Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Journal of Discrete Algorithms 5 (2007) 243–249
JOURNAL OF DISCRETE ALGORITHMS
www.elsevier.com/locate/jda

Linear time algorithm for the longest common repeat problem

Inbok Lee [a,b,1], Costas S. Iliopoulos [b], Kunsoo Park [a,*]

[a] *School of Computer Science & Engineering, Seoul National University, Seoul, Republic of Korea*
[b] *Department of Computer Science, King's College London, London, United Kingdom*

Available online 16 May 2006

Abstract

Given a set of strings $U = \{T_1, T_2, \ldots, T_\ell\}$, the longest common repeat problem is to find the longest common substring that appears at least twice in each string of U. We also consider reversed and reverse-complemented repeats as well as normal repeats. We present a linear time algorithm for the longest common repeat problem.

Keywords: String algorithm; Repeat; Generalized suffix tree

1. Introduction

Repetitive or periodic strings have a great importance in a variety of applications including computational molecular biology, data mining, data compression, and computer-assisted music analysis. For example, it is assumed that repetitive substrings in a biological sequence have important meanings and functions [1]. Finding common substrings in a set of strings is also important. For example, motifs or short strings common to protein sequences are assumed to represent a specific property of the sequences [3].

In this paper we want to find common repetitive substrings in a set of strings. We especially focus on finding the longest common repeat in a set since the number of the common repeats in a set can be quite large. We also consider reversed and reverse-complemented strings in finding repeats. Formally we define our problem as follows.

Let T be a string over an alphabet Σ. We assume $\Sigma = \{A, C, G, T\}$ or $\Sigma = \{A, C, G, U\}$ since a major application of the problem is computational molecular biology. $T[i]$ denotes the ith character of T. $T[i..j]$ is the substring $T[i]T[i+1]\cdots T[j]$ of T. The *left character* of a suffix $T[i..|T|]$ means $T[i-1]$. T^R denotes the *reverse string* of T where $|T^R| = |T|$ and $T^R[i] = T[|T| - i + 1]$ for $1 \leqslant i \leqslant |T|$. T^{RC} denotes the *reverse-complemented string* of T where $|T^{RC}| = |T|$ and, $T^{RC}[i]$ and $T[|T| - i + 1]$ form a Watson–Crick pair ($A \equiv (T$ or $U)$ and $C \equiv G$) for $1 \leqslant i \leqslant |T|$.

A *repeat* of T is a substring of T which appears at least twice in T. There are three kinds of repeats.

[*] Corresponding author.
E-mail addresses: iblee@theory.snu.ac.kr (I. Lee), csi@dcs.kcl.ac.uk (C.S. Iliopoulos), kpark@theory.snu.ac.kr (K. Park).

[1] This work was supported by the Korea Research Foundation Grant funded by Korean Government (MOEHRD, Basic Research Promotion Fund) M01-2004-000-20344-0.

1570-8667/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.jda.2006.03.019

- Normal repeat: A string p is called a *normal repeat* of T if $p = T[i..i + |p| - 1]$ and $p = T[i'..i' + |p| - 1]$ for $i \neq i'$.
- Reversed repeat: A string p is called a *reversed repeat* of T if $p = T[i..i + |p| - 1]$ and $p^R = T[i'..i' + |p| - 1]$.
- Reverse-complemented repeat: A string p is called a *reverse-complemented repeat* if $p = T[i..i + |p| - 1]$ and $p^{RC} = T[i'..i' + |p| - 1]$.

There are two reasons why we consider reversed and reverse-complemented repeats: (i) We don't know the directions of the strings in advance. (ii) In some situations, reversed and reverse-complemented repeats play an important role. For example, RNA secondary structures are determined by reverse-complemented repeats.

The longest common repeat problem can be defined as follows.

Problem 1. Given a set of strings $U = \{T_1, T_2, \ldots, T_\ell\}$, the (k, ℓ) longest common repeat problem is to find the longest repeat (normal, reversed or reverse-complemented) which are common to k strings in U for $1 \leqslant k \leqslant \ell$.

For finding the longest normal repeat in a text T, Karp, Miller, and Rosenberg first proposed $O(|T| \log |T|)$ time algorithm [8]. However, it is an easy application of the suffix tree [4,10,13] to find it in $O(|T|)$ time.

For approximate normal repeats, Landau and Schmidt gave an $O(k|T| \log k \log |T|)$ time algorithm for finding approximate squares where the allowed edit distance is at most k [9]. Schmidt also gave an $O(|T|^2 \log |T|)$ time algorithm for finding approximate tandem or non-tandem repeats [12].

The longest common repeat problem resembles the longest common substring problem. The difference is that the common substring should appear at least twice in each sequence in the longest common repeat problem. For the longest common substring problem with a set of strings $\{T_1, T_2, \ldots, T_\ell\}$, Hui showed an $O(\sum_{i=1}^{\ell} |T_i|)$ time algorithm [7]. As far as we know, our algorithm is the first one that solves the longest common repeat problem.

2. Preliminaries

A *generalized suffix tree* stores all the suffixes of a set of strings just as a suffix tree stores all the suffixes of a string. It is easy to extend the suffix tree construction algorithm [13] to building a generalized suffix tree [5, p. 116]. Fig. 1 is an example of the generalized suffix tree for $T_1 = AACTG$ and $T_2 = ACTGCTG$. We use a special character \$ which is not in Σ to denote the end of a string. Each leaf node has an ID representing the original string where the suffix came. Identical suffixes of two or more strings are considered as different ones. In this example, T_1 and T_2 share three identical suffixes CTG, TG, and G. Each of these suffixes has two leaves with different IDs.

From now on, let $ST(T)$ denote the suffix tree of T and $GST(T_1..T_\ell)$ denote the generalized suffix tree of $T_1, T_2, \ldots, T_\ell$. Let $L(v)$ denote the string obtained by concatenating the edge labels on the path from the root to a node v in a suffix tree or a generalized suffix tree.

We define *corresponding nodes* between $ST(T_i)$ and $GST(T_1..T_\ell)$ $(1 \leqslant i \leqslant \ell)$.

Definition 1. The *corresponding node* of an internal node v in $ST(T_i)$ $(1 \leqslant i \leqslant \ell)$ is a node v' in $GST(T_1..T_\ell)$ such that $L(v) = L(v')$.

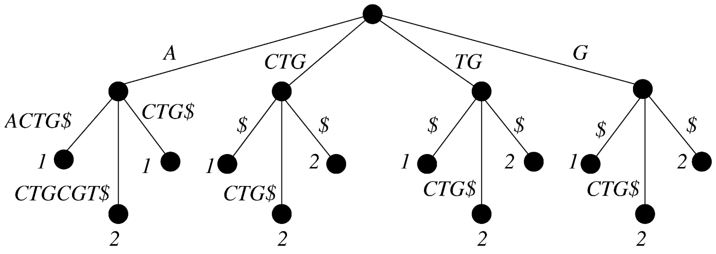

Fig. 1. The generalized suffix tree for $T_1 = AACTG$ and $T_2 = ACTGCTG$.