Efficient Construction of Long Orientable Sequences

Daniel Gabrić
University of Guelph, Canada

Joe Sawada
University of Guelph, Canada

Abstract

An orientable sequence of order n is a cyclic binary sequence such that each length-n substring appears at most once *in either direction*. Maximal length orientable sequences are known only for $n \leq 7$, and a trivial upper bound on their length is $2^{n-1} - 2^{\lfloor (n-1)/2 \rfloor}$. This paper presents the first efficient algorithm to construct orientable sequences with asymptotically optimal length; more specifically, our algorithm constructs orientable sequences via cycle-joining and a successor-rule approach requiring $O(n)$ time per bit and $O(n)$ space. This answers a longstanding open question from Dai, Martin, Robshaw, Wild [Cryptography and Coding III (1993)]. Our sequences are applied to find new longest-known orientable sequences for $n \leq 20$.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics

Keywords and phrases orientable sequence, de Bruijn sequence, concatenation tree, cycle-joining, universal cycle

Digital Object Identifier 10.4230/LIPIcs...

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Orientable sequences were introduced by Dai, Martin, Robshaw, and Wild [7] with applications related to robotic position sensing. In particular, consider an autonomous robot with limited sensors. To determine its location on a cyclic track labeled with black and white squares, the robot scans a window of n squares directly beneath it. For the position *and* orientation to be uniquely determined, the track must designed with the property that each length n window can appear at most once in *either direction*. A cyclic binary sequence (track) with such a property is called an *orientable sequence* of order n (an $\mathcal{OS}(n)$). By this definition, an orientable sequence does not contain a length-n substring that is a palindrome.

Example 1 Consider $\mathcal{S} = 001011$. In the forward direction, including the wraparound, $\mathcal{S}$ contains the six 5-tuples 00101, 01011, 10110, 01100, 11001, and 10010; in the reverse direction $\mathcal{S}$ contains 11010, 10100, 01001, 10011, 00110, and 01101. Since each substring is unique, $\mathcal{S}$ is an $\mathcal{OS}(5)$ with length (period) six.

Orientable sequences do not exist for $1 < n < 5$, and somewhat surprisingly, the maximum length M_n of an $\mathcal{OS}(n)$ is known only for $n = 1, 5, 6, 7$. Since the number of palindromes of length n is $2^{\lfloor (n+1)/2 \rfloor}$, a trivial upper bound on M_n is $(2^n - 2^{\lfloor (n+1)/2 \rfloor})/2 = 2^{n-1} - 2^{\lfloor (n-1)/2 \rfloor}$.

In addition to providing a tighter upper bound, Dai, Martin, Robshaw, and Wild [7] provide a lower bound on M_n by demonstrating the *existence* of $\mathcal{OS}(n)$s via cycle-joining with length L_n asymptotic to their upper bound; see Section 1.1 for the explicit upper and lower bounds.. They conclude by stating the following open problem relating to orientable sequences whose lengths (periods) attain the lower bound.

> *We note that the lower bound on the maximum period was obtained using an existence construction . . . It is an open problem whether a more practical procedure exists for the construction of orientable sequences that have this asymptotically optimal period.*

Recently, some progress was made in this direction by Mitchell and Wild [20]. They apply Lempel's lift [18] to obtain an $\mathcal{OS}(n)$ recursively from an $\mathcal{OS}(n-1)$. This construction can generate orientable sequences in $O(1)$-amortized time per bit; however, it requires exponential space, and there is an exponential time delay before the first bit can be output. Furthermore, they state that they "only *partially* answer the question, since the periods/lengths of the sequences produced are not asymptotically optimal."

Main result: By developing a parent rule to define a cycle-joining tree, we construct an $\mathcal{OS}(n)$ of length L_n in $O(n)$ time per bit using $O(n)$ space.

Outline. In Section 1.1, we review the lower bound L_n and upper bound U_n from [7]. In Section 2, we present necessary background definitions and notation, including a review of the cycle-joining technique. In Section 3, we provide a parent rule for constructing a cycle-joining tree composed of "reverse-disjoint" cycles. This leads to our $O(n)$ time per bit construction of orientable sequences of length L_n. In Section 4 we discuss the algorithmic techniques used to extend our constructed orientable sequences to find longer ones for $n \leq 20$. We conclude in Section 5 with a summary of our results and directions for future research. An implementation of our construction is available for download at `http://debruijnsequence.org/db/orientable`.

1.1 Bounds on M_n

Dai, Martin, Robshaw, and Wild [7] gave a lower bound L_n and an upper bound U_n on the maximum length M_n of an $\mathcal{OS}(n)$.[1] Their lower bound L_n is the following, where μ is the Möbius function:

$$L_n = \left(2^{n-1} - \frac{1}{2}\sum_{d|n}\mu(n/d)\frac{n}{d}H(d)\right), \quad \text{where} \quad H(d) = \frac{1}{2}\sum_{i|d} i\left(2^{\lfloor\frac{i+1}{2}\rfloor} + 2^{\lfloor\frac{i}{2}\rfloor+1}\right).$$

Their upper bound U_n is the following:[1]

$$U_n = \begin{cases} 2^{n-1} - \frac{41}{9}2^{\frac{n}{2}-1} + \frac{n}{3} + \frac{16}{9} & \text{if } n \bmod 4 = 0, \\ 2^{n-1} - \frac{31}{9}2^{\frac{n-1}{2}} + \frac{n}{3} + \frac{19}{9} & \text{if } n \bmod 4 = 1, \\ 2^{n-1} - \frac{41}{9}2^{\frac{n}{2}-1} + \frac{n}{6} + \frac{20}{9} & \text{if } n \bmod 4 = 2, \\ 2^{n-1} - \frac{31}{9}2^{\frac{n-1}{2}} + \frac{n}{6} + \frac{43}{18} & \text{if } n \bmod 4 = 3. \end{cases}$$

These bounds are calculated in Table 1 for n up to 20. This table also illustrates the length R_n of the $\mathcal{OS}(n)$ produced by the recursive construction by Mitchell and Wild [20], starting from an initial orientable sequence of length 80 for $n = 8$. The column labeled L_n^* indicates the longest known orientable sequences we discovered by applying a combination of techniques (discussed in Section 4) to our orientable sequences of length L_n.

n	R_n	L_n	L_n^*	U_n
5	-	0	**6**	6
6	-	6	**16**	17
7	-	14	**36**	40
8	80	48	92	96
9	161	126	174	206
10	322	300	416	443
11	645	682	844	918
12	1290	1530	1844	1908
13	2581	3276	3700	3882
14	5162	6916	7694	7905
15	10325	14520	15394	15948
16	20650	29808	31483	32192
17	41301	61200	63135	64662
18	82602	124368	128639	129911
19	165205	252434	257272	260386
20	330410	509220	519160	521964

Table 1 Lower bounds R_n, L_n, L_n^* and upper bound U_n for M_n.

1.2 Related work

Recall the problem of determining a robot's position and orientation on a track. Suppose now that we allow the track to be non-cyclic. That is, the beginning of the track and the end of the track are not

[1] These bounds correspond to $\tilde{L}_n$ and $\tilde{U}_n$, respectively, as they appear in [7].

connected. Then the corresponding sequence that allows one to determine orientation and position is called an *acyclic orientable sequence*. One does not consider the substrings in the wraparound for this variation of an orientable sequence. Note that one can always construct an acyclic $\mathcal{OS}(n)$ from a cyclic $\mathcal{OS}(n)$ by taking the cyclic $\mathcal{OS}(n)$ and appending its prefix of length $n-1$ to the end. See the paper by Burns and Mitchell [5] for more on acyclic orientable sequences, which they call *aperiodic 2-orientable window sequences*. Alhakim et al. [2] generalize the recursive results of Mitchell and Wild [20] to construct orientable sequences over an alphabet of arbitrary size $k \geq 2$; they also generalize the upper bound, by Dai et al. [7], on the length of an orientable sequence. Rampersad and Shallit [21] showed that for every alphabet size $k \geq 2$ there is an infinite sequence such that for every sufficiently long substring, the reversal of the substring does not appear in the sequence. Fleischer and Shallit [11] later reproved the results of the previous paper using theorem-proving software. See [6, 19] for more work on sequences avoiding reversals of substrings.

2 Preliminaries

Let $\mathbf{B}(n)$ denote the set of all length-n binary strings. Let $\alpha = \mathtt{a}_1\mathtt{a}_2\cdots\mathtt{a}_n \in \mathbf{B}(n)$ and $\beta = \mathtt{b}_1\mathtt{b}_2\cdots\mathtt{b}_m \in \mathbf{B}(m)$ for some $m, n \geq 0$. Throughout this paper, we assume $0 < 1$ and use lexicographic order when comparing two binary strings. More specifically, we say that $\alpha < \beta$ either if α is a prefix of β or if $\mathtt{a}_i < \mathtt{b}_i$ for the smallest i such that $\mathtt{a}_i \neq \mathtt{b}_i$. We say that α is a *rotation* of β if $m = n$ and there exist strings x and y such that $\alpha = xy$ and $\beta = yx$. The *weight* (density) of a binary string is the number of 1s in the string. Let $\overline{\mathtt{a}}_i$ denote the complement of bit $\mathtt{a}_i$. Let α^R denote the reversal $\mathtt{a}_n\cdots\mathtt{a}_2\mathtt{a}_1$ of α; α is a *palindrome* if $\alpha = \alpha^R$. For $j \geq 1$, let α^j denote j copies of α concatenated together. If $\alpha = \gamma^j$ for some non-empty string γ and some $j > 1$, then α is said to be *periodic*[2]; otherwise, α is said to be *aperiodic* (or *primitive*). For example, the English word $\mathtt{hotshots} = (\mathtt{hots})^2$ is periodic, but $\mathtt{hots}$ is aperiodic.

A *necklace class* is an equivalence class of strings under rotation; let $[\alpha]$ denote the set of strings in α's necklace class. We say α is a *necklace* if it is the lexicographically smallest string in $[\alpha]$. Let $\mathbf{N}(n)$ denote the set of length-n necklaces. A *bracelet class* is an equivalence class of strings under rotation and reversal; let $\langle\alpha\rangle$ denote the set of strings in α's bracelet class. Thus, $\langle\alpha\rangle = [\alpha] \cup [\alpha^R]$. We say α is a *bracelet* if it is the lexicographically smallest string in $\langle\alpha\rangle$. Note that in general, a bracelet is always a necklace, but a necklace need not be a bracelet. For example, the string 001011 is both a bracelet and a necklace, but the string 001101 is a necklace and is not a bracelet.

A necklace α is *symmetric* if it belongs to the same necklace class as α^R, i.e., both α and α^R belong to $[\alpha]$. By this definition, a symmetric necklace is necessarily a bracelet. If a necklace or bracelet is not symmetric, it is said to be *asymmetric*. Let $\mathbf{A}(n)$ denote the set of all asymmetric bracelets of order n. Table 2 lists all 60 necklaces of length $n = 9$ partitioned into asymmetric necklace pairs and symmetric necklaces. The asymmetric necklace pairs belong to the same bracelet class, and the first string in each pair is an asymmetric bracelet. Thus, $|\mathbf{A}(9)| = 14$. In general, $|\mathbf{A}(n)|$ is equal to the number of necklaces of length n minus the number of bracelets of length n; for $n = 6, 7, \ldots 15$, this sequence of values $|\mathbf{A}(n)|$ is given by 1, 2, 6, 14, 30, 62, 128, 252, 495, 968 and it corresponds to sequence A059076 in The On-Line Encyclopedia of Integer Sequences [25]. Asymmetric bracelets have been studied previously in the context of efficiently ranking/unranking bracelets [1]. One can test whether a string is an asymmetric bracelet in linear time using linear space; see Theorem 1.

[2] Periodic strings are are also known as *powers* in the literature. The term *periodic* is sometimes used to denote a string of the form $(\alpha\beta)^i\alpha$ where α is non-empty, β is possibly empty, $i \geq 1$, and $\frac{|(\alpha\beta)^i\alpha|}{|\alpha\beta|} \geq 2$. Under this definition, the word $\mathtt{alfalfa}$ is periodic, but $\mathtt{bonobo}$ is not.

Asymmetric necklace pairs	Symmetric necklaces		
000001011 , 000001101	000000000	000100011	001110111
000010011 , 000011001	000000001	000101101	001111111
000010111 , 000011101	000000011	000110011	010101011
000100101 , 000101001	000000101	000111111	010101111
000100111 , 000111001	000000111	001001001	010111111
000101011 , 000110101	000001001	001001111	011011011
000101111 , 000111101	000001111	001010011	011011111
000110111 , 000111011	000010001	001010101	011101111
001001011 , 001001101	000010101	001011101	011111111
001010111 , 001110101	000011011	001100111	111111111
001011011 , 001101101	000011111	001101011	
001011111 , 001111101			
001101111 , 001111011			
010110111 , 010111011			

Table 2 A listing of all 60 necklaces in $\mathbf{N}(9)$ partitioned into asymmetric necklace pairs and symmetric necklaces. The first column of the asymmetric necklaces corresponds to the 14 asymmetric bracelets $\mathbf{A}(9)$.

▶ **Theorem 1.** *One can determine whether a string α is in $\mathbf{A}(n)$ in $O(n)$ time using $O(n)$ space.*

Proof. A string α will belong to $\mathbf{A}(n)$ if α is a necklace and the necklace of $[\alpha^R]$ is lexicographically larger than α. These tests can be computed in $O(n)$ time using $O(n)$ space [3]. ◀

Lemma 2 is considered a folklore result in combinatorics on words; see Theorem 4 in [4] for a variant of the lemma. We provide a short proof for the interested reader.

▶ **Lemma 2.** *A necklace α is symmetric if and only if there exists palindromes β_1 and β_2 such that $\alpha = \beta_1\beta_2$.*

Proof. Suppose α is a symmetric necklace. By definition, it is equal to the necklace of $[\alpha^R]$. Thus, there exist strings β_1 and β_2 such that $\alpha = \beta_1\beta_2 = (\beta_2\beta_1)^R = \beta_1^R\beta_2^R$. Therefore, $\beta_1 = \beta_1^R$ and $\beta_2 = \beta_2^R$, which means β_1 and β_2 are palindromes. Suppose there exists two palindromes β_1 and β_2 such that $\alpha = \beta_1\beta_2$. Since β_1 and β_2 are symmetric, we have that $\alpha^R = (\beta_1\beta_2)^R = \beta_2^R\beta_1^R = \beta_2\beta_1$. So α belongs to the same necklace class as α^R and hence is symmetric. ◀

▶ **Corollary 3.** *If $\alpha = 0^s\beta$ is a symmetric bracelet such that the string β begins and ends with 1 and does not contain 0^s as a substring, then β is a palindrome.*

2.1 Cycle joining

Given $\mathbf{S} \subseteq \mathbf{B}(n)$, a *universal cycle* U for $\mathbf{S}$ is a cyclic sequence of length $|\mathbf{S}|$ that contains each string in $\mathbf{S}$ as a substring (exactly once). Thus, an orientable sequence is a universal cycle. If $\mathbf{S} = \mathbf{B}(n)$ then U is known as a *de Bruijn sequence*. Given a universal cycle U for $\mathbf{S}$, a *successor rule* for U is a function $f : \mathbf{S} \to \{0, 1\}$ such that $f(\alpha)$ is the bit following α in U.

Cycle-joining is perhaps the most fundamental technique applied to construct universal cycles; for some applications, see [8, 9, 10, 12, 14, 16, 17, 23, 24]. If $\mathbf{S}$ is closed under rotation, then it can be partitioned into necklace classes (cycles); each cycle is disjoint. Let $\alpha = \mathtt{a}_1\mathtt{a}_2\cdots\mathtt{a}_n$ and $\hat{\alpha} = \overline{\mathtt{a}}_1\mathtt{a}_2\cdots\mathtt{a}_n$; we say $(\alpha, \hat{\alpha})$ is a *conjugate pair*. Two disjoint cycles can be joined if they each contain one string of a *conjugate pair* as a substring. This approach resembles Hierholzer's algorithm to construct an Euler cycle in an Eulerian graph [15].

Example 2 Consider disjoint subsets $\mathbf{S}_1 = [011111] \cup [001111]$ and $\mathbf{S}_2 = [010111] \cup [010101]$, where $n = 6$. Then $U_1 = 110011\underline{110111}$ is a universal cycle for $\mathbf{S}_1$ and $U_2 = 01\underline{010111}$ is a universal cycle for $\mathbf{S}_2$. Since $(110111, 010111)$ is a conjugate pair, $U = 110011\underline{110111} \cdot 01\underline{010111}$ is a universal cycle for $\mathbf{S}_1 \cup \mathbf{S}_2$.

If all necklace cycles can be joined via conjugate pairs to form a cycle-joining tree, then the tree defines a universal U for $\mathbf{S}$ with a corresponding successor rule (see Section 3 for an example).

For most universal cycle constructions, a corresponding cycle-joining tree can be defined by a rather simple *parent rule*. For example, when $\mathbf{S} = \mathbf{B}(n)$, the following are perhaps the *simplest* parent rules that define how to construct cycle-joining trees with nodes corresponding to $\mathbf{N}(n)$ [13, 22].

- **Last-0**: rooted at 1^n and the parent of every other node $\alpha \in \mathbf{N}(n)$ is obtained by flipping the last 0.
- **First-1**: rooted at 0^n and the parent of every other node $\alpha \in \mathbf{N}(n)$ is obtained by flipping the first 1.
- **Last-1**: rooted at 0^n and the parent of every other node $\alpha \in \mathbf{N}(n)$ is obtained by flipping the last 1.
- **First-0**: rooted at 1^n and the parent of every other node $\alpha \in \mathbf{N}(n)$ is obtained by flipping the first 0.

These rules induce the cycle-joining trees T_1, T_2, T_3, T_4 illustrated in Figure 1 for $n = 6$. Note that for T_3 and T_4, the parent of a node α is obtained by first flipping the highlighted bit and then rotating the string to its lexicographically least rotation to obtain a necklace. Each node α and its parent β are joined by a conjugate pair, where the highlighted bit in α is the first bit in one of the conjugates. For example, the nodes $\alpha = 011011$ and $\beta = 001011$ in T_2 from Figure 1 are joined by the conjugate pair $(110110, 010110)$.

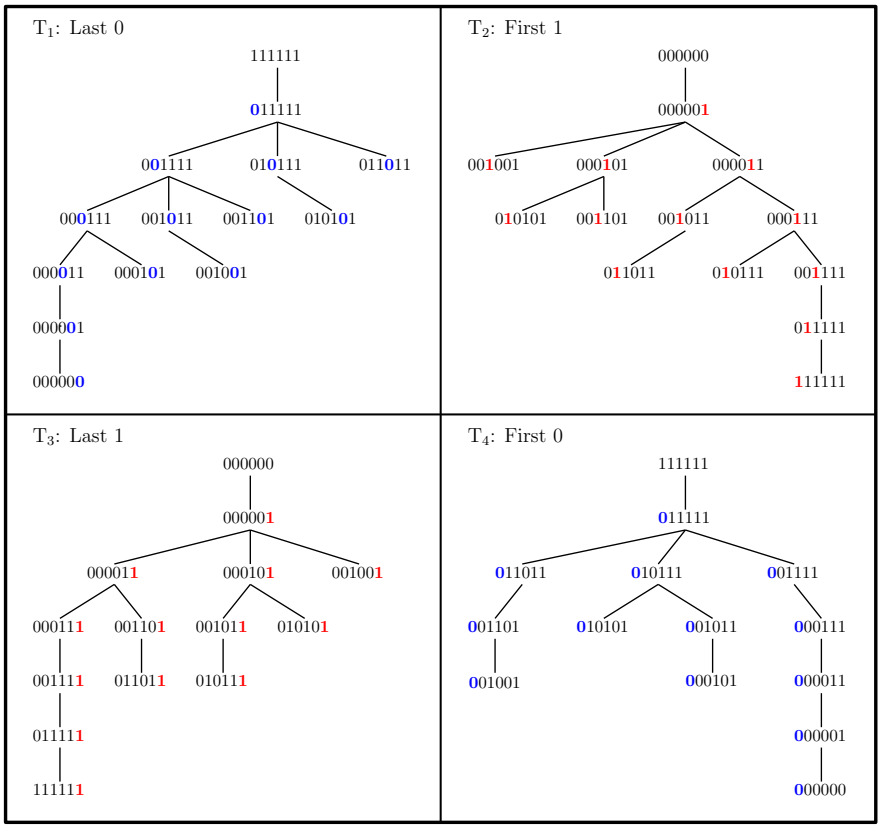

Figure 1 Cycle-joining trees for $\mathbf{B}(6)$ from simple parent rules.

3 An efficient cycle-joining construction of orientable sequences

Consider the set of asymmetric bracelets $\mathbf{A}(n) = \{\alpha_1, \alpha_2, \ldots, \alpha_t\}$. Recall, that each symmetric bracelet is a necklace. Let $\mathbf{S}(n) = [\alpha_1] \cup [\alpha_2] \cup \cdots \cup [\alpha_t]$. From [7], we have $|\mathbf{S}(n)| = L_n$. By its definition, there is no string $\alpha \in \mathbf{S}(n)$ such that $\alpha^R \in \mathbf{S}(n)$. Thus, a universal cycle for $\mathbf{S}(n)$ is an $\mathcal{OS}(n)$. For the rest of this section, we assume $n \geq 8$.

To construct a cycle-joining tree with nodes $\mathbf{A}(n)$, we apply a combination of three of the four simple parent rules described in the previous section. First, we demonstrate that there is no such parent rule, using at most two rules in combination. Observe, there are no necklaces in $\mathbf{A}(n)$ with weight 0, 1, 2, $n-2$, $n-1$, or, n. Thus, $0^{n-4}1011$ and $0^{n-5}10011$ are both necklaces in $\mathbf{A}(n)$ with minimal weight three. Similarly, 00101^{n-4} and 001101^{n-5} are necklaces in $\mathbf{A}(n)$ with maximal weight $n-3$. Therefore, when considering a parent rule for a cycle-joining tree with nodes $\mathbf{A}(n)$, the rule must be able to flip a 0 to a 1, or a 1 to a 0, i.e., if the rule applies a combination of the four rules from Section 2.1, it must include one of First-0 or Last-0, and one of First-1 and Last-1.

Let $\alpha = \mathtt{a}_1\mathtt{a}_2\cdots\mathtt{a}_n$ denote a necklace in $\mathbf{A}(n)$; it must begin with 0 and end with 1. Then let

- $\mathrm{first1}(\alpha)$ be the necklace $\mathtt{a}_1\cdots\mathtt{a}_{i-1}\mathbf{0}\mathtt{a}_{i+1}\cdots\mathtt{a}_n$, where i is the index of the first 1 in α;
- $\mathrm{last1}(\alpha)$ be the necklace of $[\mathtt{a}_1\mathtt{a}_2\cdots\mathtt{a}_{n-1}\mathbf{0}]$;
- $\mathrm{first0}(\alpha)$ be the necklace of $[\mathbf{1}\mathtt{a}_2\cdots\mathtt{a}_n]$;
- $\mathrm{last0}(\alpha)$ be the necklace $\mathtt{a}_1\cdots\mathtt{a}_{j-1}\mathbf{1}\mathtt{a}_{j+1}\cdots\mathtt{a}_n$, where j is the index of the last 0 in α.

Note that $\mathrm{first1}(\alpha)$ and $\mathrm{last0}(\alpha)$ are necklaces (easily observed by definition) obtained by flipping the i-th and j-th bit in α, respectively; $\mathrm{last1}(\alpha)$ and $\mathrm{first0}(\alpha)$ are the result of flipping a bit and rotating the resulting string to obtain a necklace. The next example illustrates that no two of the previous four parent rules can be applied in combination to obtain a spanning tree with nodes in $\mathbf{A}(n)$.

Example 3 Suppose $\mathrm{p}(\alpha)$ is a parent rule that applies a combination of the four parent rules, first1, last1, first0, last0, to construct a cycle-joining tree with nodes $\mathbf{A}(n)$. The following examples are for $n = 10$ but generalize to larger n. In both cases, we see that at least three of the parent rules must be applied in p.

Suppose p does not use first0; it must apply last0. Consider three asymmetric bracelets in $\mathbf{A}(10)$: $\alpha_1 = 0000001011$, $\alpha_2 = 0000010111$, and $\alpha_3 = 0011001011$. Clearly, $\mathrm{first1}(\alpha_1)$, $\mathrm{last1}(\alpha_1)$, and $\mathrm{last0}(\alpha_1)$ are symmetric. Thus, α_1 must be the root. Both $\mathrm{first1}(\alpha_2)$ and $\mathrm{last0}(\alpha_2)$ are symmetric, so p must apply last1. Note $\mathrm{last0}(\alpha_3)$ is symmetric and $\mathrm{last1}(\alpha_3) = 0001100101$ is not a bracelet, so p must apply first1.

Suppose p does not use last0; it must apply first0. Consider three asymmetric bracelets in $\mathbf{A}(10)$: $\beta_1 = 0000100011$, $\beta_2 = 0001001111$, and $\beta_3 = 0001100111$. Clearly, $\mathrm{first1}(\beta_1)$, $\mathrm{last1}(\beta_1)$, and $\mathrm{first0}(\beta_1)$ are symmetric. Thus, β_1 must be the root. Both $\mathrm{first1}(\beta_2)$ and $\mathrm{first0}(\beta_2)$ are symmetric, so p must apply last1. Both $\mathrm{last1}(\beta_3)$ and $\mathrm{first0}(\beta_3)$ are symmetric, so p must apply first1.

Let r_n denote the asymmetric bracelet $0^{n-4}1011$. We choose to use r_n to be the root of our cycle-joining tree since it is the lexicographically smallest asymmetric bracelet of length n.

Parent rule for cycle-joining $\mathbf{A}(n)$: Let r_n be the root. Let α denote a non-root node in $\mathbf{A}(n)$. Then

$$\mathrm{par}(\alpha) = \begin{cases} \mathrm{first1}(\alpha) & \text{if } \mathrm{first1}(\alpha) \in \mathbf{A}(n); \\ \mathrm{last1}(\alpha) & \text{if } \mathrm{first1}(\alpha) \notin \mathbf{A}(n) \text{ and } \mathrm{last1}(\alpha) \in \mathbf{A}(n); \\ \mathrm{last0}(\alpha) & \text{otherwise.} \end{cases} \tag{1}$$

▶ **Theorem 4.** *The parent rule* $\mathrm{par}(\alpha)$ *in* (1) *induces a cycle-joining tree with nodes* $\mathbf{A}(n)$ *rooted at* r_n.

Let $\mathbb{T}_n$ denote the cycle-joining tree with nodes $\mathbf{A}(n)$ induced by the parent rule in (1); Figure 2 illustrates $\mathbb{T}_9$. The proof of Theorem 4 relies on the following lemma.

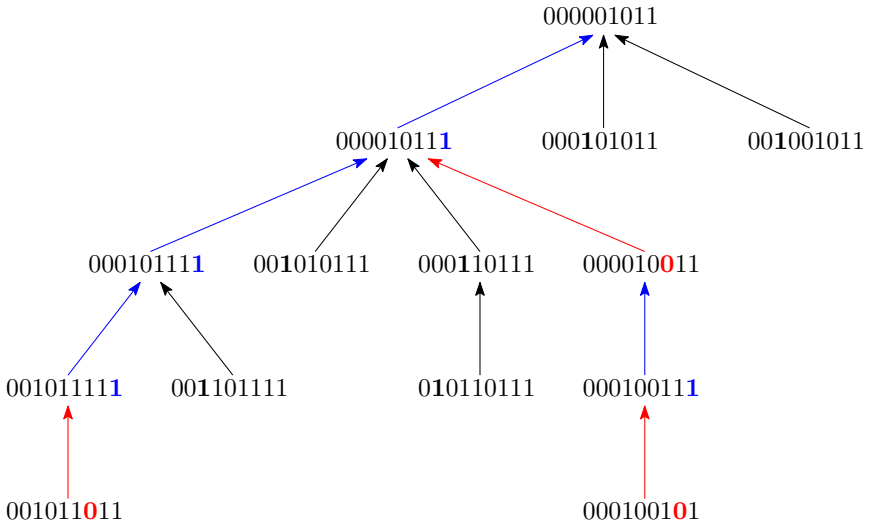

Figure 2 The cycle-joining tree $\mathbb{T}_9$. The black edges indicate that $\mathrm{par}(\alpha) = \mathrm{first1}(\alpha)$; the blue edges indicate that $\mathrm{par}(\alpha) = \mathrm{last1}(\alpha)$; the red edges indicate that $\mathrm{par}(\alpha) = \mathrm{last0}(\alpha)$.

▶ **Lemma 5.** *Let* $\alpha \neq r_n$ *be an asymmetric bracelet in* $\mathbf{A}(n)$. *If neither* $\mathrm{first1}(\alpha)$ *nor* $\mathrm{last1}(\alpha)$ *are in* $\mathbf{A}(n)$, *then the last* 0 *in* α *is at index* $n-2$ *or* $n-1$, *and both* $\mathrm{last0}(\alpha)$ *and* $\mathrm{last1}(\mathrm{last0}(\alpha))$ *are in* $\mathbf{A}(n)$.

Proof. Since α is an asymmetric bracelet, it must have the form $\alpha = 0^i1\beta01^j$ where $i, j \geq 1$ and $\beta 0$ does not contain 0^{i+1} as a substring. Furthermore, $1\beta01^j < (1\beta01^j)^R$, which implies $\beta01^{j-1} < (\beta01^{j-1})^R$.

Suppose $j > 2$**.** Since $\mathrm{last1}(\alpha) = 0^{i+1}1\beta01^{j-1}$ is not an asymmetric bracelet, we have $1\beta01^{j-1} \geq (1\beta01^{j-1})^R$. Thus, β begins with 1. Since $\mathrm{first1}(\alpha) = 0^{i+1}\beta01^j$ is not an asymmetric bracelet, Lemma 2 implies $\beta01^j \geq (\beta01^j)^R$, contradicting the earlier observation that $\beta01^{j-1} < (\beta01^{j-1})^R$. Thus, the last 0 in α is at index $n-2$ or $n-1$.

Suppose $j = 1$ **or** $j = 2$**.** Then the last 0 in α must be at position $n-2$ or $n-1$. Write $\alpha = x0y$ where $y = 1$ or $y = 11$. Since α is a bracelet, it is straightforward to see that $\mathrm{last0}(\alpha) = x1y$ is also a bracelet. If it is symmetric, Lemma 2 implies there exist palindromes β_1 and β_2 such that $\mathrm{last0}(\alpha) = x1y = \beta_1\beta_2$. However, flipping the 1 in $x1y$ that allows us to obtain α implies that α is greater than or equal to the necklace in $[\alpha^R]$, contradicting the assumption that α is an asymmetric bracelet. Thus, $\mathrm{last0}(\alpha)$ is an asymmetric bracelet.

Consider $\mathrm{last1}(\mathrm{last0}(\alpha)) = 0^{i+1}1\beta1^j$. Let $\beta = \mathtt{b}_1\mathtt{b}_2\cdots\mathtt{b}_m$. Suppose that $m = 0$. Then $\mathrm{last1}(\mathrm{last0}(\alpha)) = 0^{i+1}1^{j+1} \Rightarrow \mathrm{last0}(\alpha) = 0^i1^{j+2}$. Since $j = 1$ or $j = 2$, we have that $\mathrm{last0}(\alpha) = 0^i111$ or $\mathrm{last0}(\alpha) = 0^i1111$. Now α is the result of flipping one of the 1s in $\mathrm{last0}(\alpha)$ to a 0 and performing the appropriate rotation. But in every case, we end up with α being a symmetric necklace, a contradiction. Thus, assume $m \geq 1$. Suppose $\beta = 1^m$. Then, α is not an asymmetric bracelet, a contradiction. Suppose $\beta = 0^m$. If $j = 1$, then α is symmetric, a contradiction; if $j = 2$, then $\mathrm{last1}(\mathrm{last0}(\alpha)) = 0^{i+1}10^m11$ which is in $\mathbf{A}(n)$. For all other cases, β contains at least

one 1 and at least one 0; $m \geq 2$. Since β does not contain 0^{i+1} as a substring, by Lemma 2, we must show that (i) $\beta 1^{j-1} < 1^{j-1}\beta^R$, which implies $1\beta 1^j < 1^j\beta^R 1$, recalling that (ii) $\beta 01^{j-1} < 1^{j-1}0\beta^R$. Let ℓ be the largest index of β such that $\mathtt{b}_\ell = 1$. Then $\mathtt{b}_{\ell+1}\cdots\mathtt{b}_m = 0^{m-\ell}$; note that $\mathtt{b}_{\ell+1}\cdots\mathtt{b}_m$ is the empty string when $\ell = m$. Suppose $j = 1$. From (ii), we have $\mathtt{b}_1 = 0$ and $\mathtt{b}_2\cdots\mathtt{b}_{\ell-1}10^{m-\ell} < 0^{m-\ell}1\mathtt{b}_{\ell-1}\cdots\mathtt{b}_2$. But this implies that $\mathtt{b}_2\cdots\mathtt{b}_{m-\ell+1} = 0^{m-\ell}$. Therefore, we have $\beta = 0^{m-\ell+1}\mathtt{b}_{m-\ell+2}\cdots\mathtt{b}_m < 0^{m-\ell}1\mathtt{b}_{\ell-1}\cdots\mathtt{b}_1 = \beta^R$, hence (i) is satisfied. Suppose $j = 2$. If $\mathtt{b}_1 = 0$, then (i) is satisfied. Otherwise $\mathtt{b}_1 = 1$ and from (ii) $\mathtt{b}_2 = 0$. From (ii), we get that $\mathtt{b}_3\cdots\mathtt{b}_{\ell-1}10^{m-\ell} < 0^{m-\ell}\mathtt{b}_{\ell-1}\cdots\mathtt{b}_3$. This inequality implies that $\mathtt{b}_3\cdots\mathtt{b}_{m-\ell+2} = 0^{m-\ell}$. Therefore, we have $\beta 1 = 10^{m-\ell+1}\mathtt{b}_{m-\ell+3}\cdots\mathtt{b}_m 1 < 10^{m-\ell}1\mathtt{b}_{\ell-1}\cdots\mathtt{b}_1 = 1\beta^R$, hence (i) is satisfied. Thus, $\mathrm{last1}(\mathrm{last0}(\alpha))$ is an asymmetric bracelet. ◀

Proof of Theorem 4. Let α be an asymmetric bracelet in $\mathbf{A}(n) \setminus \{r_n\}$. We demonstrate that the parent rule par from (1) induces a path from α to r_n, i.e., there exists an integer j such that $\mathrm{par}^j(\alpha) = r_n$. Note that r_n is the lexicographically smallest asymmetric bracelet of order n. By Lemma 5, $\mathrm{par}(\alpha) \in \mathbf{A}(n)$. In the first two cases of the parent rule, $\mathrm{par}(\alpha)$ is lexicographically smaller than α. If the third case applies, let $\alpha = 0^s1\beta$. From Lemma 5, $\mathrm{last1}(\mathrm{last0}(\alpha))$ is an asymmetric bracelet. Thus, $\mathrm{par}(\mathrm{par}(\alpha))$ is either $\mathrm{first1}(\mathrm{last0}(\alpha))$ or $\mathrm{last1}(\mathrm{last0}(\alpha))$; in each case the resulting asymmetric bracelet has 0^{s+1} as a prefix and is therefore lexicographically smaller than α. Therefore, the parent rule induces a path from α to r_n.

3.1 A successor rule

Each application of the parent rule $\mathrm{par}(\alpha)$ in (1) corresponds to a conjugate pair. For instance, consider the asymmetric bracelet $\alpha = 00010111\mathbf{1}$. The parent of α is obtained by flipping the last 1 to obtain $00010111\mathbf{0}$ (see Figure 2). The corresponding conjugate pair is $(\mathbf{1}00010111, \mathbf{0}00010111)$. Let $\mathbf{C}(n)$ denote the set of all strings belonging to a conjugate pair in the cycle-joining tree $\mathbb{T}_n$. Then the following is a successor rule for an $\mathcal{OS}(n)$:

$$f(\alpha) = \begin{cases} \bar{\mathtt{a}}_1 & \text{if } \alpha \in \mathbf{C}(n); \\ \mathtt{a}_1 & \text{otherwise.} \end{cases}$$

For example, if $\mathbf{C}(9)$ corresponds to the conjugate pairs to create the cycle-joining tree $\mathbb{T}_9$ shown in Figure 2, then the corresponding universal cycle is:

$$0\underline{\mathbf{0}0001011111}00101101100101111001101111\underline{\mathbf{1}0001011}100101011100011011$$
$$10101101110000100111000100101000100110000101100100101100010101 1,$$

where the two underlined strings belong to the conjugate pair $(\mathbf{1}00010111, \mathbf{0}00010111)$. In general, this rule requires exponential space to store the set $\mathbf{C}(n)$. However, in some cases, it is possible to test whether a string is in $\mathbf{C}(n)$ without pre-computing and storing $\mathbf{C}(n)$. In our successor rule for an $\mathcal{OS}(n)$, we use Theorem 1 to avoid pre-computing and storing $\mathbf{C}(n)$, thereby reducing the space requirement from exponential in n to linear in n.

Successor-rule g to construct an $\mathcal{OS}(n)$ of length L_n

Let $\alpha = \mathtt{a}_1\mathtt{a}_2\cdots\mathtt{a}_n \in \mathbf{S}(n)$ and let

- $\beta_1 = 0^{n-i}\mathbf{1}\mathtt{a}_2\cdots\mathtt{a}_i$ where i is the largest index of α such that $\mathtt{a}_i = 1$ (first 1);
- $\beta_2 = \mathtt{a}_2\mathtt{a}_3\cdots\mathtt{a}_n\mathbf{1}$ (last 1);
- $\beta_3 = \mathtt{a}_j\mathtt{a}_{j+1}\cdots\mathtt{a}_n\mathbf{0}1^{j-2}$ where j is the smallest index of α such that $\mathtt{a}_j = 0$ and $j > 1$ (last 0).

Let

$$g(\alpha) = \begin{cases} \bar{\mathtt{a}}_1 & \text{if } \beta_1 \text{ and } \text{first1}(\beta_1) \text{ are in } \mathbf{A}(n); \\ \bar{\mathtt{a}}_1 & \text{if } \beta_2 \text{ and } \text{last1}(\beta_2) \text{ are in } \mathbf{A}(n), \text{ and } \text{first1}(\beta_2) \text{ is not in } \mathbf{A}(n); \\ \bar{\mathtt{a}}_1 & \text{if } \beta_3 \text{ and } \text{last0}(\beta_3) \text{ are in } \mathbf{A}(n), \text{ and neither } \text{first1}(\beta_3) \text{ nor } \text{last1}(\beta_3) \text{ are in } \mathbf{A}(n); \\ \mathtt{a}_1 & \text{otherwise.} \end{cases}$$

Starting with any string in $\alpha \in \mathbf{S}(n)$, we can repeatedly apply $g(\alpha)$ to obtain the next bit in a universal cycle for $\mathbf{S}(n)$.

▶ **Theorem 6.** *The function g is a successor rule that generates an $\mathcal{OS}(n)$ with length L_n for the set $\mathbf{S}(n)$ in $O(n)$-time per bit using $O(n)$ space.*

Proof. Consider $\alpha = \mathtt{a}_1\mathtt{a}_2 \cdots \mathtt{a}_n \in \mathbf{S}(n)$. If α belongs to some conjugate pair in $\mathbb{T}_n$, then it must satisfy one of three possibilities stepping through the parent rule in 1:

- Both β_1 and $\text{first1}(\beta_1)$ must be in $\mathbf{A}(n)$. Note, β_1 is a rotation of α when $\mathtt{a}_1 = 1$, where $\mathtt{a}_1$ corresponds to the first one in β_1.
- Both β_2 and $\text{last1}(\beta_2)$ must both be in $\mathbf{A}(n)$, but additionally, $\text{first1}(\beta_2)$ can not be in $\mathbf{A}(n)$. Note, β_2 is a rotation of α when $\mathtt{a}_1 = 1$, where $\mathtt{a}_1$ corresponds to the last one in β_2.
- Both β_3 and $\text{last0}(\beta_3)$ must both be in $\mathbf{A}(n)$, but additionally, both $\text{first1}(\beta_3)$ and $\text{last1}(\beta_3)$ can not be in $\mathbf{A}(n)$. Note, β_3 is a rotation of α when $\mathtt{a}_1 = 0$, where $\mathtt{a}_1$ corresponds to the last zero in β_3.

Thus, g is a successor rule on $\mathbf{S}(n)$ that generates a cycle of length $|\mathbf{S}(n)| = L_n$. By Theorem 1, one can determine whether a string is in $\mathbf{A}(n)$ in $O(n)$ time using $O(n)$ space. Since there are a constant number of tests required by each case of g, the corresponding $\mathcal{OS}(n)$ can be computed in $O(n)$-time per bit using $O(n)$ space.

◀

4 Extending orientable sequences

The values from the column labeled L_n^* in Table 1 were found by extending an $\mathcal{OS}(n)$ of length L_n constructed in the previous section. Given an $\mathcal{OS}(n)$, $\mathtt{o}_1 \cdots \mathtt{o}_m$, the following approaches were applied to find longer $\mathcal{OS}(n)$s for $n \leq 20$:

1. For each index i, apply a standard backtracking search to see whether $\mathtt{o}_i \cdots \mathtt{o}_m\mathtt{o}_1 \cdots \mathtt{o}_{i-1}$ can be extended to a longer $\mathcal{OS}(n)$. We followed several heuristics: (a) find a maximal length extension for a given i, and then attempt to extend starting from index $i+1$; (b) find a maximal length extension over all i, then repeat; (c) find the "first" possible extension for a given i, and then repeat for the next index $i+1$. In each case, we repeat until no extension can be found for any starting index. This approach was fairly successful for even n, but found shorter extensions for n odd. Steps (a) and (b) were only applied to n up to 14 before the depth of search became infeasible.
2. Refine the search in the previous step so the resulting $\mathcal{OS}(n)$ of length m' has an odd number of 1s and at most one substring 0^{n-4}. Then we can apply the recursive construction by Mitchell and Wild [20] to generate an $\mathcal{OS}(n+1)$ with length $2m'$ or $2m'+1$. Then, starting from the sequences generated by recursion, we again apply the exhaustive search to find minor extensions (the depth of recursion is significantly reduced). This approach found significantly longer extensions to obtain $\mathcal{OS}(n+1)$s when $n+1$ is odd.

5 Future research directions

We present the first efficient algorithm to construct orientable sequences with asymptotically optimal length; it is a successor-rule-based approach that requires $O(n)$ time per bit and uses $O(n)$ space. This answers a long-standing open question by Dai, Martin, Robshaw, and Wild [7]. The full version of this paper includes an application of the recent concatenation-tree framework [22] that leads to constructions of our $\mathcal{OS}(n)$s in $O(1)$-amortized time per bit. It also includes the results of applying our $\mathcal{OS}(n)$s to find some longer acyclic orientable sequences than reported in [5]. The binary results have recently been extended to arbitrary sized alphabets like $\{C, G, A, T\}$.

References

1 D. Adamson, V. V. Gusev, I. Potapov, and A. Deligkas. Ranking bracelets in polynomial time. In Paweł Gawrychowski and Tatiana Starikovskaya, editors, *32nd Annual Symposium on Combinatorial Pattern Matching (CPM 2021)*, volume 191 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CPM.2021.4.

2 A. Alhakim, C. J. Mitchell, J. Szmidt, and P. R. Wild. Orientable sequences over non-binary alphabets. *manuscript, 2023*.

3 K. S. Booth. Lexicographically least circular substrings. *Inform. Process. Lett.*, 10(4/5):240–242, 1980. doi:10.1016/0020-0190(80)90149-0.

4 S. Brlek, S. Hamel, M. Nivat, and C. Reutenauer. On the palindromic complexity of infinite words. *Internat. J. Found. Comp. Sci.*, 15(02):293–306, 2004. doi:10.1142/S012905410400242X.

5 J. Burns and C. J. Mitchell. Position sensing coding schemes. In *Cryptography and Coding III (M.J.Ganley, ed.)*, pages 31–66. Oxford University Press, 1993.

6 J. Currie and P. Lafrance. Avoidability index for binary patterns with reversal. *Electronic J. Combinatorics*, 23((1) P1.36):1–14, 2016. doi:10.37236/5483.

7 Z. D. Dai, K. M. Martin, M. J. B. Robshaw, and P. R. Wild. Orientable sequences. In *Cryptography and Coding III (M.J.Ganley, ed.)*, pages 97–115. Oxford University Press, 1993.

8 T. Etzion. An algorithm for generating shift-register cycles. *Theoret. Comput. Sci.*, 44(2):209–224, 1986. doi:10.1016/0304-3975(86)90118-0.

9 T. Etzion. Self-dual sequences. *J. Combin. Theory Ser. A*, 44(2):288–298, 1987. doi:10.1016/0097-3165(87)90035-5.

10 T. Etzion and A. Lempel. Algorithms for the generation of full-length shift-register sequences. *IEEE Trans. Inform. Theory*, 30(3):480–484, 1984. doi:10.1109/TIT.1984.1056919.

11 L. Fleischer and J. O. Shallit. Words that avoid reversed factors, revisited. Arxiv preprint arXiv:1911.11704 [cs.FL], available at http://arxiv.org/abs/1911.11704, 2019.

12 H. Fredricksen. A survey of full length nonlinear shift register cycle algorithms. *SIAM Review*, 24(2):195–221, 1982. doi:10.1137/1024041.

13 D. Gabrić, J. Sawada, A. Williams, and D. Wong. A framework for constructing de Bruijn sequences via simple successor rules. *Discrete Math.*, 241(11):2977–2987, 2018. doi:10.1016/j.disc.2018.07.010.

14 D. Gabrić, J. Sawada, A. Williams, and D. Wong. A successor rule framework for constructing k-ary de Bruijn sequences and universal cycles. *IEEE Trans. Inform. Theory*, 66(1):679–687, 2020. doi:10.1109/TIT.2019.2928292.

15 C. Hierholzer. Ueber die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren. *Math. Annalen*, 6:30–32, 1873. doi:10.1007/BF01442866.

16 Y. Huang. A new algorithm for the generation of binary de Bruijn sequences. *J. Algorithms*, 11(1):44–51, 1990. doi:10.1016/0196-6774(90)90028-D.

17 C. J. A. Jansen, W. G. Franx, and D. E. Boekee. An efficient algorithm for the generation of DeBruijn cycles. *IEEE Trans. Inform. Theory*, 37(5):1475–1478, 1991. doi:10.1109/18.133272.

18 A. Lempel. On a homomorphism of the de Bruijn graph and its applications to the design of feedback shift registers. *IEEE Trans. Comput.*, C-19(12):1204–1209, 1970. doi:10.1109/T-C.1970.222859.

19 R. Mercaş. On the aperiodic avoidability of binary patterns with variables and reversals. *Theoret. Comput. Sci.*, 682:180–189, 2017. doi:10.1016/j.tcs.2016.12.022.

20 C. J. Mitchell and P. R. Wild. Constructing orientable sequences. *IEEE Trans. Inform. Theory*, 68(7):4782–4789, 2022. doi:10.1109/TIT.2022.3158645.

21 N. Rampersad and J. O. Shallit. Words that avoid reversed subwords. *J. Combin. Math. Combin. Comput.*, 54:157–164, 2005.

22 J. Sawada, J. Sears, A. Trautrim, and A. Williams. Concatenation trees: A framework for efficient universal cycle and de Bruijn sequence constructions. Arxiv preprint arXiv:2308.12405 [math.CO], available at https://arxiv.org/abs/2308.12405, 2023.

23 J. Sawada and A. Williams. Constructing the first (and coolest) fixed-content universal cycle. *Algorithmica*, 85(6):1754–1785, 2023. doi:10.1007/s00453-022-01047-2.

24 J. Sawada and D. Wong. Efficient universal cycle constructions for weak orders. *Discrete Math.*, 343(10):112022, 2020. doi:10.1016/j.disc.2020.112022.

25 N. J. A. Sloane et al. OEIS Foundation Inc. (2024), The On-Line Encyclopedia of Integer Sequences, https://oeis.org.