Paths of Lyndon Words

L. J. Cummings
University of Waterloo
and
J. L. Yucas
Southern Illinois University

Abstract. The set of Lyndon words of length n, Λ_n, is the set obtained by choosing those strings of length n over any finite alphabet Σ of cardinality σ which are lexicographically least in the primitive or aperiodic equivalence classes determined by cyclic permutation. It is well-known that Λ_n is a maximal synchronizable code with bounded synchronization delay for fixed word length n. If the Lyndon words of length n are represented as vertices of the n-cube we show that they form a connected set for arbitrary alphabets. Indeed, we show that between any two Lyndon words there is a path consisting of at most $2n$ Lyndon words in the n-cube. Further, we show that there always exists a path of $n(\sigma-1)-1$ Lyndon words in the n-cube.

Let Σ be a nonempty totally ordered finite set, called the alphabet, with cardinality σ. A mapping $s : \{1, \cdots, n\} \longrightarrow \Sigma$ is a *string* of length n. We denote a string of length n by $s[1]s[2]\cdots s[n]$. Let Σ^n denote the set of all σ^n strings of length n over Σ and set $\Sigma^* = \cup\Sigma^n$. We suppose further that Σ^* contains the empty string, λ. For notational convenience, set $\Sigma^+ = \Sigma^* - \{\lambda\}$.

Define an equivalence relation on Σ^* by: $u \sim v$ if there are strings $x, y \in \Sigma^+$ such that $u = xy$ and $v = yx$. If $u \sim v$ then u and v are said to be *conjugate*. The resulting equivalence classes are sometimes referred to as "circular strings". Equivalently, two strings are conjugate if and only if one is a cyclic shift of the other.

In the sequel we are concerned with just those circular strings that are primitive. A string w is *primitive* if $w \neq u^k$, for any $u \in \Sigma^+$ and positive integer k. Here, exponential notation is used to indicate the concatenation of k copies of the substring u. Note that if w is primitive and $v \sim w$, then v is primitive.

An easy counting argument using elementary Möbius inversion shows that the number of primitive strings with fixed length n is

$$S(n,\sigma) = \Sigma\mu(n/d)\sigma^d, \tag{1}$$

where the summation is over all positive divisors d of n and μ is the Möbius function of elementary number theory. (For a proof using Möbius inversion see, for example, [1].)

It will be convenient to consider the set Σ^* as ordered by the usual lexicographical ordering, that is, strings u and v in Σ^+ satisfy $u < v$ if

(i) $v = uv'$ for some $v' \in \Sigma^+$ or,
(ii) $u = ras, v = rbt$ and $a < b$ for $a, b \in \Sigma$; $r, s, t \in \Sigma^*$.

We state as Lemma 1 two well-known properties of lexicographical ordering:

Dedicated to the memory of Rodger Lyndon

Lemma 1. *If* $u, v \in \Sigma^+$ *then*

(i) $u < v$ *if and only if* $wu < wv$ *for all* $w \in \Sigma^*$

(ii) *if* $v \neq uv'$ *for any* $v' \in \Sigma^*$ *then* $u < v$ *implies* $uw < vx$ *for all* $w, x \in \Sigma^*$.

$\Lambda_n = \Lambda_n(\sigma)$ is the set of strings of length $n \geq 1$ in Σ^n which are lexicographically least in the primitive equivalence classes determined by $\sim$. The strings in Λ_n are called *Lyndon words*. We set $\Lambda_1 = \Sigma$ and $\Lambda = \cup\Lambda_n$.

From (1) the cardinality of $\Lambda_n(\sigma)$ is $\frac{1}{n}S(n, \sigma)$ since each equivalence class of $\sim$ containing primitive strings has n elements. Our interest in Λ_n stems from the work of Golomb and Gordon [4] who first proved that Λ_n is a maximal block code with bounded synchronization delay. We now state two well-known properties of Λ as Lemmas:

Lemma 2.

(i) *A string* $w \in \Lambda$ *if and only if* $w = uv$ *where* $u, v \in \Lambda$ *and* $u < v$ *in the lexicographical order.*

(ii) *If* $u, v \in \Lambda$ *and* $u < v$ *then* $u^k v \in \Lambda$ *for every* $k \geq 1$.

Proof: For a proof of (i) see [5]. Notice that (ii) follows from repeated applications of (i) of this lemma and (i) of the definition of lexicographical order. ∎

Lemma 2 (i) yields a recursive algorithm to generate all the strings in Λ_n. But the difficulty lies in the fact that many repetitions of the same string may be generated, necessitating frequent "lookups" in any program to generate all the strings of Λ_n without repetition. For example, in Λ when $\Sigma = \{0, 1\}$:

$$010111 = (01)(0111) = (01011)1.$$

Both factorications above are in Λ since $01 \in \Lambda_2$, $0111 \in \Lambda_4$, $01011 \in \Lambda_5$, and $1 \in \Lambda_1 = \{0, 1\}$.

If $w = uv$ and u is non-empty the v is a *proper right factor* of w. For a proof of the following lemma see [5].

Lemma 3. *A string* $w \in \Lambda$ *if and only if* w *is strictly less in the lexicographical ordering than each of its proper right factors.*

Lemma 3 appears to yield a more efficient algorithm for generating Λ_n but it still requires testing each of the 2^n binary strings of length n.

Recently, Duval [3] has given an algorithm which lists the words of Λ_n in lexicographical order in linear time without auxiliary memory. Our concern here is with a "geometrical" ordering of Λ_n. In particular, we are interested in listing Λ_n with only a single bit change between adjacent strings.

Proposition 4. *If* $w = w_1 w_2 \cdots w_m \in \Lambda_{mn}$ *with* $w_i \in \Lambda_n$ *and if* $x \in \Lambda_n$ *with* $w_i < x \leq w_k$ *for all* $i \neq 1$ *and all* $k > i$ *then*

$$w' = w_1 w_2 \cdots w_{i-1} x w_{i+1} \cdots w_m \in \Lambda_{mn}.$$

Proof: By Lemma 3 it suffices to show that each proper right factor y of w' is larger than w'. Notice first that if y begins with a proper right factor of one of the w_j's then $y > w_j$ since $w_j \in \Lambda_n$ hence $y > w_j w_{j+1} \cdots w_m$. Similarly, if y begins with a proper right factor of x then $y > x w_{i+1} \cdots w_m$. Thus it suffices to show that each proper right factor y of w' beginning with a w_j or x is larger than w'. Since $w_i w_{i+1} \cdots w_m$ is a proper right factor of w it follows that $w_i \geq w_1$ and hence $x > w_1$. Consequently, $x w_{i+1} \cdots w_m > w'$. On the other hand suppose $y = w_j w_{j+1} \cdots w_m$. If $j > i$ then $w_1 \leq w_i < x \leq w_j$ implies $w_j > w_1$ and hence $y > w'$. If $j < i$ then $y = w_j \cdots x \cdots w_m$. First notice that if there is a p with $1 \leq p \leq i-1$ and $w_{j+p-1} > w_p$ then taking the smallest such p yields $w_j \cdots w_{j+p-1} > w_1 \cdots w_p$ hence $y > w'$. So assume $w_{j+p-1} = w_p$ for $p = 1, 2, \cdots, i-1$. Here notice that $w_j \cdots w_{i-1} w_i \geq w_1 \cdots w_{i-j} w_{i-j+1}$ since $w \in \Lambda_{mn}$ so $w_i \geq w_{i-j+1}$. Since $x > w_i$ we have $x > w_{i-j+1}$ thus $w_j \cdots w_{i-1} x > w_1 \cdots w_{i-j} w_{i-j+1}$ and $y > w'$. ∎

The *n-cube* over an alphabet Σ is the graph whose vertices are the strings of Σ^n with an edge between distinct vertices α and β if $d(\alpha, \beta) = 1$, where $d(\alpha, \beta)$ denotes the Hamming distance between α and β; i.e., the number of bits in which α and β differ as strings. A set S of distinct vertices $v_1, v_2, \cdots, v_k$ in the n-cube over Σ determine a *path* if there is an ordering

$$v_{\sigma(1)}, v_{\sigma(2)}, \cdots, v_{\sigma(k)}$$

of the vertices in S such that

$$d(v_{\sigma(i)}, v_{\sigma(i+1)}) = 1 \text{ for } i = 1, 2, \cdots, k-1.$$

We say that the path is *ordered* if

$$v_{\sigma(1)} < v_{\sigma(2)} < \cdots < v_{\sigma(k)}.$$

We proceed with applications of Proposition 4. In [2] it was shown that the code Λ_n is a connected subset of the n-cube for $\Sigma = \{0, 1\}$. The connectivity of Λ_n for arbitrary alphabets is given in Theorem 6 which follows from Lemma 5.

Lemma 5. *Let* $u, v \in \Lambda_n$. *There is a path of at most* $2n$ *Lyndon words in the* n*-cube starting at* u *and ending at* v.

Proof: If $n = 1$ then u, v is the desired path. Suppose $n > 1$. Let z be the largest element of $\Sigma = \Lambda_1$ and suppose $u = x_1 x_2 \cdots x_n, x_i \in \Sigma$. Let i be the largest integer such that $x_i < z$. Such an i exist since $z^n \notin \Lambda_n$. If $i \neq 1$ then

$$u' = x_1 \cdots x_{i-1} z x_{i+1} \cdots x_n \in \Lambda_n$$

by Proposition 4 and $d(u, u') = 1$. Repeating this process we see that there is a path of at most n Lyndon words in the n-cube from u to $x_1 z^{n-1}$. $x_1 z^{n-1}$ is in Λ_n by Lemma 2. Similarly, there is a path of at most n Lyndon words in the n-cube from v to $y_1 z^{n-1}$ for some $y_1 \in \Sigma \setminus \{z\}$. Notice that $d(x_1 z^{n-1}, y_1 z^{n-1}) \leq 1$. Thus these paths can be joined to form a path from u to v of at most $2n$ Lyndon words in the n-cube. ∎

Theorem 6. *If $n \geq 1$ then Λ_n is a connected subset of the n-cube over any finite alphabet.*

Theorem 7. *If there is an ordered path of m Lyndon words in the n-cube then for every integer $r \geq 1$ there is a path of $r(m-1)-1$ Lyndon words in the rn-cube.*

Proof: Suppose $w_1 < w_2 < \cdots < w_m$ is an ordered path of Lyndon words in the n-cube. For $r = 1, w_1, w_2, \cdots, w_{m-2}$ will work so assume $r > 1$. By Lemma 2(ii), $w_i^{r-1} w_{i+1} \in \Lambda_{rn}$ for $i = 1, 2, \cdots, m-1$ and notice that

$$d(w_i^{r-j} w_{i+1}^j, w_i^{r-j-1} w_{i+2}^{j+1}) = d(w_i, w_{i+1}) = 1$$

for $j = 1, 2, \cdots, r-2$. By repeated applications of Proposition 4, we see that

$$w_i^{r-1} w_{i+1}, w_i^{r-2} w_{i+1}^2, \cdots, w_i w_{i+1}^{r-1}$$

is a path of $r-1$ Lyndon words in the rn-cube. Finally notice that each of these paths can be joined to the next path by inserting the Lyndon word $w_i w_{i+1}^{r-2} w_{i+2}$. This yields a path from $w_1^{r-1} w_2$ to $w_{m-1} w_m^{r-1}$ consisting of $r(m-1)-1$ Lyndon words in the rn-cube. ∎

Corollary 8. *For every integer $r \geq 1$ there is a path of $r(\sigma - 1) - 1$ Lyndon words in the r-cube.*

Proof: The ordered alphabet Σ is an ordered path of σ Lyndon words in the 1-cube, thus the result follows from Theorem 6. ∎

References

1. L.J. Cummings, *Aspects of synchronizable coding*, The Journal of Combinatorial Mathematics and Combinatorial Computing **1** (1987), 67–84.
2. L.J. Cummings, *Connectivity of Lyndon words in the n-cube*, The Journal of Combinatorial Mathematics and Combinatorial Computing **3** (1988), 93–96.
3. J.-P. Duval, *Génération d'une section des classes de conjugation et arbre des mots de Lyndon de longueur bornée*, LITP report 88–20, Paris (March 1988).
4. S.W. Golomb and B. Gordon, *Codes with bounded synchronization delay*, Information and Control **8** (1965), 355–372.
5. M. Lothaire, "Combinatorics on Words", Addison-Wesley, Reading, Massachusetts, 1983.