DE BRUIJN SEQUENCES WITH VARYING COMBS

Abbas Alhakim[1]
Department of Mathematics, American University of Beirut, Beirut, Lebanon
aa145@aub.edu.lb

Steve Butler
Department of Mathematics, Iowa State University, Ames, Iowa
butler@iastate.edu

Ron Graham
Department of Computer Science and Engineering, University of California San Diego, La Jolla, California
graham@ucsd.edu

Received: 4/1/13, Accepted: 10/28/13, Published: 5/20/14

Abstract

For a given alphabet $\mathcal{A}$ and length n, a de Bruijn sequence corresponds to a string of length $|\mathcal{A}|^n$ where every string of length n occurs as a consecutive substring (and we allow the ends to wrap around). We consider the relaxation wherein the letters of the substring are not consecutive but rather fixed by some pattern, called a comb. We give several constructions showing how to construct some sequences for combs, as well as give several ways to form combs without de Bruijn sequences.

1. Introduction

Given an alphabet $\mathcal{A}$ and a string length n, a de Bruijn sequence is a string of length $|\mathcal{A}|^n$ which contains each possible string of length n composed of letters from $\mathcal{A}$ as a consecutive substring (with wraparound allowed at the ends). These sequences were named after N. G. de Bruijn who studied them for large alphabets [1], though they had been previously studied by Camille Flye Sainte-Marie in 1894 when $\mathcal{A} = \{0, 1\}$.

Much of the research of de Bruijn sequences has focused on their construction, enumeration (van Aardenne-Ehrenfest and de Bruijn showed that there are $(|\mathcal{A}|!)^{|\mathcal{A}|^{n-1}}/|\mathcal{A}|^n$ such sequences), or in establishing various extremal properties of such sequences.

[1]Research partially supported by the Center for Advanced Mathematical Sciences, American University of Beirut and CNRS, Lebanon ref. 01-01-13.

11101000			11100100			11010100		
000*****	→	111	00**0***	→	110	0*0*0***	→	100
*000****	→	110	*00**0**	→	111	*0*0*0**	→	111
000*	→	101	**00**0*	→	100	**0*0*0*	→	000
***000**	→	010	***00**0	→	000	***0*0*0	→	110
****000*	→	100	0***00**	→	011	0***0*0*	→	001
*****000	→	000	*0***00*	→	101	*0***0*0	→	101
0*****00	→	001	**0***00	→	001	0*0***0*	→	010
00*****0	→	011	0**0***0	→	010	*0*0***0	→	011

Table 1: Three examples of combs: 000, 00**0, 0*0*0.

In this note we will look at what happens when we relax the condition that the letters in the substring occur consecutively. Namely, we will allow for a "comb" which has some pattern of teeth (marked 0) through which we can read entries in the string and coverings (marked *) which we cannot read through. A de Bruijn sequence for a given comb is then a string of length $|\mathcal{A}|^n$ which contains each possible string in some shift of the comb where again we allow for the comb to wrap around at the ends. It is worth mentioning that de Bruijn sequences for a special comb are implicitly used in the construction of the well known generalized feedback shift register sequences (aka, GFSRs) out of regular linear feedback shift register sequences (aka, LFSRs), see [6].

As a demonstration we show in Table 1 three de Bruijn sequences for $\mathcal{A} = \{0, 1\}$ and $n = 3$, one for 000 (the original variation), one for 00**0, and one for 0*0*0.

In this note we will look at some very basic results about some simple combs for the case $\mathcal{A} = \{0, 1\}$, as well as examining a comb in relation to the de Bruijn sequences generated by linear feedback shift registers. Previous work for de Bruijn combs can be found in Krahn [5] and Cooper and Graham [2], the latter of which was highlighted by Diaconis and Graham [3, Chs. 2–3].

1.1. Combs

An alternative way to express a comb is to indicate which entries correspond to teeth in some cyclic shift (where by convention we start at 0). So for example, the combs illustrated above, 000, 00**0, and 0*0*0, are respectively $[0, 1, 2]$, $[0, 1, 4]$ and $[0, 2, 4]$.

Observation 1. Suppose we have a given comb $[a_1, a_2, \ldots, a_n]$ and a corresponding de Bruijn sequence $d_1 d_2 d_3 \ldots d_{|\mathcal{A}|^n}$ for that comb. Then for any k, ℓ with $\gcd(k, |\mathcal{A}|) = 1$ we have that $[ka_1 + \ell, ka_2 + \ell, \ldots, ka_n + \ell]$ is a comb for the sequence $d_k d_{2k} d_{3k} \ldots d_{k|\mathcal{A}|^n}$ where the subscripts are taken modulo $|\mathcal{A}|^n$.

This is easy to see since cyclic shifts of combs will not effect the sequence (i.e., we are going to look at arbitrary cyclic shifts regardless), and then if we scale both the entries of the comb and the relative locations of the sequence then we still have the same strings (visited in the same order as before). Note that a reflection of a comb can be achieved by scaling by -1 and then appropriately shifting.

In particular, this greatly reduces the number of combs that need to be examined. For example, up to scaling and shifting there are four combs when $|\mathcal{A}| = 2$ and $n = 3$. In addition to the 3 given above, the comb `00*0`$= [0, 1, 3]$ has no de Bruijn sequence.

The number of combs up to scaling and shifting for $|\mathcal{A}| = 2$ for the first few values of n are given in the table below.

n	2	3	4	5	6	7
number of combs	2	4	25	454	38494	3136831

More information about the combs and the corresponding number of such sequences for $n = 4, 5$ can be found in the Appendix.

2. Combs in Arithmetic Progression

The easiest combs to work with are those whose teeth form an arithmetic progression. Given Observation 1 we can conclude that we only need to be concerned about arithmetic progressions where the step size between teeth involves only factors from $|\mathcal{A}|$ (i.e., all other factors can be "scaled out"). In the case of $\mathcal{A} = \{0, 1\}$ this means we can limit ourselves to step sizes of the form 2^k for some k. Note the case $k = 0$ is the original form of the de Bruijn sequences.

Let us start by considering the comb `0*0*0`, with the corresponding de Bruijn sequence 11010100. We can split this sequence into two parts, i.e.,

$$11010100 \;\rightarrow\; {}^{1}{}_{1}{}^{0}{}_{1}{}^{0}{}_{1}{}^{0}{}_{0} \;\rightarrow\; 1000 \text{ and } 1110.$$

What we are doing is pulling out the two subsequences (two equals the step size between teeth in our comb) which our comb will alternate between. In other words, the comb `0*0*0` for the sequence 11010100 alternates between what the comb `000` produces for the pair of sequences 1000 and 1110.

This has a simple interpretation via the de Bruijn graph. Given an alphabet $\mathcal{A}$ and n, the de Bruijn graph is a directed graph whose vertices consist of strings of length $n - 1$ from $\mathcal{A}$ and $u \rightarrow v$ if the last $n - 2$ letters of u agree with the first $n - 2$ letters of v. For example, the case for $n = 3$ and $\mathcal{A} = \{0, 1\}$ is shown on the left in Figure 1.

Each edge in the de Bruijn graph is naturally associated with a string of length n and walks correspond to sequences of strings which can occur consecutively. In particular, a de Bruijn sequence corresponds to an Eulerian circuit. In our case w here

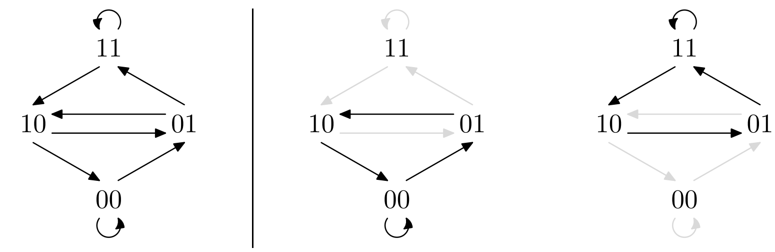

Figure 1: The de Bruijn graph for $\mathcal{A} = \{0, 1\}$ and $n = 3$, and a decomposition of its edges into two circuits of length 4.

we have split our sequence into two parts, 1000 and 1110, each part corresponding to a circuit in the graph. Namely the circuits

$$10 \to 00 \to 00 \to 01 \to 10 \quad \text{and} \quad 11 \to 11 \to 10 \to 01 \to 11,$$

illustrated on the right in Figure 1.

With the de Bruijn graph in hand we now have the following two observations.

Observation 2. Given a de Bruijn sequence for the comb $[0, 2^k, 2{\cdot}2^k, \ldots, (n{-}1){\cdot}2^k]$ then there is a decomposition of the de Bruijn graph into 2^k circuits of length 2^{n-k}. Namely, by forming the circuits corresponding to the substrings found by taking the 2^kth terms.

Observation 3. For each decomposition of the de Bruijn graph into 2^k circuits of length 2^{n-k}, there are $(2^k{-}1)!{\cdot}2^{(n-k)(2^k-1)}$ rotationally distinct de Bruijn sequences for the comb $[0, 2^k, 2 \cdot 2^k, \ldots, (n-1) \cdot 2^k]$.

The latter observation simply follows by noting that we must interlace these 2^k circuits. We do this by simply fixing one of them and placing the rest relative to that fixed circuit. In particular there are $2^k - 1$ circuits left to place which can be put down in any order, and each one of those circuits has length 2^{n-k} for which we can choose any rotational shift. As an illustration of this last idea we can rotate part of the de Bruijn sequence for `0*0*0` independently to derive another de Bruijn sequence. This is shown below.

$$11010100 \;\to\; {}^{1}{}_{1}{}^{0}{}_{1}{}^{0}{}_{1}{}^{0}{}_{0} \;\to\; {}^{1}{}_{0}{}^{0}{}_{1}{}^{0}{}_{1}{}^{0}{}_{1} \;\to\; 10010101$$

The problem of finding a de Bruijn sequence for a given arithmetic progression now has been reduced to finding a decomposition of the de Bruijn graph into circuits of equal length. Further, if we know the number of such decompositions the preceding observations give us a precise count for the number of such sequences. By computer search we have determined the number of such decompositions for some corresponding small combs:

comb	decompostions
$[0,2,4]$	1
$[0,2,4,6]$	3
$[0,2,4,6,8]$	112
$[0,4,8,12,16]$	1
$[0,4,8,12,16,20]$	3260
$[0,8,16,24,32,40]$	1
$[0,8,16,24,32,40,48]$	235290

This data indicates that for many cases there appears to be a large number of possible decompositions. We will prove something much weaker, namely the following result.

Theorem 4. *Given k and n so that $2^{n-k-1} \le n < 2^{n-k}$, there is a decomposition of the de Bruijn graph for $\mathcal{A} = \{0,1\}$ and n into circuits of length 2^{n-k}, in particular we have a de Bruijn sequence for the comb $[0, 2^k, 2\cdot 2^k, \ldots, (n-1)\cdot 2^k]$.*

Before giving the proof of the theorem we will need some basic facts about binomial coefficients and systems of equations.

Kummer's Theorem. *Given $n \ge m \ge 0$, and p a prime, the maximum k so that p^k divides $\binom{n}{m}$ is equal to the number of carries when m is added to $n-m$ in base p.*

Lemma 5. *If $i, j < 2^t$ then* $\binom{j}{i} \equiv \binom{j+2^t}{i} \equiv \binom{j+2^t}{i+2^t} \pmod 2$.

Proof. We claim a binomial coefficient $\binom{j}{i}$ will only be odd if for each 1 in the binary expansion of i the binary expansion of j will also have a 1 in the same position.

To show this, suppose $i > j$ then comparing the leading binary digits we have that the first digit where they differ will have a 1 in the digit for i and a 0 in the digit for j establishing this case. Otherwise we can apply Kummer's Theorem and note that 2 will divide $\binom{j}{i}$ only if there is at least one carry when adding the binary numbers $j-i$ and i. If there is a 1 in the expansion of i without a corresponding 1 in the expansion of j then there had to have been a carry (i.e., we have a 1 in that slot which needs to be added to something which gives a 0 in the slot; this could only happen if a carry at some stage was involved).

Finally we note that $\binom{j+2^t}{i}$ or $\binom{j+2^t}{i+2^t}$ will not change the existence of a 1 in the bottom term with a 0 in the top term. This establishes the result. ☐

Lemma 6. *For $0 \le i < 2^t$ and $s \in \{0,1\}$, let $x_0, \ldots, x_{2^t-1}$ satisfy the following system of equations modulo 2,*

$$\sum_{j=0}^{2^t-1} \binom{j}{\ell} x_j = \begin{cases} 0 & \text{if } \ell < i \\ s & \text{if } \ell = i. \end{cases}$$

Then for any m we also have

$$\sum_{j=0}^{2^t-1}\binom{j}{i}x_{j+m}=s$$

where the subscripts on the x terms are taken modulo 2^t.

Proof. We proceed by induction on i. First note that for $i=0$ we have the single equation

$$x_0+x_1+\cdots+x_{2^t-1}=s$$

which has all coefficients 1 and so is invariant under a cyclic shift of the indices.

Now suppose the statement holds up through i, and consider the statement for $i+1$. First we note that it suffices to show that the result holds for $m=-1$, i.e.,

$$\sum_{j=0}^{2^t-1}\binom{j}{i+1}x_{j-1}=s.$$

This follows since by the induction hypothesis we can shift all of the indices in the first i equations by 1, then we simply repeat this shifting by 1 to the full set of equations as many times as needed, and establish the result.

Finally, we note by combining the equations for $\ell=i+1$ and $\ell=i$ we have

$$\begin{aligned}s=0+s=\sum_{j=0}^{2^t-1}\binom{j}{i}x_j+\sum_{j=0}^{2^t-1}\binom{j}{i+1}x_j&=\sum_{j=0}^{2^t-1}\left(\binom{j}{i}+\binom{j}{i+1}\right)x_j\\&=\sum_{j=0}^{2^t-1}\binom{j+1}{i+1}x_j=\sum_{j=0}^{2^t-1}\binom{j}{i+1}x_{j-1}\end{aligned}$$

In the last step we used Lemma 5 and $1\le\ell\le 2^t-1$ to get $\binom{2^t}{\ell}\equiv\binom{0}{\ell}\equiv 0 \pmod 2$, which allows us to wrap the binomial coefficient around. □

Proof of Theorem 4. The desired decomposition will be formed by collecting all sequences of the form $x_0x_1\ldots x_{2^{n-k}-1}$, where the x_i satisfy the following system of equations modulo 2:

$$\sum_{j=0}^{2^{n-k}-1}\binom{j}{\ell}x_j=\begin{cases}0 & \text{if } 0\le\ell<2^{n-k}-n-1\\ 1 & \text{if } \ell=2^{n-k}-n-1.\end{cases}$$

Given any string $a_0a_1\ldots a_{n-1}$ we can set $a_i=x_{2^{n-k}-n+i}$, this leaves us with a total of $2^{n-k}-n$ variables to determine with a linear system of $2^{n-k}-n$ equations. Further, this is an invertible system (written in matrix form this is an upper diagonal matrix with 1's on the diagonal) so we can solve for the remaining x_i. Therefore we have that every string shows up at least once in our collection.

Further, by Lemma 6 once we specify the n entries of our string to appear anywhere consecutively among the x_i then the resulting solution will be the same string up to a cyclic shift. Therefore if we group all the resulting sequences by those equivalent under cyclic shifts and take a representative from each group, then each string of length n will appear in precisely one of these representatives.

It remains to show that no string of length n appears twice in one of our representatives. If a string did appear twice then by application of Lemma 6 we would conclude that the resulting string was periodic. Since the length of our string is a power of 2 then we must conclude that our solution is invariant under a shift by the variables 2^{n-k-1}, i.e., for $0 \le j < 2^{n-k-1}$ we have $x_j = x_{j+2^{n-k-1}}$. Combining this with Lemma 5 we have

$$\begin{aligned}
&\sum_{j=0}^{2^{n-k}-1} \binom{j}{2^{n-k}-n-1} x_j \\
&\quad = \sum_{j=0}^{2^{n-k-1}-1} \left(\binom{j}{2^{n-k}-n-1} x_j + \binom{j+2^{n-k-1}}{2^{n-k}-n-1} x_{j+2^{n-k-1}} \right) \\
&\qquad = 2 \sum_{j=0}^{2^{n-k-1}-1} \binom{j}{2^{n-k}-n-1} x_j = 0 \pmod 2.
\end{aligned}$$

which contradicts the set of equations. Therefore we can conclude that among our representative solutions each string of length n appears precisely once, i.e., we have our desired decomposition of the de Bruijn graph. □

Examples of the construction from Theorem 4 are given in Table 2.

We note that this is only one decomposition and this technique will not capture every decomposition. For example, the following is a decomposition of the de Bruijn graph for $n = 6$ into 4 circuits which has circuits containing both an even and an odd number of 1s, i.e., automatically fails the first equation:

1101100001000000
1110010010101000
1111011101001100
1111110001011010

Nevertheless, at one extreme this captures the *unique* decomposition.

Corollary 7. *For the de Bruijn graph with $\mathcal{A} = \{0, 1\}$ and for strings of length $n = 2^k - 1$, the unique way to decompose the graph into circuits of length 2^k is to take all possible strings of length 2^k (up to cyclic shifting) that have an odd number of* 1*s.*

n and k	equations	decomposition
$n=6$ $k=3$	$x_0+x_1+x_2+x_3+x_4+x_5+x_6+x_7=0$ $x_1+x_3+x_5+x_7=1$	11000000 01000010 11000101 01000111 11001010 01001101 11001111 11011011
$n=5$ $k=2$	$x_0+x_1+x_2+x_3+x_4+x_5+x_6+x_7=0$ $x_1+x_3+x_5+x_7=0$ $x_2+x_3+x_6+x_7=1$	10100000 01100011 11100100 11101011
$n=4$ $k=1$	$x_0+x_1+x_2+x_3+x_4+x_5+x_6+x_7=0$ $x_1+x_3+x_5+x_7=0$ $x_2+x_3+x_6+x_7=0$ $x_3+x_7=1$	11110000 11010010

Table 2: Examples of the construction from Theorem 4.

Proof. In Theorem 4 this corresponds to the situation where $\sum x_i = 1 \pmod 2$. We know that this gives a decomposition. It remains to show that this decomposition is unique.

This follows by combining several observations. First, the all zero string must show up somewhere and so it can only be either $00\ldots00$ or $00\ldots01$, but the first one is impossible in a de Bruijn decomposition as $00\ldots0$ would appear multiple times. So we can conclude that $00\ldots01$ is one of our circuits in our decomposition. Further, in any circuit knowing the edge that is used coming into a vertex and out of that vertex will completely determine the circuit that it lies on, i.e., this determines $n = 2^k$ consecutive terms in our circuit. Finally, in the de Bruijn graph the in-degree and the out-degree at each vertex is 2.

Combining all of this we start with the one circuit which we know must be in the sequence, then we simply look for any vertex it passed through, it has one remaining edge coming in and out and we must use those in combination in another circuit which in turn forces other circuits and repeating this we are forced in our selection of circuits for the entire graph. □

3. Combs Related to LFSR Sequences

One of the best known techniques for generating de Bruijn sequences is to use linear feedback shift registers (see Golomb [4]). These are based off of irreducible polynomials in the ring $\mathbb{Z}_2[x]$ and using the polynomials to build a linear recursion. So for example, x^4+x^3+1 is irreducible and this corresponds to the linear feedback shift register $x_i = x_{i-3} + x_{i-4}$ where we work modulo 2 (note the powers in the polynomial give the corresponding shifts of terms to examine). If we initiate the sequence with 0001 then this will generate the sequence:

$$\underbrace{000100110101111}_{\text{a near de Bruijn sequence}}\ 000100110101111000100110101111000100110101111\ldots$$

In particular this will generate a pattern with period 15 that hits every string of length 4 other than 0000. This can be easily fixed by taking the unique occurrence of 000 and replacing it with 0000. We can also rearrange the terms of the recurrence so that we can run it backwards, i.e., we also have $x_{i-4} = x_{i-3} + x_i$ (one of the advantages of working modulo 2). This allows us to start with any initial seed and repeatedly prepend the sequence. Using either approach we will generate the same 15-periodic sequence which can be corrected to form a de Bruijn sequence.

In general, given any irreducible polynomial the above procedure gives a de Bruijn sequence. We will show that in some cases these irreducible polynomials can produce other combs as well.

Theorem 8. *Consider an irreducible polynomial of degree $n \geq 3$. Then exactly one of x or x^{n-1} is in the polynomial if and only if the resulting de Bruijn sequence constructed using the above technique will also work for the comb* `0*00...0*0`*.*

Proof. We start by establishing that if precisely one of x or x^{n-1} is in the polynomial then the comb works. First we will show that the sequence we generate where we do *not* correct for the missing $00\ldots0$ term hits all but one term using the comb. We will then show that by making the needed correction we will hit the missing term and still keep the appearance of every other term. We will work through the case when the polynomial has x^{n-1} but not x (the other case is handled similarly).

Since the recurrence for x_k has no term x_{k-1}, the LFSR sends both $yx_1\ldots x_{n-2}z$ *and* $yx_1\ldots x_{n-2}\overline{z}$ respectively to $x_1\ldots x_{n-2}zw$ and $x_1\ldots x_{n-2}\overline{z}w$, where y, z, w are 0-1 and $\overline{z}$ is the complement of z. This follows since the next term will be independent of the value in the z ($\overline{z}$) position.

This in turn implies that $\overline{y}x_1\ldots x_{n-2}z$ *and* $\overline{y}x_1\ldots x_{n-2}\overline{z}$ respectively go to $x_1\ldots x_{n-2}z\overline{w}$ and $x_1\ldots x_{n-2}\overline{zw}$ since the recurrence does involve the term x_{k-n}.

We can also run the recurrence backward, and since our polynomial of degree n has a term x^{n-1} then the term we prepend will depend on the current first term. Doing so we get that $yx_1\ldots x_{n-2}z$ *and* $\overline{y}x_1\ldots x_{n-2}z$ came respectively from $tyx_1\ldots x_{n-2}$ *and* $\overline{t}\overline{y}x_1\ldots x_{n-2}$ for some 0-1 value t. So that $yx_1\ldots x_{n-2}\overline{z}$ *and* $\overline{y}x_1\ldots x_{n-2}\overline{z}$ come respectively from $\overline{t}yx_1\ldots x_{n-2}$ *and* $t\overline{y}x_1\ldots x_{n-2}$.

Combining all the above we get the following occurring in our sequence

$$\begin{array}{l} t\overline{y}x_1\ldots x_{n-2}\overline{z}\overline{w} \\ \overline{t}yx_1\ldots x_{n-2}\overline{z}w \\ \overline{t}\overline{y}x_1\ldots x_{n-2}z\overline{w} \\ tyx_1\ldots x_{n-2}zw. \end{array}$$

Now if we apply the comb and observe that $x_1\ldots x_{n-2}$ occurs 4 times in the original LFSR sequence (except for $00\ldots 0$ that occurs three times), we see that every pattern of size n now occurs once in this sequence; the only missing term is $00\ldots 0$ (this would have come from the case when $y = x_1 = \cdots = x_{n-2} = z = 0$ which in turn would give $t = w = 0$).

We now proceed to the second step, which is to show that we can insert the 0 into the sequence and the result will be to pick up the missing $00\ldots 0$ term without losing/gaining any other terms. To do this we note that we initially have the following for some x and y:

$$\ldots x11\underbrace{00\ldots 0}_{n-1}10y\ldots.$$

This follows by first noting that we must have the $100\ldots 01$ term in our sequence and then applying the recurrence to get the adjacent terms. If we now apply our comb to this pattern we have:

$$\begin{array}{lcl} \ldots x1100\ldots 0010y\ldots & & \\ \texttt{0*000...*0***} & \to & x10\ldots 000 \\ \texttt{*0*00...0*0**} & \to & 100\ldots 001 \\ \texttt{**0*0...00*0*} & \to & 100\ldots 000 \\ \texttt{***0*...000*0} & \to & 000\ldots 01y \end{array}$$

Next we compare it to what happens *after* we add the missing 0 term.

$$\begin{array}{lcl} \ldots x11000\ldots 00010y\ldots & & \\ \texttt{0*0000...*0****} & \to & x10\ldots 000 \\ \texttt{*0*000...0*0***} & \to & 100\ldots 000 \\ \texttt{**0*00...00*0**} & \to & 100\ldots 001 \\ \texttt{***0*0...000*0*} & \to & 000\ldots 000 \\ \texttt{****0*...0000*0} & \to & 000\ldots 01y \end{array}$$

Note that no other shifted comb will be effected by the insertion other than the ones given above. Comparing the above two sequences we see that we have kept the same patterns (though in different order) and we have now picked up the missing all 0 pattern. In particular the resulting sequence has each possible occurrence once in a shifted comb, as desired.

To establish the other direction we note that if both x and x^{n-1} occur in the primitive polynomial then by following through on the first half of the above argument we will see that the following patterns all occur in the unaltered LFSR sequence:

$$\begin{array}{c} t\overline{y}x_1\dots x_{n-2}\overline{z}w \\ tyx_1\dots x_{n-2}zw \\ \overline{t}yx_1\dots x_{n-2}\overline{zw} \\ \overline{ty}x_1\dots x_{n-2}z\overline{w} \end{array}$$

Applying the comb 0*00...0*0 will result in multiple double occurrences of words in the sequence, which altering will only effect a small number of and so we cannot get a de Bruijn sequence.

On the other hand if both x and x^{n-1} do not occur in the primitive polynomial then by following through on the second half of the above argument we see that the location of the placement of the extra 0 will occur at the following:

$$\dots x01\underbrace{00\dots0}_{n-1}10y\dots.$$

Now let us consider what happens *after* we add the missing 0 term.

$$\begin{array}{lcl} \dots x01000\dots00010y\dots & & \\ \texttt{0*0000...*0****} & \to & x10\dots000 \\ \texttt{*0*000...0*0***} & \to & 000\dots000 \\ \texttt{**0*00...00*0**} & \to & 100\dots001 \\ \texttt{***0*0...000*0*} & \to & 000\dots000 \\ \texttt{***0*0...0000*0} & \to & 000\dots01y \end{array}$$

This gives two occurrences of $00\dots0$ and so the sequence cannot be de Bruijn for the given comb. □

Experimentation with de Bruijn sequences generated from small LFSRs have some suggestive patterns for combs; see Table 3. In particular, it appears that the combs generically fall into the class $W\texttt{0}^k\overline{W}$ where W is a word in $\{\texttt{0},*\}$ and $\overline{W}$ is the word formed by swapping 0 and *. There still remains a lot of work to do in this direction in determining when a particular comb applies to a given LFSR.

Another option to consider when working with LFSRs is the placement of the missing 0. In the original formation of the de Bruijn sequence from an LFSR we have no freedom in our placement since there is only one location it can go into to give us the missing all 0 term. However, when we are dealing with more general combs, this no longer needs to be the case. A different placement might work for a particular comb. While experimentation has not revealed anything satisfying, we did come across the following interesting case which is based off of the irreducible polynomial $x^6 + x^4 + x^3 + x + 1$:

$$\begin{array}{l} 0000011100000100100011011001011010111011110011000101010011111101 \\ \qquad\quad\uparrow \end{array}$$

where we have marked the location of the added 0 into the sequence (note that normally we would have placed it in the first block of 0s). This sequence has several

Irreducible polynomial	Some combs for the corresponding de Bruijn sequence
x^2+x+1	000
x^3+x+1	000, 0*0*0
x^4+x+1	0000, 0*00*0, 00*0**0, 0**0*00
x^5+x^2+1	00000
$x^5+x^3+x^2+x+1$	00000, 0*000*0, 00*00**0
$x^5+x^4+x^2+x+1$	00000
x^6+x+1	000000, 0*0000*0, 00*000**0, 0**000*00, 000*00***0, 00**00**00, 0*0*00*0*0, 0***00*000, 0000*0****0, 000**0***00, 00*0*0**0*0, 0*00*0*0**0, 00***0**000, 0*0**0*0*00, 0**0*0*00*0, 0****0*0000
$x^6+x^4+x^3+x+1$	000000, 0*0000*0

Table 3: Combs for some small value LFSRs.

combs that work including 0**00*00**0, 00*00**0**0, and 0**0**00*00. Again, what happens in general is an open question.

4. Impossible Combs

So far we have looked at ways to find sequences which work for various combs. There is also the converse problem, namely to identify combs for which there are no such sequences. As mentioned before the only comb which does not work for $n = 3$ is 00*0$= [0, 1, 3]$ (and anything which can be found by scaling/shifting this comb). The lists of all combs which do not have a sequence for $n = 4, 5$ are given in the Appendix.

Some of these combs are easy to see for $n = 4$, namely $[0, 4, 8, 12]$, $[0, 2, 8, 10]$ and $[0, 1, 8, 9]$. For these patterns suppose we had a de Bruijn sequence, then if we shift by 8 our teeth in the comb will be lined up over the same slots, and so in particular 0000 will occur twice as we run through the possible shifts which contradicts the sequence being de Bruijn.

More generally we have the following construction.

Construction 1. Let $n = a2^k + b$ with $0 \le b < k$ and let T be any pattern of distributing a teeth in 2^{n-k} entries. Construct the comb by dividing up 2^n into blocks of length 2^{n-k} and in each block placing T (i.e., so any two occurrences of T differ by a shift of some multiple of 2^{n-k}) and then the remaining b terms arbitrary to any remaining slots.

Verification that this does not have a de Bruijn sequence. Suppose that there were a de Bruijn sequence for the comb. Then it would follow that there is some cyclic shift which has all 0s. From that shift now consider the 2^k shifts where at each stage we move 2^{n-k} slots. The teeth in our comb which came from the equally spaced occurrences of T would still remain 0, and hence the only entries which would vary are the b remaining slots. However, these slots can only take on at most 2^b values and so if $2^b < 2^k$ then there must be a repetition that occurred in these cyclic shifts, a contradiction. □

So for example for $n = 5$ we can use this to rule out $[0, 1, 8, 16, 24]$, $[0, 2, 8, 16, 24]$, and $[0, 4, 8, 16, 24]$.

We can also bootstrap our way to larger combs by using smaller combs via the following construction.

Construction 2. Let $n = a2^k + k$ and let $C \in \{\mathtt{0}, *\}^{2^k}$ be a comb with k teeth which does not have a de Bruijn sequence. Further let T be any pattern of distributing a teeth in 2^{n-k} entries and let T' be T with one additional tooth added arbitrarily. Construct a new comb by "blowing up" C, namely by replacing `0` with T' and `*` with T.

Verification that this does not have a de Bruijn sequence. Suppose that there were a de Bruijn sequence for the comb. Then it would follow that there is some cyclic shift which has all 0s. From that shift now consider the 2^k shifts where at each stage we move 2^{n-k} slots. If we now restrict our attention to the location of the extra slot that was formed in T', this gives us a sequence of length 2^k that the comb C will go over. Further, we need to have all of these be distinct (since all other slots are 0's) and therefore we need to have a de Bruijn sequence for C, a contradiction. □

For example, starting with the forbidden comb `00*0****` for $n = 3$ we can form the following forbidden comb for $n = 11$ (here $T =$ `0**...` and $T' =$ `00*...`),

$$[0, 1, 256, 257, 512, 768, 769, 1024, 1280, 1536, 1792]$$

5. Concluding Comments

We have looked at some very specific combs and the corresponding de Bruijn sequences for our alphabet of $\mathcal{A} = \{0, 1\}$. For combs in arithmetic progressions we have some simple constructions which can produce some special combs, though this construction is far from exhaustive and some more general techniques are still waiting to be developed to handle finding all ways to decompose the de Bruijn graph into equal length circuits. We note that the construction we gave in Section 2 works for alphabets of prime size by similar arguments.

We have also looked at LFSRs and seen an example of how they can give nontrivial combs. However these combs again appear to have a very restrictive structure and constructions for more general combs would be of interest.

One can also consider various problems related to the possible structure of such sequences. For example, for normal de Bruijn sequences for strings of length n there must be n consecutive 0's and n consecutive 1's but no longer consecutive equal-valued strings. For general combs this no longer needs to hold (particularly for combs in arithmetic progressions). For example, consider the following combs and corresponding de Bruijn sequences where the first one has no four consecutive identical terms, while the second one has eight consecutive ones:

$$\begin{array}{rl} [0,1,8,9,16]: & 11100110011100011011001100100100 \\ [0,1,7,17,23]: & 11111111001001000101001010001100 \end{array}$$

Another extremal problem would be to focus on the possible 0-1 sequences and determine which such sequence has the most combs for which it works.

The biggest open problem in this area concerns the determination of whether a given comb has a de Bruijn sequence. We have given some simple constructions that can rule out a few simple combs. However, there is currently no better method than exhaustion at this point to rule out a generic comb (and exhaustion is very exhausting considering the search space involved). While we have highlighted what happens here for the small cases of $n = 3, 4, 5$ (where 1/4, 16/25, and 224/454 of all combs do not have de Bruijn sequences), the general case remains elusive. In particular it appears that most combs are very sensitive and have only one or two de Bruijn sequences, or none at all (when the step sizes are all multiples of 2 or 4 the high numbers can be misleading), this leads us to propose the following conjecture.

Conjecture. As n gets large almost all combs do not have a de Bruijn sequence, i.e., the fraction of combs with a de Bruijn sequence goes to 0.

We can also change the problem to form sequences of length $p \cdot |\mathcal{A}|^n$ and then insist that every string of length n occurs exactly p times as a substring. We have considered the case when $p = 1$, and Krahn [5] considered the case when $p = 2$ (i.e., which has the interpretation that every edge in the de Bruijn graph is used exactly twice).

In general, given a comb $C = [0, a_1, \ldots, a_{k-1}]$ where $0 < a_1 < \cdots < a_{k-1}$ we will define the *weight* of C to be k and *span* of C to be $a_{k-1} + 1$ (i.e., the distance between the two furthest teeth). If we let the *index* of C be the smallest possible p so that there is a de Bruijn sequence where each substring appears exactly p times then we have the following.

Fact 1. For any comb C we have $index(C) \leq 2^{span(C)-weight(C)}$.

This follows by simply noting that we can form a de Bruijn sequence for $n = a_{k-1} + 1$ and then each comb will appear precisely 2 raised to the power of the number of times that we have non-windows in our pattern. In particular, for every comb C there is some smallest p that works. We have already noted for the comb

$[0, 1, 3]$ that $p > 1$, and the above fact shows that $p \leq 2$; the latter case can be done by using a de Bruijn sequence, though other possibilities exist such as 1111100101001000. We note that every comb on four teeth has $index(C) \leq 2$. It is unknown if the index of a comb can be arbitrarily large.

We look forward to seeing more progress about de Bruijn sequences for combs.

Acknowledgements We thank Hal Fredricksen for helpful discussions and also for pointing out several important references in the literature in regards to combs.

References

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

[2] J. Cooper and R. Graham, Generalized de Bruijn circuits, *Ann. Comb.* **8** (2004), 13–25.

[3] P. Diaconis and R. Graham, *Magical Mathematics: The mathematical ideas that animate great magic tricks*, Princeton, New Jersey, 2012.

[4] S. Golomb, *Shift Register Sequences*, Holden-Day, Inc., 1967.

[5] G. Krahn, *Double Eulerian circuits on de Bruijn digraphs*, Ph.D. dissertation, Naval Postgraduate School, 1994.

[6] T. G. Lewis and W. H. Payne, Generalized feedback shift register pseudorandom number algorithm, *J. ACM* **3** (1973), 456–468.

Appendix: Combs of Length 4 and 5

In the table below is the list of every comb of length 4 which has a 0-1 de Bruijn sequence corresponding to that comb, as well as all such sequences (up to cyclic shifts and swapping 0 and 1).

[0, 2, 4, 6]	1111101100001000, 1111101100000100, 1111101000100100, 1111100100100010, 1111100100000110, 1111100010000110, 1111011100001000, 1111010100011000, 1111000110001010, 1111000100001110, 1110110100010010, 1110101100010100, 1110100100010110, 1110010100011010, 1101010100110010, 1101010100101100
[0, 1, 2, 3]	1111011001010000, 1111011000010100, 1111010110010000, 1111010011000010, 1111010010110000, 1111010000110010, 1111001011010000, 1111001010000110
[0, 1, 2, 7]	1111010100110000, 1111010010110000, 1111001101010000, 1111001011010000, 1111000101000110
[0, 1, 3, 14]	1111011001010000, 1111010110010000, 1111010010110000, 1111001011010000
[0, 1, 2, 6]	1111011010001000, 1111000101000110, 1111000100010110
[0, 1, 3, 8]	1111100010010100, 1111010000110010
[0, 1, 3, 4]	1111100101000100, 1111100100010100
[0, 1, 3, 7]	1111100101000100
[0, 1, 3, 9]	1111100100101000

The following combs of length 4 have no corresponding sequence for the de Bruijn comb.

[0, 1, 2, 8]	[0, 1, 2, 5]	[0, 1, 3, 5]	[0, 1, 4, 5]	[0, 1, 7, 8]	[0, 2, 6, 8]
[0, 2, 8, 10]	[0, 1, 3, 12]	[0, 4, 8, 12]	[0, 1, 4, 8]	[0, 2, 4, 10]	
[0, 1, 2, 4]	[0, 1, 2, 9]	[0, 1, 4, 9]	[0, 1, 8, 9]	[0, 2, 4, 8]	

In the table on the following pages is the list of every comb of length 5 which has a 0-1 de Bruijn sequence corresponding for that comb, the number of such sequences (up to cyclic shifts and swapping 0 and 1), and an example of one such sequence.[2]

[2]A complete list of all sequences for each comb of length 5 is available from the second author.

[0, 4, 8, 16, 20]	3072	11111111100011001010110000100000
[0, 4, 8, 12, 20]	1536	11111110110110101010010010000000
[0, 4, 8, 12, 16]	1536	11111110110110101010010010000000
[0, 1, 2, 3, 4]	1024	11111011100110101100010100100000
[0, 2, 4, 6, 8]	912	11111110110100111001000001001000
[0, 2, 4, 8, 28]	304	11111110110100111000010010001000
[0, 2, 6, 16, 18]	192	11111111100001000100101100110000
[0, 2, 4, 16, 18]	192	11111111001010011001011001000000
[0, 1, 2, 16, 17]	192	11111101101100010110010000101000
[0, 1, 2, 4, 30]	144	11111011100110101100010100100000
[0, 2, 8, 10, 16]	128	11111110111001000011001000101000
[0, 2, 6, 10, 28]	128	11111110101100110000010010100100
[0, 2, 4, 8, 12]	112	11111110110001110000001100101000
[0, 2, 4, 6, 18]	112	11111111001010001001010010011000
[0, 2, 6, 10, 14]	112	11111110110001110000000100111000
[0, 2, 4, 8, 18]	96	11111110110001101001100010100000
[0, 1, 8, 9, 16]	96	11111110111000100101010001001000
[0, 1, 7, 16, 17]	88	11111110100101000011001001011000
[0, 2, 4, 8, 20]	80	11111110110001101001100010100000
[0, 2, 6, 10, 18]	80	11111110101011000011001001100000
[0, 2, 4, 8, 16]	80	11111110110001101001100010100000
[0, 2, 4, 8, 24]	80	11111110110001101001100010100000
[0, 2, 4, 12, 20]	80	11111110110000101001100010100100
[0, 2, 4, 12, 16]	80	11111110110000101001100010100100
[0, 2, 4, 14, 22]	64	11111110110010110000001000111000
[0, 2, 4, 6, 16]	64	11111111001001101001100101000000
[0, 2, 6, 14, 16]	64	11111110110001101001100010100000
[0, 2, 6, 8, 16]	64	11111110101100001000011011001000
[0, 2, 4, 12, 14]	64	11111110110010110000001000111000
[0, 2, 4, 12, 18]	64	11111110110000101001100010100100
[0, 2, 6, 10, 16]	64	11111110101011000011001001100000
[0, 1, 7, 17, 23]	48	11111111001001000101001010001100
[0, 1, 7, 8, 16]	48	11111110010110000011001010010100
[0, 2, 4, 14, 16]	48	11111111001001101001100101000000
[0, 2, 4, 8, 26]	48	11111110011101001000000110010100
[0, 2, 4, 14, 20]	48	11111111001001101001100101000000
[0, 1, 4, 8, 17]	34	11111110001011100101010010000100
[0, 1, 2, 3, 17]	32	11111011101011000011001010010000
[0, 1, 4, 12, 28]	32	11111110011010000101010011000010
[0, 2, 4, 6, 14]	32	11111110110001110000000100111000

[0, 1, 4, 12, 17]	32	11111110011010000101010011000010
[0, 1, 4, 8, 16]	32	11111110001011100101010010000100
[0, 1, 4, 8, 24]	32	11111110001011100101010010000100
[0, 1, 5, 13, 16]	32	11111110110010100101010001100000
[0, 2, 4, 8, 22]	32	11111110110001110000000100111000
[0, 1, 4, 12, 16]	32	11111110011010000101010011000010
[0, 1, 4, 20, 28]	32	11111110011010000101010011000010
[0, 1, 4, 17, 28]	32	11111110011010000101010011000010
[0, 2, 4, 8, 10]	32	11111110011010001011100001001000
[0, 1, 4, 8, 20]	32	11111110001011100101010010000100
[0, 1, 2, 13, 21]	31	11111101100000110001011001010100
[0, 1, 4, 18, 19]	30	11111101001100100101011100010000
[0, 1, 2, 12, 13]	23	11111101101010011000100110100000
[0, 1, 2, 6, 28]	21	11111011010001101011100100000100
[0, 1, 2, 15, 19]	20	11111101011001001101100101000000
[0, 1, 3, 5, 30]	20	11111101100101010011100010010000
[0, 1, 2, 10, 11]	19	11111101100011001010001011000010
[0, 1, 2, 8, 9]	18	11111100011000110100010101011000
[0, 1, 2, 4, 18]	18	11111100101100010101100010000110
[0, 1, 2, 6, 7]	17	11111101100101001011000000100110
[0, 1, 4, 8, 28]	17	11111110001101101010100001001000
[0, 2, 4, 6, 12]	16	11111111001010010010001000110100
[0, 2, 4, 10, 12]	16	11111111001010010010001000110100
[0, 1, 2, 3, 5]	16	11111011100011010011001010000010
[0, 2, 4, 6, 10]	16	11111110100111000001000111000100
[0, 2, 6, 8, 18]	16	11111110010101100101001100000100
[0, 1, 2, 14, 15]	16	11111101011001001101100101000000
[0, 2, 4, 10, 16]	16	11111011101100001001110000101000
[0, 1, 2, 7, 8]	15	11111101100101001011000000100110
[0, 1, 4, 5, 9]	15	11111110010110000100011010100010
[0, 1, 4, 8, 12]	14	11111110101001110000010001010010
[0, 1, 4, 5, 8]	14	11111110010101100010110000100010
[0, 1, 4, 17, 20]	12	11111111000100011010100101001000
[0, 1, 4, 17, 18]	12	11111110010101100011010001001000
[0, 1, 2, 4, 5]	12	11111100110101100000011001010100
[0, 1, 4, 5, 17]	11	11111110101010010010011100001000
[0, 1, 2, 6, 12]	10	11110111011010001001011100001000
[0, 1, 4, 13, 29]	10	11111101110001011010100010010000
[0, 1, 2, 4, 19]	10	11111100110100110010101011000000
[0, 1, 2, 15, 16]	10	11111100000101100110001001011010

[0, 1, 2, 4, 17]	10	11111100110100101010110000001100
[0, 1, 2, 4, 6]	10	11111010010100011000001011100110
[0, 1, 2, 4, 29]	10	11111100110101100000011001010100
[0, 1, 3, 4, 17]	9	11111011001001110001101010000010
[0, 1, 2, 9, 10]	9	11111101010011000100000101100110
[0, 1, 2, 10, 22]	9	11111100001010001000110110100110
[0, 1, 2, 4, 8]	9	11111100010110010000110001010110
[0, 1, 4, 5, 16]	8	11111001100100100101001100111000
[0, 1, 4, 16, 20]	8	11111101011000100001101000110010
[0, 1, 2, 4, 28]	8	11111100101100010001010110000110
[0, 1, 2, 8, 26]	8	11111101100101010110010000001100
[0, 1, 4, 16, 17]	8	11111101001010010011001110001000
[0, 1, 4, 13, 17]	8	11111101110001011010100010010000
[0, 1, 2, 13, 14]	7	11111100001011001101010001100010
[0, 1, 2, 5, 6]	7	11111100110101101100001010001000
[0, 1, 2, 6, 20]	7	11111011000100001110100100010110
[0, 1, 2, 11, 12]	7	11111101000110010100000110100110
[0, 1, 2, 12, 14]	7	11111000010001001011101101000110
[0, 1, 2, 14, 16]	7	11111011011010001000111001010000
[0, 1, 2, 3, 11]	7	11111011100101011010000100110000
[0, 1, 3, 4, 14]	7	11111100010011101010000100101100
[0, 1, 4, 15, 18]	7	11111100100110101110010100001000
[0, 1, 2, 3, 9]	6	11111010010100001011000011001110
[0, 1, 2, 4, 22]	6	11111001000101101110101100010000
[0, 1, 2, 4, 14]	6	11110111011010001001011100001000
[0, 1, 2, 4, 20]	6	11111011100100010101100000110100
[0, 1, 3, 17, 20]	6	11111100100010000110011101001010
[0, 1, 2, 7, 27]	6	11111100110000101100001101010100
[0, 1, 3, 5, 7]	6	11111101100010100011001011000010
[0, 1, 3, 5, 28]	5	11111101100101000001100101011000
[0, 1, 2, 3, 14]	5	11111011101000011001010010110000
[0, 1, 2, 3, 16]	5	11111011011000001010011100010100
[0, 1, 2, 3, 10]	5	11111010111010110001001100100000
[0, 1, 4, 5, 18]	5	11111100110001010110010110000100
[0, 1, 2, 6, 8]	5	11111101101010001100001001101000
[0, 1, 3, 4, 10]	5	11111100111010110000100010010100
[0, 1, 2, 11, 23]	4	11111011100001000110101100010100
[0, 1, 2, 3, 15]	4	11111011000001110101001011000100
[0, 1, 3, 7, 10]	4	11111011100101000110101100000100
[0, 1, 2, 3, 8]	4	11111011010100001110010010110000

[0, 1, 4, 14, 18]	4	11111011010001001011001100011000
[0, 1, 2, 10, 12]	4	11111101100101101100010001010000
[0, 1, 2, 8, 20]	4	11111011100101101011001000100000
[0, 1, 4, 9, 28]	4	11111010110011011000110000010100
[0, 1, 2, 10, 23]	4	11111011100100001100001001010110
[0, 1, 2, 9, 24]	4	11111101001100001010110000110100
[0, 1, 5, 16, 19]	4	11111001100100011001101001011000
[0, 1, 4, 6, 8]	4	11111101101010001110001001001000
[0, 1, 2, 4, 11]	4	11111100110101010010110011000000
[0, 1, 3, 14, 19]	3	11111100001000101001100111010010
[0, 1, 3, 12, 30]	3	11110111010010001001011110001000
[0, 1, 2, 14, 18]	3	11110111011100010001011010010000
[0, 1, 2, 12, 21]	3	11111011011101001000011010000100
[0, 1, 3, 10, 26]	3	11111011010111000101100010000100
[0, 1, 3, 8, 11]	3	11111100101100001101010000110010
[0, 1, 3, 5, 8]	3	11111100110101100000101000110010
[0, 1, 3, 13, 23]	3	11111110001001010110010100110000
[0, 1, 3, 5, 23]	3	11111011001010011101011000100000
[0, 1, 3, 14, 30]	3	11111101001001010110000111000100
[0, 1, 2, 3, 13]	3	11111011100001010010100001101100
[0, 1, 4, 12, 29]	3	11111001011110110001000010100010
[0, 1, 2, 4, 16]	2	11110110100101110111000100010000
[0, 1, 3, 8, 9]	2	11111001011000011010110101100000
[0, 1, 2, 6, 10]	2	11110101111000100010100011011000
[0, 1, 3, 7, 8]	2	11111100010110010000110001010110
[0, 1, 3, 19, 21]	2	11111100010100011110010010100100
[0, 1, 2, 3, 6]	2	11111010101100001110010000100110
[0, 1, 2, 7, 26]	2	11111101100100000011010010100110
[0, 1, 2, 7, 10]	2	11111010100011001000100001110110
[0, 1, 3, 5, 16]	2	11111110001011000110010100100010
[0, 1, 4, 8, 13]	2	11111110010101100001000101100010
[0, 1, 3, 15, 19]	2	11110110100101110111000100010000
[0, 1, 3, 14, 16]	2	11111101001001010010110001110000
[0, 1, 3, 8, 25]	2	11111100101100001101010000110010
[0, 1, 2, 6, 18]	2	11111010101011000100111000011000
[0, 1, 3, 5, 14]	2	11110111010010001001011110001000
[0, 1, 3, 24, 25]	2	11111100101100001101010000110010
[0, 1, 2, 6, 27]	2	11111101010011000011010000100110
[0, 1, 2, 4, 13]	2	11111011100101100000110101000100
[0, 1, 3, 4, 8]	2	11111101011001001110001010000100

[0, 1, 2, 14, 19]	2	111110111010000110010100101100000
[0, 1, 2, 8, 10]	2	111101110001011010010001011100000
[0, 1, 3, 12, 14]	2	111101110100100010010111100010000
[0, 1, 3, 14, 17]	2	111111001010011010110001011000000
[0, 1, 2, 9, 11]	2	111111011011010001100010001010000
[0, 1, 3, 13, 16]	2	111111000001010100101100110001100
[0, 1, 2, 7, 25]	2	111111011000010110000110010010100
[0, 1, 4, 8, 21]	2	111111101010011100000100010100100
[0, 1, 2, 5, 27]	2	111111010000110100011000101001100
[0, 1, 4, 9, 29]	2	111110110110001110000100100010100
[0, 1, 2, 5, 17]	2	111111010011000110110001010000100
[0, 1, 2, 5, 29]	2	111110111000011001001000011010100
[0, 1, 3, 21, 23]	2	111101110100100010010111100010000
[0, 1, 2, 4, 21]	2	111101110000101101110100100010000
[0, 1, 4, 12, 15]	2	111110011100100010011011010100000
[0, 1, 4, 6, 26]	2	111110110001010001101011000001100
[0, 1, 2, 12, 18]	2	111110101000110011100000110100100
[0, 1, 2, 10, 13]	2	111111000001010010101101100110000
[0, 1, 3, 12, 15]	2	111111011100001010011001010001000
[0, 1, 2, 13, 15]	2	111110110100001011100011000101000
[0, 1, 3, 9, 10]	1	111110011101101000001100100010100
[0, 1, 3, 7, 26]	1	111110011101101010001011001000000
[0, 1, 2, 6, 24]	1	111101011110001101100010100010000
[0, 1, 3, 21, 27]	1	111111001010100000110010110001100
[0, 1, 3, 8, 22]	1	111110100101100010001011000011100
[0, 1, 3, 14, 22]	1	111101100010110100010000111100100
[0, 1, 2, 7, 13]	1	111111001101101011000101000100000
[0, 1, 2, 8, 15]	1	111101111010001011000101100001000
[0, 1, 2, 7, 21]	1	111110101001100010110001110000100
[0, 1, 2, 11, 18]	1	111111001101100000101011010010000
[0, 1, 3, 25, 28]	1	111111001100010110100000110010100
[0, 1, 3, 7, 22]	1	111101110000111010001000100101100
[0, 1, 4, 5, 12]	1	111110001010001000001110110101100
[0, 1, 2, 5, 11]	1	111111010000110100010100110001100
[0, 1, 2, 6, 22]	1	111110101010110001001110000110000
[0, 1, 3, 13, 20]	1	111111011001000101000101110010000
[0, 1, 2, 4, 15]	1	111110010110001001110101000001100
[0, 1, 3, 10, 25]	1	111111010010100000111001010011000
[0, 1, 4, 8, 18]	1	111101100110001011010011001100000
[0, 1, 3, 19, 23]	1	111110111010011100001000101001000

[0, 1, 2, 5, 23]	1	11111101000011010001010011000110
[0, 1, 2, 8, 14]	1	11111000011010101010000111001100
[0, 1, 2, 8, 22]	1	11111001101011000010111000010100
[0, 1, 2, 5, 12]	1	11111101010001100110000011010010
[0, 1, 2, 12, 16]	1	11111000110000111001000110101010
[0, 1, 2, 7, 9]	1	11111101001010011000000110100110
[0, 1, 4, 13, 23]	1	11110110011000010110011010011000
[0, 1, 2, 4, 10]	1	11111010110110010001110001000010
[0, 1, 2, 8, 11]	1	11111001101011010111000100100000
[0, 1, 3, 9, 19]	1	11111100100110100010111000010100
[0, 1, 2, 9, 12]	1	11111100000110101101010001100100
[0, 1, 3, 13, 17]	1	11110110100100010001011101110000
[0, 1, 4, 8, 9]	1	11111110001101101010100001001000
[0, 1, 4, 16, 19]	1	11111011001010000010101111000100
[0, 1, 2, 5, 10]	1	11111011100011001010011010000010
[0, 1, 3, 9, 11]	1	11111100100101101110001000101000
[0, 1, 2, 7, 16]	1	11110110001000101001110101110000
[0, 1, 5, 8, 13]	1	11111010001100010110000100101110
[0, 1, 3, 4, 13]	1	11111011101011000001100010100100
[0, 1, 2, 7, 18]	1	11111101001100001011001000010110
[0, 1, 3, 8, 28]	1	11111100100001010110001101100010
[0, 1, 2, 3, 7]	1	11111001000010100001101110101100
[0, 1, 2, 12, 20]	1	11111001011101100001101001010000
[0, 1, 3, 7, 28]	1	11111100100010100101000100111100
[0, 1, 3, 12, 17]	1	11111010001000010010111001110100
[0, 1, 2, 7, 17]	1	11111000010010000101101110101100
[0, 1, 2, 5, 28]	1	11111101000101001100001011000110
[0, 1, 3, 5, 19]	1	11111101001011100110010100010000
[0, 1, 4, 9, 12]	1	11111110001011001000110001010010
[0, 1, 2, 5, 18]	1	11111010101100001110010000100110

Below and on the next page are listed all combs of length 5 which have no corresponding sequence for the de Bruijn comb.

$[0, 1, 3, 13, 28]$	$[0, 1, 3, 17, 21]$	$[0, 1, 3, 7, 19]$	$[0, 1, 3, 13, 22]$	$[0, 1, 3, 4, 12]$
$[0, 1, 2, 9, 15]$	$[0, 4, 8, 16, 24]$	$[0, 1, 4, 17, 26]$	$[0, 1, 4, 6, 14]$	$[0, 1, 2, 6, 19]$
$[0, 2, 6, 18, 22]$	$[0, 1, 2, 8, 18]$	$[0, 1, 5, 7, 16]$	$[0, 1, 3, 15, 27]$	$[0, 1, 3, 23, 24]$
$[0, 1, 3, 8, 24]$	$[0, 2, 4, 10, 14]$	$[0, 1, 2, 7, 15]$	$[0, 1, 3, 7, 27]$	$[0, 1, 2, 5, 19]$
$[0, 1, 2, 6, 17]$	$[0, 1, 2, 4, 27]$	$[0, 1, 2, 10, 20]$	$[0, 1, 3, 5, 26]$	$[0, 1, 4, 6, 18]$
$[0, 1, 2, 6, 26]$	$[0, 1, 3, 5, 13]$	$[0, 1, 3, 12, 28]$	$[0, 1, 3, 23, 27]$	$[0, 2, 8, 16, 18]$

$[0, 1, 2, 7, 14]$	$[0, 1, 3, 7, 16]$	$[0, 1, 2, 4, 26]$	$[0, 1, 4, 14, 17]$	$[0, 1, 4, 18, 26]$
$[0, 1, 3, 9, 13]$	$[0, 1, 7, 15, 16]$	$[0, 1, 3, 8, 14]$	$[0, 1, 3, 7, 15]$	$[0, 1, 3, 10, 12]$
$[0, 1, 3, 17, 23]$	$[0, 1, 4, 9, 15]$	$[0, 1, 2, 9, 17]$	$[0, 1, 4, 9, 17]$	$[0, 1, 3, 13, 27]$
$[0, 1, 3, 8, 12]$	$[0, 1, 2, 9, 20]$	$[0, 1, 2, 9, 19]$	$[0, 1, 2, 5, 26]$	$[0, 1, 3, 9, 26]$
$[0, 1, 2, 7, 11]$	$[0, 1, 2, 9, 21]$	$[0, 1, 2, 4, 9]$	$[0, 1, 3, 16, 27]$	$[0, 1, 4, 6, 20]$
$[0, 1, 3, 4, 9]$	$[0, 1, 2, 5, 9]$	$[0, 1, 3, 5, 9]$	$[0, 1, 2, 6, 21]$	$[0, 1, 3, 17, 19]$
$[0, 1, 4, 15, 28]$	$[0, 1, 2, 9, 16]$	$[0, 1, 3, 5, 15]$	$[0, 1, 3, 7, 24]$	$[0, 1, 2, 10, 15]$
$[0, 1, 3, 8, 16]$	$[0, 1, 3, 10, 23]$	$[0, 1, 2, 4, 23]$	$[0, 1, 2, 7, 22]$	$[0, 1, 3, 13, 21]$
$[0, 1, 2, 6, 25]$	$[0, 1, 3, 8, 10]$	$[0, 2, 4, 10, 24]$	$[0, 1, 4, 12, 14]$	$[0, 1, 2, 6, 9]$
$[0, 1, 3, 21, 25]$	$[0, 1, 2, 11, 15]$	$[0, 1, 2, 10, 16]$	$[0, 1, 4, 15, 20]$	$[0, 1, 2, 3, 12]$
$[0, 1, 2, 5, 13]$	$[0, 1, 3, 19, 27]$	$[0, 1, 3, 8, 17]$	$[0, 1, 4, 13, 14]$	$[0, 1, 2, 13, 20]$
$[0, 1, 3, 15, 16]$	$[0, 1, 3, 5, 11]$	$[0, 1, 3, 22, 25]$	$[0, 1, 2, 12, 15]$	$[0, 1, 2, 10, 17]$
$[0, 1, 2, 6, 14]$	$[0, 1, 2, 8, 23]$	$[0, 1, 3, 9, 25]$	$[0, 1, 3, 12, 27]$	$[0, 1, 7, 8, 17]$
$[0, 1, 3, 10, 15]$	$[0, 1, 3, 7, 11]$	$[0, 1, 3, 15, 28]$	$[0, 1, 2, 16, 18]$	$[0, 1, 3, 10, 13]$
$[0, 1, 2, 7, 23]$	$[0, 1, 3, 5, 12]$	$[0, 1, 2, 6, 23]$	$[0, 1, 2, 4, 7]$	$[0, 1, 7, 9, 15]$
$[0, 1, 3, 15, 21]$	$[0, 1, 2, 11, 16]$	$[0, 2, 4, 8, 14]$	$[0, 1, 2, 13, 18]$	$[0, 1, 3, 7, 9]$
$[0, 1, 3, 23, 25]$	$[0, 1, 3, 20, 24]$	$[0, 1, 2, 6, 11]$	$[0, 1, 2, 10, 18]$	$[0, 1, 2, 6, 15]$
$[0, 1, 4, 14, 28]$	$[0, 1, 2, 8, 24]$	$[0, 1, 2, 9, 13]$	$[0, 1, 3, 15, 22]$	$[0, 1, 2, 4, 24]$
$[0, 1, 3, 4, 11]$	$[0, 1, 2, 5, 8]$	$[0, 1, 3, 5, 17]$	$[0, 1, 2, 8, 16]$	$[0, 1, 3, 5, 25]$
$[0, 2, 4, 10, 18]$	$[0, 1, 8, 16, 17]$	$[0, 1, 3, 13, 25]$	$[0, 1, 3, 16, 25]$	$[0, 1, 3, 15, 23]$
$[0, 2, 8, 16, 24]$	$[0, 1, 4, 6, 12]$	$[0, 1, 2, 8, 19]$	$[0, 1, 4, 15, 16]$	$[0, 1, 4, 14, 29]$
$[0, 1, 3, 4, 16]$	$[0, 1, 2, 5, 7]$	$[0, 1, 3, 13, 15]$	$[0, 1, 3, 7, 23]$	$[0, 1, 3, 9, 28]$
$[0, 1, 6, 8, 17]$	$[0, 1, 8, 16, 24]$	$[0, 1, 2, 8, 17]$	$[0, 1, 3, 7, 17]$	$[0, 1, 3, 12, 13]$
$[0, 1, 3, 24, 28]$	$[0, 2, 4, 16, 20]$	$[0, 1, 4, 6, 17]$	$[0, 1, 3, 9, 30]$	$[0, 1, 3, 27, 28]$
$[0, 1, 2, 7, 19]$	$[0, 1, 2, 5, 14]$	$[0, 2, 4, 10, 26]$	$[0, 1, 2, 9, 25]$	$[0, 1, 3, 8, 20]$
$[0, 1, 3, 15, 20]$	$[0, 1, 2, 10, 14]$	$[0, 1, 3, 8, 19]$	$[0, 1, 3, 22, 24]$	$[0, 1, 4, 12, 18]$
$[0, 1, 2, 14, 17]$	$[0, 1, 3, 16, 21]$	$[0, 1, 5, 8, 16]$	$[0, 1, 3, 12, 16]$	$[0, 1, 3, 5, 24]$
$[0, 1, 2, 12, 19]$	$[0, 1, 2, 15, 18]$	$[0, 1, 2, 6, 13]$	$[0, 1, 3, 9, 20]$	$[0, 1, 2, 9, 14]$
$[0, 1, 2, 11, 13]$	$[0, 1, 3, 5, 21]$	$[0, 1, 2, 5, 25]$	$[0, 1, 3, 21, 24]$	$[0, 1, 2, 5, 24]$
$[0, 1, 4, 5, 13]$	$[0, 1, 2, 11, 14]$	$[0, 1, 2, 5, 15]$	$[0, 1, 3, 9, 16]$	$[0, 1, 3, 17, 27]$
$[0, 1, 7, 9, 16]$	$[0, 1, 2, 5, 20]$	$[0, 1, 3, 25, 27]$	$[0, 1, 3, 5, 22]$	$[0, 1, 3, 10, 30]$
$[0, 1, 2, 12, 17]$	$[0, 1, 2, 5, 16]$	$[0, 1, 4, 8, 23]$	$[0, 1, 2, 9, 22]$	$[0, 1, 3, 16, 19]$
$[0, 1, 3, 10, 20]$	$[0, 1, 3, 14, 15]$	$[0, 1, 3, 16, 17]$	$[0, 1, 2, 5, 21]$	$[0, 1, 3, 8, 23]$
$[0, 1, 3, 17, 25]$	$[0, 1, 2, 8, 12]$	$[0, 1, 4, 15, 23]$	$[0, 1, 2, 4, 12]$	$[0, 1, 3, 12, 24]$
$[0, 1, 3, 10, 16]$	$[0, 1, 2, 11, 19]$	$[0, 1, 2, 6, 16]$	$[0, 1, 3, 16, 23]$	$[0, 1, 4, 26, 28]$
$[0, 1, 3, 15, 25]$	$[0, 1, 3, 7, 12]$	$[0, 1, 2, 13, 16]$	$[0, 1, 3, 4, 15]$	$[0, 1, 3, 17, 28]$
$[0, 1, 3, 9, 17]$	$[0, 1, 3, 7, 30]$	$[0, 1, 3, 10, 14]$	$[0, 1, 7, 9, 17]$	$[0, 1, 4, 9, 16]$
$[0, 1, 2, 5, 22]$	$[0, 1, 2, 7, 24]$	$[0, 1, 3, 8, 21]$	$[0, 1, 2, 8, 13]$	$[0, 1, 2, 8, 25]$
$[0, 1, 4, 16, 26]$	$[0, 1, 4, 9, 20]$	$[0, 1, 2, 4, 25]$	$[0, 1, 3, 12, 25]$	