Large Order Binary de Bruijn Sequences via Zech's Logarithms

Zuling Chang, Martianus Frederic Ezerman, Adamas Aqsa Fahreza, San Ling, and Huaxiong Wang

***Abstract*—This work shows how to efficiently construct binary de Bruijn sequences, even those with large orders, using the cycle joining method. The cycles are generated by an LFSR with a chosen period e whose irreducible characteristic polynomial can be derived from any primitive polynomial of degree n satisfying $e = \frac{2^n-1}{t}$ by t-decimation. The crux is our proof that determining Zech's logarithms is equivalent to identifying conjugate pairs shared by any pair of cycles. The approach quickly finds enough number of conjugate pairs between any two cycles to ensure the existence of trees containing all vertices in the adjacency graph of the LFSR.**

When the characteristic polynomial $f(x)$ is a product of distinct irreducible polynomials, we combine the approach via Zech's logarithms and a recently proposed method to determine the conjugate pairs. This allows us to efficiently generate de Bruijn sequences with larger orders. Along the way, we establish new properties of Zech's logarithms.

***Index Terms*—Binary de Bruijn sequence, conjugate pair, decimation, Zech's logarithm.**

I. Introduction

A binary *de Bruijn sequence* of order $n \in \mathbb{N}$ has period $N = 2^n$ in which each n-tuple occurs exactly once. There are $2^{2^{n-1}-n}$ such sequences [1].

We developed a de Bruijn sequence generator in [2] and demonstrated that it performed well up to some decent design parameters. The characteristic polynomials were mostly products of distinct irreducible polynomials. As the period grows large, *i.e.*, as the degree n of the characteristic polynomial increases, we run into time complexity issues.

Motivated mostly by cryptographic purposes, there has been a sustained interest in efficiently generating a good number of de Bruijn sequences of large order.

Adding a 0 to the longest string of 0s in a *maximal length sequence* (also known as an m-sequence) of period $2^n - 1$ produces a de Bruijn sequence of order n. There are $\Lambda_n := \frac{1}{n}\phi(2^n-1)$ such shift-distinct m-sequences where $\phi(.)$ is the Euler totient function. There is a bijection between the set of all such sequences and the set of primitive polynomials of degree n in $\mathbb{F}_2[x]$. As n grows large, Λ_n soon becomes miniscule compared to $2^{2^{n-1}-n}$.

It is then natural to widen the choice of the characteristic polynomials from primitive polynomials to irreducible polynomials. Let $\mu(n)$ be the Möbius function. There are $\displaystyle\frac{1}{n}\sum_{d|n}\mu(d)2^{\frac{n}{d}}$ irreducible polynomials of degree n over $\mathbb{F}_2$. All irreducible polynomials of (prime) degree n are primitive if and only if $2^n - 1$ is prime. Such an n is called a *Mersenne exponent*. Mersenne exponents are sparse [3] although not known to be finite. Thus, for most n, there are many more irreducible than primitive polynomials.

Each irreducible non-primitive polynomial may yield a large number of de Bruijn sequences if one can efficiently identify the conjugate pairs to apply the cycle joining method on them. A recent contribution using irreducible characteristic polynomials of large degree n is [4]. It shows how to generate de Bruijn sequences of large orders *e.g.*, 128 without either time or space complexity analysis.

In this paper we focus on producing large order de Bruijn sequences and aim to vastly improve on the existing results. To be more specific, we show how to handle much larger orders than had been attempted and generate many of them faster.

At the core of our method is a new insight on how Zech's logarithms characterize the conjugate pairs.

We start by identifying a primitive polynomial $p(x) \in \mathbb{F}_2[x]$ of degree n having a (primitive) root α. Several approaches to do so are given in [5, Section 4.4]. Many computer algebra systems have routines that output primitive polynomial(s) of a specified degree.

Combining a decimation technique and the Berlekamp-Massey algorithm on input $p(x)$ and a suitable divisor t of $2^n - 1$ we derive the associated irreducible polynomial $f(x)$. It has degree n, order e, and a root $\beta = \alpha^t$ with $e \cdot t = 2^n - 1$. The cycle structure of the LFSR with $f(x)$ as a characteristic polynomial has t distinct nonzero cycles.

We establish a novel approach that transforms the problem of finding conjugate pairs between any two distinct cycles into computing Zech's logarithms with respect to α. A property, useful to compute new Zech's logarithms from the already known values, is proved. Taking modulo t of the appropriate Zech's logarithms characterizes the exact positions of the corresponding conjugate pairs. This eliminates the need to store the pairs, reducing the memory requirement tremendously. These positional markings pave the way to speedy generation of the resulting sequences.

Z. Chang's work is supported by the Joint Fund of the National Natural Science Foundation of China Grant U1304604. Research Grants TL-9014101684-01 and MOE2013-T2-1-041 support the research carried out by M. F. Ezerman, S. Ling, and H. Wang. Singapore Ministry of Education Grant M4011381 provides a partial support for A. A. Fahreza.

Z. Chang is with the School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China, e-mail: zuling_chang@zzu.edu.cn.

M. F. Ezerman, A. A. Fahreza, S. Ling, and H. Wang are with the School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, Singapore 637371, e-mails: {fredezerman,adamas,lingsan,HXWang}@ntu.edu.sg.

Trading completeness for efficiency, we compute for enough number of conjugate pairs between any two distinct cycles to create a connected subgraph $\widetilde{\mathcal{G}} = (V_{\widetilde{\mathcal{G}}}, E_{\widetilde{\mathcal{G}}})$ of the full adjacency graph $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$ associated with $f(x)$, requiring that $V_{\widetilde{\mathcal{G}}} = V_{\mathcal{G}}$. Any spanning tree Υ in $\widetilde{\mathcal{G}}$ can be used to complete the cycle joining method, yielding de Bruijn sequences. The method works for any order n and determining the number of constructible sequences is straightforward.

We briefly discuss the case when $f(x)$ is the product of distinct irreducible polynomials. The conjugate pairs in this case can similarly be determined by the Zech's logarithms relative to the roots of the primitive polynomials associated with the respective irreducible polynomials. We end with some conclusions.

II. Preliminaries

Let $\mathbb{N}$ denote the set of positive integers. For integers $k < \ell$, let $[\![k, \ell]\!]$ denote $\{k, k+1, \ldots, \ell-1, \ell\}$. We recall some definitions and results from [6, Chapter 4].

An *n-stage shift register* is a clock-regulated circuit with n consecutive storage units, each containing a bit. As the clock pulses, each bit is shifted to the next stage in line. A shift register generates a binary code if one adds a feedback loop that outputs a new bit s_n based on the n bits $\mathbf{s}_0 = (s_0, \ldots, s_{n-1})$ called an *initial state* of the register. The Boolean function h that outputs s_n on input $\mathbf{s}_0$ is called its *feedback function*.

A feedback shift register (FSR) outputs a binary sequence $\mathbf{s} = s_0, s_1, \ldots, s_n, \ldots$ satisfying the recursive relation $s_{n+\ell} = h(s_\ell, s_{\ell+1}, \ldots, s_{\ell+n-1})$ for $\ell \geq 0$. If $s_{i+N} = s_i$ for all $i \geq 0$, then $\mathbf{s}$ is *N-periodic* or *with period N* and one denotes $\mathbf{s} = (s_0, s_1, s_2, \ldots, s_{N-1})$. We call $\mathbf{s}_i = (s_i, s_{i+1}, \ldots, s_{i+n-1})$ *the i-th state* of $\mathbf{s}$ and $\mathbf{s}_{i+1}$ the *successor* of $\mathbf{s}_i$. The all zero sequence $\mathbf{0}$ has period 1. We also use $\mathbf{0}$ to denote a zero vector.

In an FSR, distinct initial states generate distinct sequences, forming the set $\Omega(h)$ of 2^n elements.

A *state operator* T turns $\mathbf{s}_i$ into $\mathbf{s}_{i+1}$, *i.e.*, $\mathbf{s}_{i+1} = T\mathbf{s}_i$. If $\mathbf{s}$ has a state $\mathbf{s}_i$ and period e, then the e *distinct* states of $\mathbf{s}$ are $\mathbf{s}_i, T\mathbf{s}_i = \mathbf{s}_{i+1}, \ldots, T^{e-1}\mathbf{s}_i = \mathbf{s}_{i+e-1}$. The *shift operator* L sends $\mathbf{s}$ to $L\mathbf{s} = L(s_0, s_1, \ldots, s_{N-1}) = (s_1, s_2, \ldots, s_{N-1}, s_0)$ with the convention that $L^0\mathbf{s} = \mathbf{s}$. The set $[\mathbf{s}] := \{\mathbf{s}, L\mathbf{s}, L^2\mathbf{s}, \ldots, L^{N-1}\mathbf{s}\}$ is a *shift equivalent class* or a *cycle* in $\Omega(h)$. One partitions the set of sequences in $\Omega(h)$ into cycles and writes the *cycle structure* as

$$[\mathbf{s}_1] \cup [\mathbf{s}_2] \cup \ldots \cup [\mathbf{s}_r] \text{ if } \bigcup_{i=1}^{r} [\mathbf{s}_i] = \Omega(h).$$

A *conjugate pair* consists of a state $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ and its *conjugate* $\widehat{\mathbf{v}} = (v_0 + 1, v_1, \ldots, v_{n-1})$. Cycles C_1 and C_2 in $\Omega(h)$ are *adjacent* if they are disjoint and there exists $\mathbf{v}$ in C_1 whose conjugate $\widehat{\mathbf{v}}$ is in C_2. Adjacent cycles merge into a single cycle by interchanging the successors of $\mathbf{v}$ and $\widehat{\mathbf{v}}$. The feedback function of the resulting cycle is

$$\widehat{h}(x_0, \ldots, x_{n-1}) = h(x_0, \ldots, x_{n-1}) + \prod_{i=1}^{n-1} (x_i + v_i + 1).$$

Continuing this step, all cycles in $\Omega(h)$ join into one cycle, which is a de Bruijn sequence. This is the *cycle joining method* [7]. The feedback functions of the resulting de Bruijn sequences are completely determined once the corresponding conjugate pairs are found.

Definition 1. *[8] The adjacency graph $\mathcal{G}$ of an FSR with feedback function h is an undirected multigraph whose vertices correspond to the cycles in $\Omega(h)$. There exists an edge between two vertices if and only if they are adjacent. A conjugate pair labels every edge. The number of edges between any pair of cycles is the number of conjugate pairs that they share.*

Clearly $\mathcal{G}$ contains no loops. There is a bijection between the spanning trees of $\mathcal{G}$ and the de Bruijn sequences constructed by the cycle joining method (see. *e.g.*, [8] and [9]). A variant of the BEST (de **B**ruijn, **E**hrenfest, **S**mith, and **T**utte) Theorem from [10, Section 7] provides the counting formula.

Theorem 1. *(BEST) Let $\mathcal{G}$ be the adjacency graph of an FSR with $V_{\mathcal{G}} := \{V_1, V_2, \ldots, V_k\}$. Let $\mathcal{M} = (m_{i,j})$ be the $k \times k$ matrix derived from $\mathcal{G}$ in which $m_{i,i}$ is the number of edges incident to vertex V_i and $m_{i,j}$ is the negative of the number of edges between vertices V_i and V_j for $i \neq j$. Then the number of the spanning trees of $\mathcal{G}$ is the cofactor of any entry of $\mathcal{M}$.*

The *cofactor* of entry $m_{i,j}$ in $\mathcal{M}$ is $(-1)^{i+j}$ times the determinant of the matrix obtained by deleting the i-th row and j-th column of $\mathcal{M}$.

An FSR is called *linear* or an LFSR if its feedback function is linear, and *nonlinear* or an NLFSR otherwise. Henceforth, all FSRs are linear.

The *characteristic polynomial* of an n-stage LFSR is

$$f(x) = x^n + c_{n-1}x^{n-1} + \ldots + c_1 x + c_0 \in \mathbb{F}_2[x] \qquad (1)$$

when the feedback function is $h(x_0, \ldots, x_{n-1}) = \sum_{i=0}^{n-1} c_i x_i$. To ensure that all generated sequences are periodic, $c_0 \neq 0$. A sequence $\mathbf{s}$ may have many characteristic polynomials with the one having the lowest degree being its *minimal polynomial*. It represents the LFSR of shortest length that generates $\mathbf{s}$. Given an LFSR with characteristic polynomial $f(x)$, the set $\Omega(h)$ is also denoted by $\Omega(f(x))$.

If the minimal polynomial of $\mathbf{s}$ is primitive with degree n, then $\mathbf{s}$ is the corresponding m-sequence with period $2^n - 1$, which is the maximal period among all sequences generated by any LFSR with minimal polynomial of degree n. Let $\mathbf{m}$ denote an m-sequence.

A sequence $\mathbf{u}$ is a *d-decimation* sequence of $\mathbf{s}$, denoted by $\mathbf{u} = \mathbf{s}^{(d)}$ if $u_j = s_{d \cdot j}$ for all $j \geq 0$. A d-decimation $\mathbf{m}^{(d)}$ of $\mathbf{m}$ is also an m-sequence if and only if $\gcd(d, 2^n - 1) = 1$.

More properties of sequences in relation to their characteristic and minimal polynomials can be found in [6, Chapter 4] and [11, Chapter 8].

To $f(x)$ in (1), one associates a matrix

$$A_f := \begin{pmatrix} 0 & 0 & \ldots & 0 & c_0 \\ 1 & 0 & \ldots & 0 & c_1 \\ 0 & 1 & \ldots & 0 & c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & c_{n-1} \end{pmatrix}. \qquad (2)$$

On input state $\mathbf{s}_0$, the state vectors of the resulting sequence are $\mathbf{s}_j = \mathbf{s}_0 A^j$ for $j \in \{0, 1, 2, \ldots\}$ and $\mathbf{s}_{i+1} = \mathbf{s}_i A = T\mathbf{s}_i$.

Let $f(x)$ be an irreducible but not a primitive polynomial with degree n. Suppose that $\beta \in \mathbb{F}_{2^n}$ is a root of $f(x)$, then there exists a primitive polynomial $p(x)$ with degree n and a root $\alpha \in \mathbb{F}_{2^n}$ satisfying $\beta = \alpha^t$ for some $1 < t \in \mathbb{N}$ and $e = \frac{2^n-1}{t}$ is the order of β. Notice that n is the least integer satisfying $2^n \equiv 1 \pmod{e}$.

Among computational devices over finite fields we have *Zech's logarithm* (see *e.g.*, [6, page 39]). The logarithm is also often referred to as *Jacobi's logarithm* [11, Exercise 2.8 and Table B]. It was Jacobi who introduced the notion and tabulated the values for $\mathbb{F}_p$ with $p \leq 103$ in [12]. For $\ell \in [\![1, 2^n - 1]\!] \cup \{-\infty\}$, the Zech's logarithm $\tau(\ell)$ relative to α is defined by $1 + \alpha^\ell = \alpha^{\tau(\ell)}$ where $\alpha^{-\infty} = 0$. It induces a permutation on $[\![1, 2^n - 2]\!]$.

A *cyclotomic coset* of 2 modulo $2^n - 1$ containing i is $D_i = \{i, 2i, \ldots, 2^{n_i - 1} i\}$ with n_i the least positive integer such that $i \equiv 2^{n_i} i \pmod{2^n - 1}$. Obviously $n_i \mid n$. For each i, call the least integer in D_i its coset leader. The set of all coset leaders form a *complete set of coset representatives*, denoted by $\mathcal{R}_n$.

The *cyclotomic classes* $C_i \subseteq \mathbb{F}_{2^n}$, for $0 \leq i < t$, are

$$C_i = \{\alpha^{i+s\cdot t} \mid 0 \leq s < e\} = \{\alpha^i \beta^s \mid 0 \leq s < e\} = \alpha^i C_0. \tag{3}$$

The *cyclotomic numbers* $(i, j)_t$, for $0 \leq i, j < t$, are

$$(i, j)_t = \left|\{\xi \mid \xi \in C_i, \xi + 1 \in C_j\}\right|. \tag{4}$$

III. The Cycle Structure

This section orders the cycles in $\Omega(f(x))$ using a method from [9] to benefit from some useful relationships between suitable sequences and cyclotomic classes.

Let $\{1, \beta, \ldots, \beta^{n-1}\}$ be a basis for the $\mathbb{F}_2$-vector space $\mathbb{F}_{2^n}$. Then $\alpha^j = \displaystyle\sum_{i=0}^{n-1} a_{j,i}\beta^i$ with $a_{j,i} \in \mathbb{F}_2$ for $j \in [\![0, 2^n - 2]\!]$. In vector form, the expression becomes

$$\alpha^j = (a_{j,0}, a_{j,1}, \ldots, a_{j,n-1}). \tag{5}$$

Define the mapping $\varphi : \mathbb{F}_{2^n} \to \mathbb{F}_2^n$ by

$$\varphi(0) = \mathbf{0}, \quad \varphi(\alpha^j) = (a_{j,0}, a_{j+t,0}, \ldots, a_{j+(n-1)t,0}), \tag{6}$$

where the subscripts are reduced modulo $2^n - 1$. By the recursive relation determined by (5), φ is a bijection. Let

$$\mathbf{u}_i := (a_{i,0}, a_{i+t,0}, \ldots, a_{i+(e-1)t,0}). \tag{7}$$

It is now straightforward to verify that

$$\Omega(f(x)) = [\mathbf{0}] \cup [\mathbf{u}_0] \cup [\mathbf{u}_1] \cup \ldots \cup [\mathbf{u}_{t-1}] \tag{8}$$

with $\varphi(\alpha^i)$ as the initial state of $\mathbf{u}_i$ for $i \in [\![0, t-1]\!]$. In particular, the initial state of $\mathbf{u}_0$ is $(1, \mathbf{0}) \in \mathbb{F}_2^n$.

Note that φ induces a correspondence between C_i and $[\mathbf{u}_i]$ (see [9, Thm. 3]). In other words, $\mathbf{u}_i$ and the sequence of states of $\mathbf{u}_i$, namely $((\mathbf{u}_i)_0, (\mathbf{u}_i)_1, \ldots, (\mathbf{u}_i)_{e-1})$, where

$$(\mathbf{u}_i)_j = (a_{i+jt,0}, a_{i+(j+1)t,0}, \ldots, a_{i+(j+n-1)t,0}) = \varphi(\alpha^i \beta^j)$$

for $j \in [\![0, e-1]\!]$, are equivalent. The state $\varphi(\alpha^i\beta^j)$ corresponds to the element $\alpha^i\beta^j \in C_i$. Hence, $\mathbf{u}_i \longleftrightarrow C_i$. This provides a convenient method to find the exact position of any state $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1}) \in \mathbb{F}_2^n$ in some cycle in $\Omega(f(x))$.

Let $\mathbf{v} = \varphi(\alpha^i\beta^j) = \varphi(\alpha^{i+tj})$ for some $i \in [\![0, t-1]\!]$. Then $\mathbf{v}$ must be the j-th state of $\mathbf{u}_i$. Let $k = i + tj$ and suppose that $\alpha^k = a_0 + a_1\beta + \ldots + a_{n-1}\beta^{n-1}$. We have $a_0 = v_0$ from the definition of $\varphi(\alpha^k)$. Note that if $f(x)$ in (1) is irreducible, then $c_0 = 1$. Since $\beta = \alpha^t$,

$$\alpha^{k+t} = \sum_{\ell=0}^{n-1} a_\ell \beta^{\ell+1} = \sum_{\ell=0}^{n-2} a_\ell \beta^{\ell+1} + \sum_{\ell=0}^{n-1} a_{n-1} c_\ell \beta^\ell =$$
$$a_{n-1} + (a_0 + a_{n-1}c_1)\beta + \ldots + (a_{n-2} + a_{n-1}c_{n-1})\beta^{n-1}.$$

Hence, one has

$$\begin{cases} a_0 &= v_0 \\ a_{n-1} &= v_1 \\ a_{n-2} &= v_1 c_{n-1} + v_2 \\ a_{n-3} &= v_1 c_{n-2} + v_2 c_{n-1} + v_3 \\ \vdots & \vdots \\ a_1 &= v_1 c_2 + \ldots + v_{n-2} c_{n-1} + v_{n-1} \end{cases}. \tag{9}$$

Once $a_0 + a_1\beta + \ldots + a_{n-1}\beta^{n-1}$ and the Zech's logarithms relative to α are known, one gets α^k and, thus, $\mathbf{v}$'s position.

How can one efficiently generate the cycles in $\Omega(f(x))$? Directly using the above definition is not practical since it requires costly computations over $\mathbb{F}_{2^n}$. Simply generating them by the LFSR with characteristic polynomial $f(x)$ may fail to guarantee that their respective initial states are $\varphi(\alpha^i)$ for $i \in [\![0, t-1]\!]$. We show that decimation is the right tool.

Equation (5) ensures that $\mathbf{m} = (a_{0,0}, a_{1,0}, \ldots, a_{2^n-2,0})$ is an m-sequence with characteristic polynomial $p(x)$ [6, Chapter 5]. The *trace function* Tr maps $\delta \in \mathbb{F}_{2^n}$ to $\sum_{i=0}^{n-1} \delta^{2^i} \in \mathbb{F}_2$. Recall, *e.g.*, from [6, Section 4.6] that the entries in $\mathbf{m}$ are $a_{i,0} = \mathrm{Tr}(\gamma\alpha^i) : 0 \neq \gamma \in \mathbb{F}_{2^n}$ for $i \in [\![0, 2^n - 2]\!]$. From $\mathbf{m}$, construct t distinct t-decimation sequences of period e:

$$\mathbf{u}_0 = \mathbf{m}^{(t)}, \mathbf{u}_1 = (L\mathbf{m})^{(t)}, \ldots, \mathbf{u}_{t-1} = (L^{t-1}\mathbf{m})^{(t)}.$$

The resulting sequences have $(\mathbf{u}_k)_j = \mathrm{Tr}(\gamma\alpha^{k+t\cdot j})$ for $k \in [\![0, t-1]\!]$ and $j \in [\![0, e-1]\!]$. Each $[\mathbf{u}_i]$ is a cycle in $\Omega(f(x))$ since $\beta = \alpha^t$. We need to find an initial state $\mathbf{v}$ of $\mathbf{m}$ such that the initial state of $\mathbf{u}_0$ is $(1, \mathbf{0}) \in \mathbb{F}_2^n$.

Let A_p be the associate matrix of $p(x)$. Then the respective first elements of $\mathbf{v}A_p^{(i\cdot t)}$ for $t \in [\![0, n-1]\!]$ must be $1, 0, \ldots, 0$. Construct a system of equations to derive $\mathbf{v}$. Let κ be the number of 1s in the binary representation of $i \cdot t$. Computing $A_p^{(i\cdot t)}$ is efficient using the square-and-multiply method, taking at most $\log_2 \lfloor i \cdot t \rfloor$ squarings and κ multiplications.

Use $\mathbf{v}$ and $p(x)$ to derive the first $n \cdot t$ entries of $\mathbf{m}$. The respective initial states $\varphi(\alpha^0), \ldots, \varphi(\alpha^{t-1})$ of $\mathbf{u}_0, \ldots, \mathbf{u}_{t-1}$ in $\Omega(f(x))$ immediately follow by decimation. Thus, one gets $\varphi(\alpha^j)$ for any j. This allows us to quickly find the desired initial state of any cycle, even for large n. Given the state $(\mathbf{u}_i)_j = \varphi(\alpha^i\beta^j)$, we have $T^k[(\mathbf{u}_i)_j] = \varphi(\alpha^i\beta^{j+k})$.

At this point, given an irreducible polynomial $f(x)$ with root β and order $e = \frac{2^n-1}{t}$, we need a primitive polynomial $p(x)$

with the same degree as $f(x)$ and root α satisfying $\beta = \alpha^t$. In general, such a $p(x)$ is not unique [2, Section 3]. Here we provide a method to find one.

For $k \in [\![1, \Lambda_n]\!]$, let $p_k(x)$ be a primitive polynomial of degree n that generates the m-sequence $\mathbf{m}_k$. The set of all shift inequivalent m-sequences with period $2^n - 1$ is $\{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{\Lambda_n}\}$. The elements are the d_j-decimation sequences of any $\mathbf{m}_k$ for all d_j satisfying $\gcd(d_j, 2^n - 1) = 1$. Derive $\mathbf{m}_k^{(t)}$ of period e and check if it shares a common string of $2n$ consecutive elements with a sequence whose characteristic polynomial is $f(x)$. If yes, then associate $p_k(x)$ with $f(x)$. Testing all ks guarantees a match between $f(x)$ and some $p_k(x)$ without costly operations over $\mathbb{F}_{2^n}$.

As n or t grows, finding one such $p(x)$ becomes more computationally involved. To work around this limitation, one starts instead with any primitive polynomial $p(x)$ with a root α and find the corresponding irreducible $f(x)$ having $\beta = \alpha^t$ as a root. There are tools from finite fields (see, *e.g.*, [11]) that can be deployed. We prefer another approach that does not require computations over $\mathbb{F}_{2^n}$.

Any primitive polynomial $p(x)$ generates an m-sequence $\mathbf{m}$. Taking t-decimation, we get $\mathbf{m}^{(t)}$. Input any $2n$ consecutive bits of $\mathbf{m}^{(t)}$ into the Berlekamp-Massey algorithm [13, Section 6.2.3] to get an irreducible polynomial $f(x)$ having α^t as a root. Note that there are instances where $f(x)$ has degree $m \mid n$ with $m < n$. This implies that, for this t, there is no irreducible polynomial of degree n that can be associated with $p(x)$.

As k traverses $\mathcal{R}_n$, by k-decimation and the Berlekamp-Massey algorithm, the process provides all irreducible polynomials with root $\alpha^{t \cdot k}$. The resulting polynomials form the set of all irreducible polynomials with degree $m \mid n$.

Remark 1. *Constructing irreducible polynomials is important in the study of finite fields. The Ben-Or and Rabin tests for irreducibility are widely used. Their implementation source codes are given in [5, Section 4.4]. The latter is often more efficient but requires a number of* $\gcd$ *computations for polynomials which increases the running cost in many other instances. A running time comparison is available in [14].*

Our procedure determines irreducible polynomials of degree m satisfying $m \mid n$ given any primitive polynomial of degree n using only decimation and the Berlekamp-Massey algorithm, which is much faster than both of the above tests. The running time is $\mathcal{O}(n^2)$

IV. Conjugate Pairs and Zech's Logarithms

This section proves that the conjugate pairs shared by any two distinct cycles in $\Omega(f(x))$ are characterized by Zech's logarithms. We further discover that determining the respective initial states of $[\mathbf{u}_i] : i \in [\![1, t-1]\!]$ is not necessary. Ensuring that $(1, \mathbf{0})$ is the initial state of $[\mathbf{u}_0]$ is sufficient for implementation.

Our main contribution is the following theorem.

Theorem 2. *Let α be a root of a primitive polynomial $p(x)$ of degree n and $\tau()$ be the Zech's logarithm with respect to α. Let $f(x)$ be the irreducible polynomial of degree n and order $e = \frac{2^n-1}{t}$ having a root $\beta = \alpha^t$,* i.e., *$f(x)$ is associated with $p(x)$.*

Let $[\mathbf{u}_i]$ and $[\mathbf{u}_\ell]$ be distinct nonzero cycles in $\Omega(f(x))$ constructed above, i.e., *$i, \ell \in [\![0, t-1]\!]$. Let $\mathbf{v} := T^j \varphi(\alpha^i) = \varphi(\alpha^{i+tj})$ be the j-th state of $[\mathbf{u}_i]$ and $\widehat{\mathbf{v}} := T^k \varphi(\alpha^\ell) = \varphi(\alpha^{\ell+tk})$ be the k-th state of $[\mathbf{u}_\ell]$. Then $(\mathbf{v}, \widehat{\mathbf{v}})$ forms a conjugate pair if and only if $\ell + tk = \tau(i + tj)$.*

Proof. Let η and γ be elements of $\mathbb{F}_{2^n}$. Then $\varphi(\eta)$ is a state of $\mathbf{u}_i$ if and only if $\eta = \alpha^j$ and $j \in C_i$. It is therefore clear that $\varphi(\eta) + \varphi(\gamma) = \varphi(\eta + \gamma)$. Observe that $\varphi(\alpha^0 = 1)$ and $\mathbf{0}$ are conjugate. The conjugate of $\varphi(\alpha^j)$ with $j \in [\![1, 2^n - 2]\!]$ is

$$\widehat{\varphi(\alpha^j)} = \varphi(1) + \varphi(\alpha^j) = \varphi(1 + \alpha^j) = \varphi(\alpha^{\tau(j)}).$$

The conjugate of an arbitrary state $\varphi(\alpha^j)$ belonging to cycle $[\mathbf{u}_{j \pmod t}]$ must then be $\varphi(\alpha^{\tau(j)})$, which belongs to cycle $[\mathbf{u}_{\tau(j) \pmod t}]$. In other words, the conjugate of the j-th state of cycle $[\mathbf{u}_i]$, which is $T^j \varphi(\alpha^i) = \varphi(\alpha^i \beta^j) = \varphi(\alpha^{i+tj})$, must be $\varphi(\alpha^{\tau(i+tj)})$. Writing $\tau(i+tj) = kt + \ell$ with $k \in [\![0, e-1]\!]$ and $\ell \in [\![0, t-1]\!]$, $\varphi(\alpha^{\tau(i+tj)}) = T^k \varphi(\alpha^\ell)$ belongs to $[\mathbf{u}_\ell]$.

Thus, knowing the Zech's logarithms relative to α enables us to easily determine all conjugate pairs between two arbitrary cycles in $\Omega(f(x))$. By the definition of cyclotomic numbers, $[\mathbf{u}_i]$ and $[\mathbf{u}_j]$ share $(i, j)_t$ conjugate pairs.

Conversely, knowing all of the conjugate pairs allows us to derive the Zech's logarithms relative to α. Let a conjugate pair $(\mathbf{v}, \widehat{\mathbf{v}})$ with $\mathbf{v} = T^j \varphi(\alpha^i) = \varphi(\alpha^{i+tj})$ and $\widehat{\mathbf{v}} = T^k \varphi(\alpha^\ell) = \varphi(\alpha^{\ell+tk})$ be given. Then $\tau(i + tj) = \ell + tk$ since $\widehat{\mathbf{v}}$ must be $\varphi(\alpha^{\tau(i+tj)})$. If all of the conjugate pairs are known, a complete Zech's logarithm table, relative to α, follows. □

Example 1. *Given $f(x) = x^4 + x^3 + x^2 + x + 1$, which is irreducible, of order 5 with β as a root, choose $p(x) = x^4 + x + 1$ with a root α satisfying $\alpha^3 = \beta$ as the associated primitive polynomial. Let $\mathbf{m}$ be the corresponding m-sequence with initial state* $(1, 0, 0, 0)$. *By 3-decimating one derives $\Omega(f(x)) = [\mathbf{0}] \cup [\mathbf{u}_0] \cup [\mathbf{u}_1] \cup [\mathbf{u}_2]$ with $\mathbf{u}_0 = (1, 0, 0, 0, 1)$, $\mathbf{u}_1 = (0, 1, 1, 1, 1)$, and $\mathbf{u}_2 = (0, 0, 1, 0, 1)$. The nonzero 4-stage states are:*

$$\begin{array}{c}
\varphi(\alpha^0) = \varphi(\alpha^0\beta^0) = (1,0,0,0) = (\mathbf{u}_0)_0 \\
\varphi(\alpha^1) = \varphi(\alpha^1\beta^0) = (0,1,1,1) = (\mathbf{u}_1)_0 \\
\varphi(\alpha^2) = \varphi(\alpha^2\beta^0) = (0,0,1,0) = (\mathbf{u}_2)_0 \\
\varphi(\alpha^3) = \varphi(\alpha^0\beta^1) = (0,0,0,1) = (\mathbf{u}_0)_1 \\
\varphi(\alpha^4) = \varphi(\alpha^1\beta^1) = (1,1,1,1) = (\mathbf{u}_1)_1 \\
\varphi(\alpha^5) = \varphi(\alpha^2\beta^1) = (0,1,0,1) = (\mathbf{u}_2)_1 \\
\varphi(\alpha^6) = \varphi(\alpha^0\beta^2) = (0,0,1,1) = (\mathbf{u}_0)_2 \\
\varphi(\alpha^7) = \varphi(\alpha^1\beta^2) = (1,1,1,0) = (\mathbf{u}_1)_2 \\
\varphi(\alpha^8) = \varphi(\alpha^2\beta^2) = (1,0,1,0) = (\mathbf{u}_2)_2 \\
\varphi(\alpha^9) = \varphi(\alpha^0\beta^3) = (0,1,1,0) = (\mathbf{u}_0)_3 \\
\varphi(\alpha^{10}) = \varphi(\alpha^1\beta^3) = (1,1,0,1) = (\mathbf{u}_1)_3 \\
\varphi(\alpha^{11}) = \varphi(\alpha^2\beta^3) = (0,1,0,0) = (\mathbf{u}_2)_3 \\
\varphi(\alpha^{12}) = \varphi(\alpha^0\beta^4) = (1,1,0,0) = (\mathbf{u}_0)_4 \\
\varphi(\alpha^{13}) = \varphi(\alpha^1\beta^4) = (1,0,1,1) = (\mathbf{u}_1)_4 \\
\varphi(\alpha^{14}) = \varphi(\alpha^2\beta^4) = (1,0,0,1) = (\mathbf{u}_2)_4
\end{array}.$$

The Zech's logarithm table is

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$\tau(i)$	4	8	14	1	10	13	9	2	7	5	12	11	6	3

.

All conjugate pairs between any two nonzero cycles can be determined from the table by Theorem 2. Knowing $\tau(3) = 14$, for example, one concludes that $[\mathbf{u}_0]$ and $[\mathbf{u}_2]$ share the pair $\varphi(\alpha^3) = (0, 0, 0, 1)$ and $\varphi(\alpha^{14}) = (1, 0, 0, 1)$. Conversely, knowing a conjugate pair is sufficient to determine the corresponding Zech's logarithm. Since $(0, 0, 1, 1) = \varphi(\alpha^6)$ and $(1, 0, 1, 1) = \varphi(\alpha^{13})$ form a conjugate pair between $[\mathbf{u}_0]$ and $[\mathbf{u}_1]$, for instance, one gets $\tau(6) = 13$.

Remark 2. *In Theorem 2, if $f(x)$ is primitive, then the output sequence is an m-sequence $\mathbf{m} = (m_0, m_1, \ldots, m_{2^n-2})$. Let $\mathbf{m}_0 := (m_0, \ldots, m_{n-1}) = (1, \mathbf{0})$. The i-th state $\mathbf{m}_i$ and the $\tau(i)$-th state $\mathbf{m}_{\tau(i)}$ form a conjugate pair. To compute $\tau(i)$ for $i \in [\![1, 2^n - 2]\!]$ it suffices to determine the position of the state $\mathbf{m}_{(\tau(i))} = \mathbf{m}_i + \mathbf{m}_0$ by searching. This fact can be used to find the Zech's logarithms when n is not very large.*

Huber established some important properties of Zech's logarithms over $\mathbb{F}_q$ and provided the Zech's logarithm tables for $\mathbb{F}_{2^n}$ with $2 \leq n \leq 11$ in [15]. We recall relevant results.

1) It suffices to find the logarithms relative to one primitive element. Let distinct primitive polynomials $p(x)$ and $q(x)$ be of degree n with respective roots α and δ. Then $\delta = \alpha^b$ for some integer b with $\gcd(b, 2^n - 1) = 1$. Let $\tau_{n,\alpha}(k)$ and $\tau_{n,\delta}(k)$ denote the respective logarithms of $k \in [\![1, 2^n - 2]\!]$ relative to α and δ. Note that
$$1 + \delta^k = 1 + \alpha^{b \cdot k} = \alpha^{\tau_{n,\alpha}(b \cdot k)} = \alpha^{bb^{-1}\tau_{n,\alpha}(b \cdot k)} = \delta^{b^{-1}\tau_{n,\alpha}(b \cdot k)}.$$
Hence, $\tau_{n,\delta}(k) \equiv b^{-1}\tau_{n,\alpha}(b \cdot k) \pmod{(2^n - 1)}$. With a primitive element fixed, we use the notation $\tau(k)$, or $\tau_n(k)$ to emphasize n.
2) The Flip map is given by $\tau_n(\tau_n(k)) = k$. Knowing $\tau_n(k)$ for any k in a cyclotomic coset D_j is sufficient to find $\tau_n(2^i k)$ by using the Double map
$$\tau_n(2k) \equiv 2\tau_n(k) \pmod{(2^n - 1)}. \quad (10)$$
Based on Flip and Double, the Zech's logarithm maps cosets onto cosets of the same size.
3) Let $\mathrm{Inv}(j) = 2^n - 1 - j \equiv -j \pmod{(2^n - 1)}$. Then
$$\tau_n(\mathrm{Inv}(j)) \equiv \tau_n(j) - j \pmod{(2^n - 1)}. \quad (11)$$
$\mathrm{Inv}(k)$ maps a coset onto a coset of the same size.
4) Let $h(x)$ be a primitive polynomial of degree m having a root β. Let $m \mid n$ and $\beta = \alpha^r$ with $r = {}^{(2^n - 1)}/_{(2^m - 1)}$. If the Zech's logarithms relative to β are known, then $1 + \alpha^{r \cdot j} = 1 + \beta^j = \beta^{\tau_m(j)} = \alpha^{r \cdot \tau_m(j)}$. Hence,
$$\tau_n(r \cdot j) \equiv r \cdot \tau_m(j) \pmod{(2^n - 1)}. \quad (12)$$

Repeatedly applying Flip and Inv induces a cycle of 6 cosets, except in 3 rare cases (see [15]). Using the Double map, one then gets the value of $\tau_n(k)$ for all k in the union of these cosets. To complete the table, Huber suggested the use of *key elements*, each corresponding to a starting coset. The following lemma reduces the storage need.

Lemma 1. *For known $\tau_n(i)$, $\tau_n(j)$, and $\tau_n(i - j)$,*
$$\tau_n(\tau_n(i) - \tau_n(j)) \equiv \tau_n(i - j) + j - \tau_n(j) \pmod{(2^n - 1)}. \quad (13)$$

TABLE I
ZECH'S LOGARITHMS FOR ELEMENTS IN REMAINING COSETS

(i, j)	$\tau(\tau(i) - \tau(j))$	Cosets in the induced cycle D_k	#
$(12, 5)$	$\tau(550) = 512$	$k \in \{77, 1, 511, 19, 251, 187\}$	60
$(12, 7)$	$\tau(43) = 523$	$k \in \{43, 23, 125, 63, 15, 245\}$	60
$(76, 28)$	$\tau(11) = 200$	$k \in \{11, 25, 223, 45, 189, 253\}$	60
$(3, 1)$	$\tau(956) = 78$	$k \in \{239, 39, 123, 439, 73, 49\}$	60
$(3, 2)$	$\tau(879) = 948$	$k \in \{447, 237, 75, 375, 69, 9\}$	60
$(12, 2)$	$\tau(909) = 874$	$k \in \{111, 347, 149, 35, 247, 57\}$	60
$(12, 10)$	$\tau(37) = 161$	$k \in \{37, 379, 31\}$	30
$(37, 31)$	$\tau(426) = 316$	$k \in \{213, 79, 59, 55, 121, 171\}$	60
$(37, 6)$	$\tau(141) = 744$	$k \in \{93, 105, 183\}$	30
$(77, 43)$	$\tau(501) = 142$	$k \in \{351, 71, 119, 235, 83, 21\}$	60
$(77, 34)$	$\tau(402) = 958$	$k \in \{147, 479, 17, 221, 89, 219\}$	60
$(68, 12)$	$\tau(181) = 971$	$k \in \{181, 191, 13, 157, 91, 173\}$	60
$(749, 255)$	$\tau(29) = 566$	$k \in \{29, 109, 151, 207, 51, 95\}$	60
$(702, 136)$	$\tau(343) = 746$	$k \in \{343, 85, 155\}$	30
$(434, 109)$	$\tau(27) = 206$	$k \in \{27, 103, 115, 205, 179, 159\}$	60
$(349, 333)$	$\tau(33) = 660$	$k \in \{33, 165, 363, 99, 231, 495\}$	30
$(785, 151)$	$\tau(87) = 619$	$k \in \{87, 215, 117, 367, 101, 41\}$	60
$(274, 51)$	$\tau(107) = 376$	$k \in \{107, 47, 175, 61, 167, 53\}$	60

Proof. For $i, j \in [\![1, 2^n - 2]\!]$, $1 + \alpha^i = \alpha^{\tau_n(i)}$ if and only if $\alpha^j + \alpha^{i+j} = \alpha^{\tau_n(i)+j}$. This is equivalent to $\alpha^{\tau_n(j)} + \alpha^{\tau_n(i+j)} = \alpha^{\tau_n(i)+j}$. Thus, $1 + \alpha^{\tau_n(i+j)-\tau_n(j)} = \alpha^{\tau_n(i)+j-\tau_n(j)}$. □

To apply Lemma 1, one looks for an (i, j) pair such that the respective Zech's logarithms of i, j, and $i - j$ are already known, *i.e.*, i, j, and $i - j$ are elements in the union U of cosets with known Zech's logarithms, but $\tau(i) - \tau(j) \notin U$. In many cases, a given Zech's logarithm is sufficient to deduce all others values.

Example 2. *We reproduce the table in [15, Appendix] for $p(x) = x^{10} + x^3 + 1$ without any key element. The computations are done modulo $2^{10} - 1$ with $=$ replacing $\equiv$ for brevity. There are 107 cyclotomic cosets of 2 modulo 1023: the trivial coset D_0, a coset of cardinality 2, 6 cosets, each of cardinality 5, and 99 cosets, each of cardinality 10.*

The coset $\{341, 682\}$ implies $\tau(341) = 682$.

The cycle of 6 cosets beginning from $\tau(3) = 10$ is
$$3 \in D_3 \underset{\longleftrightarrow}{\mathrm{Flip}} 10 \in D_5 \underset{\longleftrightarrow}{\mathrm{Inv}} 1013 \in D_{383} \underset{\longleftrightarrow}{\mathrm{Flip}} 1016 \in D_{127}$$
$$\underset{\longleftrightarrow}{\mathrm{Inv}} 7 \in D_7 \underset{\longleftrightarrow}{\mathrm{Flip}} 1020 \in D_{255} \underset{\longleftrightarrow}{\mathrm{Inv}} 3 \in D_3,$$
giving the logarithms of all 60 elements in $\bigcup D_k$ with $k \in \{3, 5, 7, 127, 255, 383\}$.

To generate the remaining logarithms, search for an (i, j) pair such that $\tau(i) - \tau(j)$ is not in any of previously known cosets but $i - j$ is. A simple `python` *routine completes the task. Table I summarizes the computations for the remaining cosets. The rows follow chronological order.*

Remark 3. *The approach in Example 2 may fail to yield the complete table. We tested all trinomials $x^n + x^j + 1$ with $n \in \{15, 17, 18, 20, 22\}$ and $j \leq \lfloor n/2 \rfloor$. The reciprocals give identical conclusions.*

For $n = 15$, the full table is obtained for $j \in \{1, 4, 7\}$. For $n = 17$, Lemma 1 produces the full table for $j \in \{3, 5\}$ while it fails for $j = 6$. It also fails for $n = 18, j = 7$ but works for $n = 20, j = 3$ and $n = 22, j = 1$. In case of failure, incorporating the computation in (12) becomes necessary.

V. Spanning Trees

This section takes a close look at the spanning trees in the adjacency graph $\mathcal{G}$.

A. Constructing Some Spanning Trees

When n is large or when t is a large valid divisor of $2^n - 1$, building the complete adjacency graph $\mathcal{G}$ is possible but consumes too much resources. It is also very often unnecessary to generate all of the de Bruijn sequences that the construction can produce. We aim to find enough Zech's logarithms to identify some spanning trees in $\mathcal{G}$. Since there is a unique pair that joins $[\mathbf{0}]$ and $[\mathbf{u}_0]$ into one cycle, the focus is on the set of nonzero cycles $\{[\mathbf{u}_i] : i \in [\![0, t-1]\!]\}$.

Let $j = a_1 \cdot t + a_2$. If $\tau_n(j) = b_1 \cdot t + b_2$ with $a_2, b_2 \in [\![0, t-1]\!]$, then $[\mathbf{u}_{a_2}]$ and $[\mathbf{u}_{b_2}]$ are adjacent. They are joined into one cycle using the conjugate pair

$$\mathbf{v} = \varphi(\alpha^j) = T^{a_1}\varphi(\alpha^{a_2}) \text{ and } \widehat{\mathbf{v}} = \varphi(\alpha^{\tau_n(j)}) = T^{b_1}\varphi(\alpha^{b_2}). \tag{14}$$

with $\varphi(\alpha^{a_2})$ the initial state of $\mathbf{u}_{a_2}$ and $\varphi(\alpha^{b_2})$ that of $\mathbf{u}_{b_2}$. Continue the process by identifying some conjugate pairs between enough pairs of adjacent cycles until all of the cycles in $\Omega(f(x))$ can be joined into one. The identified spanning tree(s) in $\mathcal{G}$ generates de Bruijn sequences. Using more Zechs logarithms yields more spanning trees, producing more such sequences.

Let $\tau(j)$ for some $j \in [\![1, 2^n - 2]\!]$ be known. This induces a mapping from D_j onto $D_{\tau(j)}$ with $n_j := |D_j| = |D_{\tau(j)}|$. If $j \not\equiv \tau(j) \pmod{t}$, then, for $i \in [\![0, n_j - 1]\!]$, states $\varphi(\alpha^{2^i j})$ and $\varphi(\alpha^{2^i \tau(j)})$ belong to distinct cycles. These states join their corresponding cycles into one.

Let m_j be the least positive integer such that $(2^{m_j} - 1)j \equiv 0 \pmod{t}$. Observe that $\varphi(\alpha^j)$ and $\varphi(\alpha^{j \cdot 2^{m_j}})$ are states of the same cycle and $m_j \mid n_j$. Hence, given cosets D_j and $D_{\tau(j)}$, one derives $\frac{n_j}{m_j}$ distinct conjugate pairs between each of the m_j distinct pairs of cycles.

The Zech's logarithms supply the exact positions of the conjugate pair(s) in the relevant cycles. Once enough conjugate pairs to construct a spanning tree are identified, the precise positions to apply the cycle joining method, *i.e.*, to exchange successors, appear. Thus, with $(1, \mathbf{0})$ as the initial state of $\mathbf{u}_0$, we just need to keep track of the precise positions, in terms of the operator T and the power of α, governed by the $(j, \tau_n(j))$ pair. The actual construction of the de Bruijn sequences no longer requires storing the initial states of the t nonzero sequences in $\Omega(f(x))$.

Example 3. *Consider $p(x) = x^{300} + x^7 + 1$ and let α be a root of $p(x)$, implying $\tau(7) = 300$. Choosing $t = 31$, the Berlekamp-Massey algorithm outputs*

$$\begin{aligned} f(x) = x^{300} + x^{194} + x^{176} + x^{158} + x^{97} + x^{88} + x^{79} + \\ x^{52} + x^{43} + x^{25} + x^{16} + x^7 + 1. \end{aligned}$$

Hence, $\Omega(f(x)) = [\mathbf{0}] \cup \bigcup_{i=0}^{30} [\mathbf{u}_i]$. Let $(1, \mathbf{0}) \in \mathbb{F}_2^{300}$ be the initial state of $\mathbf{u}_0$. For $i \in \{1, 3, 5, 7, 15, 35\}$, let $Z_i := \tau(i)$. Knowing a specific (i, Z_i) pair gives us $\{(j, Z_j) : j \in D_i\}$. Note that $|D_i| = 300$ for all i.

Since $Z_7 \equiv 21 \pmod{31}$, there are 5 distinct pairs of cycles, each sharing 60 conjugate pairs. The indices of the cycles are the (a_2, b_2) pairs $(7, 21), (14, 11), (28, 22), (25, 13), (19, 26)$. One of the 60 conjugate pairs between $[\mathbf{u}_7]$ and $[\mathbf{u}_{21}]$ is $(\varphi(\alpha^7), T^9 \varphi(\alpha^{21}))$ since $\left\lfloor \frac{7}{31} \right\rfloor = 0$ and $\left\lfloor \frac{300}{31} \right\rfloor = 9$. Computing a_1 and b_1 are easy given the relevant logarithms, so we omit them from the rest of this example.

Since $Z_1 \equiv Z_3 \equiv 0 \pmod{31}$, $[\mathbf{u}_0]$ shares 60 conjugate pairs each with $[\mathbf{u}_j]$ for $j \in \{1, 2, 4, 8, 16\} \cup \{3, 6, 12, 24, 17\}$. Similarly, adjacent cycles in Table II share 60 conjugate pairs.

TABLE II
The Rest of the Adjacent Cycles in Example 3

$(i, Z_i \pmod{31})$	Indices of Adjacent Cycles
$(5, 3)$	$(3, 5), (6, 10), (12, 20), (24, 9), (17, 18)$
$(15, 22)$	$(21, 27), (11, 23), (22, 15), (13, 30), (26, 29)$
$(35, 7)$	$(4, 7), (8, 14), (16, 28), (1, 25), (2, 19)$

We order the cycles as $[\mathbf{0}], [\mathbf{u}_0], [\mathbf{u}_1], \ldots, [\mathbf{u}_{30}]$ and build an adjacency subgraph $\widetilde{\mathcal{G}}$ from the computational results. Applying Theorem 1 with $\mathcal{G}$ replaced by $\widetilde{\mathcal{G}}$, the approach produces $\approx 2^{177.21}$ de Bruijn sequences. Figure 1 is a spanning tree. Using more Zech's logarithms leads to a larger number of de Bruijn sequences.

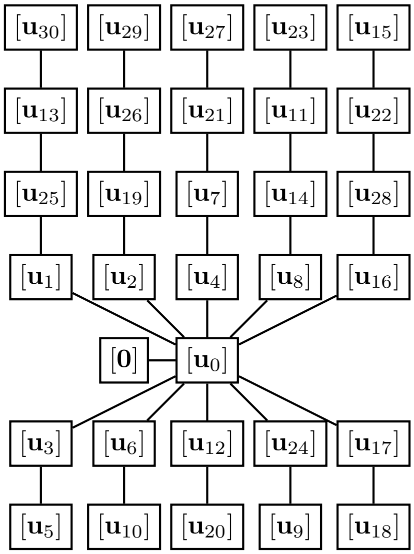
$[\mathbf{u}_{30}]$ $[\mathbf{u}_{29}]$ $[\mathbf{u}_{27}]$ $[\mathbf{u}_{23}]$ $[\mathbf{u}_{15}]$
$[\mathbf{u}_{13}]$ $[\mathbf{u}_{26}]$ $[\mathbf{u}_{21}]$ $[\mathbf{u}_{11}]$ $[\mathbf{u}_{22}]$
$[\mathbf{u}_{25}]$ $[\mathbf{u}_{19}]$ $[\mathbf{u}_{7}]$ $[\mathbf{u}_{14}]$ $[\mathbf{u}_{28}]$
$[\mathbf{u}_{1}]$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

Fig. 1. A spanning tree in an adjacency subgraph $\widetilde{\mathcal{G}}$ of $\Omega(f(x))$.

Our `python` implementation performs the following tasks as a proof of concept. The input consists of $p(x)$, a valid t, and a desired number of de Bruijn sequences.

1) Produce the associated irreducible polynomial $f(x)$.
2) Generate the (possibly partial) Zech's logarithm table.
3) Ensure that $(1, \mathbf{0})$ is the initial state of $[\mathbf{u}_0]$.
4) Attempt to build an adjacency subgraph from the table. If not all vertices are connected, then identify pairs of vertices that still need to be connected. Use the information to get needed Zech's logarithms through calls to MAGMA online calculator http://magma.maths.usyd.edu.au/calc/ until a connected adjacency subgraph containing all vertices is obtained.

5) Choose the required number of spanning trees and apply the cycle joining method to produce de Bruijn sequences using the procedure explained in [2, Section VI].

Example 4. *There are* $\approx 2^{145.73}$ *de Bruijn sequences that can be produced on input* $x^{10} + x^3 + 1$ *and* $t = 31$*. The program takes* 0.072 *seconds and a negligible amount of memory to output one of the sequences. On input* $x^{22} + x + 1$ *and* $t = 89$*, it consumes* 480.10 *seconds and around* 680 *MB of memory to produce one of* $\approx 2^{1286.65}$ *de Bruijn sequences.*

B. A Note on Dong and Pei's Construction

We examine a recent construction of de Bruijn sequences with large order proposed by Dong and Pei in [4]. Given an irreducible characteristic polynomial $f(x)$ of degree n, order e, and $t = \frac{2^n-1}{e}$, they defined a shift register matrix T in the form of (2) satisfying $f^*(T) = 0$ where $f^*(x)$ is the reciprocal polynomial of $f(x)$.

Given the sequence $\mathbf{u}_0$ with initial state $\alpha_0 = (1, \mathbf{0})$, write any sequence as $g(T)\mathbf{u}_0$, where $g(x)$ is some polynomial with degree less than n. If $(1 + x^k)^e \not\equiv 1 \pmod{f(x)}$, then $[\mathbf{u}_0]$ and $[(1 + T^k)^{2^j}\mathbf{u}_0]$ are distinct cycles sharing the conjugate pair $\left(T^{k2^j}\alpha_0, (1 + T^k)^{2^j}\alpha_0\right)$. Here $T^{k2^j}\alpha_0$ is a state of $[\mathbf{u}_0]$. The claim is that $[\mathbf{u}_0]$ shares some conjugate pairs with each of the other nonzero cycles. This does <u>not</u> hold in general.

First, as n and e grow large, it soon becomes prohibitive to compute $(1 + x^k)^e \pmod{f(x)}$.

Second, after $(1 + x^k)^e \not\equiv 1 \pmod{f(x)}$ is verified, it remains unclear which cycle $[(1 + T^k)^{2^j}\mathbf{u}_0]$ corresponds to. One is still unable to judge whether it is possible to join all of the cycles in $\Omega(f(x))$ even after a lot of the conjugate pairs have been determined.

Third, and most importantly, $t < \sqrt{2^n - 1}$ is a necessary condition for their method to work [4, Section 5]. In fact, a sufficient and necessary condition is $(0, i)_t > 0$ for all $i \in [\![1, t-1]\!]$. This does not hold in general. Take, *e.g.*, $n = 10$ with $p(x) = x^{10} + x^3 + 1$ and $t = 31 < \sqrt{31 \cdot 33}$. All values $1 \leq i \leq 2^{10} - 2$ such that $\tau(i) \equiv 0 \pmod t$ forms the set

$$\begin{aligned} X := \{&85, 105, 141, 170, 210, 277, 282, 291, 325, 337, 340, \\ &341, 379, 420, 431, 493, 554, 564, 582, 650, 657, 674, \\ &680, 682, 701, 727, 758, 840, 862, 875, 949, 986\}. \end{aligned}$$

Hence, $[\mathbf{u}_0]$ can be joined only to $[\mathbf{u}_\ell]$ with

$$\ell \in \{3, 6, 7, 12, 14, 15, 17, 19, 23, 24, 25, 27, 28, 29, 30\}.$$

Since only 15 out of the required 30 cycles can be joined with $[\mathbf{u}_0]$, Dong and Pei's approach fails to produce de Bruijn sequences here. We show in the next subsection that our method handles such a situation perfectly.

C. Star and Almost-Star Spanning Trees

In cases where $[\mathbf{u}_0]$ is adjacent to $[\mathbf{u}_j]$ for $j \in [\![1, t-1]\!]$, our characterization via Zech's logarithm rapidly certifies the existence of a star spanning tree centered at $[\mathbf{u}_0]$. This helps in quickly generating at least a de Bruijn sequence of order n. The certificate contains the following information.

1) A *witness* $\mathcal{W}$ that generate $\Delta_{\mathcal{W}} := \{i := k \cdot t \text{ for } k \in \mathcal{W}\}$ satisfying, with $Y_i := \tau(i) \pmod t$,
$$[\![1, t-1]\!] \subset \bigcup_{i \in \Delta_{\mathcal{W}}} \{j \pmod t : j \in D_{Y_i}\}. \qquad (15)$$
2) There are $\geq \#cp$ conjugate pairs between $[\mathbf{u}_0]$ and $[\mathbf{u}_j]$.
3) A matrix $\widetilde{M}$ derived from the adjacency subgraph $\widetilde{\mathcal{G}}$.
4) The number $\#dBSeqs$ of de Bruijn sequences.

Algorithm 1 outlines the required computations.

Algorithm 1 Certifying a Star Spanning Tree Exists

```
Input: $n$ and $p(x)$.
Output: Witness $\mathcal{W}$, $\#cp$, Matrix $\widetilde{M}$ and $\#dBSeqs$.
 1: $N \leftarrow 2^n - 1$
 2: $\mathbb{F}_{2^n}$ is the extension field of $\mathbb{F}_2$ defined by $p(x)$
 3: for $t$ from 3 to $s$ do                ▷ $\widetilde{M}$ will be a $t \times t$ matrix
 4:     if $t \mid N$ then
 5:         $f(x) \leftarrow BM(p(x), t)$    ▷ Berlekamp-Massey Alg.
 6:         if $\deg(f(x)) < n$ then
 7:             Go to next $t$
 8:         end if
 9:         Create sets $Done := \{0\}$, $MinSet$, and $\mathcal{W}$
10:         Create $t \times t$ zero matrix $\widetilde{M} = (m_{i,j})$
11:         for $i \in \{(2k-1)t : 1 \leq k \leq z\}$ do
12:             $L \leftarrow \tau(i) \pmod t$
13:             Construct the coset $D_L$
14:             $c_L \leftarrow$ coset leader of $D_L$
15:             if $c_L \pmod t \notin Done$ then
16:                 Append $i$ to $MinSet$
17:                 Append $2k - 1$ to $\mathcal{W}$
18:                 $Done \leftarrow Done \cup \{y \pmod t : y \in D_L\}$
19:                 if $|Done| = t$ then
20:                     Output $\mathcal{W}$ and $\#cp \leftarrow \frac{n}{|D_L|}$
21:                     $m_{1,1} \leftarrow \#cp \cdot (t-1) + 1$
22:                     for $r$ from 2 to $t$ do
23:                         $m_{r,r} \leftarrow \#cp$
24:                         $m_{1,r} = m_{r,1} \leftarrow -\#cp$
25:                     end for
26:                     Output $\#dBSeqs \leftarrow \det(\widetilde{M})$
27:                     break $i$
28:                 end if
29:             end if
30:         end for
31:         if $|Done| < t$ then
32:             No star spanning tree centered at $[\mathbf{u}_0]$ found
33:         end if
34:     end if
35: end for
```

The value of s in Line 3 of Algorithm 1 upper bounds the choice of t. Running the algorithm with a random choice of $p(x)$ for all $n \in [\![10, 300]\!]$, we set $s = 2000$ for $n \leq 100$ and $s = 1000$ for $100 < n \leq 300$. Limiting z in Line 11 to 2000 was enough for all but very few parameter sets.

Algorithm 1 is practical. Averaging over 10 randomly selected primitive polynomials of degree 300, it took about 3 hour 46 minutes to compute the certificates for <u>all</u> valid $3 \leq t < 1000$ given a $p(x)$.

TABLE III
STAR AND ALMOST-STAR SPANNING TREE CERTIFICATES

No.	n	$p(x)$	t	$f(x)$	Star Witness $\mathcal{W}$	$\#cp$	$\#dBSeqs$	Time
1	20	$\{3\}$	155	$\{16,15,14,13,11,10,9,8,5,4,3\}$	$\{1,3,5,7,9,11,13,15,17,23,27,61,157\}$	4	2^{308}	0.01s
2	60	$\{1\}$	31	$\{39,35,31,20,18,16,10,8,4,2,1\}$	$\{1,5,7,9,11,25\}$	12	$\approx 2^{107.55}$	0.00s
3	100	$\{37\}$	25	$\{96,68,64,37,36,32,4\}$	$\{1,15\}$	25	$\approx 2^{111.45}$	0.00s
4	120	$\{49,2,1\}$	341	$\{117,116,112,103,100,96,95,94,93,92,91,88,86,84,79,78,77,74,71,70,68,66,65,64,63,62,57,51,48,43,42,39,38,37,35,34,32,30,27,26,24,21,17,16,15,12,10,8,7,5,3,2,1\}$	$\{1,3,5,7,9,11,15,17,19,21,23,25,31,33,35,37,39,41,49,53,59,61,65,67,77,87,91,93,97,113,115,119,137,159,167,189,199\}$	12	$\approx 2^{1218.89}$	4.39s
5	128	$\{7,2,1\}$	255	$\{128,127,126,125,123,122,119,118,117,114,112,109,108,106,105,103,99,98,96,94,93,91,89,87,83,82,81,74,70,68,67,66,65,63,60,59,56,53,52,51,50,46,45,44,43,42,41,40,39,38,36,31,30,29,28,27,26,25,24,22,15,13,11,10,9,6,4,1\}$	$\{1,3,5,7,9,11,13,15,17,19,21,27,29,31,33,37,43,45,47,53,57,65,77,79,83,101,107,123,133,141,145,177,187,929\}$	64	2^{1524}	1m37s
6	130	$\{3\}$	93	$\{97,89,64,63,48,47,43,42,21,11,10,5,3,2,1\}$	$\{1,3,5,9,11,15,23,31,33,35,43,73,101\}$	26	$\approx 2^{432.44}$	0.10s
7	300	$\{7\}$	15	$\{280,260,240,220,200,180,160,140,120,100,80,60,40,20,7\}$	$\{1,3,5,9\}$	150	$\approx 2^{101.20}$	9.01s
8			77	$\{273,220,219,193,192,191,165,164,139,111,110,86,85,83,82,30,29,28,7,6,5,4,3,2,1\}$	$\{1,3,5,97,125\}$	100	$\approx 2^{504.93}$	57.57s
No.	n	$p(x)$	t	$f(x)$	Almost-Star Witness $\mathcal{W}$ and ℓ	$\#cp$	$\#dBSeqs$	Time
1	20	$\{3\}$	205	$\{18,17,15,14,9,8,4,2,1\}$	$\{1,3,5,7,9,11,21,23,25,41,53,155\}$, $\ell=2$	20	$\approx 2^{881.67}$	0.04s
2			825	$\{19,14,11,5,4,1\}$	$\{1,3,5,7,9,11,13,15,17,23,25,27,31,35,39,41,43,45,47,49,53,57,61,63,65,69,71,89,93,105,111,115,123,171,175,179,187,211,237,239,255,265,377,381,685,821,1297\}$, $\ell=10$	4	$\approx 2^{1913.27}$	0.40s
3	29	$\{2\}$	233	$\{24,22,20,18,16,15,14,13,12,11,10,9,8,7,5,4,2\}$	$\{1,3,5,9,15,17,19,33,79\}$, $\ell=2$	29	$\approx 2^{1127.05}$	0.08s
4	128	$\{7,2,1\}$	255	See Entry 5 above	$\{1,3,5,7,9,11,13,17,19,23,25,29,31,33,37,45,49,53,55,57,67,89,91,103,107,111,119,139,143,159,201,237,251,343,465\}$, $\ell=2$	128	2^{1778}	7m38s
5	130	$\{3\}$	131	$\{96,72,65,48,36,34,24,17,12,10,5,4,2\}$	$\{1,171\}$, $\ell=2$	130	$\approx 2^{912.91}$	3.88s

In instances where no star spanning tree can be certified, one sets aside the unique edge E_0 between $[\mathbf{0}]$ and $[\mathbf{u}_0]$ and modifies Algorithm 1 to find a certificate for star trees centered at $[\mathbf{u}_\ell]$ for a chosen $\ell \in [\![1, t-1]\!]$ with vertices $[\mathbf{u}_i]$ for all $i \in [\![0, t-1]\!] \setminus \ell$ as leaves. Adding E_0 back yields *almost-star spanning trees centered at* $[\mathbf{u}_\ell]$.

To be more precise, we replace 0 by ℓ in Line 9 and replace $(2k-1)t$ by $(2k-1)t+\ell$ in Line 11. The entries of Matrix $\widetilde{M}$ defined in Lines 21 to 25 are now given as follows.

```
$m_{1,1} \leftarrow 1 + \#cp$, $m_{1,\ell+1} = m_{\ell+1,1} \leftarrow -\#cp$
for $r$ from 2 to $t$ do
    $m_{r,r} \leftarrow \#cp$, $m_{\ell+1,r} = m_{r,\ell+1} \leftarrow -\#cp$
end for
$m_{\ell+1,\ell+1} \leftarrow \#cp \cdot (t-1)$
```

We have seen that for $x^{10} + x^3 + 1$ and $t = 31$ there is no star spanning tree centered at $[\mathbf{u}_0]$. There are $\approx 2^{99.68}$ almost-star spanning trees centered at $[\mathbf{u}_6]$ with witness $\mathcal{W} = \{1,3,7,9,13,17,21\}$. The associated irreducible polynomial is $x^{10} + x^9 + x^5 + x + 1$. Similarly, for $x^{20} + x^3 + 1$ and $t \in \{165, 341, 451, 465, 615, 775, 825\}$, there is no star spanning tree certificate produced. Going through $\ell \in [\![1, t-1]\!]$ produces certificates for almost-star spanning trees, *e.g.*, with $\ell = 10$ for $t \in \{615, 825\}$ and $\ell = 2$ for all other ts.

There are parameter sets for which there is a unique star spanning tree each, yielding only 1 de Bruijn sequence. Examples include $x^{20} + x^3 + 1$ with $t \in \{41, 123, 205, 275\}$, $x^{29} + x^2 + 1$ with $t \in \{233, 1103, 2089\}$, and $x^{130} + x^3 + 1$ with $t = 131$. There are certificates for almost-star spanning trees for all of them, ensuring the existence of a large number of de Bruijn sequences in each case. The search takes longer but remains practical.

Counting the number of, respectively, star and almost-star spanning trees (with center $[\mathbf{u}_2]$) for $p(x) = x^{128} + x^7 + x^2 + x + 1$ and $t = 255$ gives us 2^{1524} and 2^{1778} sequences while [4, Example 3] yields 2^{1032} sequences.

Table III lists down some samples in two parts: star and almost-star. For a compact presentation we use sparse primitive polynomials. They are either trinomials, *i.e.*, $p(x) = x^n + x^k + 1$ with $1 \leq k < n$, or $p(x) = x^n + x^k + x^j + x^i + 1$ with $1 \leq i < j < k < n$. MAGMA [16] easily generates a primitive polynomial of either type for $n \leq 920$.

Given n and t, the polynomials $p(x)$ and $f(x)$ are presented as sets whose elements are the powers of x between 1 and $n-1$ whose coefficients are 1. Hence, for $n = 130$ and $t = 31$, $p(x) = x^{130} + x^3 + 1$ and $f(x) = x^{130} + x^{63} + x^{31} + x^{15} + x^7 + x^3 + 1$. Its witness $\mathcal{W} = \{1, 3, 7, 9, 17, 45\}$ generates $\Delta_{\mathcal{W}} = \{31, 93, 217, 279, 527, 1395\}$, implying

$$\{Y_i = \tau(i) \pmod{31} : i \in \Delta_{\mathcal{W}}\} = \{20, 16, 14, 13, 23, 12\}.$$

The corresponding sets $\{j \pmod{t} : j \in D_{Y_i}\}$ are

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

Their union is $[\![1, 30]\!]$. Note that $[\mathbf{u}_0]$ and $[\mathbf{u}_\ell]$ share $\frac{130}{5} = 26$ conjugate pairs for $\ell \in [\![1, 30]\!]$. Computing for $\#dBSeqs$ is then straightforward. The other entries can be similarly interpreted. The recorded running time is for the specified $(n, p(x), t)$ with ℓ added for cases where the center of the almost-star trees is $[\mathbf{u}_\ell]$.

Computations for the certificates are done on a laptop with Ubuntu 16.04 OS powered by an Intel Hasswell i5-4300U CPU 1.90GHz running MAGMA V2.20-10. The current limit for an efficient computation is $n = 300$ without a companion file of size 1.8 GB. With the file, n goes up to 460.

VI. Product of Irreducibles

The approach via Zech's logarithm can be used in tandem with the one we proposed in [2] to generate de Bruijn sequences of even larger orders. To keep the exposition brief, we retain the notations from the said reference.

Let $\{f_1(x), f_2(x), \ldots, f_s(x)\}$ be a set of s pairwise distinct irreducible polynomials over $\mathbb{F}_2$. Each $f_i(x)$ has degree n_i, order e_i with $t_i = \frac{2^{n_i}-1}{e_i}$, and a root β_i. Let the respective associated primitive polynomials be $p_i(x)$ with degree n_i and root α_i. Hence, $\Omega(f_i(x)) = [\mathbf{0}] \cup [\mathbf{u}_0^i] \cup [\mathbf{u}_1^i] \cup \ldots \cup [\mathbf{u}_{t_i-1}^i]$.

Let the initial state of $\mathbf{u}_0^i$ be $(1, \mathbf{0}) \in \mathbb{F}_2^{n_i}$. The initial states of $\mathbf{u}_j$ for $j \in [\![1, t_i - 1]\!]$ follows by decimating the appropriate m-sequence $\mathbf{m}_i$ generated by $p_i(x)$. From hereon, let

$$f(x) := \prod_{i=1}^{s} f_i(x) \text{ and } n := \sum_{i=1}^{s} n_i. \tag{16}$$

We use the expression for the cycle structure of $\Omega(f(x))$ given in [2, Lemma 3 Eq. (7)]. For any cycle $\Gamma_1 := [\mathbf{u}_{i_1}^1 + L^{\ell_2}\mathbf{u}_{i_2}^2 + \cdots + L^{\ell_s}\mathbf{u}_{i_s}^s]$ containing a state $\mathbf{v}$ the goal is to identify a cycle Γ_2 that contains $\widehat{\mathbf{v}}$.

Letting $\mathcal{P}$ be the matrix defined in [2, Section III.B],

$$\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_s)\mathcal{P} \text{ with } \mathbf{v}_i := \varphi(\alpha_i^{j_i}) \text{ for } i \in [\![1, s]\!].$$

One then gets a state $\mathbf{a}_i$ of some nonzero sequence in $\Omega(f_i(x))$ satisfying $(\mathbf{a}_1, \ldots, \mathbf{a}_s)\mathcal{P} = (1, \mathbf{0})$. The exact position of each $\mathbf{a}_i$ in the corresponding cycle in $\Omega(f_i(x))$, *i.e.*, the exact value of γ_i satisfying $\mathbf{a}_i = \varphi(\alpha_i^{\gamma_i})$ is computed using the method from Section III or by an exhaustive search when n_i is small. The conjugate state $\widehat{\mathbf{v}} = (\mathbf{b}_1, \ldots, \mathbf{b}_s)\mathcal{P}$ of $\mathbf{v}$ must then be

$$\begin{aligned} \mathbf{b}_i &= \varphi(\alpha_i^{j_i}) + \varphi(\alpha_i^{\gamma_i}) = \varphi(\alpha_i^{j_i} + \alpha_i^{\gamma_i}) \\ &= \varphi(\alpha_i^{\gamma_i}(1 + \alpha_i^{j_i - \gamma_i})) = \varphi(\alpha_i^{\gamma_i + \tau_i(j_i - \gamma_i)}), \end{aligned} \tag{17}$$

with τ_i based on $p_i(x)$. If $\mathbf{b}_i$ is the j_i-th state of $\mathbf{u}_{k_i}^i$ for all i, then $\widehat{\mathbf{v}}$ must be in cycle $\left[L^{j_1}\mathbf{u}_{k_1}^1 + L^{j_2}\mathbf{u}_{k_2}^2 + \cdots + L^{j_s}\mathbf{u}_{k_s}^s\right]$.

Thus, given any nonzero cycle Γ_1 in $\Omega(f(x))$ we can determine any of its state $\mathbf{v}$, find the conjugate state $\widehat{\mathbf{v}}$, and the cycle Γ_2 that $\widehat{\mathbf{v}}$ is a state of. If so desired, all conjugate pairs shared by any adjacent cycles can be determined explicitly. Finally, to produce actual de Bruijn sequences, one follows the steps detailed in [2, Sections IV and VI].

Using $f(x)$ in (16) may become crucial when substantially more than Λ_n de Bruijn sequences of order a Mersenne exponent n need to be produced. The simplest choice is to use $s = 2$ with $f_1(x)$ an irreducible polynomial of a small degree, *e.g.*, $1 + x$ or $1 + x + x^2$, and $f_2(x)$ any irreducible non-primitive polynomial of degree $n-1$ or $n-2$, respectively. If even more de Bruijn sequences are required, one uses $s \geq 3$ and choose small n_i for $i \in [\![1, s - 1]\!]$ since computing the Zech logarithm tables relative to small n_is is easy.

VII. Conclusions

We propose a novel approach to generate binary de Bruijn sequences via Zech's logarithms. It is guaranteed to work for all order n. Its practical feasibility is demonstrated by producing many such sequences of large orders within reasonable time and memory expenditures. Many design parameters certify the existence of star or near-star spanning trees. The information supplied by the certificate significantly expedites the sequence generation.

We establish a salient property of Zech's logarithm and a rapid method to generate irreducible polynomials of degree $m \mid n$ from a primitive polynomial of degree n.

Our approach is capable of generating de Bruijn sequences of orders larger than the current limit of MAGMA's efficient computation of Zech's logarithms. One uses LFSRs whose characteristic polynomial are products of distinct irreducible polynomials to reach this goal.

For large n, storing a de Bruijn sequence of order n is not feasible. One may opt to output subsequent states up to some specified total length, *e.g.*, measured in file size. The initial state and the spanning tree(s) can be chosen randomly to cater to application-specific requirements.

Acknowledgements

M. F. Ezerman gratefully acknowledges the hospitality of the School of Mathematics and Statistics of Zhengzhou University, in particular Prof. Yongcheng Zhao, during a visit where this work was initiated.

References

[1] N. G. de Bruijn, “A combinatorial problem,” *Koninklijke Nederlandse Akademie v. Wetenschappen*, vol. 49, pp. 758–764, 1946.

[2] Z. Chang, M. F. Ezerman, S. Ling, and H. Wang, “On binary de Bruijn sequences from LFSRs with arbitrary characteristic polynomials,” *CoRR*, vol. abs/1611.10088, 2016. [Online]. Available: http://arxiv.org/pdf/1611.10088

[3] N. J. A. Sloane, “Mersenne exponents,” primes p such that $2^p - 1$ is (Mersenne) prime. [Online]. Available: https://oeis.org/A000043

[4] J. Dong and D. Pei, “Construction for de Bruijn sequences with large stage,” *Designs, Codes and Cryptography*, pp. 1–16, Dec 2016. [Online]. Available: http://dx.doi.org/10.1007/s10623-016-0309-1

[5] J. Arndt, *Matters Computational: Ideas, Algorithms, Source Code*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.

[6] S. W. Golomb and G. Gong, *Signal Design for Good Correlation: for Wireless Communication, Cryptography, and Radar*. New York: Cambridge Univ. Press, 2004.

[7] H. Fredricksen, “A survey of full length nonlinear shift register cycle algorithms,” *SIAM Review*, vol. 24, no. 2, pp. 195–221, 1982.

[8] E. R. Hauge and J. Mykkeltveit, “On the classification of de Bruijn sequences,” *Discrete Math.*, vol. 148, no. 13, pp. 65 – 83, 1996.

[9] E. R. Hauge and T. Helleseth, “De Bruijn sequences, irreducible codes and cyclotomy,” *Discrete Math.*, vol. 159, no. 1-3, pp. 143–154, 1996.

[10] T. van Aardenne-Ehrenfest and N. G. de Bruijn, “Circuits and trees in oriented linear graphs,” *Simon Stevin*, vol. 28, pp. 203–217, 1951.

[11] R. Lidl and H. Niederreiter, *Finite Fields*, ser. Encyclopaedia of Mathematics and Its Applications. New York: Cambridge Univ. Press, 1997.

[12] C. Jacobi, “Über die kreistheilung und ihre anwendung auf die zahlentheorie.” *Journal für die reine und angewandte Mathematik*, vol. 30, pp. 166–182, 1846.

[13] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1996.

[14] S. Gao and D. Panario, “Tests and constructions of irreducible polynomials over finite fields,” in *Foundations of Computational Mathematics: Selected Papers of a Conference, Rio de Janeiro, January 1997*, F. Cucker and M. Shub, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 346–361.

[15] K. Huber, “Some comments on Zech’s logarithms,” *IEEE Trans. on Inform. Theory*, vol. 36, no. 4, pp. 946–950, Jul 1990.

[16] W. Bosma, J. Cannon, and C. Playoust, “The Magma algebra system. I. The user language,” *J. Symbolic Comput.*, vol. 24, no. 3-4, pp. 235–265, 1997.