On-line Construction of Compact Suffix Vectors and Maximal Repeats

Élise Prieur and Thierry Lecroq
elise.prieur@univ-rouen.fr

Laboratoire d’Informatique de Traitement de l’Information et des Systèmes.

Journées Montoises
August 30th, 2006, Rennes
Plan

1. Introduction
2. Suffix Vectors
3. Computing maximal repeats
4. Conclusion
1 Introduction
- Motivation
- Suffix trees
- Ukkonen’s algorithm

2 Suffix Vectors
- Introduction
- Compact Suffix Vectors
- On-line construction of a compact suffix vector

3 Computing maximal repeats

4 Conclusion
Motivation

Detecting repeats in long biological sequences.

Adapted index structure.
**Notations**

$y$ is a sequence of length $n$ on the alphabet $A$.

$\$ is a terminator symbol.

**Suffix tree**

- index structure;
- all substrings represented;
- edges labeled (begin position, length);
- leaves represent suffixes.
Ukkonen’s algorithm

- On-line algorithm
- Construction split into \( n \) phases which are also split into extensions.
- During the phase \( i \), construction of the implicit tree of \( y[0..i] \) from the one of \( y[0..i-1] \).
- During the extension \( j \) of the phase \( i \), the suffix \( y[j+1..i] \) is added to the tree.
- The last added substring is \( w = y[j+1..i-1] \).
Ukkonen’s algorithm is based on 3 rules expressed by Gusfield\(^1\):

**Rule 1**

\[ w = y[j+1...i-1] \]
Ukkonen’s algorithm is based on 3 rules expressed by Gusfield:

**Rule 1**

\[ wy[i] = y[j+1...i] \]
Ukkonen’s algorithm is based on 3 rules expressed by Gusfield:

**Rule 2**

\[ w \rightarrow x \]
Ukkonen’s algorithm is based on 3 rules expressed by Gusfield:

**Rule 2**
Ukkonen’s algorithm is based on 3 rules expressed by Gusfield:

**Rule 3**

\[ wy[i]x \]
Some properties

- leaves are added in increasing order;
- rule 1 does not need any treatment;
- phase $i$ begins at the extension $j_\ell + 1$, where $j_\ell$ is the number of the last created leaf;
- phase $i$ ends at the first extension $j > j_\ell$ such that rule 3 is applied.
Introduction
- Motivation
- Suffix trees
- Ukkonen’s algorithm

Suffix Vectors
- Introduction
- Compact Suffix Vectors
- On-line construction of a compact suffix vector

Computing maximal repeats

Conclusion
Introduction to suffix vectors

Root

(0, 2) − (1, 1) − (4, 1)

0 1 2 3 4
t a t a $

2 3 (4,1)
1 3 (4,1)
Introduction to suffix vectors
Introduction to suffix vectors

- Alternative data structure to suffix trees
- same information in reduced space
- introduced by K. Monostori in 2001
Introduction to suffix vectors

Definition
A succession of boxes whose lines contain:
- the depth of the node;
- the natural edge;
- the edge list.

The root is a special box.

Notations
- $B_j$: box at position $j$ in $y$,
- The natural edge of a line in $B_j$ is the end position of the edge beginning by $y[j + 1]$. 
**Example**

$tatt$ is a substring of $y$?

The root contains the edge $(2, 1)$ beginning by $t$ leading to $B_2$. The edge $(5, 1)$ by $a$ leads to $B_5$. The natural edge begins by $tt$. 
**Definition**

A *group of nodes* is a set of nodes which are in the same box and have exactly the same edges.
Compact suffix vectors

3 rules of compaction of a box:

**Rule A** the node with depth $d - 2$ has the same edges as the node with depth $d - 1$,

**Rule B** the node with depth $d - 1$ has the same edges as the node with depth $d$ and some extra edges,

**Rule C** the node with depth $d - 3$ has different edges to the node with depth $d - 2$.
Compacting $\nu(aatttatattattata$)

Root

(0, 1) – (2, 1) – (13, 1)
\( y \xrightarrow{\text{Monostori}} O(n) \xrightarrow{\text{Extended vector}} O(n) \xrightarrow{\text{Monostori}} \text{Compact vector} \)
On-line construction of a compact vector

\[ y \xrightarrow{O(n)} \text{extended vector} \xrightarrow{O(n)} \text{compact vector} \]

Faster and more space economical construction.

\[ O(n) \]

Elise Prieur

Faster and more space economical construction.
**Proposition**

When an edge is added to the node $w$ of depth $d$ in a box $B_p$, this edge will be added to all the nodes in $B_p$ of depth smaller than $d$ in the group of nodes of $w$. 
Skip $k - 1$ extensions where $k$ is the number of the nodes in the group into the edge is added.
1 Introduction
   • Motivation
   • Suffix trees
   • Ukkonen’s algorithm

2 Suffix Vectors
   • Introduction
   • Compact Suffix Vectors
   • On-line construction of a compact suffix vector

3 Computing maximal repeats

4 Conclusion
Definition

A maximal repeat in a string is a substring such that there exist at least 2 occurrences: \( a_1ub_1 \) and \( a_2ub_2 \) with \( a_1 \neq a_2, \ b_1 \neq b_2 \) and \( a_1, a_2, b_1, b_2 \in A \).

Example

\( y = \text{aatttatttattta}$

\( \text{tta} \) is a maximal repeat at positions 5 and 12.
Applying to suffix vectors

**Proposition**

The deepest node of each group of nodes represents a maximal repeat.
Example

Boxes 0, 2, 5 et 7 are reduced: a, t, tta, atttatt are maximal repeats.
Box $B_3$ is extended, the 2 lines have different edges: att, tt are maximal repeats.
1 Introduction
   - Motivation
   - Suffix trees
   - Ukkonen’s algorithm

2 Suffix Vectors
   - Introduction
   - Compact Suffix Vectors
   - On-line construction of a compact suffix vector

3 Computing maximal repeats

4 Conclusion
More economical construction of the compact suffix vector.

Linear method to compute maximal repeats with a compact suffix vector.