On the Validity of Hierarchical Decompositions*

Irene Finocchi and Rossella Petreschi

Department of Computer Science
University of Rome "La Sapienza"
{finocchi, petreschi}@dsi.uniroma1.it

Abstract. Hierarchical decompositions are a useful tool for drawing large graphs. Such decompositions can be represented by means of a data structure called hierarchy tree. In this paper we introduce the notion of $\mathcal{P}$*-validity* of hierarchy trees with respect to a given property $\mathcal{P}$: this notion reflects the similarity between the topological structure of the original graph and of any high-level representation of it obtained from the hierarchy tree. We study the $\mathcal{P}$*-validity* when the clustered graph is a tree and property $\mathcal{P}$ is the acyclicity, presenting a structure theorem for the $\mathcal{P}$*-validity* of hierarchy trees under these hypotheses.

1 Introduction and Preliminaries

Many graph drawing algorithms have been designed during the last years [2] and most of them move from the following assumption: in order to find a pleasant layout of a graph, we first of all should be able to recognize its graph-theoretical properties, e.g., acyclicity, planarity, bipartiteness, and so on. Actually, exploiting the relevant features of a graph helps produce better visualizations for it. It is also well accepted that only few graph drawing algorithms scale up well and are able to visualize the more and more large graphs that arise in practical applications. Clustering techniques are a useful tool to overcome this drawback and make it possible dealing with graphs not fitting in the screen [3,4,5,6]. Recursively clustering the vertices of a graph leads to a hierarchical decomposition of the graph itself: each cluster in the decomposition is considered as a single node which is visualized instead of a set of vertices of the original graph, considerably reducing the dimension of the drawing. Thanks to the parent relationships between clusters in the hierarchical decomposition, the viewer can move from a high-level representation of the graph to another one by detailing or shrinking clusters. The considerations above immediately lead to the following problem: *Given a hierarchical decomposition of a graph G and a graph-theoretical property $\mathcal{P}$ satisfied by G, does any high-level representation of G obtained from the hierarchical decomposition satisfy property $\mathcal{P}$?* Though this question naturally arises when using hierarchical decompositions for visualizing large graphs, as far as we

* Work partially supported by the project "Algorithms for Large Data Sets: Science and Engineering" of the Italian Ministry of University and of Scientific and Technological Research (MURST 40%).

J. Wang (Ed.): COCOON 2001, LNCS 2108, pp. 368–374, 2001.

know it has not been previously studied in literature. Some related work is due to Feng *et al.*, who focus on clustered planar graphs and investigate the concept of compound planarity [5].

In this paper we introduce the general notion of $\mathcal{P}$-*validity* of hierarchical decompositions of graphs with respect to a given property $\mathcal{P}$. This notion reflects the similarity between the topological structure of the original graph and of any high-level representation of it and allows us to compare different hierarchical decompositions associated to the same graph. We then focus on the case where the clustered graph is a tree and property $\mathcal{P}$ is the acyclicity and we present a structural characterization of the $\mathcal{P}$-*validity* under these hypotheses, i.e., conditions on the structure of the clusters necessary and sufficient to guarantee the $\mathcal{P}$-*validity* of the decomposition.

We start with preliminary definitions and notation. A hierarchical decomposition of a graph can be represented by means of a data structure called *hierarchy tree* (or sometimes *cluster* or *inclusion* tree) [1,5]. A hierarchy tree $HT = (N, A)$ associated to a graph $G = (V, E)$ is a rooted tree whose leaves are the vertices of G. Nodes of a hierarchy tree are called *clusters*: a cluster c represents a set V_c of vertices of G, namely, the vertices that are the leaves of the subtree rooted at c. We say that such vertices are *covered* by c and we refer to their number as *cardinality* of c. For brevity, we write $u \prec c$ to indicate that a vertex $u \in V$ is covered by a cluster $c \in N$. We call *singleton cluster* a cluster with cardinality equal to 1. W.l.o.g. we assume that the vertices covered by a cluster c are a proper subset of the vertices covered by the parent of c in HT, i.e., in the hierarchy tree there is no node with a unique child. Under this hypothesis, the number of nodes of a hierarchy associated to a n-vertex graph is at most $2n - 1$. This implies that adding a hierarchy tree on the top of a graph requires only space linear in the number of vertices of the graph itself.

It is possible to visit a hierarchy tree in a top-down fashion by performing expand and contract operations on the clusters visualized at any instant of time: these operations define a *navigation* of HT. The previous ideas can be formalized by defining the concepts of *covering* and of *view* [1,6]:

Definition 1. *A covering C of a graph $G = (V, E)$ on a hierarchy tree $HT = (N, A)$ is a subset of the nodes of HT such that each vertex of G is covered by exactly a node in C.*

A view of graph G on HT is a graph $W = (C, L)$ such that C is a covering of G on HT and $L = \{(c, c') \mid c, c' \in C,\ c \neq c',\ \exists (u, v) \in E\ :\ u \prec c \text{ and } v \prec c'\}$.

We refer to the edges of W as *links*. It is worth observing that any view is a simple graph: if there exist two edges $(u, v) \in E$ and $(u', v') \in E$ which lead to a connection between two clusters c and c', only a single link (c, c') is considered in L. We also call W_r and $W_l = G$ the views generated by the root of the hierarchy tree and by all its leaves, respectively: in other words, W_r and W_l are the least and the most detailed representations of the graph, respectively.

2 $\mathcal{P}$-Validity of Views and Hierarchy Trees

A navigation of a hierarchy tree should help the viewer to focus on particular interesting regions of the graph, according to his/her necessities; hence, if the clusters are generated not taking into account the topological structure of the graph, no benefit may derive for the viewer from the clustering structure. Informally speaking, only vertices in the same "locality" of the graph should be grouped together to form a cluster.

A motivating example of the concept of $\mathcal{P}$-*validity* is illustrated in Figure 1, where two different hierarchical decompositions associated to a chain with 8 vertices are considered (Figure 1(a) and Figure 1(d)). The hierarchy trees HT_1 (Figure 1(b)) and HT_2 (Figure 1(e)) corresponding to these decompositions are both complete binary trees of height 3 and differ only in the permutation of their leaves. Figure 1(c) and Figure 1(f) report three views related to coverings $\{\mathtt{E}, \mathtt{F}\}$, $\{\mathtt{E}, \mathtt{C}, \mathtt{D}\}$, and $\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}\}$ in the two decompositions, respectively. All the views built from HT_1 maintain the structural property of the original graph to be a chain, while the views from HT_2 loose this property introducing cycles, up to become even a clique.

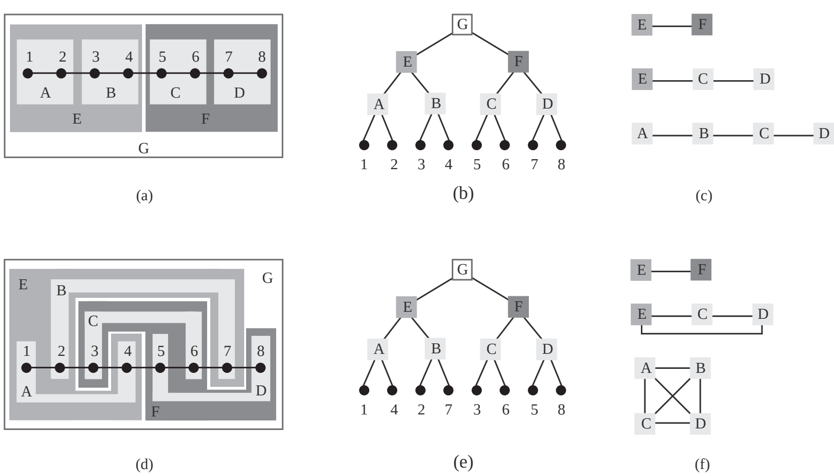

Fig. 1. Hierarchical decompositions, hierarchy trees, and views of a 8-vertex chain

In order to characterize the "semantic" differences of hierarchy trees with "syntactically" similar or even identical structure it is then natural to introduce a notion of validity of a view and of a hierarchy tree with respect to a certain property $\mathcal{P}$: from now on we refer to this concept as $\mathcal{P}$-*validity*.

Definition 2. *Let HT be a hierarchy tree associated to a graph G and let $\mathcal{P}$ be a property satisfied by G. A view W of G obtained from HT is $\mathcal{P}$-valid iff W*

satisfies property $\mathcal{P}$. HT is $\mathcal{P}$-valid iff all the views of G obtained from it are $\mathcal{P}$-valid.

In view of the fact that Definition 2 is parametric in the property $\mathcal{P}$ to be considered, different notions of validity may be thought for different classes of graphs. For example, we may require a view of a bipartite graph to be bipartite or a view of a planar graph to be planar. But even more sophisticated definitions for $\mathcal{P}$ can be considered: as an example, we refer to the *c-planarity* property introduced by Feng *et al.* [5]. The interest in characterizing $\mathcal{P}$-valid hierarchy trees naturally follows from the considerations in Section 1. In this paper we focus on the case where the clustered graph is a tree and property $\mathcal{P}$ is the acyclicity: for brevity, we will speak of valid views/hierarchy trees under these hypotheses. In Section 3 we prove necessary and sufficient conditions for a hierarchy tree to be valid.

3 A Structural Characterization of Valid Hierarchy Trees

Before presenting the structure theorem for the validity of a hierarchy tree, we give preliminary definitions and lemmas useful for proving it. First of all we study the connectivity of views obtained from hierarchy trees of connected graphs.

Lemma 1. *Each view obtained from a hierarchy tree associated to a connected graph is connected.*

In the following we denote with $S(c)$ the subgraph of $T = (V, E)$ induced by the vertices covered by a node c of HT. For each $u, v \in T$, let $path_T(u, v)$ be the path joining u and v in T and let $dist_T(u, v)$ be the length of this path.

Definition 3. *Let c be a node of a hierarchy tree HT associated to a free tree T. Let u and v be two vertices of T covered by c. u, v are a broken pair of cluster c iff they are neither coincident nor connected in $S(c)$. A broken pair u, v is a minimum-distance broken pair of c iff $w \not\prec c$, $\forall w \in path_T(u, v)$ such that $w \neq u, v$.*

Lemma 2. *Let W be a view on a hierarchy tree HT associated to a free tree T. If W is not acyclic, then in each cycle $\mathcal{C}$ there is at least a cluster containing a broken pair.*

Proof. Let $\mathcal{C} = (c_1, \ldots, c_h)$ be a cycle in W. Each cluster $c_i \in \mathcal{C}$ is endpoint of two links (c_{i-1}, c_i) and (c_i, c_{i+1}), respectively (to simplify the notation we assume that $h + 1 = 1$). Let us call (r_{i-1}, l_i) and (r_i, l_{i+1}) two tree edges which derive links (c_{i-1}, c_i) and (c_i, c_{i+1}), respectively. For each cluster c_i we therefore identify two vertices covered by it, named l_i and r_i (see Figure 2). If no broken pair exists in any cluster of $\mathcal{C}$, for each $i \in [1, h]$ vertices l_i and r_i are either coincident or connected in $S(c_i)$. This implies the existence of a cycle in tree T, that is a contradiction.

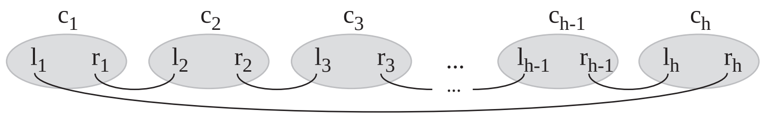

Fig. 2. Vertices, edges, clusters, and links involved in cycle $\mathcal{C}$ in the proof of Lemma 2

Theorem 1. *Let $T = (V, E)$ be a free tree and let $HT = (N, A)$ be a hierarchy tree associated to T. HT is valid iff for each minimum-distance broken pair u, v of HT $dist_T(u, v) = 2$.*

Proof. We first prove the necessary condition, showing that the existence in HT of a minimum-distance broken pair u, v such that $dist_T(u, v) > 2$ implies that HT is not valid. Let c be a node of HT such that u, v are a minimum-distance broken pair in c. Let us consider the view $W = (C, L)$ where covering C consists only of singleton clusters, except for cluster c: $C = \{c\} \cup \{\{x\},\ x \in V \text{ and } x \not\prec c\}$. Let z and w be the vertices in $path_T(u, v)$ such that $(u, z) \in E$ and $(w, v) \in E$. Since u, v is a minimum-distance broken pair of c, $z \not\prec c$, $w \not\prec c$, and any other vertex in $path_T(u, v)$ is not covered by c. Moreover, since $dist_T(u, v) > 2$, $dist_T(z, w) > 0$, i.e., z and w are distinct vertices. Hence $\langle c, \{z\}, \ldots, \{w\}, c\rangle$ is a cycle in W, proving that HT is not valid.

We now focus on the sufficient condition, proving that a contradiction can be derived if we suppose that HT is not valid while satisfying the property on the minimum-distance broken pairs formulated in the statement of the theorem. Namely, we assume that in HT there exists a non-valid view $W = (C, L)$, i.e., a view W that is not a tree. Since W must be connected due to Lemma 1, it must contain a simple cycle $\mathcal{C}$. We prove that this assumption leads to a contradiction.

The general idea of the proof is to convert W into another view W', still existing on the hierarchy tree, such that: (a) the number of singletons of W' is strictly greater than the number of singletons of W; (b) the cycle $\mathcal{C}$ of W is also changed into a simple cycle $\mathcal{C}'$ in W'. The sequence of manipulations that we perform has finite length, as the number of singletons is clearly upper bounded by $n = |V|$. We can therefore prove that at some step during this process we find a contradiction due to Lemma 2, since we obtain a cycle with no broken pairs.

Let $\mathcal{C} = (c_1, \ldots, c_h)$ be a simple cycle of length h in W and let c_i, l_i, and r_i be defined as in the proof of Lemma 2 for $1 \le i \le h$ (see also Figure 2). Due to Lemma 2 a broken pair must exist in $\mathcal{C}$. Let l_i and r_i be the vertices in such a broken pair of $\mathcal{C}$. We remark that l_i and r_i are not necessarily a minimum-distance broken pair.

Let us consider the path from l_i to r_i in T, which is unique and not completely contained in $S(c_i)$. On this path we can univocally identify a set of k vertices u_j, for $1 \le j \le k$, such that $u_j \prec c_i$ but its successor on $path_T(l_i, r_i)$ is not covered by c_i. Analogously, we can univocally identify a set of k vertices v_j, for $1 \le j \le k$, such that $v_j \prec c_i$ but its predecessor on $path_T(l_i, r_i)$ is not covered

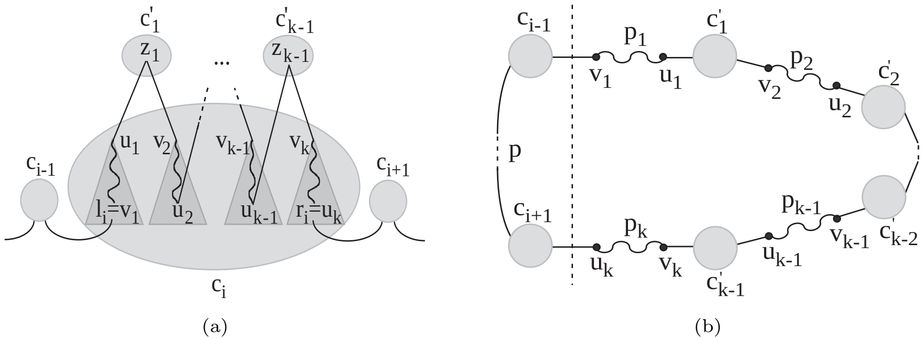

Fig. 3. (a) Broken pair l_i, r_i in cycle $\mathcal{C}$; (b) cycle $\hat{\mathcal{C}}$ in the proof of Theorem 1

by c_i. Observe that it could be $u_j = v_j$ for some j, but vertices with different indexes always belong to different connected components of $S(c_i)$.

It is clear that $v_1 = l_i$ and $u_k = r_i$. Besides, it is easy to see that the pairs u_j, v_{j+1}, for $1 \le j \le k-1$, are minimum distance broken pairs of cluster c_i and, by hypothesis, we know that $dist_T(u_j, v_{j+1}) = 2$. Let us call z_j the unique vertex of T in the path between u_j and v_{j+1} and let c'_j be the cluster of W that covers z_j. The configuration is illustrated in Figure 3(a).

We now change view W into W' by expanding cluster c_i at the singleton level, i.e., by substituting c_i with the set of singleton clusters corresponding to the vertices in $S(c_i)$. In the following we prove that we are able to exhibit in W' a simple cycle $\mathcal{C}'$. Let us first consider the cycle $\hat{\mathcal{C}}$ shown in Figure 3(b). $\hat{\mathcal{C}}$ clearly exists in W', being obtained from $\mathcal{C}$ by unrolling $path_T(l_i, r_i)$ (compare vertices and clusters involved in Figure 3(a) and in Figure 3(b), respectively). However, $\hat{\mathcal{C}}$ is not necessarily simple. W.l.o.g. we can assume that $c'_j \neq c'_s$ $\forall j, s \in [1, k-1]$, $j \neq s$. We can always reduce to this situation as follows: while $\hat{\mathcal{C}}$ contains a pair j, s such that $1 \le j < s \le k-1$ and $c'_j = c'_s$, substitute the path $\langle c'_j, \ldots, c'_s \rangle$ with the path $\langle c'_j \rangle$. After this operation all the c'_j are distinct. If $\forall j \in [1, k-1]$ $c'_j \notin \mathcal{C}$, $\mathcal{C}' = \hat{\mathcal{C}}$ is a simple cycle. Otherwise $\exists j \in [1, k-1]$ such that $c'_j \in \mathcal{C}$ and two cases may happen:

- $c'_1 = c_{i-1}$: change $\hat{\mathcal{C}}$ by replacing the non-simple path $\langle c_{i-1}, p_1, c'_1, p_2 \rangle$ with the simple path $\langle c_{i-1}, p_2 \rangle$. If $k = 1$ or $\forall j \in [2, k-1]$ $c'_j \notin \mathcal{C}$, then the modified $\hat{\mathcal{C}}$ is a simple cycle and we can choose $\mathcal{C}' = \hat{\mathcal{C}}$. Otherwise, let s be the smallest value in $[2, k-1]$ such that $c'_s \in \mathcal{C}$, i.e., $c'_s = c_t$ for some $t \in [1, h]$, $t \neq i$. Moving clockwise on $\hat{\mathcal{C}}$, we find the simple cycle $\mathcal{C}' = \langle c_{i-1}, p_2, c'_2, \ldots, p_s, c'_s = c_t \sim c_{i-1} \rangle$, where $\sim$ indicates a subpath of p.

- $c'_1 \neq c_{i-1}$: let s be the smallest value in $[1, k-1]$ such that $c'_s \in \mathcal{C}$, i.e., $c'_s = c_t$ for some $t \in [1, h]$, $t \neq i$. If $s = 1$ then $\mathcal{C}' = \langle c_{i-1}, p_1, c'_1 = c_t \sim c_{i-1} \rangle$ is a simple cycle in W'. Note that the length of the path $\langle c_t \sim c_{i-1} \rangle$ is ≥ 1, because $c_t = c'_1 \neq c_{i-1}$. If $s>1$ then let $\mathcal{C}' = \langle c_{i-1}, p_1, c'_1, p_2, c'_2, \ldots, p_s, c'_s = c_t \sim c_{i-1} \rangle$.

In this case it could be $c_t = c_{i-1}$; however, the fact that the path from c_{i-1} to c'_s has length ≥ 3 guarantees $\mathcal{C}'$ to be a cycle of W'. By construction we also know that $\mathcal{C}'$ is simple.

In any case we are therefore able to find a view W' containing more singletons than W and to exhibit a simple cycle $\mathcal{C}'$ in W'. Iterating this reasoning, we obtain either a cycle with no broken pairs, which is absurd due to Lemma 2, or a cyclic view containing only singleton clusters, i.e., W_l. This is also a contradiction because W_l is acyclic being equal to T.

Theorem 1 implies that the validity of a hierarchy tree can be checked in polynomial time. We remark that this does not immediately follow from Definition 2, since the number of views in a hierarchy tree may be exponential.

References

1. Buchsbaum, A.L., and Westbrook, J.R.: *Maintaining hierarchical graph views*, Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA'00), 566-575, 2000.
2. Di Battista, G., Eades, P., Tamassia, R., and Tollis, I.: *Graph Drawing: Algorithms for the visualization of graphs*, Prentice Hall, Upper Saddle River, NJ, 1999.
3. Duncan, C.A., Goodrich, M.T., and Kobourov, S.G.: *Balanced aspect ratio trees and their use for drawing large graphs*, Journal of Graph Algorithms and Applications, 4(3), 19-46, 2000.
4. Eades, P., and Feng, Q.: *Multilevel visualization of clustered graphs*, Proc. 4th Symposium on Graph Drawing (GD'96), LNCS 1190, 101-112, 1997.
5. Feng, Q.W., Cohen, R.F., and Eades, P.: *Planarity for clustered graphs*, Proc. European Symposium on Algorithms (ESA'95), LNCS 979, 213-226, 1995.
6. Finocchi, I., and Petreschi, R.: *Hierarchical clustering of trees: algorithms and experiments*, Proc. 3rd Workshop on Algorithm Engineering and Experiments (ALENEX'01), January 2001.