Discrete Applied Mathematics 205 (2016) 35–44

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

On the readability of overlap digraphs[☆]

Rayan Chikhi [a,b], Paul Medvedev [a,d,e,*], Martin Milanič [c], Sofya Raskhodnikova [a]

[a] Department of Computer Science, The Pennsylvania State University, USA
[b] CNRS, UMR 9189, France
[c] UP IAM and UP FAMNIT, University of Primorska, Slovenia
[d] Department of Biochemistry and Molecular Biology, The Pennsylvania State University, USA
[e] Genome Sciences Institute of the Huck, The Pennsylvania State University, USA

ARTICLE INFO

Article history:
Received 25 July 2015
Received in revised form 30 November 2015
Accepted 9 December 2015
Available online 5 January 2016

Keywords:
Graph parameter
Stringology
Bioinformatics
Readability
Overlap graph

ABSTRACT

We introduce the graph parameter *readability* and study it as a function of the number of vertices in a graph. Given a digraph D, an injective overlap labeling assigns a unique string to each vertex such that there is an arc from x to y if and only if x properly overlaps y. The readability of D is the minimum string length for which an injective overlap labeling exists. In applications that utilize overlap digraphs (e.g., in bioinformatics), readability reflects the length of the strings from which the overlap digraph is constructed. We study the asymptotic behavior of readability by casting it in purely graph theoretic terms (without any reference to strings). We prove upper and lower bounds on readability for certain graph families and general graphs.

1. Introduction

In this paper, we introduce and study a graph parameter called readability, motivated by applications of overlap graphs in bioinformatics. A string x *overlaps* a string y if there is a non-empty suffix of x that is equal to a prefix of y. They overlap *properly* if, in addition, the suffix and prefix are both proper. The *overlap digraph* of a set of strings S is the digraph where each string is a vertex and there is an arc from x to y (possibly with $x = y$) if and only if x properly overlaps y. Walks in the overlap digraph of S represent strings that can be spelled by stitching strings of S together, using the overlaps between them. Overlap digraphs have various applications, e.g., they are used by approximation algorithms for the Shortest Superstring Problem [16]. Their most impactful application, however, has been in bioinformatics. Their variants, such as de Bruijn graphs [7] and string graphs [12], have formed the basis of nearly all genome assemblers used today (see [11,13] for a survey), successful despite results showing that assembly is a hard problem in theory [5,10,14]. In this context, the strings of S represent known fragments of the genome (called *reads*), and the genome is represented by walks in the overlap digraph of S. However, do the overlap digraphs generated in this way capture all possible digraphs, or do they have any properties or structure that can be exploited?

Braga and Meidanis [4] showed that overlap digraphs capture all possible digraphs, i.e., for every digraph D, there exists a set of strings S such that their overlap digraph is D. Their proof takes an arbitrary digraph and shows how to construct an *injective overlap labeling*, that is, a function assigning a unique string to each vertex, such that (x, y) is an arc if and only if

[☆] This is a full version of a conference paper of the same title at the 26th Annual Symposium on Combinatorial Pattern Matching (CPM 2015).

[*] Correspondence to: 343J IST Bldg, University Park, PA 16801, USA.
E-mail address: paul.medvedev@psu.edu (P. Medvedev).

http://dx.doi.org/10.1016/j.dam.2015.12.009
0166-218X/© 2015 Elsevier B.V. All rights reserved.

the string assigned to x properly overlaps the string assigned to y. However, the *length* of strings produced by their method can be exponential in the number of vertices. In the bioinformatics context, this is unrealistic, as the read size is typically much smaller than the number of reads.

To investigate the relationship between the string length and the number of vertices, we introduce a graph parameter called *readability*. The readability of a digraph D, denoted $r(D)$, is the smallest nonnegative integer r such that there exists an injective overlap labeling of D with strings of length r. The result by [4] shows that readability is well defined and is at most $2^{\Delta+1} - 1$, where Δ is the maximum of the in- and out-degrees of vertices in D. However, nothing else is known about the parameter, though there are papers that look at related notions [1,2,6,8,9,15,17,18].

In this paper, we study the asymptotic behavior of readability as a function of the number of vertices in a graph. We define readability for undirected bipartite graphs and show that the two definitions of readability are asymptotically equivalent. We capture readability using purely graph theoretic parameters (i.e., without any reference to strings). For trees, we give a parameter that characterizes readability exactly. For the larger family of bipartite C_4-free graphs, we give a parameter that approximates readability to within a factor of 2. Finally, for general bipartite graphs, we give a parameter that is bounded on the same sets of graphs as readability.

We apply our purely graph theoretic interpretation to prove readability upper and lower bounds on several graph families. We show, using a counting argument, that almost all digraphs and bipartite graphs have readability of at least $\Omega(n/\log n)$. Next, we construct a graph family inspired by Hadamard codes and prove that it has readability $\Omega(n)$. Finally, we show that the readability of trees is bounded from above by their radius, and there exist trees of arbitrary readability that achieve this bound.

2. Preliminaries

2.1. General definitions and notation

We use ϵ to denote the empty string. Let x be a string. We denote the length of x by $|x|$. We use $x[i]$ to refer to the ith character of x, and denote by $x[i..j]$ the substring of x from the ith to the jth character, inclusive. We let $\mathrm{pre}_i(x)$ denote the prefix $x[1..i]$ of x, and we let $\mathrm{suf}_i(x)$ denote the suffix $x[|x| - i + 1..|x|]$. Let y be another string. We denote by $x \cdot y$ the concatenation of x and y. We say that x *overlaps* y if there exists an i with $1 \le i \le \min\{|x|, |y|\}$ such that $\mathrm{suf}_i(x) = \mathrm{pre}_i(y)$. In this case, we say that x overlaps y by i. If $i < \min\{|x|, |y|\}$, then we call the overlap *proper*. Define $\mathrm{ov}(x, y)$ as the minimum i such that x overlaps y by i, or 0 if x does not overlap y. For a positive integer n, we denote by $[n]$ the set $\{1, \dots, n\}$.

We refer to finite simple undirected graphs simply as graphs and to finite directed graphs without parallel arcs in the same direction as digraphs. For a vertex v in a graph, we denote the set of neighbors of v by $N(v)$. A P_4 denotes the path on 4 vertices and 3 edges. A *biclique* is a complete bipartite graph. Note that the one-vertex graph is a biclique (with one of the parts of its bipartition being empty). Two vertices u, v in a graph are called *twins* if they have the same neighbors, i.e., if $N(u) = N(v)$. If, in addition, $N(u) = N(v) \neq \emptyset$, vertices u, v are called *non-isolated twins*. A *matching* is a graph of maximum degree at most 1, though we will sometimes slightly abuse the terminology and not distinguish between matchings and their edge sets. A cycle (respectively, path) on i vertices is denoted by C_i (respectively, P_i). For graph terms not defined here, see, e.g., [3].

We denote by $\mathcal{B}_{n\times n}$ the set of balanced bipartite graphs with nodes $[n]$ in each part, and by $\mathcal{D}_n$ the set of all digraphs with nodes $[n]$.

2.2. Readability of digraphs

A *labeling* ℓ of a graph or digraph is a function assigning a string to each vertex such that all strings have the same length, denoted by $len(\ell)$. We define $\mathrm{ov}_\ell(u, v) = \mathrm{ov}(\ell(u), \ell(v))$. An *overlap labeling* of a digraph $D = (V, A)$ is a labeling ℓ such that $(u, v) \in A$ if and only if $0 < \mathrm{ov}_\ell(u, v) < len(\ell)$. An overlap labeling is said to be *injective* if it does not generate duplicate strings. Recall that the readability of a digraph D, denoted $r(D)$, is the smallest nonnegative integer r such that there exists an injective overlap labeling of D of length r. We note that in our definition of readability we do not place any restrictions on the alphabet size. Braga and Meidanis [4] gave a reduction from an overlap labeling of length ℓ over an arbitrary alphabet Σ to an overlap labeling of length $\ell(2 \log |\Sigma| + 1)$ over the binary alphabet.

2.3. Readability of bipartite graphs

We also define a modified notion of readability that applies to balanced bipartite graphs as opposed to digraphs. We found that readability on balanced bipartite graphs is simpler to study but is asymptotically equivalent to readability on digraphs. Let $G = (V, E)$ be a bipartite graph with a given bipartition of its vertex set $V(G) = V_s \cup V_p$. (We also use the notation $G = (V_s, V_p, E)$.) We say that G is *balanced* if $|V_s| = |V_p|$. An *overlap labeling of* G is a labeling ℓ of G such that for all $u \in V_s$ and $v \in V_p$, $(u, v) \in E$ if and only if $\mathrm{ov}_\ell(u, v) > 0$. In other words, overlaps are exclusively between the suffix of a string assigned to a vertex in V_s and the prefix of a string assigned to a vertex in V_p. The *readability* of G is the smallest nonnegative integer r such that there exists an overlap labeling of G of length r. Note that we do not require injectivity of the labeling, nor do we require the overlaps to be proper. As before, we use $r(G)$ to denote the readability of G.