Weighted Ancestors in Suffix Trees Revisited

Djamal Belazzougui
Centre de Recherche sur l'Information Scientifique et Technique, Algiers, Algeria

Dmitry Kosolobov
Institute of Natural Sciences and Mathematics, Ural Federal University, Ekaterinburg, Russia

Simon J. Puglisi
Department of Computer Science, University of Helsinki, Helsinki, Finland

Rajeev Raman
Department of Informatics, University of Leicester, Leicester, United Kingdom

Abstract

The weighted ancestor problem is a well-known generalization of the predecessor problem to trees. It is known to require $\Omega(\log\log n)$ time for queries provided $\mathcal{O}(n \operatorname{polylog} n)$ space is available and weights are from $[0..n]$, where n is the number of tree nodes. However, when applied to suffix trees, the problem, surprisingly, admits an $\mathcal{O}(n)$-space solution with constant query time, as was shown by Gawrychowski, Lewenstein, and Nicholson (Proc. ESA 2014). This variant of the problem can be reformulated as follows: given the suffix tree of a string s, we need a data structure that can locate in the tree any substring $s[p..q]$ of s in $\mathcal{O}(1)$ time (as if one descended from the root reading $s[p..q]$ along the way). Unfortunately, the data structure of Gawrychowski et al. has no efficient construction algorithm, limiting its wider usage as an algorithmic tool. In this paper we resolve this issue, describing a data structure for weighted ancestors in suffix trees with constant query time and a linear construction algorithm. Our solution is based on a novel approach using so-called irreducible LCP values.

2012 ACM Subject Classification Theory of computation → Pattern matching

Keywords and phrases suffix tree, weighted ancestors, irreducible LCP, deterministic substring hashing

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Funding *Dmitry Kosolobov*: Supported by the grant 20-71-00050 of the Russian Foundation of Basic Research (for the final version of the data structure and for its construction algorithm).

1 Introduction

The suffix tree is one of the central data structures in stringology. Its primary application is to determine whether an arbitrary string t occurs as a substring of another string, s, which can be done in time proportional to the length of t by traversing the suffix tree of s downward from the root and reading off the symbols of t along the way. Many algorithms using suffix trees perform this procedure for substrings $t = s[p..q]$ of s itself. In this important special case, the traversal can be executed much faster than $\mathcal{O}(q - p + 1)$ time provided the tree has been preprocessed to build some additional data structures [1, 11, 13]; particularly surprising is that the traversal can be performed in constant time using only linear space, as shown by Gawrychowski et al. [13]. In this paper, we describe the first linear construction algorithm for such a data structure. The lack of an efficient construction algorithm for the result of [13] has been, apparently, the main obstacle hindering its wider adoption. Our solution is completely different from that of [13] and is based on a combinatorial result of Kärkkäinen et al. [17] (see also [16]) that estimates the sum of irreducible LCP values (precise definitions follow).

As one might expect, the described data structure has a multitude of applications: [1, 7, 8, 11, 19, 20, 21], to name a few. We, however, do not dive further into a discussion of these