# Combinatorics of Words (Fall 2019)
# Exercise sheet 2

Mathilde Bouvel

---

**Time-line:** This exercise sheet will be on-line on Thursday, December 19, 2019. A "questions and answers" session will be organized in January 2020 by Jacopo Borga (he will organize a doodle pool to fix the exact date). You should prepare this exercise sheet for this session. Indeed, part of this session will be dedicated to the solution of these exercises. There is no need to hand in your solution in advance (solutions are not graded), just bring it with you for the "questions and answers" session. If you are not able to attend this session, but would like to have feedback on your solution to this exercise sheet, please contact Jacopo.

---

**Exercise 1: Lyndon-preserving morphisms**
Let $\Sigma$ be a finite alphabet, totally ordered by $\leq$. We also denote by $\leq$ the associated lexicographic order on $\Sigma^*$, and write $<$ for the underlying "strict order".
Recall that $w \in \Sigma^*$ is a Lyndon word if and only if $w < v$ for every non-empty proper suffix $v$ of $w$.
We say that a morphism $\tau$ on $\Sigma^*$ is *Lyndon-preserving* when for all $w$ which are Lyndon words, $\tau(w)$ is also a Lyndon word. We also say that $\tau$ is order-preserving when $\tau(a) < \tau(b)$ for all $a, b \in \Sigma$ such that $a < b$.

The goal of this exercise is to show that a morphism $\tau$ is Lyndon-preserving if and only if $\tau$ is order-preserving and satisfies that $\tau(x)$ is a Lyndon word for every letter $x$ in $\Sigma$.

We first show the $\Rightarrow$ implication. Assume $\tau$ is a Lyndon-preserving morphism.

1. Justify that $\tau(x)$ is a Lyndon word for every letter $x$ in $\Sigma$.

   As a consequence, we note that $\tau$ is non-erasing (since $\varepsilon$ is not a Lyndon word.)

2. For any $a, b \in \Sigma$ such that $a < b$, show that $\tau(a) < \tau(b)$.
   *Hint:* Consider the image by $\tau$ of the Lyndon word $ab$.

We now show the $\Leftarrow$ implication. Assume $\tau$ is an order-preserving morphism such that $\tau(x)$ is a Lyndon word for every letter $x$ in $\Sigma$. As before, $\tau$ is necessarily non-erasing.

Let $w \in \Sigma^*$ be a Lyndon word. Our goal is to show that $\tau(w)$ is also a Lyndon word. For this, let $s$ be a non-empty proper suffix of $\tau(w)$; we want to prove that $\tau(w) < s$.

3. Assume first that there exist $u, v \in \Sigma^*$ such that $w = uv$ and $s = \tau(v)$. In this case, argue that $u$ is not empty, and prove that $\tau(w) < s$.

4. Otherwise, explain why $w$ can be written $w = uav$ for some $a \in \Sigma$ and $u, v \in \Sigma^*$, in such a way that there exists non-empty $p, q \in \Sigma^+$ satisfying $\tau(a) = pq$ and $s = q\tau(v)$.

5. Using that $\tau(a)$ is a Lyndon word, show that $\tau(av) < q\tau(v)$.

6. Recalling that $w$ is a Lyndon word, and that $\tau$ is order-preserving, conclude the proof that $\tau(w) < s$.

**Exercise 2: Another description of the Fibonacci word**
Let $\varphi$ denote the Fibonacci morphism (defined by $\varphi(a) = ab$ and $\varphi(b) = a$). Let also $f = \varphi^\omega(a)$ be the infinite Fibonacci word.
Define in addition the morphism $\rho$ by $\rho(a) = aab$ and $\rho(b) = ab$.

1. Prove by induction that for all $n \geq 0$, $\begin{cases} a\varphi^{2n}(a) & = \rho^n(a)a \\ a\varphi^{2n}(ba) & = \rho^n(ab)a \end{cases}$.

   *Hint:* Prove both statements in the same induction. For the induction step, a useful intermediate step in the computation is to establish that $a\varphi^{2n+2}(a) = a\varphi^{2n}(a)\varphi^{2n}(ba)$

2. Show that for all $n$, $\rho^n(a)a$ is a prefix of $\rho^{n+1}(a)$.

3. Explain why the word $\rho^\omega(a)$ can be defined, and deduce from the previous question that for all $n$, $\rho^n(a)a$ is a prefix of $\rho^\omega(a)$.

4. Show that for all $n$, $a\varphi^n(a)$ is a prefix of $\rho^\omega(a)$.
   *Hint:* Start with the case where $n$ is even. For the odd case, use a relation between $f_n = \varphi^n(a)$ and $f_{n+1} = \varphi^{n+1}(a)$ coming from the recursive definition of finite Fibonacci words.

5. Conclude that $af = \rho^\omega(a)$.

**Bonus exercise: a combination**
Using the results of Exercises 1 and 2, you can show that the word $af$ is an infinite Lyndon word. (Those are by definition the infinite words which are lexicographically strictly smaller than all their proper suffixes.)

**Exercise 3: Refined characterization of unbalanced words**
Recall that in the lecture, we proved the following:

> Let $\Sigma = \{a, b\}$ and $w \in \Sigma^\omega$. $w$ is unbalanced if and only if there exists a *word* $v \in \Sigma^*$ such that $ava$ and $bvb$ are both factors of $v$.

We now prove a refined statement, namely:

> Let $\Sigma = \{a, b\}$ and $w \in \Sigma^\omega$. $w$ is unbalanced if and only if there exists a *palindrome* $v \in \Sigma^*$ such that $ava$ and $bvb$ are both factors of $v$.

1. Briefly justify the $\Leftarrow$ implication.

   To prove the other implication, consider as in the lecture $u$ and $u'$ two factors of $w$ of the same length $n$ such that $\big||u|_a - |u'|_a\big| \geq 2$, and choose $n$ minimal with this property. Then, as in the lecture, we can prove that there exist words $v, s, s' \in \Sigma^*$ such that $u = avas$ and $u' = bvbs'$. (You don't need to rewrite this part of the proof.)

2. Explain why necessarily we have $s = s' = \varepsilon$. As a consequence, $u = ava$ and $u' = bvb$.

   The final step is to show that $v$ is a palindrome. We proceed by contradiction. So, assume $v$ is not a palindrome.

3. Considering the first letter where $v$ and $\overleftarrow{v}$ differ, argue that there exists $p \in \Sigma^*$ and $\alpha, \beta$ with $\{\alpha, \beta\} = \{a, b\}$ such that $p\alpha$ is a prefix of $v$ and $\beta\overleftarrow{p}$ is a suffix of $v$.

   This implies that $ap\alpha$ is a (proper) prefix of $u = ava$ and $\beta\overleftarrow{p}b$ is a (proper) suffix of $u' = bvb$.

4. Prove by contradiction that $\alpha = b$ and $\beta = a$.

   Consequently, $u = apbx$ and $u' = x'a\overleftarrow{p}b$ for some words $x, x' \in \Sigma^*$.

5. Derive from this a contradiction to the minimality of $n$.

6. Conclude the proof.

**Exercise 4: Recurrence of Sturmian words**
In the lecture, we stated (without proof) that Sturmian words are uniformly recurrent. In this exercise, we prove the weaker statement that Sturmian words are recurrent.
Let $w \in \Sigma^\omega$ for $\Sigma = \{a, b\}$ be a Sturmian word, and assume $w$ is not recurrent. This means that there exists a factor $u$ of $w$ which occurs only finitely many times in $w$. Denote by $n$ the length of $u$.

1. Show that it implies the existence of $p \in \Sigma^*$ and $w' \in \Sigma^\omega$ such that $w = pw'$ and $P_{w'}(n) \leq n$.
2. Which property of $w'$ does this imply?
3. Deduce that $w$ is ultimately periodic.
4. By deriving a contradiction, conclude the proof that $w$ is recurrent.

**Exercise 5: Equivalent definition of the Thue-Morse word**
In the last lecture of the semester, we defined the Thue-Morse word $t = \mu^\omega(a)$ for the morphism $\mu$ given by $\mu(a) = ab$ and $\mu(b) = ba$. We saw at the very beginning of the semester a different definition, namely that $t = t_0t_1t_2\dots$ where for all $n \geq 0$ $t_n = a$ (resp. $b$) if there is an even (resp. odd) number of 1 in the binary expansion of $n$. In this exercise, we prove that both definitions are equivalent.

We will make use of the "complement" function $\overline{\cdot}$ defined by $\overline{a} = b$ and $\overline{b} = a$.

For the rest of the exercise, we let $t = \mu^\omega(a)$, and we let $t = t_0t_1t_2\dots$ be the expression of $t$ as a concatenation of letters, namely, $t_n \in \{a, b\}$ for all $n \geq 0$.

1. Recalling that $t$ is a fixed point of $\mu$, and noting that the image of each letter by $\mu$ is a word of size 2, show that for all $n \geq 0$, $\mu(t_n) = t_{2n}t_{2n+1}$.
2. Looking at the definition of $\mu$, for $\alpha \in \{a, b\}$, what is the first letter of $\mu(\alpha)$? And what is the second letter?
3. Deduce that for all $n \geq 0$, $t_{2n} = t_n$ and $t_{2n+1} = \overline{t_n}$.
4. Prove by induction that for all $n \geq 0$, $t_n = a$ (resp. $b$) if there is an even (resp. odd) number of 1 in the binary expansion of $n$. In the induction step, it is useful to distinguish cases depending on whether $n$ is even or odd.