Journal of Universal Computer Science, vol. 10, no. 8 (2004), 967-984
submitted: 22/3/04, accepted: 28/6/04, appeared: 28/8/04 © J.UCS

Modelling Lexical Databases with Formal Concept Analysis

Uta Priss
(School of Computing, Napier University, Edinburgh, UK
u.priss@napier.ac.uk)
L. John Old
(School of Computing, Napier University, Edinburgh, UK
j.old@napier.ac.uk)

Abstract: This paper provides guidelines and examples for visualising lexical relations using Formal Concept Analysis. Relations in lexical databases often form trees, imperfect trees or poly-hierarchies which can be embedded into concept lattices. Many-to-many relations can be represented as concept lattices where the values from one domain are used as the formal objects and the values of the other domain as formal attributes. This paper further discusses algorithms for selecting meaningful subsets of lexical databases, the representation of complex relational structures in lexical databases and the use of lattices as basemaps for other lexical relations.
Key Words: Formal Concept Analysis, lexical databases, semantic relations
Category: I 2.4

1 Introduction

This paper is concerned with modelling lexical databases so that they can be explored in visualisations. Modern lexical databases are available in electronic formats and thus accessible for computerised visualisations. But it can be challenging to automatically derive visualisations because these can be large and complex. An interesting question is how to find a suitable visualisation for a given task. In the second section of this paper, we discuss why this is a difficult question and why there are no formal solutions for this question.

A common practice in the area of interface design is to provide guidelines (for example, [Nielsen and Tahir 2001]) and heuristics. These can be combined with classifications of tasks and solutions. (This is similar to how in psychology, classifications of data have been proposed to facilitate the selection of experiments [Coombs 1964].) Shneiderman [Shneiderman 1996] combines the use of guidelines with a classification of visualisation tasks and of types of visual data. Once a user can classify a problem as being of a certain type, the user then has a choice of guidelines which apply to the type of problem.

In this paper, we attempt to answer the question of finding suitable representations for the application domain of lexical databases by providing a set of guidelines for visualisations. Our analysis differs in three ways from Shneiderman's: First, because of the selection of a specific domain, our guidelines are more specific than Shneiderman's guidelines. Second, we discuss the impact of using basic structures from relational databases because lexical databases are often implemented in that manner. Third, we consider a specific data analysis theory, Formal Concept Analysis [Ganter and Wille 1999], and discuss its use for the different types of visualisation tasks.

The primary lexical database used in this paper is Roget's International Thesaurus [Roget 1962] but in principle the guidelines proposed in this paper also apply to other databases such as library thesauri and formal ontologies. We are using the following relational database versions of Roget's Thesaurus:

- RIT11: a relational database version of Roget's Thesaurus [Roget 1911] which was compiled by the second author based on the text version published by the Gutenberg Project.
- RIT3: a relational database version of the 3rd edition of Roget's International Thesaurus [Roget 1962] which was converted to electronic form by [Sedelow and Sedelow 1993] under NSF grants in the 1980s and later converted to relational format by the second author.

If referring to structures which occur in both electronic versions (but may or may not occur in the printed version) we use the term "ROGET".

2 A semiotic background for visualisation tasks

Visualisations can be considered with respect to two contrasting semiotic theories. Bertin's semiology of graphics [Bertin 1967] describes how formal graphics, such as cartographic maps and scientific visualisations, can be constructed from seven visual variables, such as form, orientation and position. This construction is based on grammar-like rules similar to how a sentence can be constructed from words and grammatical rules. A contrasting theory is Gibson's ecological theory of perception [Gibson 1979]. According to Gibson's theory, images are not constructed from formal primitives using a grammar, but instead are vague and polysemic. The two theories represent two major trends in semiotics: Bertin's theory instantiates Saussure's semiology; whereas Gibson's theory is related to Peirce's semiotics [Peirce 1894] (according to [Keeler 2002]).

The difference between the two theories can be stated in terms of formal and associative structures [Priss 2002]. This distinction has been discovered in many fields (for example, [Pinker 1991] and [McGarry et al. 1999]) and also relates to Gibson's notion of (verbal) description and (visual) depiction. Formal structures are rule-based, logical, precisely defined, and, using Peirce's terminology, predominantly "symbolic". Associative structures depend on external structures of a domain and a perceiving agent and can be represented iconically, indexically or symbolically [Priss 2002]. In these terms, Bertin's theory focuses on formal aspects, whereas Gibson's theory focuses on associative aspects.

When formal structures are represented graphically, they induce associative structures (indexical or iconical) in a user's mind. For example, cartographic maps have an unambiguous formal meaning, which is a direct sum of the employed formal symbols of the map. A reader of a cartographic map, however, can discover associative information in the map that is not explicitly encoded but emerges from the spatial relations and the reader's knowledge of the domain. Vagueness of associative representations is not a shortcoming but an advantage because it provides an opportunity for the discovery of tacit knowledge, emergent structures (see [Clark 1997] and [Priss 2001]) and the continuing evolution of meaning, which Peirce calls "semiosis" [Keeler 2002]. The associative nature of visualisations contributes to what types of visualisation software users select. For example, people often find it easier to read bar graphs instead of tables with values; and prefer graphical interfaces instead of purely text-based interfaces.

The difference between associative and formal structures can also be stated in terms of "information content". While formal content can be identified as a mathematical representation of a visualisation, associative content is more difficult to grasp. Formally equivalent representations can have different associations. For example, if the content of a graph is represented in XML then it is quite likely that the graph is more human-readable than the XML representation. While formally equivalent, users will most likely value both representations differently.

An interesting question arises as to what the (associatively) best representation of formal content is for a given purpose. Or, because associative information is not easily quantifiable, the question might be better stated as "what are suitable representations for certain purposes". One can argue that this is a question of "semantics" because it asks what representations mean to users. But this is definitely not "formal semantics" because only formal content can be covered in a formal model. Guidelines and heuristics provide a means for encoding associative knowledge (expert or tacit knowledge). The solutions proposed in this paper are associative in two ways: because of the contextual embedding (purpose, domain, data structures) and because of the vagueness of the guidelines.

3 Types of visualisation problems

Shneiderman distinguishes seven data types for visualisations and seven tasks [Shneiderman 1996]. The types are 1-, 2-, 3- and multi-dimensional data, trees, networks and temporal data. The tasks are overview, zoom, filter, details-on-demand, relate, history and extract. For lexical databases, the data types of "tree" and "network" are most relevant because lexical data are not usually

cardinality	domain	examples	FCA
one-to-many	bipartite	partition, classification	anti-chain
one-to-many	autorelation	directed graph, tree	add bottom node
many-to-many	bipartite	bipartite graph	objects and attributes
many-to-many	autorelation	graph, poly-hierarchy	concepts, embedding

Figure 1: Modelling relations in relational databases

recorded with temporal or spatial coordinates. A temporal comparison of different versions of lexical databases at different times does not yield temporal data in Shneiderman's sense because the data are not real-valued and do not have start points, intervals, overlaps and so on. It is possible to represent non spatial-temporal data using spatial metaphors (cf. Information Cartography [Old 2002]) but that shall not be discussed in this paper.

We argue that in addition to Shneiderman's two aspects, data type and task, the details of implementations provide a third aspect of visualisations. We assume lexical databases to be stored as relational databases. Even though in theory a lexical database can be considered a formal structure, the details of the implementation add associative features. Although we do not have a formal proof for this, it appears to us that visualisations are driven by what is technically simple and intuitive from an implementation viewpoint. For example, tree hierarchies pose a practical challenge for relational databases, even though from a theoretical view, they are straightforward. Therefore, we believe that visualisation guidelines should start with implementation-related basics of relational databases, such as domain values and cardinality of binary relations (e.g.. one-to-many, many-to-many). We call binary relations whose two domains are identical "autorelations" and those whose two domains are not identical "bipartite relations". We consider only binary relations in this paper, which can be classified into four types [Fig. 1].

We further discuss which types of relations can suitably be represented by Formal Concept Analysis (FCA, [Ganter and Wille 1999]). Formal Concept Analysis provides a means for representing the information contained in a "formal context" (a cross-table or binary relation, [Fig. 6]) as a concept lattice. For the details of FCA, we refer to [Ganter and Wille 1999]. We believe that FCA provides unique opportunities for visualisations but that these have not yet been extensively discussed in a manner which is accessible for information designers. In our terminology, a concept lattice is derived from a bipartite relation between sets of objects and attributes while the graph of the lattice is an autorelation on the set of concepts.

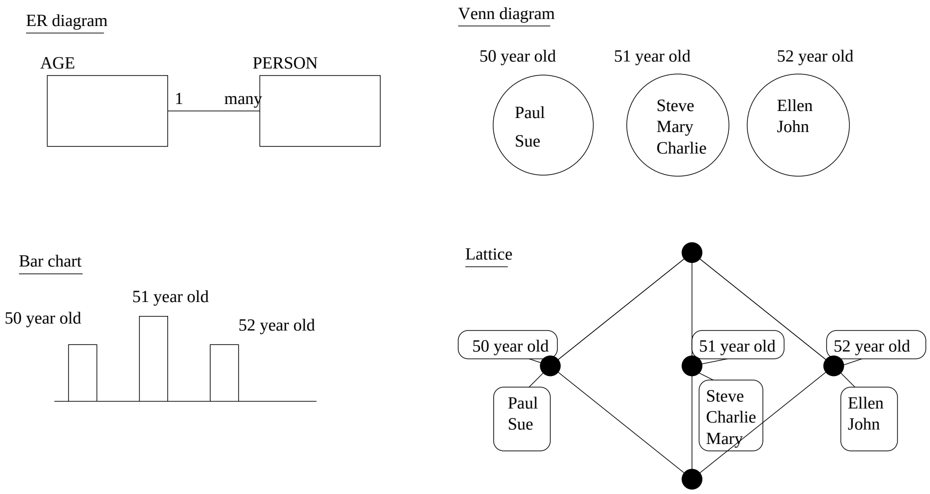

Figure 2: One-to-many bipartite relation

3.1 One-to-many bipartite relations

A one-to-many bipartite relation forms a partition of a set where the values of the relation at the "many" end become the elements of the subsets while the values of the relation at the "one" end become labels of the subsets. The example in [Fig. 2] shows a relation between people and age where people are partitioned according to their age groups. Each age value is used to label an age group, such as "50 year old". The process of assigning individuals to partitions is called "classification".

Partitions can be visualised as Venn diagrams without intersections [Fig. 2, top right] or as labelled lists. If only the number of elements in the partitions are of interest (not their values) then bar charts or pie charts can be used [Fig. 2, bottom left]. In FCA, relations of this type can be recognised by the fact that either each row of a formal context or, dually, each column of a formal context has at most one cross. The lattices of these relations form anti-chains, i.e., apart from the top and bottom node, all concepts are in one level [Fig. 2, bottom right]. It may not be feasible to visualise partitions as concept lattices if the anti-chains become too long. Furthermore, unless it is intended to combine the concept lattice with other lattices, FCA representations may not offer much value in this case.

3.2 Autorelations

Autorelations in a relational database are relations where both columns contain values from the same domain. Autorelations can be visualised as graphs (or networks) consisting of a set of nodes and a set of edges. If the rows of database tables are not selected using "distinct", multiple edges can run between a single pair of nodes. An edge can connect a node to itself. Because database columns have unique headers, by default the graphs are "directed", which means that all edges are arcs (or arrows). Of course, the directionality can be ignored if irrelevant for a certain purpose.

3.3 One-to-many autorelations: Trees

One-to-many autorelations can be visualised as directed graphs where either every node has at most one out-going arc, or dually, at most one incoming arc. If such a graph is acyclic (i.e., one can never get back to the same point after following several arcs) and connected (i.e., all nodes are connected), then such a graph forms a "tree", also called a "tree hierarchy". As explained above, classification or partitioning can be seen as a bipartite one-to-many relation between elements and classes. But the classes themselves are often arranged in a tree, which is an autorelation among classes. Such trees are called "classification systems". Tree hierarchies are important for lexical databases because they represent classification systems which underlie cognitive structures. Roget's thesaurus contains six (or eight depending on the edition) tree hierarchies which are called the "Synopsis of Categories" and are printed at the beginning of the book. If counted from the root (i.e., the top node), trees contain levels which correspond to the length of the path from a node to the root. ROGET contains 6 levels which are indicated by the label "CLASS", Roman numerals, letters, numbers, paragraphs and groups delimited by semicolons, respectively [Fig. 3]. Each word can occur in multiple semicolon groups.

Structures in lexical databases are frequently only "almost" trees. That means that although most nodes have only one outgoing arc, a few nodes may have more than one arc. We define an "imperfect tree (hierarchy)" as a many-to-many autorelation where the percentage of nodes with more than one outgoing arc is low. This is a fuzzy notion because the meaning of "low" is left undefined. Imperfect trees are very common. Hierarchies in WordNet (hypernymy, meronymy, toponymy etc) are imperfect trees [Fellbaum 1998]. A reason for the frequent use of imperfect trees may be that humans like to impose a tree structure on domains, although these domains may contain elements which cannot uniquely be classified. Imperfection of trees even occurs in fields, such as botanical taxonomy [Pullan et al. 2000], if one considers the changes in taxonomic data over time.

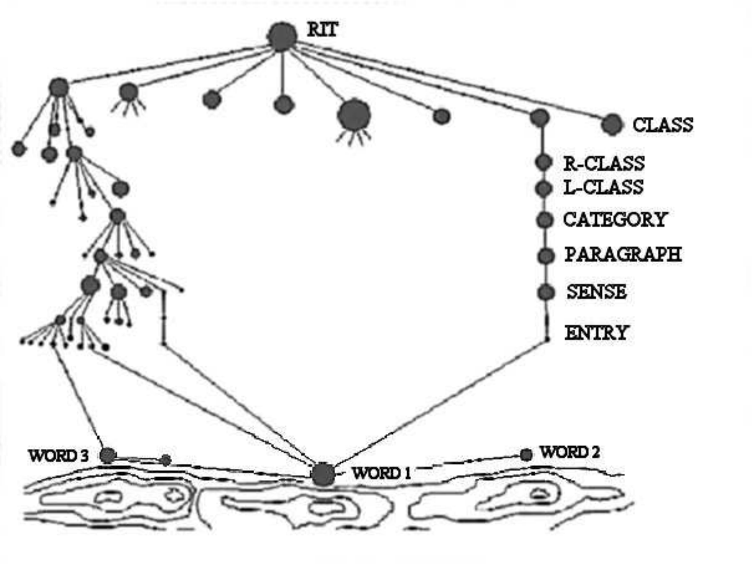

Figure 3: Roget's Thesaurus: tree hierarchy and word/sense relation [Old 2003]

In printed form, tree hierarchies are often represented as lists which have alpha-numerical indicators for levels and/or visual indentation of levels. The left half of [Fig. 4] depicts a part of Roget's Synopsis of Categories [Roget 1962]. It shows the level of sophistication employed in visualising a tree in the printed medium. Each level has a special alpha-numerical indicator: bold-face uppercase CLASS plus letters for the top level, then bold-face Roman numerals (I, II, ...), then bold-face single letters (A, B, ...), and finally numbers (1, 2, ...). In addition to these, the different levels are also distinguished by indentation, except for the top level which is centred. In the electronic medium, trees are often represented as "file system displays" or "directory trees" but also as Venn Diagrams or nested maps [Johnson and Shneiderman 1991]. Because these are interactive (users can expand, collapse and navigate the hierarchy) they may require less complex graphical elements than printed representations.

It should be noted that trees (or imperfect trees) can emerge from a union of different relations and trees can be imperfected by a union of different relations. In ROGET, the full tree hierarchy is identified by [Old 2004] as the union of the structures of the Synopsis with an implicit structure in the main part of the thesaurus [Fig. 4, right half]. The lower two levels of the thesaurus consist of a grouping of terms into Paragraphs and a further subdivision into groups delimited by semicolons (Semicolon Groups). As another example, a Unix files

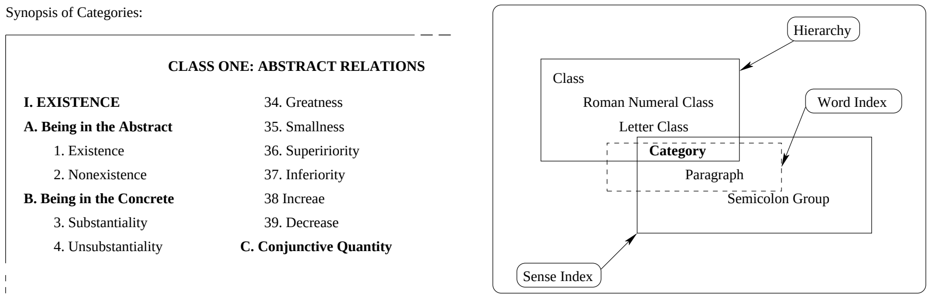

Figure 4: Roget's Thesaurus: Synopsis and Indices

system usually forms a tree of normal directory entries, but forms an imperfect tree consisting of a union of normal directory entries and different types of symbolic links. The Yahoo classification system forms an imperfect tree consisting of subclasses and pointers to subclasses in other parts of the tree. In ROGET, the addition of cross-references creates an imperfect tree.

3.4 Hierarchies and concept lattices

A graph (i.e., autorelation) which is directed and acyclic but not a tree is called a "poly-hierarchy". That means it is a general hierarchy without any restriction as to how many upper or lower neighbours any node can have. Imperfect trees are poly-hierarchies.

Any poly-hierarchy can be embedded into a concept lattice (using the so-called Dedekind-MacNeille completion [Ganter and Wille 1999]). This completion retains all information contained in the original poly-hierarchy. But it also ensures that every set of nodes has a unique lowest upper bound and a unique highest lower bound. Calculating these bounds can in some cases simplify the graph structure and can have computational advantages.

With respect to FCA, trees can always be represented as concept lattices if one formally adds a bottom node. But if a lexical database contains other relations than just a tree, it may be a challenge to embed all these relations into a lattice which retains the original structures without adding too many unnecessary nodes. This problem is discussed by [Groh, Strahringer and Wille 1998] with respect to thesauri which contain several trees. Trees can also be embedded in a manner that allows admissible Boolean combinations of thesaurus query terms to form the nodes of a concept lattice [Skorsky 1997].

The most common visualisation of concept lattices is as line diagrams such as the one in the right half of [Fig. 5]. But users need to be trained in how to read such diagrams. Therefore, some end-user interfaces of FCA applications (such as

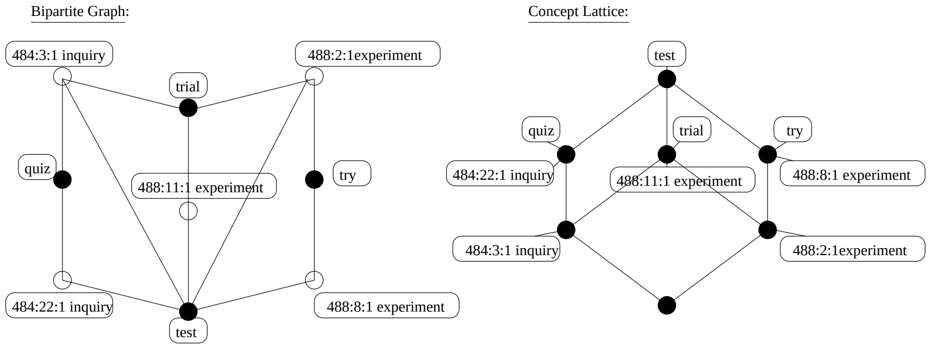

Figure 5: Words connected to "test" in RIT3

CREDO, see `http://credo.fub.it`) convert lattices into tree hierarchies before displaying these to users. These tree hierarchies can be navigated by users similar to "file system displays". Any lattice can be converted into a tree hierarchy by duplicating (or multiplying) nodes. Obviously, many features of lattices (such as the unique lower bounds) are not visible in such displays.

3.5 Many-to-many relations: bipartite graphs

Bipartite relations can be represented as bipartite graphs which are graphs where all nodes can be partitioned into two sets such that all edges are between the sets but not within the sets. The nodes of the graph can be coloured to show which node belongs to which set. The left half of [Fig. 5] shows a bipartite graph for some words closely related to the the word "test" in RIT3. All words of the thesaurus which share at least two senses with the word "test" were selected. These words, "test", "quiz", "try" and "trial" are represented by black nodes in the graph. The senses which they share are represented by white nodes.

The right half of [Fig. 5] shows the same information as a concept lattice. In this case, the words are represented as formal attributes and the senses as formal objects. The lattice represents additional implicit information which is not easily available from the bipartite graph. Implications among objects or attributes are shown. For example, the lattice shows that "quiz" implies "test" because "quiz" labels a node below "test". This means that all senses of "quiz" also belong to "test". Words or senses which have exactly the same crosses in a formal context are grouped at the same node in the lattice ([Fig. 5] does not contain such an example). Again this kind of information is not easily available in the bipartite graph.

On the other hand, bipartite graphs also provide some features which lattices do not have. For example, the nodes can be moved anywhere on the plane, whereas in lattices the up-down movement of nodes is restricted by the ordering (unless, of course, the ordering is displayed using other means such as arrows instead of edges). The nodes of a bipartite graph could thus be arranged according to some external purpose to display further information, for example, semantic distances in a vector space.

There are usually many ways of representing data which can lead to exploration of different kinds of implicit structures. Both bipartite graphs and lattices reach limits of their representational capabilities if too many edge crossings occur, because the graphs can then become unreadable. Different means have been suggested to overcome this limitation using techniques, such as fish-eye views [Furnas 1981], mapping the graphs to hyper-spheres, and navigational displays (e.g. `http://www.plumbdesign.com/thesaurus`) which reveal and hide nodes and links as users click their way through the graphs.

3.6 Combinations of relations

In lexical databases, interesting structures often relate to combinations of different relations. In Roget's Thesaurus an important relation is the one between words and senses [Fig. 6]. The Sense Index, which forms the main part of the first half of the book, can be distinguished [Old 2003] from the Word Index, which forms the second half of the book [Fig. 4]. The Word Index lists words in alphabetical order and indicates senses by Category numbers combined with Paragraph numbers. Within the Sense Index, words are grouped into Categories, Paragraphs and Semicolon Groups (i.e., groups delimited by semicolons). The Semicolon Groups are not numbered in the printed version of the thesaurus, but are numbered in the electronic version. In the printed version, Semicolon Group numbers are not needed because the Paragraphs are short enough to be scanned by a reader in search of a word. The relation between words and senses is a many-to-many relation because each word can occur in several senses (i.e., different Categories, Paragraphs and Semicolon Groups) and each sense usually contains several words.

Only two levels of Roget's Synopsis (Category and Paragraph) are explicitly used in the Word Index. The full structure of the thesaurus can only be deduced if the tree hierarchy of the Synopsis is combined with the many-to-many relation of the Sense and Word Indices [Fig. 3]. The Word and Sense Indices duplicate some information because both contain Categories and Paragraphs. A conversion of the data into electronic format showed that the overlapping information is not entirely consistent. To some degree this is due to omissions in the Word Index, which may have been deliberate in order to shorten and simplify the Word Index in the printed medium. Thus the full structure of the thesaurus is ambiguous.

WORDS → SYNONYMY

SENSES ↓ POLYSEMY

	over	above	beyond	across	past	in excess of	...
9:12:1	X						
206:24:2	X	X					
206:27:4	X	X	X			X	
227:40:1	X	X					
...	X	X	X	X	X		

Figure 6: The word/sense relation in Roget's Thesaurus [Old 2003]

A crucial difference between ROGET and WordNet is that although WordNet also contains a many-to-many relation between words and senses combined with an imperfect tree hierarchy, in WordNet the sense groupings (also called "synsets" or "synonym sets" [Fellbaum 1998]) can occur at any level of the tree whereas in ROGET they occur only at the lowest level. Thus in ROGET a classification system sits on top of a level of Semicolon Groups whereas in WordNet the classification structure consists of synsets at every level. Another difference is that in WordNet the synsets do not have explicit identifiers apart from index numbers which are contained in the ASCII files and hidden from the users. In ROGET, every Semicolon Group can be identified uniquely by its Category number, Paragraph number and Semicolon Group number. In WordNet, a synset can only be identified by the synonyms it contains and by its hyper- and hyponyms (i.e., neighbours in the tree). But this identification need not be reliable, nor unique.

Even though ROGET contains identifiers, neither in ROGET nor in WordNet, is it easy to follow a single Semicolon Group or synset across different editions. The question is what constitutes an identity of such a grouping across different editions: its content, its identifier or its location? Depending on how one answers this question a different picture of the evolution of a lexical database over time is painted. This problem is shared with other hierarchical structures as well, and discussed in detail for botanical taxonomies by [Pullan et al. 2000].

Apart from the word/sense relation and the main imperfect tree of hypernymy, WordNet contains many more relations, such as meronymy, antonymy and entailment. Many combinations are possible, which lead to interesting emergent structures. Roget's Thesaurus contains a few other relations, for example, cross-references which if combined with the Synopsis yield imperfect trees.

4 Restricting Domains

Lexical databases such as ROGET or WordNet are so large that it is not feasible to visually represent their complete trees or to calculate a lattice of their complete word/sense relations or to embed their complete imperfect trees into lattices. It is therefore essential to restrict domains in a meaningful way to subsets which are interesting for a certain purpose. The example in [Fig. 7] shows a "neighbourhood" of the word "test" in RIT3. Obviously, there are many ways to define such neighbourhoods. A first definition was proposed by Wille in an unpublished manuscript and was subsequently used by [Sedelow and Sedelow 1993] and [Priss 1998]. In general, a semantic neighbourhood of a word in a thesaurus consists of all words that share senses with the original word. The set of objects of a "neighbourhood lattice" consists of such a semantic neighbourhood of a word. The attributes are either all senses of the original word, or all senses of all the words in the neighbourhood. The choice depends on whether a larger or a smaller semantic environment is intended.

The operation which underlies the selection of elements in a neighbourhood has been called the "plus operator" as opposed to the "prime operator" used in concept formation. The prime operator is used in FCA to select all attributes which are shared among a set of objects or all objects which are shared by all attributes in a certain set. Applying the prime operator twice yields a closure operator [Ganter and Wille 1999]. That means that further applications of the prime operator do not yield further elements.

The plus operator behaves differently. For a many-to-many relation one starts by selecting an element (or set of elements) from the first domain. Then one selects all elements from the second domain which are related to the first element or set. Then one goes back to the first domain and selects all elements related to the second set and so on. This algorithm was probably re-invented many times. The algorithm is mentioned by [Wunderlich 1980] with respect to bilingual dictionaries: select all German translations of an English word, then all English translations of those German words, and so on. Dyvik applies this method to a parallel corpus of English and Norwegian texts and extracts monolingual thesaurus entries [Dyvik 1998]. (In addition to the plus operator he employs an interesting algorithm which generates feature lattices that are similar to concept lattices. But a more detailed description of his methods is beyond the scope of this paper.)

In contrast to the prime operator, the plus operator does not yield a closure operator after a predetermined number of applications. The plus operator may converge after a different number of applications for different elements. For some elements it may only stop after all elements from both domains have been selected. In Dyvik's parallel corpus examples, 5 applications of the plus operator to a single start term yield up to 7000 words (out of 200,000 entries). With re-

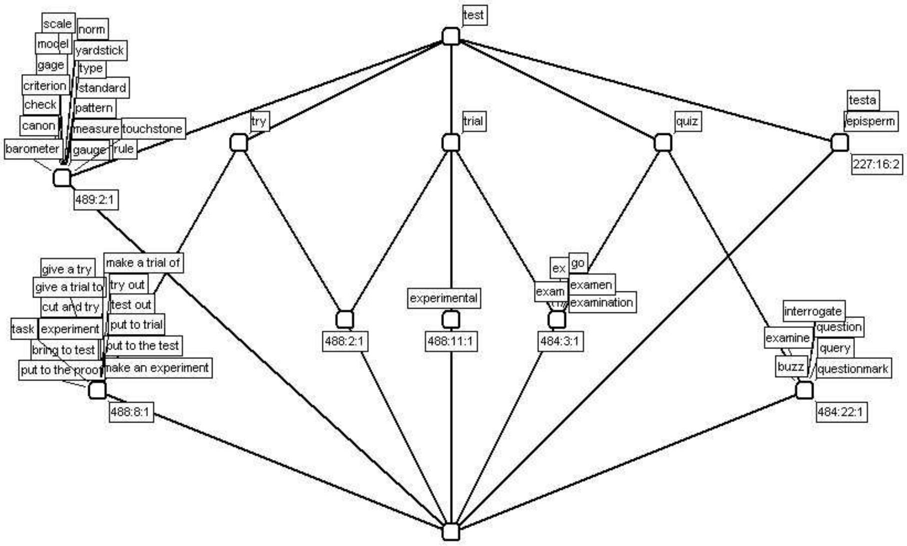

Figure 7: A neighbourhood of "test" (2 plus operators)

spect to RIT3, applying the plus operator two or three times is usually sufficient to obtain semantically relevant sets that can be graphically represented. [Fig. 7] shows the neighbourhood of "test" in RIT3 after applying the plus operator twice. In this case, starting with "test" the first plus operator generates all senses of "test" (which become the objects). The second plus operator generates all other words which have at least one of those senses. These words are taken as attributes.

Old explores a different means of restricting neighbourhoods in RIT3 to their core structures [Old 2003]. The neighbourhood of "test" in [Fig. 7] differs from the neighbourhood in [Fig. 5] because only those words are included in [Fig. 5] which share at least two senses with the original word. Old calls [Fig. 5] a "restricted neighbourhood". The figures show that [Fig. 7] contains all the information from [Fig. 5] plus additional information. But the core structure of "test" is more visible in [Fig. 5].

It is not necessary to include both domains which are involved in the plus operator also in the final graph. [Fig. 8] shows a restricted neighbourhood for the RIT3 senses of "over". [Fig. 9] uses the same set of words as [Fig. 8]. But in this case the senses are omitted. An edge in the graph means that the words share at least one sense. Five clusters are visible in the graph, each forming a complete subgraph. The five clusters correspond to the five co-atoms (neighbours of the

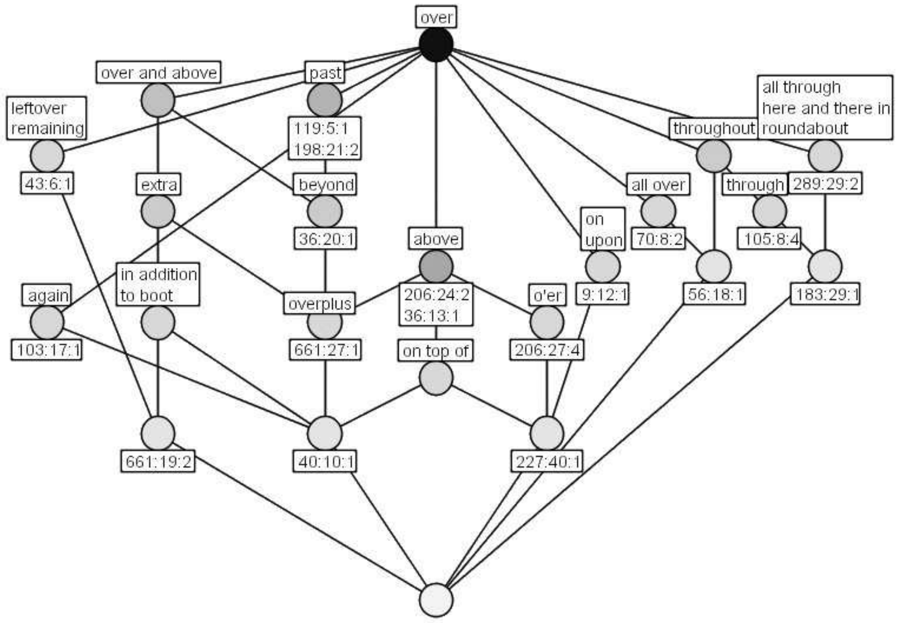

Figure 8: A restricted neighbourhood of "over" (2 plus operators)

bottom node) in [Fig. 8]. Therefore, formally, all of the information of [Fig. 9] is contained in [Fig. 8], but not vice versa. But, associatively, [Fig. 9] makes information explicit which is implicit in [Fig. 8] by emphasising these clusters.

Other means of restricting domains rely on using one relation to select elements and another relation to visualise them. For example, [Skorsky 1997] uses the hypernymy relation in WordNet to select words. In this case, a user can choose the length of a path to navigate up or down the imperfect tree around a target word. The lattice is then constructed based on a different relation, which describes how terms can be combined in queries. This approach is similar to the one suggested by [Groh, Strahringer and Wille 1998].

5 Using lattices as basemaps for other relations

In information cartography [Old 2002], a 3-dimensional coordinate system is used to represent non-spatial information. This can be implemented using Geographic Information System (GIS) software. Lattices can be embedded in such as a system for the purpose of using the navigational features of the GIS software. But this approach can also be turned around: instead of using a 3-dimensional

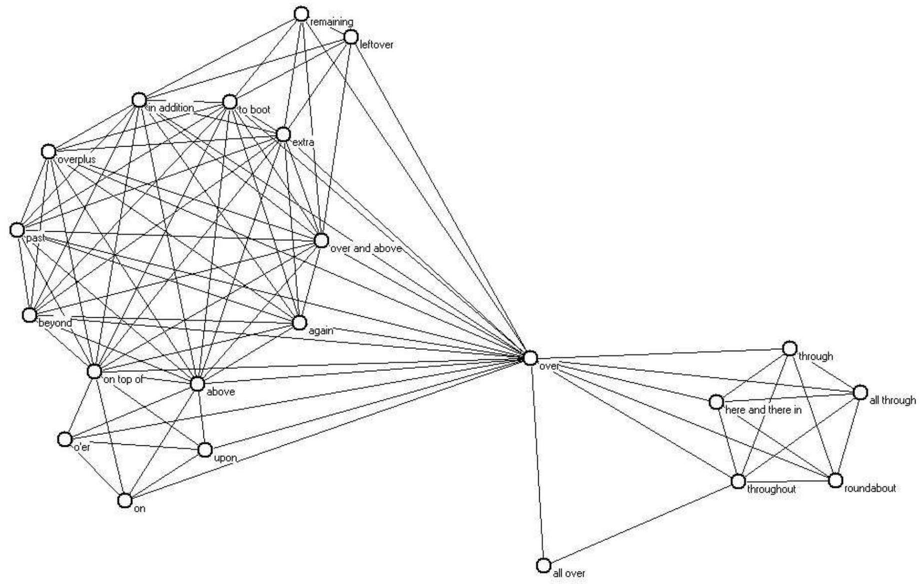

Figure 9: The core senses of "over"

coordinate system as a basemap (or background structure), lattices can be used as a basemap for other relations.

Section 3.6 discussed how central structures of a lexical database are often distributed over different relations. The question there was how to create a lattice which is constructed from such different relations. But in many cases, lexical databases contain a central hierarchical structure, which is complemented by information from other relations, including unary relations. In this case, the hierarchical structure can be used as a basemap for the other relations. For example, [Fig. 10] shows the same lattice as in [Fig. 8] (although up-side-down). An additional unary relation is represented in the shading of the nodes. Each type of shading represents a different part of speech as contained in the thesaurus.

This approach is not restricted to unary relations. Using relational database JOINs, any relation that contains a column that acts as a foreign key to the objects or attributes of a lattice can be mapped onto the lattice. [Fig. 11] shows a restricted neighbourhood lattice where the set of objects corresponds to the descendants of the Indo-European Root "SEK" (modern "cut"). The Roots were selected from a database which contains a relation between Roots and their modern meanings. The modern meanings were taken as a JOIN condition for combining the Roots with words in RIT3. The lattice was then constructed as described in section 4. Instead of displaying the words as objects, the sets of

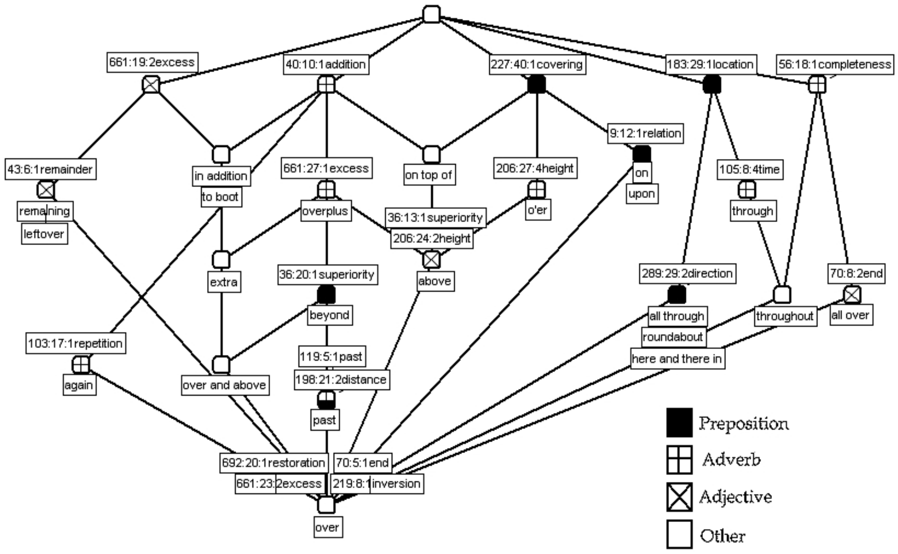

Figure 10: Color-coding nodes in a lattice according to features

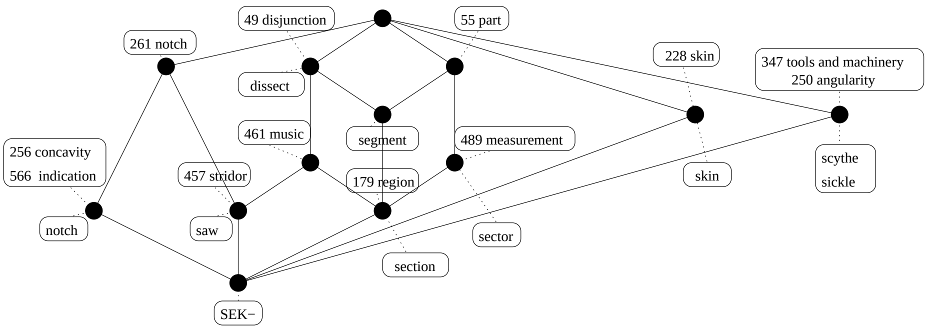

Figure 11: Using a lattice as a basemap: dependents of IE SEK (modern "cut")

Roots could be displayed as objects (because the relation between a database key and another column is always one-to-many). But for this paper we chose to display the words as objects, instead of the Roots as objects, because the Roots are meaningless for anybody who is not an expert in Indoeuropean etymology. The approach of using a lattice as a basemap for another relation is also described by [Priss 2000] where different editions of the Dewey Decimal Classification are compared in that manner.

6 Conclusion

This paper provides guidelines and examples for visualising lexical relations using Formal Concept Analysis. Autorelations in lexical databases often form trees, imperfect trees or poly-hierarchies. These can be embedded into concept lattices. Bipartite many-to-many relations can be represented as bipartite graphs or as concept lattices where the values from one domain are used as the formal objects, the values of the other domain as formal attributes. Because domains of lexical databases are often quite large, algorithms for selecting meaningful subsets of domains are important, such as the use of a plus operator as discussed in section 4. Two sections discuss combinations of relations: section 3.6 describes how central structures of lexical databases can be generated by several relations. Section 5, on the other hand, discusses how to use one lattice as a basemap for other relations.

The guidelines provided in this paper are stated in general terms because every lexical database may have very specific requirements. But we are hoping that this paper can serve as a starting point for a collection of guidelines and examples, which eventually could grow into a collection of examples of "best practises" for modelling lexical databases with FCA. Such a collection would be helpful for users who want to visualise lexical databases without having to spend an enormous amount of time on evaluating visualisation software and on defining modelling techniques.

References

[Bertin 1967] Bertin, Jacques: "Semiology of Graphics: Diagrams, Networks, Maps", Madison, WI: University of Wisconsin Press (1967/1983).

[Clark 1997] Clark, Andy: "Being There. Putting Brain, Body, and World Together", MIT Press (1997).

[Coombs 1964] Coombs, Clyde H.: "A Theory of Data", Wiley, New York (1964).

[Dyvik 1998] Dyvik, Helge: "A Translational Basis for Semantics", In: Johansson and Oksefjell (eds.): Corpora and Crosslinguistic Research: Theory, Method and Case Studies, Rodopi, (1998), 51-86.

[Fellbaum 1998] Fellbaum, C.: "WordNet: An Electronic Lexical Database and Some of its Applications", MIT press (1998).

[Furnas 1981] Furnas, G. W.: "The FISHEYE View: A New Look at Structured Files", Bell Laboratories Technical Memorandum (1981).

[Ganter and Wille 1999] Ganter, Bernhard; Wille, Rudolf: "Formal Concept Analysis. Mathematical Foundations", Berlin-Heidelberg-New York: Springer, Berlin-Heidelberg (1999).

[Gibson 1979] Gibson, James: "The Ecological Approach to Visual Perception", Houghton Mifflin Co (1979).

[Groh, Strahringer and Wille 1998] Groh, Bernd; Strahringer, Selma; Wille, Rudolf: "TOSCANA-Systems Based on Thesauri", Proceedings of the 6th International Conference on Conceptual Structures, LNAI 1453, Springer Verlag (1998), 127-138.

[Johnson and Shneiderman 1991] Johnson, Brian; Shneiderman, Ben: "Treemaps a space filling approach to the visualization of hierarchical information structures", Proceedings of IEEE Visualization'91, IEEE CS Press (1991), 275-282.

[Keeler 2002] Keeler, Mary: "The Place of Images in a World of Text", Computers and the Humanities 36 (2002), 75-93.

[McGarry et al. 1999] McGarry, K.; Wermter, S.; MacIntyre, J.: "Hybrid Neural Systems: From Simple Coupling to Fully Integrated Neural Networks", Neural Computing Surveys 2 (1999), 62-93.

[Nielsen and Tahir 2001] Nielsen, Jakob; Tahir, Marie: "Homepage Usability: 50 Websites Deconstructed", New Riders Publishing, Indianapolis (2001).

[Old 2002] Old, L. John: "Information Cartography: Using GIS for visualizing nonspatial data", Proceedings of the ESRI International Users Conference, San Diego, CA, ESRI Library (2002). On-line available at `http://gis.esri.com/library/userconf/proc02/pap0239/p0239.htm`

[Old 2003] Old, L. John: "The Semantic Structure of Roget's, A Whole-Language Thesaurus", PhD Dissertation. Indiana University (2003).

[Old 2004] Old, L. John: "Unlocking the Semantics of Roget's Thesaurus", In P. Eklund, (Ed.), Proceedings, Second International Conference on Formal Concept Analysis, Springer-Verlag (2004), 236-243.

[Peirce 1894] Peirce, Charles: "What is a Sign", In: Houser; Kloesel (eds). The Essential Peirce. Selected Philosophical Writings. Vol 2. (1893-1913), Indiana University Press.

[Pinker 1991] Pinker, S.: "Rules of language", Science, 253 (1991), 530-535.

[Priss 2002] Priss, Uta: "Associative and Formal Concepts", In: Priss; Corbett; Angelova (eds.), Conceptual Structures: Integration and Interfaces. Proceedings of the 10th International Conference on Conceptual Structures, Springer Verlag, LNAI 2393 (2002), 354-368.

[Priss 2001] Priss, Uta: "Multilevel Approaches to Concepts and Formal Ontologies", In: Efthimiadis (ed.), Advances in Classification Research, Volume 12. Proceedings of the 12th ASIST SIG/CR Classification Research Workshop, Information Today (2001).

[Priss 2000] Priss, Uta: "Comparing Classification Systems using Facets", In: Beghtol, C.; Howarth, L.; Williamson N. (eds.), Dynamism and Stability in Knowledge Organization. Proceedings of the 6th International ISKO Conference (2000), 170-175.

[Priss 1998] Priss, Uta: "Relational Concept Analysis: Semantic Structures in Dictionaries and Lexical Databases", (PhD Thesis) Verlag Shaker, Aachen (1998).

[Pullan et al. 2000] Pullan, M. R.; Watson, M. F.; Kennedy, J. B.; Raguenaud, C.; Hyam, R.: "The Prometheus Taxonomic Model: a practical approach to representing multiple classifications", Taxon, 49 (2000), 55-75.

[Roget 1962] Roget, Peter Mark: "Roget's International Thesaurus", 3rd Edition Thomas Crowell, New York (1962).

[Roget 1911] Roget, Peter Mark: "Roget's Thesaurus", Available from the Project Gutenberg (1911), `http://promo.net/pg`

[Sedelow and Sedelow 1993] Sedelow, S.; Sedelow, W.: "The Concept 'concept"', Proceedings of the Fifth International Conference on Computing and Information, Sudbury, Ontario, Canada (1993), 339-343.

[Shneiderman 1996] Shneiderman, Ben: "The eyes have it: A task by data type taxonomy of information visualizations", Proceedings of the IEEE Symposium on Visual Languages, IEEE, Los Alamos, CA (September 1996), 336-343.

[Skorsky 1997] Skorsky, Martin: "Graphische Darstellung eines Thesaurus", Deutscher Dokumentartag, Regensburg (1997).

[Wunderlich 1980] Wunderlich, Dieter: "Arbeitsbuch Semantik", Athenaeum, Königstein im Taunus (1980).