Efficient Universal Cycle Constructions for Weak Orders

Joe Sawada   Dennis Wong

Abstract

A weak order is a way n competitors can rank in an event, where ties are allowed. A weak order can also be thought of as a relation on $\{1, 2, \ldots, n\}$ that is transitive and complete. We provide the first efficient algorithms to construct universal cycles for weak orders by considering both rank and height representations. Each algorithm constructs the universal cycles in $O(n)$ time per symbol using $O(n)$ space.

1 Introduction

An ordering of how n competitors can rank in an event, where ties are allowed, is known as a *weak order*. As an example, the times for the 100m men's butterfly final in the 2016 Summer Olympics were:

Lane	Name	Country	Time	Rank
1	Sadovnikov	RUS	51.84	8
2	Phelps	USA	**51.14**	2
3	Li	CHN	51.26	5
4	Schooling	SGP	50.39	1
5	Le Clos	RSA	**51.14**	2
6	Cseh	HUN	**51.14**	2
7	Shields	USA	51.73	7
8	Metella	FRA	51.58	6

The result was a three way tie for the silver medal. No bronze was awarded. This outcome corresponds to the weak ordering 82512276 that represents the *rank* of each competitor. Let $\mathbf{W_r}(n)$ denote the set of weak orders in a competition with n competitors (teams) under this rank representation. For example, when $n = 3$, the 13 different weak orders are

$$\mathbf{W_r}(3) = \{111, 113, 131, 311, 122, 212, 221, 123, 132, 213, 231, 312, 321\}.$$

The number of weak orders of order n are also known as the the ordered Bell numbers or Fubini numbers and their enumeration sequence is A000670 in the Online Encyclopedia of Integer Sequences [9]. The first six terms in this sequence starting at $n = 1$ are 1, 3, 13, 75, 541, and 4683 respectively.

Given a set of strings $\mathbf{S}$ of length n, a *universal cycle* for $\mathbf{S}$ is a sequence of length $|\mathbf{S}|$ that when considered cyclicly contains each string in $\mathbf{S}$ as a substring. Note this definition implies that each string in $\mathbf{S}$ will appear as a substring exactly once. As an example, a universal cycle for $\mathbf{W_r}(3)$ is 1113212213123. The existence of universal cycles for $\mathbf{W_r}(n)$ was proved by Leitner and Godbole [8] using the terminology *ranked permutations*. Using a height-based representation for weak orders defined later in this section, Diaconis and Graham [2] discuss the existence of universal cycles using the terminology *permutations with ties*; subsequently, Horan and Hurlbert [5] prove their existence using standard graph techniques, extending their results to related objects called s-overlap cycles. However, the more difficult problem of efficiently constructing universal cycles for weak orders, which was posed by Diaconis and Ruskey in Problem 477 of [6], remained open. In recent work, Jacques and Wong [7] proposed greedy constructions, but they require exponential space.

In this paper we present the first efficient universal cycle constructions for weak orders by considering both rank and height representations, thus answering the open problem described above. Our algorithms apply the k-ary universal cycle framework developed in [4], which generalizes a binary framework in [3], to construct the universal cycles using $O(n)$ time per symbol and $O(n)$ space. Implementations of our algorithms in C are available for download at `http://debruijnsequence.org`.

1.1 Representations for weak orders

A *weak order* can be thought of as a binary relation $\preceq$ on $\Sigma = \{1, 2, \ldots, n\}$ that is transitive and complete (or connex). The latter property meaning that $x \preceq y$ or $y \preceq x$ (or both) for each $x, y \in \Sigma$. We write $x \equiv y$ if $x \preceq y$ and $y \preceq x$, and we write $x \prec y$ if $x \preceq y$ but $y \npreceq x$. Using this notation, a weak order can be written as a permutation where each element is separated by either $\equiv$ or $\prec$. For example

$$4 \prec 2 \equiv 5 \equiv 6 \prec 3 \prec 8 \prec 7 \prec 1$$

corresponds to the weak ordering from our earlier Summer Olympics example. We will use this ordering to formally define our two representations for weak orders.

The *height* of element j is the number of $\prec$ symbols that precede j in the weak order. By replacing each element j by its height, the weak order $4 \prec 2 \equiv 5 \equiv 6 \prec 3 \prec 8 \prec 7 \prec 1$ can be represented by 51201143. Let $\mathbf{W_h}(n)$ denote the set of all weak orders of order n using this height representation. This is the representation used in [2, 5, 6]. As an example,

$$\mathbf{W_h}(3) = \{000, 001, 010, 100, 011, 101, 110, 012, 120, 201, 021, 210, 102\}.$$

The *rank* of element j is one plus the number of elements that precede the rightmost $\prec$ symbol to the left of j in the weak order. By replacing each element j by its rank, as done in [8], the weak order $4 \prec 2 \equiv 5 \equiv 6 \prec 3 \prec 8 \prec 7 \prec 1$ can be represented by 82512276. This rank-based representation is equivalent to the strings we described to define our set $\mathbf{W_r}(n)$.

Remark 1.1 *The sets $\mathbf{W_r}(n)$ and $\mathbf{W_h}(n)$ are both closed under string rotation.*

2 Universal cycle construction framework

In this section we recall elements from the framework developed in [4] that are applied to develop efficient universal cycle constructions for weak orders. We begin by introducing the most general results that applied in our universal cycle construction for $\mathbf{W_h}(n)$. Then we present a simplified special case that is applied in the construction of a universal cycle for $\mathbf{W_r}(n)$.

Let Σ denote a finite alphabet $\{1, 2, \ldots, k\}$ and assume that $n, k \geq 2$. A function $f : \Sigma^n \to \Sigma$ is said to be a *feedback function*. A feedback function f is a *UC-successor* of $\mathbf{S}$, a subset of Σ^n, if there exists a universal cycle U for $\mathbf{S}$ such that each string $\omega \in \mathbf{S}$ is followed by the symbol $f(\omega)$ in U. A partition of $\mathbf{S}$ into subsets $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$ is a *UC-partition with respect to f* if f is a UC-successor for each $\mathbf{S}_i$ where $i \in \{1, 2, \ldots, m\}$.

Example 1 Consider the feedback function f defined by $f(w_1 w_2 \cdots w_n) = w_1$. Observe that f is a UC-successor for each equivalence class of strings under rotation. Thus,

$$\mathbf{S}_1 = \{\mathbf{111}\}, \mathbf{S}_2 = \{\mathbf{113}, 131, 311\}, \mathbf{S}_3 = \{\mathbf{122}, 221, 212\}, \mathbf{S}_4 = \{\mathbf{123}, 231, 312\}, \mathbf{S}_5 = \{\mathbf{132}, 321, 213\}$$

is a UC-partition of $\mathbf{W_r}(3)$ with respect to f.

Definition 2.1 *Let $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$ be an ordered partition of $\mathbf{S}$. For $2 \leq i \leq m$, let $x_i, y_i, z_i \in \Sigma$ and let $\beta_i \in \Sigma^{n-1}$. A sequence of tuples $(\beta_2, x_2, y_2, z_2), (\beta_3, x_3, y_3, z_3), \ldots, (\beta_m, x_m, y_m, z_m)$ is a spanning sequence of the partition if for each (β_i, x_i, y_i, z_i):*

(i) $y_i\beta_i \in \mathbf{S}_i$,

(ii) *if $i = \mathit{first}(\beta_i)$ then $x_i\beta_i \in \mathbf{S}_j$ for some $j < i$,*

(iii) *$x_iy_iz_i$ is a substring of the cyclic string created by starting with $x_{first(\beta_i)}$ then appending each y_j from tuples (β_j, x_j, y_j, z_j) where $\beta_j = \beta_i$ in increasing order of index j,*

where $\mathit{first}(\beta_i)$ is the smallest index of a tuple containing β_i.

The two main results from [4] are Theorem 2.8 and Theorem 2.9. They apply the above definitions to give a framework for constructing universal cycles. We combine them in the following theorem.

Theorem 2.2 [4] *Let $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$ be a UC-partition of $\mathbf{S}$ with respect to f with spanning sequence*

$$(\beta_2, x_2, y_2, z_2), (\beta_3, x_3, y_3, z_3), \ldots, (\beta_m, x_m, y_m, z_m)$$

for some $m \geq 2$. Then the following feedback functions g and g' are UC-successors for $\mathbf{S}$:

$$g(\omega) \ = \begin{cases} f(y_i\beta_i) & \text{if } \omega = x_i\beta_i \text{ for some } i \in \{2,3,\ldots,m\} \text{ and } i = first(\beta_i); \\ f(z_i\beta_i) & \text{if } \omega = y_i\beta_i \text{ for some } i \in \{2,3,\ldots,m\}; \\ f(\omega) & \text{otherwise,} \end{cases}$$

$$g'(\omega) = \begin{cases} f(x_i\beta_i) & \text{if } \omega = y_i\beta_i \text{ for some } i \in \{2,3,\ldots,m\} \text{ and } i = first(\beta_i); \\ f(y_i\beta_i) & \text{if } \omega = z_i\beta_i \text{ for some } i \in \{2,3,\ldots,m\}; \\ f(\omega) & \text{otherwise.} \end{cases}$$

Next we present a special case for the definition of a spanning sequence and the above theorem when each β_i is unique. This simplified result will be used to develop a universal cycle for $\mathbf{W_r}(n)$ in the next section. The more general result will be used for $\mathbf{W_h}(n)$.

Definition 2.3 *Let $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$ be an ordered partition of $\mathbf{S}$. For $2 \leq i \leq m$, let $x_i, y_i \in \Sigma$ and let $\beta_i \in \Sigma^{n-1}$. A sequence of tuples $(\beta_2, x_2, y_2), (\beta_3, x_3, y_3), \ldots, (\beta_m, x_m, y_m)$ is a simplified spanning sequence of the partition if each β_i is unique and for each i the string $y_i\beta_i \in \mathbf{S}_i$ implies the string $x_i\beta_i \in \mathbf{S}_j$ for some $j < i$.*

Theorem 2.4 [4] *Let $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$ be a UC-partition of $\mathbf{S}$ with respect to f with simplified spanning sequence $(\beta_2, x_2, y_2), (\beta_3, x_3, y_3), \ldots, (\beta_m, x_m, y_m)$ for some $m \geq 2$. Then the following feedback function $g(\omega)$ is a UC-successor for $\mathbf{S}$:*

$$g(\omega) = \begin{cases} f(x_i\beta_i) & \text{if } \omega = y_i\beta_i \text{ for some } i \in \{2,3,\ldots,m\}; \\ f(y_i\beta_i) & \text{if } \omega = x_i\beta_i \text{ for some } i \in \{2,3,\ldots,m\}; \\ f(\omega) & \text{otherwise.} \end{cases}$$

In the above theorem, the modifications of f to get g correspond to repeatedly applying a standard cycle joining technique. The way the cycles are joined are directed by the simplified spanning sequence. Such a cycle joining (gluing) approach has been exploited in many de Bruijn sequence constructions (see [3, 4]).

3 A universal cycle construction for $\mathbf{W_r}(n)$

In this section we apply Theorem 2.4 to develop a UC-successor for $\mathbf{W_r}(n)$. Let $\mathbf{W'_r}(n)$ be the subset of all weak orders in $\mathbf{W_r}(n)$ that have no repeating symbol except for possibly the symbol 1. For example, $\mathbf{W'_r}(3) = \mathbf{W_r}(3) - \{122, 212, 221\}$. Additionally, let $num_\omega(v)$ denote the number of occurrences of symbol v in the string ω.

Let $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$ be a UC-partition of $\mathbf{W_r}(n)$ with respect to $f(w_1 w_2 \cdots w_n) = w_1$. Let the lexicographically smallest representatives for each part be given by $\alpha_1, \alpha_2, \ldots, \alpha_m$ respectively. Let $\mathbf{R_r}(n) = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$. Consider the partition to be ordered first by the number of 1s in the representatives (smallest to largest), and then by reverse lexicographic order of the representatives. Note, that $\mathbf{S}_1 = \{1^n\}$ for all n. Define a sequence $\mathcal{S}_r = (\beta_2, x_2, y_2), (\beta_3, x_3, y_3), \ldots, (\beta_m, x_m, y_m)$ for this ordered partition where each (β_i, x_i, y_i) is defined as follows assuming $\alpha_i = a_1 a_2 \cdots a_n$:

$$(\beta_i, x_i, y_i) = \begin{cases} (a_{j+1} \cdots a_n a_1 \cdots a_{j-1},\ 1,\ a_j) & \text{if } \alpha_i \in \mathbf{W'_r}(n); \\ (a_{k+1} \cdots a_n a_1 \cdots a_{k-1},\ a_k + num_{\alpha_i}(a_k) - 1,\ a_k) & \text{otherwise,} \end{cases}$$

where j is the index of the unique symbol $(num_{\alpha_i}(1) + 1)$, and k is the largest index[1] such that $a_k \neq 1$ and $num_{\alpha_i}(a_k) > 1$.

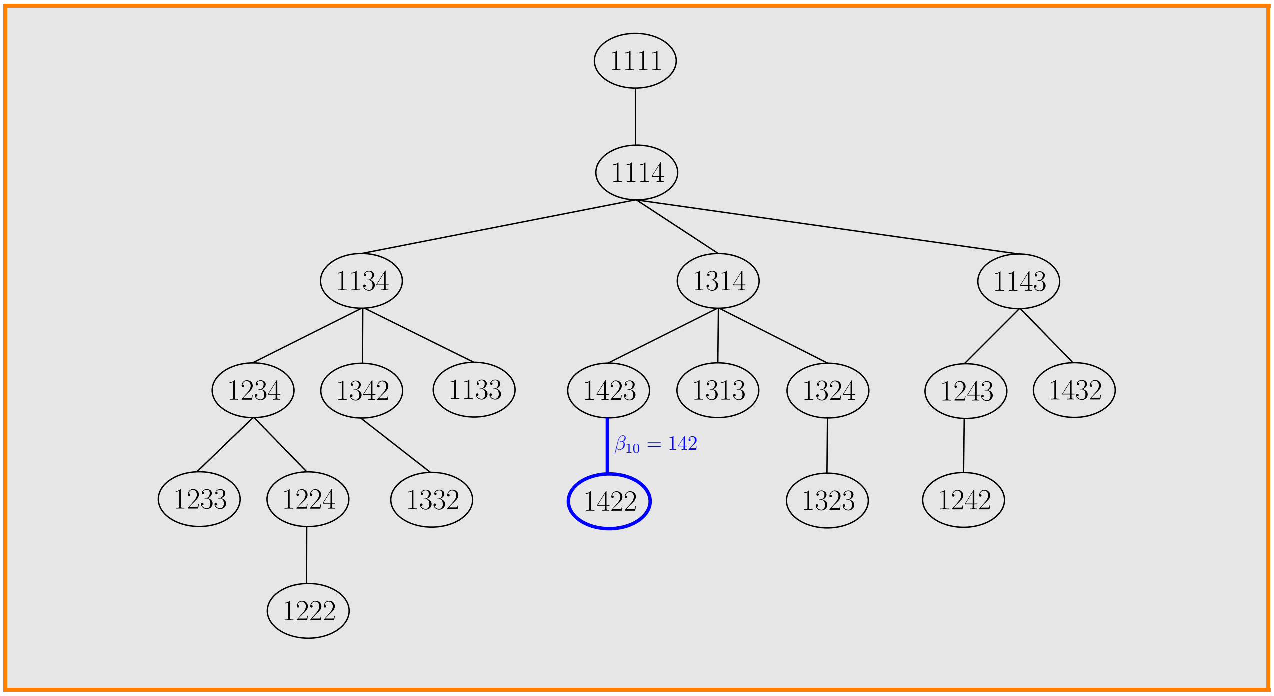
Example 2 Consider the UC-partition $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_{20}$ of $\mathbf{W_r}(4)$ with respect to $f(w_1 w_2 \cdots w_n) = w_1$ ordered first by the number of 1s in the representatives, then in reverse lexicographic order. The following illustrates this ordered partition by its representatives $\alpha_1, \alpha_2, \ldots, \alpha_{20}$ along with the sequence $\mathcal{S}_r$ defined above.

i	α_i	(β_i, x_i, y_i)
1	1111	-
2	1114	(111,1,4)
3	1314	(141,1,3)
4	1313	(131,4,3)
5	1143	(114,1,3)
6	1134	(411,1,3)
7	1133	(113,4,3)
8	1432	(143,1,2)
9	1423	(314,1,2)
10	1422	(142,3,2)

i	α_i	(β_i, x_i, y_i)
11	1342	(134,1,2)
12	1332	(213,4,3)
13	1324	(413,1,2)
14	1323	(132,4,3)
15	1243	(431,1,2)
16	1242	(124,3,2)
17	1234	(341,1,2)
18	1233	(123,4,3)
19	1224	(412,3,2)
20	1222	(122,4,2)

The sequence $\mathcal{S}_r$ induces the following tree where the nodes are the representatives α_i of each $\mathbf{S}_i$. Observe that the parent of the representative of each $y_i\beta_i$ is the representative for $x_i\beta_i$. Also if α_j is the parent of α_i, then $j < i$. For instance for $\alpha_{10} = 1422$ we have $\beta_{10} = 142$ and its parent is $\alpha_9 = 1423$ which is a rotation of $x_{10}\beta_{10} = 3142$.

[1] The smallest index (among other possible choices for k) also works, producing an alternate simplified spanning sequence



Lemma 3.1 *$\mathcal{S}_r$ is a simplified spanning sequence of the ordered partition $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_m$.*

Proof. Based on the definition of a simplified spanning sequence we must show three things about $\mathcal{S}_r$: (1) each β_i is unique, (2) each $y_i\beta_i \in \mathbf{S}_i$, and (3) each $x_i\beta_i \in \mathbf{S}_j$ for some $j < i$. Consider (β_i, x_i, y_i) for some $2 \leq i \leq m$. Since $y_i\beta_i$ is a rotation of α_i it is in $\mathbf{S}_i$, thus satisfying (2). Observe that the definitions of the indices j and k imply that $y_i > 1$. Furthermore, by the definition of these indices, observe that $x_i\beta_i$ is in $\mathbf{W_r}(n)$, and its corresponding representative α_j either has more 1s or is lexicographically larger than α_i. Thus $x_i\beta_i$ is in some $\mathbf{S}_j$ where $j < i$, thus satisfying (3). Finally, to demonstrate (1), suppose there exists $j \neq i$ such that $\beta_j = \beta_i$. Since $\alpha_i \neq \alpha_j$, this means $y_j \neq y_i$. If y_i is not found in β_i then $num_{\alpha_i}(y_i) = 1$, α_i must be in $\mathbf{W'_r}(n)$, and $y_i = num_{\alpha_i}(1) + 1$. By the definition of a weak order, the only other possible value for y_j such that $y_j\beta_j$ is in $\mathbf{W_r}(n)$ is $y_j = 1$. However this contradicts our earlier claim that all $y_j > 1$. Otherwise, assume y_i appears in β_i which means $num_{\alpha_i}(y_i) > 1$. Because $\alpha_i \in \mathbf{W_r}(n)$ there is no symbol $y_i + num_{\alpha_i}(y_i) - 1$ in α_i. Thus since $y_j\beta_j$ is in $\mathbf{W_r}(n)$, either $y_j = y_i$ (a contradiction) or $y_j = y_i + num_{\alpha_i}(y_i) - 1$. But from the previous argument since $y_i + num_{\alpha_i}(y_i) - 1$ would have to be unique in α_j, which was just ruled out in the first case. Thus each β_i is unique, satisfying (1). $\square$

Using the simplified spanning sequence $\mathcal{S}_r$, illustrated in Example 2, we can immediately apply Theorem 2.4 to define a UC-successor g_r for $\mathbf{W_r}(n)$. However, storing the simplified spanning sequence will require an exponential amount of memory, and hence the UC-successor will not be efficient. Thus we need to determine when a weak order $\omega = w_1w_2\cdots w_n \in \mathbf{W_r}(n)$ belongs to the set $\mathbf{U} = \{x_i\beta_i \mid i \in \{2,3,\ldots,m\}\} \cup \{y_i\beta_i \mid i \in \{2,3,\ldots,m\}\}$. We consider the following four cases noting that $\mathbf{X}_1 \cup \mathbf{Y}_1 \cup \mathbf{X}_2 \cup \mathbf{Y}_2 = \mathbf{U}$:

- $\mathbf{X}_1 = \{x_i\beta_i \mid \alpha_i \in \mathbf{W'_r}(n)\}$,
- $\mathbf{Y}_1 = \{y_i\beta_i \mid \alpha_i \in \mathbf{W'_r}(n)\}$,
- $\mathbf{X}_2 = \{x_i\beta_i \mid \alpha_i \notin \mathbf{W'_r}(n)\}$,
- $\mathbf{Y}_2 = \{y_i\beta_i \mid \alpha_i \notin \mathbf{W'_r}(n)\}$.

Algorithm 1 Pseudocode for the UC-successor $g_r(\omega)$ where $\omega = w_1w_2\cdots w_n$ for the set $\mathbf{W_r}(n)$.

function $g_r(\omega)$

  if $\omega \in \mathbf{W}'_\mathbf{r}(n)$ **and** $w_1 = num_\omega(1) + 1$ **then return** 1 $\triangleright\ \omega \in \mathbf{Y}_1$

  if $\omega \in \mathbf{W}'_\mathbf{r}(n)$ **and** $w_1 = 1$ **then return** $num_\omega(1)$ $\triangleright\ \omega \in \mathbf{X}_1$

  $\triangleright\ \omega \in \mathbf{Y}_2$
  if $\omega \notin \mathbf{W}'_\mathbf{r}(n)$ **and** $num_\omega(w_1) > 1$ **and** $w_1 > 1$ **then**
    if $num_\omega(w_i) = 1$ **or** $w_i = 1$ **then**
      for all $2 \le i <$ RepStart(ω) **return** $w_1 + num_\omega(w_1) - 1$

  $\triangleright\ \omega \in \mathbf{X}_2$
  if $num_\omega(w_1) = 1$ **and** $w_1 > 1$ **then**
    $p \leftarrow$ the largest symbol in ω less than w_1
    if $p \ne 1$ **then**
      if $w_i \ne p$ **and** $(num_\omega(w_i) = 1$ **or** $w_i = 1)$ **then**
        for all $2 \le i <$ RepStart($pw_2w_3\cdots w_n$) **return** p

  return w_1

We start by examining $\mathbf{X}_1$ and $\mathbf{Y}_1$. From our definition of $\mathcal{S}_r$, a string ω will belong to $\mathbf{Y}_1$ if and only if it is in $\mathbf{W}'_\mathbf{r}(n)$ (since ω is a rotation of α_i) and $w_1 = num_\omega(1) + 1$. For each such ω, by replacing w_1 with a 1 yields a string ω' in $\mathbf{X}_1$. Since ω' will also belong to $\mathbf{W}'_\mathbf{r}(n)$ it will belong to $\mathbf{X}_1$ if and only if it is in $\mathbf{W}'_\mathbf{r}(n)$ and $w_1 = 1$. Testing for membership in $\mathbf{X}_2$ and $\mathbf{Y}_2$ is a bit more complicated. Let t denote the index of ω such that $w_t \cdots w_n w_1 \cdots w_{t-1}$ is the representative of ω. Let the function RepStart(ω) return the index t such that $w_t \cdots w_n w_1 \cdots w_{t-1}$ is the representative of ω. For example RepStart(82512276) $= 4$. Now from our definition of $\mathcal{S}_r$, ω will belong to $\mathbf{Y}_2$ if it is not in $\mathbf{W}'_\mathbf{r}(n)$, $w_1 \ne 1$, and $num_\omega(w_1) > 1$, and every symbol in $w_2w_3\cdots w_{t-1}$ is equal to 1 or appears exactly once in ω. For each such ω, replacing w_1 with $w'_1 = w_1 + num_\omega(w_1) - 1$ will yield a string ω' in $\mathbf{X}_2$. Note that since ω is a weak order, $num_{\omega'}(w'_1) = 1$ and $w'_1 > 1$. Furthermore, it is possible that ω' is in $\mathbf{W}'_\mathbf{r}(n)$ even though ω (which is a rotation of some α_i) is not. Given the former two constraints ω' will be in $\mathbf{X}_2$ if and only if every symbol in $w_2w_3\cdots w_{t-1}$ is not equal to w_1 and is either equal to 1 or appears exactly once in ω.

Using the membership conditions for the sets $\mathbf{X}_1, \mathbf{Y}_1, \mathbf{X}_2$ and $\mathbf{Y}_2$ outlined above, Algorithm 1 provides pseudocode for a UC-successor $g_r(\omega)$ for $\mathbf{W_r}(n)$ by applying Theorem 2.4.

Theorem 3.2 *The function $g_r : \Sigma^n \to \Sigma$ presented in Algorithm 1 is a UC-successor for $\mathbf{W_r}(n)$.*

3.1 Implementation and analysis

Pseudocode in Algorithm 2 applies the UC-successor $g_r(\omega)$ given in Algorithm 1 to construct a universal cycle for $\mathbf{W_r}(n)$ starting with the weak order 1^n. The algorithm initializes ω to 1^n and initializes the values $num_\omega(v)$ for each symbol v from 1 to n. Each iteration of the repeat loop prints w_1, calls the function $g_r(\omega)$, then updates ω and the num values. The loop terminates when ω returns to the initial string 1^n which occurs when $num_\omega(1) = 1$. This requires a constant amount of work for each call to $g_r(\omega)$. Note that we can start from any initial weak order, but to test for that string in the termination condition would require $O(n)$ time. In the function $g_r(\omega)$, given in Algorithm 1, testing whether or not a string belongs to $\mathbf{W}'_\mathbf{r}(n)$ can easily be done in $O(n)$ time. Also, it is well known that the function RepStart($w_1w_2\cdots w_n$) can be computed in $O(n)$ time [1] and each iteration of the two loops in the function $g_r(\omega)$ requires only a constant amount of time. Thus, the function $g_r(\omega)$ runs in $O(n)$ time.

Algorithm 2 Applying the UC-successor $g_r(\omega)$ to construct a universal cycle for $\mathbf{W_r}(n)$.

procedure $UC(\ n)$
    $\omega = w_1w_2\cdots w_n \leftarrow 1^n$
    for i **from** 2 **to** n **do** $num_\omega(i) \leftarrow 0$
    $num_\omega(1) \leftarrow n$
    repeat
        $\text{PRINT}(w_1)$
        $v \leftarrow g(\omega)$
        $num_\omega(w_1) \leftarrow num_\omega(w_1) - 1$
        $num_\omega(v) \leftarrow num_\omega(v) + 1$
        $\omega \leftarrow w_2w_3\cdots w_nv$
    until $num_\omega(1) = n$

Theorem 3.3 *A universal cycle for $\mathbf{W_r}(n)$ can be constructed using the successor $g_r(\omega)$ presented in Algorithm 1 starting from any initial weak order ω in $O(n)$ time per symbol using $O(n)$ space.*

The following are the universal cycles for $\mathbf{W_r}(n)$ generated by Algorithm 2 for $n = 3$ and $n = 4$:

▷ $n = 3$: 1113213122123;
▷ $n = 4$: 111143214312421243114132313241313142214231411331134213321341222122412331234.

4 A universal cycle construction for $\mathbf{W_h}(n)$

In this section we consider the height representation for weak orders and $\mathbf{W_h}(n)$. For $n = 3$, the feedback function $f(w_1w_2\cdots w_n) = w_1$ partitions $\mathbf{W_h}(n)$ into 5 sets with the lexicographically largest element from each set being 210, 201, 110, 100, and 000 respectively. In order to define a simplified spanning sequence, we must define four unique β_i. However there are only a total of three possible values for β_i, namely $\{00, 01, 10\}$. Thus in order to develop UC-successors for $\mathbf{W_h}(n)$ using this feedback function, we define an appropriate spanning sequence and then apply Theorem 2.2.

Let $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_m$ be a UC-partition of $\mathbf{W_h}(n)$ with respect to $f(w_1w_2\cdots w_n) = w_1$. Let the lexicographically largest representatives for each part be given by $\alpha_1, \alpha_2, \ldots, \alpha_m$ respectively. Let $\mathbf{R_h}(n) = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$. Consider the partition to be ordered in lexicographic order with respect to their representatives α_i. Thus, $\mathbf{T}_1 = \{0^n\}$ for all n. Assuming $i > i$, let $\alpha_i = a_1a_2\cdots a_n$, $\alpha_i^- = (a_1{-}1)a_2a_3\cdots a_n$, and $\alpha_i^+ = (a_1{+}1)a_2a_3\cdots a_n$. We define a sequence $\mathcal{S}_h = (\beta_2, x_2, y_2), (\beta_3, x_3, y_3), \ldots, (\beta_m, x_m, y_m)$ for this ordered partition where each (β_i, x_i, y_i, z_i) is defined as follows:

▷ $\beta_i = a_2a_3\cdots a_n$,
▷ $x_i = a_i - 1$,
▷ $y_i = a_1$,
▷ $z_i = \begin{cases} a_1 + 1 & \alpha_i^+ \in \mathbf{R_h}(n) \\ a_1 - 2 & \alpha_i^- \in \mathbf{R_h}(n) \\ a_1 - 1 & \text{otherwise.} \end{cases}$

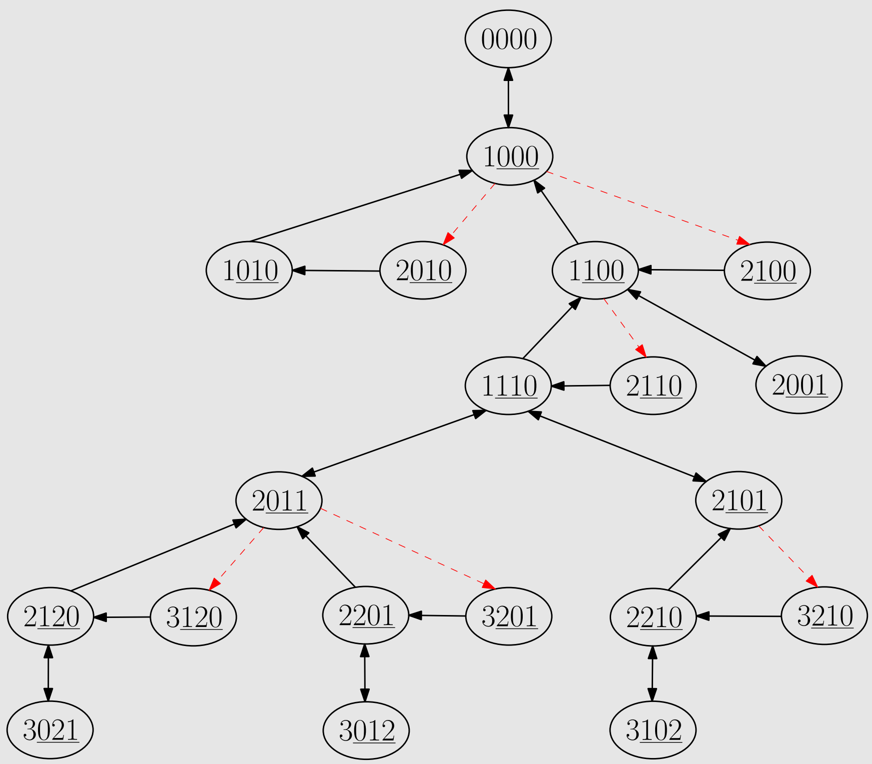
When proving that $\mathcal{S}_h$ is a spanning sequence in the next lemma, we show that it is not possible for both α_i^+ and α_i^- to be in $\mathbf{R_h}(n)$. Thus, the definition of z_i is well-defined.

Example 3 Consider the UC-partition $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_{20}$ of $\mathbf{W_h}(4)$ with respect to $f(w_1w_2\cdots w_n) = w_1$ where the sets are listed in lexicographic order with respect to their unique representatives in $\mathbf{R_h}(n)$. The following table illustrates this ordered partition by its representatives $\alpha_1, \alpha_2, \ldots, \alpha_{20}$ along with its corresponding sequence $\mathcal{S}_h$.

i	α_i	(β_i, x_i, y_i, z_i)
1	0000	-
2	1000	(000, 0, 1, 0)
3	1010	(010, 0, 1, 2)
4	1100	(100, 0, 1, 2)
5	1110	(110, 0, 1, 2)
6	2001	(001, 1, 2, 1)
7	2011	(011, 1, 2, 1)
8	2010	(010, 1, 2, 0)
9	2100	(100, 1, 2, 0)
10	2101	(101, 1, 2, 1)

i	α_i	(β_i, x_i, y_i, z_i)
11	2110	(110, 1, 2, 0)
12	2120	(120, 1, 2, 3)
13	2201	(201, 1, 2, 3)
14	2210	(210, 1, 2, 3)
15	3012	(012, 2, 3, 2)
16	3021	(021, 2, 3, 2)
17	3102	(102, 2, 3, 2)
18	3120	(120, 2, 3, 1)
19	3201	(201, 2, 3, 1)
20	3210	(210, 2, 3, 1)

The partition and sequence $\mathcal{S}_h$ is illustrated by the following graph. The nodes are the representatives α_i of each $\mathbf{T}_i$ where each β_i is underlined. Consider two nodes α_i and α_j where $j < i$. If β_i is unique, then there is a bi-directional edge (α_i, α_j) if $x_i\beta_i \in \mathbf{T}_j$. If β_i is not unique, there is a uni-directional edge (α_i, α_j) if $x_i\beta_i \in \mathbf{T}_j$ and a dashed (red) uni-directional edge (α_j, α_i) if $z_i\beta_i \in \mathbf{T}_j$.



Lemma 4.1 *$\mathcal{S}_h$ is a spanning sequence of the ordered partition $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_m$.*

Proof. We demonstrate that $\mathcal{S}_h$ satisfies the three conditions of its definition. Consider $1 < i \leq m$. By definition $y_i\beta_i = \alpha_i$ and thus $y_i\beta_i \in \mathbf{T}_i$, satisfying condition (i). Since $\alpha_i \in \mathbf{W_h}(n)$, every symbol from 0 to a_1 appears in α_i. Thus, $x_i\beta_i = (a_1 - 1)\beta_i$ is a valid weak order and moreover, it must belong to some set $\mathbf{T}_j$ where $j < i$ by the ordering of the sets. Thus condition (ii) is satisfied. For condition (iii) we consider the following two cases:

(a) If α_i is the only string in $\mathbf{R_h}(n)$ with the suffix β_i then $x_i = z_i = a_1 - 1$. Thus, the cyclic string described in (iii) is simply $x_i y_i$ and clearly $x_i y_i z_i$ is a substring of this string when the former is considered cyclicly.

(b) If α_i is not the only string in $\mathbf{R_h}(n)$ with the suffix β_i, then we show that there can be at most one more string with the suffix β_i. Suppose a_1 is the uniquely largest element in α_i. Then any string $(a_1+t)\beta_i$ for $t > 0$ will not be in $\mathbf{W_h}(n)$. Moreover, any string $(a_1 - t)\beta_i$ for $t > 1$ will not be the lexicographically largest amongst all its rotations and hence is not in $\mathbf{R_h}(n)$. Thus it must be that $\alpha_i^- \in \mathbf{R_h}(n)$. For this case $x_i = a_1 - 1, y_i = a_1$ and $z_i = a_1 - 2$. The representative for α_i^- must be for some set $\mathbf{T}_j$ where $j < i$ and by definition $x_j = a_1 - 2$, $y_j = a_1 - 1$ and $z_j = a_1$. Furthermore, j will be the smallest index of a tuple containing β_i. Thus, the cyclic string described in (iii) is $(a_1{-}2)(a_1{-}1)(a_1)$ and $x_i y_i z_i$ is a substring of this string when the former is considered cyclicly, satisfying condition (iii).

Otherwise, it must be that a_1 is one of the largest but not the uniquely largest element in α_i. In this case, any string $(a_1 + t)\beta_i$ for $t > 1$ will not be in $\mathbf{W_h}(n)$. Moreover, any string $(a_1 - t)\beta_i$ for $t > 0$ will not be the lexicographically largest amongst all its rotations and hence is not in $\mathbf{R_h}(n)$. Thus it must be that $\alpha_i^+ \in \mathbf{R_h}(n)$. For this case $x_i = a_1 - 1, y_i = a_1$ and $z_i = a_1 + 1$. The representative for α_i^+ must be for some set $\mathbf{T}_j$ where $i < j$ and will have $x_j = a_1$, $y_j = a_1 + 1$, and $z_j = a_1 - 1$. Furthermore, i will be the smallest index of a tuple containing β_i. Thus, the cyclic string described in (iii) is $(a_1{-}1)(a_1)(a_1{+}1) = x_i y_i z_i$. Thus condition (iii) is satisfied.

□

With the spanning sequence $\mathcal{S}_h$ we can apply Theorem 2.2 to obtain two UC-successors g_h and g'_h for $\mathbf{W_h}(n)$ where $\omega = w_1 \cdots w_n \in \mathbf{W_h}(n)$, and

$$\omega' = (w_1{+}1)w_2 \cdots w_n, \qquad \omega'' = (w_1{+}2)w_2 \cdots w_n, \qquad \omega^- = (w_1{-}1)w_2 \cdots w_n.$$

$$g_h(\omega) = \begin{cases} w_1 + 1 & \text{if } \omega' \in \mathbf{R_h}(n); \\ w_1 - 1 & \text{if } \omega' \notin \mathbf{R_h}(n) \text{ and } \omega \in \mathbf{R_h}(n) \text{ and } \omega^- \notin \mathbf{R_h}(n); \\ w_1 - 2 & \text{if } \omega' \notin \mathbf{R_h}(n) \text{ and } \omega \in \mathbf{R_h}(n) \text{ and } \omega^- \in \mathbf{R_h}(n); \\ w_1 & \text{otherwise.} \end{cases}$$

$$g'_h(\omega) = \begin{cases} w_1 - 1 & \text{if } \omega \in \mathbf{R_h}(n); \\ w_1 + 1 & \text{if } \omega \notin \mathbf{R_h}(n) \text{ and } \omega' \in \mathbf{R_h}(n) \text{ and } \omega'' \notin \mathbf{R_h}(n); \\ w_1 + 2 & \text{if } \omega \notin \mathbf{R_h}(n) \text{ and } \omega' \in \mathbf{R_h}(n) \text{ and } \omega'' \in \mathbf{R_h}(n); \\ w_1 & \text{otherwise,} \end{cases}$$

Theorem 4.2 *The functions g_h and g'_h are UC-successors for $\mathbf{W_h}(n)$.*

Proof. The function g'_h for $\mathbf{W_h}(n)$ is obtained from the spanning sequence $\mathcal{S}_h$ by applying Theorem 2.2 and then the observations from Lemma 4.1. The first case of Lemma 4.1 is handled by the first and second lines of g'_h. The second case of the Lemma 4.1 is handled by the first, second, and third line of g'_h. The sub-case with a_1 being the uniquely largest element in α_i is handled by the first and third line of g'_h, while the sub-case with a_1 being one of the largest but not the uniquely largest element in α_i is handled by the first and second lines of g'_h. The proof for g_h is similar. □

Testing whether or not a string is in $\mathbf{R_h}(n)$ can be done in $O(n)$ time by applying a result from [1]. Thus, we obtain the following corollary.

Corollary 4.3 *A universal cycle for $\mathbf{W_h}(n)$ can be constructed using either the successor $g_h(\omega)$ or $g'_h(\omega)$ starting from any initial weak order $\omega \in \mathbf{W_h}(n)$. Each universal cycle can be constructed in $O(n)$ time per symbol using $O(n)$ space.*

By applying the UC-successor g_h starting from 0^n, we obtain the following two universal cycles for $\mathbf{W_h}(n)$ for $n = 3$ and $n = 4$:

- $\triangleright$ $n = 3$: 0001012011021;
- $\triangleright$ $n = 4$: 000010102010012001101210231022103210112021302120312012301220132011102110021.

By applying the UC-successor g'_h starting from 0^n, we obtain the following two universal cycles for $\mathbf{W_h}(n)$ for $n = 3$ and $n = 4$:

- $\triangleright$ $n = 3$: 0001021012011;
- $\triangleright$ $n = 4$: 000010201010021001200110211012103210231022101120312021302120132012301220111.

5 Acknowledgement

The research of Joe Sawada is supported by the *Natural Sciences and Engineering Research Council of Canada* (NSERC) grant RGPIN 400673-2012. The research of Dennis Wong is supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Consilience Creative Program (IITP-2019-H8601-15-1011) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

References

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

[2] P. Diaconis and R. L. Graham. Products of universal cycles, in *A Lifetime of Puzzles. E. Demaine, M. Demaine and T. Rodgers, eds., A K Peters/CRC Press*, pages 35–55, 2008.

[3] D. Gabric, J. Sawada, A. Williams, and D. Wong. A framework for constructing de Bruijn sequences via simple successor rules. *Discrete Mathematics*, 341(11):2977–2987, 2018.

[4] D. Gabric, J. Sawada, A. Williams, and D. Wong. A successor rule framework for constructing k-ary de Bruijn sequences and universal cycles. *IEEE Transactions on Information Theory*, 2019.

[5] V. Horan and G. Hurlbert. Universal cycles for weak orders. *SIAM J. Discrete Math.*, 27(3):1360–1371, 2013.

[6] B. Jackson, B. Stevens, and G. Hurlbert. Research problems on Gray codes and universal cycles. *Discrete Mathematics*, 309(17):5341 – 5348, 2009.

[7] M. Jacques and D. Wong. Greedy universal cycle constructions for weak orders. In *Proceedings of the 6th International Conference on Algorithms and Discrete Applied Mathematics (CALDAM), India*, 2020.

[8] A. Leitner and A. Godbole. Universal cycles of classes of restricted words. *Discrete Mathematics*, 310(23):3303 – 3309, 2010.

[9] N. Sloane. The on-line encyclopedia of integer sequences, `http://oeis.org`. Sequence A000670.