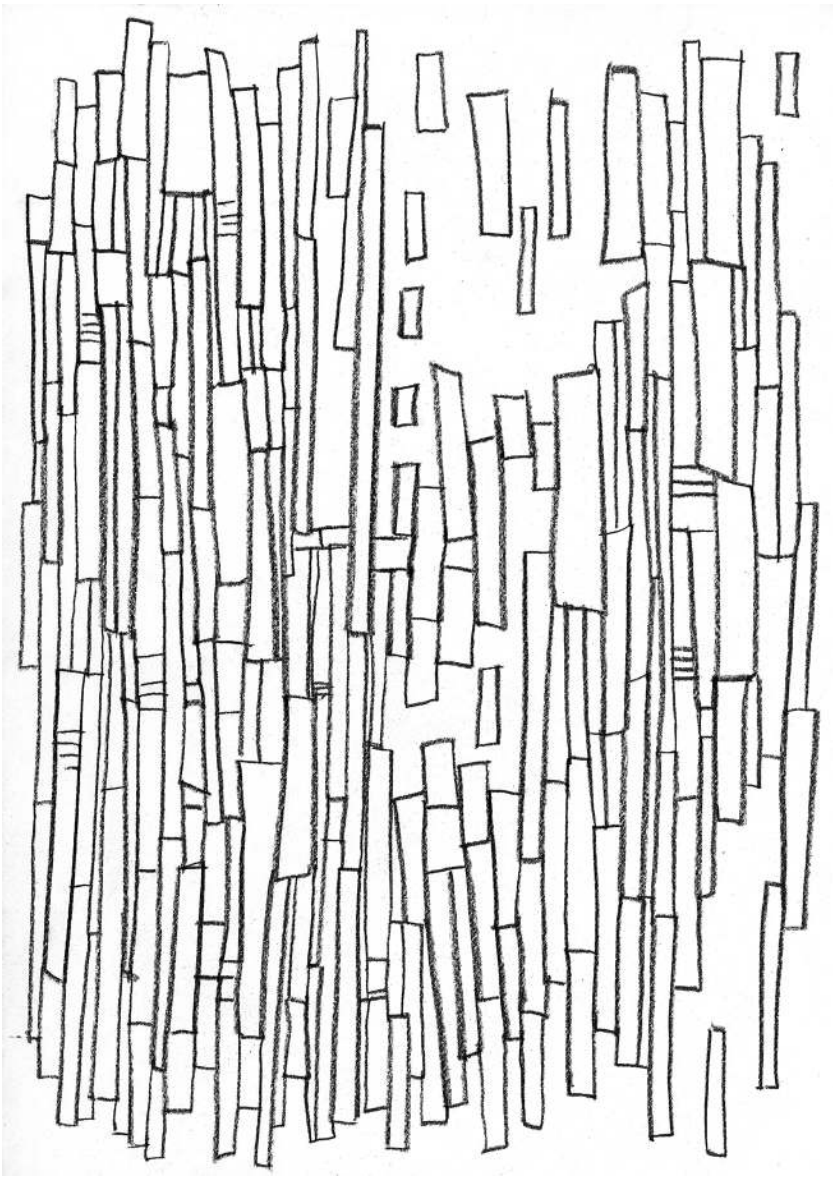
1 The Very Basics of Stringology



In this chapter we introduce basic notation and definitions of words and sketch several constructions used in text algorithms.

Texts are central in 'word processing' systems, which provide facilities for the manipulation of texts. Such systems usually process objects that are quite large. Text algorithms occur in many areas of science and information processing. Many text editors and programming languages have facilities for processing texts. In molecular biology, for example, text algorithms arise in the analysis of biological molecular sequences.

Words

An ***alphabet*** is a non-empty set whose elements are called ***letters*** or symbols. We typically use alphabets $\mathbf{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$, $\mathbf{B} = \{\mathtt{0}, \mathtt{1}\}$ and natural numbers. A ***word*** (*mot*, in French) or ***string*** on an alphabet A is a sequence of elements of A.

The zero letter sequence is called the ***empty word*** and is denoted by ε. The set of all finite words on an alphabet A is denoted by A^*, and $A^+ = A^* \setminus \{\varepsilon\}$.

The ***length*** of a word x, length of the sequence, is denoted by $|x|$. We denote by $x[i]$, for $i = 0, 1, \ldots, |x| - 1$, the letter at ***position*** or ***index*** i on a non-empty word x. Then $x = x[0]x[1]\cdots x[|x| - 1]$ is also denoted by $x[0\,..\,|x|-1]$. The set of letters that occur in the word x is denoted by $alph(x)$. For the example $x = \mathtt{abaaab}$ we have $|x| = 6$ and $alph(x) = \{\mathtt{a}, \mathtt{b}\}$.

The ***product*** or ***concatenation*** of two words x and y is the word composed of the letters of x followed by the letters of y. It is denoted by xy or by $x \cdot y$ to emphasise the decomposition of the resulting word. The neutral element for the product is ε and we denote respectively by zy^{-1} and $x^{-1}z$ the words x and y when $z = xy$.

A ***conjugate***, ***rotation*** or ***cyclic shift*** of a word x is any word y that factorises into vu, where $uv = x$. This makes sense because the product of words is obviously non-commutative. For example, the set of conjugates of $\mathtt{abba}$, its conjugacy class because conjugacy is an equivalence relation, is $\{\mathtt{aabb}, \mathtt{abba}, \mathtt{baab}, \mathtt{bbaa}\}$ and that of $\mathtt{abab}$ is $\{\mathtt{abab}, \mathtt{baba}\}$.

A word x is a ***factor*** (sometimes called ***substring***) of a word y if $y = uxv$ for two words u and v. When $u = \varepsilon$, x is a ***prefix*** of y, and when $v = \varepsilon$, x is a ***suffix*** of y. Sets $Fact(x)$, $Pref(x)$ and $Suff(x)$ denote the sets of factors, prefixes and suffixes of x respectively.

When x is a non-empty factor of $y = y[0\,..\,n-1]$ it is of the form $y[i\,..\,i+|x|-1]$ for some i. An ***occurrence*** of x in y is an interval $[i\,..\,i+|x|-1]$ for which $x = y[i\,..\,i+|x|-1]$. We say that i is the ***starting position*** (or left position) on y of this occurrence, and that $i+|x|-1$ is its ***ending position*** (or right position). An occurrence of x in y can also be defined as a triple (u, x, v) such that $y = uxv$. Then the starting position of the occurrence is $|u|$. For example, the starting and ending positions of $x = \texttt{aba}$ on $y = \texttt{babaababa}$ are

i	0	1	2	3	4	5	6	7	8
$y[i]$	b	a	b	a	a	b	a	b	a
starting positions		1			4		6		
ending positions				3			6		8

For words x and y, $|y|_x$ denotes the number of occurrences of x in y. Then, for instance, $|y| = \Sigma\{|y|_a : a \in alph(y)\}$.

The word x is a ***subsequence*** or ***subword*** of y if the latter decomposes into $w_0x[0]w_1x[1]\ldots x[|x|-1]w_{|x|}$ for words $w_0, w_1, \ldots, w_{|x|}$.

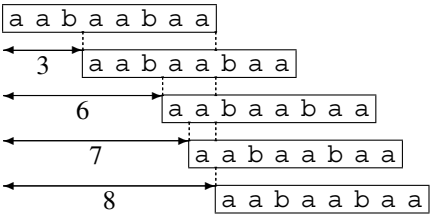
A factor or a subsequence x of a word y is said to be ***proper*** if $x \neq y$.

Periodicity

Let x be a non-empty word. An integer p, $0 < p \leq |x|$, is called a ***period*** of x if $x[i] = x[i+p]$ for $i = 0, 1, \ldots, |x|-p-1$. Note that the length of a word is a period of this word, so every non-empty word has at least one period. ***The period*** of x, denoted by $per(x)$, is its smallest period. For example, 3, 6, 7 and 8 are periods of the word $\texttt{aabaabaa}$, and $per(\texttt{aabaabaa}) = 3$. Note that if p is a period of x, its multiples not larger than $|x|$ are also periods of x.

Here is a series of properties equivalent to the definition of a period p of x. First, x can be factorised uniquely as $(uv)^k u$, where u and v are words, v is non-empty, k is a positive integer and $p = |uv|$. Second, x is a prefix of ux for a word u of length p. Third, x is a factor of u^k, where u is a word of length p and k a positive integer. Fourth, x can be factorised as $uw = wv$ for three words u, v and w, verifying $p = |u| = |v|$.

The last point leads to the notion of border. A ***border*** of x is a proper factor of x that is both a prefix and a suffix of x. ***The border*** of x, denoted by $Border(x)$, is its longest border. Thus, ε, $\texttt{a}$, $\texttt{aa}$, and $\texttt{aabaa}$ are the borders of $\texttt{aabaabaa}$ and $Border(\texttt{aabaabaa}) = \texttt{aabaa}$.

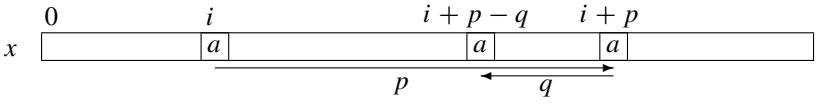


Borders and periods of x are in one-to-one correspondence because of the fourth point above: a period p of x is associated with the border $x[p\,..\,|x|-1]$.

Note that, when defined, the border of a border of x is also a border of x. Then $\langle Border(x), Border^2(x), \ldots, Border^k(x) = \varepsilon\rangle$ is the list of all borders of x. The (non-empty) word x is said to be ***border free*** if its only border is the empty word or equivalently if its only period is $|x|$.

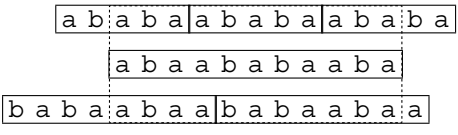
Lemma* 1 *(Periodicity lemma) *If p and q are periods of a word x and satisfy $p+q-\gcd(p,q) \le |x|$ then $\gcd(p,q)$ is also a period of x.*

The proof of the lemma may be found in textbooks (see Notes). The Weak Periodicity lemma refers to a variant of the lemma in which the condition is strengthened to $p+q \le |x|$. Its proof comes readily as follows.



The conclusion obviously holds when $p = q$. Else, w.l.o.g. assume $p > q$ and let us show first that $p-q$ is a period of x. Indeed, let i be a position on x for which $i+p < |x|$. Then $x[i] = x[i+p] = x[i+p-q]$ because p and q are periods. And if $i+p \ge |x|$, the condition implies $i-q \ge 0$. Then $x[i] = x[i-q] = x[i+p-q]$ as before. Thus $p-q$ is a period of x. Iterating the reasoning or using a recurrence as for Euclid's algorithm, we conclude that $\gcd(p,q)$ is a period of x.

To illustrate the Periodicity lemma, let us consider a word x that admits 5 and 8 as periods. Then, if we assume moreover that x is composed of at least two distinct letters, $\gcd(5,8) = 1$ is not a period of x. Thus, the condition of the lemma cannot hold, that is, $|x| < 5+8-\gcd(5,8) = 12$.



The extreme situation is displayed in the picture and shows (when generalised) that the condition required on periods in the statement of the Periodicity lemma cannot be weakened.

Regularities

The powers of a word x are defined by $x^0 = \varepsilon$ and $x^i = x^{i-1}x$ for a positive integer i. The kth ***power*** of x is x^k. It is a ***square*** if k is a positive even integer and a ***cube*** if k is a positive multiple of 3.

The next lemma states a first consequence of the Periodicity lemma.

Lemma 2 *For words x and y, $xy = yx$ if and only if x and y are (integer) powers of the same word. The same conclusion holds when there exist two positive integers k and ℓ for which $x^k = y^\ell$.*

The proofs of the two parts of the lemma are essentially the same (in fact the conclusion derives from a more general statement on codes). For example, if $xy = yx$, both x and y are borders of the word, then both $|x|$ and $|y|$ are periods of it and $\gcd(|x|, |y|)$ as well by the Periodicity lemma. Since $\gcd(|x|, |y|)$ divides also $|xy|$, the conclusion follows. The converse implication is straightforward.

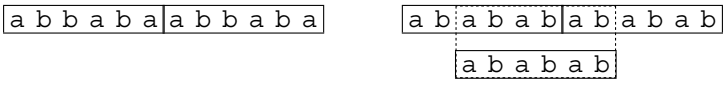
The non-empty word x is said to be ***primitive*** if it is not the power of any other word. That is to say, x is primitive if $x = u^k$, for a word u and a positive integer k, implies $k = 1$ and then $u = x$. For example, `abaab` is primitive, while ε and `bababa` $= (\texttt{ba})^3$ are not.

It follows from Lemma 2 that a non-empty word has exactly one primitive word it is a power of. When $x = u^k$ and u is primitive, u is called the ***primitive root*** of x and k is its ***exponent***, denoted by $\exp(x)$. More generally, the exponent of x is the quantity $\exp(x) = |x|/per(x)$, which is not necessarily an integer, and the word is said to be ***periodic*** if its exponent is at least 2.

Note the number of conjugates of a word, the size of its ***conjugacy class***, is the length of its (primitive) root.

Another consequence of the Periodicity lemma follows.

Lemma 3 (Primitivity Lemma, Synchronisation lemma) *A non-empty word x is primitive if and only if it is a factor of its square only as a prefix and as a suffix, or equivalently if and only if $per(x^2) = |x|$.*



The picture illustrates the result of the lemma. The word `abbaba` is primitive and there are only two occurrences of it in its square, while `ababab` is not primitive and has four occurrences in its square.

The notion of ***run*** or ***maximal periodicity*** encompasses several types of regularities occurring in words. A run in the word x is a maximal occurrence of a periodic factor. To say it more formally, it is an interval $[i \,..\, j]$ of positions on x for which $\exp(x[i \,..\, j]) \geq 2$ and both $x[i-1 \,..\, j]$ and $x[i \,..\, j+1]$ have periods larger than that of $x[i \,..\, j]$ when they exist. In this situation, since the occurrence is identified by i and j, we also say abusively that $x[i \,..\, j]$ is a run.

Another type of regularity consists in the appearance of reverse factors or of palindromes in words. The ***reverse*** or ***mirror image*** of the word x is the word $x^{\mathrm{R}} = x[|x|-1]x[|x|-2]\cdots x[0]$. Associated with this operation is the notion of ***palindrome***: a word x for which $x^{\mathrm{R}} = x$.

For example, `noon` and `testset` are English palindromes. The first is an even palindrome of the form uu^{R} while the second is an odd palindrome of the form uau^{R} with a letter a. The letter a can be replaced by a short word, leading to the notion of gapped palindromes as useful when related to folding operations like those occurring in sequences of biological molecules. As another example, integers whose decimal expansion is an even palindrome are multiples of 11, such as $1661 = 11 \times 151$ or $175571 = 11 \times 15961$.

Ordering

Some algorithms benefit from the existence of an ordering on the alphabet, denoted by $\leq$. The ordering induces the ***lexicographic ordering*** or ***alphabetic ordering*** on words as follows. Like the alphabet ordering, it is denoted by $\leq$. For $x, y \in A^*$, $x \leq y$ if and only if either x is a prefix of y or x and y can be decomposed as $x = uav$ and $y = ubw$ for words u, v and w, letters a and b, with $a < b$. Thus, `ababb` $<$ `abba` $<$ `abbaab` when considering `a` $<$ `b` and more generally the natural ordering on the alphabet $\mathbf{A}$.

We say that x is ***strongly less*** than y, denoted by $x \ll y$, when $x \leq y$ but x is not a prefix of y. Note that $x \ll y$ implies $xu \ll yv$ for any words u and v.

Concepts of ***Lyndon words*** and of ***necklaces*** are built from the lexicographic ordering.

A Lyndon word x is a primitive word that is the smallest among its conjugates. Equivalently but not entirely obvious, x is smaller than all its proper non-empty suffixes, and as such is also called a ***self-minimal word***. As a consequence, x is border-free. It is known that any non-empty word w factorises uniquely into $x_0 x_1 \cdots x_k$, where x_is are Lyndon words and

$x_0 \geq x_1 \geq \cdots \geq x_k$. For example, the word aababaabaaba factorises as aabab · aab · aab · a, where aabab, aab and a are Lyndon words.

A necklace or ***minimal word*** is a word that is the smallest in its conjugacy class. It is a (integer) power of a Lyndon word. A Lyndon word is a necklace but, for example, the word $\mathtt{aabaab} = \mathtt{aab}^2$ is a necklace without being a Lyndon word.

Remarkable Words

Besides Lyndon words, three sets of words have remarkable properties and are often used in examples. They are Thue–Morse words, Fibonacci words and de Bruijn words. The first two are prefixes of (one-way) infinite words. Formally an ***infinite word*** on the alphabet A is a mapping from natural numbers to A. Their set is denoted by A^∞.

The notion of (monoid) ***morphism*** is central to defining some infinite sets of words or an associate infinite word. A morphism from A^* to itself (or another free monoid) is a mapping $h : A^* \mapsto A^*$ satisfying $h(uv) = h(u)h(v)$ for all words u and v. Consequently, a morphism is entirely defined by the images $h(a)$ of letters $a \in A$.

The ***Thue–Morse word*** is produced by iterating the ***Thue–Morse morphism*** μ from $\{\mathtt{a},\mathtt{b}\}^*$ to itself, defined by

$$\begin{cases} \mu(\mathtt{a}) = \mathtt{ab}, \\ \mu(\mathtt{b}) = \mathtt{ba}. \end{cases}$$

Iterating the morphism from letter a gives the list of Thue–Morse words $\mu^k(\mathtt{a})$, $k \geq 0$, that starts with

$$\begin{aligned} \tau_0 = \mu^0(\mathtt{a}) &= \mathtt{a} \\ \tau_1 = \mu^1(\mathtt{a}) &= \mathtt{ab} \\ \tau_2 = \mu^2(\mathtt{a}) &= \mathtt{abba} \\ \tau_3 = \mu^3(\mathtt{a}) &= \mathtt{abbabaab} \\ \tau_4 = \mu^4(\mathtt{a}) &= \mathtt{abbabaabbaababba} \\ \tau_5 = \mu^5(\mathtt{a}) &= \mathtt{abbabaabbaababbabaababbaabbabaab} \end{aligned}$$

and eventually produces its infinite associate:

$$\mathbf{t} = \lim_{k\to\infty} \mu^k(\mathtt{a}) = \mathtt{abbabaabbaababbabaababbaabbabaab}\cdots .$$

An equivalent definition of Thue–Morse words is provided by the following recurrence:

$$\begin{cases} \tau_0 = \mathtt{a}, \\ \tau_{k+1} = \tau_k \overline{\tau_k}, \quad \text{for } k \geq 0, \end{cases}$$

where the bar morphism is defined by $\overline{\mathtt{a}} = \mathtt{b}$ and $\overline{\mathtt{b}} = \mathtt{a}$. Note the length of the kth Thue–Morse word is $|\tau_k| = 2^k$.

A direct definition of $\mathbf{t}$ is as follows: the letter $\mathbf{t}[n]$ is $\mathtt{b}$ if the number of occurrences of digit $\mathtt{1}$ in the binary representation of n is odd, and is $\mathtt{a}$ otherwise.

The infinite Thue–Morse word is known to contain no overlap (factor of the form $auaua$ for a letter a and a word u), that is, no factor of exponent larger than 2. It is said to be ***overlap-free***.

The ***Fibonacci word*** is similarly produced by iterating a morphism, the ***Fibonacci morphism*** ϕ, from $\{\mathtt{a}, \mathtt{b}\}^*$ to itself, defined by

$$\begin{cases} \phi(\mathtt{a}) = \mathtt{ab}, \\ \phi(\mathtt{b}) = \mathtt{a}. \end{cases}$$

Iterating the morphism from letter $\mathtt{a}$ gives the list of Fibonacci words $\phi^k(\mathtt{a})$, $k \geq 0$, that starts with

$$\begin{aligned} fib_0 = \phi^0(\mathtt{a}) &= \mathtt{a} \\ fib_1 = \phi^1(\mathtt{a}) &= \mathtt{ab} \\ fib_2 = \phi^2(\mathtt{a}) &= \mathtt{aba} \\ fib_3 = \phi^3(\mathtt{a}) &= \mathtt{abaab} \\ fib_4 = \phi^4(\mathtt{a}) &= \mathtt{abaababa} \\ fib_5 = \phi^5(\mathtt{a}) &= \mathtt{abaababaabaab} \\ fib_6 = \phi^6(\mathtt{a}) &= \mathtt{abaababaabaababaababa} \end{aligned}$$

and eventually its infinite associate:

$$\mathbf{f} = \lim_{k \to \infty} \phi^k(\mathtt{a}) = \mathtt{abaababaabaababaababaabaababaabab} \cdots .$$

An equivalent definition of Fibonacci words comes from the recurrence relation:

$$\begin{cases} fib_0 = \mathtt{a}, \\ fib_1 = \mathtt{ab}, \\ fib_{k+1} = fib_k fib_{k-1}, \quad \text{for } k \geq 1. \end{cases}$$

The sequence of lengths of these words is the sequence of Fibonacci numbers, that is, $|fib_k| = F_{k+2}$. Recall that ***Fibonacci numbers*** are defined by the recurrence

$$\begin{cases} F_0 = 0, \\ F_1 = 1, \\ F_{k+1} = F_k + F_{k-1}, \quad \text{for } k \geq 1. \end{cases}$$

Among many properties they satisfy are

- $\gcd(F_n, F_{n-1}) = 1$, for $n \geq 2$,
- F_n is the nearest integer of $\Phi^n/\sqrt{5}$, where $\Phi = \frac{1}{2}(1+\sqrt{5}) = 1.61803\cdots$ is the ***golden ratio***.

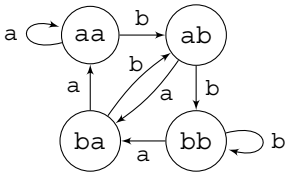
The interest in Fibonacci words comes from the combinatorial properties they satisfy and the large number of repeats they contain. However, the infinite Fibonacci word contains no factor of exponent larger than $\Phi^2 + 1 = 3.61803\cdots$.

De Bruijn words are defined here on the alphabet $A = \{\mathtt{a},\mathtt{b}\}$ and are parameterised by a positive integer k. A word $x \in A^+$ is a de Bruijn word of order k if each word of A^k occurs exactly once in x. As a first example, `ab` and `ba` are the only two de Bruijn words of order 1. As a second example, the word `aaababbbaa` is a de Bruijn word of order 3, since its eight factors of length 3 are the eight words of A^3, that is, `aaa`, `aab`, `aba`, `abb`, `baa`, `bab`, `bba` and `bbb`.

The existence of a de Bruijn word of order $k \geq 2$ can be verified with the help of the ***de Bruijn automaton*** defined by

- States are the words of A^{k-1}.
- Arcs are of the form (av, b, vb) with $a, b \in A$ and $v \in A^{k-2}$.

The picture displays the automaton for de Bruijn words of order 3. Note that exactly two arcs exit each of the states, one labelled by `a`, the other by `b`; and that exactly two arcs enter each of the states, both labelled by the same letter. The graph associated with the automaton thus satisfies the Euler condition: every vertex has an even degree. It follows that there exists an Eulerian circuit in the graph. Its label is a ***circular de Bruijn word***. Appending to it its prefix of length $k-1$ gives an ordinary de Bruijn word.



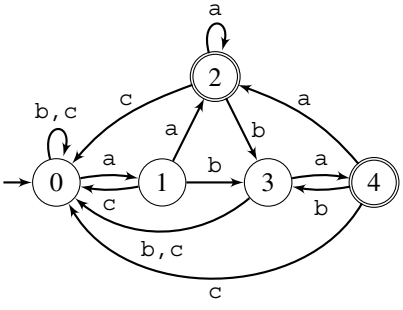
It can also be verified that the number of de Bruijn words of order k is exponential in k.

De Bruijn words can be defined on larger alphabets and are often used as examples of limit cases because they contain all the factors of a given length.

Automata

A finite ***automaton*** M on the finite alphabet A is composed of a finite set Q of ***states***, of an ***initial*** state q_0, of a set $T \subseteq Q$ of ***terminal*** states and of a set $F \subseteq Q \times A \times Q$ of ***labelled edges*** or ***arcs*** corresponding to state ***transitions***. We denote the automaton M by the quadruplet (Q, q_0, T, F) or sometimes by just (Q, F) when, for example, q_0 is implicit and $T = Q$. We say of an arc (p, a, q) that it leaves state p and enters state q; state p is the ***source*** of the arc, letter a its ***label*** and state q its ***target***. A graphic representation of an automaton is displayed below.

The number of arcs exiting a given state is called the ***outgoing degree*** of the state. The ***incoming degree*** of a state is defined in a dual way. By analogy with graphs, the state q is a ***successor*** by the letter a of the state p when $(p, a, q) \in F$; in the same case, we say that the pair (a, q) is a ***labelled successor*** of state p.



A ***path*** of length n in the automaton $M = (Q, q_0, T, F)$ is a sequence of n consecutive arcs $\langle (p_0, a_0, p'_0), (p_1, a_1, p'_1), \ldots, (p_{n-1}, a_{n-1}, p'_{n-1}) \rangle$ that satisfies $p'_k = p_{k+1}$ for $k = 0, 1, \ldots, n-2$. The ***label*** of the path is the word $a_0 a_1 \ldots a_{n-1}$, its ***origin*** the state p_0 and its ***end*** the state p'_{n-1}. A path in the automaton M is ***successful*** if its origin is the initial state q_0 and if its end is in T. A word is ***recognised*** or ***accepted*** by the automaton if it is the label of a successful path. The language composed of the words recognised by the automaton M is denoted by $Lang(M)$.