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

\[ x = \text{aabaaccaabaa} \]

Definition

A \( k \)-approximate enhanced cover \( w \) of \( x \) is
Example

\[ x = \text{aabaaccaabaa} \]

\[ \text{bord} = \{a, aa, aabaa\} \]

Definition

A \( k \)-approximate enhanced cover \( w \) of \( x \) is

1. a border of \( x \)
Example

\[ x = \text{aabaaccaabaa} \quad k = 1 \]

bord = \{a, aa, aabaa\}

\[
\begin{array}{|c|c|c|c|}
\hline
\text{covered} & \text{N/A} & 12 & 10 \\
\hline
\end{array}
\]

Definition

A \( k \)-approximate enhanced cover \( w \) of \( x \) is

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

\[ x = \text{aabaaccaabaa} \quad k = 1 \]

\[ \text{bord} = \{a, aa, aabaa\} \]

\begin{center}
\begin{tabular}{|c|c|c|c|}
\hline
\text{covered} & N/A & 12 & 10 \\
\hline
\end{tabular}
\end{center}

Definition

A \( k \)-approximate enhanced cover \( w \) of \( x \) is:

1. a border of \( x \)

2. number of positions of \( x \) that lie within some \( k \)-approximate occurrence of \( w \) in \( x \) under Hamming distance is the maximum among all borders of \( x \)
Suffix automaton

Nondeterministic
Suffix automaton

Deterministic

---

O. Guth (Czech Tech Univ) Approximate Enhanced Covers 5 / 17
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;
\[
\text{for } i \in 1..|x| \text{ do}
\]
\[
\text{e is a d-subset element such that } \text{depth}(e) \leftarrow i;
\]
\[
\text{if } x[1] = x[i] \text{ then}
\]
\[
\text{level}(e) \leftarrow 0;
\]
\[
\text{append } e \text{ to } q_1
\]
\[
\text{else if } k > 0 \text{ then}
\]
\[
\text{level}(e) \leftarrow 1;
\]
\[
\text{append } e \text{ to } q_1
\]
\[
\text{end}
\]
\[
\text{end}
\]
Construction of the backbone
Nondeterministic automaton is not needed

Example

\[ x = a \quad a \quad b \quad a \quad a \quad c \quad c \quad a \quad a \quad b \quad a \quad a \quad a \]
Construction of the backbone
Nondeterministic automaton is not needed

Example

\[\begin{array}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\times = & a & a & b & a & a & c & c & a & a & b & a & a \\
\end{array}\]
Construction of the backbone
Nondeterministic automaton is not needed

Example

\[
\begin{array}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\times = & a & a & b & a & a & c & c & a & a & b & a & a \\
\end{array}
\]
Construction of the backbone

Nondeterministic automaton is not needed

Example

\[ x = a \ a \ b \ a \ a \ c \ c \ a \ a \ b \ a \ a \]

\[
\begin{array}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\end{array}
\]
Construction of the backbone
Non-deterministic automaton is not needed

Construction of a next backbone state:

for \( i \in 2..|x| \) do
  for \( e_p \in q_p \) do
    if depth\((e_p)\) < \(|x|\) then
      \( e_n \) is new \( d \)-subset element;
      depth\((e_n)\) ← depth\((e_p)\) + 1;
      if \( x[i] = x[\text{depth}(e_n)] \) then
        level\((e_n)\) ← level\((e_p)\);
        append \( e_n \) to \( q_n \);
      else if level\((e_p)\) < \( k \) then
        level\((e_n)\) ← level\((e_p)\) + 1;
        append \( e_n \) to \( q_n \);
    end
  end
end
Construction of the backbone

Nondeterministic automaton is not needed

Example

\[ x = \text{a a b a a c c a a b a a} \]
Construction of the backbone
Nondeterministic automaton is not needed

Example

\[ x = a \ a \ b \ a \ a \ c \ c \ a \ a \ b \ a \ a \]

\[
\begin{array}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\end{array}
\]
Construction of the backbone
Nondeterministic automaton is not needed

Example

\[
x = \begin{array}{ccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\end{array}
\]

\[
\begin{array}{ccccccccccccc}
\x = & a & a & b & a & a & a & c & c & a & a & b & a & a \\
\end{array}
\]
Construction of the backbone
Removal of states

Example

\[ x = a \ a \ b \ a \ a \ c \ c \ a \ a \ b \ a \ a \]

\[
\begin{array}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\end{array}
\]
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
  - $O(n)$ space and $O(n^2)$ time
reduce time complexity
- reduce time complexity
- variants of the problem
• reduce time complexity
• variants of the problem
• $k$-approximate enhanced cover array