Combinatorics on Finite Words and Data Structures

Gabriele Fici

Dipartimento di Informatica ed Applicazioni
Università di Salerno (Italy)

Laboratoire I3S - Université de Nice-Sophia Antipolis

13 March 2009
Combinatorics of Words

AMS 2000 Mathematics Subject Classification: 68R15

Number Theory

Discrete Dynamical Systems

Algorithmics

Logic

Probability Theory

Algebra

Bioinformatics

Automata Theory
Combinatorics of Words

Gabriele Fici: Combinatorics of Finite Words and Suffix Automata. Submitted.
$A$ is a finite set of letters (the **alphabet**).

A **finite word** $w$ is an element of $A^*$.

Its **length** $|w|$ is the number of its letters.

The **empty word** $\varepsilon$ has length 0.

Let $w = a_1 a_2 \ldots a_n$ be a word.

- $a_1 \ldots a_i$, with $1 \leq i \leq n$, and $\varepsilon$ are the **prefixes** of $w$.
- $a_j \ldots a_n$, with $1 \leq j \leq n$, and $\varepsilon$ are the **suffixes** of $w$.
- $a_j \ldots a_i$, with $1 \leq i, j \leq n$, and $\varepsilon$ are the **factors** of $w$. 
Example

$A = \{a, n, b, c\}, \quad w = \text{banana}$

$|\text{banana}| = 6$

$ba$ is a prefix of $\text{banana}$

$nana$ is a suffix of $\text{banana}$

$a, ba, \varepsilon, \text{banana}$ are factors of $\text{banana}$
Some famous classes of finite words:

- palindromes: $w^R = w$. Ex. *level*.
Some famous classes of finite words:

- **palindromes**: $w^R = w$. Ex. *level*.
- **balanced words** (over two letters): all the factors of the same length have the same number of *a*'s and *b*'s up to 1. Ex. *abaababaabaab*.
Some famous classes of finite words:

- palindromes: $w^R = w$. Ex. *level*.
- balanced words (over two letters): all the factors of the same length have the same number of $a$’s and $b$’s up to 1. Ex. *abaababaabaab*.
- differentiable words: words over \{1, 2\} such that their Run Length Encoding is still a word over \{1, 2\}. Ex. 22112122122111
Some famous classes of finite words:

- **palindromes**: $w^R = w$. Ex. *level*.

- **balanced words** (over two letters): all the factors of the same length have the same number of $a$’s and $b$’s up to 1. Ex. *abaababaabaab*.

- **differentiable words**: words over \{1, 2\} such that their Run Length Encoding is still a word over \{1, 2\}. Ex. 2211212212211

- **finite prefixes of (right) infinite words**: Thue-Morse, Fibonacci, Kolakoski,...
Some famous classes of finite words:

- palindromes: \( w^R = w \). Ex. \textit{level}.

- balanced words (over two letters): all the factors of the same length have the same number of \( a \)'s and \( b \)'s up to 1. Ex. \textit{abaababaabaab}.

- differentiable words: words over \( \{1, 2\} \) such that their Run Length Encoding is still a word over \( \{1, 2\} \). Ex. \textit{2211212212211}.

- finite prefixes of (right) infinite words: Thue-Morse, Fibonacci, Kolakoski, ...

- many many others.
Some famous classes of finite words:

- palindromes: $w^R = w$. Ex. *level*.
- balanced words (over two letters): all the factors of the same length have the same number of $a$’s and $b$’s up to 1. Ex. *abaababaabaab*.
- differentiable words: words over \{1, 2\} such that their Run Length Encoding is still a word over \{1, 2\}. Ex. 2211212212211
- finite prefixes of (right) infinite words: Thue-Morse, Fibonacci, Kolakoski,...
- many many others.

Intersections: 12112112121 is a balanced differentiable palindromic prefix of the Fibonacci word over \{1, 2\}...
What’s the target?

Classify the words through their combinatorial properties.
The suffix automaton

**Definition (Blumer et al. 1985 - Crochemore 1986)**

The **suffix automaton** of the word $w$ is the minimal deterministic automaton recognizing the suffixes of $w$.

**Example**

The suffix automaton of $aabbabb$:
Theorem (Blumer et al. 1985 - Crochemore 1986)

The suffix automaton of a word $w$ over a fixed alphabet $A$ can be built in time and space $O(|w|)$. 
One way to build the SA

Build a non-deterministic automaton:

\[ w = aabbabb \]

\[
\begin{array}{cccccccc}
0 & a & 1 & a & 2 & b & 3 & b & 4 & a & 5 & b & 6 & b & 7 \\
\end{array}
\]
One way to build the SA

Build a non-deterministic automaton:

\[ w = aabbabb \]

Determinize by subset construction:
We associate to each factor \( v \) of \( w \) the set of ending positions of \( v \) in \( w \).

**Example**

\[
w = aabbabb\\1234567\\\]

\[
\text{Endset}(b) = \{3, 4, 6, 7\}, \text{Endset}(abb) = \text{Endset}(bb) = \{4, 7\}.
\]
We associate to each factor $v$ of $w$ the set of ending positions of $v$ in $w$.

**Example**

$$w = aabbabb$$

1234567

$\text{Endset}(b) = \{3, 4, 6, 7\}$, $\text{Endset}(abb) = \text{Endset}(bb) = \{4, 7\}$.

We define on $\text{Fact}(w)$ the equivalence:

$$u \sim v \iff \text{Endset}(u) = \text{Endset}(v)$$
We associate to each factor $v$ of $w$ the set of ending positions of $v$ in $w$.

**Example**

$w = aabbabb$

```
1234567
```

$\text{Endset}(b) = \{3, 4, 6, 7\}$, $\text{Endset}(abb) = \text{Endset}(bb) = \{4, 7\}$.

We define on $\text{Fact}(w)$ the equivalence:

$$u \sim v \iff \text{Endset}(u) = \text{Endset}(v)$$

Then $\text{Fact}(w)/\sim$ is the set of states of the SA of $w$. 
The number of states (classes) of the SA is noted $|Q_w|$. The bounds on $|Q_w|$ are well known:

$$|w| + 1 \leq |Q_w| \leq 2|w| - 1$$
The number of states (classes) of the SA is noted $|Q_w|$. The bounds on $|Q_w|$ are well known:

$$|w| + 1 \leq |Q_w| \leq 2|w| - 1$$

The upper bound is reached for $w = ab^{|w|-1}$, with $a \neq b$. 

---

**Gabriele Fici**

**Combinatorics on Finite Words and Data Structures**
The number of states (classes) of the SA is noted $|Q_w|$.  

The bounds on $|Q_w|$ are well known:

$$|w| + 1 \leq |Q_w| \leq 2|w| - 1$$

The upper bound is reached for $w = ab^{|w|-1}$, with $a \neq b$.

And for the lower bound?
**Definition**

- \( v \) is a **left special factor** of \( w \) if there exist \( a \neq b \) such that \( av \) and \( bv \) are factors of \( w \).

- \( v \) is a **right special factor** of \( w \) if there exist \( a \neq b \) such that \( va \) and \( vb \) are factors of \( w \).

- \( v \) is a **bispecial factor** of \( w \) if it is both left and right special.

---

Example: \( w = aabbabb \)

- \( LS = \{ \epsilon, a, b, ab, abb \} \)
- \( RS = \{ \epsilon, a, b \} \)
- \( BIS = \{ \epsilon, a, b \} \)
Special Factors

Definition

- \( v \) is a **left special factor** of \( w \) if there exist \( a \neq b \) such that \( av \) and \( bv \) are factors of \( w \).

- \( v \) is a **right special factor** of \( w \) if there exist \( a \neq b \) such that \( va \) and \( vb \) are factors of \( w \).

- \( v \) is a **bispecial factor** of \( w \) if it is both left and right special.

Example \((w = aabbabb)\)

\[
LS = \{\varepsilon, a, b, ab, abb\}, \quad RS = \{\varepsilon, a, b\}, \quad BIS = \{\varepsilon, a, b\}
\]
The number of states

Theorem (Sciortino, Zamboni 2007)

If $|A| = 2$ then the following conditions are equivalent for a word over $A$:

- $|Q_w| = |w| + 1$
- Every left special factor of $w$ is a prefix of $w$
- $w$ is a prefix of a standard sturmian word.

Without restriction on the cardinality of $A$ we have the formula:

**Lemma**

$$|Q_w| = |w| + 1 + |D(w)|$$

where $D(w)$ is the set of left special factors of $w$ which are not prefixes.
The number of states

**Theorem (Sciortino, Zamboni 2007)**

If $|A| = 2$ then the following conditions are equivalent for a word over $A$:

- $|Q_w| = |w| + 1$
- Every left special factor of $w$ is a prefix of $w$
- $w$ is a prefix of a standard sturmian word.

Without restriction on the cardinality of $A$ we have the formula:

**Lemma**

$$|Q_w| = |w| + 1 + |D(w)|$$

where $D(w)$ is the set of left special factors of $w$ which are not prefixes.
Characterize the class of words having the property that every left special factor is a prefix, over an arbitrary fixed alphabet A.
The binary case

For binary words we can give a more precise formula:

\[ |Q_w| = 2|w| - H_w - P_w \]

*H*<sub>*w*</sub> is the minimal length of a prefix of *w* occurring only once, *P*<sub>*w*</sub> is the maximal length of a left special prefix of *w*. 
The binary case

For binary words we can give a more precise formula:

\[ |Q_w| = 2|w| - H_w - P_w \]

*H*<sub>*w*</sub> is the minimal length of a prefix of *w* occurring only once, *P*<sub>*w*</sub> is the maximal length of a left special prefix of *w*.

As a corollary we obtain a new characterization of standard sturmian words:

**Corollary**

\[ w \text{ is a prefix of a stand. sturm. word } \iff |w| = H_w + P_w + 1. \]
Example (\(w = aabbabb\))

\[H_w = 2\] since \(aa\) occurs only once.
\[P_w = 1\] since \(a\) is left special.

\[|Q_w| = 2 \cdot 7 - 2 - 1 = 11\]
What about the number of edges $\mathcal{E}_w$?
The number of edges

What about the number of edges $E_w$?

The bounds on $E_w$ are well known:

$$|w| \leq E_w \leq 3|w| - 4$$
The number of edges

What about the number of edges $\mathcal{E}_w$?

The bounds on $\mathcal{E}_w$ are well known:

$$|w| \leq \mathcal{E}_w \leq 3|w| - 4$$

For binary words we give the formula:

$$\mathcal{E}_w = |Q_w| + |G(w)| - 1$$

$G(w)$ is the union of the set of bispecial factors of $w$ and the set of right special prefixes of $w$. 
Example \((w = aabbabb)\)

\[
G(w) = BIS(w) \cup (\text{Pref}(w) \cap \text{RS}(w)) = \{\varepsilon, a, b\} \cup \{\varepsilon, a\}
\]

\[
|G(w)| = 3 \quad \Rightarrow \quad \varepsilon_w = 11 + 3 - 1 = 13.
\]
Further Research

Problem

Does this approach can be applied to other data structures (factor oracles, suffix tries, suffix arrays, etc.)?