Extracting and Visualising Tree-like Structures from Concept Lattices

Cassio Melo[1], Bénédicte Le-Grand[2], Marie-Aude Aufaure[1] and Anastasia Bezerianos[1]
[1]École Centrale Paris – MAS Laboratoire, [2]Laboratoire d'Informatique 6 – LIP6
{cassio.melo, marie-aude.aufaure, anastasia.bezerianos}@ecp.fr, benedicte.le-grand@lip6.fr

Abstract

Traditional software in Formal Concept Analysis makes little use of visualization techniques, producing poorly readable concept lattice representations when the number of concepts exceeds a few dozens. This is problematic as the number of concepts in such lattices grows significantly with the size of the data and the number of its dimensions. In this work we propose several methods to enhance the readability of concept lattices firstly though colouring and distortion techniques, and secondly by extracting and visualizing trees derived from concept lattice structures. These contributions represent an important step in the visual analysis of conceptual structures, as domain experts may visually explore larger datasets that traditional visualizations of concept lattice cannot represent effectively.

Keywords--- **Concept Lattices, Formal Concept Analysis, Tree Extraction**.

1. Introduction

The vast amount of data generated over the last decades has brought new challenges to the analytics science. Visual data analysis and knowledge representation employ methods such as Formal Concept Analysis (FCA) in order to identify groupings of patterns from the analysis process [26]. FCA provides an intuitive understanding of generalization and specialization relationships among objects and their attributes in a structure known as a *concept lattice*. A concept lattice is traditionally represented by a *Hasse diagram* illustrating the groupings of objects described by common attributes. A Hasse diagram is a graph where concepts appear as vertices on the plane connected by line segments or curves. The layout of the partially ordered set may be seen as a layered diagram [2]. Lattices visualization becomes a problem as the number of clusters grows significantly with the number of objects and attributes. Interpreting the lattice through a direct visualization of the line diagram rapidly becomes impossible and more synthetic representations are needed.

In this work we propose alternatives to the traditional lattice representation, firstly by enhancing the readability of concept lattices though colouring and distortion techniques; secondly by extracting and visualizing trees derived from the lattices structure. The tree extraction from the original lattice has some unique advantages: it eliminates all edges crossing and the resulting hierarchy is also easier to interpret and to represent. Moreover, this representation still provides an overview of the dataset, highlighting significant properties of the lattice. In order to extract trees from lattices, we define a set of parent concept selection criteria, including the *stability* and *support* indexes [1,4] provided by FCA literature, *confidence* index as well as topological features of the lattice.

The paper is organized as follows. Section 2 provides background on lattice representations; Section 3 proposes a set of criteria for transforming concept lattices into trees; Section 4 discusses colouring and distortion techniques for enhancing interpretations of lattices. Section 5 presents instantiations of the suggested criteria and visualizations in the biology domain, followed by a discussion in section 6. Section 7 finally concludes and presents perspectives for future work.

2. Visual Representation of Concept Lattices

As mentioned above, FCA analysis produces lattices, usually represented as layered directed acyclic graph graphs, named Hasse diagrams, that illustrate the groupings of objects described by common attributes. Hasse diagrams display the *partially ordered sets* (*posets*) between concepts in a hierarchical fashion, where each concept may have several parent concepts. In the following example about animal's features, the formal context in table 1 generated the concept lattice illustrated in figure 1. The partial order among concepts of the lattice is materialized through the generalization and specialization relationships: for instance Concept 4 (representing the set of *flying birds,* containing *Finch* and *Eagle* objects), is more specific than Concept 1 (which contains all *birds* –flying or not-), and thus contains a smaller number of objects (Concept 1 has an extra one, the *ostrich*). This partial order provides different levels of abstraction and native navigation links from a given concept.

1550-6037/11 $26.00 © 2011 IEEE
DOI 10.1109/IV.2011.46

As mentioned earlier, such diagrams are usually layered graphs, where concept vertices are assigned to horizontal layers according of the number of common attributes, and are ordered within each layer to reduce edge crossings. FCA lattices in particular suffer from considerable edge crossings, especially if the number of concepts exceeds a few dozen as is the case in more real word applications [13], which leads to reduced graph readability and aesthetics [3].

To reduce the complexity of lattices, simplified diagrams can be produced by condensing or clustering concepts according to similarity [4]. Visualisations can also be restricted to portions of the data [5], and concept size reduction is possible by incorporating conditions into the data mining process [6]. Finally, conceptual measures can be applied to identify the most relevant concepts and filter outliers [7].

To deal specifically with the visual complexity of *Hasse* diagrams, several approaches allow users to dynamically explore and reveal specific parts of the diagram, using visual query languages [8-10]. However these techniques do not provide a clear view of the entire lattice.

Other FCA visualization approaches map the distances between concepts to visual variables, in order to highlight patterns. For example in [11] similar concepts are represented as similarly coloured pixels placed in the 2D space along a *Peano-Hilbert curve,* so that similar concepts are placed close from one another. Nevertheless in these representations detailed relationships between concepts are lost. Finally, systems often provide users with hybrid/combined lattice visualization, e.g. showing both a general *Hasse* diagram and a tag cloud for representing the neighbours of a specific concept (for a review see [12]).

Our approach consists in representing lattices not as *Hasse* diagrams, but as trees. We use different criteria to extract trees from lattices, and visualize the resulting trees. Trees are inherently simpler hierarchical structures than *Hasse* diagrams and due to their applicability in many domains, there is a plethora of tree representations. These include: indented outline trees, sometimes called a “tree list” (common in file browsers such as windows Explorer), traditional layered node-link diagrams in 2D or 3D (e.g. *ConeTrees* [14]), spatially transformed tree diagrams (e.g. *Radial* [15]) as well as several space optimization (*Space Optimized trees* [16]) and space-filling tree visualization techniques (e.g. *TreeMaps* [17]).

Table 1. A formal context of animals.

Animal	Preying	Mammal	Flying	Bird
Lion	X	X		
Finch			X	X
Eagle	X		X	X
Hare		X		
Ostrich				X

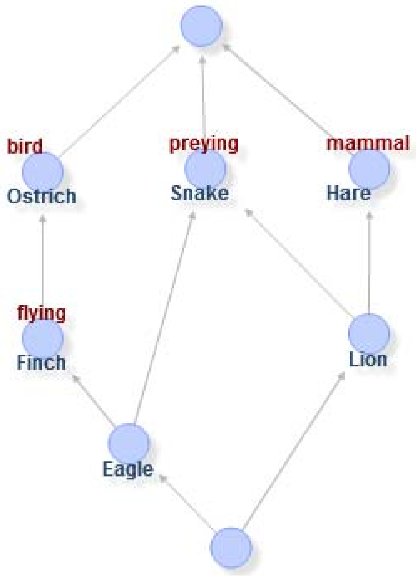


Figure 1. Generated concept lattice for the animal’s context in table 1.

3. Tree Extraction from Concept Lattices

Trees are a common and easily understandable visual representation. We consider them as a visualization alternative to large cluttered concept lattices, which preserves all lattice entities and some of its structure. In order for a tree visualization to be an effective alternative to a lattice, the extraction of the tree from the lattice needs to preserve the most essential features of the original structure.

The present approach consists in extracting a tree from a concept lattice by choosing one single parent concept for each concept of the lattice. We start from the most specific concepts i.e. the parent concepts of the lower bound of the lattice, at the bottom of the *Hasse* diagram (concepts 5 and 6 in Figure 1) and select a single parent concept for each of them, and reproduce this recursively. Choosing a single parent concept at each step leads to an information loss. Our goal is to minimize this loss by selecting parents using the most relevant criteria according to the kind of analysis performed by the analyst. Before proceeding, we briefly recall the FCA terminology [18]. Given a (*formal*) *context* $K = (G,M, I)$, where G is called a set of *objects or extent*, M is called a set of *attributes or intent*, and the binary relation $I \subseteq G \times M$ specifies which objects have which attributes, the *derivation* operators (·)’ are defined for $A \subseteq G$ and $B \subseteq M$:

$$A' = \{m \in M \mid \forall g \in A : g\mathrm{I}m\};$$
$$B' = \{g \in G \mid \forall m \in B : g\mathrm{I}m\}.$$

In the following sections we consider various strategies for selecting parent concepts, including the *stability* and *support* indexes from FCA literature, *confidence*, as well as topological features of the lattice.

3.1. Parent Selection based on the highest Stability or Support

The stability index measures the proportion of subsets of *objects* of a given concept whose derivation is equal to the *intent* of this concept [1]. In other words, the *stability* indicates the probability of preserving a concept *intent* while removing some objects of its *extent*. We recall the definition of stability:

Definition 1. *Let K = (G,M,I) be a formal context and (A,B) be a formal concept of K. Card is a cardinality function. The stability index of (A,B) is defined as:*

$$\sigma(A,B) = \frac{Card(\{C \subseteq A \mid C' = B\})}{2^{Card(A)}}$$

Using the context in table 1 as an example, we calculate the *stability* for concepts 2 and 4 in order to select a parent for concept 5 (0.25 and 0.5 respectively); we keep the one with highest *stability*, in this case we therefore remove the edge between concepts 2 and 5. The idea behind the choice of the parent concept with the highest *stability* is that we expect to keep parent concept's meaning even if some of the objects or attributes are removed.

On the other hand, the support measure is the relation between the *intent* closure and the number of total of objects [4]:

Definition 2. *Let B ⊂ M. The support count of the attribute set B in K is:*

$$\varphi(B) = \frac{Card(B')}{Card(G)}$$

The use of support as parent selection criteria may lead to trees containing concepts that have fewer specialization levels since in general, generic concepts have higher support values than their most specific counterparts [4]. Concept *stability* and *support* measures have been widely used in FCA and their combination has been promising [1] in reducting the lattice.

3.2. Parent Selection Based on Shared Attributes and Objects

This approach relies on clustering parent and child concepts which share most of their attributes or objects. Parent and child having a great number of attributes in common are supposed to be grouped together following the principle of similarity clustering and local predictability [19]. Its definition is:

Definition 3. *Let ConceptParent (A,B) be such that A ⊂ G and B ⊂ M. Let ConceptChild (C,D) be C ⊂ G and D ⊂ M. The shared attribute index of an edge E (C,D)•(A,B):*

$$\phi(E) = \frac{Card(B \cap D)}{Card(M)}$$

In the same animal's context in table 1, we have potential parent concepts 2 and 4 sharing the same number of objects with concept 5, but concept 4 has more attributes in common with 5, so it should be chosen as the unique parent of concept 5.

3.3. Parent Selection Based on Confidence

The *confidence* value of a concept estimates how likely an object which has an attribute set A, also has an attribute set C [18]. In other words, it tries to measure how strong the *implication* of the parent attributes in the child objects is. For instance, considering the formal context in table 1, what is the probability of a given object that is {*Bird, Flying*} to be also {*Bird, Flying, Preying*}? The following paragraph formalizes its definition.

Definition 4. *Let ConceptParent (A,B) be such that A ⊂ G and B ⊂ M. Let ConceptChild (C,D) be C ⊂ G and D ⊂ M. The confidence of an edge E (C,D)•(A,B):*

$$\delta(E) = \frac{Card(C)}{Card(A)}$$

An advantage of this method is its consistency with the interpretation of concept lattices. Taking our animals context as example, there is a 50% probability that an animal that is a *flying bird* is also a *flying* and *preying bird*. By contrast, an animal that is *preying* has only 33% of chance to be also a *flying bird*.

3.4. Tree Transformation Based on the Minimum Spanning Tree

This topological-driven approach seeks to choose the tree configuration on the graph which has the shortest paths among all concepts. A detailed definition of the algorithm can be found in [20]. In this case, we choose the parent concept which minimizes the number of steps needed to reach the top of the lattice, and therefore most generic parents (closer to the top) will be preferred rather than the specialized ones. This is the only proposed strategy that takes into account the topology of a lattice, and it yields concepts with similar depth (*specialization*) levels.

4. Using extraction criteria to enhance Lattice and Tree Interpretation through Drawing, Sizing and Shaping

Common graph drawing techniques include the assignment of different colours, shapes and sizes to nodes and edges, according to different dimensions or properties. This approach is underused in traditional lattice visualizations, where the main visual variable used is node/link colour to reflect user selections or node size to indicate the immediate presence of an extent or intent as displayed in *ConExp*[1].

In our work we use these as well as other visual variables in a *Hasse* diagram to represent possible tree extraction criteria. This provides several benefits to lattice and extracted tree understanding. First, it enables users to rapidly associate the dimension/criteria in question (e.g. *stability*, *support* in Figure 2 with concepts, thus justifying the choices made during the tree extraction process. Second, visualizing different extraction criteria using various visual variables, allows users to compare these criteria in order to choose the one that better fits their needs. Third, irrespective of the tree extraction process, matching visual attributes to concept attributes establishes a benchmark/comparison among concepts, making it possible to compare at a glance different concepts, even if they do not have a link in common, as well as gain insights on the whole lattice itself. Finally, prominent features of the lattice like specialization and generalization can be better understood: for instance the power of *implications* of different concepts can be rendered by edge thickness. The concept node itself can be a visual metaphor for the intent and extent. In the example of figure 2, a pie chart replaces the traditional box representation to depict the proportion of *objects* (blue) and *attributes* (yellow). In this way users can be guided in understanding and choosing criteria for extracting trees to simplify the lattice representation.

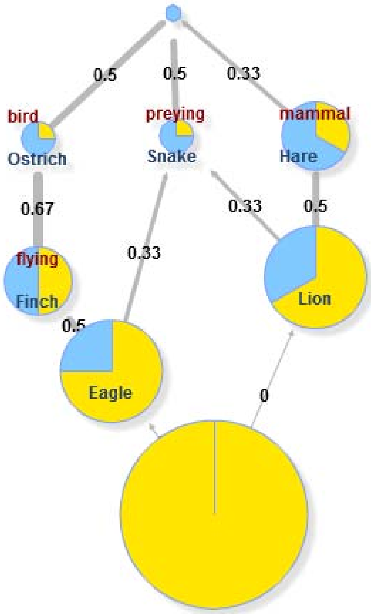


Figure 2. Animal lattice with nodes as pie charts sized by stability, and edge thickness by confidence. Pie charts indicate the ratio intent/extent of the concept.

5. A Qualitative Analysis of the Proposed Parent Selection Criteria

In this section we discuss a case study of a concept lattice to qualitatively examine the nature of the trees resulting from different criteria. The techniques for lattice transformation and drawing were implemented in a visual analytics tool called *CUBIST Analytics* and applied to a dataset[2] containing 8 animals and 9 attributes which produced a lattice with 19 concepts (figure 3). Each of the measures proposed revealed particular aspects on the analysis of a lattice, illustrated in table 2.

Table 2 a) shows the tree generated with *stability* as parent selection criterion. In practice, it resulted in a tree with very stable concepts more likely to retain their subsequent children. For instance, the concept {*lives in land*} was the preferred parent of the concept that holds our notion for amphibians: {*lives on land, lives in water*} because it is more stable than its counterparts.

The measure of *shared objects* was the criterion that generated the tree in table 2 b). Parent concepts sharing most objects with child concept were the preferred candidates. As an example, the concept {*lives on land*} shares more objects with {*lives on land, needs chlorophyll*} than concept {*needs chlorophyll*} does, therefore it was the chosen parent in this case.

Table 2 c) the tree was generated from *confidence* criterion, therefore children nodes are associated with the parent with which the relationship of confidence is the highest among the candidates. As a result, the relation {*can move, has limbs*} has a stronger implication in {*lives on land*} than {*lives on land*} has for {*can move, has limbs*}, for example.

Table 2 d) depicts the tree generated by the minimum spanning tree criterion. In this configuration, concepts are arranged in a way that they are closer semantically from the rest of lattice and hence it features a more symmetric structure in comparison with previous approaches.

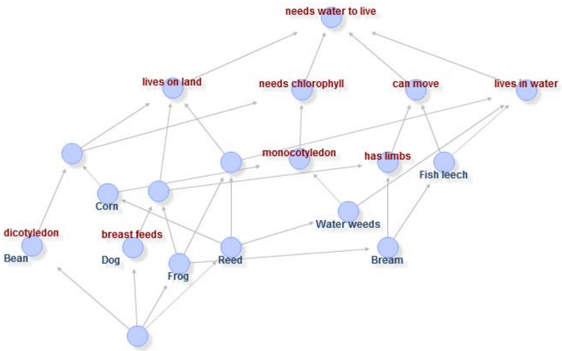


Figure 3. Concept lattice of the biology domain.

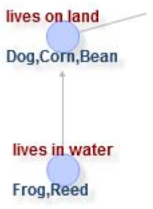
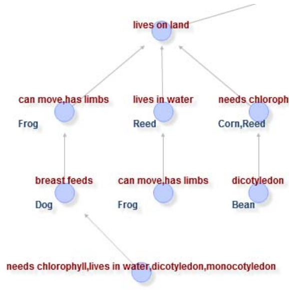
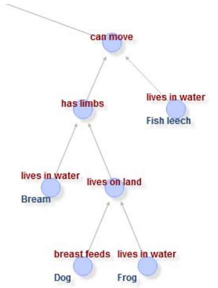
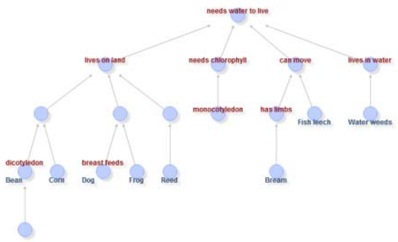
[1] *ConceptExplorer*. http://conexp.sourceforge.net/

[2] The “Needs water to live” dataset is available at http://www.upriss.org.uk/fca/examples.html

6. Discussion

Some may argue that due to the tree construction, the present approach breaks the original lattice meaning, and therefore subsequent mathematical models based on this structure. It is noteworthy to observe however, that only the links in the lattice graph structure are removed and the lattice structure remains semantically valid, since there is no need to take out the attributes or objects that concepts have in common with their parents.

Table 2. Trees generated from the lattice in figure 4 for each one of the proposed measure.

	Example
a) Stability	 lives on land Dog,Corn,Bean lives in water Frog,Reed
b) Shared objects	 lives on land can move,has limbs Frog lives in water Reed needs chlorophyll Corn,Reed breast feeds Dog can move,has limbs Frog dicotyledon Bean needs chlorophyll,lives in water,dicotyledon,monocotyledon
c) Confidence	 can move has limbs lives in water Fish leech lives in water Bream lives on land breast feeds Dog lives in water Frog
d) MST	 needs water to live lives on land needs chlorophyll can move lives in water monocotyledon has limbs Fish leech Water weeds dicotyledon Bean Corn breast feeds Dog Frog Reed Bream

The choice of parent selection criteria for tree transformation corresponds to a classification problem to some extent. Deciding if a Lion is more "mammal" than it is "preying" it's not always straightforward, hence we rely on the measures that attempt to keep the context semantics when looking at the entire concept lattice. For instance, if we have more objects described by mammal which are "closer" to Lion than other concepts, then it may reasonable to be chosen as its parent. As general recommendations, one should use the criteria that best fits to their analysis task (table 3).

In addition to the tree-extraction strategies, the use of colours, size, shaping and thickness for both nodes and edges in the original lattice to represent the criteria metrics (such as *stability*, *support*, *specialization* or *implication*) can enhance the interpretation of a concept lattice, and aid users in their choice and interpretation of the created trees.

The labelling strategy for identifying concepts should be taken into account as well. Merely placing attributes and objects names on concepts may be cumbersome for large lattice analysis (used in most FCA visualizations). In this case, it is recommended to represent the concept's *intent* and *extent* with visual metaphors like the *pie chart* shown in figure 2.

Conclusions and Future Work

Traditional software in FCA makes little use of visualization techniques, producing poorly readable lattice graphs when the number of concepts exceeds a few dozens. In this work we have presented a transformation approach to extract trees from concept lattices, attempting to minimize both semantic and conceptual loss in favour of readability and interpretation. We have also presented ways to visually show the extraction criteria in the original lattice. This is an important step in the visual analysis of conceptual structures, as the resulting tree structures are visually easier to understand than cluttered lattice graphs. Domain experts can thus visually explore larger datasets that traditional visualizations of concept lattice cannot represent effectively. Each of the tree construction measures proposed in our work provides particular insights valuable to different analysis tasks, identified in our paper as recommendations.

In the future we plan to combine two or more criteria for parent selection with other lattice reduction techniques (e.g. *icebergs* lattices [4]). We also plan to conduct user experiments to understand when users want to have full lattice views *vs*. tree views, which metrics for creating trees are of most interest to them and under which circumstances, and assess if our visual indications allow users to understand the extraction tree process.

Acknowledgements

This work is partly funded by the CUBIST project ("Combining and Uniting Business Intelligence with Semantic Technologies"), funded by the European Commission's 7th Framework Programme of ICT, under topic 4.3: Intelligent Information Management.'

Table 3. General guidelines on the usage of the proposed metrics.

Criteria	Description	Rationale	Suitable for
Stability	It measures how likely a concept is to change if some of their attributes or objects are removed.	Stable concepts are less impacted by noise and usually represent strong correlation with real world entities (e.g.: a concept that encapsulates our notion of “mammal”).	Observing real world analogies
Support	It measures the frequency of the concept itemset.	Frequent concepts are usually generic concepts since they aggregate a larger number of objects than the specialized ones.	Frequent pattern analysis
Shared objects / attributes	It represents the degree of similarity between parent and child nodes.	Concepts that share most attributes or objects should be linked together because they are similar.	Similarity analysis
Confidence	It measures how strong the implication is between a parent concept in a child concept.	Implication is one of the desired interpretation of a concept lattice.	Confidence analysis
MST	It extracts a minimum spanning tree from the concept lattice.	It looks at concepts that have some degree of similarity with all other concepts.	Topological analysis

References

[1] Kuznetsov, S.O.: Stability as an estimate of the degree of substantiation of hypotheses derived on the basis of operational similarity. Nauchn. Tekh. Inf., Ser.2 (Automat. Document. Math. Linguist.) 12 (1990) 21–29

[2] Di Battista, G.; Tamassia, R. (1988), "Algorithms for plane representation of acyclic digraphs", Theoretical Computer Science 61: 175–178.

[3] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. Information Visualization, 1(2):103–110, 2002.

[4] Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., and Lakhal, L. Computing iceberg concept lattices with Titanic. In Data & Knowledge Engineering, Volume 42, Issue 2, pp. 189-222, 2002.

[5] Ducrou, J., Eklund, P., and Wilson, T. An Intelligent User Interface for Browsing and Searching MPEG-7 Images Using Concept Lattices. In S. Ben Yahiaet al. (Eds.): CLA 2006, LNAI 4923, pp. 1–21, Springer-Verlag Berlin Heidelberg 2008.

[6] Zaki, M.J., Hsiao, C-J. Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure. In IEEE Transactions on Knowledge and Data Mining, Vol. 17, No. 4, IEE Computer Soc., 2005.

[7] Le Grand, B., Soto, M., Aufaure, M.-A. (2009) “Conceptual and Spatial Footprints for Complex systems Analysis: Application to the Semantic Web”, in 20th International Conference on Database and Expert Systems Applications 2009, pp.114-127.

[8] Blau, H., Immerman, N.,and Jensen, D.. A Visual Language for Querying and Updating Graphs. University of Massachusetts Amherst, Computer Science Department Tech: Report 2002-037. 2002.

[9] Cruz, I. F., Mendelzon, A. O., and Wood, P. T.. A Graphical Query Language Supporting Recursion. In Proc. of the Association for Computing Machinery Special Interest Group on Management of Data, pages 323–330. ACM Press, May 1987.

[10] Consens, M., and Mendelzon, A. Hy+: a Hygraph-based query and visualization system. SIGMOD Record, 22(2):511–516, 1993.

[11] Michel Soto, Benedicte Le Grand, Marie-Aude Aufaure, "Spatial Visualisation of Conceptual Data," International Conference Information Visualisation, pp. 57-61, 2009.

[12] Eklund, Peter, Villerd, Jean. A Survey of Hybrid Representations of Concept Lattices in Conceptual Knowledge Processing Formal Concept Analysis. Lecture Notes in Computer Science 2010, Springer Berlin/Heidelberg, pp. 296- 311

[13] C. Roth, S. Obiedkov, D. G. Kourie. ″Towards Concise Representation for Taxonomies of Epistemic Communities", CLA 4th Intl Conf on Concept Lattices and their Applications. 2006.

[14] George Robertson, Jock D. Mackinlay, Stuart Card. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In Proceedings of the ACM CHI 91 Human Factors in Computing Systems Conference, pages 189--194, Association for Computing Machinery. 1991.

[15] P. D. Eades. Drawing free trees. Bulletin of the Institute for Combinatorics and its Applications, 5:10–36, 1992.

[16] Quang Vinh Nguyen; Mao Lin Huang. A space-optimized tree visualization, Information Visualization, 2002. INFOVIS 2002, pp. 85- 92, 2002

[17] B. Johnson, B. Shneiderman: Tree maps: A Space-Filling Approach to Visualization of Hierarchical Information Structures. IEEE Visualization 1991: 284-291

[18] Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999)

[19] Hannan, T., Pogel, A.: Spring-based lattice drawing highlighting conceptual similarity. In: Proceedings of the International Conference on Formal Concept Analysis, ICFCA 2006, Berlin. LNCS, vol. 3974, pp. 264–279. Springer, Heidelberg (2006)

[20] Gallager, R.G., Humblet, P.A., and Spira, P.M. 1983. A Distributed Algorithm for Minimum-Weight Spanning Trees. ACM Trans. Program. Lang. Syst. 5, 1 (Jan. 1983), 66-77.