Gray code order for Lyndon words
Vincent Vajnovszki

To cite this version:

HAL Id: hal-00966527
https://hal.inria.fr/hal-00966527
Submitted on 26 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Gray code order for Lyndon words

Vincent Vajnovszki

1 LE2I – UMR CNRS, Université de Bourgogne, B.P. 47 870, 21078 DIJON-Cedex France
vvajnov@u-bourgogne.fr


At the 4th Conference on Combinatorics on Words, Christophe Reutenauer posed the question of whether the dual reflected order yields a Gray code on the Lyndon family. In this paper we give a positive answer. More precisely, we present an $O(1)$-average-time algorithm for generating length $n$ binary pre-necklaces, necklaces and Lyndon words in Gray code order.

Keywords: Lyndon words, Gray codes, generating algorithms

1 Introduction and motivation

A $k$-Gray code for a set of binary strings $B \subset \{0, 1\}^n$ is an ordered list for $B$ such that the Hamming distance between any two consecutive strings in the list is at most $k$; if $k$ is minimal then the list is called a minimal Gray code.

The conjugacy class of a string $x = uv$ is the set of all strings $vu$. A (binary) necklace is a binary string which is minimal with respect to lexicographic order in its conjugacy class and a Lyndon word is an aperiodic necklace. A pre-necklace is a binary string which is the prefix of some necklace, or equivalently (12), of some Lyndon word. Generation of necklaces and Lyndon words lexicographically have been widely studied in the literature: (2; 3; 5; 6; 8; 13; 14; 15; 16; 17; 22).

Binary strings (necklaces, Lyndon words) with a given number of ones are called binary strings (necklaces, Lyndon words) with fixed density. There are two algorithms for generating a Gray code for necklaces with fixed density. The first one published is due to Wang and Savage in 1996 (23) (where the representative used is not the lexicographically smallest); the second one, where Lyndon words with fixed density are also considered, is due to Ueda in 2000 (18). It is worth mentioning that the second paper was submitted two years before the first one. Both Gray codes are 2-Gray codes and therefore minimal.

To the best of our knowledge, there is no Gray code for necklaces and Lyndon words with no density restriction. The aim of this note is to present the first one and it is an updated form of the preliminary conference version of the paper (20).

In the following, lower case bold letters represent length $n$ binary strings, e.g., $x = x_1x_2 \ldots x_n$. The lexicographical order is defined as: $x$ is less than $y$ iff $x_k = 0$ and $y_k = 1$, where $k$ is the leftmost position with $x_k \neq y_k$; and any proper prefix of a string $x$ is less than $x$. The reflected
Gray code order on \( \{0, 1\}^n \), due to Frank Gray in 1953 (7), is given by (4): \( x < y \) if \( \sum_{j=1}^{k} x_j \) is even (and \( \sum_{j=1}^{k} y_j \) is odd), where \( k \) is the leftmost position with \( x_k \neq y_k \).

We say that an order relation on a set of strings induces a \( k \)-Gray code if the set listed in this order yields a \( k \)-Gray code. For example, the reflected Gray code order induces a 1-Gray code on \( \{0, 1\}^n \); and its restriction to the strings with fixed density (i.e., strings with a constant number of 1’s) induces a 2-Gray code, called revolving door code by Nijenhuis and Wilf (11).

The next definition gives a slight modification of reflected order and like lexicographical order, both of them are particular cases of genlex order (21), that is, any set of strings listed in such an order has the property that strings with a common prefix are contiguous.

**Definition 1** (19) We say that \( x \) is less than \( y \) in dual reflected order (or local reflected order), denoted by \( x \prec y \), if \( x_1x_2\ldots x_k, \) the length \( k \) prefix of \( x \) contains an odd number of 0’s, where \( k \) is the leftmost position with \( x_k \neq y_k \).

**Remark 1**

1. \( x \prec y \) if \( \overline{x} > \overline{y} \) in reflected Gray code order, with \( \overline{x} \) and \( \overline{y} \) the bitwise complement of \( x \) and \( y \).

2. Like reflected Gray code order, dual reflected order \( \prec \) induces a 1-Gray code on \( \{0, 1\}^n \) and a 2-Gray code on the strings in \( \{0, 1\}^n \) with fixed density.

3. In (19) and (1) it is shown that, unlike reflected Gray code order, dual reflected order \( \prec \) induces a 1-Gray code on the set of generalized Fibonacci strings (the set of length \( n \) binary
strings such that there are no $p$ consecutive 1’s) and on the set of generalized Lucas strings (the set of Fibonacci strings where the pattern $1^p$ is regarded circularly).

See Table 1.

## 2 The Gray code

**Definition 2** A binary string set $X \subset \{0,1\}^n$ is called absorbent if for any $x \in X$ and any $k$, $1 \leq k < n$, $x_1x_2\ldots x_k1^{n-k}$ is also a string in $X$.

The next lemma shows that, in contrast to the Fibonacci or Lucas strings, the sets of pre-necklaces, necklaces and Lyndon words are absorbent (see again Table 1). Its proof is based on the following result (10, Proposition 5.1.2): a binary string is a Lyndon word if and only if it is lexicographically less than any of its proper suffixes. Equivalently, using the definition of lexicographic order, we have the following characterization: a binary string is a Lyndon word if and only if its proper prefixes is less than its suffix of same length.

**Lemma 1** The sets of length $n$ Lyndon words, necklaces and pre-necklaces are absorbent.

**Proof:** Let $x = x_1x_2\ldots x_n$ be a binary string and $x' = x_1x_2\ldots x_{k-1}1^{n-k} \neq x$ with $1 \leq k < n$. When $x$ is a Lyndon word it is easy to check that $x'$ satisfies the above characterization of Lyndon words.

If $x$ is a necklace or a pre-necklace (but not a Lyndon word) then there exists $j$, $1 \leq j < n$, such that the prefix $x_1x_2\ldots x_j$ is a Lyndon word and $x = (x_1x_2\ldots x_j)^p x_{j+1}x_{j+2}\ldots x_r$ with $p = \lfloor \frac{n-j}{2} \rfloor$ and $r = n \mod j$. The final segment $x_1x_2\ldots x_r$ is empty iff $x$ is a necklace, see for instance (12). Thus a suffix of $x'$ must have the form $y_1x_2\ldots x_{j-1}x_j\ldots x_{n-1}1^{n-k}$ with the initial segment $x_{j-1}x_{j-2}\ldots x_1$ possibly empty. Such a suffix is greater than the same length prefix and therefore $x'$ is a Lyndon word. \qed

Let $X \subset \{0,1\}^n$, $x \in X$ and $k$ be the leftmost position where $x$ differs from its predecessor in $X$ in $\prec$ order (assuming that $x$ is not the first string in $X$ in this order). In other words, since $\prec$ is a genlex order, $x_1x_2\ldots x_k$ is the shortest prefix of $x$ such that for all $y \prec x$ one has $y_1y_2\ldots y_k \prec x_1x_2\ldots x_k$. We call $x_{k+1}x_{k+2}\ldots x_n$ the first discriminant suffix of $x$. Similarly, $x_{k+1}x_{k+2}\ldots x_n$, the last discriminant suffix of $x$, is the $n-k$ suffix of $x$ where $k$ is the leftmost position where $x$ differs from its successor in $\prec$ order (assuming that $x$ is not the last string in $X$ in this order). Let $|y|_0$ denote the number of 0’s in the binary string $y$.

**Lemma 2** If $x$ is a string in an absorbent subset $X$ of $\{0,1\}^n$ and $x_{k+1}x_{k+2}\ldots x_n$ the first or the last discriminant suffix of $x$, then $|x_{k+1}x_{k+2}\ldots x_n|_0 \leq 1$.

**Proof:** The proof is by contradiction. Let $x_{k+1}x_{k+2}\ldots x_n$ be a first discriminant suffix of $x$ such that $|x_{k+1}x_{k+2}\ldots x_n|_0 > 1$. Then there are at least two indices $i$ and $j$, $k+1 \leq i < j \leq n$, such that $x_i = x_j = 0$ and $x_{i+1} = x_{i+2} = \ldots = x_{j-1} = 1$. Since $X$ is absorbent, $x' = x_1x_2\ldots x_{i-1}1^{n-i}1^{n-j}1^{n-j+1}$ and $x'' = x_{i+1}x_{i+2}\ldots x_{j-1}1^{n-j+1}$ are both in $X$. In addition, by Definition 1, either $x' \prec x$ or $x'' \prec x$. So, since $\prec$ is a genlex order on $X$, the leftmost position where $x$ differs from its predecessor is greater than $k$; this is in contradiction with $x_{k+1}x_{k+2}\ldots x_n$ is the first
discriminant suffix of \( x \). The case where \( x_{k+1}x_{k+2}\ldots x_n \) is the last discriminant suffix of \( x \) is similar.

\[ \blacksquare \]

**Corollary 1** If \( X \) is an absorbent subset of \( \{0,1\}^n \) then \( X \) listed in \( \prec \) order is a 3-Gray code.

**Proof:** Let \( y \) be the successor of \( x \) in \( X \) with respect to \( \prec \) order and \( k \) be the leftmost position where \( x \) differs from \( y \). Then, \( x_{k+1}x_{k+2}\ldots x_n \) (resp. \( y_{k+1}y_{k+2}\ldots y_n \)) is the last (resp. the first) discriminant suffix of \( x \) (resp. \( y \)). By Lemma 2, \( x_i = y_i = 1 \) for all \( i, k+1 \leq i \leq n \), except possibly two of them. Since \( x_i = y_i \) for all \( i, 1 \leq i \leq k-1 \), the Hamming distance between \( x \) and \( y \) is at most three.

\[ \blacksquare \]

We now state the main result of this note, which follows directly from Lemma 1 and Corollary 1.

**Theorem 1** The \( \prec \) order yields a 3-Gray code on the sets of pre-necklaces, necklaces and Lyndon words of length \( n \).

Notice that similar results can be obtained for bracelets (minimal necklaces under the operation of reversal). Also, the first element in these Gray code lists is \( 01^{n-1} \) and the last one is \( 1^n \) (for pre-necklaces and necklaces) and \( 001^{n-2} \) (for Lyndon words and \( n \geq 3 \)). Thus all these Gray codes are circular, i.e., the first and last strings differ also in at most three positions.

### 3 Generating algorithm

Here we give an efficient algorithm for generating length \( n \) pre-necklaces, necklaces and Lyndon words in dual reflected order. It is based on Ruskey’s implementation (12) of the FKM algorithm (6; 8), which generates pre-necklaces, necklaces and Lyndon words in lexicographical order. In our Gray code version, when two values are possible for \( a[t] \), the current entry in the generated string, then either 0 or 1 is assigned first, depending upon the parity of the number of 0’s in the prefix \( a[1]a[2]\ldots a[t-1] \). This suffices to impose dual reflected order on the outputs and is obtained by adding a new binary parameter \( z \) which has the same parity as the number of 0’s in this prefix.

Like the Ruskey-FKM algorithm, our algorithm \( \text{gen\_Gray} \) below generates length \( n \) pre-necklaces; if \( \text{Print}(a) \) is called only when \( p \) is a divisor of \( n \) or when \( p = n \), then the algorithm lists, respectively, necklaces or Lyndon words. The main call is \( \text{gen\_Gray}(0,1,1) \) and \( a[0] = 0 \).

In (14) is shown that the Ruskey-FKM algorithm, and therefore also our Gray code version, generates necklaces and Lyndon words in constant average time; see also (12). The output of our algorithm for \( n = 5 \) is shown in Table 2.

\[ \begin{align*}
\text{procedure} &\quad \text{gen\_Gray}(z, t, p) \\
&\quad \text{if} \ t > n \text{ then } \text{Print}(a) \\
&\quad \text{else if} \ a[t-p] = 1 \\
&\quad \quad \text{then} \ a[t] := a[t-p]; \ \text{gen\_Gray}(z, t+1, p) \\
&\quad \quad \text{else} \ a[t] := z; \ \text{gen\_Gray}(1, t+1, p) \\
&\quad \quad \quad a[t] := 1 - z; \ \text{gen\_Gray}(0, t+1, t) \\
&\quad \text{end.}
\end{align*} \]
Gray code order for Lyndon words

<table>
<thead>
<tr>
<th>pre-necklaces</th>
<th>necklaces</th>
<th>Lyndon words</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 1 1 1</td>
<td>0 1 1 1 1</td>
<td>0 1 1 1 1</td>
</tr>
<tr>
<td>0 1 1 1 0</td>
<td>0 1 0 1 1</td>
<td>0 1 0 1 1</td>
</tr>
<tr>
<td>0 1 1 0 1</td>
<td>0 0 0 1 1</td>
<td>0 0 0 1 1</td>
</tr>
<tr>
<td>0 1 0 1 0</td>
<td>0 0 0 0 0</td>
<td>0 0 0 0 1</td>
</tr>
<tr>
<td>0 1 0 1 1</td>
<td>0 0 0 0 1</td>
<td>0 0 1 0 1</td>
</tr>
<tr>
<td>0 0 0 1 1</td>
<td>0 0 1 0 1</td>
<td>0 0 1 1 1</td>
</tr>
<tr>
<td>0 0 0 1 0</td>
<td>0 0 1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0</td>
<td>1 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 1 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 1 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 1 1 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tab. 2: The set of length 5 pre-necklaces, necklaces and Lyndon words in dual reflected order. Note that for $n \leq 5$ the Hamming distance between successive strings never exceeds two.

4 Final remarks

It can be shown that, in general, there is no listing for length $n$ necklaces in which successive representatives differ in just one bit (23). The existence of Gray codes for necklaces or Lyndon words where successive objects differ in at most two positions and a loopless implementation ($O(1)$ in the worst case) of our algorithm remain open problems. Finally, the author would like to thank Frank Ruskey for providing improvements to the algorithm and one of the referees for helpful comments.

Note added in proof: A recent subsequent generalization of our results to a $k$-ary alphabet ($k > 2$) appears in (24).
References


