BULLETIN
OF THE BRAZILIAN
MATHEMATICAL
SOCIETY

Bull Braz Math Soc, New Series 34(3), 457-477

Symbolic analysis of finite words: the complexity function

Sébastien Jaeger, Ricardo Lima and Brigitte Mossé

Abstract. We present several properties of the complexity function of finite words, the function counting the number of different factors in a word, for each length. To establish a first set of properties, we use the de Bruijn graphs and the suffix tree representations of a word. This allows us to show some inequalities that control the variation as well as the maximal value of the complexity function. Motivated by the applications, we discuss the change of the complexity function when sliding or increasing the size of a window laid down on a sequence to be analysed.

Keywords: complexity, symbolic analysis, finite words.

Mathematical subject classification: Primary: 68R15; Secondary: 92B05, 92D20.

1 Introduction

For any given finite word ω, the complexity function quantifies the diversity of factors of the word ω. More precisely, for each natural number n (from 1 up to the length of ω), the complexity function is defined as the number of different factors of length n in ω. Such a function is well known in the literature, where the reader may find important results on this topic, see [11] and references therein, but mainly for the case of infinite sequences of symbols.

It was recently recognized that such a function, in the case of the analysis of a finite sequence in which we are interested, obeys certain constraints ([6],[5]). When using the complexity function to analyze finite symbolic sequences, it is important to notice first that there are some general properties of this function that are independent of any specific sequence.

Then we consider some other issues which are related with several applications: the study of the complexity of a window that is either moved along the sequence, or the size of which increases around a fixed position.

Received 14 March 2003.

We hope that the present work may be useful for a better understanding of the analysis of a specific sequence by means of the complexity function as well as for the tuning of the algorithms used to compute it.

The present paper is organized as follows. In **section 2** we recall some definitions, whereas **section 3** is devoted to a presentation of some useful tools for the study of the complexity function, together with some new basic general results. The tools include a graph representation of words (de Bruijn and Rauzy graphs) as well as a tree representation (the suffix tree). In **section 4** we prove a family of inequalities which gives a new insight on the still open problem of completely characterizing the set of all possible functions arising as the complexity function of a finite symbolic sequence. Typically, for a two letters alphabet, the ratio of the gradients of the complexity in two successive points is bounded by two; this fact implies, in particular, the known constraints on the variation of the complexity function. It is worth noticing that a word with the maximal complexity always exists, a fact that we also prove in this section. In the **last section** we prove some other new results which can be used in the analysis of the complexity of a window, which either is sliding along a sequence, or the width of which is changing. Besides their own interest, we will use these results in a forthcoming work, [7], when analyzing the entropy function of finite sequences.

2 Basic concepts and notation

Let us denote by $\mathcal{A}$ a finite alphabet of λ elements, called letters. Then, $\mathcal{A}^N$ stands for the set of all words of length N in $\mathcal{A}$. We shall write $\omega = \omega_0 \ldots \omega_{N-1}$, where $\omega_i \in \mathcal{A}$ and $0 \leq i \leq N-1$, and in such a case, we set $N = |\omega|$. For convenience, we also introduce the empty word ϵ, with $|\epsilon| = 0$.

The words $\omega[i, i+n-1] = \omega_i\omega_{i+1} \ldots \omega_{i+n-1}$ are the factors of ω, and are called prefixes when $i = 0$, and suffixes when $i+n = N$. We denote by $\mathcal{F}act(\omega)$ the set of all factors of a word ω, by $\mathcal{F}act_n(\omega)$ the subset of factors with a given length n , $\mathcal{P}ref(\omega)$ (resp. $\mathcal{S}uff(\omega)$) the set of all prefixes (resp. suffixes) of ω. By convention, we add the empty word ϵ to $\mathcal{F}act(\omega)$, $\mathcal{P}ref(\omega)$ and $\mathcal{S}uff(\omega)$. We also write $L_n(\omega)$ (resp. $R_n(\omega)$) to denote the prefix (resp. suffix) of length n of ω.

For each factor v of ω, $\mathcal{D}(v)$ (resp. $\mathcal{G}(v)$) denotes the set of all possible right (resp. left) extensions of v in $\mathcal{F}act_{|v|+1}(\omega)$, and we set $d(v) = |\mathcal{D}(v)|$ (resp. $g(v) = |\mathcal{G}(v)|$). A factor v is right (resp. left) special if $|d(v)| > 1$ (resp. $|g(v)| > 1$).

Finally, $occ_\omega(v)$ denotes the number of occurrences of v in ω.

The complexity function of ω is defined as

$$p_\omega(n) = \#\mathcal{F}act_n(\omega),\ \text{if } 0 \leq n \leq |\omega|$$

and

$$p_\omega(n) = 0,\ \text{if } n > |\omega|\ .$$

3 Useful tools and basic results

In this section we present some tools that are useful when dealing with the combinatorics of the factors of a given word, as well as when building algorithms used to quantify them. We also use such tools in order to show some new results.

3.1 The de Bruijn an Rauzy graphs

The reader may find in [2] the background in graph theory used in what follows.

It was N.G. de Bruijn ([3]) who first introduced a graph representation for the linking of a sequence of words; namely, for any integer k, he defined the oriented graph G_k whose vertices are the words in $\mathcal{A}^k$ and arrows are connecting a word $v \in \mathcal{A}^k$ to a word $v' \in \mathcal{A}^k$ when v' is suffix of va, for some letter $a \in \mathcal{A}$. In this case we say that a labels such an arrow. The graphs G_k are nowadays known as de Bruijn graphs.

The paths in G_k are in a one-to-one correspondence with the words of length $\geq k$, in a natural way: any path $(v_0, v_1, ..., v_r)$ whose arrows are labelled by $a_1, a_2, ..., a_r$ is associated with the word $\omega = v_0a_1...a_r$. The support of such a path is called the Rauzy graph of order k associated to ω ([12]).

It is clear how an arrow of G_k can be identified with a vertex of G_{k+1}, as it is clear that the incoming arrows in a vertex v represent $\mathcal{G}(v)$ and the outcoming arrows represent $\mathcal{D}(v)$.

The graph G_k is connected and pseudo-symmetric; therefore, by a classical theorem in graph theory, it contains Eulerian cycles (each arrow appears in the cycle exactly once). By the corresponding property in G_{k+1}, it contains also Hamiltonian cycles (passing through each vertex exactly once).

The following remark will be used in section 4.

Proposition 1. *Let H_k be a Hamiltonian cycle in G_k. Then, any connected component of the graph obtained from G_k by erasing the arrows of H_k has an Eulerian cycle.*

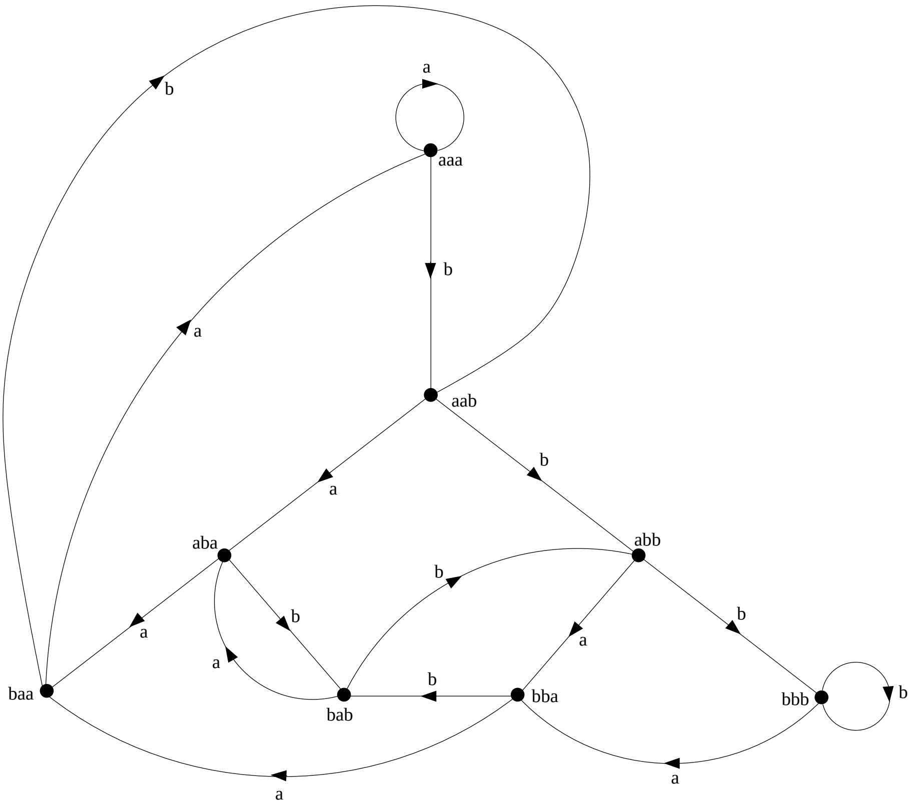

Figure 1: The graph G_3 with alphabet $\{a, b\}$.

Proof. The graph so obtained is again pseudo-symmetric, each vertex having exactly $(\lambda - 1)$ incoming and outgoing arrows. □

We shall now give examples of such situations. First, for a two letters alphabet $\{a, b\}$, we show two very different cases.

In figure 2 we can see that the complement of the Hamiltonian cycle H_3 in G_3 is connected, up to the vertices supporting the loops, whereas in figure 3 we see the complement of a Hamiltonian cycle H_4 in G_4, that has five different connected components. Because of the existence of loops, we never obtain a connected graph.

The following result shows that this situation cannot happen if the cardinality of the alphabet is at least three. We didn't find any convincing proof of this result in the literature.

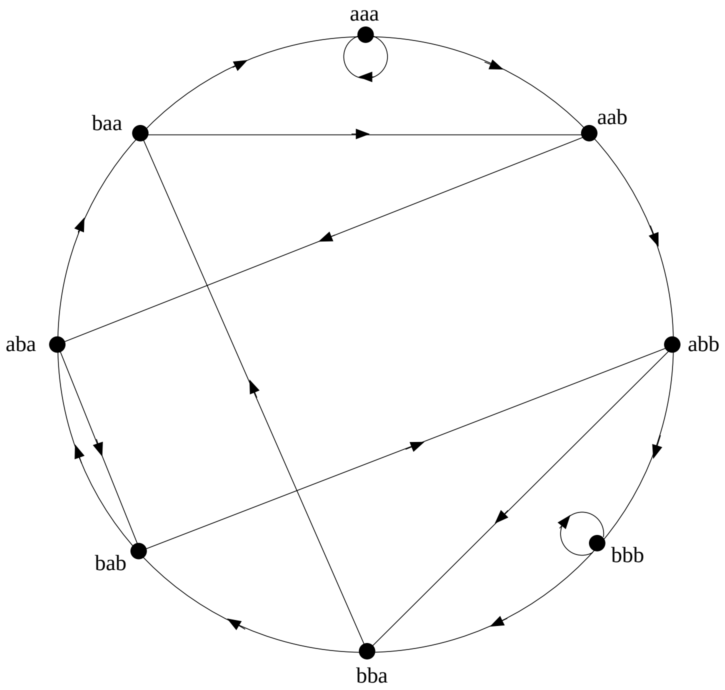

Figure 2: The graph G_3 with alphabet $\{a, b\}$ (representation showing a Hamiltonian cycle H_3.)

Proposition 2. *If $\#\mathcal{A} = \lambda$ is at least three, the graph G'_k obtained from G_k by erasing the arrows of a Hamiltonian cycle H_k is connected.*

Proof. Let uy be a vertex in G_k, with $|u| = k-1$ and $y \in \mathcal{A}$. By construction, there are $\lambda - 1$ letters x_i, $1 \leq i \leq \lambda - 1$, such that there exists an arrow $x_i u$ in G'_k arriving at uy.

Since $\lambda - 1$ is at least equal to two, for any couple (i, j) of different elements in $\{1, ..., \lambda - 1\}$ and any letter $z \in \mathcal{A}$, at least one of the words $x_i u$ or $x_j u$ is connected by an arrow in G'_k to uz. For if not, the cycle H_k would go through the vertex uz twice, which is not allowed. Therefore the words $x_i u$ $(1 \leq i \leq \lambda - 1)$ and all the words uz $(z \in \mathcal{A})$ are in the same connected component of G'_k. Alltogether, if $\alpha\omega\beta$ is in a connected component C of G'_k, with α and β in $\mathcal{A}$, then for any letter z, the word $\alpha\omega z$ is a vertex of C. But since ωz is in turn a prefix of a vertex of C, any element of $\mathcal{A}^k$ is in C, and consequently G'_k is connected. □

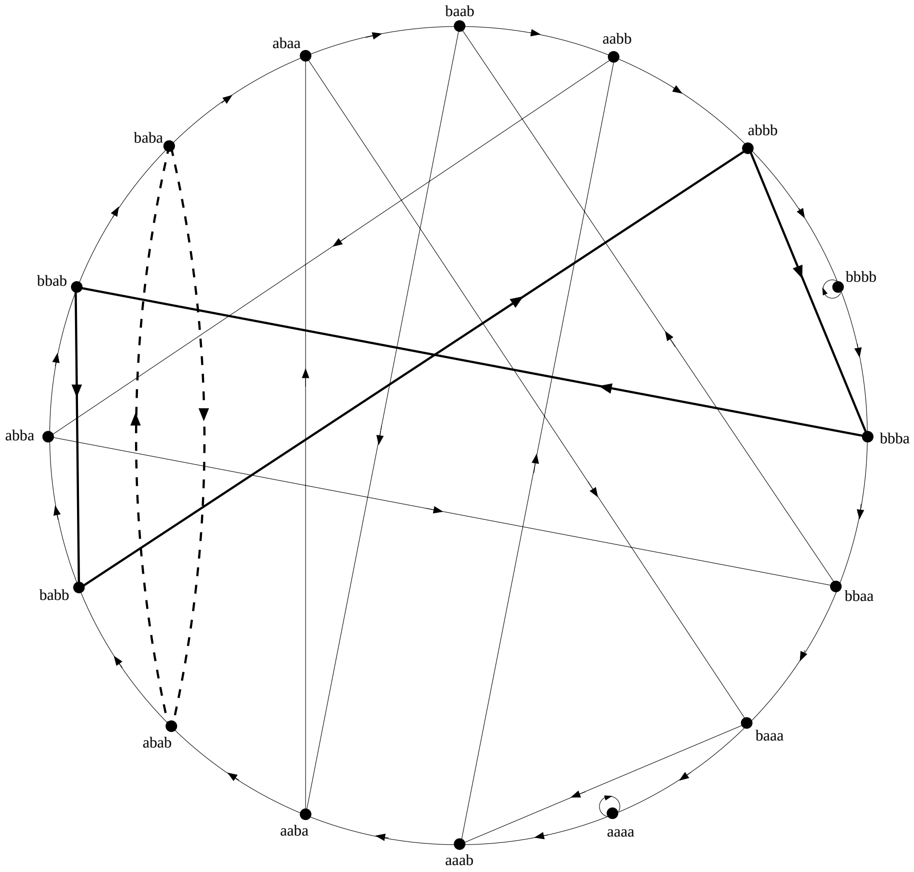

Figure 3: Graph G_4 with alphabet $\{a, b\}$

From Propositions 1 and 2 we deduce the following:

Corollary 1. *If $\#\mathcal{A} = \lambda$ is greater than or equal to three and ω is a word of lenght $\lambda^k + k - 1$ associated to a Hamiltonian cycle in G_k, then ω is a prefix of a word ω' of length $\lambda^{k+1} + k$ associated to a Hamiltonian cycle of G_{k+1}.*

This means that, when $\lambda \geq 3$, one can construct an infinite word ω such that, for each k, every factor of length k appears exactly once in the prefix of ω of length $\lambda^k + k - 1$.

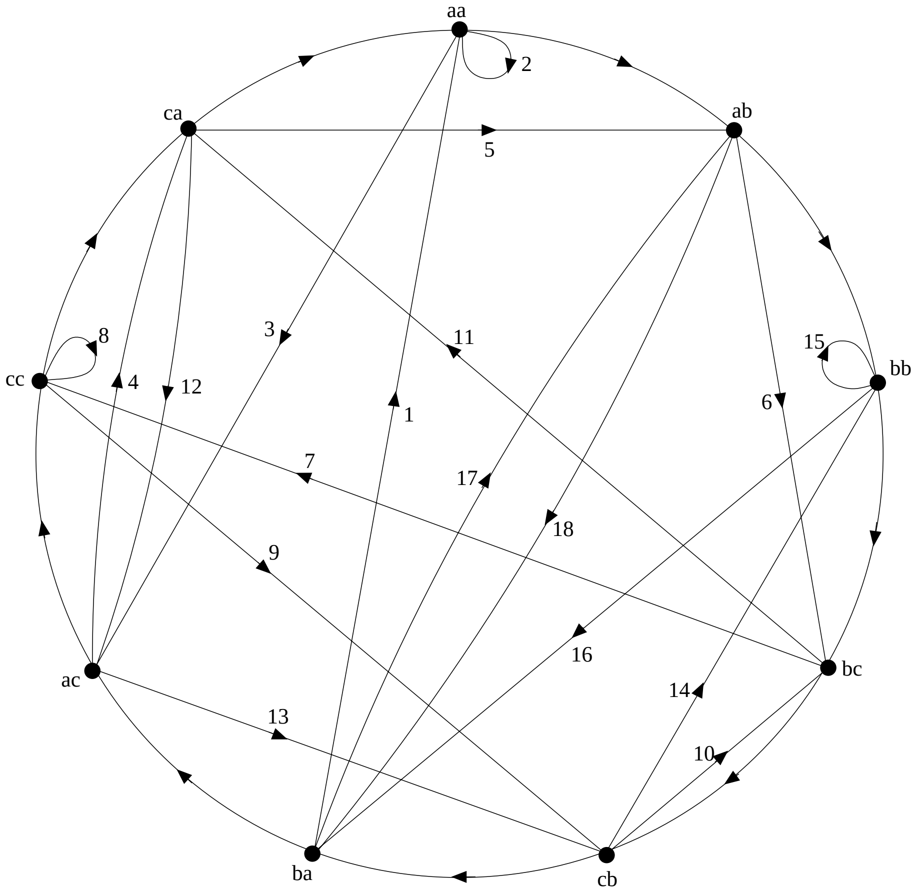

Figure 4: Graph G_2 with alphabet $\{a, b, c\}$: an Eulerian cycle is represented (follow the arrows from 1 to 17).

3.2 The suffix tree

Another classic representation of the structure of the set of factors of a given word is the suffix tree of ω. Here the symbol \$ designates a new letter which does not belong to the alphabet $\mathcal{A}$.

The suffix tree is a tree whose root is the empty word and leaves correspond to the factors of ω; an arrow labelled by a joins v to v' when $v' \in \mathcal{D}(\omega)$ and $v' = va$.

This representation, which is commonly used in algorithmic for the searching of patterns inside a text, as well as for data compression, may be presented as a suffix automaton; it is obtained from $ST(\omega)$ by deletion of the marker \$ and by identifying the factors of ω ending at the same ranks in ω. In figure 5 we give an example of such a construction.

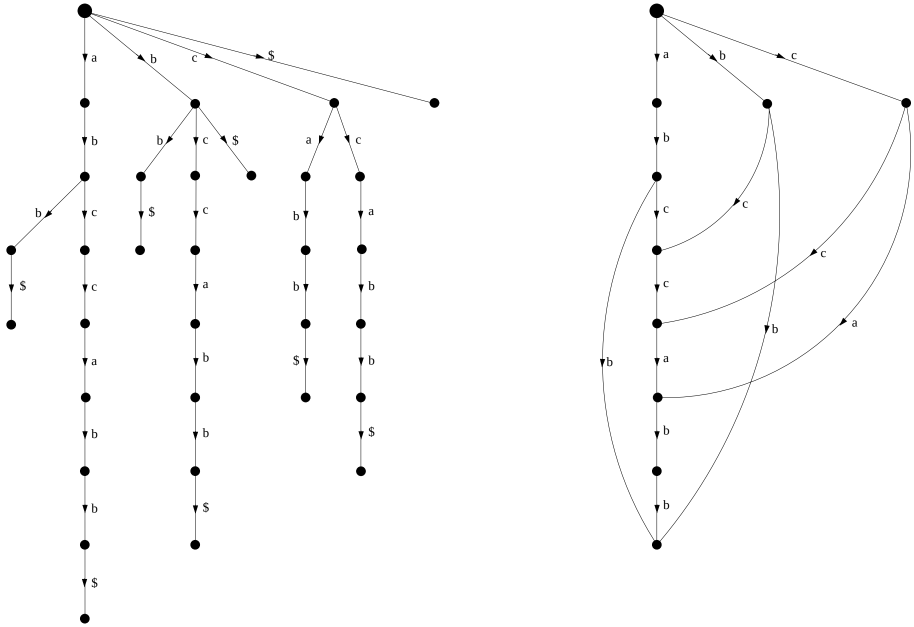

Figure 5: Suffix tree and suffix automaton of $\omega = abccabb$

For theoretical purposes, $ST(\omega)$ is certainly appropriate when working with right extensions of factors (or left extensions, after reversing ω), and when we are interested simultaneously in factors of ω of different lengths. In such cases, it is easier to handle $ST(\omega)$ than the corresponding path in the de Bruijn graph G_k. Therefore the choice among the two representations will depend of the context, a fact that we shall explore in section 3.

3.3 The particular factors $L(\omega)$ and $R(\omega)$

When studying the complexity of a finite word ω, there is a suffix that plays a special role; this is the shorter suffix of ω that appears only once in ω. Since $occ_\omega(\omega) = 1$, such a suffix always exists.

We denote it by $R(\omega)$, and by $r(\omega) = |R(\omega)|$ its length. Notice that $R(\omega)$ is also the shorter suffix of ω that has no right extension as a factor of ω. It also appears as the vertex that is the most distant from the root in the suffix automaton of ω. Notice also that, by construction, $R(\omega\$) = 1$. We similarly denote by $L(\omega)$ the shorter prefix of ω that appears only once in ω, and by $l(\omega) = |L(\omega)|$ the corresponding length.

We shall use such factors in a systematic way in what follows; the reader may also see [5] and references therein for other interesting applications of these factors.

We now introduce the following set:

$$E(\omega) = \{n \in \{1, \dots, |\omega| - 1\}\ ;\ L_n(\omega) = R_n(\omega) \text{ and } occ_\omega(L_n(\omega)) = 2\}\ .$$

Proposition 3.

1. *The set $E(\omega)$ has at most one element, denoted by $b(\omega)$.*

2. *If $E(\omega) \neq \emptyset$, then $b(\omega) < r(\omega)$ and $b(\omega) < l(\omega)$; more precisely $r(\omega) = l(\omega) = b(\omega) + 1$.*

Proof.

1. Let p and q be two different elements of $E(\omega)$ with, for instance, $p < q \leq |\omega|$; then $L_q(\omega) = R_q(\omega)$ and also $L_p(\omega) = R_p(\omega)$. But then $L_p(\omega)$ appears at least 3 times in ω, as prefix and suffix of ω and as suffix of $L_q(\omega)$. This implies that p cannot be an element of $E(\omega)$.

2. Clearly we have $b(\omega) < r(\omega)$ and $b(\omega) < l(\omega)$. Now, by definition of $b(\omega)$, the word $L_{b(\omega)}(\omega) = R_{b(\omega)}(\omega)$ appears twice in ω, one of which appearing as a suffix. Therefore $L_{b(\omega)+1}(\omega)$ is the unique right extension of $L_{b(\omega)}(\omega)$ and appears only once in ω. Finally we get $l(\omega) = b(\omega) + 1$ and, on the same way, $r(\omega) = b(\omega) + 1$. □

4 General properties of the complexity of finite words

4.1 The variations of the complexity function

The constraints appearing in the behavior of the complexity function of a finite word were investigated by several authors. For the time being, the full characterization of the functions from $\mathbb{N}$ to $\mathbb{N}$ that are the complexity function of a finite word, in a fixed alphabet, remains unknown. Only very recently, see [8] for more details, it has been possible to give a characterization of the possible values of the total complexity, i.e. the total number of factors of a finite word, without restraints on the cardinality of the alphabet.

In [6] first, and in [5] in full generality, appears a description of some constraints on the possible variations of the complexity function $p_\omega(n)$, that we now recall.

For any word ω of length N on an alphabet $\mathcal{A}$ of cardinality λ, the number $p_\omega(n)$ is clearly bounded from above by $\min(\lambda^n, N-n+1)$; the first item refers to the size of the alphabet and the second to the number of possible different positions of a window of length n in a word of length N.

The following result gives more information on the complexity function.

Theorem 1. *([5]) Let ω be a word of length N in an alphabet $\mathcal{A}$ of cardinal λ. There exist three natural integers $n_0(\omega)$, $n_1(\omega)$ and $n_2(\omega)$ such that $0 \leq n_0(\omega) \leq n_1(\omega) \leq n_2(\omega)$, for which*

- $p_\omega(n) = \lambda^n$ *on* $[0, n_0(\omega)]$,
- $p_\omega(n)$ *is increasing on* $[n_0(\omega), n_1(\omega)]$ *and* $p_\omega(n) < \lambda^n$ *on* $]n_0(\omega), n_1(\omega)]$,
- $p_\omega(n)$ *is constant on* $[n_1(\omega), n_2(\omega)]$,
- $p_\omega(n) = N - n + 1$ *on* $[n_2(\omega), N]$.

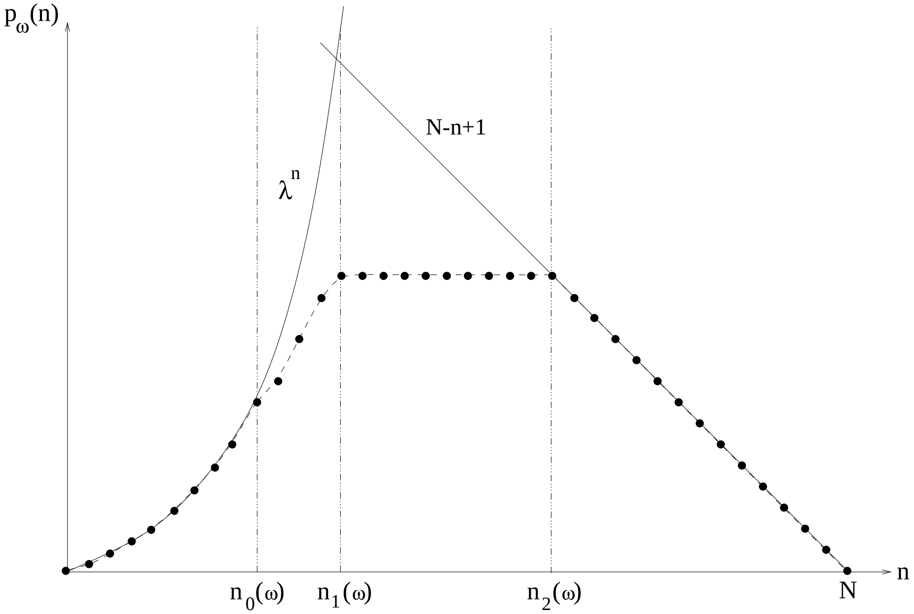

Figure 6: The variations of $(p_\omega(n))_n$

4.2 Maximal value of the complexity function

By repeating the same letter N times, we build a word of minimal complexity, namely $p_\omega(n) = 1$, for $0 < n \leq N$. The natural question concerning the existence of a word of maximal complexity then arises. We show in the following that a word whose complexity is equal to the natural upper bound always exists. We shall came back to this fact in [7], where some of these words will be shown, in a precise sense, to have the maximal randomness: this is somehow surprising for finite words on a finite alphabet. The present proof also shows how the de Bruijn graph may be used. (The reader will find in [5] another - closely related - proof of the same result.)

Proposition 4. *Let $\mathcal{A}$ be an alphabet of cardinal λ. For any integer N there exists a word ω of length N such that*

$$p_\omega(n) = \min(\lambda^n, N - n + 1) \text{ , for } 0 \leq n \leq N \text{ .}$$

Proof. Let k be the largest integer such that $\lambda^k < N - k + 1$ and $m = N - (k + 1) + 1$. Following the notation in Proposition 1, let us consider a Hamiltonian cycle H_k and the graph G'_k obtained from G_k by deleting all the arrows of H_k. Let us denote by $v_1, ..., v_{\lambda^k}$ the arrows of H_k, and by $C_1, ..., C_r$ the connected components of G'_k, two by two disjoint, labelled in increasing order from 1 to r, as they appear while running H_k from v_1 to v_{λ^k}. For each component C_i, $1 \leq i \leq r$, let us denote by u_i the initial extremity of the arrow v_{α_i} where we meet C_i for the first time in H_k (in particular $v_{\alpha_1} = v_1$). Let c_i be the number of arrows of C_i.

We have $\lambda^k + c_1 + \cdots + c_r = \lambda^{k+1}$, and there exists an integer $j \in \{1, ..., r\}$ for which m fulfills the following identity:

$$m = \lambda^k + d_j + c_{j+1} + \cdots + c_r \text{ ,} \quad \text{with} \quad 1 \leq d_j \leq c_j \text{ .}$$

Let u denote a vertex of G_k belonging to the connected component C_j, and linked by an Eulerian path of length d_j to the vertex u_j.

We now consider a cycle in G_k constructed by concatenating the following arrows:

- d_j different successive arrows of C_j, starting at u and ending at u_j,
- the arrows of H_k from v_{α_j} to $v_{\alpha_{j+1}-1}$, starting at u_j and ending at u_{j+1},
- the arrows of an Eulerian cycle of C_{j+1}, starting at u_{j+1} and ending at u_{j+1},

- the arrows of H_k from $v_{\alpha_{j+1}}$ to $v_{\alpha_{j+2}-1}$, starting at u_{j+1} and ending at u_{j+2},
- *etc...*,
- the arrows of an Eulerian cycle of C_r, starting at u_r and ending at u_r,
- the arrows of H_k from v_{α_r} to v_{λ^k}, starting at u_r and ending at u_1,
- finally the arrows of H_k from v_1 to v_{α_j-1}, starting at u_1 and ending at u_j.

The word associated to the cycle defined above has length $M = N - k$ and exactly λ^k factors of length k and m factors of length $k + 1$. Therefore its complexity function has the desired property.

Note that, in the particular case where $j = r$ (which will be the case when $\lambda \geq 3$, as needed in Proposition 3) the construction is somehow simpler. It is sufficient to build the desired cycle by concatenating

- d_j different successive arrows of C_j, starting at u and ending at u_j,
- finally the arrows of H_k, starting at u_r and ending at u_r. □

4.3 A familly of inequalities

We shall state now a set of inequalities fulfiled by the complexity function $p_\omega(n)$, which show the "smoothness" of such a function. This will make more precise the results stated in Theorem 1. In doing so, we shall focus on the suffix tree representation of the word ω.

Theorem 2.

1. *Let $\mathcal{A}$ be an alphabet of cardinal λ and ω a word of lenght N on $\mathcal{A}$. Let d_n^k (resp. g_n^k) be the maximal number of right (resp. left) extensions of an element of $\mathcal{F}act_n(\omega)$ as elements of $\mathcal{F}act_{n+k}(\omega)$. Then*

$$p_\omega(n+k+1) - p_\omega(n+k) \leq \min\{g_n^k, d_n^k\}\ (p_\omega(n+1) - p_\omega(n))\ ,$$

   if $n + k + 1 \leq N + 1$ and $p_\omega(n+1) - p_\omega(n) + 1 > 0$.

2. *Any complexity function $p(n)$ of a word of length N on $\mathcal{A}$ satisfies the following inequalities:*

$$p(n+k+1) - p(n+k) \leq \min\{p(k), N - n + 2 - p(n)\} \\ \times(p(n+1) - p(n)),$$

   if $n + k + 1 \leq N + 1$ and $p(n+1) - p(n) + 1 > 0$.

Remarks.

1. The inequalities above contain the fact that

$$p_\omega(n+1) - p_\omega(n) = 0$$
$$\text{entails} \quad p_\omega(n+k+1) - p_\omega(n+k) = 0 \text{ or } -1 \,,$$

as stated in Theorem 1.

2. The following equality holds for the sequence $u(n) = \lambda^n$ itself:

$$u(n+k+1) - u(n+k) = u(k)(u(n+1) - u(n)) \,.$$

3. Note that the same inequalities as in Theorem 2 may be stated in the following somehow different form:

$$p_\omega(m+1) - p_\omega(m) \le \min_{n \le m} [\min(g_n^{m-n}, d_n^{m-n}) \, (p_\omega(n+1) - p_\omega(n))]$$
$$\le \min_{n \le m} [\min(p_\omega(m-n), N - n + 2 - p_\omega(n)) \, (p_\omega(n+1) - p_\omega(n))] \,.$$

The remainder of section 4 is devoted to the complete proof of Theorem 2.

The reader shall notice that, in the 6 following lemmas, we consider a word $\omega = u\$$ of length N, where $\$ \notin \mathcal{A}$, the set $\mathcal{A}$ is replaced by $\mathcal{A} \cup \{\$\}$ and we consider the suffix tree of u.

In this tree, the deep of a vertex is defined to be the length of the path which connects the vertex to the root of the tree. Let us denote by $\mathcal{E}(i, j)$ the sequence, with repetitions, of the labels of the paths joining the vertex of deep i to those of deep j $(i \le j)$, in the lexicographic order; let $E(i, j)$ be the set of such labels.

For instance, there is no repetition in $\mathcal{E}(0, n)$, and $E(0, n)$ is just the set $\mathcal{F}act_n(\omega)$, containing $p_\omega(n)$ elements $(n \le N)$. The sequence $\mathcal{E}(1, n+1)$ is a sequence of $p_\omega(n+1)$ different words of length n, each element of $E(1, n+1)$ appearing as many times as it has left extensions by a letter $(n + 1 \le N)$.

Lemma 1. *We have*

$$\sum_{\substack{v \in \mathcal{F}act_n(\omega) \\ v \notin Suff(\omega)}} (d(v) - 1) = p_\omega(n+1) - p_\omega(n) + 1 \,,$$

for $n + 1 \le N$.

Proof. This lemma follows immediately from the equality:

$$\begin{aligned} p_\omega(n+1-p_\omega(n)) &= \sum_{v\in\mathcal{F}act_n(\omega)} (d(v)-1) \\ &= \sum_{\substack{v\in\mathcal{F}act_n(\omega)\\ v\notin Suff(\omega)}} (d(v)-1)-1\ . \end{aligned} \qquad \square$$

From Lemma 1 we get a first upper bound of $p_\omega(n+2)-p_\omega(n+1)$:

Lemma 2. *Let* $g_n = \max\{g(v)\ ;\ v\in\mathcal{F}act_n(\omega)\}$. *Then*

$$p_\omega(n+2)-p_\omega(n+1) \le g_n(p_\omega(n+1)-p_\omega(n)+1)-1\ , \quad \text{for } n+1\le N\ .$$

Proof. Each element v of $E(1,n+1)$, except the suffix of length n of ω, appears no more than g_n times in $\mathcal{E}(1,n+1)$; moreover, if $\alpha v\in\mathcal{F}act_{n+1}(\omega)$ then $d(\alpha v)\le d(v)$.

On the other hand, the suffix $R_n(\omega)$ has a unique left extension, $R_{n+1}(\omega)$, the last having no right extension in $\mathcal{F}act(u)$. By using Lemma 1, we end up the proof by setting:

$$\begin{aligned} p_\omega(n+2)-p_\omega(n+1) &\le \sum_{\substack{v\in\mathcal{F}act_n(\omega)\\ v\notin Suff(\omega)}} g(v)(d(v)-1)-1 \\ &\le g_n(p_\omega(n+1)-p_\omega(n)+1)-1\ . \end{aligned} \qquad \square$$

We shall now sharpen the previous inequalities, using an argument which is implicit in [5] and which may be useful in its own right.

Lemma 3. *Suppose that v_n is a right special factor with length $n\ge 1$ of ω, appearing in the last (right) position among all the right special factors of ω with length n. If in this position the word v_n is followed by a letter c, then there is no other occurrence of v_nc in ω.*

Remarks.

1. Any non empty suffix S of ω ends with \$ and satisfies $d(S)=0$; therefore the word v_n, if it exists, cannot be suffix, which shows then the existence of c.

2. The assertion of the lemma is false without the use of \$, as we can see in the following counter-example: $n=2$ and $\omega = bacabac$.

Proof. Let $\omega = \omega_0 \dots \omega_{N-1}$. Let i be the starting position of the last occurrence of $v_n c$. Let j, $j \neq i$, be another starting position of $v_n c$. Then we can write $\omega_i \dots \omega_{N-1} = v_n c \alpha_1 \dots \alpha_r$ and $\omega_j \dots \omega_{N-1} = v_n c \beta_1 \dots \beta_s$, with $r \geq 1$ and $s > r \geq 1$. Since $\alpha_r = \$$ and $r < s$, there is an integer $k \geq 1$ such that $v_n c \alpha_1 \dots \alpha_k$ and $v_n c \beta_1 \dots \beta_k$ are identical, but for the last letter. Setting $v_n c \alpha_1 \dots \alpha_k = v \alpha_k$, and $v_n c \beta_1 \dots \beta_k = v \beta_k$, we see that the suffix v is right special and has length exactly n, which is in contradiction with the assumption on v_n. □

From Lemma 3 immediately follows the next:

Lemma 4. *With the same notation as before, we have* $g(v_n c) = 1$.

We use this lemma to prove the following:

Lemma 5. *With the same notation as before, we have*

$$p_\omega(n+2) - p_\omega(n+1) \leq g_n(p_\omega(n+1) - p_\omega(n)) \ ,$$

if $p_\omega(n+1) - p_\omega(n) + 1 > 0$.

Proof. Under the above assumption on n, there exists a right special factor of length n in ω.

According to Lemma 4, it is the possible to refine the inequality of Lemma 2: if $\alpha v_n c$ and βv_n are two factors of ω, then $d(\beta v_n) \leq d(v_n) - 1$.

Therefore

$$\begin{aligned} p_\omega(n+2) - p_\omega(n+1) &\leq \sum_{\substack{v \in \mathcal{F}act_n(\omega) \\ v \notin Suff(\omega) \\ v \neq v_n}} g(v)(d(v) - 1) \\ &\quad + (g(v_n) - 1)(d(v_n) - 2) + (d(v_n) - 1) - 1 \\ &\leq \sum_{\substack{v \in \mathcal{F}act_n(\omega) \\ v \notin Suff(\omega) \\ v \neq v_n}} g_n(d(v) - 1) + (g_n - 1)(d(v_n) - 2) + (d(v_n) - 1) - 1 \\ &= g_n \sum_{\substack{v \in \mathcal{F}act_n(\omega) \\ v \notin Suff(\omega)}} (d(v) - 1) - g_n = g_n(p_\omega(n+1) - p_\omega(n)) \ . \end{aligned}$$ □

Lemma 6. *With the same notation as before, we have*

$$p_\omega(n+k+1) - p_\omega(n+k) \leq g_n^k(p_\omega(n+1) - p_\omega(n)) \ ,$$

if $n + k + 1 \leq N$ *and* $p_\omega(n+1) - p_\omega(n) + 1 > 0$.

Proof. It remains to extend Lemma 5, by considering now the sequence $\mathcal{E}(k, n+k)$ inside $ST(u)$.

In this sequence, every word is repeated as many times as the number of its left extensions by words of k letters. This is why we defined g_n^k as being the maximal number of left extensions by words of k letters of an element of $\mathcal{F}act_n(\omega)$ in elements of $\mathcal{F}act_{n+k}(\omega)$.

Let us first consider the case where v_n does not appear in the sequence $\mathcal{E}(k, n+k)$; in this case, since v_n is the last among all the right special factors of length n in ω, there is no right special factor in $\mathcal{E}(k, n+k)$, and then

$$p_\omega(n+k+1) - p_\omega(n+k) = -1 \quad \text{and} \quad g_n^k(p_\omega(n+1) - p_\omega(n)) \geq 0 \, ,$$

and the lemma follows. On the contrary, if v_n appears in the sequence $\mathcal{E}(k, n+k)$, we only need to repeat the proof of Lemma 5, replacing g_n by g_n^k. □

Proof of Theorem 2.

1. Let us now come back to the case of a word ω of length N on $\mathcal{A}$, without adding the extra symbol \$.

   From one side

   $$p_{\omega\$}(n) = p_\omega(n) + 1, \quad \text{if } 1 \leq n \leq |\omega| + 1 \, ,$$

   and therefore, for $n+k+1 \leq N+1$, we get

   $$p_\omega(n+k+1) - p_\omega(n+k) = p_{\omega\$}(n+k+1) - p_{\omega\$}(n+k) \, .$$

   From the other side, the maximal number of left extensions of factors of length n as factors of length $n+k$ is the same for ω and ω\$, since $L_n(\omega\$)$ does not occur in ω. We thus get, for n and k such that $n+k+1 \leq N+1$ and $p_\omega(n+1) - p_\omega(n) + 1 > 0$,

   $$p_\omega(n+k+1) - p_\omega(n+k) \leq g_n^k(p_\omega(n+1) - p_\omega(n)) \, .$$

   From there we deduce the desired inequalities, using the symmetry of role of left and right extensions of the factors of ω.

2. It remains to get an upper bound for g_n^k. It is clear that $g_n^k < p(k)$. Moreover g_n^k is certainly smaller than the maximal number of occurrences of a factor v of length n in ω. Since there are $p_\omega(n)$ such factors, we get

   $$occ_\omega(v) \leq (N - n + 1) - (p_\omega(n) - 1) \, .$$ □

4.4 Localization of $l(\omega)$ and of $r(\omega)$

It is now possible to use Theorem 1 to localize $l(\omega)$ and $r(\omega)$, with respect to $n_0(\omega)$, $n_1(\omega)$ and $n_2(\omega)$.

Proposition 5. *Both integers $l(\omega)$ and $r(\omega)$ belong to the set $[n_0(\omega), n_1(\omega)] \cup \{n_2(\omega)\}$.*

Proof. If $n < n_0(\omega)$, then the word $L_n(\omega)$ has λ left extensions, and therefore $l(\omega) > n$; in the same way, $r(\omega) > n$. For $n \geq n_2(\omega)$, the word $L_n(\omega)$ has no left extensions and then $l(\omega) \leq n_2(\omega)$, as well as $r(\omega) \leq n_2(\omega)$.

Finally, if the interval $[n_1(\omega), n_2(\omega)[$ is not empty, there are two possible cases:

- either all the factors of length n of ω have a unique left extension, for all $n \in [n_1(\omega), n_2(\omega)[$,
- or, for any $n \in [n_1(\omega), n_2(\omega)[$, the word $L_n(\omega)$ has no left extension, one factor of length n has two left extensions and all the others have only one extension.

Therefore, we can never have $l(\omega) \in]n_1(\omega), n_2(\omega)[$, and similarly for $r(\omega)$. □

5 The complexity function of a window

We shall now use the complexity function as a tool for the analysis of symbolic sequences.

Let a finite alphabet $\mathcal{A}$ and a (large) sequence S that we want to analyze be given. In this case we can think about a word $\omega \in \mathcal{A}^N$, $N < |S|$, as the factor of the sequence S appearing in a window of length N, set down at some position of S. It is then natural to consider observables Φ, being functions of $\mathcal{A}^n$ to some set of numbers, functions, histograms, *etc...*, that give some piece of information about the sequence.

In such a situation, it may be interesting to describe the fluctuations of $\Phi(\omega)$ when ω slides along S ($\omega = S[i, i+n-1]$ and i varies), or when the size of the window increases around a given position in S ($\omega = S[i-n, i+n]$ and n varies).

In the present section we shall analyze this situation when $\Phi(\omega)$ stands for the complexity function of ω, a case met, for instance, when analyzing the variety of patterns present in a sequence S. It is then important to have a control of the possible fluctuations of $\Phi(\omega)$ along S.

5.1 The complexity of a sliding window

Let S be a sequence on an alphabet $\mathcal{A}$, and N an integer which is the length of the sliding window.

Given a factor ω of S of length N, let us write $\omega = S[i, i + N - 1]$ with $i + N < |S|$; we now intend to compare the complexity functions of ω and of $\sigma\omega = S[i + 1, i + N]$.

In the sequel, ω' denotes the word $S[i, i + N]$.

Proposition 6. *With the same notation as before, we have*

1. *if* $n < \min(l(\omega'), r(\omega'))$, *then*
$$p_\omega(n) = p_{\omega'}(n) = p_{\sigma\omega}(n) \ ,$$
2. - *if* $l(\omega') < r(\omega')$ *and* $l(\omega') \leq n < r(\omega')$, *then*
$$p_\omega(n) = p_{\omega'}(n) = p_{\sigma\omega}(n) + 1 \ ,$$
   - *if* $r(\omega') < l(\omega')$ *and* $r(\omega') \leq n < l(\omega')$, *then*
$$p_\omega(n) + 1 = p_{\omega'}(n) = p_{\sigma\omega}(n) \ ,$$
3. *if* $n \geq \max\{l(\omega'), r(\omega')\}$, *then*
$$p_\omega(n) + 1 = p_{\omega'}(n) = p_{\sigma\omega}(n) + 1 \ .$$

Proof. It is clear that $p_{\omega'}(n) \in \{p_\omega(n), p_\omega(n) + 1\}$: the only factor of ω' that can be in $\mathcal{F}act_n(\omega') \setminus \mathcal{F}act_n(\omega)$ is $R_n(\omega')$. But this only happens when $R_n(\omega')$ occurs once in ω', that is to say when $n > r(\omega')$.

To complete the proof, it remains to use the symmetry between ω and σw, by reversing S. □

Let us now denote $\sigma^i = S[i, i + N - 1]$, $l_i = l(S[i, i + N])$ and $r_i = r(S[i, i + N])$, for $0 \leq i \leq |S| - N - 1$.

A consequence of Proposition 6 is that knowing the complexity function of the first window $\sigma^0 = S[0, N - 1]$ and the sequences $(l_i)_{0 \leq i \leq |S| - N - 1}$ and $(r_i)_{0 \leq i \leq |S| - N - 1}$, we are able to reconstruct the complexity functions of all the successive windows of length N in S (see figure 7):

- if $n < \min\{l_i, r_i\}$ or if $n \geq \max\{l_i, r_i\}$, then $p_{\sigma^{i+1}}(\omega) = p_{\sigma^i}(\omega)$,

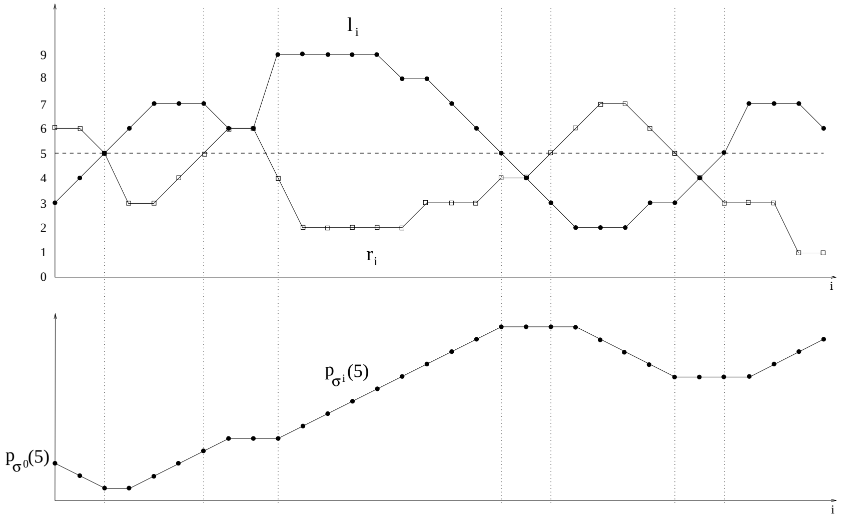

Figure 7: The sequences (l_i) and (r_i) and the reconstruction of p_{σ^i}.

- if $l_i \leq n < r_i$, then $p_{\sigma^{i+1}}(\omega) = p_{\sigma^i}(\omega) - 1$,
- if $r_i \leq n < l_i$, then $p_{\sigma^{i+1}}(\omega) = p_{\sigma^i}(\omega) + 1$.

Note that, in Proposition 6, only appear the lenghts $l(\omega')$ and $r(\omega')$, and not directly $l(\omega)$, $r(\omega)$, $l(\sigma\omega)$ or $r(\sigma\omega)$. The following proposition allows us to make connections between these quantities. Furthermore, it enables us to make more precise the variations of l_i and r_i.

Proposition 7.

1. *With the same notation as before, we have*
   - *either $B(\omega') \neq \emptyset$, and then*

     $$l(\omega) + 1 = l(\omega') = r(\omega') = r(\sigma\omega) + 1 \ ,$$

   - *or $B(\omega') = \emptyset$ and then*

     $$l(\omega) = l(\omega') \qquad \textit{and} \qquad r(\omega') = r(\sigma\omega) \ .$$

2. *In each case, we have*

$$\begin{aligned} & l(\omega) \leq l(\sigma\omega) + 1 \\ \textit{and} \qquad & r(\sigma\omega) \leq r(\omega) + 1 \ . \end{aligned}$$

Proof.

1. If $L(\omega)$ is a suffix of ω', we have $occ_{\omega'}(L(\omega)) = 2$ and $occ_{\sigma\omega}(L(\omega)) = 1$, and then $B(\omega') = \{L(\omega)\}$ and $R(\sigma\omega) = L(\omega)$; but by proposition 3 we get
$$l(\omega') = r(\omega') = b(\omega') + 1 = l(\omega) + 1 = r(\sigma\omega) + 1 .$$
If, on the contrary, $L(\omega)$ is not suffix of ω', then $occ_{\omega'}(L(\omega)) = 1$ and $L(\omega) = L(\omega')$; thus $l(\omega) = l(\omega')$, and also $r(\omega') = r(\sigma\omega)$ by reversing ω'.

2. Let α be the last letter of ω'. We do not have $occ_{\omega}(R(\omega)\alpha) > 1$, for otherwise $occ_{\omega}(R(\omega)) > 1$. Thus $r(\sigma\omega) \leq r(\omega) + 1$, and also $l(\omega) \leq l(\sigma\omega) + 1$ by reversing ω'. □

The following result is then immediate (see figure 7):

Corollary 2.

1. *If $r_{i+1} > r_i$, then $r_{i+1} = r_i + 1$.*
2. *If $l_{i+1} < l_i$, then $l_{i+1} = l_i - 1$.* □

5.2 The complexity of a window of increasing size

We want now to compare the complexity functions of the factors $w = S[i, i + N - 1]$ and $W = S[i - 1, i + N]$ of the sequence S, with $1 \leq i \leq |S| - N - 1$.

For this purpose, we use the set $E(W) = \{n < |W|\ ;\ L_n(W) = R_n(W)$ and $occ_W(L_n(W)) = 2\}$ introduced in section 2.

Let us recall that, following Proposition 3, the set $E(W)$ is either empty or reduced to a single element $\{b(W)\}$. We then get the following assertion:

Proposition 8. *With the same notation as before, we have*

- *either $E(W) \neq \emptyset$, and then*
$$p_W(n) = \begin{cases} p_w(n) \text{ if } n < b(W) \\ p_w(n) + 1 \text{ if } n = b(W) \\ p_w(n) + 2 \text{ if } n > b(W) \end{cases} ,$$
- *or $E(W) = \emptyset$, and then*
$$p_W(n) = \begin{cases} p_w(n) \text{ if } n < \min(l(W), r(W)) \\ p_w(n) + 1 \text{ if } \min(l(W), r(W)) \leq n < \max(l(W), r(W)) \\ p_w(n) + 2 \text{ if } n \geq \max(l(W), r(W)) \end{cases} .$$

Proof. It is sufficient to remark that $p_W(n) - p_w(n) \in \{0, 1, 2\}$, since only $L_n(\omega)$ and $R_n(\omega)$ may be elements of $\mathcal{F}act_n(W) \smallsetminus \mathcal{F}act_n(w)$. □

References

[1] C. Berge. *Graphes et hypergraphes*, Dunod, (1973).

[2] W.T. Tutte. *Graph Theory*, Encyclopedia of mathematics and its applications, volume **21:** (1984).

[3] N.G. de Bruijn. A combinatorial problem, *Nederl. Akad. Wetensch. Proc.*, **49:** (1946), 758–764, et *Indag. Math.*, **8:** (1946), 461–467.

[4] M. Crochemore and W. Rytter. *Text Algorithms*, Oxford University Press, (1994).

[5] J. Cassaigne and M-C. Anisiu. Properties of the complexity function for finite words, preprint, (2001).

[6] A. de Luca. On the combinatorics of finite words, *Theor. Comput. Sci.*, **218:** (1999), 13–39.

[7] S. Jaeger, R. Lima and B. Mossé. Symbolic analysis of finite words, II, The entropy fonction, in preparation.

[8] F. Levé, P. Séébold. Proof of a conjecture on word complexity, *Bull. Belg. Math. Soc.*, **8:** (2001), 275–289.

[9] M. Lothaire. *Combinatorics on Words*, Cambridge University Press, (1997).

[10] D. Gusfield. *Algorithms on strings, trees, and sequences*, Cambridge University Press, (1997).

[11] J. Cassaigne. Complexité et facteurs spéciaux, *Bull. Belg. Math. Soc. Simon Stevin*, **4**(1): (1997), 67–88.

[12] P. Arnoux and G. Rauzy. Représentation géométrique des suites de complexité $2n + 1$, *Bull. Soc. Math. France*, **119**(2): (1991), 199–215.

Sébastien Jaeger and Ricardo Lima
Centre de Physique Théorique (FRUMAM)
CNRS Luminy Case 907
13288 Marseille CEDEX 09
FRANCE
jaeger@cpt.univ-mrs.fr and lima@cpt.univ-mrs.fr

Brigitte Mossé
Institut de Mathématiques de Luminy (FRUMAM)
CNRS Luminy Case 907
13288 Marseille CEDEX 09
FRANCE
mosse@iml.univ-mrs.fr