Computing All Approximate Enhanced Covers with the Hamming Distance

Ondřej Guth
ondrej.guth@fit.cvut.cz

Department of Theoretical Computer Science
Faculty of Information Technology
Czech Technical University in Prague

1. Introduction
    - About the Topics
2. Preliminaries
    - The Notions
    - Automata
3. The Algorithm
    - Basic Idea
    - Time and Space Complexity
    - Details
4. Conclusions
    - Summary
    - Future Work

Background

- repetitive structures of strings: periods, squares, covers, seeds, etc.

Background

- repetitive structures of strings: periods, squares, covers, seeds, etc.
    - compact description of a string

Background

- repetitive structures of strings: periods, squares, covers, seeds, etc.
    - compact description of a string
- quite restrictive

Background

- repetitive structures of strings: periods, squares, covers, seeds, etc.
    - compact description of a string
- quite restrictive
- attempts to introduce more relaxed regularities

Background

- repetitive structures of strings: periods, squares, covers, seeds, etc.
  - compact description of a string
- quite restrictive
- attempts to introduce more relaxed regularities
- enhanced covers, approximate enhanced covers

Example

$\boldsymbol{x} = \texttt{aabaaccaabaa}$

Definition

A k-approximate enhanced cover $\boldsymbol{w}$ of $\boldsymbol{x}$ is

Example

$\boldsymbol{x} = \texttt{aabaaccaabaa}$
$\text{bord} = \{\texttt{a}, \texttt{aa}, \texttt{aabaa}\}$

Definition

A k-approximate enhanced cover $\boldsymbol{w}$ of $\boldsymbol{x}$ is

1. a border of $\boldsymbol{x}$

Example

$\boldsymbol{x} = \texttt{aabaaccaabaa}$ $\quad k = 1$

$\text{bord} = \{\cancel{\texttt{a}}, \texttt{aa}, \texttt{aabaa}\}$

	a	aa	aabaa
covered	N/A	12	10

Definition

A k-approximate enhanced cover $\boldsymbol{w}$ of $\boldsymbol{x}$ is

1. a border of $\boldsymbol{x}$
2. number of positions of $\boldsymbol{x}$ that lie within some k-approximate occurrence of $\boldsymbol{w}$ in $\boldsymbol{x}$ under Hamming distance is the maximum among all borders of $\boldsymbol{x}$

Example

$\boldsymbol{x} = \texttt{aabaaccaabaa}$ $k = 1$

bord $= \{\cancel{\texttt{a}}, \texttt{aa}, \texttt{aabaa}\}$

	a	aa	aabaa
covered	N/A	12	10

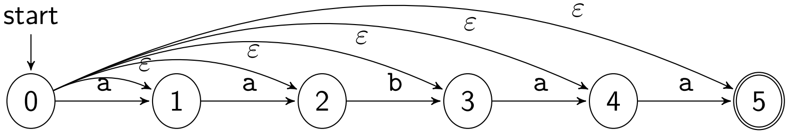
Definition

A k-approximate enhanced cover $\boldsymbol{w}$ of $\boldsymbol{x}$ is

1. a border of $\boldsymbol{x}$
2. number of positions of $\boldsymbol{x}$ that lie within some k-approximate occurrence of $\boldsymbol{w}$ in $\boldsymbol{x}$ under Hamming distance is the maximum among all borders of $\boldsymbol{x}$

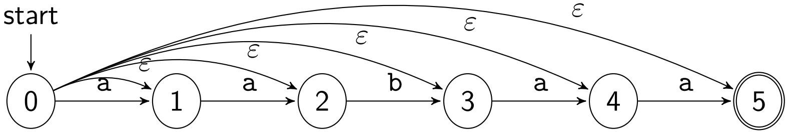
Suffix automaton

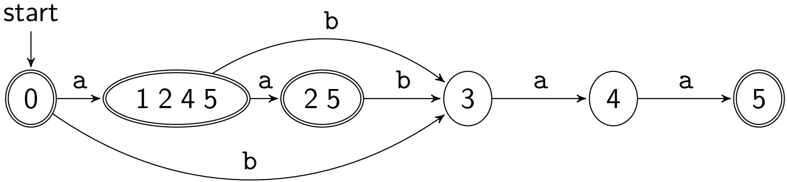
Nondeterministic



Suffix automaton

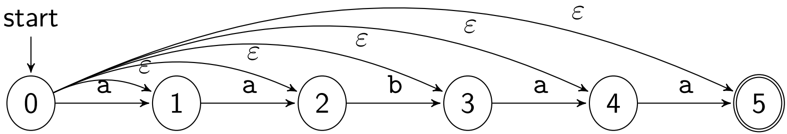
Deterministic





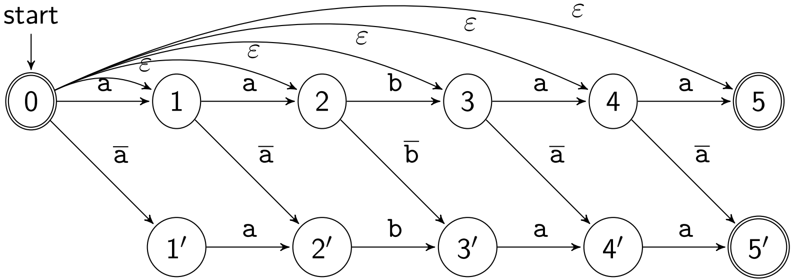
Suffix automaton

Nondeterministic



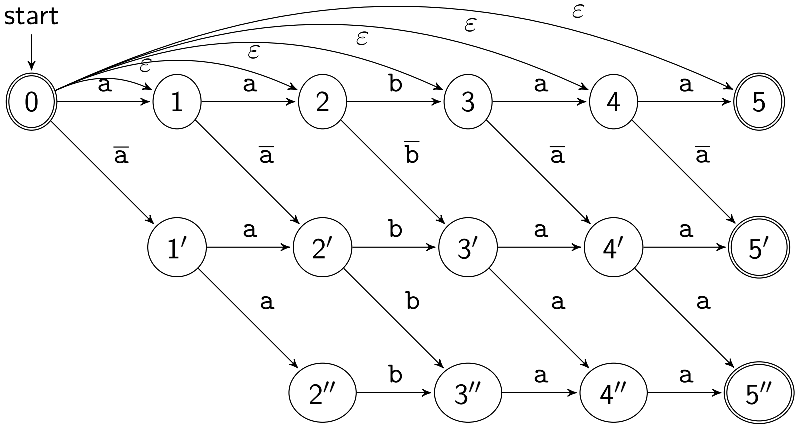
Suffix automaton

Nondeterministic Hamming $k = 1$



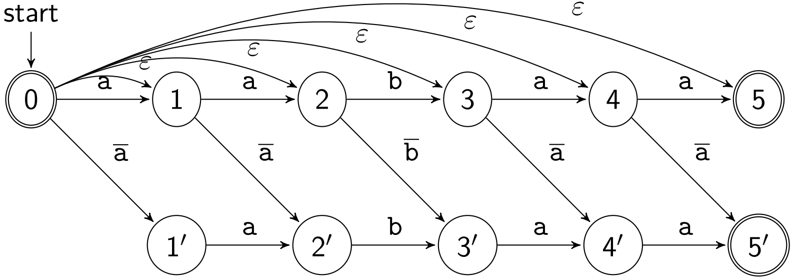
Suffix automaton

Nondeterministic Hamming $k = 2$



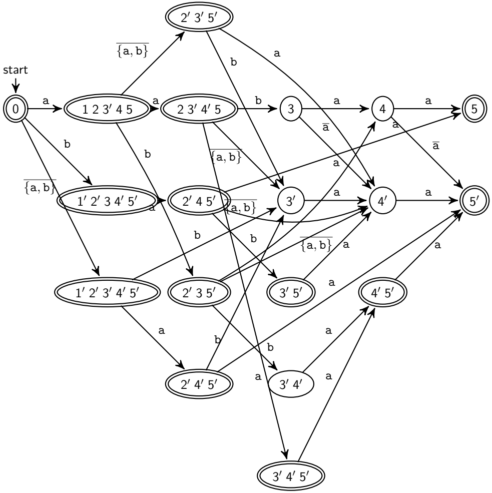
Suffix automaton

Nondeterministic Hamming $k = 1$



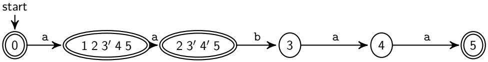
Suffix automaton

Deterministic, Hamming $k = 1$



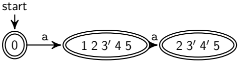
Suffix automaton

Backbone, Hamming $k = 1$



Suffix automaton

Backbone multiple front end, Hamming $k = 1$



Outline of the algorithm

1. Construct nondeterministic k-approximate suffix automaton

Outline of the algorithm

1. Construct nondeterministic k-approximate suffix automaton
2. Construct backbone using subset construction

Outline of the algorithm

1. Construct nondeterministic k-approximate suffix automaton
2. Construct backbone using subset construction
3. For each appropriate state, count covered positions and remember

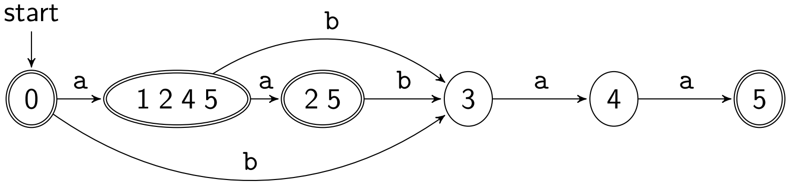
- Space complexity: $\mathcal{O}(n)$
- Time complexity: $\mathcal{O}(n^2)$

Construction of the backbone

Backbone only

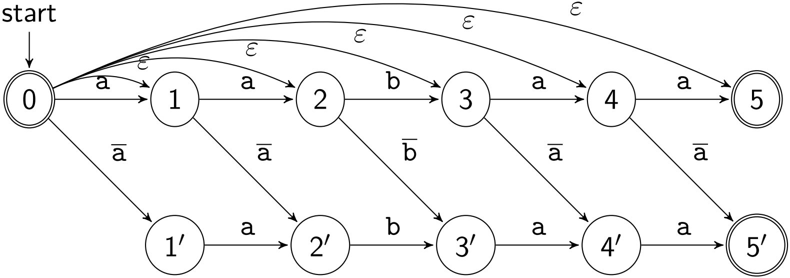
Just backbone is constructed, nothing else is needed

Example



Construction of the backbone

Nondeterministic automaton is not needed



Construction of the backbone

Nondeterministic automaton is not needed

Construction of the first backbone state:

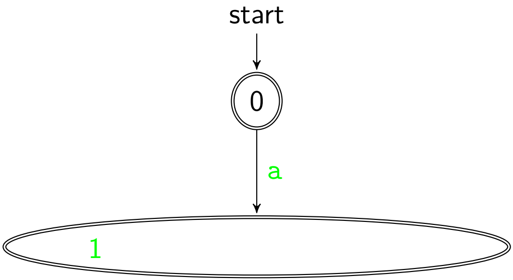
```
$q_1$ is a state;
for $i \in 1..|\mathbf{x}|$ do
    $e$ is a d-subset element such that depth($e$) $\leftarrow i$;
    if $\mathbf{x}[1] = \mathbf{x}[i]$ then
        level($e$) $\leftarrow 0$;
        append $e$ to $q_1$
    else if $k > 0$ then
        level($e$) $\leftarrow 1$;
        append $e$ to $q_1$
    end
end
```

Construction of the backbone

Nondeterministic automaton is not needed

Example

	1	2	3	4	5	6	7	8	9	10	11	12
$\boldsymbol{x} =$	a	a	b	a	a	c	c	a	a	b	a	a

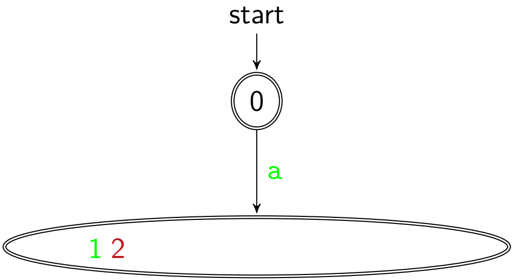


Construction of the backbone

Nondeterministic automaton is not needed

Example

$$\begin{array}{ccccccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ \boldsymbol{x} = & \mathsf{a} & \mathsf{a} & \mathsf{b} & \mathsf{a} & \mathsf{a} & \mathsf{c} & \mathsf{c} & \mathsf{a} & \mathsf{a} & \mathsf{b} & \mathsf{a} & \mathsf{a} \end{array}$$

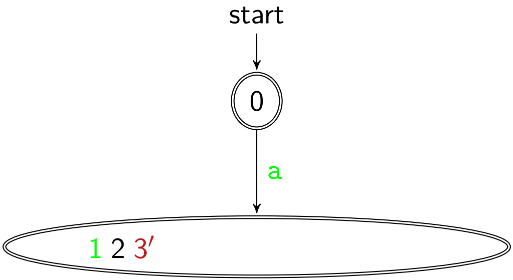


Construction of the backbone

Nondeterministic automaton is not needed

Example

	1	2	3	4	5	6	7	8	9	10	11	12
$\boldsymbol{x} =$	a	a	b	a	a	c	c	a	a	b	a	a

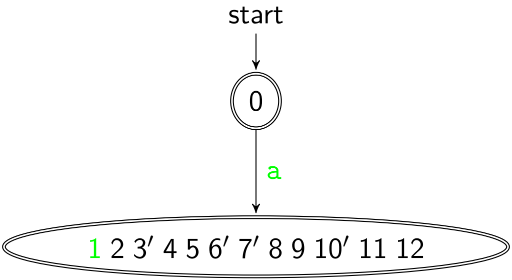


Construction of the backbone

Nondeterministic automaton is not needed

Example

	1	2	3	4	5	6	7	8	9	10	11	12
$\boldsymbol{x} =$	a	a	b	a	a	c	c	a	a	b	a	a



Construction of the backbone

Nondeterministic automaton is not needed

Construction of a next backbone state:

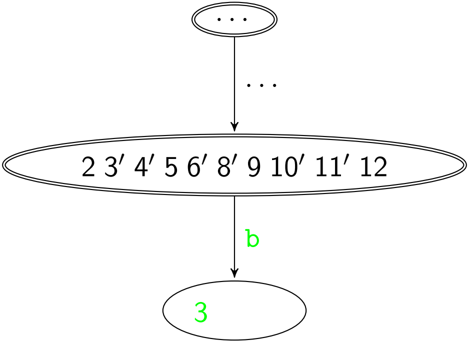
```
for $i \in 2..|\boldsymbol{x}|$ do
    for $e_p \in q_p$ do
        if $\mathrm{depth}(e_p) < |\boldsymbol{x}|$ then
            $e_n$ is new d-subset element;
            $\mathrm{depth}(e_n) \leftarrow \mathrm{depth}(e_p) + 1$;
            if $\boldsymbol{x}[i] = \boldsymbol{x}[\mathrm{depth}(e_n)]$ then
                $\mathrm{level}(e_n) \leftarrow \mathrm{level}(e_p)$;
                append $e_n$ to $q_n$;
            else if $\mathrm{level}(e_p) < k$ then
                $\mathrm{level}(e_n) \leftarrow \mathrm{level}(e_p) + 1$;
                append $e_n$ to $q_n$;
            end
        end
    end
end
```

Construction of the backbone

Nondeterministic automaton is not needed

Example

	1	2	3	4	5	6	7	8	9	10	11	12
$\boldsymbol{x} =$	a	a	b	a	a	c	c	a	a	b	a	a

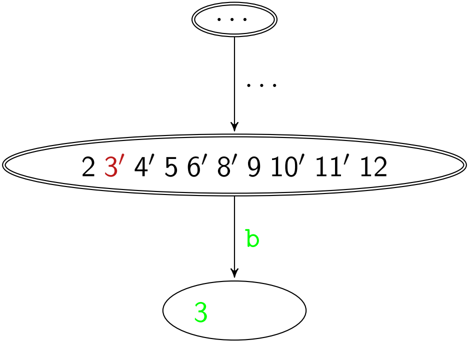


Construction of the backbone

Nondeterministic automaton is not needed

Example

	1	2	3	4	5	6	7	8	9	10	11	12
$\boldsymbol{x} =$	a	a	b	a	a	c	c	a	a	b	a	a

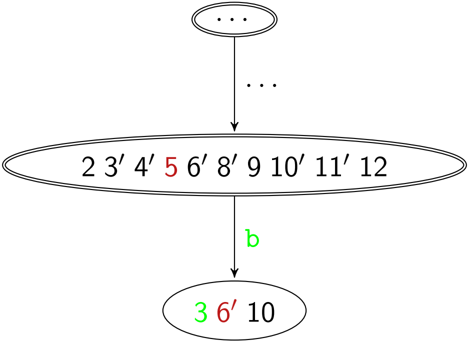


Construction of the backbone

Nondeterministic automaton is not needed

Example

	1	2	3	4	5	6	7	8	9	10	11	12
$\boldsymbol{x} =$	a	a	b	a	a	c	c	a	a	b	a	a

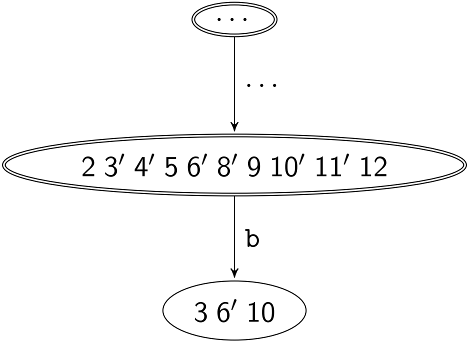


Construction of the backbone

Removal of states

Example

	1	2	3	4	5	6	7	8	9	10	11	12
$\boldsymbol{x} =$	a	a	b	a	a	c	c	a	a	b	a	a



- new type of string regularity under Hamming distance

- new type of string regularity under Hamming distance
- algorithm to find all k-approximate enhanced covers of a given string

- new type of string regularity under Hamming distance
- algorithm to find all *k*-approximate enhanced covers of a given string
    - based on finite automata

- new type of string regularity under Hamming distance
- algorithm to find all k-approximate enhanced covers of a given string
    - based on finite automata
    - $\mathcal{O}(n)$ space and $\mathcal{O}(n^2)$ time

- reduce time complexity

- reduce time complexity
- variants of the problem

- reduce time complexity
- variants of the problem
- *k*-approximate enhanced cover array