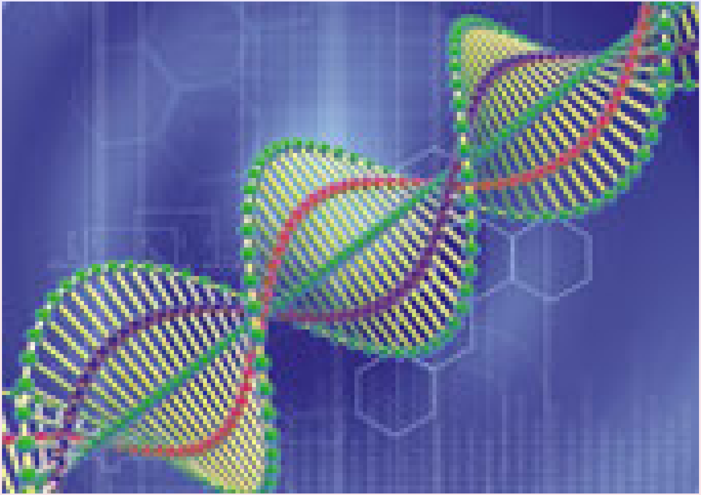
Computing All Pure Squares In Compressed Texts

Lesha Khvorost

Ural State University

July 14, 2011

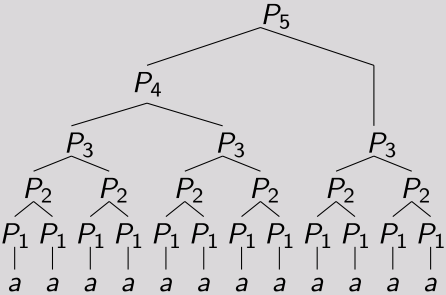
Decoding DNA and Feature Extraction



Searching a Criminal by His Photo

SLP that Generates a^{12}

Example


P_5
P_4
P_3
P_3
P_3
P_2 P_2 P_2 P_2 P_2 P_2
P_1 P_1 P_1 P_1 P_1 P_1 P_1 P_1 P_1 P_1 P_1 P_1
a a a a a a a a a a a a

Simple Definitions

Primitive string

A string x is **primitive** if $x = u^k$ for some k implies that $k = 1$ and $u = x$.

Pure square and repetition

A **pure** square is a square xx where x is primitive. Otherwise, xx is called a **repetition**.

Main Problem

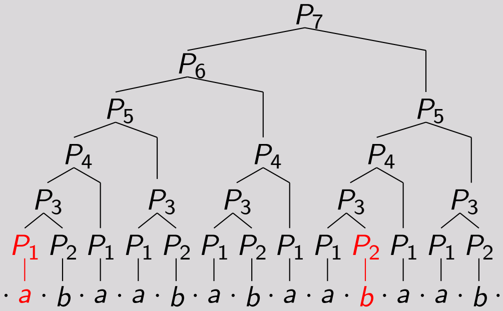
Computing All Pure Squares Problem

Input: an SLP S that derives some text S; Output: a data

structure (a PS-table) that contains information about all pure squares in S in a compressed form;

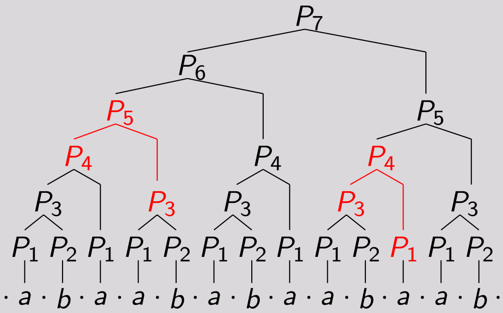
SLPs Background

Example


P_7
P_6
P_5
P_5
P_4
P_4
P_4
P_3
P_3
P_3
P_3
P_3
P_1 P_2 P_1 P_1 P_2 P_1 P_2 P_1 P_1 P_2 P_1 P_1 P_2
$\cdot a \cdot b \cdot a \cdot a \cdot b \cdot a \cdot b \cdot a \cdot a \cdot b \cdot a \cdot a \cdot b \cdot$

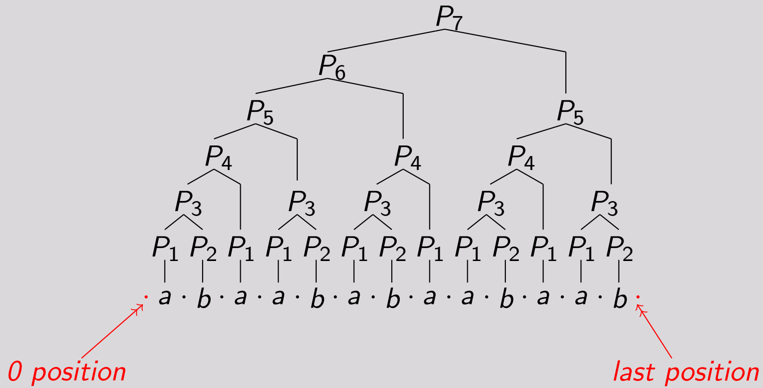
SLPs Background

Example


P_7
P_6 P_5
P_5 P_4 P_4
P_4 P_3 P_3 P_3 P_3
P_3 P_1
P_1 P_2 P_1 P_1 P_2 P_1 P_2 P_1 P_1 P_2 P_1 P_1 P_2
$\cdot\, a \cdot b \cdot a \cdot a \cdot b \cdot a \cdot b \cdot a \cdot a \cdot b \cdot a \cdot a \cdot b \,\cdot$

SLPs Background

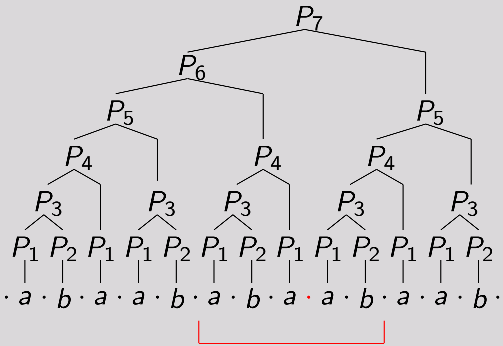
Example



P_7
P_6
P_5
P_5
P_4
P_4
P_4
P_3
P_3
P_3
P_3
P_3
P_1 P_2 P_1 P_1 P_2 P_1 P_2 P_1 P_1 P_2 P_1 P_1 P_2
$\cdot a \cdot b \cdot a \cdot a \cdot b \cdot a \cdot b \cdot a \cdot a \cdot b \cdot a \cdot a \cdot b \cdot$
0 position
last position

SLPs Background

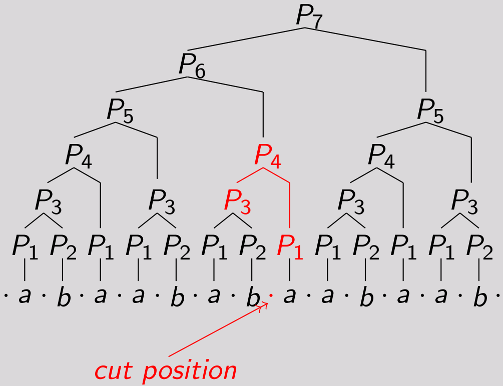
Example



P_7
P_6
P_5
P_5
P_4
P_4
P_4
P_3
P_3
P_3
P_3
P_3
P_1 P_2 P_1 P_1 P_2 P_1 P_2 P_1 P_1 P_2 P_1 P_1 P_2
$\cdot\, a \cdot b \cdot a \cdot a \cdot b \cdot a \cdot b \cdot a \cdot a \cdot b \cdot a \cdot a \cdot b \,\cdot$

SLPs Background

Example


P_7
P_6
P_5
P_4
P_5
P_4
P_3
P_3
P_3
P_4
P_3
P_3
P_1 P_2 P_1 P_1 P_2 P_1 P_2 P_1 P_1 P_2 P_1 P_1 P_2
$\cdot a \cdot b \cdot a \cdot a \cdot b \cdot a \cdot b \cdot a \cdot a \cdot b \cdot a \cdot a \cdot b \cdot$
cut position

Pure Squares Table

PS-table Properties

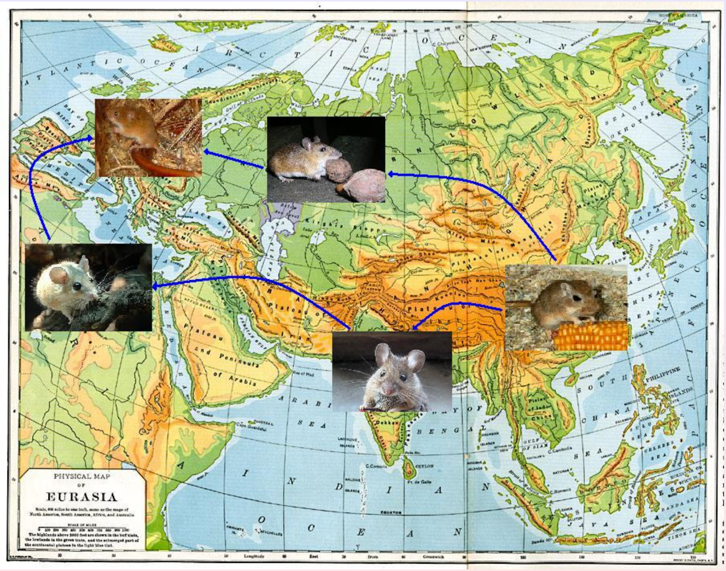
- The size of *PS*-table is equal to $(\lfloor \log |S| \rfloor + 1) \times (|\mathbb{S}| + 1)$;
- The cell $PS(i,j)$ with $i,j > 0$ contains information about the family of all pure squares such that
  1. they touch the cut position of the rule $\mathbb{S}_j$;
  2. they are contained in the text S_j;
  3. lengths of their roots belong to the interval $[2^{i-1}, 2^i)$.

Summarize

Features of the algorithm

1. The algorithm runs on $O(\max(|\mathbb{S}|^5 \log |S|, |\mathbb{S}|^3 \log^3 |\mathbb{S}| \log |S|))$ time and requires $O(\mathbb{S}^3)$ space;
2. The algorithm is divided into independent steps in contrast to classical algorithms in this area which consecutively accumulate information about required objects. As a result it can be parallelized;
3. The algorithm is not excluded that the constants hidden in the "O" notation are actually very big;

Genesis and Migration of Mice Problem



Open Problems

Computing All Squares Problem

Can we compress all other families of repetitions?

Optimization

Can we optimize a time complexity of the algorithm?

Haven't you slept yet?