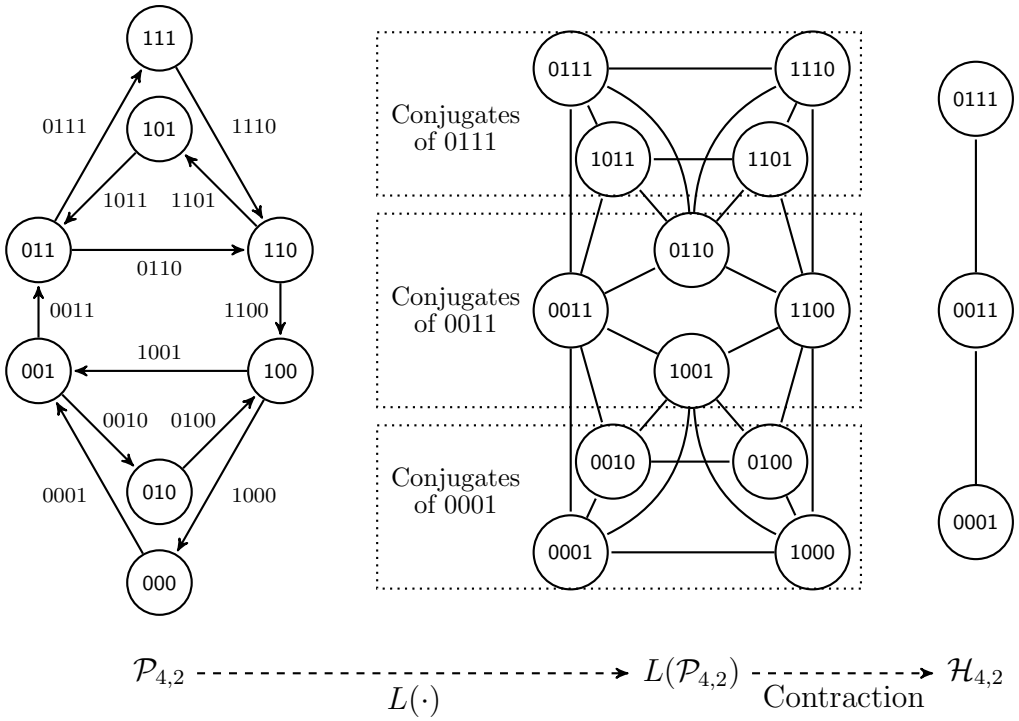

Figure 3: Transforming $\mathcal{P}_{4,2}$ to $\mathcal{H}_{4,2}$

In fact, if we extract the minimal ingredients we used the above argument, we obtain the following slightly stronger statement:

Corollary 13. *Let $\mathcal{D} \subseteq \Sigma_k^n$ be a dictionary that satisfies* (1)*, and has the property that for every pair of Lyndon words $u\alpha v, u\beta v \in \Sigma_k^n$ where $u, v \in \Sigma_k^*$ and $\alpha, \beta \in \Sigma_k$,*

$$\mathcal{D} \cap \{\alpha vu, vu\alpha\} \neq \emptyset \quad \text{and} \quad \mathcal{D} \cap \{\beta vu, vu\beta\} \neq \emptyset.$$

Then there is a de Bruijn word for $\mathcal{D}$.

Proof. Consider the de Bruijn graph $G^{\mathcal{D}}$, and let H be the graph obtained from contracting all the conjugate classes of the line graph of $G^{\mathcal{D}}$. Notice that the Lyndon words $u\alpha v, u\beta v$ differ by exactly one bit, and thus are adjacent in $\mathcal{L}_{n,k}$. Now if $\mathcal{D} \cap \{\alpha vu, vu\alpha\} \neq \emptyset$ and $\mathcal{D} \cap \{\beta vu, vu\beta\} \neq \emptyset$, that means $\mathcal{D}$ contains a conjugate of $u\alpha v$ and a conjugate of $u\beta v$ such that those two edges are both incident with the vertex vu in $G^{\mathcal{D}}$. As a result, $u\alpha v$ and $u\beta v$ are joined by an edge in H, and thus H contains $\mathcal{L}_{n,k}$ as a subgraph. This implies that H is connected, and consequently the underlying undirected graph of $G^{\mathcal{D}}$ is connected. Together with the fact that $\mathcal{D}$ satisfies (1), we conclude that $\mathcal{D}$ has a de Bruijn word. □

It would be interesting to know if any other properties of primitive words (or other families of words) and Lyndon words can be uncovered by this relation between their corresponding graphs. Establishing a tighter connection between these families of graphs (e.g. finding a transformation on $\mathcal{P}_{n,k}$ that yields exactly $\mathcal{L}_{n,k}$) could also lead to new and interesting findings.

3 Short sequences containing powers

While an arbitrary dictionary $\mathcal{D}$ may not have a de Bruijn word, there might be words of length not much larger than $|\mathcal{D}|$ that contains all words in $\mathcal{D}$ as circular factors. For instance, while we mentioned in the previous section that $\mathcal{D} := \{0000, 0001, 0011, 0111\}$ does not have a de Bruijn word, there are many sequences that contain all fours words in $\mathcal{D}$ as factors, with 0000111 being the shortest such sequence. Thus, in this regard, we can consider the word 0000111 as the closest thing to a de Bruijn word for $\mathcal{D}$, as there are no shorter sequences that contain all words in $\mathcal{D}$.

This motivates the following question: Given an arbitrary dictionary $\mathcal{D} \subseteq \Sigma_k^n$, what is the shortest word that contains all words in $\mathcal{D}$ as circular factors? Such a sequence can be seen as a generalization of de Bruijn words, since if a dictionary $\mathcal{D}$ has a de Bruijn word, that word must also be the shortest possible sequence that contains all words in $\mathcal{D}$ as circular factors.

In this section, we tackle the above question for a particular family of dictionaries, and try to find the shortest sequence that contains all p-powers in Σ_k^{pn} as circular factors. For $p = 1$, it is obvious that there is a de Bruijn word for all p-powers (it would just be a de Bruijn word for Σ_k^n). However, this does not apply for any $p > 1$. For instance, $\mathcal{D} = \{0000, 0101, 1010, 1111\}$ are the set of all squares in $\{0, 1\}^4$, and the shortest sequence that contains all four words as circular factors is $w = 000010101111$, which has length 12. More generally, if we let $\mathcal{D}$ to be the set of p-powers in Σ_k^{pn}, then $G^{\mathcal{D}}$ has as many components as the number of conjugacy classes in Σ_k^n. In fact, we shall soon see that any sequence that contains all k^n p-powers in Σ_k^{pn} must contain at least $(p-1)k^n$ factors of length pn that are not p-powers.

Define an equivalence relation on Σ_k^n, where $u \sim v$ if and only if they are conjugates of each other, and let $C(n, k)$ denote the number of conjugacy classes in Σ_k^n. It is well known that $C(n, k) = \sum_{d \geq 1 : d | n} \frac{\phi(d)}{n} k^{\frac{n}{d}}$, where $\phi(d)$ is Euler's totient function — the number of integers between 1 and d that are coprime with d. Note that $C(n, k) \geq \frac{k^n}{n}$ for all n, k.

Then we have the following:

Proposition 14. *Suppose $w \in \Sigma_k^*$ contains every p-power in Σ_k^{pn} as factors. Then $|w| \geq k^n + (p-1)nC(n, k) \geq pk^n$.*

Proof. Given $x, y \in \Sigma_k^n$, observe that if $x \not\sim y$, then any word that contains both x^p and y^p as factors has length at least $2pn - n + 1$. Therefore, every time two consecutive p-powers in w belong to different conjugacy classes, there are at least $(p-1)n$ factors of length pn in w in between that are not p-powers. Since there are $C(n, k)$ conjugacy classes in Σ_k^n, we see that w contains at least $(p-1)n(C(n, k) - 1)$ factors of length pn that are not p-powers.

Since w must also contain at least k^n factors that are p-powers, there are a total of at least $k^n + (p-1)n(C(n, k) - 1))$ factors of length pn in w. Hence

$$|w| \geq k^n + (p-1)n(C(n, k) - 1) + pn - 1 \geq k^n + (p-1)nC(n, k) \geq pk^n,$$

and our claim follows. □

Next, we show that there is a word w of length $\approx (p + \frac{1}{k})k^n$ over Σ_k that contains all p-powers of length pn. Given $u \in \Sigma_k^n$, define

$$\delta(u) := \frac{\min\left\{i \geq 1 : u^{(i)} = u\right\}}{n}.$$

Equivalently, $\delta(u)$ is the reciprocal of $\max\{p \geq 1 : u \text{ is a } p\text{-power}\}$. Note that $\delta(u) = 1$ if and only if u is primitive, and that $u^{p+\delta(u)-(1/n)}$ contains all p-powers of all conjugates of u as factors exactly once.

Next, we say that a word $s \in \Sigma_k^*$ is a *conjugate cover* of Σ_k^n if for every $u \in \Sigma_k^n$, s contains some circular factor of length $n-1$ in u. Conjugate covers exist for all n, k. For instance, if we take t to be a de Bruijn word for Σ_k^{n-1}, then $s := t \cdot t[1\,..\,n-2]$ is a conjugate cover, since it contains all words in Σ_k^{n-1} as factors. We then construct a word w that contains all p-powers in Σ_k^{pn} by the following algorithm:

Algorithm D. *Generating a sequence w that contains all p-powers in Σ_k^{pn}*

Input: *Integers n, k, p where $n, k \geq 2, p \geq 1$, and s a conjugate cover of Σ_k^n*
Set $w = \epsilon$ (the empty string)
Set $L = \Sigma_k^n$
for $j = 1, \ldots, |s| - n + 2$ **do**
  for $\alpha = 0, 1, \ldots, k-1$ **do**
    Set $u = s[j\,..\,j+n-2]\alpha$
    if $u \in L$ **then**
      Accept α *and append $u^{p+\delta(u)-1}$ to the end of w*
      Remove all conjugates of u from L
    else
      Reject α *and do not append anything*
    end
  end
  Append $s[j]$ to w
end
Append $s[|s| - n + 3\,..\,|s|]$ to w
return w

For example, consider the case $n = k = 3$ and $p = 2$. The word $s := 0221201100$ is a conjugate cover of $\{0,1,2\}^3$. In this case, Algorithm D would execute as follows:

j	$s[j\,..\,j+1]$	Accepted α's	Append to w	Removed from L
1	02	$0, 1, 2$	02002002102102202 20	Conjugates of $020, 021, 022$
2	22	$1, 2$	22122122222	Conjugates of $221, 222$
3	21	1	2112112	Conjugates of 211
4	12	0	1201201	Conjugates of 120
5	20	None	2	None
6	01	$0, 1$	0100100110110	Conjugates of $010, 011$
7	11	1	11111	111
8	10	None	1	None
9	00	0	00000	000

The algorithm finally appends 0 (the last symbol of s) to w, and outputs the word

$$\begin{aligned} w \;=\; & 02002002102102202\,20\;22122122222\;2112112\;1201201\;2 \\ & 0100100110110\;11111\;1\;00000\;0, \end{aligned}$$

which contains all squares of length 6 over $\{0, 1, 2\}$. Next, we show that the word generated by Algorithm D is not "too much" longer than the lower bound shown in Proposition 14.

Theorem 15. *Let w be the word constructed by Algorithm D. Then w contains x^p as a factor for all $x \in \Sigma_k^n$. Moreover, $|w| = k^n + (p-1)nC(n,k) + |s|$.*

Proof. Recall that, given $x \in \Sigma_k^n$, $x^{(i)} = x[i+1\,..\,n]x[1\,..\,i]$. We first prove that each p-power appears in w at least once by showing that for every $x \in \Sigma_k^n$, there exists $i \in \{1, \ldots, n\}$ such that w contains $(x^{(i)})^{p+\delta(x)-(1/n)}$ as a factor.

Let j be the smallest index such that $s[j\,..\,j+n-2]$ is a prefix of some conjugate of x, say $x^{(i)}$. Since s is a conjugate cover, such an index j must exist. Then we know that the algorithm would accept $\alpha = x^{(i)}[n]$ at step j, and $(x^{(i)})^{p-1+\delta(x)}$ is appended to w.

If at step j, some symbol larger than α is accepted, then we know the block $s[j\,..\,j+n-2] = x^{(i)}[1\,..\,n-1]$ immediately follows, giving us the desired power of $x^{(i)}$. Otherwise, we know that $s[j]$ gets added to w at the end of step j.

Then, if any symbol is accepted in step $j+1$, then $s[j+1\,..\,j+n-1]$ is added to w, and we get our desired power of $x^{(i)}$. Otherwise, we just add $s[j+1]$ at the end of step $j+1$. Proceeding in this manner, we see that the algorithm always adds $s[j\,..\,j+n-2]$ immediately after adding $(x^{(i)})^{p-1+\delta(x)}$ at step j. Since this holds for all $x \in \Sigma_k^n$, we see that w contains all p-powers in Σ_k^{pn}.

Next, we compute $|w|$. We have already found k^n factors of length pn that are p-powers. To count the other factors in w, we need to observe that, after accepting α_1 at step j, if the next symbol accepted by the algorithm is α_2 during step $j+\ell$, then there are exactly $(p-1)n+\ell$ factors of length pn in w between the last p-power in $(s[j\,..\,j+n-2]\alpha_1)^{p+\delta}$ and the first p-power in $(s[j+\ell\,..\,j+n+\ell-2]\alpha_2)^{p+\delta}$. Note that there could be p-powers among these blocks (e.g. when $\ell = 1$ and $\alpha_2 = s[j+n-1]$), but we nonetheless count them under the "other factors" category. Also, if the last symbol accepted by Algorithm D is α at step $|s| - n + 2 - \ell$, then there are ℓ factors of length pn in w after the last p-power in $u^{p+\delta(u)}$, where $u = s[|s| - n + 2 - \ell\,..\,|s| - \ell]\alpha$.

Since each symbol accepted by Algorithm D corresponds to a unique conjugacy class in Σ_k^n, we see that a total of $C(n,k)$ symbols are accepted throughout the algorithm. Therefore, w contains exactly $(p-1)n(C(n,k)-1) + |s| - n + 1$ of these "other factors" of length pn. Thus,

$$\begin{aligned} |w| &= k^n + (p-1)n(C(n,k)-1) + (|s| - n + 1) + pn - 1 \\ &= k^n + (p-1)nC(n,k) + |s|, \end{aligned}$$

and we are finished. □

As mentioned before, we can always construct a conjugate cover out of a de Bruijn word for Σ_k^{n-1}. In fact, we could do slightly better than that when $n-1$ is not prime:

Corollary 16. *Suppose $n, k \geq 2$, and let $\mathcal{D}$ be the set of primitive words in Σ_k^{n-1}. Then there exists a word w of length $k^n + (p-1)nC(n,k) + |\mathcal{D}| + n + k - 2$ that contains all p-powers in Σ_k^{pn} as factors.*

Proof. By Theorem 15, it suffices to show that there is a conjugate cover of Σ_k^n of length $|\mathcal{D}| + n + k - 2$. Let t be the de Bruijn word for $\mathcal{D}$ constructed by concatenating Lyndon words as described in Theorem 9. Then $|t| = |\mathcal{D}|$, and t contains α^{n-2} as a factor at least once for all $\alpha \in \Sigma_k$. We obtain s by replacing an instance of α^{n-2} in t by α^{n-1} for each $\alpha \in \Sigma_k$, and then appending 0^{n-2} at the end. It is easy to see that $|s| = |\mathcal{D}| + n + k - 2$, and s contains all words in $\mathcal{D}$, as well as α^{n-1} for all $\alpha \in \Sigma_k$, as factors.

To show that s is a conjugate cover, it suffices to show that for all $u \in \Sigma_k^n$, either it has a circular factor of length $n-1$ that is primitive, or $u = \alpha^n$ for some symbol α. Observe that, for any $i \in \{1, \ldots, n\}$, if neither $u[i+1\,..\,n]u[1\,..\,i-1]$ nor $u[i+2\,..\,n]u[1\,..\,i]$ is primitive, then $u[i+1] = u[i]$ by Lemma 3 and 4. Applying this argument on all i yields that $u = \alpha^n$ for some $\alpha \in \Sigma_k$, and it follows that s is a conjugate cover. □

Since the number of primitive words in Σ_k^{n-1} is less than k^{n-1}, we have now shown that the shortest sequence that contains all p-powers in Σ_k^{pn} has length roughly between pk^n and $(p + \frac{1}{k})k^n$. For $p = 1$, we know the truth is much closer to the lower bound, as there is a word of length $k^n + n - 1$ that contains all words in Σ_k^n as factors — any de Bruijn word of Σ_k^n with the first $n-1$ symbols repeated at the end would do.

Computational evidence suggests that this seems to be the case for $p = 2$ as well. Suppose we consider the special case of $k = p = 2$, and build a sequence that contains all squares in $\{0,1\}^{2n}$ by the following procedure:

Algorithm E. *Constructing a word w that contains all squares of length $2n$ over $\{0,1\}$*

Input*: Integer $n \geq 2$*
Set $w = 0^{2n}$
Set $L = \{0,1\}^n$
while $L \neq \emptyset$ **do**
  Pick $u \in L$ such that the prefix of u overlaps the most with the current suffix of w. If there is a tie, pick the lexicographically smallest u. Append to w such that w now has suffix $u^{2+\delta(u)-(1/n)}$.
  Remove all conjugates of u from L
end
return w

For any integer n, let $g(n)$ be the length of the sequence obtained by Algorithm E, and let $f(n) := \frac{g(n)}{2^n + nC(n,2)}$. Figure 4 illustrates the behaviour of $f(n)$ for $n \in \{4, \ldots, 25\}$.

By Corollary 16, the length of shortest word that contains all squares in $\{0,1\}^{2n}$ is bounded above by roughly $\frac{5}{4}(2^n + nC(n,2))$. However, we see that $f(n)$ appears to approach 1 as n increases, and there seems to be room for improvement for the upper bound. Perhaps constructing the shortest possible conjugate covers can improve the upper bound to, say, $k^n + (p-1)nC(n,k) + O(k^{n-1}/n)$. Also, we remark that the lower bound in Proposition 14 also holds for fractional powers p (given a positive real number p where pn is an integer, we can define $x^p := x^{\lfloor p \rfloor} x[1\,..\,(p - \lfloor p \rfloor)n]$). It would be interesting to know if "short" sequences that contains all p-powers for a fractional p exist, and whether there are efficient algorithms that generate short sequences that contains all p-powers in general.

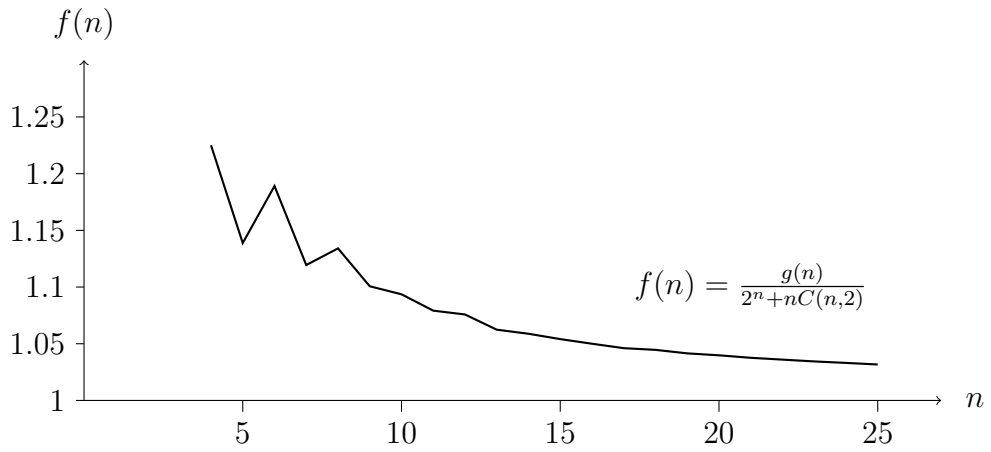

Figure 4: Computational results for $f(n)$

4 Acknowledgements

We would like to deeply thank Jeffrey Shallit, who brought to our attention the problems tackled in this manuscript. In particular, it was his suggestion that a greedy algorithm could be applied to generate a de Bruijn word for primitive words. He also provided many helpful comments on the earlier drafts of this manuscript.

Furthermore, we would like to express our gratitude towards the anonymous referees who reviewed this manuscript, and gave extremely detailed and helpful suggestions that have improved both the content and the presentation of this paper.

Finally, some of the findings in this manuscript were obtained while the author was at the University of Waterloo, supported in part by an NSERC Scholarship, a Tutte Scholarship and a Sinclair Scholarship.

References

[Alh10] Abbas M. Alhakim. A Simple Combinatorial Algorithm for de Bruijn Sequences. *American Mathematical Monthly*, 117(8):728–732, 2010.

[CDG92] Fan Chung, Persi Diaconis, and Ron Graham. Universal Cycles for Combinatorial Structures. *Discrete Mathematics*, 110(1):43–59, 1992.

[Cum88] Larry J. Cummings. Connectivity of Synchronizable Codes in the n-cube. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 3:93–96, 1988.

[dB46] Nicolaas Govert de Bruijn. A Combinatorial Problem. *Nederl. Akad. Wetensch., proc.*, 49:758–764, 1946.

[Duv88] Jean-Pierre Duval. Génération d'une section des classes de conjugaison et arbre des mots de lyndon de longueur bornée. *Theoretical Computer Science*, 60(3):255–283, 1988.

[FM78] Harold Fredricksen and James Maiorana. Necklaces of Beads in k Colors and k-ary de Bruijn Sequences. *Discrete Mathematics*, 23:207–210, 1978.

[Fre82] Harold Fredricksen. A Survey of Full Length Nonlinear Shift Register Cycle Algorithms. *SIAM Review*, 24(2):195–221, 1982.

[Joh09] J. Robert Johnson. Universal Cycles for Permutations. *Discrete Mathematics*, 309(17):5264–5270, 2009.

[Mar94] C. Flye-Sainte Marie. Solution to Problem Number 58. *l'Intermediare des Mathematiciens*, 1:107–110, 1894.

[Mar34] Monroe H. Martin. A Problem in Arrangements. *Bulletin of the American Mathematical Society*, 40(12):859–864, 1934.

[Mor05] Eduardo Moreno. De Bruijn sequences and De Bruijn graphs for a general language. *Inf. Process. Lett.*, 96(6):214–219, 2005.

[RSW92] Frank Ruskey, Carla Savage, and Terry Min Yih Wang. Generating Necklaces. *Journal of Algorithms*, 13(3):414–430, 1992.