Several theoretical results from Frank Ruskey's **Combinatorial Generation** concerning prenecklaces, necklaces, and Lyndon words

typeset by M. Markov, the author is of course Frank Ruskey

May 3, 2018

Abstract

The presented theorems and their theoretical background is based on Chapter 7 of Frank Ruskey's ***Combinatorial Generation*** *[Rus03].*

1 Preliminaries

An alphabet is a finite set of *letters*. We think of letters as natural numbers. Typically, our alphabet is $\Sigma_k = \{0, 1, \ldots, k-1\}$ for some $k \geqslant 2$. The set of all *strings* over Σ_k is Σ_k^*. The set of all strings of length n is Σ_k^n. Thus, $\Sigma_k^* = \bigcup_{n\in\mathbb{N}} \Sigma_k^n$. Furthermore, $\Sigma_k^+ = \bigcup_{n\in\mathbb{N}^+} \Sigma_k^n$.

The length of any string x is denoted by $|x|$.

The symbol 'ϵ' denotes the empty string. Clearly, $\epsilon = \Sigma_k^* \backslash \Sigma_k^+$ and $|\epsilon| = 0$.

The concatenation operation is denoted with no symbol. If $\sigma_1 \in \Sigma_k^{n_1}$ and $\sigma_2 \in \Sigma_k^{n_2}$, the concatenation of σ_1 and σ_2, in that order, is denoted by $\sigma_1\sigma_2$. Clearly, $\sigma_1\sigma_2 \in \Sigma_k^{n_1+n_2}$. Let $\sigma_1\sigma_2$ be called x. We say that $\sigma_1\sigma_2$ is a *factorisation* of x and σ_1, σ_2 are *factors* of x. We say the factorisation is *nontrivial* iff $\sigma_1 \neq \epsilon$ and $\sigma_2 \neq \epsilon$.

A prefix of any string x is any string u such that there is a factorisation, not necessarily nontrivial, $x = uv$. *A proper prefix* of x is any prefix of x that is not equal to x. Likewise, v is *a suffix* of x, and *a proper suffix* is one that does not equal the whole string x.

The *lexicographic order relation over* Σ_k, shortly the *lex order*, is the relation $\preceq$ over $\Sigma_k^* \times \Sigma_k^*$ defined as follows. For all $x, y \in \Sigma_k^*$, $x \preceq y$ iff:

* $x = \epsilon$ or
* $x = ax_1$ and $y = by_1$ and

$\star\star$ $a < b$ or

$\star\star$ $a = b$ and $x_1 \preceq y_1$

where $a, b \in \Sigma_k$ and $x_1, y_1 \in \Sigma_k^*$.

We define another relation over the same domains and call it '$\prec$'. For all $x, y \in \Sigma_k^*$, $x \prec y$ iff $x \preceq y$ and $x \neq y$.

The notation "$x = y^t$" says that x is the $(t-1)$-times concatenation of the string y with itself:

$$x = \underbrace{yy\cdots y}_{\text{t factors}}$$

Clearly, for every string $x \in \Sigma_k^*$ it is the case that $x = y^t$, for some $t \geqslant 1$ where y is some string in Σ_k^* such that $|x| = t \cdot |y|$.

Let $x \in \Sigma_k^n$ for some $n > 0$. We say x is *periodic* iff $x = y^t$ for some $t \geqslant 2$. x is *aperiodic* iff x is not periodic; in other words, x is aperiodic iff $x = y^t$ for and only for $t = 1$. If $x = y^t$ and y is aperiodic we say y is *the periodic reduction* of x and $|y|$ is *the period* of x. E.g., 010100 is aperiodic and 010101 is periodic; furthermore, the periodic reduction of 010101 is 01 and thus the period is 2.

Let $x, y \in \Sigma_k^n$. We say x *is a rotation of* y iff there exists a factorisation of x, not necessarily nontrivial, say $x = uv$ such that $y = vu$. We define the relation $\sim_n \subseteq \Sigma_k^n \times \Sigma_k^n$ as follows:

$\forall x, y \in \Sigma_k^n : x \sim_n y$ iff x is a rotation of y

Proposition 1. *For all $n \in \mathbb{N}$, the relation $\sim_n$ is an equivalence relation.*

The classes of equivalence of $\sim_n$ are called *the necklaces*. For simplicity, each such class is identified with its smallest—in lex order—element. n is *the size* of each of those necklaces. E.g., there are precisely six necklaces of size four:

$$0000, 0001, 0011, 0101, 0111, 1111$$

The set of all necklaces denoted by $\mathcal{N}$. The set of all necklaces of size n over Σ_k^n is denoted by $\mathcal{N}_k(n)$. Equivalently:

$$\mathcal{N}_k(n) = \{x \in \Sigma_k^* \mid \forall y \text{ such that } y \sim_n x : x \preceq y\}$$

As already mentioned, $\mathcal{N}_2(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}$.

Any aperiodic necklace is called *Lyndon word*. The set of all Lyndon words is denoted by $\mathcal{L}$. The set of all Lyndon words of size n over Σ_k^n is denoted by $\mathcal{L}_k(n)$. Equivalently:

$$\mathcal{L}_k(n) = \{x \in \mathcal{N}_k(n) \mid x \text{ is aperiodic}\}$$

Clearly, $\mathcal{L}_2(4) = \{0001, 0011, 0111\}$.

A prenecklace is any string that is a prefix of some necklace. Clearly, every necklace is a prenecklace. Note that not every string is a prenecklace. E.g., 1000 is not a prenecklace because for any $\beta \in \Sigma_k^*$ it is the case that $1000\beta \not\leqslant 000\beta 1$. The set of all prenecklaces of size n over Σ_k^n is denoted by $\mathcal{P}_k(n)$. Equivalently:

$$\mathcal{P}_k(n) = \{x \in \Sigma_k^* \mid \exists m \in \mathbb{N} \text{ such that } \exists y \in \Sigma_k^m \text{ such that } xy \in \mathcal{N}_k(n+m)\}$$

Clearly, $\mathcal{P}_2(4) = \mathcal{N}_2(4) \cup \{0010, 0110\}$.

2 Lemmas and Theorems on Prenecklaces, Necklaces and Lyndon Words

Let $\alpha \in \Sigma_k^n$ be the string $\alpha = a_0 a_1 \cdots a_{n-1}$.

Lemma 1 (Lemma 7.1 in [Rus03]). *If* $\alpha = xy = yx$ *and* $x \neq \epsilon$ *and* $y \neq \epsilon$ *then* $\alpha = (a_0 a_1 \cdots a_{d-1})^{\frac{n}{d}}$ *where* $d = gcd(n, |x|)$.

Proof: Let $|x| = m$. Then $xy = yx$ implies:

$$\begin{aligned}
a_0 &\equiv a_m \pmod n\\
a_1 &\equiv a_{m+1} \pmod n\\
a_2 &\equiv a_{m+2} \pmod n\\
&\cdots\\
a_{m-1} &\equiv a_{2m-1} \pmod n\\
a_m &\equiv a_{2m} \pmod n\\
&\cdots\\
a_{n-1} &\equiv a_{m+n-1} \pmod n, \text{ i.e. } a_{n-1} \equiv a_{m-1} \pmod n
\end{aligned}$$

Shortly, $a_i \equiv a_{i+m} \pmod n$, for $0 \leqslant i \leqslant n-1$. The remainder of the proof relies on Lemma 2.2 in [Rus03] that says that, given m and n, iterating $jm \pmod n$ for $0 \leqslant j \leqslant n-1$ yields a multiset of n numbers (so far, it is obvious) that consists of d copies of each of the $\frac{m}{d}$ distinct numbers kd, $0 \leqslant k \leqslant \frac{m}{d} - 1$, where $d = \gcd(m, n)$. The proof of Lemma 2.2 in [Rus03] in its turn relies on Section 4.8 in [GKP94]. □

Corollary 1 (Corollary 7.1 in [Rus03]). *If* $\alpha = xy = yx$ *and* $x \neq \epsilon$ *and* $y \neq \epsilon$ *then* α *is periodic.*

Theorem 1 (Theorem 7.3 in [Rus03]). *The following formulae are valid for all* $n \geqslant 1$, $k \geqslant 1$*:*

$$|\mathcal{L}_k(n)| = \frac{1}{n} \sum_{d|n} \mu\left(\frac{n}{d}\right) k^d \tag{1}$$

$$|\mathcal{N}_k(n)| = \frac{1}{n} \sum_{j=1}^{n} k^{gcd(j,n)} = \frac{1}{n} \sum_{d|n} \phi(d) k^{\frac{n}{d}} \tag{2}$$

$$|\mathcal{P}_k(n)| = \sum_{j=1}^{n} |\mathcal{L}_k(j)| \tag{3}$$

Proof: Let $\mathcal{A}_k(n)$ be the set of all aperiodic strings of length n over Σ_k. But every string in Σ_k^n is an "integral power" of some aperiodic string, so $k^n = \sum_{d|n} |\mathcal{A}_k(d)|$. Apply Möbius inversion to obtain $|\mathcal{A}_k(n)| = \sum_{d|n} \mu\left(\frac{n}{d}\right) k^d$. Now note that $|\mathcal{A}_k(n)| = n|\mathcal{L}_k(n)|$ because all n circular shifts of an aperiodic string produce distinct results. Equation (1) follows.

Equation (2) is proved with Burnside's lemma. Necklaces are obtained as a result of the action of the cyclic group $\mathbb{C}_n$ on all strings in Σ_k^n. Let σ be the function *cyclic shift left by one position*. Then $\mathbb{C}_n = \{\sigma^0, \sigma^1, \ldots, \sigma^{n-1}\}$. We are interested in the fixpoints; more precisely, we would like to determine for how many strings α it is the case that $\sigma^j(\alpha) = \alpha$. Use Lemma 1 and conclude that can occur only if $\alpha = \beta^t$ where β is aperiodic, *i.e.* $\beta \in \mathcal{L}$, and $|\beta| = \gcd(n, j)$. The number of these strings is clearly $k^{|\beta|} = k^{\gcd(j,n)}$. By Burnside's lemma, the number of the equivalence classes is $\sum_{j=1}^{n} k^{\gcd(j,n)}$ divided by the cardinality of $\mathbb{C}$, *i.e.* by n.

The fact that $\sum_{j=1}^{n} k^{\gcd(j,n)} = \sum_{d|n} \phi(d) k^{\frac{n}{d}}$ follows from equation (2.11) in [Rus03]. It says that:

$$\sum_{j=1}^{n} f(\gcd(n, j)) = \sum_{d|n} \phi\left(\frac{n}{d}\right) f(d) = \sum_{d|n} \phi(d) f\left(\frac{n}{d}\right)$$

The derivation uses the fact that $d = \gcd(n, j)$ iff $1 = \gcd\left(j, \frac{n}{d}\right)$, *i.e.* j and $\frac{n}{d}$ are relatively prime; having noted that, the derivation is trivial because the three sums describe different arrangements of the same summands. The preliminaries of (2.11) do not specify anything about f. We can safely assume f is any arithmetic function.

The proof of (3) relies on Lemma 4. According to Lemma 4, each pre-necklace is a prefix of some string β^* where β is a Lyndon word. Furthermore,

each prenecklace is uniquely determined by its longest prefix that is a Lyndon word. Having that in mind, (3) follows immediately. □

Theorem 2 (Theorem 7.4 in [Rus03]). *The following conditions characterise* $\mathcal{N}$ *and* $\mathcal{L}$. *Suppose* $\alpha \in \Sigma_k^+$.

1. $\alpha \in \mathcal{N}$ *iff for any factorisation of* α, *say* $\alpha = xy$, *it is the case that* $xy \preceq yx$.

2. $\alpha \in \mathcal{L}$ *iff for any nontrivial factorisation of* α, *say* $\alpha = xy$, *it is the case that* $xy \prec yx$.

Proof: The first statement follows immediately from the definition of "necklace". We prove the second statement.

In one direction, assume α is a Lyndon word. Then assume there exists a nontrivial factorisation of α, say $\alpha = xy$, such that $xy \nprec yx$. But that is equivalent to $xy \succeq yx$. The following two cases are exhaustive.

* If $xy \succ yx$ then there is a rotation of α, namely yx, that is smaller than α and thus $\alpha \notin \mathcal{N}$. Then $\alpha \notin \mathcal{L}$, contrary to the former assumtion that α is a Lyndon word.

* If $xy = yx$ we conclude α is periodic – that follows from the assumption the factorisation xy is nontrivial and Corollary 1. That conclusion, however, contradicts the former assumption that α is a Lyndon word.

In the other direction, assume for every nontrivial factorisation of α, say $\alpha = xy$, it is the case that $xy \prec yx$. Assume that $\alpha \notin \mathcal{L}$. However, it must be the case that $\alpha \in \mathcal{N}$ since $xy \prec yx$ for every nontrivial factorisation xy of α. It must be the case that α is periodic. But then there exists a nontrivial factorisation of α, say $\alpha = x'y'$, such that $\alpha = y'x'$. Then $x'y' = y'x'$ in contradiction of the assumption that for every nontrivial factorisation of α, say $\alpha = xy$, it is the case that $xy \prec yx$. □

Lemma 2 (Lemma 7.2 in [Rus03]). *If* $\alpha \in \mathcal{N}$, *then* $\alpha^t \in \mathcal{N}$ *for all* $t \geqslant 1$.

Proof: If $t = 1$ there is nothing to prove. Assume $t > 1$. Consider any factorisation of α^t, say $\alpha^t = xy$. Clearly, yx has the form $\gamma\alpha^{t-1}\delta$ where $\delta\gamma$ is a factorisation of α. But α is a necklace and therefore $\delta\gamma \preceq \gamma\delta$. It follows that $(\delta\gamma)^t \preceq (\gamma\delta)^t$. The left-hand side equals α^t. The right-hand side equals $\gamma\alpha^{t-1}\delta$. Then $\alpha^t \preceq \gamma\alpha^{t-1}\delta$. Then $xy \preceq yx$. Use Theorem 2 and conclude α^t is a necklace. □

Lemma 3 (Lemma 7.3 in [Rus03]). *If* $\alpha \in \mathcal{L}$ *and* α *has factorisation* $\alpha = \beta\gamma$ *such that* $\gamma \neq \epsilon$, *then for any* $t \geqslant 1$:

1. $\alpha^t\beta \in \mathcal{P}$, *and*

2. $\alpha^t\beta \in \mathcal{N}$ *iff* $\beta = \varnothing$.

Proof: Consider the first claim. α is a necklace because it is a Lyndon word. Use Lemma 2 to conclude both α^t and α^{t+1} are necklaces. Then clearly $\alpha^t\beta$ is a necklace since β is a prefix of α.

Consider the second claim. If $\beta = \epsilon$ then obviously $\alpha^t\beta = \alpha^t$ and it is a necklace. Now assume $\beta \neq \epsilon$. Recall that $\beta\gamma$ is a factorisation of α and α is a Lyndon word. By Theorem 2 it is the case that $\beta\gamma \prec \gamma\beta$. Then $(\beta\gamma)^t \prec (\gamma\beta)^t$ and $\beta(\beta\gamma)^t \prec \beta(\gamma\beta)^t$. But $\beta(\gamma\beta)^t = \beta\alpha^t$ and $\beta(\beta\gamma)^t = (\beta\gamma)^t\beta = \alpha^t\beta$. It follows $\beta\alpha^t \prec \alpha^t\beta$ and thus $\alpha^t\beta$ is not a necklace by Theorem 2. □

For any nonempty string α, let $\mathrm{lyn}(\alpha)$ be the length of the longest prefix of α that is a Lyndon word. Since a single letter is a Lyndon word, $\mathrm{lyn}(\alpha) \geqslant 1$.

Lemma 4 (Lemma 7.4 in [Rus03])**.** *Let* $\alpha = a_1a_2\cdots a_n$ *be a string and* $p = \mathit{lyn}(\alpha)$. $\alpha \in \mathcal{P}$ *iff* $a_{j-p} = a_j$, *for* $j = p+1, p+2, \ldots, n$.

Proof, part 1: Assume

$$\begin{aligned} a_1 &= a_{p+1} \\ a_2 &= a_{p+2} \\ &\cdots \\ a_{n-p} &= a_p \end{aligned}$$

Let the longest prefix of α that is a Lyndon word be σ. Clearly, $|\sigma| = p$. Then α has a factorisation $\sigma^t z$ for some $t \geqslant 1$ where z is a prefix of α. In other words, from left to right, α is a repetition of the same Lyndon word σ followed by some suffix z that is a prefix of α. Use Lemma 3 to conclude α is a prenecklace.

Proof, part 2: Assume α is a prenecklace. We have to show that $a_1 = a_{p+1}$, $a_2 = a_{p+2}$, *etc.*, $a_{n-p} = a_p$. Assume the opposite.

Define j to be the smallest index such that $p+1 \leqslant j \leqslant n$ and $a_{j-p} \neq a_j$. Let the longest prefix of α that is a Lyndon word be σ. Then, for some $t \geqslant 1$, α has the form:

$$\alpha = \sigma^t\delta a_j a_{j+1}\cdots a_n$$

where δ is a proper prefix of α. Every letter from δ matches the letter that is at distance p left of it (that letter is in the rightmost copy of σ) and the "problematic" letter a_j does *not* match the letter that is at distance p to the

left of it (that letter is a_{j-p} and is in the rightmost copy of σ, too). It follows $\sigma = \delta\gamma$ for some nonempty suffix γ where a_{j-p} is the first letter of γ.

Assume $a_{j-p} > a_j$. Since $\alpha \in \mathcal{P}$ there is a string η such that $\alpha\eta$ is a necklace. Consider $\alpha\eta$:

$$\alpha\eta = \underbrace{\sigma\sigma\cdots\sigma}_{\text{t factors}}\delta a_j a_{j+1}\cdots a_n\eta$$

Consider the rightmost σ in details, noting the rightmost letter of the rightmost σ is a_{tp}:

$$\alpha\eta = \underbrace{\sigma\sigma\cdots\sigma}_{\text{t}-1\text{ factors}}\underbrace{\delta\underbrace{a_{j-p}\cdots a_{tp}}_{\gamma}}_{\text{the rightmost }\sigma}\delta a_j a_{j+1}\cdots a_n\eta$$

The "problematic" pair of symbols a_{j-p}, a_j that do not match is coloured. Now imagine that $\alpha\eta$ is cyclically shifted left p positions. Call the result τ:

$$\tau = \underbrace{\sigma\sigma\cdots\sigma}_{\text{t}-2\text{ factors}}\underbrace{\delta\underbrace{a_{j-p}\cdots a_{tp}}_{\gamma}}_{\sigma}\delta a_j a_{j+1}\cdots a_n\eta\sigma$$

Compare τ with $\alpha\eta$. The red letter a_j in τ is precisely in the same position as the green letter a_{j-p} in $\alpha\eta$. To see why, note that the prefix of τ left of the red letter a_j is precisely the same as the prefix of $\alpha\eta$ left of the green letter a_{j-p} – in both strings, the said prefix is $\sigma^{t-1}\delta$. So, the result of the comparison of both strings depends on the comparison between a_j and a_{j-p}. However, we assumed that $a_{j-p} > a_j$. It follows that $\tau < \alpha\eta$. Then a nontrivial rotation of $\alpha\eta$, namely τ, is lexicographically smaller than $\alpha\eta$, and by Theorem 2, $\alpha\eta$ is not a necklace.

In the remainder of the proof assume $a_{j-p} < a_j$. Consider the prefix of α up to and including a_j. Call that prefix ρ. Then $\rho = \sigma^t\delta a_j$. Ignore the rest of α and focus on ρ only. Note that $|\rho| > |\sigma|$ because even if t is as small as 1 and δ is empty, still ρ has at least one more letter than σ, namely a_j. Recall that σ is the longest prefix of α that is a Lyndon word. The proof proceeds by analyses of cases and subcases but the idea is always the same: show that ρ is a Lyndon word, in contradiction of the earlier assymption that the longest prefix of α that is a Lyndon word is σ. The tool we use to show that ρ is a Lyndon word is Theorem 2: we consider an arbitrary nontrivial factorisation xy of ρ and show that $xy < yx$; by Theorem 2 that means ρ is a Lyndon word. So, consider an arbitrary nontrivial factorisation $\rho = xy$.

As $\rho = \sigma^t \delta a_j$, there are two possibilities for x relative to $\sigma^t \delta a_j$: σ^t is a prefix of x or x is a prefix of σ^t.

Case 1: σ^t is a prefix of x. Then $x = \sigma^t u$ where u may or may not be empty. Then $y = v a_j$ where, clearly, $uv = \delta$. y cannot be empty because a_j is in y. We already note that δ is a proper prefix of σ. In the current notation, uv is a proper prefix of σ. It follows $\sigma = uvw$ for some nonempty string w. Furthermore, a_{j-p} is the first letter of w because a_{j-p} is the first letter right of δ in σ.

- Assume $u \neq \epsilon$. But $uvw \prec vwu$ because uvw, that is σ, is a Lyndon word and $\boxed{u}\boxed{vw}$ is a nontrivial factorisation of its; recall that both u and vw are nonempty. However, $vwu \prec v a_j$ because the first letter of w is a_{j-p} and we did assume $a_{j-p} < a_j$. By the transitivity of $\prec$ we infer $uvw \prec v a_j$. Then

$$\underbrace{\underbrace{uvw}_{\sigma}\, \sigma^{t-1}\delta a_j}_{xy} \prec \underbrace{v a_j}_{y}\, \underbrace{\sigma^t u}_{x}$$

  We proved that $xy \prec yx$ under the current assumptions.

- Assume $u = \epsilon$. In this subcase $uvw = vw$. We want to prove that $uvw \prec v a_j$, that is, $vw \prec v a_j$. But that is true for precisely the same reason as above: the first letter of w is smaller than a_j. Having shown that $uvw \prec v a_j$, we derive $xy \prec yx$ precisely as above.

Case 2: x is a prefix of σ^t. Then $x = \sigma^{m_1} u$ and $y = v\sigma^{m_2}\delta a_j$ where uv is a factorisation of σ and $m_1 + m_2 = t - 1$.

- $u = \epsilon$ or $v = \epsilon$. These two possibilities are effectively the same thing because both of them imply the "slice" between x and y does not "cut" any σ in two (nonempty) parts. So, without loss of generality, assume $u = \epsilon$ and thus $v = \sigma$. Then

$$xy = \underbrace{\sigma^{m_1}\epsilon}_{x}\, \underbrace{\sigma\sigma^{m_2}\delta a_j}_{y} = \sigma^{m_1}\sigma^{m_2+1}\delta a_j$$

  We claim that

$$\underbrace{\sigma^{m_1}}_{x}\, \underbrace{\sigma^{m_2+1}\delta a_j}_{y} \prec \underbrace{\sigma^{m_2+1}\delta a_j}_{y}\, \underbrace{\sigma^{m_1}}_{x}$$

But that follows immediately from the observation that the letter a_j in the right-hand side is matched against a letter a_{j-p} in the left-hand side; this a_{j-p} is either in σ^{m_1} or in σ^{m_2+1}. To see why a_j is matched against a_{j-p} note that the prefix $\sigma^{m_2+1}\delta$ is found as a prefix of the left-hand side where it is being followed by an a_{j-p} letter. Since $a_{j-p} < a_j$, it follows the left-hand side is indeed smaller in the lex order.

Now observe the left-hand side is xy and the right-hand side is yx. Once again we showed that $xy \prec yx$.

$\star$ $u \neq \epsilon$ and $v \neq \epsilon$ and thus uv is a nontrivial factorisation of σ.

$\star\star$ Assume $m_2 = 0$. The string $\rho = xy$ looks like:

$$xy = \underbrace{\sigma^{m_1}u}_{x}\,\underbrace{v\delta a_j}_{y} = \underbrace{\sigma\sigma^{m_1-1}u}_{x}\,\underbrace{v\delta a_j}_{y}$$

Clearly, yx looks like:

$$yx = \underbrace{v\delta a_j}_{y}\,\underbrace{\sigma^{m_1}u}_{x}$$

Note that xy starts with a σ and when we compare xy with yx to see which one is lexicographically smaller we compare that σ with $v\delta a_j$. Note that both u and δ are prefixes of σ and we do not know which one of them is longer.

$\star\star\star$ δ is a proper prefix of u and that means δa_j is not longer than u. So, when we compare xy with yx, the string δa_j that yx begins with is matched against the prefix u of the leftmost σ of xy. But $\sigma \in \mathcal{L}$ and uv is a nontrivial factorisation of σ, therefore $uv \prec vu$ by Theorem 2. Furthermore, it must be the case that $vu \prec v\delta a_j$ because they have a common prefix and $u \prec \delta a_j$ since δ is a proper prefix of u and a_j is matched against a smaller letter in u, namely one that equals a_{j-p}.

$\star\star\star$ δ is not a proper prefix of u and that means δa_j is longer than u. When we compare xy with yx, the string σ that xy begins with is matched against the prefix $v\delta$ of yx. But now δ is at least as long as u so we effectively compare the prefix σ of xy with the prefix vu of yx. Since $\sigma = uv$ and $uv \prec vu$, the desired result follows.

$\star\star$ Now assume $m_2 > 0$. We can rewrite

$$xy = \sigma^{m_1}uv\sigma^{m_2}\delta a_j = u(vu)^{m_1}(vu)^{m_2}v\delta a_j = u(vu)^{m_1+m_2}v\delta a_j$$

Let us compare xy with yx. That is, compare

$$uvu(vu)^{m_1+m_2-1}v\delta a_j \quad vs \quad vuv(uv)^{m_2-1}\delta a_j\sigma^{m_1}u$$

We can rewrite xy and yx in this way because $m_2 > 0$. However, $uv \prec vu$ because $uv = \sigma$ is a Lyndon word. Then certainly $uvz_1 \prec vuz_2$ for any strings z_1, z_2. Thus we demontrated that $xy \prec yx$. □

The following theorem is called "The Fundamental Theorem of Necklaces" in [Rus03].

Theorem 3 (Theorem 7.5 in [Rus03]). *Let* $\alpha = a_1a_2\cdots a_{n-1} \in \mathcal{P}_k(n-1)$. *Let* $p = lyn(\alpha)$. *Let* $b \in \Sigma_k$. *Then* $\alpha b \in \mathcal{P}_k(n)$ *iff* $b \in \{a_{n-p}, a_{n-p}+1, \ldots, k-1\}$. *Furthermore,*

$$lyn(\alpha b) = \begin{cases} p, & \text{if } b = a_{n-p} \\ n, & \text{if } b > a_{n-p} \end{cases}$$

Proof: Follows right away from the proof of Lemma 4. □

The recursive FKM (Fredricksen, Kessler, Maiorana) algorithm that generates necklaces or prenecklaces or Lyndon words or De Brujin sequences, in lex order. Its correctness can be proved using Theorem 3 and Lemma 5 below. Assume k and n are global variables, the string $a_1a_2\cdots a_n$ is globally visible, and $a_0 = 0$ is appended at the beginning.

```
FKM(t, p ∈ ℕ)
 1  if t > n
 2      PrintIt(p)
 3  else
 4      a_t ← a_{t−p}
 5      FKM(t + 1, p)
 6      for j ← a_{t−p} to k − 1
 7          a_t ← j
 8          FKM(t + 1, t)
```

The initial call is, of course, FKM(1, 1). The function PrintIt can be implemented in four different ways, each implementation outputting one type of objects from necklaces, prenecklaces, Lyndon words or De Brujin sequences.

* If we wish prenecklaces then PrintIt(p) is Println($a_1a_2\cdots a_n$), ignoring p.

* If we wish necklaces then PRINTIT(p) is
  if $n \bmod p = 0$ **then** PRINTLN($a_1 a_2 \cdots a_n$) **else** **do-nothing**.

* If we wish Lyndon words then PRINTIT(p) is
  if $p = n$ **then** PRINTLN($a_1 a_2 \cdots a_n$) **else** **do-nothing**.

* If we wish De Bruijn sequences then PRINTIT(p) is
  if $n \bmod p = 0$ **then** PRINT($a_1 a_2 \cdots a_p$) **else** **do-nothing**.

The following statement in [Rus03] is not called "lemma" or "proposition" or "theorem" but merely "definition" (Definition 7.1, to be precise). However, it is not a definition by its nature because the successor function is not just any function – it is well-defined by a definition that is completely different and is based on the definitions of "prenecklace" and "lex order". Here we claim a certain property holds for the successor function. The proof follows from Lemma 4.

Consider the list formed by the elements of $\mathcal{P}_k(n)$ in the lex order. Clearly, the first element is 0^n. For any $\alpha \in \mathcal{P}_k(n)$ such that $\alpha \prec (k-1)^n$, succ(α) is the string that follows immediately α in the said list.

Lemma 5 (Definition 7.1 in [Rus03])**.** *Let* $\alpha = a_1 a_2 \cdots a_n$ *and* $0^n \preceq \alpha \prec (k-1)^n$. *Then*

$$succ(\alpha) = (a_1 a_2 \cdots a_{i-1}(a_i + 1))^t a_1 a_2 \cdots a_j$$

where i *is the largest integer such that* $1 \leqslant i \leqslant n$ *and* $a_i < k-1$, $t = \left\lfloor \frac{n}{i} \right\rfloor$, *and* $j = n \bmod i$.

We also define the predecessor function pred(α). *If succ*(α) $= \beta$, *then pred*(β) $= \alpha$. □

3 Theoretical Results on De Bruijn Sequences

A De Bruijn cycle, or, alternatively, *a cyclic De Bruijn sequence*, or shortly *a De Bruijn sequence*, over Σ_k and relative to parameter n, is a cyclic string S such that every element of Σ_k^n occurs in it precisely once and each letter in S is the first letter of one element of Σ_k^n. Intuitively, assuming we "drag" a window of width n over S, we see in this window every possible k-ary string of length n precisely once. Relative to $k = 2$ and $n = 3$, an example of De Bruijn cycle is 00011101.

The length of any De Bruijn cycle—we still have not proved existence—relative to k and n is k^n because that is the number of k-ary strings of length n and each letter of the cycle is the start of precisely one of them.

The theoretical foundation for the generation of De Bruijn cycles by FKM is Theorem 4.

Theorem 4 (Theorem 7.6 in [Rus03])**.** *The list of successive periodic reductions of necklaces as produced by the FKM algorithm forms a De Bruijn cycle.*

Proof: Let us first compute the length of the output of FKM$(1,1)$ relative to some k and n when the algorithm is tuned (via function PrintIt) to output De Bruijn cycle. The output is a single string that is a concatenation of the outputs of all individual calls of PrintIt for, and only for, the values of p that divide n. Let us call that string D. Each such output of PrintIt has length p. For each p that divides n the number of outputs is $|\mathcal{L}_k(p)|$; on the other hand, if p does not divide n then nothing is output. Thus:

$$|D| = \sum_{p|n} p|\mathcal{L}_k(p)|$$

Let $\mathcal{A}_k(n)$ denote the set of all aperiodic strings of length n over Σ_k – as in the proof of Theorem 1. As we said there, $|\mathcal{A}_k(n)| = n|\mathcal{L}_k(n)|$. So we can rewrite the formula for $|D|$ thus:

$$|D| = \sum_{p|n} |\mathcal{A}_k(p)|$$

As we said in the proof of Theorem 1, the right-hand side of this formula equals k^n. It follows that $|D| = k^n$. On the other hand, every De Bruijn cycle over Σ_k and relative to n has length k^n. We see that at least the length of D is right. Next we prove that indeed every k-ary string of length n occurs precisely one in this output. Clearly, it suffices to show that every such string occurs at least one because of the already proved fact that $|D| = k^n$.

The first two outputs are 0 and $0^{n-1}1$, in that order. Thus 0^n1 is a prefix of D. The last two outputs are $(k-2)(k-1)^{n-1}$ and $(k-1)$. Thus $(k-2)(k-1)^n$ is a suffix of D. Since D is a circular string, it contains $0^n(k-1)^n$ as a substring. It follows that all strings $0^p(k-1)^{n-p}$ occur in D, for $0 \leqslant p \leqslant n$. So far we have proved the occurrence of $n+1$ strings in D.

The key observation is that *every other string* has the form $(k-1)^p(xy)^t$ where $0 \leqslant p \leqslant n-1$, $t \geqslant 1$, x starts with a letter that is not $(k-1)$, xy contains a non-zero letter, and—that is crucially important—$yx \in \mathcal{L}$. The remainder of the proof is in eight cases because we consider three orthogonal possibilities:

* $p = 0$ or $p > 0$,
* $t = 1$ or $t > 1$,
* $y = \epsilon$ or $y \neq \epsilon$.

Case 1: $p = 0, t = 1, y = \epsilon$**.** Clearly, $\alpha = x$. Since $yx \in \mathcal{L}$ and $y = \epsilon$ it must be the case that $x \in \mathcal{L}$. So, in this case we consider the Lyndon words. We already know that FKM outputs all Lyndon words so there is nothing more to prove here.

Case 2: $p = 0, t = 1, y \neq \epsilon$**.** Now $\alpha = xy$ and $yx \in \mathcal{L}$. We know that FKM outputs every Lyndon word over Σ_k of length $\leqslant n$ and so it outputs yx.

What is the next output after yx? Note that $\text{succ}(yx) = yz$ for some string z since not every letter in x is $(k-1)$. Why? Because at least one letter of x is not $(k-1)$. Recall the essence of the successor function from Lemma 5: the rightmost symbol that is not $(k-1)$ is increased by 1. Surely that symbol is in x as we consider $\text{succ}(yx)$ and therefore the prefix y is left intact by the successor function. What is the periodic reduction of yz?

We prove the periodic reduction of yz is a string with prefix y. Indeed, it cannot be the case that $yz = \beta^r$ for some β that is a proper prefix of y because:

* By Theorem 3 we know that if β is a Lyndon word, then β^r is the smallest necklace of length $r|\beta|$ with prefix β in the lex order.
* yx is a smaller that yz necklace in the lex order since $yz = \text{succ}(yx)$; if yz were β^t for some proper prefix β of y, then $yz = \beta^r$ would be lexicographically smaller than yx.

We conclude it cannot be the case that after yx the next output string is a mere proper prefix of y. It follows that y appears right after yx in D and thus xy is a substring of D. That is, α is a substring of D.

Case 3: $p = 0, t > 1, y = \epsilon$**.** Now $\alpha = x^t$ and $x \in \mathcal{L}$. As $x \in \mathcal{L}$, x is output and it appears somewhere in D.

What string is output next after x? We claim that string is $x^{t-1}S(x)$ where $S(x)$ is *the necklace*[†] that is right after x in the lex order. Furthermore,

[†] $S(x)$ is not necessarily the same thing as $\text{succ}(x)$. The latter is the prenecklace that is immediately after x in the lex order. The former is the necklace that first appears after x in the lex order. E.g., let $k = 3$ and $n = 5$. Then 01002 is a Lyndon word. succ(01002) = 01010 but 01010 is not output because it is merely a prenecklace and not a necklace. The necklace that follows 01002 is 01011 and so 01011 is output right after 01002.

$(t-1)|x| + |S(x)| = n$. Why is that the case? Recall how PRINTIT operates when it is tuned for De Bruijn cycles: if $\alpha = a_1 \cdots a_n$ is a periodic necklace, that is, if p divides n, then only the periodic reduction $a_1 \cdots a_p$ is output. That periodic reduction is x because $x \in \mathcal{L}$ and thus it is "atomic". Next the string $a_1 \cdots a_n$ is "incremented" so that what was the rightmost periodic reduction, that is the rightmost copy of x, gets incremented to its successor succ(x). Now the whole string is aperiodic and the next call of PRINTIT outputs an n-letter output.

We showed that the next output after x has prefix x^{t-1}. This means that xx^{t-1} occurs as a substring in D. It follows that α is a substring of D.

Case 4: $p = 0, t > 1, y \neq \epsilon$**.** Now $\alpha = (xy)^t$ and $yx \in \mathcal{L}$. Being a Lyndon word, yx is output by FKM.

What is the next string that is output? Similarly to the previous case, the next string that is output is the aperiodic $S((yx)^t) = (yx)^{t-1}y\,S(x)$. To see why that is the case, recall that x has a letter that is not $(k-1)$ and so the successor function increments a letter in the rightmost copy of x; thus the rightmost y is left intact.

We conclude that $yx(yx)^{t-1}y\,S(x)$ is a substring of D. But that contains $(xy)^t = \alpha$ as a substring.

Case 5: $p > 0, t = 1, y = \epsilon$**.** Now $\alpha = (k-1)^p x$ and $x \in \mathcal{L}$. Consider the string pred(x)$(k-1)^p$ and call it β. Here, pred( ) is the function defined in Lemma 5. Note that pred(x) exists because x cannot be all zeroes and the pred( ) function is defined for all strings except for the all-zeroes string. And x cannot be all zeroes because xy, which in this case is x, by the initial assumptions has a non-zero letter in it.

But D contains β as a substring because $\beta \in \mathcal{L}$. To see why β = pred(x)$(k-1)^p$ is a Lyndon word recall that x is a Lyndon word of length smaller than n.

Think of the successor of β. The letter in β that gets incremented is somewhere in pred(x) because the suffix $(k-1)^p$ has no letter to be incremented. So, the successor of β has prefix succ(pred(x)) and that equals x.

That means D contains a substring pred(x)$(k-1)^p x$ and thus contains as a substring $(k-1)^p x$. But then D contains as a substring α because $\alpha = (k-1)^p x$.

Case 6: $p > 0, t = 1, y \neq \epsilon$**.** Now $\alpha = (k-1)^p xy$ where $p \geqslant 1$ and $yx \in \mathcal{L}$. Let γ be the smallest circular shift of α in the lex order. We consider the following distinct possibilities for γ that are exhaustive.

* $\gamma = xy(k-1)^p$. In this case γ is a Lyndon word. We proceed exactly

as is **Case 5** with the current xy substituted for x in **Case 5**.

* $\gamma = y(k-1)^p x$.

  ** If γ is periodic then the periodic reduction must be $y(k-1)^p$ and $\gamma = (y(k-1)^p)^q$, which means $x = (y(k-1)^p)^{q-1}$. The algorithm outputs $\text{pred}(y)(k-1)^{n-|y|}$, followed by $y(k-1)^p$, followed by $(y(k-1)^p)^{q-1}S(y(k-1)^p)$. Then this is a substring of D:

  $$\text{pred}(y)(k-1)^{n-|y|}\, y(k-1)^p (y(k-1)^p)^{q-1} S(y(k-1)^p)$$

  In this subcase we have

  $$\alpha = (k-1)^p \underbrace{(y(k-1)^p)^{q-1}}_{x} y$$

  Clearly, D has α as a substring.

  ** If γ is aperiodic then γ is output by the algorithm. The next output has the form $y(k-1)^p z$. Then $y(k-1)^p xy(k-1)^p z$ is a substring of D and so α is a substring of D.

Case 7: $p > 0, t > 1, y = \epsilon$**.** Now $\alpha = (k-1)^p x^t$ and $x \in \mathcal{L}$. Note that $\beta = \text{pred}(x)(k-1)^{n-|x|}$ is a Lyndon word and, therefore, is found in D. Let $r = n \bmod |x|$ and $m = \left\lfloor \frac{n}{|x|} \right\rfloor$.

* If $r = 0$ then $|x|$ divides n. In this subcase the algorithm outputs x, which is a Lyndon word, followed by $x^{m-1}S(x)$. Then D contains the following string:

  $$\text{pred}(x)(k-1)^{n-|x|} x x^{m-1} S(x)$$

  We see D has a copy of α in it.

* If $r > 0$ then $|x|$ does not divide n then $\text{succ}(\beta)$, which is $x^m S(z)$, is a Lyndon word where z is the r-letter prefix of x. Since $m \geqslant t$, the string α occurs in D.

Case 8: $p > 0, t > 1, y \neq \epsilon$**.** Now $\alpha = (k-1)^p (xy)^t$ with $p > 0$, $t > 1$, and $yx \in \mathcal{L}$. As is **Case 7**, let γ be the lexicographically smallest circular shift of α. It must be the case that $\gamma = y(xy)^{t-1}(k-1)^p x$ and this is a Lyndon word. The next output string is $y(xy)^{t-1}(k-1)^p S(x)$—recall that x has a letter smaller than $(k-1)$. So, D contains this string as a substring:

$$y(xy)^{t-1}(k-1)^p xy(xy)^{t-1}(k-1)^p S(x)x$$

Clearly, D has a copy of α in it. □

References

[GKP94] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1994.

[Rus03] Frank Ruskey. Combinatorial generation, 2003.