

Combinatorics on Words

Examples and Problems

Jean Berstel
Institut Gaspard–Monge
Université de Marne–la–Vallée and CNRS (UMR 8049)

Outline

- Words
- Squares
- Finding squares
- Fibonacci
- Thue-Morse
- Some open problems

Words

- Words
 - Syntax: programming and natural languages
 - Text: web, images, television
 - Genomics: structure of genes
- Algorithms
 - Translation : compilation, automata
 - Data processing : pattern matching, image analysis
 - Data compression
- Combinatorics
 - Structure: equations, classification, generation
 - (Un)avoidable regularities
 - Finding regularities
 - Special families of words

The beginning of a small piece in a human gene

```
>ref|NT_029490.4|Hs21_29649:1-490233 Homo sapiens chromosome 21 genomic contig
AATTCTGAGAACTTCTTTGTGAGGGTTGGATTCATTTACACATTTGAACATTTCTTTGATTGAAGATT
TGGAAACAGTCTTTTTTGTAATACTATAAAGGGATAATTGTGAACCTTTGAGGCCTAGGGTGAAGTAGG
AAATATCTTCACATAAAAACTACACAGAAATTTTCTGAGAAACGTTTTAGTGATGCGTGCAATTCATCTCA
CAGAGTTGAACCTTTCCTTTGCTAGAGCACTTTGGAAACAGTCCTATTGTAGAATCCCCAAAGGAATACT
TCTCAGCCGATTGAGGCCTTTGGTGATATTGGAAATATCTTCACATAAAAGCTAGACAGAACTTTCTGA
GAAACTTATTTTTAATGAGTGCTCTCATCTCAAAGAGTTAAGTGTTCCTTTTGAATGAGCAGTTTGGAAA
CACTCTTTTTTGCATAATCTGCAAATGGATAATTGGAGCGTTTTTGAGGCCTATGGTGAAAAAGGAAATATC
TTCACATAAAAACTAAACAGAAGCTTTCTGAGAACTACTTTGTAATGTGTGCATTCATCTCACAGCGTT
GAAAACTTCTTTTGATTGAGCAGTTTGTAACAGTCTTTTTTTGTAGAATCTGCAAATGGGTATTTGGAGT
GCTCTGAGTTCTATAGTGAAAAAGGAAATATCTTCCAAAAAAAACCTAGAAAGAAACATTCTGAGAACTT
CTTTGTGATATGTACTTTCATCTCACAGAGTTGAACCTTTCCTTTTCATTGAGCAGTTTGGAAACAGACTT
TTTATAGAATCTGGAAATGCATATTTGGAGAGCTTTGAGGCCTATGGAGAAAAAGGAAATATCTTCAGAT
AAACACTAAACAGAAGCTTTCTGAGAACTTCTTTGTGATGTCTGCATTCATATCACAGAGCTGAAACTT
TCTTTTGATTTAGCAGTTTGTAACAGTCTTTTGGTAGAATCTGCAAATAGATACTTGGAGTGCTTTGAG
GCCTATGTTGAAAAAGGAAATATCTTCACAAAAAATCTAGAAAGATACATTCTGAGAACTTCTTTGTGA
. . .
```

Repetition

- A **square** is a sequence that is repeated. For instance `ti` is a square in repetition. In `TTCTGAGAACTT`, there are `TT` (twice) and `GAGA`
- A square is called a **tandem repeat** in computational biology.
- A **run** is a maximal repetition (called also **tandem array**). `AAA` is a run.
- A word is **square-free** if it contains no square. For instance, `GTGATGTCTGCAT`.

Questions

- Finding squares is difficult ?
- Avoiding squares is possible ?
- How many square may a word contain ?
- How many square-free words exist ?

Square-free words

- Axel Thue (1906) gives an infinite square-free word over four letters.
- This word is obtained by iterating the morphism

$0 \rightarrow 03121$

$1 \rightarrow 01321$

$2 \rightarrow 01231$

$3 \rightarrow 01213$

- The word starts with

$03121\ 01213\ 01321\ 01231\ 01321\ \dots$

- Construction of the morphism: take 121 , insert 3 at all possible places, and start with 0 .

Another square-free word by Axel Thue

- Axel Thue gives, in the same paper, an infinite ternary square-free word.
- Three step construction, starting with a square-free word, e. g. *abac*

1. Replace *c* by *b**a* if *c* is preceded by *a*, by *a**b* otherwise:

$$abac \rightarrow abab\bar{a}$$

2. Insert a *c* after each letter:

$$abab\bar{a} \rightarrow acb\bar{c}ac\bar{b}\bar{c}ac$$

3. Replace each *a* by *aba* and each *b* by *bab*, and then erase underbars:

$$acb\bar{c}ac\bar{b}\bar{c}ac \rightarrow abacbab\bar{c}abac\bar{b}c\bar{c}ac$$

- Repeat the construction.

Other constructions of this word

The word is

abac babc abac bcac babc abac babc acbc abac babc abac bcac babc acbc abac ...

1. By iterating a (modified) substitution:

$a \mapsto abac$

$b \mapsto babc$

$c \mapsto bcac$ if c is preceded by a

$c \mapsto acbc$ otherwise

This gives

a

abac

abac babc abac bcac

abac babc abac bcac babc abac babc acbc ...

2. By iterating a substitution on four letters and then identifying two of them:

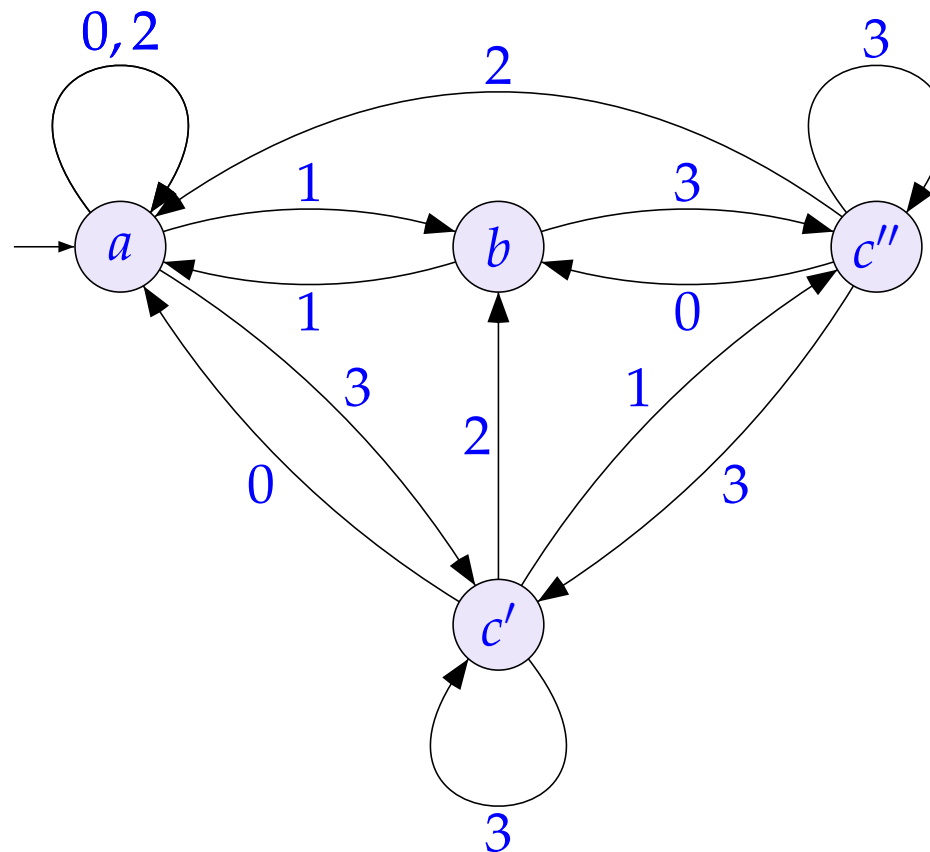
$$\begin{aligned} a &\mapsto abac' \\ b &\mapsto babc'' \\ c' &\mapsto bc''ac' \\ c'' &\mapsto ac'bc'' \end{aligned}$$

and then erase the primes and seconds This gives

$$\begin{aligned} &a \\ &abac' \\ &abac' babc'' abac' bc''ac' \\ &\dots \end{aligned}$$

Recognition of this word by a finite automaton

3. A finite automaton yields explicitly the value of the word at each position:



$13_{10} = 31_4$ and $a \cdot 31 = c' \cdot 1 = c''$, so the thirteenth symbol is c'' .

Detecting squares in a word

There exists a linear time algorithm for testing whether a word is square-free.

It is based on the so-called *c-factorization*:

$$c(x) = (x_1, x_2, \dots, x_m)$$

where each x_k is either a fresh letter, or is the longest factor that appears already before.

$$c(ababaab) = a|b|aba|ab$$

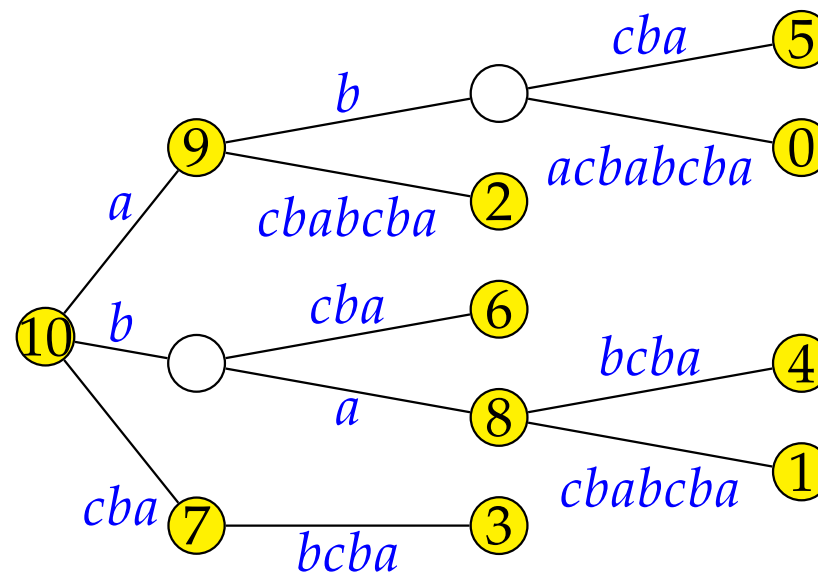
$$c(abacbabcba) = a|b|a|c|ba|b|cba$$

$$c(abaababaabaababaababa) = a|b|a|aba|baaba|ababaaba|ba$$

The computation of the *c*-factorization of x uses the *suffix tree* of the word x .

The suffix tree of a word

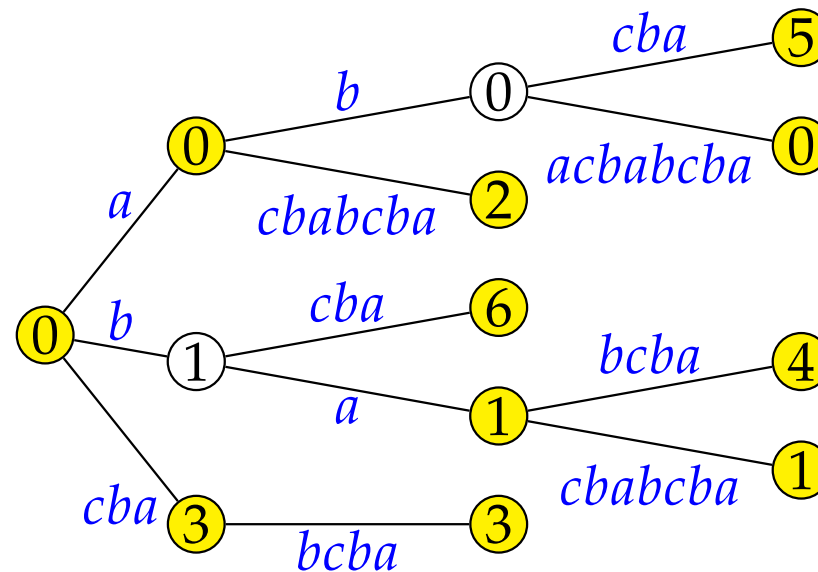
This is the suffix tree of *abacbacba*.



The suffix tree of a word can be computed in linear time.

Augmented suffix tree

At each node, the first occurrence of the factor is reported. For *abacbabcba*:



This gives in linear time the *c*-factorization:

$$c(abacbabcba) = a|b|a|c|ba|b|cba$$

Words with many squares

Theorem *At most $2n$ distinct squares may occur in a word of length n .*

Example The word *ababaababaabab* of length 14 contains 9 squares (this is maximal for a 14-letter word):

a

ab, ba

aba

ababa, babaa, abaab, baaba, aabab

Open It is not known whether there exists a word of length n having more than n occurrences of distinct squares.

Theorem A word of length n contains at most $O(n \log n)$ occurrences of primitive squares.

Example The word

$$f_6 = abaababaabaababaababa$$

of length $21 = F_6$ contains a total of $1 + 2 + 3 + 4 + 4 = 14$ distinct primitive squares:

a

ab, ba

aba, baa, aab

$ababa, abaab, baaba, aabab$

$abaababa, baababaa, aababaab, ababaaba$

and 26 occurrences of primitive squares: $4 \times a + 3 \times ab + 3 \times ba + \dots$.

Theorem The Fibonacci word of length F_n contains $2(F_{n-2} - 1)$ distinct (primitive) squares and $2/5n(F_n + F_{n-2}) - 12/5F_{n-1} - F_{n-2} + n + 1$ occurrences of squares.

Two particular infinite words

- Fibonacci word
 - The most popular of the Sturmian words
 - Has many extremal properties
- Thue-Morse word
 - The most popular of the automatic words
 - Has been introduced by Thue for proving the existence of binary infinite overlap-free words
 - Has been introduced independently by Marston Morse for proving the existence of uniformly recurrent non periodic word

Both words have been generalized in several ways.

Fibonacci word

Defined by $f_0 = a$, $f_1 = ab$, $f_{n+2} = f_{n+1}f_n$. Length of f_n is F_n .

$F_0 = 1$	$f_0 = a$
$F_1 = 2$	$f_1 = ab$
$F_2 = 3$	$f_2 = aba$
$F_3 = 5$	$f_3 = abaab$
$F_4 = 8$	$f_4 = abaababa$
$F_5 = 13$	$f_5 = abaababaabaab$
$F_6 = 21$	$f_6 = abaababaabaababaababa$
$F_7 = 34$	$f_7 = abaababaabaababaababaabaababaabaab$

The infinite Fibonacci word has all finite Fibonacci words as prefixes.

Interpretation of numerical properties

Numerical relation

$$F_n = 2 + F_0 + F_1 + \cdots + F_{n-2}$$

e.g. $F_6 = 21 = 2 + 1 + 2 + 3 + 5 + 8$.

String interpretation

$$f_n = abf_0f_1 \cdots f_{n-2}$$

e.g. $f_6 = ababababababababababab$.

Noncommutativity of words gives richer interpretations:

$$f_n = f_0^R f_1^R \cdots f_{n-2}^R (ba|ab)$$

e.g. $f_6 = ababababababababababab$.

One gets even another interpretations:

$$f_n = aw_0w_1 \cdots w_{n-2}(a|b)$$

e.g. $f_6 = ababababababababababab$.

Fibonacci number system

All natural numbers have a unique binary representation in Fibonacci numbers, provided consecutive Fibonacci numbers are not used.

F_5	F_4	F_3	F_2	F_1	n	f
8	5	3	2	1		
				0	0	a
				1	1	b
			1	0	2	a
		1	0	0	3	a
		1	0	1	4	b
	1	0	0	0	5	a
	1	0	0	1	6	b
	1	0	1	0	7	a
1	0	0	0	0	8	a
1	0	0	0	1	9	b
1	0	0	1	0	10	a

Construction of Fibonacci-even numbers

So n is *Fibonacci-even* if and only $f(n) = a$.

$$E_f = \{0, 2, 3, 5, 7, 8, 10, \dots\}$$

Construction by a *min-excluded* algorithm:

$$\begin{array}{r|ccccccc} n+1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline E_f & 0 & 2 & 3 & 5 & 7 & 8 & \\ O_f & 1 & 4 & 6 & 9 & & & \end{array}$$

The sequences E_f and O_f are complementary Beatty sequences:

$$E_f = \{\lfloor n\tau \rfloor - 1 \mid n \geq 1\}, \quad O_f = \{\lfloor n\tau^2 \rfloor - 1 \mid n \geq 1\}$$

Here $\tau = (1 + \sqrt{5})/2$.

The Thue-Morse word

$$t = 0110100110010110 \dots$$

is obtained by iterating the morphism

$$\mu : \begin{array}{l} 0 \mapsto 01 \\ 1 \mapsto 10 \end{array}$$

It is **overlap-free** : no factor of the form $uvuvu$ with u nonempty.

ID Number: A007777

URL: <http://www.research.att.com/projects/OEIS?Anum=A007777>

Sequence: 1, 2, 4, 6, 10, 14, 20, 24, 30, 36, 44, 48, 60, 60, 62, 72, 82, 88, 96, 112, 120, 120, 136, 148, 164, 152, 154, 148, 162, 176, 190, 196, 210, 216, 224, 228, 248, 272, 284, 296, 300

Name: Number of overlap-free binary words of length n .

Arithmetic definition of the Thue-Morse word

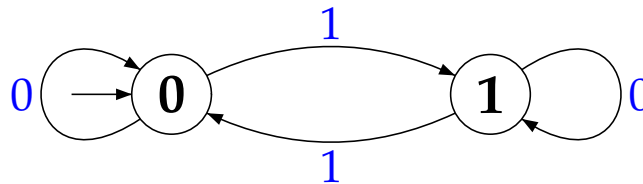
$$t = 0110100110010\mathbf{1}10 \dots$$

$$t(n) = \begin{cases} 0 & \text{if } d_2(n) \equiv 0 \pmod{2} \\ 1 & \text{otherwise} \end{cases}$$

where $d_2(n)$ is the sum of the bits of the binary expansion of n .

$13_{10} = 1101_2$, so $d_2(n) = 3 \equiv 1 \pmod{2}$, and $t(13) = 1$.

The word t is 2-automatic



$$t = 0110100110010110 \dots$$
[illegible]

16			13
	11	10	
	7	6	
4			1

Higher size and dimension

The construction

16			13
	11	10	
	7	6	
4			1

extends to sizes that are powers of 2 (excepted 2):

64	2	3	61	5	59	58	8
9	55	54	12	52	14	15	49
17	47	46	20	44	22	23	41
40	26	27	37	29	35	34	32
33	31	30	36	28	38	39	25
24	42	43	21	45	19	18	48
16	50	51	13	53	11	10	56
57	7	6	60	4	62	63	1

and can produce magic cubes etc.

Run-length encoding of the Thue-Morse sequence

- Sequence 0110100110010110100101100...
- Run-length 12112221121121122...
- Summation $S = 1, 3, 4, 5, 7, 9, 11, 12, 13, 15, 16, 17, 19, 20, 21, 23, 25 \dots$
- The set S is the smallest set of positive integers (for the lexicographic order) such that $n \in S$ if and only if $2n \notin S$.
- Construction by a *min-excluded* algorithm:

S	1	3	4	5	7	9	11	12	13
\bar{S}	2	6	8	10	14				

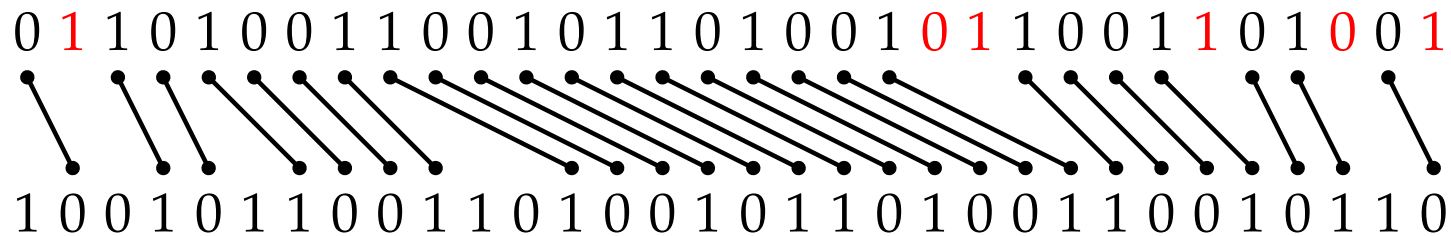
An open problem

Define the **Morse blocks** u_n and v_n by: $u_{n+1} = u_n v_n$ and $v_{n+1} = v_n u_n$, with $u_0 = 0, v_0 = 1$. Thus

$$u_1 = 01, v_1 = 10$$

$$u_2 = 0110, v_2 = 1001$$

$$u_3 = 01101001, v_3 = 10010110$$



Denote by e_n the length of a maximal common subword of u_n and v_n . The sequence e_n starts with 1, 2, 5, 12, 26, 54, 110, 226, 462, 942, 1908, ...

What is the formula for e_n ?