Structural similarity of directed universal hierarchical graphs: A low computational complexity approach

Matthias Dehmer a,b,c,d,*, Frank Emmert-Streib e

a Center for Integrative Bioinformatics Vienna, Max F. Perutz Laboratories, Dr. Bohr Gasse 9, A-1030 Vienna, Austria
b University of Vienna, Austria
c Medical University of Vienna, Austria
d University of Veterinary Medicine, Vienna, Austria
e Stowers Institute for Medical Research, 1000 E. 50th Street, Kansas City, MO 64110, USA

Abstract

In the present paper we mainly introduce an efficient approach to measure the structural similarity of so called directed universal hierarchical graphs. We want to underline that directed universal hierarchical graphs can be obtained from generalized trees which are already introduced. In order to classify these graphs, we state our novel graph similarity method. As a main result we notice that our novel algorithm has low computational complexity.

© 2007 Elsevier Inc. All rights reserved.

Keywords: Structured objects; Hierarchical models; Graph classes; Graph classification; Graph similarity; Complexity analysis

1. Introduction

Due to the increasing amount and complexity of real world data, there is a strong need to develop meaningful data mining methods [24]. In many application areas, e.g., web mining [20,33], computational linguistics [32,34], computational biology [3,18,31], and social network analysis [4,37], this data is naturally structural and can often be interpreted as relational objects, i.e., as graphs [2,9,25]. Starting from large graph databases, particularly methods for analyzing graphs structurally [31], i.e., special graph measures [22] and graph classification methods based on graph similarity approaches are extremely important [12]. In applied mathematics, there are two main concepts for comparing graphs: exact and inexact graph matching [6,12]. Exact graph matching basically means that we are looking for measures which are based on isomorphic or subgraph isomorphic relations [38,39,41]. In contrast to this, inexact graph matching mainly means that one measures the similarity (or distance) between graphs based on graph transformations, i.e., weighted transformation steps like deletions, substitutions, and insertions of vertices and edges [6]. From a machine learning [8] oriented view
point, graph classification can be divided into two major categories: *supervised* and *unsupervised* methods [8,28]. Supervised graph classification is defined as the task of learning a classification function from training data representing a certain set of graphs. But in many cases such training data is difficult to obtain. In contrast to this, unsupervised graph classification relates to extract relations between the graphs under consideration without any learning rules.

In this work we present an unsupervised learning approach for measuring the structural similarity of a novel graph class which we call *universal directed hierarchical graphs*. We call a set of graphs a graph class, if all elements possess the same structural properties according to a certain definition [5,15,36]. Further, we consider universal directed hierarchical graphs as a generalization of *generalized trees* which are already introduced [11,34]. In [11], we introduced an approach for measuring the structural similarity of such generalized trees. The main idea of this method was to derive so called property strings (see Definition 4.3) and then align the property strings by a dynamic programming technique [11]. By cumulating local similarity scores obtained from the mentioned alignments, we got the final graph similarity measure. Fig. 1 shows the main steps for measuring the structural similarity of generalized trees based on the algorithm proposed in [11]. On the one hand, our novel method for measuring the structural similarity of universal directed hierarchical graphs is also based on deriving the property strings. These strings cover structural information of our graphs. On the other hand, our new method is does not based on a dynamic programming technique to define local similarity measures which lead us to the final graph similarity measure. Instead of performing string alignments based on a dynamic programming technique, we derive other entities from the obtained property strings which cover structural information to compare universal directed hierarchical graphs. As a main result, we obtain that our novel method for measuring the structural similarity of universal directed hierarchical graphs has low computational complexity. Hence, we are able to process large graph sets containing graphs with a large number of vertices.

This paper is organized as follows: in Section 2, we introduce necessary mathematical preliminaries. In Section 3, we repeat the definition of a generalized tree and introduce the class of universal directed hierarchical graphs. In order to motivate both graph classes in practice, we give existing and future applications. The novel method to measure the structural similarity of universal directed hierarchical graphs is presented in Section 4.

Fig. 1. Steps to measure the structural similarity of generalized trees [10,11].
As a result, we define a measured value and prove that this is a similarity measure. Section 5 examines the computational complexity of our approach. The main result of this section is that the time complexity of the underlying graph similarity algorithm is polynomial. Section 6 finishes this paper with a summary and conclusion.

2. Mathematical preliminaries

In order to define the novel graph classes and develop a novel graph similarity approach, we need some graph-theoretical preliminaries [2,23,25].

**Definition 2.1.** Let \( V \neq \emptyset \) be a set of vertices. \( H = (V, E), |V| < \infty \) is a finite undirected graph, where \( E \subseteq \binom{V}{2} \). Generally, \( V \) is called the vertex and \( E \) is called the edge set.

**Definition 2.2.** Let \( V \neq \emptyset \) be the set of vertices. We call \( H = (V, E), E \subseteq V \times V, |V| < \infty \) a finite directed graph.

**Definition 2.3.** Let \( H := (V, E) \) be a finite directed graph and \( G := (\hat{V}, \hat{E}) \) be a subgraph of \( H \), that means, \( \hat{V} \subseteq V \) and \( \hat{E} \subseteq E \). We denote the subgraph \( G \) of \( H \) with \( G \subseteq H \). In the case it holds \( \hat{E} = E \cap (\hat{V} \times \hat{V}) \), then we call \( G \) an induced subgraph of \( H \).

**Definition 2.4.** Let \( H := (V, E) \) be a finite directed graph. Then we define the following characteristics:

\[ \mathcal{S}_+(v) := \{ \hat{v} \in V \setminus \{v\} | (v, \hat{v}) \in E \}, \]

\[ \mathcal{S}_-(v) := \{ \hat{u} \in V \setminus \{v\} | (\hat{u}, v) \in E \}, \]

\[ \delta_{\text{out}}(v) := |\mathcal{S}_+(v)|, \]

\[ \delta_{\text{in}}(v) := |\mathcal{S}_-(v)|. \]

**Definition 2.5.** An undirected graph is called undirected tree if this graph is connected and cycle free. We call a directed graph directed tree if the underlying undirected graph is a tree. A directed rooted tree \( T = (V_T, E_T) \) is a directed graph which has exactly one vertex \( r \in V_T \) with \( \delta_{\text{in}}(r) = 0 \). All vertices in \( T \) are uniquely accessible from \( r \) and \( r \) is called root vertex.

3. A novel graph class

In this section we mainly introduce the novel graph class of directed universal graphs. In order to motivate this graph class theoretically and practically, we first present the graph class of generalized trees recently introduced in [11,17]. For both classes, we additionally express existing and future applications.

3.1. Generalized trees

Now, we express a formal definition of a generalized tree. The starting point to define a generalized tree is an ordinary directed rooted tree as defined in Section 2.

**Definition 3.1.** Let \( T = (V, E_{\text{GT}}) \) be an directed rooted tree. The vertex set is defined by

\[ V := \{ v_{0,1}, v_{0,2}, \ldots, v_{0,|V_0|}, v_{1,1}, v_{1,2}, \ldots, v_{1,|V_1|}, v_{2,1}, v_{2,2}, \ldots, v_{2,|V_2|}, \ldots, v_{d,1}, v_{d,2}, \ldots, v_{d,|V_d|} \}, \]

and we assume \( |V| < \infty \). \(|\mathcal{L}|\) denotes the cardinality of the level set \( \mathcal{L} \) and \( d \) denotes the depth of \( T \). It holds \( |\mathcal{L}| = d + 1 \). The surjective mapping \( L : V \rightarrow \mathcal{L} \) is called a multi level function that assigns to all vertices an element of the level set \( \mathcal{L} \). \( v_{i,j} \) denotes the \( j \)th vertex on the \( i \)th level, \( 0 \leq i \leq d, 1 \leq j \leq |V_i| \). \(|V_i|\) denotes the number of vertices on level \( i \). The edge set \( E_{\text{GT}} := E_1 \cup E_2 \cup E_3 \cup E_4 \) of a finite generalized tree is defined as [11,34].
• \( E_1 \) forms the edge set of the underlying directed rooted tree \( T \).
• \((E_2)\) Up-edges associate analogously vertices of the tree hierarchy with one of their (dominating) predecessor vertices. Formally, we state
  \[
  E_2 := \left\{ (v_{i, \eta^+, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}) | \eta, \eta' \in \mathbb{N}, L(v_{i, \eta'}) = L(v_{i, \eta''}) - s, \right. \\
  \left. 1 \leq s \leq d \land \exists (v_{i, \eta'}, v_{i, \eta''}) \in E_1, \eta' \leq |V_i|, \ldots, 1 \leq |V_i| \leq |V_{i+s}|, \right. \\
  \left. 1 \leq \eta^+ \leq |V_{i+s}| \right\},
  \]
• \((E_3)\) Down-edges associate vertices of the tree hierarchy with one of their (dominated) successor vertices in terms of that tree hierarchy. Formally, we state
  \[
  E_3 := \left\{ (v_{i, \eta^+, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}) | \eta, \eta' \in \mathbb{N}, L(v_{i, \eta'}) = L(v_{i, \eta''}) + s, \right. \\
  \left. 1 < s \leq d \land \exists (v_{i, \eta'}, v_{i, \eta''}) \in E_1, \eta' \leq |V_i|, \ldots, 1 \leq |V_i| \leq |V_{i+s}|, \right. \\
  \left. 1 \leq \eta^+ \leq |V_{i+s}|, 1 \leq \eta^+ \leq |V_{i+s}| \right\},
  \]
• \((E_4)\) Across-edges associate vertices of the tree hierarchy, none of which is an (immediate) predecessor of the other in terms of the tree hierarchy. The set \( E_4 := E_4^{i-1} \cup E_4^{i+1} \cup E_4^{i+s} \) of Across-edges can be formally defined as the union of subsets of \( E_4 \), where
  \[
  E_4^{i-1} := \left\{ (v_{i, \eta^+, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}) | \eta, \eta' \in \mathbb{N}, L(v_{i, \eta'}) = L(v_{i, \eta''}) \land (\eta < \eta' \lor \eta > \eta') \right\},
  \]
  \[
  E_4^{i+1} := \left\{ (v_{i, \eta^+, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}) | \eta, \eta' \in \mathbb{N}, L(v_{i, \eta'}) = L(v_{i, \eta''}) - s, 1 \leq s \leq d \right\},
  \]
  \[
  E_4^{i+s} := \left\{ (v_{i, \eta^+, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}, v_{i, \eta''}) | \eta, \eta' \in \mathbb{N}, L(v_{i, \eta'}) = L(v_{i, \eta''}) + s, 1 \leq s \leq d \right\}.
  \]
If the sets \( E_2, E_3 \) or \( E_4 \) are nonempty, then \( H = (V, E_{GT}) \) is called a finite generalized tree.

**Remark 3.1.** It is immediately clear that Definition 3.1 can be extended to vertex-labeled generalized trees.

Fig. 2 shows a generalized tree \( H_1 \) and an ordinary directed rooted tree \( H_2 \).

By outlining some applications, we now briefly motivate generalized trees from a practical point of view.

### 3.2. Applications of generalized trees in data analysis

Novel problems dealing with discrete structures often require novel structure representations. Because of the fact that generalized trees are unexplored so far, we give some existing applications to demonstrate that a generalized tree is a useful generalization of an ordinary directed rooted tree.

1. **Web Mining** [7,35]: the subfield Web Structure Mining mainly deals with the structural analysis of web-based units, e.g., web sites or web pages. A main problem in Web Structure Mining is to extract information, e.g., certain link structures from the web and represent those structures as graphs [35].

![Fig. 2](https://example.com/fig2.png)

Fig. 2. \( H_1 \) shows a (directed) generalized tree with the edge types given in Definition 3.2. In contrast to this, \( H_2 \) shows an ordinary directed rooted tree. Apart from edges of an ordinary directed rooted tree, \( H_2 \) also contains edges \( e \in E_2, E_3, E_4 \). By applying Definition 3.1, we emphasize that an Across-edge \( e \in E_4 \) does not necessarily change a level \( i \). An Up-edge \( e \in E_2 \) can either change or over-jump a level.
example, Mehler et al. [34] inferred web site structures from the web representing generalized trees [21,34]. Then, based on a breath first search procedure, the inferred tree hierarchy represents an ordinary directed rooted tree [21,34]. Here, the inferred tree hierarchy approximates the navigation structure of this web site intended by the web site author. After inferring the tree hierarchy, the other edges types of a generalized tree were inferred, here Across-edges and Up-edges [21,34]. For example, Across-edges could be interpreted as a topic change within the web site hierarchy [12]. Because of the fact that web-based document structures, e.g., DOM-structures [7] are often represented as ordinary rooted trees, the generalized tree representation can be of considerable interest because it covers more structural information, compared to a DOM-structure. Generally, the problem to transform arbitrary graphs into generalized trees and vice versa has been treated mathematically by Emmert-Streib et al. [17].

(2) Computational and systems biology [29]:

(a) Emmert-Streib et al. [18] introduced an approach for comparing large unweighted, undirected graphs structurally. The unweighted, undirected graphs were obtained from microarray data of cervical cancer and different graphs represented different tumor stages. Hence, the classification problem of tumor stages was mapped to a classification problem of graphs. In order to solve this graph classification problem each graph was locally decomposed in a generalized tree where the hierarchy of the generalized tree is induced naturally by the Dijkstra-distance [17]. Performing this decomposition for each vertex in a graph results in a set of generalized trees for each graph. Finally, the sets of generalized trees were classified based on the method of Dehmer et al. [11]. Hence, it was demonstrated that this new graph classification method can classify graphs of order $10^4$ and, hence, tumor stages of cervical cancer successfully [18].

(b) In order to finalize the outlook of application cases, we briefly outline a problem for comparative RNA structure analysis as future work. For example, Höchstmann et al. [27] transformed RNA secondary structures into undirected rooted trees. But this method can only process secondary structures so far. However, the knowledge about tertiary structure analysis has been recently increased, see e.g., Lescoute et al. [30] and Gesell et al. [19]. For example, Gesell et al. [19] developed a general simulation framework for arbitrary site-specific interactions, including RNA tertiary interactions [19]. To perform a comparative analysis of such inferred tertiary structures, we argue that a generalized tree is an adequate graph representation. For measuring their structural similarity we refer to the already mentioned method of Dehmer et al., e.g., [11].

### 3.3. Directed universal hierarchical graphs

In this section we introduce the novel graph class of directed universal hierarchical graphs by extending the generalized tree model. We will see that if we allow more than one root vertex on level 0, this leads us to the definition of a directed universal hierarchical graph. In this sense, we call such a graph universal because these graph structures are more complex than generalized trees. That means, by maintaining the hierarchy information and allow that level 0 has more than one vertex, the graphs under consideration loose the tree character because the tree hierarchy does not longer uniquely exist. Finally, the structure of directed universal hierarchical graphs tend to network-like structures, compared to generalized trees which are typically tree-like.

Now, we state the formal definition of a directed universal hierarchical graph.

**Definition 3.2.** Let

\[ V := \{v_{0,1}, v_{0,2}, \ldots, v_{0,|P_u|}, v_{1,1}, v_{1,2}, \ldots, v_{1,|V_1|}, v_{2,1}, v_{2,2}, \ldots, v_{2,|V_2|}, \ldots, v_{d,1}, v_{d,2}, \ldots, v_{d,|V_d|}\}, \]

$|V| < \infty$ be the vertex set. \( L : V \rightarrow \mathcal{L} \) denotes again the multi level function (see Definition 3.1). It assigns to all vertices an element of the level set \( \mathcal{L} \) and it holds $d = |\mathcal{L}| - 1$. \( v_{i,j} \) denotes the \( j \)th vertex on the \( i \)th level, $0 \leq i \leq d, 1 \leq j \leq |V_i|$. \( |V_i| \) denotes the number of vertices on level \( i \). The edge set \( E_{DUHG} := E_1 \cup E_2 \cup E_3 \) is defined as follows:

- **(E₁)** Down-edges are edges which change at least one level.

\[ E_1 := \{(v_{i,p'}, v_{i + s,q'^s})|v_{i,p'}, v_{i + s,q'^s} \in V, 0 \leq i \leq d, 1 \leq j \leq |V_i|, L(v_{i + s,q'^s}) = L(v_{i,p'}) + s, 1 \leq s \leq d\}, \]
\( E_2 \) Up-edges are edges which change at least one level.

\[
E_2 := \{(v_i, v_j) | v_i, v_j \in V, 0 \leq i \leq d, 1 \leq j \leq |V|, L(v_i) = L(v_j) - s, 1 \leq s \leq d\},
\]

\( E_3 \) An Across-edge does not change a level.

\[
E_3 := \{(v_i, v_j) | v_i, v_j \in V, 0 \leq i \leq d, L(v_i) = L(v_j) \land (\hat{\eta}^i < \hat{\eta}^j \lor \hat{\eta}^i > \hat{\eta}^j)\}.
\]

Then, \( H = (V, E_{DUHG}) \) is called a finite directed universal hierarchical graph. Further, \( \mathcal{H}_{DUHG} \) denotes the set of finite directed universal hierarchical graphs.

**Remark 3.2.** As in Definition 3.1, it follows immediately that Definition 3.2 can be extended to vertex-labeled directed universal hierarchical graphs.

**Remark 3.3.** Definition 3.2 presents one possible edge type specification. We want to emphasize that the specification of edge types is generally not unique. We notice that a different differentiation of edges types leads us to different stages of usability and flexibility.

Fig. 3 shows two examples of directed universal hierarchical graphs.

### 3.4. Applications of directed universal hierarchical graphs

In this section we provide an example of a directed universal hierarchical graph representing a complex biological process. In Fig. 4 we show a simplified version of the transcriptional regulatory network of the cell cycle of *Saccharomyces cerevisiae* consisting of 19 genes [13]. Vertices in the graph correspond to genes and an edge connects two genes if there is a molecular interaction between these two genes or products thereof, e.g., proteins the genes are translated in, during the cell cycle. The direction of an edge corresponds to, e.g., gene A regulates gene B. The labels of the five hierarchy level correspond to the biological phases G1, S, G2, M and M/G1 of the cell cycle [1] which are processed in temporal order. That means, hierarchy levels are per se induced by the time arrow of the underlying biological process. The multi level function of this directed universal hierarchical graph can be defined as the phase (level) a gene plays an active role.

The possible ambiguity that a gene can be active at more than one phase (level) can be avoided by assigning the gene to the phase it appears first [13,40]. The apparent gain to represent the cell cycle of *S. cerevisiae* as an directed universal hierarchical graph and not as (normal) graph consists in the additional information provided by the hierarchy levels. Interestingly, the graph representation of the cell cycles provided in [13] was discussed in the context of the five phases of the cell cycle which are now included in the directed universal hierarchical graph. It is clear, that 'time' is not the only variable that can be used to introduce hierarchy levels. In general, all variables for which relations \((>,<)\) can be defined can be used. In the example presented above it would be possible to use the variable ‘interaction distance’ to induce different hierarchy levels. We want to remark, that the provided example is representative for all kinds of networks found omnipresently in molecular biology. Other examples are signaling, metabolic or protein–protein networks. Also, the considered
organism (*S. cerevisiae*) does not play a dominating role but can be substituted by any other organism. This implies, that all gene networks occurring in molecular and cell biology could be represented as directed universal hierarchical graphs.

4. Structural similarity of directed universal hierarchical graphs

In this section we present a method for measuring the structural similarity of directed universal hierarchical graphs. We want to notice that our novel method does not belong to the exact graph matching paradigm [6,12] because it does not deal with isomorphic and subgraph isomorphic relations. In contrast to match graph and subgraph patterns exactly based on graph isomorphisms, our approach is mainly based on the steps as follows:

1. In order to take advantage of the information induced by the hierarchy of a $H \in \mathcal{H}_{DUHG}$ to measure their structural similarity, we especially transform such a graph in a set $S = \{s_1, s_2, \ldots, s_d\}$, where the $s_i, 1 \leq i \leq d, d = |S| - 1$ are strings. These strings cover structural information of the graphs under consideration.
2. Based on the resulting string representations, we define local similarity functions. Finally, this leads us to a novel graph similarity measure.
3. Taking Step (1) and Step (2) into account, that finally means we transform a given graph similarity problem into a corresponding string similarity problem.

In the following we explain the steps stated above in more detail. We start with transforming our graphs into a certain string set.

4.1. String representation of directed universal hierarchical graphs

**Definition 4.1.** Let $A$ be a finite and nonempty set that we call alphabet. We call $s \in A$ a string of $A^*$, where $A^*$ denotes the set of all finite strings on $A$. 

Fig. 4. A directed universal hierarchical graph representing a simplified version of the transcriptional regulatory network of the cell cycle of *S. cerevisiae* consisting of 19 genes [13]. The labels of the five hierarchy level correspond to the biological phases G1, S, G2, M and M/G1 of the cell cycle [1].
**Definition 4.2.** Let \( H \in DUHG \). We call\(^1\) the set  
\[
S^H := \left\{ v^H_0, v^H_1 \circ v^H_2 \circ \cdots \circ v^H_{|V_1|}, \ldots, v^H_d \circ v^H_{d+1} \circ \cdots \circ v^H_{d|V_d|} \right\},
\]
the formal string representation of \( H \).

**Definition 4.3.** Let \( H \in DUHG \). We call  
\[
S^H_{\text{out}} := \left\{ \delta_{\text{out}}(v^H_0) \circ \delta_{\text{out}}(v^H_1) \circ \cdots \circ \delta_{\text{out}}(v^H_{|V_1|}), \ldots, \delta_{\text{out}}(v^H_1) \circ \cdots \circ \delta_{\text{out}}(v^H_{d|V_d|}) \right\},
\]
the set of out-degree property strings of \( H \) and  
\[
S^H_{\text{in}} := \left\{ \delta_{\text{in}}(v^H_0) \circ \delta_{\text{in}}(v^H_1) \circ \cdots \circ \delta_{\text{in}}(v^H_{|V_1|}), \ldots, \delta_{\text{in}}(v^H_1) \circ \cdots \circ \delta_{\text{in}}(v^H_{d|V_d|}) \right\},
\]
the set of in-degree property strings of \( H \). \( s^\text{out} \) denotes the out-degree property string and \( s^\text{in} \) the in-degree property string on level \( i \), respectively. If we set \( A = \mathbb{N} \), then \( s^\text{out}_i, s^\text{in}_i \in A^* \), \( 1 \leq i \leq d \).

As an example, we look at \( H_1 \) of Fig. 3. Here, the sets of out-degree property strings and in-degree property strings of \( H_1 \) are  
\[
S^H_{\text{out}}(H_1) = \{ 1 \circ 3 \circ 3, 2 \circ 1 \circ 0 \} \quad \text{and} \quad S^H_{\text{in}}(H_1) = \{ 1 \circ 2 \circ 2, 1 \circ 2, 1 \circ 0 \},
\]
respectively. For example, the out-degree property string \( s^\text{out}_1 \) on level +1 of \( H_1 \) is given as \( s^\text{out}_1 = 1 \circ 3 \circ 3 \).

### 4.2. A similarity measure for directed universal hierarchical graphs

In this section we present an efficient approach for measuring the structural similarity of directed universal hierarchical graphs. Our novel method is mainly based on defining similarity measures which operate on certain entities (margins) from the corresponding out-degree property and in-degree property strings (see Definition 4.3). The out-degree property and in-degree property strings cover structural information of the graphs under consideration. Finally, a cumulation of local similarity values leads to a similarity measure between directed universal hierarchical graphs.

We start with a formal definition of a similarity measure.

**Definition 4.4.** Let \( X \) be a set. A positive real function \( s : X \times X \rightarrow [0, 1] \) is called similarity measure if  
\[
\beta(x, y) = \beta(y, x) \quad \forall x, y \in X
\]
\[
\beta(x, y) \leq \beta(x, x) \quad \forall x, y \in X
\]
\[
\beta(x, x) = 1 \quad \forall x \in X
\]

**Definition 4.5.** Let \( H_1 \) and \( H_2 \) be finite directed universal hierarchical graphs.

Starting from the \( i \)th out-degree property string  
\[
s^\text{out}_i = \delta_{\text{out}}(v^H_{i,1}) \circ \delta_{\text{out}}(v^H_{i,2}) \circ \cdots \circ \delta_{\text{out}}(v^H_{i,|V_i|})
\]
and in-degree property string  
\[
s^\text{in}_i = \delta_{\text{in}}(v^H_{i,1}) \circ \delta_{\text{in}}(v^H_{i,2}) \circ \cdots \circ \delta_{\text{in}}(v^H_{i,|V_i|}),
\]

\(^1\) The symbol \( \circ \) denotes usual string concatenation.
of a $H \in \mathcal{H}_{DUHG}$, we obtain

$$s_{(i)}^{\text{out}} = \delta_{\text{out}}(v_{(i,1)}^H) \circ \delta_{\text{out}}(v_{(i,2)}^H) \circ \cdots \circ \delta_{\text{out}}(v_{(i,|V^H_i|)}^H),$$

and

$$s_{(i)}^{\text{in}} = \delta_{\text{in}}(v_{(i,1)}^H) \circ \delta_{\text{in}}(v_{(i,2)}^H) \circ \cdots \circ \delta_{\text{in}}(v_{(i,|V^H_i|)}^H),$$

where it now holds $\delta_{\text{out}}(v_{(i,j+1)}^H) \geq \delta_{\text{out}}(v_{(i,j)}^H)$ and $\delta_{\text{in}}(v_{(i,j)}^H) \geq \delta_{\text{in}}(v_{(i,j+1)}^H)$. We define $n_i$ as $n_i := \max(|V_{i,j}^H|, |V_{i,j}^H|)$. By adding zero entries, we obtain two sequences of equal length $n_\mu$, i.e.,

$$s_{(i)}^{\text{out}} = \delta_{\text{out}}(v_{(i,1)}^H) \circ \delta_{\text{out}}(v_{(i,2)}^H) \circ \cdots \circ \delta_{\text{out}}(v_{(i,|V^H_i|)}^H),$$

and

$$s_{(i)}^{\text{in}} = \delta_{\text{in}}(v_{(i,1)}^H) \circ \delta_{\text{in}}(v_{(i,2)}^H) \circ \cdots \circ \delta_{\text{in}}(v_{(i,|V^H_i|)}^H).$$

Now, let

$$s_{(i)}^{\text{out}}(H_1) = \delta_{\text{out}}(v_{(i,1)}^H_1) \circ \delta_{\text{out}}(v_{(i,2)}^H_1) \circ \cdots \circ \delta_{\text{out}}(v_{(i,|V^H_i|_1)}^H_1),$$

$$s_{(i)}^{\text{in}}(H_2) = \delta_{\text{in}}(v_{(i,1)}^H_2) \circ \delta_{\text{in}}(v_{(i,2)}^H_2) \circ \cdots \circ \delta_{\text{in}}(v_{(i,|V^H_i|_2)}^H_2),$$

the $i$th out-degree and in-degree property string of $H_1$ and $H_2$, respectively. Then, we define the $i$th out-degree margins

$$\gamma_{[j_1], \gamma_{[j_2]}, \ldots, \gamma_{[j_m]}] \in \mathbb{N},$$

and in-degree margins

$$\gamma_{[j_1], \gamma_{[j_2]}, \ldots, \gamma_{[j_m]}] \in \mathbb{N},$$

as

$$\gamma_{[j_1]} = ||\delta_{\text{out}}(v_{(i,1)}^H_1) - \delta_{\text{out}}(v_{(i,1)}^H_2)||,$$

$$\gamma_{[j_2]} = ||\delta_{\text{out}}(v_{(i,2)}^H_1) - \delta_{\text{out}}(v_{(i,2)}^H_2)||,$$

$$\ldots$$

$$\gamma_{[j_m]} = ||\delta_{\text{out}}(v_{(i,m)}^H_1) - \delta_{\text{out}}(v_{(i,m)}^H_2)||,$$

and

$$\gamma_{[j_1]} = ||\delta_{\text{in}}(v_{(i,1)}^H_1) - \delta_{\text{in}}(v_{(i,1)}^H_2)||,$$

$$\gamma_{[j_2]} = ||\delta_{\text{in}}(v_{(i,2)}^H_1) - \delta_{\text{in}}(v_{(i,2)}^H_2)||,$$

$$\ldots$$

$$\gamma_{[j_m]} = ||\delta_{\text{in}}(v_{(i,m)}^H_1) - \delta_{\text{in}}(v_{(i,m)}^H_2)||,$$

respectively.

**Lemma 4.1.** Let $\zeta_1, \zeta_2, \ldots, \zeta_n$ be integer numbers and let $O_1, O_2$ be two arbitrary objects. We define

$$\gamma(O_1, O_2, \zeta_1, \zeta_2, \ldots, \zeta_n) := \frac{\beta(\zeta_1) + \beta(\zeta_2) + \cdots + \beta(\zeta_n)}{n},$$

where $\beta(\zeta_i) \leq 1, 1 \leq i \leq n$. Then, $\gamma(\zeta_1, \zeta_2, \ldots, \zeta_n)$ is a similarity measure.

**Proof.** Because it holds $\beta(\zeta_i) \leq 1, 1 \leq i \leq n$ and $\gamma(\zeta_1, \zeta_2, \ldots, \zeta_n)$ is additively defined the symmetric condition is clear. Under the assumption if $O_1 = O_2 \beta(\zeta_i) = 0, 1 \leq i \leq n$ holds, the assertion follows. □
Remark 4.2. For example,

\[ \beta(\xi, \sigma) := \xi^{\sigma^2} \quad \text{and} \quad \beta(\xi, \sigma) := \xi^{-\sigma}, \quad \sigma \in \mathbb{R} \]

are suitable local measures to define \( \gamma(O_n, O_1, \zeta_1, \zeta_2, \ldots, \zeta_n) \) in Lemma (4.1).

Now, we define the local similarity measures which finally detect the similarity between the corresponding out-degree and in-degree property strings.

**Definition 4.6.** Starting from Eq. (15) in Lemma 4.1, we call

\[ \gamma_i^{out} = \gamma_i^{out}(s_{[i,1]}^{out}, s_{[i,2]}^{out}, \ldots, s_{[i,n]}^{out}) \in [0, 1] \]

and

\[ \gamma_i^{in} = \gamma_i^{in}(s_{[i,1]}^{in}, s_{[i,2]}^{in}, \ldots, s_{[i,n]}^{in}) \in [0, 1], \]

local similarity functions of the out-degree and in-degree property strings on level \( i \).

Now, our main result for measuring the structural similarity of directed universal hierarchical graphs is the following theorem.

**Theorem 4.3.** Let \( H_1 \) and \( H_2 \) be directed universal hierarchical graphs and we define \( v := \max(d^{H_1}, d^{H_2}) \). It holds \( d^{H_1} = |G^{H_1}| - 1 \).

\[
s(H_1, H_2) := \prod_{i=0}^{v+1} \gamma_i \tag{16}
\]

is a graph similarity measure, i.e., it holds

\[
s(H_1, H_2) = s(H_2, H_1), \tag{17}
\]

\[
s(H_1, H_1) = 1, \tag{18}
\]

\[
0 < s(H_1, H_2) \leq s(H_1, H_1). \tag{19}
\]

\( \gamma_i \) is defined as

\[
\gamma_i \triangleq \gamma_i(s_{[i,1]}^{out}, s_{[i,2]}^{out}, \ldots, s_{[i,n]}^{out}, s_{[i,1]}^{in}, s_{[i,2]}^{in}, \ldots, s_{[i,n]}^{in}) :=
\]

\[
= \alpha \cdot \gamma_i^{out}(s_{[i,1]}^{out}, s_{[i,2]}^{out}, \ldots, s_{[i,n]}^{out}) + (1 - \alpha) \cdot \gamma_i^{in}(s_{[i,1]}^{in}, s_{[i,2]}^{in}, \ldots, s_{[i,n]}^{in}), \quad \alpha \in [0, 1]. \tag{20}
\]

**Proof.** Firstly, we define \( \bar{s}(H_1, H_2) := \sum_{v+1} \gamma_i \). Due to the fact that Eq. (15) in Lemma 4.1 and Eq. (20) in Theorem 4.3 are additively defined, we conclude that the symmetry property holds for \( \bar{s}(H_1, H_2) \), i.e., \( \bar{s}(H_1, H_2) = \bar{s}(H_2, H_1) \). With **Definition 4.6**, we infer that

\[
\bar{s}(H_1, H_1) = \alpha \cdot 1 + (1 - \alpha) \cdot 1 = 1.
\]

Hence,

\[
\bar{s}(H_1, H_1) = \frac{1 + 1 + \cdots + 1}{v + 1} = 1.
\]

Because we know that \( \bar{s}(H_1, H_2) \) is symmetric, we obtain that \( s(H_1, H_2) \) (see Eq. (16)) is symmetric. In order to finalize the proof of the theorem, we now prove **Property 19** which is equivalent to \( 1 = s(H_1, H_1) \geq s(H_1, H_2) \).

By using the well known Inequality (21) proven in [26],

\[
(p_1 \cdot p_2 \cdots p_n)^{\frac{1}{n}} \leq \frac{p_1 + p_2 + \cdots + p_n}{n}, \quad p_i > 0, \quad 1 \leq i \leq n, \tag{21}
\]

we get

\[
\bar{s}(H_1, H_1) \geq \bar{s}(H_1, H_2) \leq \frac{\bar{s}(H_1, H_1)}{v + 1}.
\]
where $\gamma_i \leq 1$. Finally, we obtain
\[
1 \geq (v + 1) \frac{\gamma_0 \cdot \gamma_1 \cdots \gamma_v}{\gamma_0 + \gamma_1 + \cdots + \gamma_v}.
\] (22)

But Inequality (22) means that Property 19 holds, i.e., $1 = s(H_1, H_1) \geq s(H_1, H_2)$. ☐

### 5. Efficiency analysis

Now, we state the results of our efficiency analysis that aims to determine the computational complexity of the novel approach presented in Section 4.

In order to examine the computational complexity of our novel approach we look at Fig. 5. In Fig. 5, we see the main steps for measuring the structural similarity of directed universal hierarchical graphs. In the following we examine the computational complexity of these steps. Proposition 5.1 has already been proven in [16].

**Proposition 5.1.** Let $H \in \mathcal{H}_{DUHG}$ and it holds $|V| = |V_H^1| + |V_H^2| + \cdots + |V^H_d|$. The computation of the out-degree and in-degree property strings of $H$ requires computational complexity
\[
O(d \cdot \max((|V^H_0|)^2, (|V^H_1|^2), \ldots, (|V^H_d|^2))).
\]

**Proposition 5.2.** Let $H \in \mathcal{H}_{DUHG}$. The computation of the local similarity measures of Definition 4.6 for a given $H$ requires computational complexity
\[
O(\max(|V_H^0|, |V_H^1|) + \max(|V_H^0|, |V_H^1|) + \cdots + \max(|V^H_0|, |V^H_1|)).
\]

**Proof.** Assume that we have already calculated the out-degree and in-degree property strings on each level $i$. Then, calculating the margins $\gamma_i^{\text{out}}$, $\gamma_i^{\text{in}}$, and the measures $\gamma_i^{\text{out}}$, $\gamma_i^{\text{in}}$ altogether requires (for each level)
\[
O(\max(|V_H^0|, |V_H^1|) + \max(|V_H^0|, |V_H^1|) + \cdots + \max(|V^H_0|, |V^H_1|)).
\]

From this equation, we also obtain Eq. (23). ☐

**Fig. 5.** Exemplary steps to measure the structural similarity of directed universal hierarchical graphs.
Proposition 5.3. Let $H_1, H_2 \in \mathcal{H}_{DUHG}$. Computing the final graph similarity measure $s(H_1, H_2)$ (see Eq. (16)) needs time complexity $O(\max(d_{H_1}, d_{H_2}))$.

Proof. Follows immediately from Eq. (16). □

Now, we express our main result of our efficiency analysis.

Theorem 5.4. Let $H_1, H_2 \in \mathcal{H}_{DUHG}$. The overall computational complexity for measuring the structural similarity of $H_1$ and $H_2$ is

$$\mathcal{O}(d \cdot \max((|V_{1|}^H)^2, (|V_{2|}^H)^2, \ldots, (|V_{d|}^H)^2) + \mathcal{O}(\max(|V_{0|}^H_1, |V_{0|}^H_2|) + \max(|V_{1|}^H_1, |V_{1|}^H_2|) + \ldots + \max(|V_{d|}^H_1, |V_{d|}^H_2|)) + \mathcal{O}(\max(|L_{1|}^H_1, |L_{1|}^H_2|)).$$ (24)

If we further set,

$$f_1 := \max((|V_{1|}^H_1)^2, (|V_{2|}^H_1)^2, \ldots, (|V_{d|}^H_1)^2),$$

$$f_2 := \max(\max(|V_{0|}^H_1, |V_{0|}^H_2|) + \max(|V_{1|}^H_1, |V_{1|}^H_2|) + \ldots + \max(|V_{d|}^H_1, |V_{d|}^H_2|)),

$$f_3 := \max(d_{H_1}, d_{H_2}),$$

we clearly obtain

$$\mathcal{O}(f_1 + f_2 + f_3) = \mathcal{O}(\max(f_1 + f_2 + f_3)).$$

Proof. We obtain the overall time complexity for computing the structural similarity of $H_1$ and $H_2$ by adding up the previously obtained time complexities. □

6. Summary and conclusion

In this paper, we introduced a novel method to measure the structural similarity of a novel graph class. We now summarize the main contributions of our paper as follows:

• In Section 1 we briefly outlined our known method to measure the structural similarity of so-called generalized trees [11]. In Section 3 we introduced a novel graph class that we called directed universal hierarchical graphs. We mainly obtained such a directed universal hierarchical graph by generalizing the generalized tree concept. We notice that a directed universal hierarchical graph can cover more structural information than a generalized tree. Once again, we want to emphasize that we have presented one possible edge type specification of such a graph; hence, the specification of edge types is not unique. It follows that a different differentiation of edge types leads us to different stages of usability and flexibility.

• The main part of the present paper was about a novel algorithm to measure the structural similarity of directed universal hierarchical graphs with low computational complexity. Similarly to [11], we first derived property strings, i.e., the induced out-degree and in-degree sequences on each level $i$. Instead of performing string alignments to obtain local similarity measures [11], we derived so-called margins which cover structural information of the compared property strings. Then, we defined local similarity measures to detect the similarity between these strings. By cumulating these local similarity functions, we obtain a novel graph similarity measure for directed universal hierarchical graphs. In Section 5, we performed an efficiency analysis to determine the computational complexity of our novel approach. As the main result of Section 5, we notice that our novel approach has (asymptotically) quadratic time complexity depending on the maximum number of vertices on a level $i$.

We think that directed universal hierarchical graphs are useful graph representations to solve practical problems. To underline this we provided in Section 3.4 an example of directed universal hierarchical graphs appearing omnipresently in molecular biology by showing the cell cycle of $S. cerevisiae$. Based on our novel

graph similarity algorithm presented in Section 4 we are now able to classify those graphs in an unsupervised learning sense [8]. That means, starting from a certain set $\mathcal{H} \subseteq \mathcal{H}_{DUHG}$ we can now divide $\mathcal{H}$ into certain categories which contain structurally similar directed universal hierarchical graphs, with respect to our novel graph similarity measure $s(H_1, H_2)$, automatically.

Acknowledgements

We would like to thank [14] for freely providing the graphviz software which we used to visualize all graphs. Financial support from the Vienna Science and Technology fond (WWTF) to Matthias Dehmer is gratefully acknowledged.

References

Further Reading