# 26. Complexity of Words

The complexity of words is a continuously growing field of the combinatorics of words. Hundreds of papers are devoted to different kind of complexities. We try to present in this chapter far from being exhaustive the basic notions and results for finite and infinite words.

First of all we summarize the simple (classical) complexity measures, giving formulas and algorithms for several cases. After this, generalized complexities are treated, with different type of algorithms. We finish this chapter by presenting the palindrome complexity.

Finally, references from a rich bibliography are given.

## 26.1. Simple complexity measures

In this section simple (classical) complexities, as measures of the diversity of the subwords in finite and infinite words, are discussed. First, we present some useful notions related to the finite and infinite words with examples. Word graphs, which play an important role in understanding and obtaining the complexity, are presented in detail with pertinent examples. After this, the subword complexity (as number of subwords), with related notions, is expansively presented.

### 26.1.1. Finite words

Let $A$ be a finite, nonempty set, called ***alphabet.*** Its elements are called ***letters*** or ***symbols.*** A string $a_1a_2\dots a_n$, formed by (not necessary different) elements of $A$, is a ***word.*** The length of the word $u = a_1a_2\dots a_n$ is $n$, and is denoted by $|u|$. The word without any element is the empty word, denoted by $\varepsilon$ (sometimes $\lambda$). The set of all finite words over $A$ is denoted by $A^*$. We will use the following notations too:

$$A^+ = A^* \setminus \{\varepsilon\}, \qquad A^n = \Big\{u \in A^* \mid |u| = n\Big\} = \Big\{a_1a_2\dots a_n \mid a_i \in A\Big\},$$

that is $A^+$ is the set of all finite and nonempty words over $A$, whilst $A^n$ is the set of all words of length $n$ over $A$. Obviously $A^0 = \{\varepsilon\}$. The sets $A^*$ and $A^+$ are infinite denumerable sets.

We define in $A^*$ the binary operation called ***concatenation*** (shortly ***catenation***). If $u = a_1a_2\ldots a_n$ and $v = b_1b_2\ldots b_m$, then

$$w = uv = a_1a_2\ldots a_nb_1b_2\ldots b_m, \qquad |w| = |u| + |v|.$$

This binary operation is associative, but not commutative. Its neutral element is $\varepsilon$ because $\varepsilon u = u\varepsilon = u$. The set $A^*$ with this neutral element is a monoid. We introduce recursively the ***power of a word:***

- $u^0 = \varepsilon$
- $u^n = u^{n-1}u$, if $n \geq 1$.

A word is ***primitive*** if it is no power of any word, so $u$ is primitive if

$$u = v^n,\ v \neq \varepsilon \quad \Rightarrow \quad n = 1.$$

For example, $u = abcab$ is a primitive word, whilst $v = abcabc = (abc)^2$ is not.

The word $u = a_1a_2\ldots a_n$ is ***periodic*** if there is a value $p$, $1 \leq p < n$ such that

$$a_i = a_{i+p}, \text{ for all } i = 1, 2, \ldots, n - p,$$

and $p$ is the period of $u$. The least such $p$ is the ***least period*** of $u$.

The word $u = abcabca$ is periodic with the least period $p = 3$.

Let us denote by $(a, b)$ the greatest common divisor of the naturals $a$ and $b$. The following result is obvious.

**Theorem 26.1** *If $u$ is periodic, and $p$ and $q$ are periods, then $(p, q)$ is a period too, provided $p + q < |u|'$.*

The ***reversal*** (or ***mirror image***) of the word $u = a_1a_2\ldots a_n$ is $u^R = a_na_{n-1}\ldots a_1$. Obviously $\left(u^R\right)^R = u$. If $u = u^R$, then $u$ is a ***palindrome.***

The word $u$ is a ***subword*** (or ***factor***) of $v$ if there exist the words $p$ and $q$ such that $v = puq$. If $pq \neq \varepsilon$, then $u$ is a proper subword of $v$. If $p = \varepsilon$, then $u$ is a ***prefix*** of $v$, and if $q = \varepsilon$, then $u$ is a ***suffix*** of $v$. The set of all subwords of length $n$ of $u$ is denoted by $F_n(u)$. $F(u)$ is the set of nonempty subwords of $u$, so

$$F(u) = \bigcup_{n=1}^{|u|} F_n(u).$$

For example, if $u = abaab$, then

$F_1(u) = \{a, b\}$, $F_2(u) = \{ab, ba, aa\}$, $F_3(u) = \{aba, baa, aab\}$,
$F_4(u) = \{abaa, baab\}$, $F_5(u) = \{abaab\}$.

The words $u = a_1a_2\ldots a_m$ and $v = b_1b_2\ldots b_n$ are equal, if

- $m = n$ and
- $a_i = b_i$, for $i = 1, 2, \ldots, n$.

**Theorem 26.2** (Fine–Wilf). *If $u$ and $v$ are words of length $n$, respective $m$, and if there are the natural numbers $p$ and $q$, such that $u^p$ and $v^q$ have a common prefix of length $n + m - (n, m)$, then $u$ and $v$ are powers of the same word.*

The value $n+m-(n,m)$ in the theorem is tight. This can be illustrated by the following example. Here the words $u$ and $v$ have a common prefix of length $n+m-(n,m)-1$, but $u$ and $v$ are not powers of the same word.

$$\begin{array}{lll} u = abaab, & m = |u| = 5, & u^2 = abaababaab\,, \\ v = aba, & n = |v| = 3, & v^3 = abaabaaba\,. \end{array}$$

By the theorem a common prefix of length 7 would ensure that $u$ and $v$ are powers of the same word. We can see that $u^2$ and $v^3$ have a common prefix of length 6 ($abaaba$), but $u$ and $v$ are not powers of the same word, so the length of the common prefix given by the theorem is tight.

## 26.1.2. Infinite words

Beside the finite words we consider ***infinite*** (more precisely infinite at right) words too:

$$u = u_1 u_2 \dots u_n \dots, \quad \text{where } u_1, u_2, \dots \in A\,.$$

The set of infinite words over the alphabet $A$ is denoted by $A^\omega$. If we will study together finite and infinite words the following notation will be useful:

$$A^\infty = A^* \cup A^\omega\,.$$

The notions as subwords, prefixes, suffixes can be defined similarly for infinite words too.

The word $v \in A^+$ is a subword of $u \in A^\omega$ if there are the words $p \in A^*$, $q \in A^\omega$, such that $u = pvq$. If $p \neq \varepsilon$, then $p$ is a prefix of $u$, whilst $q$ is a suffix of $u$. Here $F_n(u)$ also represents the set of all subwords of length $n$ of $u$.

Examples of infinite words over a binary alphabet:

1) The ***power word*** is defined as:

$$p = 010011000111\dots0^n1^n\dots = 010^21^20^31^3\dots0^n1^n\dots\,.$$

It can be seen that

$F_1(p) = \{0,1\}$, $F_2(p) = \{01,10,00,11\}$,
$F_3(p) = \{010,100,001,011,110,000,111\}, \dots$

2) The ***Champernowne word*** is obtained by writing in binary representation the natural numbers $0,1,2,3,\dots$:

$$c = 0\,1\,10\,11\,100\,101\,110\,111\,1000\,1001\,1010\,1011\,1100\,1101\,1110\,1111\,10000\dots\,.$$

It can be seen that

$F_1(p) = \{0,1\}$, $F_2(p) = \{00,01,10,11\}$,
$F_3(p) = \{000,001,010,011,100,101,110,111\}, \dots$

3) The finite ***Fibonacci words*** can be defined recursively as:

$f_0 = 0$, $f_1 = 01$
$f_n = f_{n-1}f_{n-2}$, if $n \geq 2$.

From this definition we obtain:
$f_0 = 0$,
$f_1 = 01$,
$f_2 = 010$,
$f_3 = 01001$,
$f_4 = 01001010$,
$f_5 = 0100101001001$,
$f_6 = 010010100100101001010$.

The infinite Fibonacci word can be defined as the limit of the sequence of finite Fibonacci words:

$$f = \lim_{n\to\infty} f_n \,.$$

The subwords of this word are:
$F_1(f) = \{0, 1\},\ F_2(f) = \{01, 10, 00\},\ F_3(f) = \{010, 100, 001, 101\}$,
$F_4(f) = \{0100, 1001, 0010, 0101, 1010\}, \ldots$ .

The name of Fibonacci words stems from the Fibonacci numbers, because the length of finite Fibonacci words is related to the Fibonacci numbers: $|f_n| = F_{n+2}$, i.e. the length of the $n$th finite Fibonacci word $f_n$ is equal to the $(n+2)$th Fibonacci number.

The infinite Fibonacci word has a lot of interesting properties. For example, from the definition, we can see that it cannot contain the subword 11.

The number of 1's in a word $u$ will be denoted by $h(u)$. An infinite word $u$ is ***balanced,*** if for arbitrary subwords $x$ and $y$ of the same length, we have $|h(x) - h(y)| \le 1$, i.e.

$$x, y \in F_n(u) \ \Rightarrow\ |h(x) - h(y)| \le 1 \,.$$

**Theorem 26.3** *The infinite Fibonacci word $f$ is balanced.*

**Theorem 26.4** *$F_n(f)$ has $n + 1$ elements.*

If word $u$ is concatenated by itself infinitely, then the result is denoted by $u^\omega$.

The infinite word $u$ is ***periodic,*** if there is a finite word $v$, such that $u = v^\omega$. This is a generalization of the finite case periodicity. The infinite word $u$ is ***ultimately periodic,*** if there are the words $v$ and $w$, such that $u = vw^\omega$.

The infinite Fibonacci word can be generated by a (homo)morphism too. Let us define this morphism:

$$\chi : A^* \to A^*, \quad \chi(uv) = \chi(u)\chi(v), \quad \forall u, v \in A^* \,.$$

Based on this definition, the function $\chi$ can be defined on letters only. A morphism can be extended for infinite words too:

$$\chi : A^\omega \to A^\omega, \quad \chi(uv) = \chi(u)\chi(v), \quad \forall u \in A^*, v \in A^\omega \,.$$

The finite Fibonacci word $f_n$ can be generated by the following morphism:

$$\sigma(0) = 01,\ \sigma(1) = 0 \,.$$

In this case we have the following theorem.

**Theorem 26.5** $f_{n+1} = \sigma(f_n)$.

**Proof** The proof is by induction. Obviously $f_1 = \sigma(f_0)$. Let us presume that $f_k = \sigma(f_{k-1})$ for all $k \leq n$. Because

$$f_{n+1} = f_n f_{n-1},$$

by the induction hypothesis

$$f_{n+1} = \sigma(f_{n-1})\sigma(f_{n-2}) = \sigma(f_{n-1}f_{n-2}) = \sigma(f_n).$$

■

From this we obtain:

**Theorem 26.6** $f_n = \sigma^n(0)$.

The infinite Fibonacci word $f$ is the fixed point of the morphism $\sigma$.

$$f = \sigma(f).$$

### 26.1.3. Word graphs

Let $V \subseteq A^m$ be a set of words of length $m$ over $A$, and $E \subseteq AV \cap VA$. We define a digraph, whose vertices are from $V$, and whose arcs from $E$. There is an arc from the vertex $a_1a_2\ldots a_m$ to the vertex $b_1b_2\ldots b_m$ if

$$a_2 = b_1, \quad a_3 = b_2, \quad \ldots, \quad a_m = b_{m-1} \text{ and } a_1a_2\ldots a_mb_m \in E,$$

that is the last $m-1$ letters in the first word are identical to the first $m-1$ letters in the second word. This arc is labelled by $a_1a_2\ldots a_mb_m$ (or $a_1b_1\ldots b_m$).

#### De Bruijn graphs

If $V = A^m$ and $E = A^{m+1}$, where $A = \{a_1, a_2, \ldots a_n\}$, then the graph is called ***De Bruijn graph,*** denoted by $B(n, m)$.

Figures 26.1 and 26.2 illustrate De Bruijn graphs $B(2, 3)$ and $B(3, 2)$.

To a walk[1] $x_1x_2\ldots x_m$, $x_2x_3\ldots x_mx_{m+1}$, $\ldots$, $z_1z_2\ldots z_m$ in the De Bruijn graph we attach the label $x_1x_2\ldots z_{m-1}z_m$, which is obtained by maximum overlap of the vertices of the walk. In Figure 26.1 in the graph $B(2, 3)$ the label attached to the walk $001, 011, 111, 110$ (which is a path) is $001110$. The word attached to a Hamiltonian path (which contains all vertices of the graph) in the graph $B(n, m)$ is an $(n, m)$-type ***De Bruijn word.*** For example, words $0001110100$ and $0001011100$ are $(2, 3)$-type De Bruijn word. An $(n, m)$-type De Bruijn word contains all words of length $m$.

A connected digraph[2] is Eulerian[3] if the in-degree of each vertex is equal to its out-degree[4].

---

[1] In a graph a walk is a sequence of neighbouring edges (or arcs with the same orientation). If the edges or arcs of the walk are all different the walk is called trail, and when all vertices are different, the walk is a path.

[2] A digraph (oriented graph) is connected if between every pair of vertices there is an oriented path at least in a direction.

[3] A digraph is Eulerian if it contains a closed oriented trail with all arcs of the graph.

[4] In-degree (out-degree) of a vertex is the number of arcs which enter (leave) this vertex.

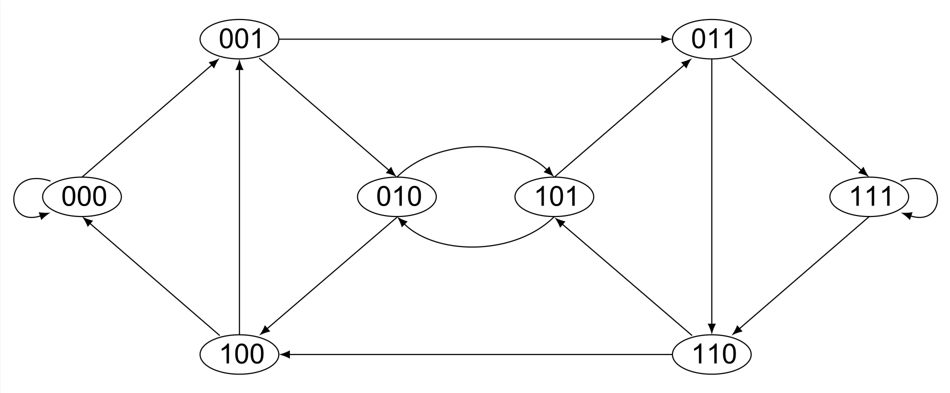


**Figure 26.1** The De Bruijn graph $B(2,3)$.

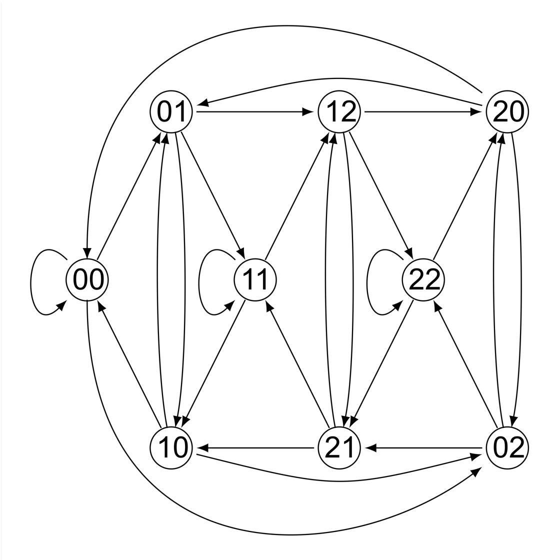


**Figure 26.2** The De Bruijn graph $B(3,2)$.

**Theorem 26.7** *The De Bruijn graph $B(n,m)$ is Eulerian.*

**Proof** a) The graph is connected because between all pair of vertices $x_1x_2\dots x_m$ and $z_1z_2\dots z_m$ there is an oriented path. For vertex $x_1x_2\dots x_m$ there are $n$ leaving arcs, which enter vertices whose first $m-1$ letters are $x_2x_3\dots x_m$, and the last letters in this words are all different. Therefore, there is the path $x_1x_2\dots x_m$, $x_2x_3\dots x_mz_1$, $\dots$, $x_mz_1\dots z_{m-1}$, $z_1z_2\dots z_m$.

b) There are incoming arcs to vertex $x_1x_2\dots x_m$ from vertices $yx_1\dots x_{m-1}$, where $y \in A$ ($A$ is the alphabet of the graph, i.e. $V = A^m$). The arcs leaving vertex $x_1x_2 \ldots x_m$ enter vertices $x_2x_3 \ldots x_my$, where $y \in A$. Therefore, the graph is Eulerian, because the in-degree and out-degree of each vertex are equal. ■

From this the following theorem is simply obtained.

**Theorem 26.8** *An oriented Eulerian trail of the graph $B(n, m)$ (which contains all arcs of graph) is a Hamiltonian path in the graph $B(n, m + 1)$, preserving the order.*

For example, in $B(2, 2)$ the sequence 000, 001, 010, 101, 011, 111, 110, 100 of arcs is an Eulerian trail. At the same time these words are vertices of a Hamiltonian path in $B(2, 3)$.

### Algorithm to generate De Bruijn words

Generating De Bruijn words is a common task with respectable number of algorithms. We present here the well-known Martin algorithm. Let $A = \{a_1, a_2, \ldots, a_n\}$ be an alphabet. Our goal is to generate an $(n, m)$-type De Bruijn word over the alphabet $A$.

We begin the algorithm with the word $a_1^m$, and add at its right end the letter $a_k$ with the greatest possible subscript, such that the suffix of length $m$ of the obtained word does not duplicate a previously occurring subword of length $m$. Repeat this until such a prolongation is impossible.

When we cannot continue, a De Bruijn word is obtained, with the length $n^m + m - 1$. In the following detailed algorithm, $A$ is the $n$-letters alphabet, and $B = (b_1, b_2, \ldots)$ represents the result, an $(n, m)$-type De Bruijn word.

Martin$(A, n, m)$

```
for i ← 1 to m
     do b_i ← a_1
i ← m
repeat
     done ← TRUE
     k ← n
     while k > 1
          do if b_{i-m+2}b_{i-m+3} ... b_i a_k ⊄ b_1b_2 ... b_i        ▷ Not a subword.
               then i ← i + 1
                    b_i ← a_k
                    done ← FALSE
                    exit while
               else k ← k − 1
until done
return B                                        ▷ B = (b_1, b_2, ..., b_{n^m+m+1}).
```

Because this algorithm generates all letters of a De Bruijn word of length $(n^m + m - 1)$, and $n$ and $m$ are independent, its time complexity is $\Omega(n^m)$. The more precise characterization of the running time depends on the implementation of line 8. The **repeat** statement is executed $n^m - 1$ times. The **while** statement is executed at most $n$ times for each step of the **repeat.** The test $b_{i-m+2}b_{i-m+3}\dots b_i a_k \not\subset b_1 b_2 \dots b_i$ can be made in the worst case in $mn^m$ steps. So, the total number of steps is not greater than $mn^{2m+1}$, resulting a worst case bound $\Theta(n^m + 1)$. If we use Knuth-Morris-Pratt string mathching algorithm, then the worst case running time is $\Theta(n^{2m})$.

In chapter **??** a more efficient implementation of the idea of Martin is presented. Based on this algorithm the following theorem can be stated.

**Theorem 26.9** *An $(n, m)$-type De Bruijn word is the shortest possible among all words containing all words of length $m$ over an alphabet with $n$ letters.*

To generate all $(n, m)$-type De Bruijn words the following recursive algorithm is given. Here $A$ is also an alphabet with $n$ letters, and $B$ represents an $(n, m)$-type De Bruijn word. The algorithm is called for each position $i$ with $m+1 \le i \le n^m + m - 1$.

All-De-Bruijn$(B, i, m)$

**for** $j \leftarrow 1$ **to** $n$<br>
    **do** $b_i \leftarrow a_j$<br>
        **if** $b_{i-m+1}b_{i-m+2}\dots b_i \not\subset b_1 b_2 \dots b_{i-1}$    ▷ Not a subword.<br>
          **then** All-De-Bruijn$(b, i+1, m)$<br>
          **else if** $length(B) = n^m + m - 1$<br>
              **then** print $B$    ▷ A De Bruijn word.<br>
                  **exit for**

The call of the procedure:

**for** $i = 1$ **to** $m$<br>
  **do** $b_i \leftarrow a_1$<br>
All-De-Bruijn $(B, m+1, m)$.

This algorithm naturally is exponential.

In following, related to the De Bruijn graphs, the so-called De Bruijn trees will play an important role.

A ***De Bruijn tree*** $T(n, w)$ with the root $w \in A^m$ is a $n$-ary tree defined recursively as follows:

*i.* The word $w$ of length $m$ over the alphabet $A = \{a_1, a_2, \dots, a_n\}$ is the root of $T(n, w)$.
*ii.* If $x_1x_2\dots x_m$ is a leaf in the tree $T(n, w)$, then each word $v$ of the form $x_2x_3\dots x_m a_1$, $x_2x_3\dots x_m a_2$, ..., $x_2x_3\dots x_m a_n$ will be a descendent of $x_1x_2\dots x_m$, if in the path from root to $x_1x_2\dots x_m$ the vertex $v$ does not appears.
*iii.* The rule *ii* is applied as many as it can.

In Figure 26.3 the De Bruijn tree $T(2, 010)$ is given.

### Rauzy graphs

If the word $u$ is infinite, and $V = F_n(u)$, $E = F_{n+1}(u)$, then the corresponding

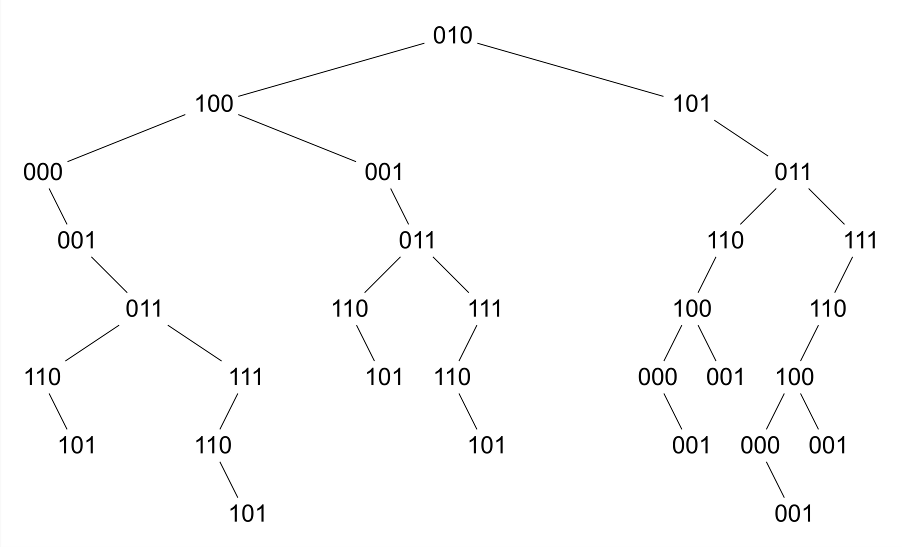


**Figure 26.3** The De Bruijn tree $T(2, 010)$.

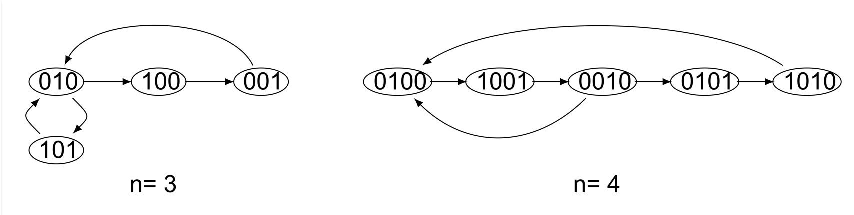


**Figure 26.4** Rauzy graphs for the infinite Fibonacci word.

word graph is called ***Rauzy graph*** (or ***subword graph***). Figure 26.4 illustrates the Rauzy graphs of the infinite Fibonacci word for $n = 3$ and $n = 4$. As we have seen, the infinite Fibonacci word is

$$f = 0100101001001010010100100101001001\ldots,$$

and $F_1(f) = \{0, 1\}$, $F_2(f) = \{01, 10, 00\}$,
$F_3(f) = \{010, 100, 001, 101\}$, $F_4(f) = \{0100, 1001, 0010, 0101, 1010\}$,
$F_5(f) = \{01001, 10010, 00101, 01010, 10100, 00100\}$.

In the case of the power word $p = 01001100011100001111\ldots0^n1^n\ldots$, where
$F_1(p) = \{0, 1\}$, $F_2(p) = \{01, 10, 00, 11\}$,
$F_3(p) = \{010, 100, 000, 001, 011, 111, 110\}$,
$F_4(p) = \{0100, 1001, 0011, 0110, 1100, 1000, 0000, 0001, 0111, 1110, 1111\}$,
the corresponding Rauzy graphs are given in Figure 26.5.

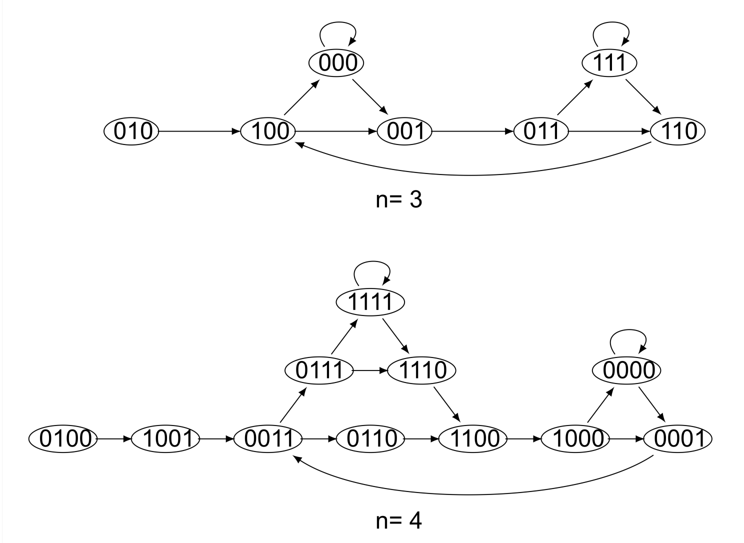


**Figure 26.5** Rauzy graphs for the power word.

As we can see in Fig, 26.4 and 26.5 there are subwords of length $n$ which can be continued only in a single way (by adding a letter), and there are subwords which can be continued in two different ways (by adding two different letters). These latter subwords are called ***special subwords.*** A subword $v \in F_n(u)$ is a ***right special subword,*** if there are at least two different letters $a \in A$, such that $va \in F_{n+1}(u)$. Similarly, $v \in F_n(u)$ is ***left special subword,*** if there are at least two different letters $a \in A$, such that $av \in F_{n+1}(u)$. A subword is ***bispecial,*** if at the same time is right and left special. For example, the special subwords in Figures 26.4 and 26.5) are:

| | |
|---|---|
| left special subwords: | 010, 0100 (Figure 26.4), |
| | 110, 000, 111, 1110, 0001, 1111, 0011 (Figure 26.5), |
| right special subwords:: | 010, 0010 ( Figure 26.4), |
| | 011, 000, 111, 0111, 1111, 0011 (Figure 26.5) |
| bispecial subwords: | 010 (Figure 26.4), |
| | 000, 111, 1111, 0011 (Figure 26.5). |

### 26.1.4. Complexity of words

The complexity of words measures the diversity of the subwords of a word. In this regard the word *aaaaa* has smaller complexity then the word *abcab*.

We define the following complexities for a word.

1) The ***subword complexity*** or simply the ***complexity*** of a word assigns to each $n \in \mathbf{N}$ the number of different subwords of length $n$. For a word $u$ the number

of different subwords of length $n$ is denoted by $f_u(n)$.

$$f_u(n) = \#F_n(u), \quad u \in A^\infty .$$

If the word is finite, then $f_u(n) = 0$, if $n > |u|$.

2) The ***maximal complexity*** is considered only for finite words.

$$C(u) = \max\{f_u(n) \mid n \geq 1, u \in A^*\} .$$

If $u$ is an infinite word, then $C_u^-(n)$ is the ***lower maximal complexity,*** respectively $C_u^+(n)$ the ***upper maximal complexity.***

$$C_u^-(n) = \min_i C(u_i u_{i+1} \dots u_{i+n-1}), \quad C_u^+(n) = \max_i C(u_i u_{i+1} \dots u_{i+n-1}) .$$

3) The ***global maximal complexity*** is defined on the set $A^n$:

$$G(n) = \max\{C(u) \mid u \in A^n\} .$$

4) The ***total complexity*** for a finite word is the number of all different nonempty subwords[5]

$$K(u) = \sum_{i=1}^{|u|} f_u(i), \quad u \in A^* .$$

For an infinite word $K_u^-(n)$ is the ***lower total complexity,*** and $K_u^+(n)$ is the ***upper total complexity:***

$$K_u^-(n) = \min_i K(u_i u_{i+1} \dots u_{i+n-1}), \quad K_u^+(n) = \max_i K(u_i u_{i+1} \dots u_{i+n-1}) .$$

5) A decomposition $u = u_1 u_2 \dots u_k$ is called a ***factorization*** of $u$. If each $u_i$ (with the possible exception of $u_k$) is the shortest prefix of $u_i u_{i+1} \dots u_k$ which does not occur before in $u$, then this factorization is called the Lempel-Ziv factorization. The number of subwords $u_i$ in such a factorization is the ***Lempel-Ziv factorization complexity*** of $u$. For example for the word $u = ababaaabb$ the Lempel-Ziv factorization is: $u = a.b.abaa.abb$. So, the Lempel-Ziv factorization complexity of $u$ is $\textsc{lz}(u) = 4$.

6) If in a factorization $u = u_1 u_2 \dots u_k$ each $u_i$ is the longest possible palindrome, then the factorization is called a palindromic factorization, and the number of subwords $u_i$ in this is the ***palindromic factorization complexity.*** For $u = aababbabbabb = aa.babbabbab.b$, so the palindromic factorization complexity of $u$ is $\textsc{pal}(u) = 3$.

7) The ***window complexity*** $P_w$ is defined for infinite words only. For $u = u_0 u_1 u_2 \dots u_n \dots$ the window complexity is

$$P_w(u, n) = \#\{u_{kn} u_{kn+1} \dots u_{(k+1)n-1} \mid k \geq 0\} .$$

[5] Sometimes the empty subword is considered too. In this case the value of total complexity is increased by 1.

### Subword complexity

As we have seen

$$f_u(n) = \#F_n(u), \quad \forall u \in A^\infty,\ n \in \mathbf{N}\,.$$

$f_u(n) = 0$, if $n > |u|$.

For example, in the case of $u = abacab$:

$$f_u(1) = 3,\ f_u(2) = 4,\ f_u(3) = 4,\ f_u(4) = 3, f_u(5) = 2,\ f_u(6) = 1\,.$$

In Theorem 26.4 was stated that for the infinite Fibonacci word:

$$f_f(n) = n + 1\,.$$

In the case of the power word $p = 010011\ldots 0^k1^k\ldots$ the complexity is:

$$f_p(n) = \frac{n(n+1)}{2} + 1\,.$$

This can be proved if we determine the difference $f_p(n+1) - f_p(n)$, which is equal to the number of words of length $n$ which can be continued in two different ways to obtain words of length $n+1$. In two different ways can be extended only the words of the form $0^k1^{n-k}$ (it can be followed by 1 always, and by 0 when $k \le n-k$) and $1^k0^{n-k}$ (it can be followed by 0 always, and by 1 when $k < n-k$). Considering separately the cases when $n$ is odd and even, we can see that:

$$f_p(n+1) - f_p(n) = n + 1\,,$$

and from this we get

$$\begin{aligned} f_p(n) &= n + f_p(n-1) = n + (n-1) + f_p(n-2) = \ldots \\ &= n + (n-1) + \ldots + 2 + f_p(1) = \frac{n(n+1)}{2} + 1\,. \end{aligned}$$

In the case of the ***Champernowne word***

$$\begin{aligned} c = u_0u_1\ldots u_n\ldots &= 0\,1\,10\,11\,100\,101\,110\,111\,1000\ldots \\ &= 0110111001011101111000\ldots, \end{aligned}$$

the complexity is $f_c(n) = 2^n$.

**Theorem 26.10** *If for the infinite word $u \in A^\omega$ there exists an $n \in \mathbf{N}$ such that $f_u(n) \le n$, then $u$ is ultimately periodic.*

**Proof** $f_u(1) \ge 2$, otherwise the word is trivial (contains just equal letters). Therefore there is a $k \le n$, such that $f_u(k) = f_u(k+1)$. But

$$f_u(k+1) - f_u(k) = \sum_{v \in F_k(u)} \Big(\#\{a \in A \mid va \in F_{k+1}(u)\} - 1\Big)\,.$$

It follows that each subword $v \in F_k(u)$ has only one extension to obtain $va \in$

$F_{k+1}(u)$. So, if $v = u_iu_{i+1}\dots u_{i+k-1} = u_ju_{j+1}\dots u_{j+k-1}$, then $u_{i+k} = u_{j+k}$. Because $F_k(u)$ is a finite set, and $u$ is infinite, there are $i$ and $j$ $(i < j)$, for which $u_iu_{i+1}\dots u_{i+k-1} = u_ju_{j+1}\dots u_{j+k-1}$, but in this case $u_{i+k} = u_{j+k}$ is true too. Then from $u_{i+1}u_{i+2}\dots u_{i+k} = u_{j+1}u_{j+2}\dots u_{j+k}$ we obtain the following equality results: $u_{i+k+1} = u_{j+k+1}$, therefore $u_{i+l} = u_{j+l}$ is true for all $l \geq 0$ values. Therefore $u$ is ultimately periodic. ■

A word $u \in A^\omega$ is ***Sturmian,*** if $f_u(n) = n+1$ for all $n \geq 1$.

Sturmian words are the least complexity infinite and non periodic words. The infinite Fibonacci word is Sturmian. Because $f_u(1) = 2$, the Sturmian words are two-letters words.

From the Theorem 26.10 it follows that each infinite and not ultimately periodic word has complexity at least $n+1$, i.e.

$$u \in A^\omega, u \text{ not ultimately periodic } \Rightarrow f_u(n) \geq n+1\,.$$

The equality holds for Sturmian words.

Infinite words can be characterized using the lower and upper total complexity too.

**Theorem 26.11** *If an infinite word $u$ is not ultimately periodic and $n \geq 1$, then*

$$C_u^+(n) \geq \left[\frac{n}{2}\right] + 1, \quad K_u^+(n) \geq \left[\frac{n^2}{4} + n\right].$$

*For the Sturmian words equality holds.*

Let us denote by $\{x\}$ the fractional part of $x$, and by $\lfloor x \rfloor$ its integer part. Obviously $x = \lfloor x \rfloor + \{x\}$. The composition of a function $R$ by itself $n$ times will be denoted by $R^n$. So $R^n = R \circ R \circ \ldots \circ R$ ($n$ times). Sturmian words can be characterized in the following way too:

**Theorem 26.12** *A word $u = u_1u_2\dots$ is Sturmian if and only if there exists an irrational number $\alpha$ and a real number $z$, such that for $R(x) = \{x+\alpha\}$*

$$u_n = \begin{cases} 0, & \textit{if } R^n(z) \in (0, 1-\alpha)\,, \\ 1, & \textit{if } R^n(z) \in [1-\alpha, 1)\,, \end{cases}$$

*or*

$$u_n = \begin{cases} 1, & \textit{if } R^n(z) \in (0, 1-\alpha)\,, \\ 0, & \textit{if } R^n(z) \in [1-\alpha, 1)\,. \end{cases}$$

In the case of the infinite Fibonacci number, these numbers are: $\alpha = z = (\sqrt{5}+1)/2$.

Sturmian words can be generated by the orbit of a billiard ball inside a square too. A billiard ball is launched under an irrational angle from a boundary point of the square. If we consider an endless move of the ball with reflection on boundaries and without friction, an infinite trajectory will result. We put an 0 in the word if the ball reaches a horizontal boundary, and 1 when it reaches a vertical one. In such a way we generate an infinite word. This can be generalized using an $(s+1)$-letter

| $u$ | $f_u(1)$ | $f_u(2)$ | $f_u(3)$ | $f_u(4)$ | $f_u(5)$ | $f_u(6)$ | $f_u(7)$ | $f_u(8)$ |
|---|---|---|---|---|---|---|---|---|
| 00100011 | 2 | 4 | 5 | 5 | 4 | 3 | 2 | 1 |
| 00100100 | 2 | 3 | 3 | 3 | 3 | 3 | 2 | 1 |
| 00100101 | 2 | 3 | 4 | 4 | 4 | 3 | 2 | 1 |
| 00100110 | 2 | 4 | 5 | 5 | 4 | 3 | 2 | 1 |
| 00100111 | 2 | 4 | 5 | 5 | 4 | 3 | 2 | 1 |
| 00101000 | 2 | 3 | 5 | 5 | 4 | 3 | 2 | 1 |
| 00101001 | 2 | 3 | 4 | 5 | 4 | 3 | 2 | 1 |
| 00101011 | 2 | 4 | 4 | 4 | 4 | 3 | 2 | 1 |
| 01010101 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |
| 11111111 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Figure 26.6** Complexity of several binary words.

alphabet and an $(s+1)$-dimensional hypercube. In this case the complexity is

$$f_u(n, s+1) = \sum_{i=0}^{\min(n,s)} \frac{n!s!}{(n-i)!i!(s-i)!} .$$

If $s = 1$, then $f_u(n,2) = f_u(n) = n+1$, and if $s = 2$, then $f_u(n,3) = n^2 + n + 1$.

### Maximal complexity

For a finite word $u$

$$C(u) = \max\{f_u(n) \mid n \geq 1\}$$

is the maximal complexity. In Figure 26.6 the values of the complexity function for several words are given for all possible length. From this, we can see for example that $C(00100011) = 5$, $C(00100100) = 3$ etc.

For the complexity of finite words the following interesting result is true.

**Theorem 26.13** *If $w$ is a finite word, $f_w(n)$ is its complexity, then there are the natural numbers $m_1$ and $m_2$ with $1 \leq m_1 \leq m_2 \leq |w|$ such that*

- $f_w(n+1) > f_w(n)$, *for* $1 \leq n < m_1$,
- $f_w(n+1) = f_w(n)$, *for* $m_1 \leq n < m_2$,
- $f_w(n+1) = f_w(n) - 1$, *for* $m_2 \leq n \leq |w|$.

From the Figure 26.6, for example, if
$w = 00100011$, then $m_1 = 3$, $m_2 = 4$,
$w = 00101001$, then $m_1 = 4$, $m_2 = 4$,
$w = 00101011$, then $m_1 = 2$, $m_2 = 5$.

### Global maximal complexity

The global maximal complexity is

$$G(n) = \max\{C(u) \mid u \in A^n\} ,$$

| $u$ | $f_u(i)$ $i=1$ | $i=2$ | $i=3$ |
|---|---|---|---|
| 000 | 1 | 1 | 1 |
| 001 | 2 | 2 | 1 |
| 010 | 2 | 2 | 1 |
| 011 | 2 | 2 | 1 |
| 100 | 2 | 2 | 1 |
| 101 | 2 | 2 | 1 |
| 110 | 2 | 2 | 1 |
| 111 | 1 | 1 | 1 |

**Figure 26.7** Complexity of all 3-length binary words

that is the greatest (maximal) complexity in the set of all words of length $n$ on a given alphabet. The following problems arise:

- what is length of the subwords for which the global maximal complexity is equal to the maximal complexity?
- how many such words exist?

**Example 26.1** For the alphabet $A = \{0, 1\}$ the Figure 26.7 and 26.8 contain the complexity of all 3-length and 4-length words.

In this case of the 3-length words (Figure 26.7) the global maximal complexity is 2, and this value is obtained for 1-length and 2-length subwords. There are 6 such words.

For 4-length words (Figure 26.8) the global maximal complexity is 3, and this value is obtained for 2-length words. The number of such words is 8.

To solve the above two problems, the following notations will be used:

$$R(n) = \{i \in \{1, 2, \dots, n\} \mid \exists u \in A^n : f_u(i) = G(n)\},$$

$$M(n) = \#\{u \in A^n : C(u) = G(n)\}.$$

In the table of Figure 26.9 values of $G(n)$, $R(n)$, $M(n)$ are given for length up to 20 over on a binary alphabet.

We shall use the following result to prove some theorems on maximal complexity.

**Lemma 26.14** *For each $k \in \mathbf{N}^*$, the shortest word containing all the $q^k$ words of length $k$ over an alphabet with $q$ letters has $q^k + k - 1$ letters (hence in this word each of the $q^k$ words of length $k$ appears only once).*

**Theorem 26.15** *If $\#A = q$ and $q^k + k \le n \le q^{k+1} + k$, then $G(n) = n - k$.*

**Proof** Let us consider at first the case $n = q^{k+1} + k$, $k \ge 1$.

From Lemma 26.14 we obtain the existence of a word $w$ of length $q^{k+1}+k$ which contains all the $q^{k+1}$ words of length $k+1$, hence $f_w(k+1) = q^{k+1}$. It is obvious that $f_w(l) = q^l < f_w(k+1)$ for $l \in \{1, 2, \dots, k\}$ and $f_w(k+1+j) = q^{k+1} - j < f_w(k+1)$ for $j \in \{1, 2, \dots q^{k+1} - 1\}$. Any other word of length $q^{k+1} + k$ will have the maximal

| $u$ | $f_u(i)$ $i=1$ | $i=2$ | $i=3$ | $i=4$ |
|---|---|---|---|---|
| 0000 | 1 | 1 | 1 | 1 |
| 0001 | 2 | 2 | 2 | 1 |
| 0010 | 2 | 3 | 2 | 1 |
| 0011 | 2 | 3 | 2 | 1 |
| 0100 | 2 | 3 | 2 | 1 |
| 0101 | 2 | 2 | 2 | 1 |
| 0110 | 2 | 3 | 2 | 1 |
| 0111 | 2 | 2 | 2 | 1 |
| 1000 | 2 | 2 | 2 | 1 |
| 1001 | 2 | 3 | 2 | 1 |
| 1010 | 2 | 2 | 2 | 1 |
| 1011 | 2 | 3 | 2 | 1 |
| 1100 | 2 | 3 | 2 | 1 |
| 1101 | 2 | 3 | 2 | 1 |
| 1110 | 2 | 2 | 2 | 1 |
| 1111 | 1 | 1 | 1 | 1 |

**Figure 26.8** Complexity of all 4-length binary words.

complexity less than or equal to $C(w) = f_w(k+1)$, hence we have $G(n) = q^{k+1} = n-k$.

For $k \geq 1$ we consider now the values of $n$ of the form $n = q^{k+1} + k - r$ with $r \in \{1, 2, \ldots, q^{k+1} - q^k\}$, hence $q^k + k \leq n < q^{k+1} + k$. If from the word $w$ of length $q^{k+1} + k$ considered above we delete the last $r$ letters, we obtain a word $w_n$ of length $n = q^{k+1} + k - r$ with $r \in \{1, 2, \ldots, q^{k+1} - q^k\}$. This word will have $f_{w_n}(k+1) = q^{k+1} - r$ and this value will be its maximal complexity. Indeed, it is obvious that $f_{w_n}(k+1+j) = f_{w_n}(k+1) - j < f_{w_n}(k+1)$ for $j \in \{1, 2, \ldots, n-k-1\}$; for $l \in \{1, 2, \ldots, k\}$ it follows that $f_{w_n}(l) \leq q^l \leq q^k \leq q^{k+1} - r = f_{w_n}(k+1)$, hence $C(w_n) = f_{w_n}(k+1) = q^{k+1} - r$. Because it is not possible for a word of length $n = q^{k+1} + k - r$, with $r \in \{1, 2, \ldots, q^{k+1} - q^k\}$ to have the maximal complexity greater than $q^{k+1} - r$, it follows that $G(n) = q^{k+1} - r = n - k$. ■

**Theorem 26.16** *If $\#A = q$ and $q^k + k < n < q^{k+1} + k + 1$ then $R(n) = \{k+1\}$; if $n = q^k + k$ then $R(n) = \{k, k+1\}$.*

**Proof** In the first part of the proof of Theorem 26.15, we proved for $n = q^{k+1} + k$, $k \geq 1$, the existence of a word $w$ of length $n$ for which $G(n) = f_w(k+1) = n - k$. This means that $k + 1 \in R(n)$. For the word $w$, as well as for any other word $w'$ of length $n$, we have $f_{w'}(l) < f_w(k+1)$, $l \neq k+1$, because of the special construction of $w$, which contains all the words of length $k+1$ in the most compact way. It follows that $R(n) = \{k+1\}$.

As in the second part of the proof of Theorem 26.15, we consider $n = q^{k+1} + k - r$

| $n$ | $G(n)$ | $R(n)$ | $M(n)$ |
|---:|---:|---:|---:|
| 1 | 1 | 1 | 2 |
| 2 | 2 | 1 | 2 |
| 3 | 2 | 1, 2 | 6 |
| 4 | 3 | 2 | 8 |
| 5 | 4 | 2 | 4 |
| 6 | 4 | 2, 3 | 36 |
| 7 | 5 | 3 | 42 |
| 8 | 6 | 3 | 48 |
| 9 | 7 | 3 | 40 |
| 10 | 8 | 3 | 16 |
| 11 | 8 | 3, 4 | 558 |
| 12 | 9 | 4 | 718 |
| 13 | 10 | 4 | 854 |
| 14 | 11 | 4 | 920 |
| 15 | 12 | 4 | 956 |
| 16 | 13 | 4 | 960 |
| 17 | 14 | 4 | 912 |
| 18 | 15 | 4 | 704 |
| 19 | 16 | 4 | 256 |
| 20 | 16 | 4, 5 | 79006 |

**Figure 26.9** Values of $G(n)$, $R(n)$, and $M(n)$.

with $r \in \{1, 2, \ldots q^{k+1} - q^k\}$ and the word $w_n$ for which $G(n) = f_{w_n}(k+1) = q^{k+1} - r$. We have again $k+1 \in R(n)$. For $l > k+1$, it is obvious that the complexity function of $w_n$, or of any other word of length $n$, is strictly less than $f_{w_n}(k+1)$. We examine now the possibility of finding a word $w$ with $f_w(k+1) = n - k$ for which $f_w(l) = n - k$ for $l \le k$. We have $f_w(l) \le q^l \le q^k \le q^{k+1} - r$, hence the equality $f_w(l) = n - k = q^{k+1} - r$ holds only for $l = k$ and $r = q^{k+1} - q^k$, that is for $w = q^k + k$.

We show that for $n = q^k + k$ we have indeed $R(n) = \{k, k+1\}$. If we start with the word of length $q^k + k - 1$ generated by the Martin's algorithm (or with another De Bruijn word) and add to this any letter from $A$, we obtain obviously a word $v$ of length $n = q^k + k$, which contains all the $q^k$ words of length $k$ and $q^k = n - k$ words of length $k+1$, hence $f_v(k) = f_v(k+1) = G(n)$. ■

Having in mind the Martin algorithm (or other more efficient algorithms), words $w$ with maximal complexity $C(w) = G(n)$ can be easily constructed for each $n$ and for both situations in Theorem 26.16.

**Theorem 26.17** *If $\#A = q$ and $q^k + k \le n \le q^{k+1} + k$ then $M(n)$ is equal to the number of different paths of length $n - k - 1$ in the de Bruijn graph $B(q, k+1)$.*

**Proof** From Theorems 26.15 and 26.16 it follows that the number $M(n)$ of the words of length $n$ with global maximal complexity is given by the number of words $w \in A^n$ with $f_w(k+1) = n-k$. It means that these words contain $n-k$ subwords of length $k+1$, all of them distinct. To enumerate all of them we start successively with each word of $k+1$ letters (hence with each vertex in $B(q, k+1)$) and we add at each step, in turn, one of the symbols from $A$ which does not duplicate a word of length $k+1$ which has already appeared. Of course, not all of the trials will finish in a word of length $n$, but those which do this, are precisely paths in $B(q, k+1)$ starting with each vertex in turn and having the length $n-k-1$. Hence to each word of length $n$ with $f_w(k+1) = n-k$ we can associate a path and only one of length $n-k-1$ starting from the vertex given by the first $k+1$ letters of the initial word; conversely, any path of length $n-k-1$ will provide a word $w$ of length $n$ which contains $n-k$ distinct subwords of length $k+1$. ■

$M(n)$ can be expressed also as the number of vertices at level $n-k-1$ in the set $\left\{T(q,w) \,\middle|\, w \in A^{k+1}\right\}$ of De Bruijn trees.

**Theorem 26.18** *If $n = 2^k + k - 1$, then $M(n) = 2^{2^{k-1}}$.*

**Proof** In the De Bruijn graph $B(2,k)$ there are $2^{2^{k-1}-k}$ different Hamiltonian cycles. With each vertex of a Hamiltonian cycle a De Bruijn word begins (containing all $k$-length subwords), which has maximal complexity, so $M(n) = 2^k \cdot 2^{2^{k-1}-k} = 2^{2^{k-1}}$, which proves the theorem. ■

A generalization for an alphabet with $q \geq 2$ letters:

**Theorem 26.19** *If $n = q^k + k - 1$, then $M(n) = (q!)^{q^{k-1}}$.*

### Total complexity

The ***total complexity*** is the number of different nonempty subwords of a given word:

$$K(u) = \sum_{i=1}^{|u|} f_u(i)\,.$$

The total complexity of a trivial word of length $n$ (of the form $a^n$, $n \geq 1$) is equal to $n$. The total complexity of a rainbow word (with pairwise different letters) of length $n$ is equal to $\dfrac{n(n+1)}{2}$.

The problem of existence of words with a given total complexity are studied in the following theorems.

**Theorem 26.20** *If $C$ is a natural number different from 1, 2 and 4, then there exists a nontrivial word of total complexity equal to $C$.*

**Proof** To prove this theorem we give the total complexity of the following $k$-length words:

$$\begin{aligned} K(a^{k-1}b) &= 2k-1, && \text{for } k \geq 1\,, \\ K(ab^{k-3}aa) &= 4k-8, && \text{for } k \geq 4\,, \\ K(abcd^{k-3}) &= 4k-6, && \text{for } k \geq 3\,. \end{aligned}$$

These can be proved immediately from the definition of the total complexity.

1. If $C$ is odd then we can write $C = 2k-1$ for a given $k$. It follows that $k = (C+1)/2$, and the word $a^{k-1}b$ has total complexity $C$.

2. If $C$ is even, then $C = 2\ell$.

2.1. If $\ell = 2h$, then $4k-8 = C$ gives $4k-8 = 4h$, and from this $k = h+2$ results. The word $ab^{k-3}aa$ has total complexity $C$.

2.2. If $\ell = 2h+1$ then $4k-6 = C$ gives $4k-6 = 4h+2$, and from this $k = h+2$ results. The word $abcd^{k-3}$ has total complexity $C$. ■

In the proof we have used more than two letters in a word only in the case of the numbers of the form $4h+2$ (case 2.2 above). The new question is, if there exist always nontrivial words formed only of two letters with a given total complexity. The answer is yes anew. We must prove this only for the numbers of the form $4h+2$. If $C = 4h+2$ and $C \geq 34$, we use the followings:

$$\begin{aligned} K(ab^{k-7}abbabb) &= 8k-46, && \text{for } k \geq 10\,, \\ K(ab^{k-7}ababba) &= 8k-42, && \text{for } k \geq 10\,. \end{aligned}$$

If $h = 2s$, then $8k-46 = 4h+2$ gives $k = s+6$, and the word $ab^{k-7}abbabb$ has total complexity $4h+2$.

If $h = 2s+1$, then $8k-42 = 4h+2$ gives $k = s+6$, and the word $ab^{k-7}ababba$ has total complexity $4h+2$. For $C < 34$ only 14, 26 and 30 are feasible. The word $ab^4a$ has total complexity 14, $ab^6a$ has 26, and $ab^5aba$ 30. Easily it can be proved, using a tree, that for 6, 10, 18 and 22 such words does not exist. Then the following theorem is true.

**Theorem 26.21** *If $C$ is a natural number different from 1, 2, 4, 6, 10, 18 and 22, then there exists a nontrivial word formed only of two letters, with the given total complexity $C$.*

The existence of a word with a given length and total complexity is not always assured, as we will prove in what follows.

In relation with the second problem a new one arises: How many words of length $n$ and complexity $C$ there exist? For small $n$ this problem can be studied exhaustively. Let $A$ be of $n$ letters, and let us consider all words of length $n$ over $A$. By a computer program we have got Figure 26.10, which contains the frequency of words with given length and total complexity.

Let $|A| = n$ and let $\phi_n(C)$ denote the frequency of the words of length $n$ over $A$ having a complexity $C$. Then we have the following easy to prove results:

$n = 2$

| $C$ | 2 | 3 |
|---|---|---|
| $\phi_n(C)$ | 2 | 2 |

$n = 3$

| $C$ | 3 | 4 | 5 | 6 |
|---|---|---|---|---|
| $\phi_n(C)$ | 3 | 0 | 18 | 6 |

$n = 4$

| $C$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|
| $\phi_n(C)$ | 4 | 0 | 0 | 36 | 48 | 144 | 24 |

$n = 5$

| $C$ | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\phi_n(C)$ | 5 | 0 | 0 | 0 | 60 | 0 | 200 | 400 | 1140 | 1200 | 120 |

$n = 6$

| $C$ | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|
| $\phi_n(C)$ | 6 | 0 | 0 | 0 | 0 | 90 | 0 | 0 |
| $C$ | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| $\phi_n(C)$ | 300 | 990 | 270 | 5400 | 8280 | 19800 | 10800 | 720 |

**Figure 26.10** Frequency of words with given total complexity

$$\begin{aligned}
&\phi_n(C) = 0, \qquad \text{if } C < n \text{ or } C > \frac{n(n+1)}{2}, \\
&\phi_n(n) = n, \\
&\phi_n(2n-1) = 3n(n-1), \\
&\phi_n\left(\frac{n(n+1)}{2} - 1\right) = \frac{n(n-1)n!}{2}, \\
&\phi_n\left(\frac{n(n+1)}{2}\right) = n!
\end{aligned}$$

As regards the distribution of the frequency 0, the following are true:

If $C = n+1, n+2, \ldots, 2n-2$, then $\phi_n(C) = 0$.
If $C = 2n, 2n+1, \ldots, 3n-5$, then $\phi_n(C) = 0$.

The question is, if there exists a value from which up to $\frac{n(n+1)}{2}$ no more 0 frequency exist. The answer is positive. Let us denote by $b_n$ the least number between $n$ and $n(n+1)/2$ for which

$$\phi_n(C) \neq 0 \quad \text{for all } C \text{ with} \quad b_n \le C \le \frac{n(n+1)}{2}.$$

The number $b_n$ exists for any $n$ (in the worst case it may be equal to $n(n+1)/2$):

**Theorem 26.22** *If $\ell \ge 2$, $0 \le i \le \ell$, $n = \dfrac{\ell(\ell+1)}{2} + 2 + i$, then*

$$b_n = \frac{\ell(\ell^2-1)}{2} + 3\ell + 2 + i(\ell+1).$$

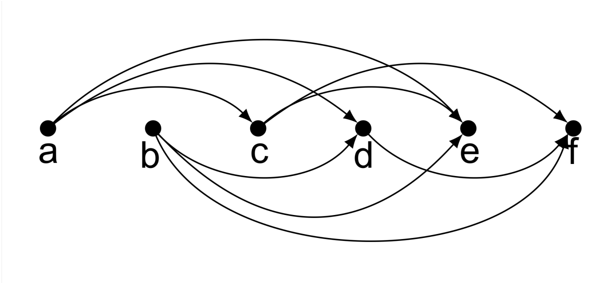


**Figure 26.11** Graph for $(2, 4)$-subwords when $n = 6$.

# 26.2. Generalized complexity measures

As we have seen in the previous section, a contiguous part of a word (obtained by erasing a prefix or/and a suffix) is a ***subword*** or ***factor.*** If we eliminate arbitrary letters from a word, what is obtained is a ***scattered subword,*** sometimes called ***subsequence.*** Special scattered subwords, in which the consecutive letters are at distance at least $d_1$ and at most $d_2$ in the original word, are called $(d_1, d_2)$-***subwords***. More formally we give the following definition.

Let $n$, $d_1 \le d_2$, $s$ be positive integers, and let $u = x_1x_2 \ldots x_n \in A^n$ be a word over the alphabet $A$. The word $v = x_{i_1}x_{i_2} \ldots x_{i_s}$, where

$i_1 \ge 1$,
$d_1 \le i_{j+1} - i_j \le d_2$, for $j = 1, 2, \ldots, s - 1$,
$i_s \le n$,

is a $(d_1, d_2)$-***subword*** of length $s$ of $u$.
For example the $(2, 4)$-subwords of *aabcade* are: *a*, *ab*, *ac*, *aba*, *aa*, *acd*, *abd*, *aae*, *abae*, *ace*, *abe*, *ad*, *b*, *ba*, *bd*, *bae*, *be*, *c*, *cd*, *ce*, *ae*, *d*, *e*.

The number of different $(d_1, d_2)$-subwords of a word $u$ is called $(d_1, d_2)$-***complexity*** and is denoted by $C_u(d_1, d_2)$.
For example, if $u = aabcade$, then $C_u(2, 4) = 23$.

## 26.2.1. Rainbow words

Words with pairwise different letters are called ***rainbow words.*** The $(d_1, d_2)$-complexity of a rainbow word of length $n$ does not depends on what letters it contains, and is denoted by $C(n; d_1, d_2)$.

To compute the $(d_1, d_2)$-complexity of a rainbow word of length $n$, let us consider the word $a_1a_2 \ldots a_n$ (if $i \ne j$, then $a_i \ne a_j$) and the corresponding digraph $G = (V, E)$, with

$V = \{a_1, a_2, \ldots, a_n\}$,
$E = \{(a_i, a_j) \mid d_1 \le j - i \le d_2,\ i = 1, 2, \ldots, n, j = 1, 2, \ldots, n\}$.

For $n = 6, d_1 = 2, d_2 = 4$ see Figure 26.11.
The adjacency matrix $A = \left(a_{ij}\right)_{\substack{i=\overline{1,n}\\ j=\overline{1,n}}}$ of the graph is defined by:

$$a_{ij} = \begin{cases} 1, & \text{if } d_1 \le j - i \le d_2, \\ 0, & \text{otherwise,} \end{cases} \qquad \text{for } i = 1, 2, \ldots, n, j = 1, 2, \ldots, n.$$

Because the graph has no directed cycles, the entry in row $i$ and column $j$ in $A^k$ (where $A^k = A^{k-1}A$, with $A^1 = A$) will represent the number of $k$-length directed paths from $a_i$ to $a_j$. If $A^0$ is the identity matrix (with entries equal to 1 only on the first diagonal, and 0 otherwise), let us define the matrix $R = (r_{ij})$:

$$R = A^0 + A + A^2 + \cdots + A^k, \text{ where } A^{k+1} = O \text{ (the null matrix).}$$

The $(d_1, d_2)$-complexity of a rainbow word is then

$$C(n; d_1, d_2) = \sum_{i=1}^{n} \sum_{j=1}^{n} r_{ij}.$$

The matrix $R$ can be better computed using a variant of the well-known Warshall algorithm:

Warshall$(A, n)$

```
W ← A
for k ← 1 to n
   do for i ← 1 to n
         do for j ← 1 to n
               do w_ij ← w_ij + w_ik w_kj
return W
```

From $W$ we obtain easily $R = A^0 + W$. The time complexity of this algorithms is $\Theta(n^3)$.

For example let us consider the graph in Figure 26.11. The corresponding adjacency matrix is:

$$A = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

After applying the Warshall algorithm:

$$W = \begin{pmatrix} 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad R = \begin{pmatrix} 1 & 0 & 1 & 1 & 2 & 2 \\ 0 & 1 & 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

and then $C(6; 2, 4) = 19$, the sum of entries in $R$.

The Warshall algorithm combined with the Latin square method can be used to obtain all nontrivial (with length at least 2) $(d_1, d_2)$-subwords of a given rainbow word $a_1 a_2 \ldots a_n$ of length $n$. Let us consider a matrix $\mathcal{A}$ with the elements $A_{ij}$ which

$n = 6$

| $d_1 \backslash d_2$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 21 | 46 | 58 | 62 | 63 |
| 2 | - | 12 | 17 | 19 | 20 |
| 3 | - | - | 9 | 11 | 12 |
| 4 | - | - | - | 8 | 9 |
| 5 | - | - | - | - | 7 |

$n = 7$

| $d_1 \backslash d_2$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 28 | 79 | 110 | 122 | 126 | 127 |
| 2 | - | 16 | 25 | 30 | 32 | 33 |
| 3 | - | - | 12 | 15 | 17 | 18 |
| 4 | - | - | - | 10 | 12 | 13 |
| 5 | - | - | - | - | 9 | 10 |
| 6 | - | - | - | - | - | 8 |

**Figure 26.12** $(d_1, d_2)$-complexity for rainbow words of length 6 and 7.

are set of words. Initially this matrix is defined as

$$A_{ij} = \begin{cases} \{a_i a_j\}, & \text{if } d_1 \le j - i \le d_2, \\ \emptyset, & \text{otherwise}, \end{cases} \qquad \text{for } i = 1, 2, \ldots, n, \; j = 1, 2, \ldots, n .$$

If $\mathcal{A}$ and $\mathcal{B}$ are sets of words, $\mathcal{AB}$ will be formed by the set of concatenation of each word from $\mathcal{A}$ with each word from $\mathcal{B}$:

$$\mathcal{AB} = \left\{ ab \,\middle|\, a \in \mathcal{A}, b \in \mathcal{B} \right\} .$$

If $s = s_1 s_2 \ldots s_p$ is a word, let us denote by $'s$ the word obtained from $s$ by erasing the first character: $'s = s_2 s_3 \ldots s_p$. Let us denote by $'A_{ij}$ the set $A_{ij}$ in which we erase from each element the first character. In this case $'\mathcal{A}$ is a matrix with entries $'A_{ij}$.

Starting with the matrix $\mathcal{A}$ defined as before, the algorithm to obtain all nontrivial $(d_1, d_2)$-subwords is the following:

Warshall-Latin$(\mathcal{A}, n)$

$\mathcal{W} \leftarrow \mathcal{A}$
**for** $k \leftarrow 1$ **to** $n$
  **do for** $i \leftarrow 1$ **to** $n$
    **do for** $j \leftarrow 1$ **to** $n$
      **do if** $W_{ik} \neq \emptyset$ and $W_{kj} \neq \emptyset$
        **then** $W_{ij} \leftarrow W_{ij} \cup W_{ik}\, 'W_{kj}$
**return** $\mathcal{W}$

The set of nontrivial $(d_1, d_2)$-subwords is $\displaystyle\bigcup_{i,j \in \{1,2,\ldots,n\}} W_{ij}$. The time complexity is also $\Theta(n^3)$.

For $n = 7$, $d_1 = 2$, $d_2 = 4$, the initial matrix is:

$$\mathcal{A} = \begin{pmatrix} \emptyset & \emptyset & \{ac\} & \{ad\} & \{ae\} & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \{bd\} & \{be\} & \{bf\} & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \{ce\} & \{cf\} & \{cg\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \{df\} & \{dg\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \{eg\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \end{pmatrix},$$

and

$$\mathcal{W} = \begin{pmatrix} \emptyset & \emptyset & \{ac\} & \{ad\} & \{ace, ae\} & \{adf, acf\} & \{aeg, aceg, adg, acg\} \\ \emptyset & \emptyset & \emptyset & \{bd\} & \{be\} & \{bdf, bf\} & \{beg, bdg\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \{ce\} & \{cf\} & \{ceg, cg\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \{df\} & \{dg\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \{eg\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \end{pmatrix}.$$

Counting the one-letter subwords too, we obtain $C(7; 2, 4) = 30$.

### The case $d_1 = 1$

In this case instead of $d_2$ we will use $d$. For a rainbow word, $a_{i,d}$ we will denote the number of $(1, d)$-subwords which finish at the position $i$. For $i = 1, 2, \ldots, n$

$$a_{i,d} = 1 + a_{i-1,d} + a_{i-2,d} + \ldots + a_{i-d,d}\,. \tag{26.1}$$

For simplicity, let us denote $C(n; 1, d)$ by $N(n, d)$. The $(1, d)$-complexity of a rainbow word can be obtained by the formula

$$N(n, d) = \sum_{i=1}^{n} a_{i,d}\,.$$

Because of (26.1) we can write in the case of $d \geq 2$

$$a_{i,d} + \frac{1}{d-1} = \left(a_{i-1,d} + \frac{1}{d-1}\right) + \cdots + \left(a_{i-d,d} + \frac{1}{d-1}\right).$$

Denoting

$$b_{i,d} = a_{i,d} + \frac{1}{d-1}\,, \qquad \text{and} \qquad c_{i,d} = (d-1)b_{i,d}\,,$$

we get

$$c_{i,d} = c_{i-1,d} + c_{i-2,d} + \ldots + c_{i-d,d}\,,$$

and the sequence $c_{i,d}$ is one of Fibonacci-type. For any $d$ we have $a_{1,d} = 1$ and from this $c_{1,d} = d$ results. Therefore the numbers $c_{i,d}$ are defined by the following recurrence equations:

$$\begin{array}{ll} c_{n,d} = c_{n-1,d} + c_{n-2,d} + \ldots + c_{n-d,d}, & \text{for} \quad n > 0, \\ c_{n,d} = 1, & \text{for} \quad n \leq 0. \end{array}$$

These numbers can be generated by the following generating function:

$$F_d(z) = \sum_{n\geq 0} c_{n,d} z^n = \frac{1+(d-2)z-z^2-\cdots-z^d}{1-2z+z^{d+1}}$$
$$= \frac{1+(d-3)z-(d-1)z^2+z^{d+1}}{(1-z)(1-2z+z^{d+1})}\,.$$

The $(1,d)$-complexity $N(n,d)$ can be expressed with these numbers $c_{n,d}$ by the following formula:

$$N(n,d) = \frac{1}{d-1}\left(\sum_{i=1}^{n} c_{i,d} - n\right), \qquad \text{for } d > 1\,,$$

and

$$N(n,1) = \frac{n(n+1)}{2}\,,$$

or

$$N(n,d) = N(n-1,d) + \frac{1}{d-1}(c_{n,d}-1), \qquad \text{for } \quad d>1, \quad n>1\,.$$

If $d=2$ then

$$F_2(z) = \frac{1-z^2}{1-2z+z^3} = \frac{1+z}{1-z-z^2} = \frac{F(z)}{z} + F(z)\,,$$

where $F(z)$ is the generating function of the Fibonacci numbers $F_n$ (with $F_0 = 0, \quad F_1 = 1$). Then, from this formula we have

$$c_{n,2} = F_{n+1} + F_n = F_{n+2}\,,$$

and

$$N(n,2) = \sum_{i=1}^{n} F_{i+2} - n = F_{n+4} - n - 3\,.$$

Figure 26.13 contains the values of $N(n,d)$ for $k \leq 10$ and $d \leq 10$.

$$N(n,d) = 2^n - 1, \qquad \text{for any} \quad d \geq n-1.$$

The following theorem gives the value of $N(n,d)$ in the case $n \geq 2d-2$:

**Theorem 26.23** *For $n \geq 2d-2$ we have*

$$N(n,n-d) = 2^n - (d-2)\cdot 2^{d-1} - 2\,.$$

The main step in the proof is based on the formula

$$N(n,n-d-1) = N(n,n-d) - d\cdot 2^{\mathrm{d}-1}\,.$$

The value of $N(n,d)$ can be also obtained by computing the number of sequences

| $n \backslash d$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 6 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 4 | 10 | 14 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 |
| 5 | 15 | 26 | 30 | 31 | 31 | 31 | 31 | 31 | 31 | 31 |
| 6 | 21 | 46 | 58 | 62 | 63 | 63 | 63 | 63 | 63 | 63 |
| 7 | 28 | 79 | 110 | 122 | 126 | 127 | 127 | 127 | 127 | 127 |
| 8 | 36 | 133 | 206 | 238 | 250 | 254 | 255 | 255 | 255 | 255 |
| 9 | 45 | 221 | 383 | 464 | 494 | 506 | 510 | 511 | 511 | 511 |
| 10 | 55 | 364 | 709 | 894 | 974 | 1006 | 1018 | 1022 | 1023 | 1023 |

**Figure 26.13** The $(1, d)$-complexity of words of length $n$

of length $k$ of 0's and 1's, with no more than $d-1$ adjacent zeros. In such a sequence one 1 represents the presence, one 0 does the absence of a letter of the word in a given $(1, d)$-subword. Let $b_{n,d}$ denote the number of $n$-length sequences of zeros and ones, in which the first and last position is 1, and the number of adjacent zeros is at most $d-1$. Then it can be proved easily that

$$\begin{aligned}
&b_{n,d} = b_{n-1,d} + b_{n-2,d} + \ldots + b_{n-d,d}\,, \text{ for } \quad k > 1\,,\\
&b_{1,d} = 1\,,\\
&b_{n,d} = 0, \text{ for all } n \le 0\,,
\end{aligned}$$

because any such sequence of length $n-i$ $(i = 1, 2, ..., d)$ can be continued in order to obtain a similar sequence of length $n$ in only one way (by adding a sequence of the form $0^{i-1}1$ on the right). For $b_{n,d}$ the following formula also can be derived:

$$b_{n,d} = 2b_{n-1,d} - b_{n-1-d,d}\,.$$

If we add one 1 or 0 at an internal position (e.g at the $(n-2)^{th}$ position) of each $b_{n-1,d}$ sequences, then we obtain $2b_{n-1,d}$ sequences of length $n$, but from these, $b_{n-1-d,d}$ sequences will have $d$ adjacent zeros.

The generating function corresponding to $b_{n,d}$ is

$$B_d(z) = \sum_{n\ge 0} b_{n,d} z^n = \frac{z}{1 - z \cdots - z^d} = \frac{z(1-z)}{1 - 2z + z^{d+1}}\,.$$

By adding zeros on the left and/or on the right to these sequences, we can obtain the number $N(k, d)$, as the number of all these sequences. Thus

$$N(k, d) = b_{k,d} + 2b_{k-1,d} + 3b_{k-2,d} + \cdots + kb_{1,d}\,.$$

($i$ zeros can be added in $i+1$ ways to these sequences: 0 on the left and $i$ on the right, 1 on the left and $i-1$ on the right, and so on).

From the above formula, the generating function corresponding to the complexities $N(k, d)$ can be obtained as a product of the two generating functions $B_d(z)$ and

$A(z) = \sum_{n\geq 0} nz^n = 1/(1-z)^2$, thus:

$$N_d(z) = \sum_{n\geq 0} N(n,d)z^n = \frac{z}{(1-z)(1-2z+z^{d+1})}\,.$$

**The case $d_2 = n-1$**

In the sequel instead of $d_1$ we will use $d$. In this case the distance between two letters picked up to be neighbours in a subword is at least $d$.

Let us denote by $b_{n,d}(i)$ the number of $(d, n-1)$-subwords which begin at the position $i$ in a rainbow word of length $n$. Using our previous example ($abcdef$), we can see that $b_{6,2}(1) = 8$, $b_{6,2}(2) = 5$, $b_{6,2}(3) = 3$, $b_{6,2}(4) = 2$, $b_{6,2}(5) = 1$, and $b_{6,2}(6) = 1$.

The following formula immediately results:

$$b_{n,d}(i) = 1 + b_{n,d}(i+d) + b_{n,d}(i+d+1) + \cdots + b_{n,d}(n)\,, \tag{26.2}$$

for $n > d$, and $1 \leq i \leq n-d$,

$$b_{n,d}(1) = 1 \text{ for } \; n \leq d\,.$$

For simplicity, $C(n;d,n)$ will be denoted by $K(n,d)$.

The $(d,n-1)$-complexity of rainbow words can be computed by the formula:

$$K(n,d) = \sum_{i=1}^{n} b_{n,d}(i)\,. \tag{26.3}$$

This can be expressed also as

$$K(n,d) = \sum_{k=1}^{n} b_{k,d}(1)\,, \tag{26.4}$$

because of the formula

$$K(n+1,d) = K(n,d) + b_{n+1,d}(1)\,.$$

In the case $d = 1$ the complexity $K(n,1)$ can be computed easily: $K(n,1) = 2^n - 1$.

From (26.2) we get the following algorithm for the computation of $b_{n,d}(i)$. The numbers $b_{n,d}(k)$ $(k = 1, 2, \ldots)$ for a given $n$ and $d$ are obtained in the array $b = (b_1, b_2, \ldots)$. Initially all these elements are equal to $-1$. The call for the given $n$ and $d$ and the desired $i$ is:

Input $(n,d,i)$
**for** $k \leftarrow 1$ **to** $n$
  **do** $b_k \leftarrow -1$
B$(n,d,i)$ ▷ Array $b$ is a global one.
Output $b_1,\ b_2,\ \ldots,\ b_n$

| $n \backslash d$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 7 | 4 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 15 | 7 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 31 | 12 | 8 | 6 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 6 | 63 | 20 | 12 | 9 | 7 | 6 | 6 | 6 | 6 | 6 | 6 |
| 7 | 127 | 33 | 18 | 13 | 10 | 8 | 7 | 7 | 7 | 7 | 7 |
| 8 | 255 | 54 | 27 | 18 | 14 | 11 | 9 | 8 | 8 | 8 | 8 |
| 9 | 511 | 88 | 40 | 25 | 19 | 15 | 12 | 10 | 9 | 9 | 9 |
| 10 | 1023 | 143 | 59 | 35 | 25 | 20 | 16 | 13 | 11 | 10 | 10 |
| 11 | 2047 | 232 | 87 | 49 | 33 | 26 | 21 | 17 | 14 | 12 | 11 |
| 12 | 4095 | 376 | 128 | 68 | 44 | 33 | 27 | 22 | 18 | 15 | 13 |

**Figure 26.14** Values of $K(n, d)$.

The recursive algorithm is the following:

B$(n, d, i)$

```
p ← 1
for k ← i + d to n
    do if b_k = −1
           then B(n, d, k)
       p ← p + b_k
b_i ← p
return
```

This algorithm is a linear one.

If the call is $B(8, 2, 1)$, the elements will be obtained in the following order: $b_7 = 1$, $b_8 = 1$, $b_5 = 3$, $b_6 = 2$, $b_3 = 8$, $b_4 = 5$, and $b_1 = 21$.

**Lemma 26.24** *$b_{n,2}(1) = F_n$, where $F_n$ is the nth Fibonacci number.*

**Proof** Let us consider a rainbow word $a_1 a_2 \ldots a_n$ and let us count all its $(2, n-1)$-subwords which begin with $a_2$. If we change $a_2$ for $a_1$ in each $(2, n-1)$-subword which begin with $a_2$, we obtain $(2, n-1)$-subwords too. If we add $a_1$ in front of each $(2, n-1)$-subword which begin with $a_3$, we obtain $(2, n-1)$-subwords too. Thus

$$b_{n,2}(1) = b_{n-1,2}(1) + b_{n-2,2}(1)\,.$$

So $b_{n,2}(1)$ is a Fibonacci number, and because $b_{1,2}(1) = 1$, we obtain that $b_{n,2}(1) = F_n$. ■

**Theorem 26.25** *$K(n, 2) = F_{n+2} - 1$, where $F_n$ is the nth Fibonacci number.*

**Proof** From equation (26.4) and Lemma 26.24:

$$\begin{aligned} K(n,2) &= b_{1,2}(1)+b_{2,2}(1)+b_{3,2}(1)+b_{4,2}(1)+\cdots+b_{n,2}(1) \\ &= F_1+F_2+\cdots+F_n \\ &= F_{n+2}-1\,. \end{aligned}$$

■

If we use the notation $M_{n,d}=b_{n,d}(1)$, because of the formula

$$b_{n,d}(1)=b_{n-1,d}(1)+b_{n-d,d}(1)\,,$$

a generalized middle sequence will be obtained:

$$\begin{aligned} &M_{n,d}=M_{n-1,d}+M_{n-d,d}\,, \quad \text{for } n\ge d\ge 2\,, \\ &M_{0,d}=0,\ M_{1,n}=1,\ \ldots,\ M_{d-1,d}=1\,. \end{aligned} \tag{26.5}$$

Let us call this sequence ***d-middle sequence.*** Because of the equality $M_{n,2}=F_n$, the $d$-middle sequence can be considered as a generalization of the Fibonacci sequence.

Then next linear algorithm computes $M_{n,d}$, by using an array $M_0,M_1,\ldots,M_{d-1}$ to store the necessary previous elements:

Middle$(n,d)$

```
M_0 ← 0
for i ← 1 to d − 1
    do M_i ← 1
for i ← d to n
    do M_{i mod d} ← M_{(i−1) mod d} + M_{(i−d) mod d}
       print M_{i mod d}
return
```

Using the generating function $M_d(z)=\sum_{n\ge 0} M_{n,d}z^n$, the following closed formula can be obtained:

$$M_d(z)=\frac{z}{1-z-z^d}\,. \tag{26.6}$$

This can be used to compute the sum $s_{n,d}=\sum_{n=1}^{n} M_{i,d}$, which is the coefficient of $z^{n+d}$ in the expansion of the function

$$\frac{z^d}{1-z-z^d}\cdot\frac{1}{1-z}=\frac{z^d}{1-z-z^d}+\frac{z}{1-z-z^d}-\frac{z}{1-z}\,.$$

So $s_{n.d}=M_{n+(d-1),d}+M_{n,d}-1=M_{n+d,d}-1$. Therefore

$$\sum_{i=1}^{n} M_{i,d}=M_{n+d,d}-1\,. \tag{26.7}$$

**Theorem 26.26** *$K(n,d) = M_{n+d,d} - 1$, where $n > d$ and $M_{n,d}$ is the nth elements of d-middle sequence.*

**Proof** The proof is similar to that in Theorem 26.25 taking into account the equation (26.7). ■

**Theorem 26.27** $K(n,d) = \sum_{k\geq 0} \binom{n-(d-1)k}{k+1}$, *for* $n \geq 2, d \geq 1$.

**Proof** Let us consider the generating function $G(z) = \dfrac{1}{1-z} = 1 + z + z^2 + \cdots$. Then, taking into account the equation (26.6) we obtain $M_d(z) = zG(z+z^d) = z + z(z+z^d) + z(z+z^d)^2 + \cdots + z(z+z^d)^i + \cdots$. The general term in this expansion is equal to

$$z^{i+1} \sum_{k=1}^{i} \binom{i}{k} z^{(d-1)k},$$

and the coefficient of $z^{n+1}$ is equal to

$$\sum_{k\geq 0} \binom{n-(d-1)k}{k}.$$

The coefficient of $z^{n+d}$ is

$$M_{n+d,d} = \sum_{k\geq 0} \binom{n+d-1-(d-1)k}{k}. \tag{26.8}$$

By Theorem 26.26 $K(n,d) = M_{n+d,d} - 1$, and an easy computation yields

$$K(n,d) = \sum_{k\geq 0} \binom{n-(d-1)k}{k+1}.$$

■

### 26.2.2. General words

The algorithm Warshall-Latin can be used for nonrainbow words too, with the remark that repeating subwords must be eliminated. For the word $aabbbaaa$ and $d_1 = 2$, $d_2 = 4$ the result is: $ab, abb, aba, abba, abaa, aa, aaa, bb, ba, bba, baa$, and with $a$ and $b$ we have $C_{aabbbaaa}(2,4) = 13$.

## 26.3. Palindrome complexity

The ***palindrome complexity function*** $\mathrm{pal}_w$ of a finite or infinite word $w$ attaches to each $n \in \mathbf{N}$ the number of palindrome subwords of length $n$ in $w$, denoted by

$\mathrm{pal}_w(n)$.

The ***total palindrome complexity*** of a finite word $w \in A^*$ is equal to the number of all nonempty palindrome subwords of $w$, i.e.:

$$P(w) = \sum_{n=1}^{|w|} \mathrm{pal}_w(n)\,.$$

This is similar to the total complexity of words.

## 26.3.1. Palindromes in finite words

**Theorem 26.28** *The total palindrome complexity $P(w)$ of any finite word $w$ satisfies $P(w) \le |w|$.*

**Proof** We proceed by induction on the length $n$ of the word $w$. For $n = 1$ we have $P(w) = 1$.

We consider $n \ge 2$ and suppose that the assertion holds for all words of length $n-1$. Let $w = a_1a_2\ldots a_n$ be a word of length $n$ and $u = a_1a_2\ldots a_{n-1}$ its prefix of length $n-1$. By the induction hypothesis it is true that $P(u) \le n-1$.

If $a_n \ne a_j$ for each $j \in \{1, 2, \ldots n-1\}$, the only palindrome in $w$ which is not in $u$ is $a_n$, hence $P(w) = P(u) + 1 \le n$.

If there is an index $j$, $1 \le j \le n-1$ such that $a_n = a_j$, then $P(w) > P(u)$ if and only if $w$ has suffixes which are palindromes. Let us suppose that there are at least two such suffixes $a_ia_{i+1}\ldots a_n$ and $a_{i+k}a_{i+k+1}\ldots a_n$, $1 \le k \le n-i$, which are palindromes. It follows that

$$\begin{aligned}
&a_i = a_n = a_{i+k}\\
&a_{i+1} = a_{n-1} = a_{i+k+1}\\
&\ldots\\
&a_{n-k} = a_{i+k} = a_n,
\end{aligned}$$

hence $a_{i+k}\ldots a_n = a_i\ldots a_{n-k}$. The last palindrome appears in $u$ (because of $k \ge 1$) and has been already counted in $P(u)$. It follows that $P(w) \le P(u) + 1 \le n$. ■

This result shows that the total number of palindromes in a word cannot be larger than the length of that word. We examine now if there are words which are 'poor' in palindromes. In the next lemma we construct finite words $w_n$ of arbitrary length $n \ge 9$, which contain precisely 8 palindromes.

Let us denote by $w^{\frac{p}{q}}$ the fractional power of the word $w$ of length $q$, which is the prefix of length $p$ of $w^p$.

**Lemma 26.29** *If $w_n = (001011)^{\frac{n}{6}}$, $n \ge 9$, then $P(w_n) = 8$.*

**Proof** In $w_n$ there are the following palindromes: 0, 1, 00, 11, 010, 101, 0110, 1001. Because 010 and 101 are situated in $w_n$ between 0 on the left and 1 on the right, these cannot be continued to obtain any palindromes. The same is true for 1001 and 0110, which are situated between 1 on the left and 0 on the right, excepting the cases when 1001 is a suffix. So, there are no other palindromes in $w_n$. ■

**Remark 26.30** *If $u$ is a circular permutation of* 001011 *and $n \geq 9$ then $P(u^{\frac{n}{6}}) = 8$ too. Because we can interchange* 0 *with* 1*, for any $n$ there will be at least* 12 *words of length $n$ with total complexity equal to* 8*.*

We shall give now, beside the upper delimitation from Theorem 26.28, lower bounds for the number of palindromes contained in finite binary words. (In the trivial case of a 1-letter alphabet it is obvious that, for any word $w$, $P(w) = |w|$.)

**Theorem 26.31** *If $w$ is a finite word of length $n$ on a* 2*-letter alphabet, then*

$$\begin{array}{ll} P(w) = n, & \text{for } 1 \leq n \leq 7\,, \\ 7 \leq P(w) \leq 8, & \text{for } n = 8\,, \\ 8 \leq P(w) \leq n, & \text{for } n \geq 9\,. \end{array}$$

**Proof** Up to 8 the computation can be made by a computer program. For $n \geq 9$, Lemma 26.29 gives words $v_n$ for which $P(v_n) = 8$. The maximum value is obtained for words of the form $a^n$, $a \in A$, $n \in \mathbf{N}$. ■

**Remark 26.32** *For all the short binary words (up to $|w| = 7$), the palindrome complexity takes the maximum possible value given in Theorem 26.28; from the words with $|w| = 8$, only four (out of $2^8$) have $P(w) = 7$, namely* 00110100, 00101100 *and their complemented words.*

In the following lemmas we construct binary words which have a given total palindrome complexity greater than or equal to 8.

**Lemma 26.33** *If $u_{k,\ell} = 0^k 10110^\ell 1$ for $k \geq 2$ and $1 \leq \ell \leq k-1$, then $P(u_{k,\ell}) = k+6$.*

**Proof** In the prefix of length $k$ of $u_{k,\ell}$ there are always $k$ palindromes $(1, \ldots, 1^k)$. The other palindromes different from these are 1, 11, 010, 101, 0110 and $10^\ell 1$ (for $\ell \geq 2$), respectively 101101 (for $\ell = 1$). In each case $P(u_{k,\ell}) = k+6$. ■

**Lemma 26.34** *If $v_{k,\ell} = (0^k 1011)^{\frac{k+\ell+5}{k+4}}$ for $k \geq 2$ and $k \leq \ell \leq n-k-5$, then $P(v_{k,\ell}) = k+6$.*

**Proof** Since $\ell \geq k$, the prefix of $u_{k,j}$ is at least $0^k 10110^k 1$, which includes the palindromes $0, \ldots, 0^k$, 1, 11, 010, 101, 0110 and $10^k 1$, hence $P(v_{k,\ell}) \geq k+6$. The palindromes 010 and 101 are situated between 0 and 1, while 0110 and $10^k 1$ are between 1 and 0 (excepting the cases when they are suffixes), no matter how large is $\ell$. It follows that $v_{k,\ell}$ contains no other palindromes, hence $P(v_{k,\ell}) = k+6$. ■

**Remark 26.35** *If $k = 2$, then the word $v_{2,\ell}$ is equal to $w_{\ell+7}$, with $w_n$ defined in Lemma 26.29.*

We can determine now precisely the image of the restriction of the palindrome complexity function to $A^n$, $n \geq 1$.

**Theorem 26.36** *Let $A$ be a binary alphabet. Then*

$$P(A^n) = \begin{cases} \{n\}, & \text{for } 1 \leq n \leq 7\,, \\ \{7, 8\}, & \text{for } n = 8\,, \\ \{8, \dots, n\}, & \text{for } n \geq 9\,. \end{cases}$$

**Proof** Having in mind the result in Theorem 26.31, we have to prove only that for each $n$ and $i$ so that $8 \leq i \leq n$, there exists always a binary word $w_{n,i}$ of length $n$ for which the total palindrome complexity is $P(w_{n,i}) = i$. Let $n$ and $i$ be given so that $8 \leq i \leq n$. We denote $k = i - 6 \geq 2$ and $\ell = n - k - 5$.

If $\ell \leq k - 1$, we take $w_{n,i} = u_{k,\ell}$ (from Lemma 26.33); if $\ell \geq k$, $w_{n,i} = v_{k,\ell}$ (from Lemma 26.34). It follows that $|w_{n,i}| = n$ and $P(w_{n,i}) = k + 6 = i$. ■

**Example 26.2** Let us consider $n = 25$ and $i = 15$. Then $k = 15-6 = 9$, $\ell = 26-15 = 11$.

Because $\ell > k - 1$, we use $v_{9,11} = (0^9 1011)^{\frac{25}{13}} = 0^9 10110^9 101$, whose total palindrome complexity is 15.

We give similar results for the case of alphabets with $q \geq 3$ letters.

**Theorem 26.37** *If $w$ is a finite word of length $n$ over a $q$-letter ($q \geq 3$) alphabet, then*

$$\begin{array}{ll} P(w) = n, & \text{for } n \in \{1, 2\}\,, \\ 3 \leq P(w) \leq n, & \text{for } n \geq 3\,. \end{array}$$

**Proof** For $n \in \{1, 2\}$ it can be checked directly. Let us consider now $n \geq 3$ and a word of length at least 3. If this is a trivial word (containing only one letter $n$ times), its total palindrome complexity is $n \geq 3$. If in the word there appear exactly two letters $a_1$ and $a_2$, it will have as palindromes those two letters and at least one of $a_1^2$, $a_2^2$, $a_1a_2a_1$ or $a_2a_1a_2$, hence again $P(w) \geq 3$. If the word contains a third letter, then obviously $P(w) \geq 3$. So, the total complexity cannot be less then 3. ■

**Theorem 26.38** *Let $A$ be a $q$-letter ($q \geq 3$) alphabet. Then for*

$$P(A^n) = \begin{cases} \{n\}, & \text{for } 1 \leq n \leq 2\,, \\ \{3, \dots, n\}, & \text{for } n \geq 3\,. \end{cases}$$

**Proof** It remains to prove that for each $n$ and $i$ so that $3 \leq i \leq n$, there exists always a word $w_{n,i}$ of length $n$, for which the total palindrome complexity is $P(w_{n,i}) = i$. Such a word is $w_{n,i} = a_1^{i-3}(a_1a_2a_3)^{\frac{n-i+3}{3}}$, which has $i - 2$ palindromes in its prefix of length $i - 2$, and other two palindromes $a_2$ and $a_3$ in what follows. ■

## 26.3.2. Palindromes in infinite words

### Sturmian words

The number of palindromes in the infinite Sturmian words is given by the following theorem.

**Theorem 26.39** *If $u$ is an infinite Sturmian word, then*

$$\mathrm{pal}_u(n) = \begin{cases} 1, & \textit{if } n \textit{ is even}, \\ 2, & \textit{if } n \textit{ is odd}. \end{cases}$$

### Power word

Let us recall the power word as being
$p = 01001100011100001111\ldots 0^n 1^n \ldots$.

**Theorem 26.40** *The palindrome complexity of the power word $p$ is*

$$\mathrm{pal}_p(n) = 2\left\lfloor \frac{n}{3} \right\rfloor + 1 + \varepsilon,$$

*where*

$$\varepsilon = \begin{cases} 0, & \textit{if } n \textit{ divisible by } 3, \\ 1, & \textit{otherwise}. \end{cases}$$

**Proof** There exist the following cases:

Case $n = 3k$. Palindrome subwords are:

$$\begin{array}{lll} 0^i 1^{3k-2i} 0^i & \text{for } i = 0, 1, \ldots k, & \\ 1^i 0^{3k-2i} 1^i & \text{for } i = 0, 1, \ldots k-1, & \text{so } \mathrm{pal}_p(3k) = 2k+1. \end{array}$$

Case $n = 3k+1$. Palindrome subwords are:

$$\begin{array}{lll} 0^i 1^{3k+1-2i} 0^i & \text{for } i = 0, 1, \ldots k, & \\ 1^i 0^{3k+1-2i} 1^i & \text{for } i = 0, 1, \ldots k, & \text{so } \mathrm{pal}_p(3k+1) = 2k+2. \end{array}$$

Case $n = 3k+2$. Palindrome subwords are:

$$\begin{array}{lll} 0^i 1^{3k+2-2i} 0^i & \text{for } i = 0, 1, \ldots k, & \\ 1^i 0^{3k+2-2i} 1^i & \text{for } i = 0, 1, \ldots k, & \text{so } \mathrm{pal}_p(3k+2) = 2k+2. \end{array}$$ ■

The palindrome subwords of the power word have the following properties:
• Every palindrome subword which contains both 0's and 1's occurs only once in the power word.
• If we use the notations $U_{iji} = 0^i 1^j 0^i$ and $V_{iji} = 1^i 0^j 1^i$ then there are the unique decompositions:

$$p = U_{111} U_{121} U_{232} U_{242} U_{353} U_{363} \ldots U_{k,2k-1,k} U_{k,2k,k} \ldots,$$

$$p = 0 V_{121} V_{232} V_{141} V_{353} V_{262} V_{474} V_{383} \ldots V_{k+1,2k+1,k+1} V_{k,2k+2,k} \ldots.$$

### Champernowne word

The Champernowne word is defined as the concatenation of consecutive binary written natural numbers:

$c = 0\,1\,10\,11\,100\,101\,110\,111\,1000\,1001\,1010\,1011\,1100\ldots$ .

**Theorem 26.41** *The palindrome complexity of the Champernowne word is*

$$\mathrm{pal}_c(n) = 2^{\lfloor \frac{n}{2} \rfloor + \varepsilon},$$

*where*

$$\varepsilon = \begin{cases} 0, & \textit{if } n \textit{ is even}, \\ 1, & \textit{if } n \textit{ is odd}. \end{cases}$$

**Proof** Any palindrome $w$ of length $n$ can be continued as $0w0$ and $1w1$ to obtain palindromes of length $n+2$. This theorem results from the following: $\mathrm{pal}_c(1) = 2$, $\mathrm{pal}_c(2) = 2$ and for $n \geq 1$ we have

$\mathrm{pal}_c(2n+1) = 2\mathrm{pal}_c(2n-1)$,
$\mathrm{pal}_c(2n+2) = 2\mathrm{pal}_c(2n)$. ■

The following algorithm generates all palindromes up to a given length of a Sturmian word beginning with the letter $a$, and generated by the morphism $\sigma$.

The idea is the following. If $p$ is the least value for which $\sigma^p(a)$ and $\sigma^p(b)$ are both of odd length (such a $p$ always exists), we consider conjugates[6] of these words, which are palindromes (such conjugates always exists), and we define the following morphism:

$\pi(a) = \mathrm{conj}\big(\sigma^p(a)\big)$,
$\pi(b) = \mathrm{conj}\big(\sigma^p(b)\big)$,

where $\mathrm{conj}(u)$ produces a conjugate of $u$, which is a palindrome.

The sequences $\big(\pi^n(a)\big)_{n\geq 0}$ and $\big(\pi^n(b)\big)_{n\geq 0}$ generate all odd length palindromes, and the sequence $\big(\pi^n(aa)\big)_{n\geq 0}$ all even length palindromes.

If $\alpha$ is a word, then $'\alpha'$ represents the word which is obtained from $\alpha$ by erasing its first and last letter. More generally, $^{m\prime}\alpha^{\prime m}$ is obtained from $\alpha$ by erasing its first $m$ and last $m$ letters.

[6] If $w = uv$ then $vu$ is a conjugate of $w$.

Sturmian-Palindromes($n$)

**if** $n$ is even<br>
   **then** $n \leftarrow n - 1$<br>
let $p$ be the least value for which $\sigma^p(a)$ and $\sigma^p(b)$ are both of odd length<br>
let define the morphism: $\pi(a) = \text{conj}\big(\sigma^p(a)\big)$ and $\pi(b) = \text{conj}\big(\sigma^p(b)\big)$<br>
$\alpha \leftarrow a$<br>
**while** $|\alpha| < n$<br>
      **do** $\alpha \leftarrow \pi(\alpha)$<br>
$m \leftarrow (|\alpha| - n)/2$<br>
$\alpha \leftarrow {}^{m\prime}\alpha^{\prime m}$<br>
$\beta \leftarrow b$<br>
**while** $|\beta| < n$<br>
      **do** $\beta \leftarrow \pi(\beta)$<br>
$m \leftarrow (|\beta| - n)/2$<br>
$\beta \leftarrow {}^{m\prime}\beta^{\prime m}$<br>
**repeat print** $\alpha, \beta$     ▷ Printing odd length palindromes.<br>
        $\alpha \leftarrow {}'\alpha'$<br>
        $\beta \leftarrow {}'\beta'$<br>
**until** $\alpha = \varepsilon$ and $\beta = \varepsilon$<br>
$\gamma \leftarrow aa$<br>
**while** $|\gamma| < n + 1$<br>
      **do** $\gamma \leftarrow \pi(\gamma)$<br>
$m \leftarrow (|\gamma| - n - 1)/2$<br>
$\gamma \leftarrow {}^{m\prime}\gamma^{\prime m}$<br>
**repeat print** $\gamma$     ▷ Printing even length palindromes.<br>
        $\gamma \leftarrow {}'\gamma'$<br>
**until** $\gamma = \varepsilon$

Because any substitution requires no more than $cn$ steps, where $c$ is a constant, the algorithm is a linear one.

In the case of the Fibonacci word the morphism $\sigma$ is defined by

$\sigma(a) = ab$, $\sigma(b) = a$,

and because

$\sigma(a) = ab$, $\sigma^2(a) = aba$, $\sigma^3(a) = abaab$, $|\sigma^3(a)| = |abaab| = 5$,

$\sigma(b) = a$, $\sigma^(b) = ab$, $\sigma^3(b) = aba$, $|\sigma^3(b)| = |aba| = 3$,

both being odd numbers, $p$ will be equal to 3.

The word $abaab$ is not a palindrome, and for the morphism $\pi$ we will use the adequate conjugate $ababa$, which is a palindrome.

In this case the morphism $\pi$ is defined by

$\pi(a) = ababa$,

$\pi(b) = aba$.

For example, if $n = 14$, the following are obtained:

$\pi^2(a) = ababa\,aba\,ababa\,aba\,ababa$, and then $\alpha = aabaababaabaa$,

$\pi^2(b) = ababa\,aba\,ababa$, and $\beta = ababaabaababa$,

$\pi^3(aa) = ababaabaababaabaababaababaabaababaabaababa$, and

$\gamma = baababaababaab$.

The odd palindromes obtained are:

| | |
|---|---|
| $aabaababaabaa$, | $ababaabaababa$, |
| $abaababaaba$, | $babaabaabab$, |
| $baababaab$, | $abaabaaba$, |
| $aababaa$, | $baabaab$, |
| $ababa$, | $aabaa$, |
| $bab$, | $aba$, |
| $a$, | $b$, |

The even palindromes obtained are:

$baababaababaab$,
$aababaababaa$,
$ababaababa$,
$babaabab$,
$abaaba$,
$baab$,
$aa$.

## Problems

***26-1 Generating function 1***

Let $b_{n,d}$ denote the number of sequences of length $n$ of zeros and ones, in which the first and last position is 1, and the number of adjacent zeros is at most $d-1$. Prove that the generating function corresponding to $b_{n,d}$ is

$$B_d(z) = \sum_{n\geq 0} b_{n,d} z^n = \frac{z(1-z)}{1-2z+z^{d+1}} .$$

*Hint.* See Subsection 26.2.1.)

***26-2 Generating function 2***

Prove that the generating function of $N(n,d)$, the number of all $(1,d)$-subwords of a rainbow word of length $n$, is

$$N_d(z) = \sum_{n\geq 0} N(n,d) z^n = \frac{z}{(1-z)(1-2z+z^{d+1})}$$

(*Hint.* (See Subsection 26.2.1.)

***26-3 Window complexity***

Compute the window complexity of the infinite Fibonacci word.

***26-4 Circuits in De Bruijn graphs***

Prove that in the De Bruijn graph $B(q,m)$ there exist circuits (directed cycles) of any length from 1 to $q^m$.

## Chapter Notes

The basic notions and results on combinatorics of words are given in Lothaire's [27, 28, 29] and Fogg's books [19]. Neither Lothaire nor Fogg is a single author, they are pseudonyms of groups of authors. A chapter on combinatorics of words written by Choffrut and Karhumäki [11] appeared in a handbook on formal languages.

The different complexities are defined as follows: total complexity in Iványi [22], maximal and total maximal complexity in Anisiu, Blázsik, Kása [3], $(1,d)$-complexity in Iványi [22] (called $d$-complexity) and used also in Kása [23]), $(d, n-1)$-complexity (called super-$d$-complexity) in Kása [25], scattered complexity in Kása [24], factorization complexity in Ilie [21] and window complexity in Cassaigne, Kaboré, Tapsoba [10].

The power word, lower/upper maximal/total complexities are defined in Ferenczi, Kása [18]. In this paper a characterization of Sturmian words by upper maximal and upper total complexities (Theorem 26.11) is also given. The maximal complexity of finite words is studied in Anisiu, Blázsik, Kása [3]. The total complexity of finite words is described in Kása [23], where the results of the Theorem 26.22 is conjectured too, and proved later by Levé and Séébold [26].

Different generalized complexity measures of words are defined and studied by Iványi [22] and Kása [23, 25, 24].

The results on palindrome complexity are described in M.-C. Anisiu, V. Anisiu, Kása [2] for finite words, and in Droubay, Pirillo [14] for infinite words. The algorithm for palindrome generation in Sturmian words is from this paper too.

Applications of complexities in social sciences are given in Elzinga [16, 15], and in biology in Troyanskaya et al. [31].

It is worth to consult other papers too, such as [4, 9, 13, 17, 30] (on complexity problems) and [1, 5, 6, 7, 8, 12, 20] (on palindromes).

# Bibliography


[1] J.-P. Allouche, M. Baake, J. Cassaigne, D. Damanik. Palindrome complexity. *Theoretical Computer Science*, 292:9–31, 2003. 1337

[2] M.-C. Anisiu, V. Anisiu, Z. Kása. Total palindrome complexity of finite words. *Discrete Mathematics*, 310:109–114, 2010. 1337

[3] M.-C. Anisiu, Z. Blázsik, Z. Kása. Maximal complexity of finite words. *Pure Mathematics and Applications*, 13:39–48, 2002. 1336, 1337

[4] P. Arnoux, C. Mauduit, I. Shiokawa, J. Tamura. Complexity of sequences defined by billiard in the cube. *Bulletin de la Société Mathématique de France*, 122(1):1–12, 1994. MR1259106 (94m:11028). 1337

[5] M. Baake. A note on palindromicity. *Letters in Mathematical Physics*, 49(3):217–227, 1999. 1337

[6] V. Berthé, L. Vuillon. Palindromes and two-dimensional sturmian sequences. *Journal of Automata, Languages and Combinatorics*, 6(2):121–138, 2001. 1337

[7] J-P. Borel, C. Reutenauer. Palindrome factors of billiard words. *Theoretical Computer Science*, 340:334–348, 2005. 1337

[8] S. Brlek, S. Hamel, M. Nivat, C. Reutenauer. On the palindromic complexity of infinite words. *International Journal of Foundations of Computer Science*, 15(2):293–306, 2004. MR2071459 (2005d:68109). 1337

[9] J. Cassaigne. Complexité et facteurs spéciaux. *Bulletin of Bull. Belgian Mathematical Society Simon Stevin*, 4(1):67–88, 1997. 1337

[10] J. Cassaigne, I. Kaboré, T. Tapsoba. On a new notion of complexity on infinite words. *Acta Univ. Sapientiae, Mathematica*, 2:127–136, 2010. 1337

[11] C. Choffrut, J. Karhumäki. Combinatorics of words. In G. Rozemberg (Ed.), *Handbook of Formal Languages,* Vol. I-III. Springer, 1997. MR2071459 (2005d:68109). 1336

[12] D. Damanik, D. Zare. Palindrome complexity bounds for primitive substitution sequences. *Discrete Mathematics*, 222:259–267, 2000. 1337

[13] A. de Luca. On the combinatorics of finite words. *Theoretical Computer Science*, 218(1):13–39, 1999. MR1687752 (2000g:68123). 1337

[14] X. Droubay, G. Pirillo. Palindromes and Sturmian words. *Theoretical Computer Science*, 223(1–2):73–85, 1999. MR1704637 (2000g:68122). 1337

[15] C. H. Elzinga. Complexity of categorial time series. *Sociological Methods & Research*, 38(3):463–481, 2010. 1337

[16] C. H. Elzinga, S. Rahmann, H. Wang. Algorithms for subsequence combinatorics. *Theoretical Computer Science*, 409:394–404, 2008. 1337

[17] S. Ferenczi. Complexity of sequences and dynamical systems. *Discrete Mathematics*, 206(1–3):145–154, 1999. MR1665394 (2000f:68093). 1337

[18] S. Ferenczi, Z. Kása. Complexity for finite factors of infinite sequences. *Theoretical Computer Science*, 218:177–195, 1999. MR1687792 (2000d:68121). 1337

[19] N. P. Fogg. *Substitutions in dynamics, arithmetics, and combinatorics (Lecture Notes in Mathematics, Vol. 1794)*. Springer, 2002. 1336

[20] T. I, S. Inenaga, H. Bannai, M. Takeda. Counting and verifying maximal palindromes. In E. Chavez, S. Lonardi (Eds.), *Proceeding of 17th Internatioonal Symposium on String Processing and Information Retrieval* (Los Cabos, Mexico, October 11–79, 2010), Lecture Notes in Computer Science **6393**, 135–146 pages, 2010. Springer-Verlag. 1337

[21] L. Ilie. Combinatorial complexity measures for strings. In *Recent Advances in Formal Languages*, 149–170 pages. Springer-Verlag, 2006. 1337

[22] A. Iványi. On the $d$-complexity of words. *Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae, Sectio Computarorica*, 8:69–90, 1987. 1336, 1337

[23] Z. Kása. On the $d$-complexity of strings. *Pure Mathematics and Applications*, 9:119–128, 1998. 1337

[24] Z. Kása. On scattered subword complexity of finite words. *Acta Univ. Sapientiae, Informatica*, 3(1), 2010 (accepted). 1337

[25] Z. Kása. Super-$d$-complexity of finite words. In *Proceedings of 8th Joint Conference on Mathematics and Computer Science,* (July 14–17, 2010, Komárno, Slovakia). János Selye University, 2011. 1337

[26] F. Levé, P. Séébold. Proof of a conjecture on word complexity. *Bull. Belg. Math. Soc. Simon Stevin*, 8:277–291, 2001. 1337

[27] M. Lothaire. *Applied Combinatorics on Words*. Cambridge University Press, 2005. 1336

[28] M. Lothaire. *Combinatorics on Words*. Cambridge University Press, 1997. 2nd edition. 1336

[29] M. Lothaire. *Algebraic Combinatorics on Words*. Cambridge University Press, 2002. 1336

[30] J. O. Shallit. On the maximum number of distinct factors in a binary string. *Graphs and Combinatorics*, 9:197–200, 1993. 1337

[31] O. G. Troyanskaya, O. Arbell, Y. Koren, G. M. Landau, A. Bolshoy. Sequence complexity profiles of prokaryotic genomic sequences: A fast algorithm for calculating linguistic complexity. *Bioinformatics*, 18:679–688, 2002. 1337

This bibliography is made by HBibTEX. First key of the sorting is the name of the authors (first author, second author etc.), second key is the year of publication, third key is the title of the document.

Underlying shows that the electronic version of the bibliography on the home page of the book contains a link to the corresponding address.

# Subject Index

This index uses the following conventions. Numbers are alphabetised as if spelled out; for example, "2-3-4-tree" is indexed as if were "two-three-four-tree". When an entry refers to a place other than the main text, the page number is followed by a tag: *exe* for exercise, *exa* for example, *fig* for figure, *pr* for problem and *fn* for footnote.

The numbers of pages containing a definition are printed in *italic* font, e.g.

time complexity, *583*.

# Name Index

This index uses the following conventions. If we know the full name of a cited person, then we print it. If the cited person is not living, and we know the correct data, then we print also the year of her/his birth and death.