Graphs and Combinatorics (2017) 33:1393–1399
DOI 10.1007/s00373-017-1778-3

ORIGINAL PAPER

A New Universal Cycle for Permutations

Dennis Wong[1]

Received: 31 August 2016 / Published online: 29 May 2017

Abstract We introduce a novel notation, the relaxed shorthand notation, to encode permutations. We then present a simple shift rule that exhaustively lists out each of the permutations exactly once. The shift rule induces a cyclic Gray code for permutations where successive strings differ by a rotation or a shift. By concatenating the first symbol of each string in the listing, we produce a universal cycle for permutations in relaxed shorthand notation. We also prove that the universal cycle can be constructed in $O(1)$-amortized time per symbol using $O(n)$ space.

Keywords Universal cycles · Permutations · de Bruijn sequences · Gray codes

1 Universal Cycles for Permutations

A *universal cycle* for a set **S** is a cyclic sequence of length $|\mathbf{S}|$ whose substrings of length n encode $|\mathbf{S}|$ distinct objects in **S**. As an example, the cyclic sequence 112132233 is a universal cycle for the set of 3-ary strings of length 2; the 9 unique substrings of length 2 when considered cyclicly are:

$$11, 12, 21, 13, 32, 22, 23, 33, 31.$$

A permutation of a character set $\langle n\rangle = \{1, 2, \ldots, n\}$ is an ordered arrangement of n distinct symbols in $\langle n\rangle$. A universal cycle for permutations of order n is a cyclic sequence of length $n!$ whose substrings encode each of the permutations of $\langle n\rangle$ exactly

✉ Dennis Wong
cwong@uoguelph.ca

1 Wenzhou-Kean University, Wenzhou, China

once. Universal cycles for permutations do not exist under standard one-line notation when $n > 2$ [1]. To demonstrate the non-existence result, suppose there exists a universal cycle for permutations in one-line notation, then the universal cycle must contain the substring $n(n-1)\cdots 1$. The next length n substring of the universal cycle starts with $(n-1)(n-2)\cdots 1$, and must end with the symbol n. By similar reasoning, the next n symbols after $n(n-1)\cdots 1$ are exactly $n(n-1)\cdots 1$, a repetition and thus contradicts with the assumption that it is a universal cycle.

Several other notations were introduced to construct universal cycles for permutations. Jackson proved that universal cycles for *k-permutations* of $\langle n\rangle$ exist when $k < n$, where a k-permutation is an ordered arrangement of k distinct symbols in $\langle n\rangle$ [4]. Knuth extended Jackson's result by introducing *shorthand notation* to encode permutations [6]. The shorthand notation of a permutation $a_1a_2\cdots a_n$ is $a_1a_2\cdots a_{n-1}$. A *shorthand universal cycle for permutations* of order n is a cyclic sequence of length $n!$ that contains each of the unique permutations of $\langle n\rangle$ in shorthand notation as a substring exactly once. For example, the cyclic sequence

$$123413243214213423143124$$

is a shorthand universal cycle for permutations of order 4; the 24 unique permutations when considered cyclicly are:

$$\begin{aligned}
&123\underline{4},\ 234\underline{1},\ 341\underline{2},\ 413\underline{2},\ 132\underline{4},\ 324\underline{1},\ 243\underline{1},\ 432\underline{1},\\
&321\underline{4},\ 214\underline{3},\ 142\underline{3},\ 421\underline{3},\ 213\underline{4},\ 134\underline{2},\ 342\underline{1},\ 423\underline{1},\\
&231\underline{4},\ 314\underline{2},\ 143\underline{2},\ 431\underline{2},\ 312\underline{4},\ 124\underline{3},\ 241\underline{3},\ 412\underline{3}.
\end{aligned}$$

The last symbol of each permutation in the listing is determined by a length 3 substring of the shorthand universal cycle. Holroyd, Ruskey and Williams provided efficient constructions to generate shorthand universal cycles for permutations in $O(1)$-amortized time per symbol and $O(1)$-amortized time per n symbols respectively using $O(n)$ space [2,3,7]. Permutations have also been encoded using relative order [1]. For example, 321341 is an order-isomorphic universal cycle for permutations of order 3 since its substrings are order-isomorphic to 321, 213, 123, 231, 312, 132. Johnson verified a conjecture in [1] to show that order-isomorphic universal cycles for permutations exist using only $n + 1$ symbols [5]. However, there is currently no known efficient construction to generate order-isomorphic universal cycles for permutations for all order n.

In this paper, a novel notation, the relaxed shorthand notation, is introduced to represent permutations. We then present a shift-based construction for producing a universal cycle for permutations in relaxed shorthand notation. The construction is based on the following function over permutations, where k is the largest possible position such that $a_k > a_{k+1}$:

$$f(a_1a_2\cdots a_n) = \begin{cases} a_2a_1a_3a_4\cdots a_n & \text{if } a_2 = 1 \text{ and } a_3a_4\cdots a_n \\ & \text{ is strictly increasing;} \\ a_2a_3\cdots a_ka_1a_{k+1}a_{k+2}\cdots a_n & \text{if } a_2 = 1 \text{ and } a_1 > a_{k+1}; \\ a_2a_3\cdots a_{k-1}a_1a_ka_{k+1}\cdots a_n & \text{if } a_2 = 1 \text{ and } a_1 < a_{k+1}; \\ a_2a_3\cdots a_na_1 & \text{otherwise,} \end{cases}$$

As an illustration, successive applications of this rule for $n = 4$ starting with the permutation 1234 produce the following listing:

$$\begin{aligned} &1234, 2341, 3412, 4123, 1423, 4231, 2314, 3142, \\ &1432, 4321, 3214, 2143, 1243, 2431, 4312, 3124, \\ &1324, 3241, 2413, 4132, 1342, 3421, 4213, 2134. \end{aligned} \tag{1}$$

Observe that each permutation of length 4 is visited exactly once and that by applying one more application of the rule, we return to the first string 1234. This property holds in general for all $n \geq 1$. This leads to the following theorem, where $\Pi(n)$ denotes the set of permutations of $\langle n \rangle$.

Theorem 1 *The shift rule f induces a cyclic ordering on $\Pi(n)$.*

The rest of the paper is outlined as follows. In Sect. 2, we introduce the relaxed shorthand notation. In Sect. 3, we prove Theorem 1, which leads to a universal cycle for permutations in relaxed shorthand notation. Then in Sect. 4, we present an algorithm that generates a universal cycle for permutations in this new notation in $O(1)$-amortized time per symbol using $O(n)$ space.

2 A Novel Notation to Represent Permutations

This section introduces the relaxed shorthand notation to represent permutations. The *relaxed shorthand notation* uses a length n string $\alpha = a_1a_2\cdots a_n$ with $n-1$ or n distinct symbols to represent a permutation of $\langle n \rangle$. If α contains n distinct symbols, then it simply represents the permutation $a_1a_2\cdots a_n$. Otherwise if α contains $n-1$ distinct symbols, then by pigeonhole principle there is a symbol which appears twice within α. Let a_i and a_j be the same symbol within α such that $i < j$. We can then obtain a length $n-1$ string with $n-1$ distinct symbols β by simply ignoring a_j, and shifting all symbols after a_j to the left by one position, that is $\beta = a_1a_2\cdots a_{j-1}a_{j+1}\cdots a_n$ when $j < n$, and $\beta = a_1a_2\cdots a_{n-1}$ if $j = n$. We then treat β as the shorthand notation of a permutation. Thus, the corresponding permutation can be obtained by appending the missing symbol in $\langle n \rangle$ to β. As an example, the permutation 1234 can be represented by 1123, 1213, 1223, 1231, 1232, 1233 and 1234 in relaxed shorthand notation.

A *relaxed shorthand universal cycle for permutations* of order n is a cyclic sequence of length $n!$ that contains each of the unique permutations of $\langle n \rangle$ in relaxed shorthand notation as a substring exactly once. As an example, the cyclic sequence

$$123414231432124313241342$$

is a relaxed shorthand universal cycle for permutations of order 4; the 24 unique permutations when considered cyclicly are listed out in (1).

Lemma 1 *A shorthand universal cycle for* $\mathbf{S} \in \Pi(n)$ *is a relaxed shorthand universal cycle for* $\mathbf{S}$.

Proof A shorthand universal cycle for $\mathbf{S}$ has its length equal to $|\mathbf{S}|$ and contains each of the length $n-1$ prefixes of permutations in $\mathbf{S}$ as a substring exactly once. Let $\alpha = a_1a_2\cdots a_n$ be a permutation in $\mathbf{S}$. Observe that appending any symbol in $\langle n\rangle$ at the end of the length $n-1$ prefix of α, that is $a_1a_2\cdots a_{n-1}$, produces a string that corresponds to α in relaxed shorthand notation. Thus, a shorthand universal cycle for $\mathbf{S}$ also contains each of the permutations in relaxed shorthand notation as a substring exactly once, and thus is a relaxed shorthand universal cycle for $\mathbf{S}$. □

3 Proof of Theorem 1

In [8], Williams introduced the cool-lex ordering that exhaustively lists out multiset permutations. A multiset is a generalization of set in which elements are allowed to appear more than once. For example, $\{1, 1, 2, 4\}$ is a multiset in which the element 1 appears twice in the multiset. A permutation of a multiset $\mathbf{M}$ is an ordered arrangement of the elements in $\mathbf{M}$. For example, the 12 unique permutations for the multiset $\{1, 1, 2, 4\}$ are:

$$1124, 1214, 2141, 1241, 2411, 4112, 1142, 1412, 4121, 1421, 4211, 2114.$$

Williams proved that the following simple shift rule exhaustively lists out each of the permutations in a multiset exactly once:

$$cool(a_1a_2\cdots a_n) = \begin{cases} a_2a_3\cdots a_na_1 & \text{if } a_2a_3\cdots a_n \\ & \text{is strictly decreasing;} \\ a_2a_3\cdots a_ka_1a_{k+1}a_{k+2}\cdots a_n & \text{if } a_1 > a_k; \\ a_2a_3\cdots a_ka_{k+1}a_1a_{k+2}a_{k+3}\cdots a_n & \text{otherwise,} \end{cases}$$

where k is the smallest value such that $a_k < a_{k+1}$. As an illustration, successive applications of this rule for the multiset $\{1, 1, 2, 4\}$ starting with 1124 produce the listing shown earlier in this Section.

A *necklace* is the lexicographically smallest string in an equivalence class of strings under rotation. Let $f^j(\alpha)$ be the string obtained from j successive applications of the shift rule f starting with α. Also, let $\overleftarrow{\alpha}$ denotes the reversal of α, that is $\overleftarrow{a_1a_2\cdots a_n} = a_na_{n-1}\cdots a_1$. We now prove Theorem 1 using the cool-lex result by Williams.

Theorem 1 *The shift rule* f *induces a cyclic ordering on* $\Pi(n)$.

Proof Let $\alpha = a_1a_2\cdots a_n \in \Pi(n)$ be a necklace. Thus $a_1 = 1$. By the definition of f, the next $n-1$ strings after α are all possible rotations of α. Thus it suffices to show that successive applications of f generate each of the necklaces in $\Pi(n)$ exactly once in cyclic order.

Since $a_1 = 1$, $f^{n-1}(\alpha) = a_na_1a_2\cdots a_{n-1} = a_n1a_2\cdots a_{n-1}$ by the definition of f. Observe that by one more application of f, we have $f(f^{n-1}(\alpha)) = f^n(\alpha) = a_1\overleftarrow{cool\ell(\overleftarrow{a_2a_3\cdots a_n})} = 1\overleftarrow{cool\ell(\overleftarrow{a_2a_3\cdots a_n})}$. Since the shift rule $cool\ell$ over permutations generates each of the permutations exactly once in cyclic order, successive applications of $\overleftarrow{cool\ell(\overleftarrow{a_2a_3\cdots a_n})}$ list out each of the permutations of $\{a_2, a_3, \ldots, a_n\}$ exactly once in cyclic order. Thus, successive applications of f^n and $1\overleftarrow{cool\ell(\overleftarrow{a_2a_3\cdots a_n})}$ on α generate each of the permutations in $\Pi(n)$ that starts with the symbol 1 exactly once in cyclic order, that is generating each of the necklaces in $\Pi(n)$ exactly once in cyclic order. □

Let U denotes the sequence created by concatenating the first symbol of each permutation in the listing generated by successive applications of f starting with $12\cdots n$. Clearly $|U| = |\Pi(n)|$. We then define a function g as follows:

$$g(a_1a_2\cdots a_n) = a_1\overleftarrow{cool\ell(\overleftarrow{a_2a_3\cdots a_n})}.$$

From the proof of Theorem 1, the sequence U can be summarized by the following formula:

$$U = \alpha_1 \cdot \alpha_2 \cdots \alpha_{(n-1)!}, \quad \text{where } \alpha_1 = 12\cdots n \quad \text{and} \quad \alpha_{i+1} = g(\alpha_i).$$

Corollary 1 *Each necklace in $\Pi(n)$ appears as a length n substring of U.*

We now prove that U is a relaxed shorthand universal cycle for permutations.

Theorem 2 *U is a relaxed shorthand universal cycle for $\Pi(n)$.*

Proof Since $|U| = |\Pi(n)|$, it suffices to show that each of the permutations in $\Pi(n)$ appears as a length n substring of U in relaxed shorthand notation.

Let $\beta = b_1b_2\cdots b_n \in \Pi(n)$. If β is a necklace, then β is a length n substring of U by Corollary 1, that is a length n substring of U in relaxed shorthand notation. Otherwise if β is not a necklace, then β is a rotation of some necklace $\alpha_i = a_1a_2\cdots a_n$, that is $\beta = a_ta_{t+1}\cdots a_na_1a_2\cdots a_{t-1}$ for some $1 < t \le n$. By the definition of U, the next n symbols in U after α_i are $g(\alpha_i) = \alpha_{i+1} = a_1a_2\cdots a_ja_na_{j+1}\cdots a_{n-1}$ for some $j < n-1$. If $t \le j$, then β is a substring of $\alpha_i \cdot a_1a_2\cdots a_j$ and U, which is also a length n substring of U in relaxed shorthand notation. Otherwise if $t > j$, then $b_1b_2\cdots b_{n-t+j+1} = a_ta_{t+1}\cdots a_na_1a_2\cdots a_j$ and $b_{n-t+j+2}b_{n-t+j+3}\cdots b_{n-1} = a_{j+1}a_{j+2}\cdots a_{t-2}$. Observe that $\gamma = a_ta_{t+1}\cdots a_na_1a_2\cdots a_ja_na_{j+1}a_{j+2}\cdots a_{t-2} = b_1b_2\cdots b_{n-t+j+1}a_nb_{n-t+j+2}b_{n-t+j+3}\cdots b_{n-1}$ is a length n substring of $\alpha_i\alpha_{i+1}$ and U with the symbol a_n appearing twice. Also, β can be represented by γ in relaxed shorthand notation. Thus, β also appears as a length n substring of U in relaxed shorthand notation. Therefore, U is a relaxed shorthand universal cycle for $\Pi(n)$. □

4 Generating our Universal Cycle Efficiently

By Corollary 1, U can be generated by starting with an arbitrary necklace in $\Pi(n)$ and repeatedly applying g until it reaches the starting necklace. The function g can be computed in $O(n)$ time. However, g is called only once for every n symbols generated. This leads to an $O(1)$-amortized time per symbol algorithm to generate U in Algorithm 1. A complete C implementation of the algorithm is given in the Appendix.

Algorithm 1 Shift-based algorithm to generate U in $O(1)$-amortized time per symbol.

1: **procedure** UCYCLEPERM
2: $\quad a_1a_2\cdots a_n \leftarrow 12\cdots n$
3: $\quad$ **do**
4: $\qquad$ Print$(a_1a_2\cdots a_n)$
5: $\qquad a_1a_2\cdots a_n \leftarrow g(a_1a_2\cdots a_n)$
6: $\quad$ **while** $a_1a_2\cdots a_n \neq 12\cdots n$

Theorem 3 *The algorithm* UCYCLEPERM *generates the relaxed shorthand universal cycle U for permutations in $\Pi(n)$ in $O(1)$-amortized time per symbol using $O(n)$ space.*

Acknowledgements The author would like to thank Joe Sawada and Aaron Williams for their helpful advice that greatly improved this paper.

Appendix: C code to Generate a Relaxed Shorthand Universal Cycle for Permutations in $\Pi(n)$ in $O(1)$-Amortized Time per Symbol Using $O(n)$ Space

```
#include<stdio.h>
int n,a[50];

//-----------------------------------------------------
// Return TRUE iff a[1..n] is 12..n
//-----------------------------------------------------
int StartStr() {
    int i;
    for (i=1; i<=n; i++) if (a[i] != i) return 0;
    return 1;
}

//-----------------------------------------------------
void g() {
    int i,j=2;

    for (i=n+1; i>1; i--) a[i] = a[i-1];
    a[1] = a[n+1];

    for (i=n; i>2; i--)
        if (a[i] < a[i-1]) {
            if (a[i] < a[1]) j = i-1;
            else j = i-2;
            break;
```

```
        }

    for (i=0; i<j; i++) a[i] = a[i+1];
    a[j] = a[0];
}

//-----------------------------------------------------------------------------
// Generate a relaxed shorthand universal cycle for permutations of order n
// in O(1)-amortized time per symbol
//-----------------------------------------------------------------------------
int main() {
    int i;

    printf("Enter n: ");
    scanf("%d", &n);

    for (i=1; i<=n; i++) a[i] = i;
    do {
        // print n symbols
        for (i=1; i<=n; i++)
            if (a[i] < 10) printf("%d", a[i]);
            else printf("%c", a[i]+87); // use characters to represent symbols >=
                10

        g();
    } while (!StartStr());

    printf("\n\n");
}
```

References

1. Chung, F., Diaconis, P., Graham, R.: Universal cycles for combinatorial structures. Discret. Math. **110**(1–3), 43–59 (1992)
2. Holroyd, A., Ruskey, F., Williams, A.: Faster generation of shorthand universal cycles for permutations. In: Computing and combinatorics, 16th annual international conference, COCOON 2010, Nha Trang, Vietnam, July 19–21, 2010. Proceedings, pp. 298–307 (2010)
3. Holroyd, A., Ruskey, F., Williams, A.: Shorthand universal cycles for permutations. Algorithmica **64**(2), 215–245 (2012)
4. Jackson, B.: Universal cycles of k-subsets and k-permutations. Discret. Math. **117**(13), 141–150 (1993)
5. Johnson, R.: Universal cycles for permutations. Discret. Math. **309**(17), 5264–5270 (2009)
6. Knuth, D.: The art of computer programming. Volume 4, fascicule 2. , Generating all tuples and permutations. The art of computer programming. Addison-Wesley, Upper Saddle River (2005) (Autre tirage: (2010))
7. Ruskey, F., Williams, A.: An explicit universal cycle for the $(n$-1)-permutations of an n-set. ACM Trans. Algor. **6**(3) , 45:1–45:12 (2010). doi:10.1145/1798596.1798598
8. Williams, A.: Loopless generation of multiset permutations using a constant number of variables by prefix shifts. In: Proceedings of the twentieth annual ACM-SIAM symposium on discrete algorithms, SODA 2009, New York, NY, USA, January 4–6, 2009, pp. 987–996 (2009)