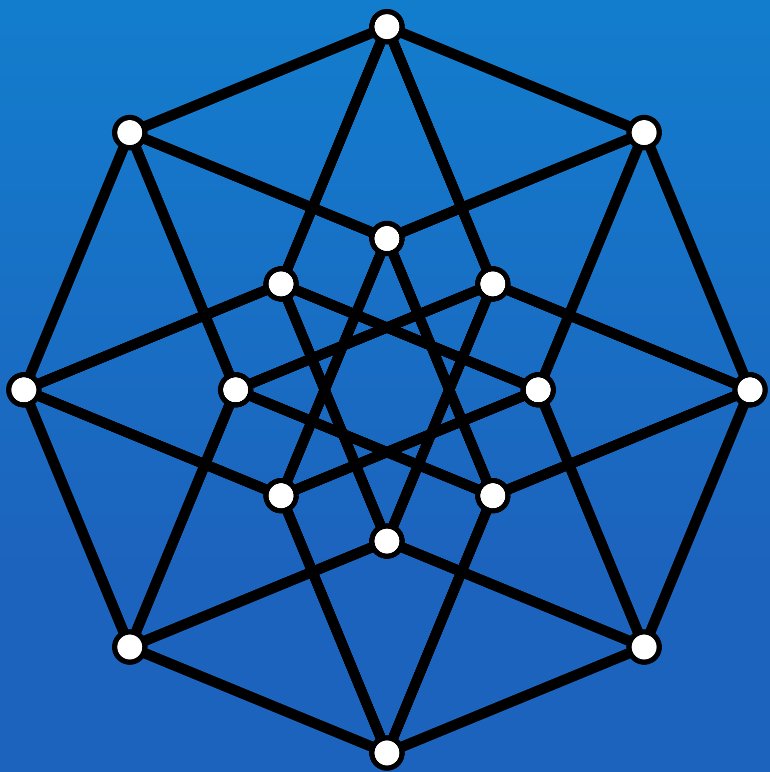
TECHNISCHE UNIVERSITÄT DRESDEN

Visualization of Conceptual Data
with Methods of Formal Concept Analysis

Francesco Kriegel

Diploma Thesis



Technische Universität Dresden
Fachrichtung Mathematik

Institut für Algebra

Graphische Darstellung begrifflicher Daten mit Methoden der formalen Begriffsanalyse

Diplomarbeit
zur Erlangung des ersten akademischen Grades

Diplommathematiker

vorgelegt von

Name: Kriegel Vorname: Francesco

geboren am: 12.06.1986 in: Jena

Tag der Einreichung: 14.07.2012

Betreuer: Prof. Dr. Bernhard Ganter
Dr. rer. nat. Frithjof Dau

Contents

1 Introduction

1.1 Acknowledgements

Zuallererst möchte ich mich bei meinen Eltern Carmen und Hans-Jürgen Kriegel für ihre tatkräftige und immerwährende Unterstützung in allen Lebenslagen bedanken. Sie haben meine Interessen und Fähigkeiten stets bestmöglich gefördert.
Also I gratefully thank my both supervisors Prof. Dr. Bernhard Ganter (TU Dresden) and Dr. Frithjof Dau (SAP) for their support and the many interesting and fruitful discussions.
Especially I also want to thank Prof. Dr. Ulrike Baumann a lot. First, for her great support when applying at SAP, and second, for always encouraging me to give student tutorials.
Last but not least thanks to Dipl.-Math. Ilse Ilsche, one of the good souls at the Institute for Algebra.
Furthermore, thanks to Matthias Lange and Jonas Stolle.
Finally also many thanks to my friends for being there and supporting me.
Sincere thanks are given to all forgotten people.

1.2 Supporting University: TU Dresden, Institute for Algebra

Research at the Dresden Institute of Algebra emphasizes on the foundations and, in equal measure, the applications of algebra. At the same time it includes the theory of ordered sets and graph theory.
In particular we see fields of application of algebra in the formal description of data and knowledge being a foundation e.g. for data analysis and knowledge processing. General algebra for data modelling, geometric algebra for data security, and lattice theory for conceptual knowledge processing have proven especially valuable as methodological foundations.

Figure 1.1: Willersbau at TU Dresden, with Institute of Algebra

The "classical" fields of activity in algebra are more intensively studied elsewhere. Here we primarily focus on general algebra and the theory of order and lattices. A major emphasis is on function and relation systems

and formal concept analysis. Both areas, the study of foundations and applications, are closely intertwined. The institute maintains contacts with leading researchers worldwide.
Research groups focus on topics from algebraic structure theory, discrete structures, methods of applied algebra, and universal algebra. As a special (however little) research highlight we would like to accentuate the field of "mathematical theory of music". There particularly the experimental micro-tonal musical instrument "MUTABOR" is being tested and developed.
Our research work is open to students. They benefit from this as they become familiar with the present work techniques in general and applied algebra, conceptual knowledge processing and graph theory. (TUD)

1.3 Supporting Corporation: SAP AG, Research Center Dresden

The Dresden region, also referred to as "Silicon Saxony," is located in the eastern part of Germany, close to the German-Polish and German-Czech borders. SAP Research aims to capitalize on this fact by turning SAP Research Dresden into the company's hub for research collaborations with companies and academics from Eastern Germany and the EU member states of Eastern Europe. The location is in close proximity to the campus of the Technische Universität Dresden (TU Dresden).

Figure 1.2: SAP Research Center Dresden in the Falkenbrunnen

SAP Research Dresden contributes significantly to three SAP Research topics: mobile computing and user experience, business intelligence, and software engineering and tools. SAP Research Dresden also manages the Future Factory Initiative living lab.

Research Environment – Working with World-Class Researchers

SAP Research Dresden finds itself in a prospering environment perfectly suited for a research location of its kind. The Dresden metro area is the home of production facilities and research labs of major semiconductor engineering, multimedia, and information management companies. Global players, as well as a considerable network of small and midsize high-tech businesses, constitute excellent cooperation opportunities for SAP Research. In addition, TU Dresden has been a longstanding loyal, beneficial, and faithful partner of SAP Research. Together with TU Dresden, the research location in Dresden offers a Ph.D. program to highly-qualified students. Covered research topics are as follows:

- Business Intelligence
- Internet Applications & Services
- Mobile Computing & User Experience
- Software Engineering & Tools

(SAP)

1.4 Research Project: CUBIST

Constantly growing amounts of data, complicated and rapidly changing economic interactions, and an emerging trend of incorporating unstructured data into analytics, is bringing new challenges to Business Intelligence (BI). Contemporary solutions involve BI users dealing with increasingly complex analyses. According to a 2008 study by Information Week, the complexity of BI tools and their interfaces is becoming the biggest barrier to success for these systems. Moreover, classical BI solutions have, so far, neglected the meaning of data, which can limit the completeness of analysis and make it difficult, for example, to remove redundant data from federated sources.

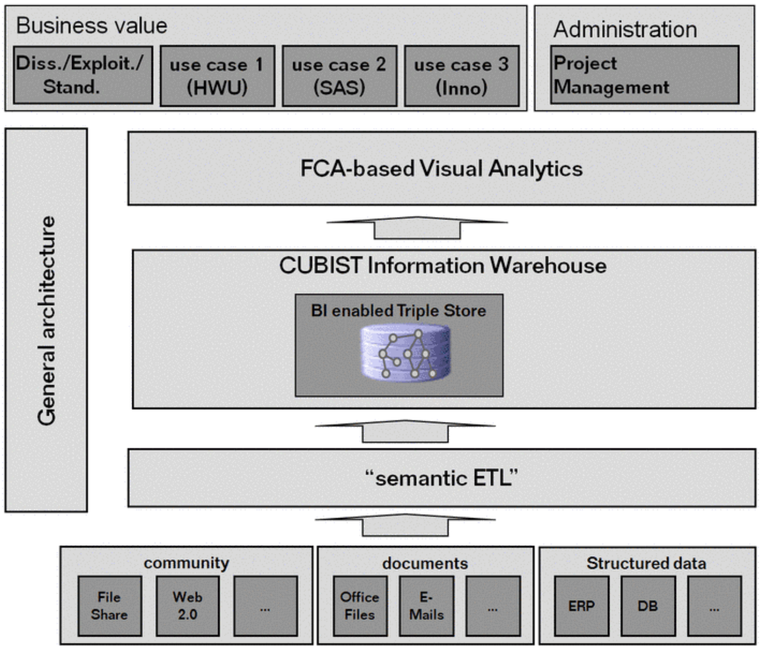


Figure 1.3: Structure of CUBIST project

Semantic Technologies, however, focus on the meaning of data and are capable of dealing with both unstructured and structured data. Having the meaning of data and a sound reasoning mechanism in place, a user can be better guided during an analysis. For example, a piece of information can be semantically explained or a new relevant fact can brought to the user's attention. In particular, we foresee a well known semantic technique called Formal Concept Analysis (FCA) to be a key element of new hybrid BI system. FCA can be used to guide a user in discovering new facts, which are not explicitly modelled in the data warehouse schema. Semantic analysis could also improve classical methods in BI, such as data reduction and duplicate detection. However, semantic technologies have traditionally operated on data sets a magnitude smaller than classical BI solutions. They also lack standard BI functionalities such as Online Analytical Processing (OLAP) queries, making it difficult to perform analysis over semantic data. The CUBIST project develops methodologies and a platform that combines essential features of Semantic Technologies and Business Intelligence. With CUBIST, we envision a system with the following core features:

- Support for the federation of data from a variety of unstructured and structured sources.
- A data persistency layer in the form of a semantic Data Warehouse; a hybrid approach based a BI enabled triple store.
- Semantic information used to improve BI best practices in, for example, data reduction and preprocessing; CUBIST enables a user to perform BI operations over semantic data.
- A semantic data warehouse that realizes the advanced mining techniques of Formal Concept Analysis (FCA).
- FCA guides the user in performing BI and helps the user discover facts not expressed explicitly by the warehouse model.
- Novel ways of applying visual analytics in which meaningful diagrammatic representations will be

used for depicting the data, navigating through the data and for visually querying the data.

CUBIST demonstrates the resulting technology stack in the fields of market intelligence, computational biology and the field of control centre operations.
CUBIST is funded by the European Commission under the 7th Framework Programme of ICT, topic 4.3: Intelligent Information Management. (Cub)

1.5 Task Description und Structure of the Diploma Thesis

My task was to investigate and implement methods for visualizing conceptual data. This thesis is subdivided into two parts: a mathematical part and an implementational part. Some fundamentals of formal concept analysis and methods for drawing them are presented in the next chapter. The third chapter contains the main theoretical part, where an algorithm for updating labeled additive concept diagram upon insertion or removal of a single attribute column in the base context is presented and proven. The fourth chapter gives some techniques for interaction with concept diagrams. The second parts starts with a requirement analysis for visualizing graphs and lattices, and interacting with them. Then in the next section some details of the implementation are given. Within the framework of this thesis a Java program has been written.

Part I

Mathematical Details

2 Fundamentals of Formal Concept Analysis

This initial chapter gives some fundamental definitions and theorems needed for formal concept analysis. The first section contains basic properties of formal concept lattices. Please see (GW99) for further details. The second section introduces diagrams for concept lattice visualization and the the third section gives some statements on context appositions.

2.1 Concepts and Concept Lattice

Definitio: Formal Context 2.1

A FORMAL CONTEXT $\mathbf{K}$ is a triple (G, M, I) such that G and M are sets and $I \subseteq G \times M$ is a binary relation between them. Elements of G are called OBJECTS and those from M are ATTRIBUTES. I is the INCIDENCE and for $(g, m) \in I$ one also writes gIm and says that g HAS m.

There are various views of a context. In data mining a context is also called BOOLEAN DATABASE or TRANSACTION DATABASE. The objects are also called TRANSACTIONS or ROWS. Analogously, attributes are ITEMS, PROPERTIES or COLUMNS. Such boolean databases can also be defined as a multiset of subsets of a given set of items. A multiset is just a bag, *i.e.* a set that can contain multiple copies of the same element, and could be modelled as a mapping $\mathcal{M}\colon M \to \mathbb{N}$ that maps each element m of the underlying set M to the number $\mathcal{M}(m)$ of copies of m within $\mathcal{M}$.

2.2 **Theorema: Contexts & Galois Connections**

Each context $\mathbf{K} = (G, M, I)$ introduces a galois connection $(\phi_{\mathbf{K}}, \psi_{\mathbf{K}})$ between G and M. The mappings are given as follows:

$$\phi_{\mathbf{K}}: \begin{array}{l} \wp(G) \to \wp(M) \\ A \mapsto A^I := \left\{ m \in M \,\middle|\, \underset{g \in A}{\forall} gIm \right\} \end{array} \quad \text{and} \quad \psi_{\mathbf{K}}: \begin{array}{l} \wp(M) \to \wp(G) \\ B \mapsto B^I := \left\{ g \in G \,\middle|\, \underset{m \in B}{\forall} gIm \right\} \end{array}$$

In the opposite direction every galois connection (ϕ, ψ) between sets X and Y introduces a context $\mathbf{K}_{(\phi,\psi)} := (X, Y, I_{(\phi,\psi)})$ with

$$I_{(\phi,\psi)} := \{(x,y) \in X \times Y \mid x \in \psi(\{y\})\} = \{(x,y) \in X \times Y \mid y \in \phi(\{x\})\}$$

Both operations are inverse to each other, *i.e.* $(\phi_{\mathbf{K}_{(\phi,\psi)}}, \psi_{\mathbf{K}_{(\phi,\psi)}}) = (\phi, \psi)$ holds for every galois connection (ϕ, ψ) and $\mathbf{K}_{(\phi_{\mathbf{K}}, \psi_{\mathbf{K}})} = \mathbf{K}$ holds for all contexts $\mathbf{K}$ respectively.

As well known from the theory of galois connections, the mappings $\phi_{\mathbf{K}}$ and $\psi_{\mathbf{K}}$ are order-reversing, *i.e.*

$$\underset{A,C \subseteq G}{\forall} A \subseteq C \Rightarrow A^I \supseteq C^I$$
$$\text{and} \underset{B,D \subseteq M}{\forall} B \subseteq D \Rightarrow B^I \supseteq D^I.$$

Furthermore, the compositions $\psi_{\mathbf{K}} \circ \phi_{\mathbf{K}}$ and $\phi_{\mathbf{K}} \circ \psi_{\mathbf{K}}$ are closure operators on G and M respectively, *i.e.* they are extensive and idempotent, *i.e.*

$$\underset{A \subseteq G}{\forall} A \subseteq A^{II} \text{ and } A^I = A^{III}$$
$$\text{and} \underset{B \subseteq M}{\forall} B \subseteq B^{II} \text{ and } B^I = B^{III}.$$

Furthermore the closures of are always of the form A^I for an object set $A \subseteq G$, and B^I for an attribute set $M \subseteq M$ respectivelly. An equivalent characterization of galois connections yields

$$\underset{\substack{A \subseteq G \\ B \subseteq M}}{\forall} A \subseteq B^I \Leftrightarrow A^I \supseteq B \Leftrightarrow A \times B \subseteq I \Leftrightarrow \underset{g \in A}{\forall} \underset{m \in B}{\forall} gIm.$$

2.3 **Definitio: Biset, Preconcept, Concept**

Let $\mathbf{K} = (G, M, I)$ be a context. A pair (A, B) is called

BISET iff $A \subseteq G$ and $B \subseteq M$,

PRECONCEPT or RECTANGLE iff $A \times B \subseteq I$,

CONCEPT iff $A^I = B$ as well as $A = B^I$.

For a concept (A, B) the object set A is called EXTENT and the attribute set B is called INTENT. The set of all concepts is denoted by $\mathfrak{B}(\mathbf{K})$, the set of all extents is $\mathrm{Ext}(\mathbf{K})$ as well as $\mathrm{Int}(\mathbf{K})$ is the set of all intents.

One can easily see that each concept is a preconcept and every preconcept is a biset. Each concept has a closed object set as its extent and a closed attribute set as its intent. Furthermore each concept has the form (A^{II}, A^I) for a suitable object set $A \subseteq G$ and also (B^I, B^{II}) for a suitable attribute set $B \subseteq M$.

Lemma 2.4

For each family of object sets $\{A_t\}_{t\in T}$ and analogously for each family of attribute sets $\{B_t\}_{t\in T}$ it holds that

$$\bigcap_{t\in T} A_t^I = \left(\bigcup_{t\in T} A_t\right)^I \text{ and } \bigcap_{t\in T} B_t^I = \left(\bigcup_{t\in T} B_t\right)^I.$$

It follows that every extent is an intersection of object extents and dually each intent is an intersection of attribute intents, *i.e.*

$$\underset{A\subseteq G}{\forall} A^I = \bigcap_{g\in A} g^I \text{ and } \underset{B\subseteq M}{\forall} B^I = \bigcap_{m\in B} m^I.$$

Theorema: Concept Lattice 2.5

The concepts of a context $\mathbf{K} = (G, M, I)$ can be ordered: A concept (A, B) is a SUBCONCEPT of a concept (C, D), and dually (C, D) is a SUPERCONCEPT of (A, B), iff $A \subseteq C$ holds for the extents or dually iff $B \supseteq D$ holds for the intents. This is symbolized by $(A, B) \leq (C, D)$. The set of all concepts $\mathfrak{B}(\mathbf{K})$ together with the subconcept relation $\leq$ introduces a complete lattice $\underline{\mathfrak{B}}(\mathbf{K}) := (\mathfrak{B}(\mathbf{K}), \leq)$ and the supremum (join) and the infimum (meet) respectively for a family of concepts $\{(A_t, B_t)\}_{t\in T}$ are given as follows:

$$\bigwedge_{t\in T} (A_t, B_t) = \left(\bigcap_{t\in T} A_t, \left(\bigcup_{t\in T} B_t\right)^{II}\right) \text{ and } \bigvee_{t\in T} (A_t, B_t) = \left(\left(\bigcup_{t\in T} A_t\right)^{II}, \bigcap_{t\in T} B_t\right)$$

The concept $\gamma(g) := (\{g\}^{II}, \{g\}^I)$ is called OBJECT CONCEPT of $g \in G$, and $\mu(m) := (\{m\}^I, \{m\}^{II})$ is called ATTRIBUTE CONCEPT of $m \in M$. Why do we need these special concepts? This is due to the possibility of displaying an arbitrary concept by a number of object concepts, or attribute concepts respectively.

Corollarium 2.6

The set of object concepts is $\bigvee$-dense and the set of attribute concepts is $\bigwedge$-dense in the concept lattice $\underline{\mathfrak{B}}(\mathbf{K})$. Each formal concept is the supremum of object concepts below, and dually is the infimum of the attribute concepts above, *i.e.*

$$\bigvee_{g\in A} (g^{II}, g^I) = \bigvee \gamma(A) = (A, B) = \bigwedge \mu(B) = \bigwedge_{m\in B} (m^I, m^{II}).$$

Hence, each $\bigvee$-irreducible concept is an object concept and dually each $\bigwedge$-irreducible concept is an attribute concept.

In many cases not the whole formal context is necessary to fully describe the structure of a concept lattice. A subcontext $(H, N, I \cap H \times N)$ is called DENSE in (G, M, I) iff $\gamma(H)$ is $\bigvee$-dense and $\mu(N)$ is $\bigwedge$-dense in $\underline{\mathfrak{B}}(G, M, I)$. Each concept lattice of a dense subcontext is isomorphic to the concept lattice of the whole context.

2.7 **Lemma**

A subcontext $(H, N, I \cap H \times N)$ is dense in (G, M, I), iff

$$A^I = (A \cap H)^I \text{ and } B^I = (B \cap N)^I$$

holds for every object set $A \subseteq G$ and attribute set $B \subseteq M$.

2.2 Visualizations of Concept Lattices

2.2.1 Transitive Closure and Transitive Reduction

Let R be a binary relation on a set M. A TRANSITIVE CLOSURE of R is a minimal transitive superrelation, symbolized by R^+. As the intersection of transitive relations is again transitive, a transitive closure must be uniquely determined by

$$R^+ = \bigcap_{\substack{S \supseteq R \\ S \text{ transitive}}} S.$$

Overthis the transitive closure can be computed directly: Let $R^1 := R$ and $R^n := R^{n-1}; R$ for all natural numbers $n > 1$, then

$$R^+ = \bigcup_{n>1} R^n$$

holds. By construction $\bigcup_{n>1} R^n$ is a transitive superrelation of R. To inductivelly proof its minimality let S be another transitive superrelation of R. The base case: S contains $R = R^1$. The inductive step: Whenever S contains R^n it must also contain $R^n; R = R^{n+1}$ since S is transitive. The relation S thus contains all powers R^n and is a superrelation of $\bigcup_{n>1} R^n$.
There is a natural isomorphism between binary relations and binary square matrices. R can be displayed by a square matrix whose rows and columns are labeled with the elements of the base set M and whose entries are either 1 iff the appropriate row and column label are in relation or 0 otherwise. This permits the computation of relation compositions like for the transitive closure by means of matrix multiplication. Furthermore there is also a canonical isomorphism between binary relations and directed graphs. R can be seen as a graph with the elements from M as nodes and there is an edge from x to y iff $x \; R \; y$. Thereby the transitive closure can also be determined using graph algorithms like the FLOYD-WARSHALL algorithm. However both naïve matrix multiplication and FLOYD-WARSHALL algorithm have time complexity $\mathcal{O}(n^3)$ where n is the cardinality of M. There are various algorithms with lower time complexity but higher constant factor. So they are only faster for huge input sets.
A TRANSITIVE REDUCTION of R is a minimal subrelation $R^- \subseteq R$ such that the transitive closure of R^- equals the transitive closure of R. For an acyclic relation R the transitive reduction is unique. Especially all (strict) order relations are acyclic. It then can be computed by means of the transitive closure and is given by

$$R^- = R \setminus (R; R^+).$$

For further information please have a look on (AGU72). In summary the transitive reduction of a relation is obtained by removing all transitively redundant pairs.

2.2.2 Neighborhood Relation

Let $(P, \leq)$ be an ordered set and $p, q \in P$. Then p is COVERED BY q iff $p < q$ and there is no element $x \in P$ with $p < x < q$, *i.e.* iff $(p, q) \in < \setminus <^2$. One then also say that q COVERS p, or p and q are NEIGHBORING, and write $p \prec q$ or $q \succ p$. Thereby a binary relation $\prec$ on P is obtained that is called NEIGHBORHOOD or COVER relation. In the finite case the order relation $\leq$ and the cover relation $\prec$ determine each other in a unique way. One can show that the neighborhood $\prec$ is the (unique) transitive reduction of the corresponding strict order $<$ and dually the strict order $<$ is the transitive closure of the neighborhood $\prec$. This is due to the fact that $p < q$ hold iff there is a finite sequence $p \prec x_0 \prec x_1 \prec \cdots \prec x_k \prec q$ in P, *i.e.* iff $p \prec^+ q$ is true. Indeed, $\prec$ is the smallest subrelation whose transitive closure equals $<$, since $p \prec q$ always implies $(p, q) \notin (\prec \setminus \{(p, q)\})^+$.

2.2.3 Line Diagram

Every finite ordered set $(P, \leq)$ can be visualized in the real plane (or more generally in the real space) by a LINE DIAGRAM. A line diagram is an arrangement of circles (nodes) and interconnecting lines (edges). First of all a PLACEMENT FUNCTION

$$\mathsf{pos}\colon P \to \mathbb{R}^2,$$

is required, that assigns a position $\mathsf{pos}(p) = (\mathsf{pos}_x(p), \mathsf{pos}_y(p))$ in the real plane to each element p of P. The placement must be injective to ensure distinguishability for different nodes in the drawn diagram. The elements of p are then depicted by circles at their position $\mathsf{pos}(p)$ in the plane, and each circle is labeled with its appropriate element p. Next, two circles at $\mathsf{pos}(p)$ and $\mathsf{pos}(q)$ are joined by a straight line segment, denoted by $\mathsf{pos}(p, q)$, iff p and q are neighboring in $(P, \leq)$. To omit arrowheads, the diagram is drawed upwards, *i.e.* the vertical coordinate $\mathsf{pos}_y(p)$ is smaller than $\mathsf{pos}_y(q)$ whenever p is smaller than q. No node at $\mathsf{pos}(p)$ intersect any edge $\mathsf{pos}(q, r)$ if $p \neq q$ and $p \neq r$. This ensures no node being on any edge except the start and end node, otherwise it would not be clear where the edge starts and ends.
A generalisation are line diagrams with continuous curves as edges: For an ordered set $(P, \leq)$ a LINE DIAGRAM WITH CURVES is defined as a mapping

$$\mathsf{pos}\colon P \cup \prec \to \mathbb{R}^2 \cup \wp(\mathbb{R}^2)$$

such that $\mathsf{pos}|_P$ is a line diagram, and $\mathsf{pos}|_\prec\colon \prec \to \wp(\mathbb{R}^2)$ assigns to each neighborhood $p \prec q$ a one-dimensional set $\mathsf{pos}(p, q)$ of points in the real plane $\mathbb{R}^2$, such that $\mathsf{pos}(p, q) = \gamma_{pq}[0, 1]$ is the image of a plane curve $\gamma_{pq}\colon [0, 1] \to \mathbb{R}^2$ starting at $\gamma_{pq}(0) = \mathsf{pos}(p)$ and ending at $\gamma_{pq}(1) = \mathsf{pos}(q)$.
The question arises whether a line diagram must be completely defined by assigning a position to each element of the underlying ordered set, or if it suffices to give position for certain elements and compute the remaining position by means of them. This leads to the additive line diagrams. For example, when $(P, \leq)$ is a finite complete lattice, then the $\bigwedge$-irreducibles form a $\bigwedge$-dense set and each element p can thus be displayed as an infimum of all $\bigwedge$-irreducibles smaller than p.
An order-preserving mapping $\mathsf{rep}\colon (P, \leq) \to (\wp(S), \subseteq)$ is called SET REPRESENTATION of $(P, \leq)$ in S. A SEED VECTOR MAP is a map $\mathsf{seed}\colon S \to \mathbb{R}^2$ with $\mathsf{seed}(s) = (\mathsf{seed}_x(s), \mathsf{seed}_y(s))$ for each element s of the representing set S. Then the mapping

$$\mathsf{pos}\colon \begin{array}{l} P \to \mathbb{R}^2 \\ p \mapsto \displaystyle\sum_{x \in \mathsf{rep}(p)} \mathsf{seed}(x) \end{array}$$

is a line diagram and is called ADDITIVE LINE DIAGRAM of $(P, \leq)$ w.r.t. rep and seed. To ensure the upward drawing convention the seed vectors must be chosen with positive vertical coordinates. It is also possible to choose a order-reversing set representation and seed vectors with negative vertical coordinates. Both possibilities yield the same diagrams for bounded ordered sets (especially lattices), as an order-preserving set representation rep with upward seed vectors seed can be transformed in an order-reversing set representation $\mathsf{rep}'\colon p \mapsto \mathsf{rep}(\top) \setminus \mathsf{rep}(p)$ with downward seed vectors $\mathsf{seed}' = -\mathsf{seed}$ such that $\mathsf{pos}'(p) = \mathsf{pos}(p) - \mathsf{pos}(\top)$ hold. Indeed:

$$\begin{aligned} \mathsf{pos}(p) - \mathsf{pos}(\top) &= \sum_{x \in \mathsf{rep}(p)} \mathsf{seed}(x) - \sum_{x \in \mathsf{rep}(\top)} \mathsf{seed}(x) \\ &= -\left(\sum_{x \in \mathsf{rep}(\top)} \mathsf{seed}(x) - \sum_{x \in \mathsf{rep}(p)} \mathsf{seed}(x) \right) \\ &= -\sum_{x \in \mathsf{rep}(\top) \setminus \mathsf{rep}(p)} \mathsf{seed}(x) \\ &= \sum_{x \in \mathsf{rep}'(p)} \mathsf{seed}'(x) \\ &= \mathsf{pos}'(p). \end{aligned}$$

2.2.4 Concept Diagram

For concept lattices there a three canonical ways to define such additive line diagrams. It is well known that the attribute concepts make up a $\bigwedge$-dense set and dually the object concepts form a $\bigvee$-dense set in $\mathfrak{B}(G, M, I)$.

(I) An ATTRIBUTE ADDITIVE line diagram of $\mathfrak{B}(G,M,I)$ is given by the intent projection as set representation $\mathsf{rep}\colon (A,B) \mapsto B$ and downward seed vectors. One should further restrict to the irreducible attributes to gain a clearer diagram. It is not possible to omit an irreducible attribute, as then some concept nodes would coincide.

(II) Dually an OBJECT ADDITIVE line diagram of $\mathfrak{B}(G,M,I)$ can be obtained by means of the extent projection as set reprentation $\mathsf{rep}\colon (A,B) \mapsto A$ and upward seed vectors. Again it is better to only choose seed vectors for irreducible attributes.

(III) One can also combine these two approaches to gain HYBRID ADDITIVE line diagrams. The intent projection is order-reversing while the extent projection is order-preserving. So to gain a suitable set representation combining both approaches, one of the projections must be reversed as described above. The reversed extent projection is $(A,B) \mapsto \mathsf{rep}(\top) \setminus \mathsf{rep}(A,B) = G \setminus A = \complement A$ and dually the reversed intent projection is $(A,B) \mapsto \mathsf{rep}(\bot) \setminus \mathsf{rep}(A,B) = M \setminus B = \complement B$. Thereby both set representations $\mathsf{rep}\colon (A,B) \mapsto A \cup \complement B$ together with upward seeds and $\mathsf{rep}\colon (A,B) \mapsto \complement A \cup B$ with downward seeds provide suitable hybrid additive line diagrams, and yield the same diagrams.

When dealing with a formal context and its concept lattice, a line diagram is hardly readable when each node is fully labeled with the corresponding formal concept. As the object concepts construct all concepts by means of suprema, one can just label each object concept γg with g and then read off the objects in a concept extent by collecting all objects that label the node itself or a node being connected by a descending path. This due to the fact, that a concept (A,B) contains an object g in its extent iff the object concept $\gamma g = (g^{II}, g^{I})$ is smaller than (A,B).[1] Dually the attribute concepts are a infimum base, and one can thus label each attribute concept μm with m. The attributes contained in a concept intent can then be read off by collecting all attributes labeling the node itself or a concept node being connected by an ascending path. This is true, since $m \in B$ hold for a formal concept (A,B) if and only if $\mu m \geq (A,B)$.

2.8 **Definitio: Labeled Additive Concept Diagram**

A LABELED ADDITIVE CONCEPT DIAGRAM for a formal context $\mathbf{K} = (G,M,I)$ is defined as a triple

$$\mathfrak{D}(\mathbf{K}) := (\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}).$$

Thereby $\mathfrak{N}(\mathbf{K})$ is the set of CONCEPT NODES. Each concept $(A,B) \in \mathfrak{B}(\mathbf{K})$ has an appropriate concept node $(A,B,A_\lambda,B_\lambda)$ with its OBJECT LABELS A_λ and ATTRIBUTE LABELS B_λ respectivelly, that are given by the conventions

$$A_\lambda := \{g \in G \mid \gamma(g) = (A,B)\} = \left\{ g \in A \,\middle|\, g^I = B \right\}$$
$$\text{and } B_\lambda := \{m \in M \mid \mu(m) = (A,B)\} = \left\{ m \in B \,\middle|\, m^I = A \right\}.$$

The concept nodes inherit the neighborhood relation from the concepts via

$$(A,B,A_\lambda,B_\lambda) \prec (C,D,C_\lambda,D_\lambda) :\Leftrightarrow (A,B) \prec (C,D).$$

Furthermore, seed is a seed vector map, such that for every object g there is an OBJECT SEED VECTOR $\mathsf{seed}(g) \in \mathbb{R}^2$ and for every attribute m there is an ATTRIBUTE SEED VECTOR $\mathsf{seed}(m) \in \mathbb{R}^2$. The POSITION of a concept node is then defined by a hybrid representation $\mathsf{rep}\colon (A,B) \mapsto \complement A \cup B$

$$\mathsf{pos}\colon (A,B) \mapsto \sum_{g \in \complement A} \mathsf{seed}(g) + \sum_{m \in B} \mathsf{seed}(m).$$

A labeled additive concept diagram is called ATTRIBUTE ADDITIVE, iff all object seed vectors equal the null vector, and analogously it is called OBJECT ADDITIVE, iff all attribute seed vectors are null vectors. In all other cases it is called HYBRID ADDITIVE.

Due to the chosen hybrid representation, the seed vectors must point downwards to ensure the upward drawing convention. A context diagram can be transformed in its coordinates by applying a TRANSFORMATION

[1] From $g \in A$ it follows $g^{II} \subseteq A^{II} = A$ and this means $\gamma g \leq (A,B)$. In the opposite direction $\gamma g \leq (A,B)$ implies $g \in A$ as surely $g \in g^{II}$ hold.

FUNCTION $\theta\colon \mathbb{R}^2 \to \mathbb{R}^2$ on the seeds. The TRANSFORMED diagram is then given as

$$\mathfrak{D}_\theta(\mathbf{K}) := (\mathfrak{N}(\mathbf{K}), \prec, \theta \circ \mathsf{seed}).$$

A toy example is given in figure 2.1.

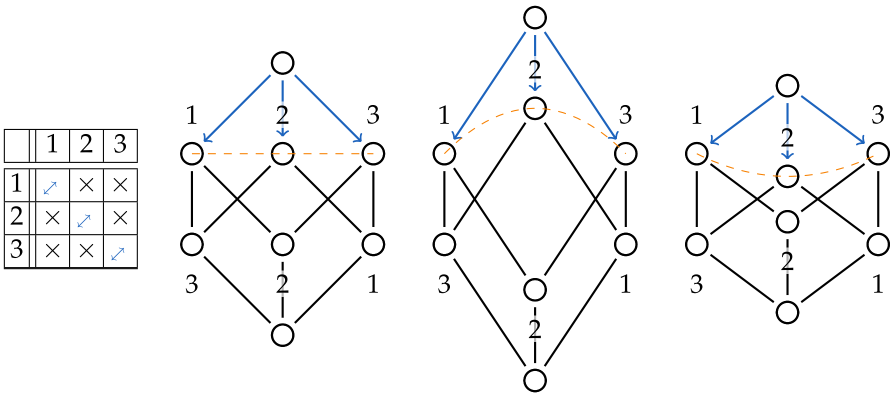


Figure 2.1: Concept diagram and transformations for the three-dimensional boolean scale context. The transformation functions for the second diagram is $\theta_1\colon (x, y) \mapsto (x, y + \frac{1}{2}x^2)$ and the one for the third diagram is $\theta_2\colon (x, y) \mapsto (x, y - \frac{1}{4}x^2)$.

2.2.5 Vertical Hybridization

For distributive concept lattices, the attribute additive approach gives the best additive line diagrams. However, in the non-distributive case the problem of a distended base can occur. This can be seen on the attribute additive line diagram for the concept lattice of a nominal scale with at least three elements.

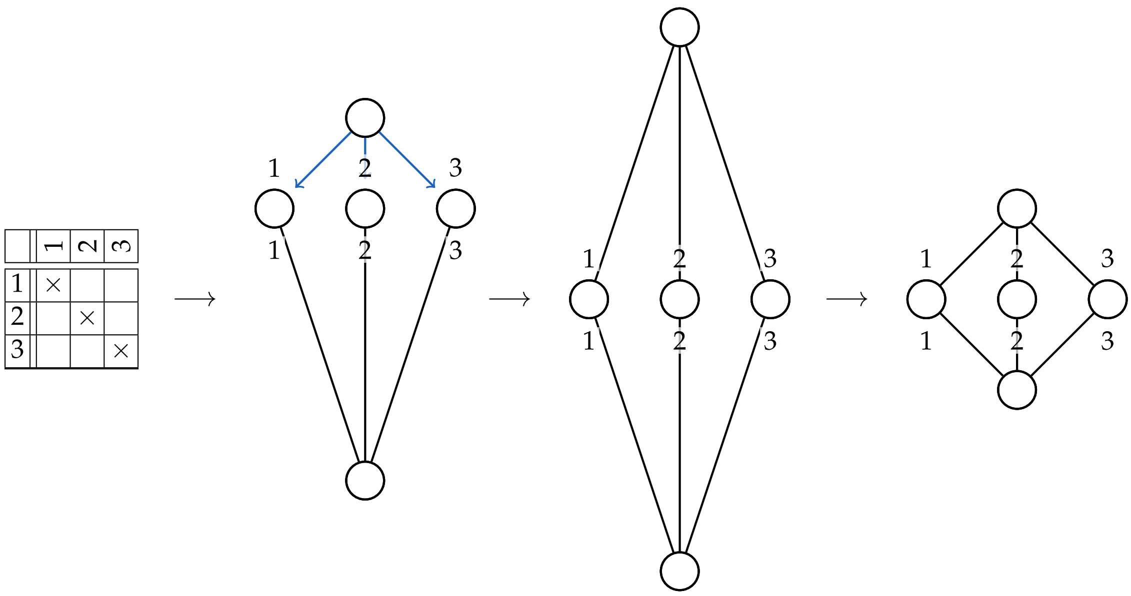


Figure 2.2: A formal context of the three-dimensional nominal scale $\mathbf{N}_3$, an attribute additive line diagram, an hybridized additive line diagram and an vertical adjusted hybridized line diagram of the concept lattice $\mathfrak{B}(\mathbf{N}_3)$.

The seed vectors for the attributes can directly be read off the diagram: They are just the vector from the upper neighbor to the appropriate attribute concept. In the line diagram 2.2 above there are three seed vectors. The attribute 1 has the seed vector $(-1, -1)$, attribute 2 has $(0, -1)$ and attribute 3 has $(1, -1)$. However, the seed vector can only be read off the line diagram in the attribute additive or in the object additive case. This is not possible for hybrid additive diagrams.

To gain more symmetry in the vertical axis, one can compute a VERTICAL HYBRIDIZATION of an attribute additive line diagram. This is done by introducing vertical seed vectors for the irreducible objects, whose vertical coordinates $\mathsf{seed}_y(g)$ are computed by means of a heuristic, *e.g.* conflict distance or symmetry metric *etc.* , or defined by means of the existing attribute seeds $(\mathsf{seed}(m))_{m\in M}$. A good choice is $\mathsf{seed}_y(g) = f\left(\sum_{gIm} \mathsf{seed}_y(m)\right)$ for a suitable function $f\colon \mathbb{R} \to \mathbb{R}$. Here the vectors $(0,-1)$ can be chosen for each object 1, 2 and 3. Then the vertical seed coordinates get adjusted to gain more symmetry and a more compact diagram. Finally these vectors are added to the seeds and the positions are recomputed.

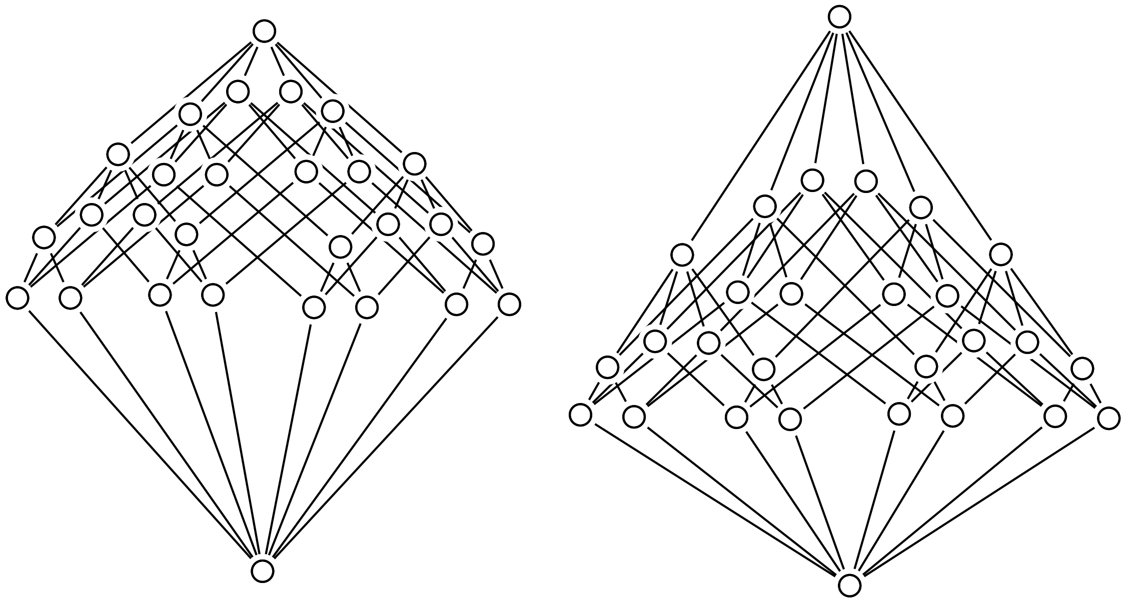


Figure 2.3: Attribute additive and hybridized line diagrams of the concept lattice of the dichotomic scale $\mathbf{D}_3$

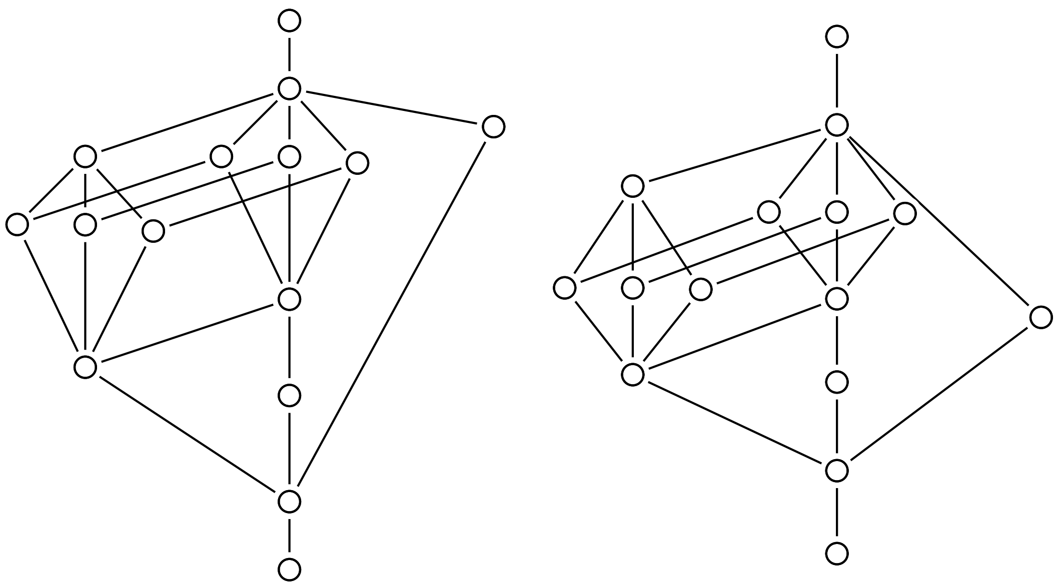


Figure 2.4: Attribute additive and hybridized line diagrams of the concept lattice of a formal context about cognac

2.2.6 Omitting the top and bottom concept node

For each formal context $\mathbf{K} = (G, M, I)$ the top concept node is given by

$$\top = \left(G, G^I, \left\{g \in G \,\middle|\, g^I = G^I\right\}, \left\{m \in M \,\middle|\, m^I = G\right\}\right)$$

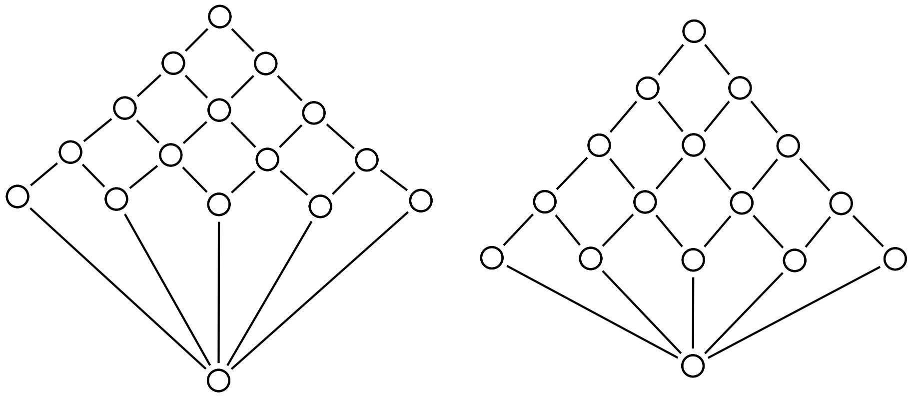


Figure 2.5: Attribute additive and hybridized line diagrams for the interordinal scale $\mathbf{I}_5$

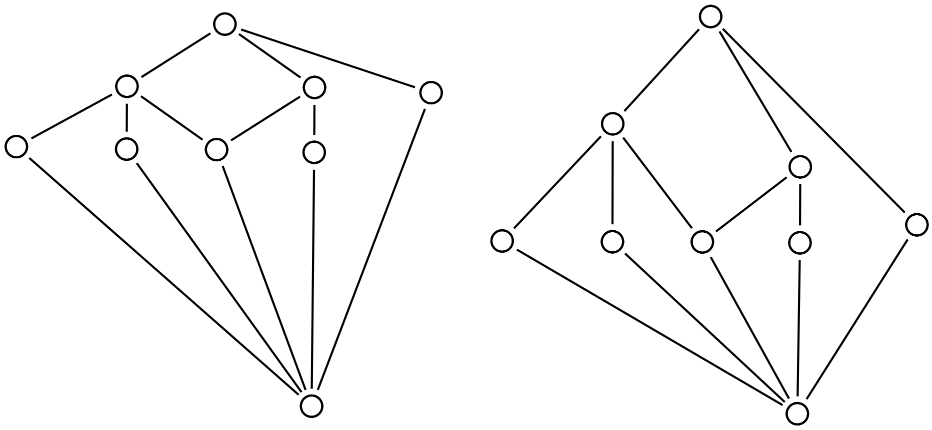


Figure 2.6: Attribute additive and hybridized line diagrams of an artificial formal context

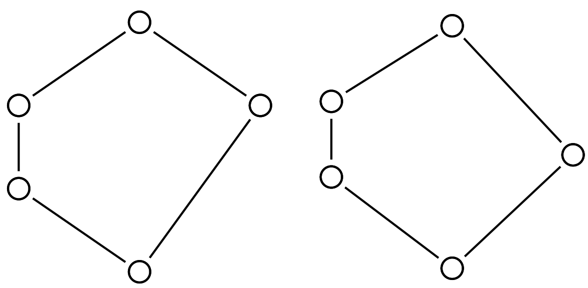


Figure 2.7: Attribute additive and hybridized line diagrams for the $\mathbf{N}_5$ lattice

In certain cases it does not provide any information about $\mathbf{K}$ in the concept diagram $\mathfrak{D}(\mathbf{K})$. This happens when neither any attribute label nor any object label in the top concept node exists. When there is no attribute that is shared by all object, the intent and thus also the set of attribute labels of the top concept node is empty. Furthermore, when the top concept is no object concept, then the set of object labels of the top concept node is empty. In summary, the top concept node can be omitted, if there is no full attribute column in $\mathbf{K}$ and also no object g with $G = g^{II}$ exists. Clearly the maximal elements of the subset $\mathfrak{B}(\mathbf{K}) \setminus \{\top\}$ must then be the coatoms of the concept lattice $\underline{\mathfrak{B}}(\mathbf{K})$.

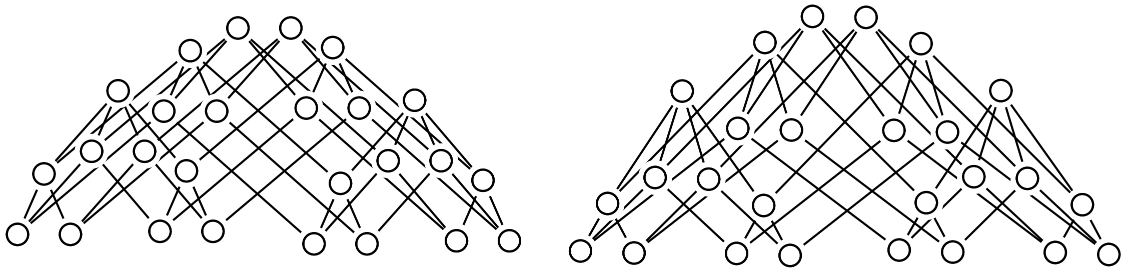


Figure 2.8: Attribute additive and hybridized line diagrams of the concept lattice of the dichotomic scale $\mathbf{D}_3$, with omitted top and bottom concept node

Dually the bottom concept node

$$\bot = \left(M^I, M, \left\{g \in G \,\middle|\, g^I = M\right\}, \left\{m \in M \,\middle|\, m^I = M^I\right\}\right)$$

can be left out in the concept diagram, if there is no full object row and furthermore no attribute m with $M = m^{II}$ exists. Obviously the minimal elements of $\mathfrak{B}(\mathbf{K}) \setminus \{\bot\}$ are then the atoms of $\underline{\mathfrak{B}}(\mathbf{K})$.

2.2.7 Actions on Concept Diagrams

When interacting with concept diagrams, one may want to adjust certain seeds. This is modelled with the map $\mathsf{move}_{\mathsf{seed}}$ that is defined as follows: For a seed element $\sigma \in G \,\dot\cup\, M$ and an adjustment vector $\delta \in \mathbb{R}^2$ let

$$\mathsf{move}_{\mathsf{seed}}((\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}), \sigma, \delta) := (\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}') \text{ and } \mathsf{seed}'(x) := \begin{cases} \mathsf{bound}_\varepsilon(\mathsf{seed}(\sigma) + \delta) & \text{if } x = \sigma \\ \mathsf{seed}(x) & \text{else} \end{cases}.$$

To ensure the upward drawing convention the movement is bound on the y-coordinate by

$$\mathsf{bound}_\varepsilon(x, y) := \begin{cases} (x, y) & \text{if } y < \varepsilon \\ (x, \varepsilon) & \text{else} \end{cases}$$

for a chosen boundary parameter $\varepsilon \in \mathbb{R}_-$. Especially in the hybrid additive concept diagrams the seed vectors cannot be read off the diagram easily, and a direct modification of them is difficult. Thus, one should introduce a mechanism for moving nodes, that preserves the additivity of the concept diagram. This is done with the map $\mathsf{move}_{\mathsf{node}}$ defined as follows: For a concept node $N \in \mathfrak{N}(\mathbf{K})$ and an adjustment vector $\delta \in \mathbb{R}^2$ let

$$\mathsf{move}_{\mathsf{node}}((\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}), N, \delta) := \mathsf{move}_{\mathsf{seed}}(\ldots \mathsf{move}_{\mathsf{seed}}((\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}), x_1, \delta') \ldots), x_k, \delta')$$

with $\mathsf{rep}(N) = \{x_1, \ldots, x_k\}$ and $\delta' := \frac{1}{k} \cdot \delta$. In summary, this can be displayed as

$$\mathsf{move}_{\mathsf{node}}((\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}), N, \delta) = (\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}'')$$

with $\mathsf{seed}''(s) := \begin{cases} \mathsf{bound}_\varepsilon(\mathsf{seed}(x) + \frac{1}{|\mathsf{rep}(N)|} \cdot \delta) & \text{if } x \in \mathsf{rep}(N) \\ \mathsf{seed}(x) & \text{else} \end{cases}.$

2.2.8 Metrics on Concept Diagrams

A METRIC on a concept diagram is a mapping metric, that assigns a non-negative real number to each concept diagram. A metric measures the subjective quality of concept diagrams under different points of view. One can for example take the conflict distance (or conflict avoidance parameter when moving seed or nodes, that involves the diagram growth and is thus bounded) from (Gan) defined by

$$\mathsf{conflictdistance}(\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}) := \bigwedge \left\{ \mathsf{distance}(\mathsf{pos}(\alpha,\beta), \mathsf{pos}(\gamma)) \;\middle|\; \begin{array}{c} \alpha,\beta,\gamma \in \mathfrak{N}(\mathbf{K}) \text{ and } \alpha \neq \gamma \\ \text{and } \beta \neq \gamma \text{ and } \alpha \prec \beta \end{array} \right\}.$$

Also the number of crossing edges yield a suitable quality metric given by

$$\mathsf{edgecrossings}(\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}) := \left| \left\{ \{(\alpha,\beta),(\gamma,\delta)\} \;\middle|\; \begin{array}{c} \alpha,\beta,\gamma,\delta \in \mathfrak{N}(\mathbf{K}) \text{ and } \alpha \neq \gamma \text{ and } \beta \neq \delta \\ \text{and } \alpha \prec \beta \text{ and } \gamma \prec \delta \text{ and } \mathsf{pos}(\alpha,\beta) \cap \mathsf{pos}(\gamma,\delta) \neq \emptyset \end{array} \right\} \right|.$$

Furthermore for measuring the readability of a concept diagram one may consider the number of distinct directions of edges with

$$\mathsf{directioncount}(\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}) := \left| \left\{ \frac{\mathsf{pos}(\beta) - \mathsf{pos}(\alpha)}{\|\mathsf{pos}(\beta) - \mathsf{pos}(\alpha)\|} \;\middle|\; \alpha,\beta \in \mathfrak{N}(\mathbf{K}) \text{ and } \alpha \prec \beta \right\} \right|,$$

and also the minimal angle between edges can be of interest, that yields a metric by

$$\mathsf{minimalangle}(\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}) := \bigwedge \left\{ \mathsf{angle}(\mathsf{pos}(\alpha,\beta), \mathsf{pos}(\alpha,\gamma)) \;\middle|\; \begin{array}{c} \alpha,\beta,\gamma \in \mathfrak{N}(\mathbf{K}) \text{ and } \beta \neq \gamma \\ \text{and } (\alpha \prec \beta \text{ and } \alpha \prec \gamma) \text{ or } (\beta \prec \alpha \text{ and } \gamma \prec \alpha) \end{array} \right\}$$

2.2.9 Heatmaps for Concept Diagrams

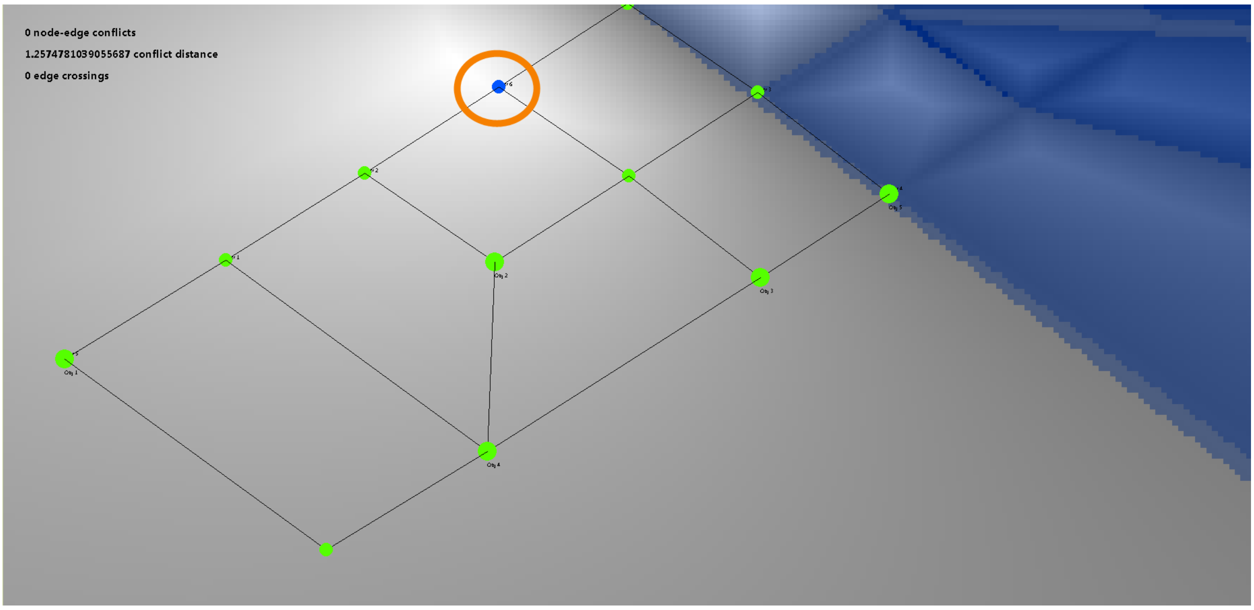


Figure 2.9: Heatmap for the context in figure 4.4 in (GW99) on movement of the left coatom

For a visual assisted adjustment of seeds, a HEATMAP *w.r.t.* to an arbitrary concept diagram metric metric is given as follows. For a seed element σ a heatmap is a function from the real plane (or for practical purposes, a finite subset of the real plane) that assigns to each adjustment vector $\delta \in \mathbb{R}^2$ the value of metric for the appropriate seed adjustment, *i.e.*

$$\mathsf{heatmap}_{\mathsf{seed}}(\mathfrak{D}(\mathbf{K}), \sigma, \mathsf{metric}) : \begin{array}{l} \mathbb{R}^2 \to \mathbb{R} \\ \delta \mapsto \begin{cases} \mathsf{metric}(\mathsf{move}_{\mathsf{seed}}(\mathfrak{D}(\mathbf{K}), \sigma, \delta)) & \text{if } \delta_y < \mathsf{seed}_y(\sigma) \\ 0 & \text{else} \end{cases} \end{array}$$

Furthermore, heatmaps can also be defined for moving concept nodes. A benefit is then the possibility to draw the heatmap in the background of the concept diagram to give advices where to place the chosen concept node. Let $N \in \mathfrak{D}(\mathbf{K})$ be the moving concept node with its original position $\pi := \mathsf{pos}(N)$ before the movement, then the heatmap is defined as

$$\mathsf{heatmap}_{\mathsf{node}}(\mathfrak{D}(\mathbf{K}), N, \mathsf{metric}) : \begin{array}{l} \mathbb{R}^2 \to \mathbb{R} \\ \pi + \delta \mapsto \begin{cases} \mathsf{metric}(\mathsf{move}_{\mathsf{node}}(\mathfrak{D}(\mathbf{K}), N, \delta)) & \text{if } \delta_y < \bigwedge_{N_1 > N} \mathsf{pos}_y(N_1) - \mathsf{pos}_y(N) \\ 0 & \text{else} \end{cases} \end{array}$$

Heatmaps on concept lattice diagrams were abstractly introduced in (Gan) by means of conflict charts *w.r.t.* node edge distances (conflict distance). Three examples for heatmaps are shown in figures 2.9, 2.10 and 2.11. The pictures were produced with a multi-threaded toy prototype, written as an Eclipse plugin.

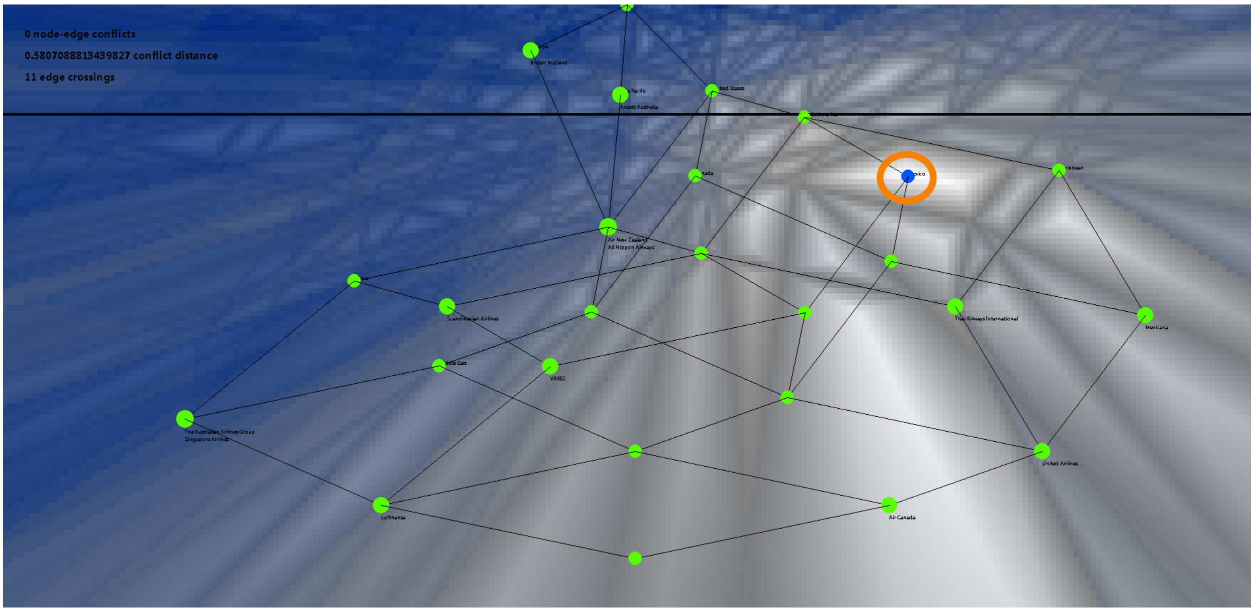


Figure 2.10: Heatmap for a context about airlines on movement of the encircled node

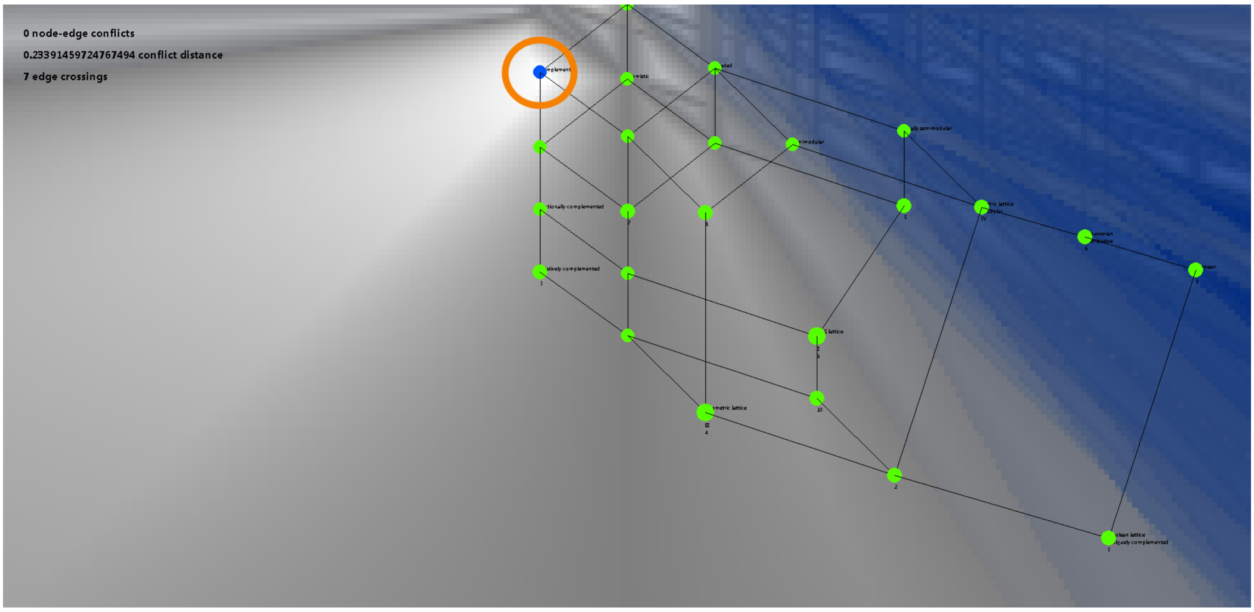


Figure 2.11: Heatmap for a context about lattices on movement of the encircled left coatom

2.2.10 Biplots of Concept Diagrams

In a paper from (GG12) the technique of an ordinal factorization of a formal context is introduced.

A FACTORISATION of a formal context (G, M, I) consists of two FACTORISATION CONTEXTS (G, F, I_{GF}) and (F, M, I_{FM}) such that an object g has an attribute m in (G, M, I) iff there is a FACTOR $f \in F$ with $g\ I_{GF}\ f$ and $f\ I_{FM}\ m$. We then write $(G, M, I) = (G, F, I_{GF}) \circ (F, M, I_{FM})$. For a subset $E \subseteq F$ the subcontext $(G, E, I_{GF} \cap (G \times E))$ is called a (MANY-VALUED) FACTOR of (G, M, I). A factor (G, F, I_{GF}) of (G, M, I) is called an **S**-FACTOR, if it has a surjective full **S**-measure. If **S** is an elementary ordinal scale, one also speaks of an ORDINAL factor. Moreover, one says that (G, M, I) has an ORDINAL FACTORISATION if it has a first factorising context that can be written as an apposition of ordinal factors.
A proposition in (GG12) states that a formal context is an ordinal factor of (G, M, I), iff its attribute extents are a linearly ordered family of concept extents of (G, M, I).
Each context has a trivial factorisation $(G, M, I) = (G, M, I) \circ (M, M, \rightarrow)$ with factors in M. Choosing the irreducible attributes $M_{\text{irr}} \subseteq M$ gives the ATTRIBUTE FACTORISATION

$$(G, M, I) = (G, M_{\text{irr}}, I \cap (G \times M_{\text{irr}})) \circ (M_{\text{irr}}, M, \rightarrow \cap (M_{\text{irr}} \times M)).$$

By computing a chain decomposition $M_1, \ldots, M_k$ of the attribute order $(M_{\text{irr}}, \rightarrow)$ an ordinal factorization can be obtained. Indeed, each factor $(G, M_j, I \cap (G \times M_j))$ is then ordinal, and obviously (G, M, I) can be written as an apposition of the ordinal factors $(G, M_j, I \cap (G \times M_j))$. Also (GG12) introduce the BIPLOT visualization of the conceptual data obtained from a formal context by chosing two ordinal factors as axes of an ordinary x-y-chart and projecting the formal concepts onto these factors to gain the coordinates. When choosing ordinal factors computed from an attribute chain decomposition, one can label the axis with the appropriate attributes. Furthermore, nodes are drawn at the coordinates of the projected concepts and labeling them with the corresponding object labels. It remains to investigate, whether the projected concepts should be drawn upon or beneath each other. All edges between concepts are omitted for a clearer structure.

$$\mathsf{biplot}((\mathfrak{N}(\mathbf{K}), \prec, \mathsf{seed}), M_1, M_2) := (\mathfrak{N}(\mathbf{K}), \emptyset, \mathsf{seed}''') \text{ and } \mathsf{seed}'''(m) := \begin{cases} e_j & \text{if } m \in M_j \\ 0 & \text{else} \end{cases}$$

Finally one could use remaining ordinal factors in other chart dimensions, *e.g.* node size, node color, node shape *etc.* Also nominal factors can be displayed. One should for the sake of readability provide legends beneath the chart for such additional dimensions.

2.2.11 Seeds Selection

Seed vectors can be chosen in many different ways.

Context Rearrangement

Suppose two enumerations $\mathsf{enum}_G \colon G \hookrightarrow\!\!\!\!\rightarrow \{0, 1, \ldots, |G| - 1\}$ and $\mathsf{enum}_M \colon M \hookrightarrow\!\!\!\!\rightarrow \{0, 1, \ldots, |M| - 1\}$ for the object set G and the attrribute set M is given. One can consider simply the position of the objects and the attributes in the corresponding cross table. Then the CENTER of an object row if given

$$\mathsf{center}_G(g) := \frac{1}{|g^I|} \cdot \sum_{gIm} \mathsf{enum}_M(m)$$

and dually the CENTER of an attribute column is defined as

$$\mathsf{center}_M(m) := \frac{1}{|m^I|} \cdot \sum_{gIm} \mathsf{enum}_G(g).$$

The contexts rows are rearranged by ascending centers, then dually the columns are rearranged by ascending centers. This procedure is repeated until no further changes occur, or a cycle is entered. In the end one can choose the seed vectors according to the attribute enumeration. This technique ensures that in many cases the horizontal coordinates of incident objects and attributes do not differ too much, and can reduce the number of edge crosses. It can also be easily implemented.

Spectral Decomposition

For a formal context $\mathbf{K}$ transform its concept lattice $\underline{\mathfrak{B}}(\mathbf{K}) = (\mathfrak{B}(\mathbf{K}), \leq)$ into an undirected graph $(V, E) = (\mathfrak{B}(\mathbf{K}), \leq \cup \geq)$ and compute its laplacian matrix $L := D - A \in \mathbb{R}^{V \times V}$ by means of the adjacency matrix

$A \in \mathbb{R}^{V \times V}$ and degree matrix $D \in \mathbb{R}^{V \times V}$ that are given by

$$A := \left(\begin{cases} 1 & \text{if } \{v, w\} \in E \text{ and } v \neq w \\ 0 & \text{else} \end{cases} \right)_{v,w \in V} \quad \text{and } D := \mathsf{diag}(\mathsf{deg}(v))_{v \in V}.$$

The laplacian matrix is symmetric and has thus exactly $|V|$ real eigenvalues and corresponding real eigenvectors. Then two eigenvectors are chosen, one for the horizontal coordinates and the other for the vertical coordinates. In figure 2.12 a toy example of a boolean cube $\mathbf{B}_5$ is shown that was drawn with this technique. However, more sophisticated details remains for future research. See (GR04) or (Ros04) for further details.

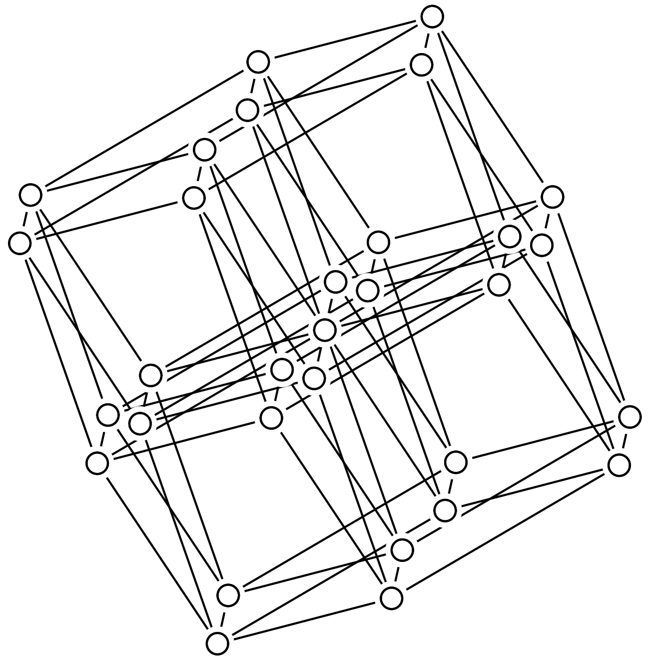


Figure 2.12: Concept diagram of the five-dimensional cube with its appropriate boolean context $\mathbf{B}_5$

Chain Decomposition

Another possibility is given by means of a chain decomposition of the attribute order $(M, \rightarrow)$ where

$$m \rightarrow n :\Leftrightarrow \mu m \leq \mu n \Leftrightarrow m^I \subseteq n^I.$$

Let $(P, \leq)$ be an arbitrary ordered set. A CHAIN DECOMPOSITION $\{C_t\}_{t \in T}$ of $(P, \leq)$ is a partition of P, such that every partition class C_t forms a chain in $(P, \leq)$.

$$\forall_{t_1 \neq t_2}\, C_{t_1} \cap C_{t_2} = \emptyset$$
$$\bigcup_{t \in T} C_t = P$$
$$\forall_{t \in T}\, \forall_{p_1, p_2 \in C_t}\, p_1 \leq p_2 \text{ or } p_2 \leq p_1$$

A MINIMAL chain decomposition is a chain decomposition with k chains, such that there is no chain decomposition with less than k chains. The cardinality of a minimal chain decomposition is also called the CHAIN COVERING NUMBER of the poset $(P, \leq)$. DILWORTH's theorem states that the width, *i.e.* the cardinality of a maximal antichain, of each ordered set equals its chain covering number.
For a context $\mathbf{K} = (G, M, I)$ an ATTRIBUTE CHAIN DECOMPOSITION is a chain decomposition $\{M_t\}_{t \in T}$ of the attribute order $(M_{\text{irr}}, \rightarrow)$. From an attribute chain decomposition a seed function can be obtained by choosing a suitable chain seed function $\mathsf{seed}' \colon T \rightarrow \mathbb{R}^2$. This yields a seed function seed for a concept diagram with $\mathsf{seed}(m) := \mathsf{seed}'(t)$ with $m \in M_t$. Good practical results can be achieved by putting the longest chain in the middle, *i.e.* with null horizontal coordinate, and sort the other chains sideways, descending by size.
Furthermore, when moving a concept node within a concept diagram with chain seeds, one can decide to move just like in the ordinary case, or to move the whole chain as well to not break the chain visualization.

Subdirect Decomposition

According to (GW99) the subdirect irreducible factors can be computed by means of the context $\mathfrak{B}(G, M, \swarrow\!\!\!\nearrow)$. In a doubly founded context a subcontext (H, N) is compatible iff it is arrow closed iff $(\complement H, N)$ is a concept of $\mathfrak{B}(G, M, \swarrow\!\!\!\nearrow)$. Furthermore the subdirect decompositions of (G, M, I) are in a one-to-one correspondence to the families $\{(G_t, M_t)\}_{t\in T}$ of arrow-closed subcontexts covering the context, *i.e.* $G = \bigcup_{t\in T} G_t$ and $M = \bigcup_{t\in T} M_t$.

When computing a diagram of a subdirect irreducible factor, the seeds can be used to obtain a diagram of the whole context. As the diagram is smaller it is easier to compute a good diagram. This leads to a divide and conquer technique for producing concept diagrams. However, it has to be ensured when mixing seed maps from different factors, that the obtained diagram is good as well.

A small subdirect irreducible decomposition can be produced by looking at the attribute order of $\mathfrak{B}(G, M, \swarrow\!\!\!\nearrow)$. First, it holds

$$\bigcup_{m\in M} m^{\swarrow\!\!\!\nearrow\swarrow\!\!\!\nearrow} = M \text{ and } \bigcap_{m\in M} m^{\swarrow\!\!\!\nearrow} = M^{\swarrow\!\!\!\nearrow} = \emptyset$$

since (G, M) is trivially arrow closed in (G, M, I), thus $(G \setminus G, M) = (\emptyset, M)$ must be the smallest concept of $(G, M, \swarrow\!\!\!\nearrow)$. Hence, choose a minimal subset $N \subseteq M$ such that

$$\bigcap_{m\in N} m^{\swarrow\!\!\!\nearrow} = \emptyset,$$

then $\left\{(G \setminus n^{\swarrow\!\!\!\nearrow}, n^{\swarrow\!\!\!\nearrow\swarrow\!\!\!\nearrow})\right\}_{n\in N}$ is a subdirect decomposition into subdirect irreducible factors. The simplest way is to choose N as the set of the minimal elements in the attribute order $(M, \rightarrow_{\swarrow\!\!\!\nearrow})$ of $\mathfrak{B}(G, M, \swarrow\!\!\!\nearrow)$. Then, concept diagrams for these subcontext are computed and merged into a concept diagram of the whole concept diagram of $\mathbf{K}$.

2.3 Apposition of Contexts

Definitio: Apposition 2.9

Let (G, M, I) and (G, N, J) be two contexts with disjoint attribute sets, *i.e.* $M \cap N = \emptyset$. Then their APPOSITION is defined as

$$(G, M, I)|(G, N, J) := (G, M \,\dot\cup\, N, I \,\dot\cup\, J).$$

Lemma: Rows and Columns in Apposition Context 2.10

Let (G, M, I) and (G, N, J) be two contexts with disjoint attribute sets. Then we have the following equations for objects $g \in G$ and attributes $m \in M$ and $n \in N$:

(I) $g(I \,\dot\cup\, J)m \Leftrightarrow gIm$ and $g(I \,\dot\cup\, J)n \Leftrightarrow gJn$

(II) $g^{I\dot\cup J} = g^I \,\dot\cup\, g^J$

(III) $m^{I\dot\cup J} = m^I$ and $n^{I\dot\cup J} = n^J$

APPROBATIO (I) This is obvious, since by construction of an apposition we have $(I \,\dot\cup\, J) \cap (G \times M) = I$ and dually $(I \,\dot\cup\, J) \cap (G \times N) = J$.

(II) For an object $g \in G$ we have

$$g^{I\dot\cup J} = \{m \in M \,\dot\cup\, N \mid g(I \,\dot\cup\, J)m\} = \{m \in M \,\dot\cup\, N \mid gIm \,\dot\vee\, gJm\} = \{m \in M \mid gIm\} \,\dot\cup\, \{m \in N \mid gJm\} = g^I \,\dot\cup\, g^J$$

(III) This follows from (i). ∎

2.11 **Lemma: Common Rows and Common Columns in Apposition Context**

Let (G, M, I) and (G, N, J) be two contexts with disjoint attribute sets. Then we have the following equations for object sets $A \subseteq G$ and attribute sets $B \subseteq M \dot\cup N$, $D \subseteq M$ and $F \subseteq N$:

(I) $A^{I\dot\cup J} \cap M = A^I$
and $A^{I\dot\cup J} \cap N = A^J$
and $A^{I\dot\cup J} = A^I \dot\cup A^J$

(II) $D^{I\dot\cup J} = D^I$
and $F^{I\dot\cup J} = F^J$
and $B^{I\dot\cup J} = (B \cap M)^I \cap (B \cap N)^J$

(III) $A^{I(I\dot\cup J)} = (A^{I\dot\cup J} \cap M)^I = A^{II}$
and $A^{J(I\dot\cup J)} = (A^{I\dot\cup J} \cap N)^J = A^{JJ}$
and $A^{(I\dot\cup J)(I\dot\cup J)} = A^{II} \cap A^{JJ}$

(IV) $D^{I(I\dot\cup J)} = D^{II} \dot\cup D^{IJ}$
and $F^{J(I\dot\cup J)} = F^{JI} \dot\cup F^{JJ}$
and $B^{(I\dot\cup J)(I\dot\cup J)} = ((B \cap M)^I \cap (B \cap N)^J)^I \dot\cup ((B \cap M)^I \cap (B \cap N)^J)^J$

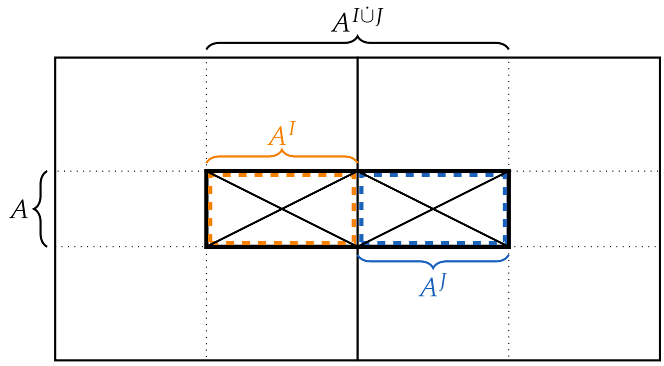


Figure 2.13: schema for closure of object sets

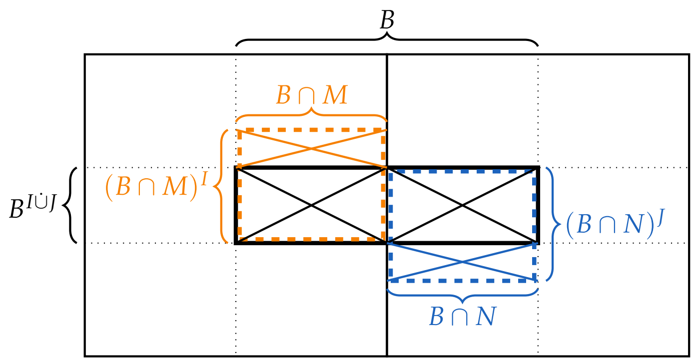


Figure 2.14: schema for closure of attribute sets

APPROBATIO (I) Let $A \subseteq G$ be an object set. Then it holds that

$$A^{I\dot\cup J} \cap M = \bigcap_{g \in A} g^{I\dot\cup J} \cap M = \bigcap_{g \in A} (g^I \dot\cup g^J) \cap M = \bigcap_{g \in A} g^I = A^I.$$

Dually we have $A^{I\dot\cup J} \cap N = A^J$. Furthermore we conclude

$$A^{I\dot\cup J} = (A^{I\dot\cup J} \cap M) \dot\cup (A^{I\dot\cup J} \cap N) = A^I \dot\cup A^J.$$

Figure 2.13 shows what happens in the context when deriving an object set A.

(II) Let now $D \subseteq M$ be an attribute set. Then

$$D^{I\dot{\cup}J} = \bigcap_{m \in D} m^{I\dot{\cup}J} = \bigcap_{m \in D} m^{I} = D^{I}$$

and dually $F^{I\dot{\cup}J} = F^{I}$. Furthermore for an attribute set $B \subseteq M \dot{\cup} N$ we have

$$B^{I\dot{\cup}J} = ((B \cap M) \dot{\cup} (B \cap N))^{I\dot{\cup}J} = (B \cap M)^{I\dot{\cup}J} \cap (B \cap N)^{I\dot{\cup}J} = (B \cap M)^{I} \cap (B \cap N)^{J}.$$

(III) For a set of objects $A \subseteq G$ it holds

$$A^{I(I\dot{\cup}J)} = (A^{I} \cap M)^{I} \cap (A^{I} \cap N)^{J} = A^{II} \cap \emptyset^{J} = A^{II} \cap G = A^{II}$$

and dually $A^{J(I\dot{\cup}J)} = A^{JJ}$. Also we have

$$(A^{(I\dot{\cup}J)} \cap M)^{I} = ((A^{I} \dot{\cup} A^{J}) \cap M)^{I} = A^{II}.$$

and $(A^{(I\dot{\cup}J)} \cap N)^{J} = A^{JJ}$ dually. It then follows that

$$A^{(I\dot{\cup}J)(I\dot{\cup}J)} = (A^{I} \dot{\cup} A^{J})^{(I\dot{\cup}J)} = A^{I(I\dot{\cup}J)} \cap A^{J(I\dot{\cup}J)} = A^{II} \cap A^{JJ}.$$

(IV) We have $D^{I(I\dot{\cup}J)} = D^{II} \dot{\cup} D^{IJ}$ by (I). Dually it follows that $F^{J(I\dot{\cup}J)} = F^{JI} \dot{\cup} F^{JJ}$. As a conclusion we get

$$B^{(I\dot{\cup}J)(I\dot{\cup}J)} = ((B \cap M)^{I} \cap (B \cap N)^{J})^{(I\dot{\cup}J)} = ((B \cap M)^{I} \cap (B \cap N)^{J})^{I} \dot{\cup} ((B \cap M)^{I} \cap (B \cap N)^{J})^{J}.$$

■

We recall the definition of a dense subcontext in this special case: (G, M, I) is dense in $(G, M, I)|(G, N, J)$ iff μM is $\bigwedge$-dense in the concept lattice of $(G, M, I)|(G, N, J)$. Trivially γG is $\bigvee$-dense. By Lemma 2.7 (G, M, I) is dense iff $B^{(I\dot{\cup}J)} = (B \cap M)^{(I\dot{\cup}J)} (= (B \cap M)^{I})$ holds for all attribute sets $B \subseteq M \dot{\cup} N$. Again, $A^{(I\dot{\cup}J)} = (A \cap G)^{(I\dot{\cup}J)}$ trivially holds for $A \subseteq G$. The context (G, N, J) is called REDUNDANT in $(G, M, I)|(G, N, J)$ iff (G, M, I) is dense in $(G, M, I)|(G, N, J)$.

Theorema: Embedding into Apposition Lattice 2.12

Let (G, M, I) and (G, N, J) be two contexts with disjoint attribute sets. Then every extent of (G, M, I) is also an extent of $(G, M, I)|(G, N, J)$ and

$$\phi\colon \begin{array}{l} \mathfrak{B}(G, M, I) \hookrightarrow \mathfrak{B}(G, M, I)|(G, N, J) \\ (A, B) \mapsto (A, A^{(I\dot{\cup}J)}) = (A, B \dot{\cup} A^{J}) \end{array}$$

is a $\bigwedge$-preserving order-embedding. Furthermore, if (G, M, I) is dense in $(G, M, I)|(G, N, J)$, then vice versa every extent of $(G, M, I)|(G, N, J)$ is an extent of (G, M, I) as well and ϕ is an isomorphism. The inverse mapping is then given by

$$\phi^{-1}\colon (A, B) \mapsto (A, B \cap M).$$

APPROBATIO Each extent of (G, M, I) has the form B^{I} for some attribute set $B \subseteq M$. By Lemma: Common Rows and Common Columns in Apposition Context 2.11 $B^{(I\dot{\cup}J)} = B^{I}$ then always hold and so B^{I} must also be an extent of $(G, M, I)|(G, N, J)$. Thus ϕ is well-defined in the sense that each ϕ-image of a (G, M, I)-concept is a $(G, M, I)|(G, N, J)$-concept. Again by the preceding Lemma: Common Rows and Common Columns in Apposition Context 2.11, $A^{(I\dot{\cup}J)} = A^{I} \dot{\cup} A^{J} = B \dot{\cup} A^{J}$ hold for the intents. As ϕ does not change the extent, it clearly must be an order-embedding. By Theorema: Concept Lattice 2.5 every infimum can be found by just intersecting extents, thereby ϕ is $\bigwedge$-preserving.
Finally let (G, M, I) be dense in $(G, M, I)|(G, N, J)$, *i.e.* $B^{(I\dot{\cup}J)} = (B \cap M)^{I}$ holds for every $B \subseteq M \dot{\cup} N$ as above. Clearly each extent $B^{(I\dot{\cup}J)}$ of $(G, M, I)|(G, N, J)$ must then be an extent of (G, M, I) as well. Furthermore ϕ is a surjection: Let $(A, B) \in \mathfrak{B}(G, M, I)|(G, N, J)$, then $(A, B \cap M)$ is a concept of (G, M, I) as

$A^I = A^{(I \dot\cup J)} \cap M = B \cap M$ and $(B \cap M)^I = B^{(I \dot\cup J)} = A$, and $\phi(A, B \cap M) = (A, A^{(I \dot\cup J)}) = (A, B)$ holds. In summary ϕ is a surjective order-embedding, *i.e.* an order-isomorphism and a lattice-isomorphism. Moreover $(A, B) \mapsto (A, B \cap M)$ is indeed the inverse of ϕ as

$$\phi^{-1}\phi(A, B) = \phi^{-1}(A, B \dot\cup A^J) = (A, (B \dot\cup A^J) \cap M) = (A, B)$$

and

$$\phi\phi^{-1}(A, B) = \phi(A, B \cap M) = (A, \underbrace{(B \cap M) \dot\cup A^J}_{\substack{=(A^{(I\dot\cup J)} \cap M) \dot\cup A^J \\ =A^I \dot\cup A^J \\ =A^{(I \dot\cup J)} \\ =B}}) = (A, B)$$

■ hold by Lemma: Common Rows and Common Columns in Apposition Context 2.11.

Theorema: Nested Concept Lattice 2.13

The concept lattice of a context apposition can be embedded in the direct product of the single concept lattices. Formal: For two contexts (G, M, I) and (G, N, J) the mapping

$$\psi\colon \begin{array}{l} \underline{\mathfrak{B}}(G,M,I)|(G,N,J) \hookrightarrow\!\!\!\!\to \underline{\mathfrak{B}}(G,M,I) \boxtimes \underline{\mathfrak{B}}(G,N,J) \\ \qquad (A,B) \mapsto ((A^{II},A^{I}),(A^{JJ},A^{J})) = (((B\cap M)^{I}, B\cap M),((B\cap N)^{J}, B\cap N)) \end{array}$$

with $\mathfrak{B}(G,M,I) \boxtimes \mathfrak{B}(G,N,J) \subseteq \mathfrak{B}(G,M,I) \times \mathfrak{B}(G,N,J)$ and

$$((A,B),(C,D)) \in \mathfrak{B}(G,M,I) \boxtimes \mathfrak{B}(G,N,J) :\Leftrightarrow (A\cap C, B \dot\cup D) \in \mathfrak{B}(G,M,I)|(G,N,J)$$

is an isomorphism. The inverse mapping of ψ is given by

$$\psi^{-1}\colon ((A,B),(C,D)) \mapsto (A\cap C, B\dot\cup D).$$

The set $\mathfrak{B}(G,M,I) \boxtimes \mathfrak{B}(G,N,J)$ together with the inherited coordinate-wise order is a complete lattice and is called NESTED CONCEPT LATTICE of (G,M,I) and (G,N,J).

APPROBATIO The order of $\mathfrak{B}(G,M,I) \boxtimes \mathfrak{B}(G,N,J)$ is the inherited coordinate-wise order from the cartesian product $\mathfrak{B}(G,M,I) \times \mathfrak{B}(G,N,J)$. The supremum equals the coordinate-wise supremum in the cartesian product, as can be seen on the intents: $(\bigcap_{t\in T} B_t) \dot\cup (\bigcap_{t\in T} D_t) = \bigcap_{t\in T}(B_t \dot\cup D_t)$ always hold for attribute sets $B_t \subseteq M$ and $D_t \subseteq N$ for all $t \in T$. So the supremum in $\mathfrak{B}(G,M,I) \boxtimes \mathfrak{B}(G,N,J)$ exist for all subsets of $\mathfrak{B}(G,M,I) \boxtimes \mathfrak{B}(G,N,J)$ and it is indeed a complete lattice.
Now let (A,B) be a concept of the apposition $(G,M,I)|(G,N,J)$, then $A^{II} \cap A^{JJ} = A^{(I\dot\cup J)(I\dot\cup J)} = A$ and $A^{I} \dot\cup A^{J} = A^{(I\dot\cup J)} = B$ hold by Lemma: Common Rows and Common Columns in Apposition Context 2.11 and thereby $\psi(A,B)$ is an element of the nested product $\mathfrak{B}(G,M,I) \boxtimes \mathfrak{B}(G,N,J)$. So ψ is well-defined. For each concept $(A,B) \in \mathfrak{B}(G,M,I)|(G,N,J)$ it follows by the same arguments as above

$$\psi^{-1}\psi(A,B) = \psi^{-1}((A^{II},A^{I}),(A^{JJ},A^{J})) = (A^{II}\cap A^{JJ}, A^{I}\dot\cup A^{J}) = (A,B).$$

Now let $((A,B),(C,D)) \in \mathfrak{B}(G,M,I) \boxtimes \mathfrak{B}(G,N,J)$, then it holds that

$$\psi\psi^{-1}((A,B),(C,D)) = \psi(A\cap C, B\dot\cup D) = (((A\cap C)^{II},(A\cap C)^{I}),((A\cap C)^{JJ},(A\cap C)^{J})).$$

By Lemma: Common Rows and Common Columns in Apposition Context 2.11 $(A\cap C)^{I} = (A\cap C)^{(I\dot\cup J)} \cap M = (B\dot\cup D)\cap M = B$ and thus $(A\cap C)^{II} = B^{I} = A$ as well. Analogously for (C,D) in (G,N,J). In summary ψ is a bijection.
ψ is order-preserving as $A \subseteq C$ always implies $A^{I} \supseteq C^{I}$ and $A^{J} \supseteq C^{J}$ and thus $\psi(A,B) \leq \psi(C,D)$ hold for all concepts $(A,B) \leq (C,D)$. Overthis ψ is order-reversing since $A^{I} \supseteq C^{I}$ and $A^{J} \supseteq C^{J}$ implies

$$B = A^{(I\dot\cup J)} = A^{I}\dot\cup A^{J} \supseteq C^{I}\dot\cup C^{J} = C^{(I\dot\cup J)} = D.$$

So ψ is an order-isomorphism and a lattice-isomorphism.
When using ψ just as an embedding into the cartesian product $\mathfrak{B}(G,M,I) \times \mathfrak{B}(G,N,J)$, then ψ is only an order-embedding and overthis $\bigvee$-preserving as can be seen on the intents. ■

3 Incremental Updates for Concept Diagrams

3.1 Insertion & Removal of a single Attribute Column

When visualizing a concept lattice by a line diagram, some interaction techniques are needed. In this section an algorithm for the insertion and removal of single attributes is constructed and proven.
Throughout the whole section let $\mathbf{K} = (G, M, I)$ be an arbitrary finite formal context. Let further be $n \notin M$ any new attribute with its attribute extent $n^J \subseteq G$ or incidence $J \subseteq G \times \{n\}$ respectivelly, *i.e.* $\mathbf{C} = (G, \{n\}, J)$ is also a context, called COLUMN. Their apposition is symbolized as

$$\mathbf{K}|\mathbf{C} = (G, M, I)|(G, \{n\}, J) = (G, M \dot{\cup} \{n\}, I \dot{\cup} J)$$

The context neither has to be clarified nor reduced. The initial point of view is the Insertion of a single attribute, thus $\mathbf{K}$ is called the old context and $\mathbf{K}|\mathbf{C}$ the new context.
The update process of the line diagram is split up in four parts: concepts, neighborhood, labels and seeds.
In certain cases there is no big change in the line diagram, *viz.* when the set of extents of the updated context $\mathbf{K}|\mathbf{C}$ equals the set of extents of the preceding context $\mathbf{K}$. Then the column $\mathbf{C}$ and the attribute n are called REDUNDANT in $\mathbf{K}|\mathbf{C}$. Two other strongly related definitions are recalled: Two attributes of a context are called EQUIVALENT, iff they have the same attribute extents. An attribute is called REDUCIBLE, iff the appropriate attribute concept is $\bigwedge$-reducible in the concept lattice. More formally: $m \in M$ is reducible in $\mathbf{K}$, iff $\mu m = (m^I, m^{II})$ is $\bigwedge$-reducible in $\mathfrak{B}(\mathbf{K})$, so, iff

$$\mu m = (\mu m)^* = \bigwedge_{\substack{(A,B)\in\mathfrak{B}(\mathbf{K}) \\ \mu m<(A,B)}} (A, B) = \bigwedge_{\substack{n\in M \\ \mu m<\mu n}} \mu n$$

and thus, iff $m^I = \bigcap_{\substack{n\in M \\ m^I\subset n^I}} n^I$. The following lemma shows that a redundant attribute is either equivalent to another attribute or is reducible by some other attributes, and vice versa each equivalent or reducible attribute is redundant.

3.1 **Lemma: Characterization of Attribute Redundancy**

The following statements are equivalent:

(I) n is redundant in $\mathbf{K}|\mathbf{C}$

(II) n^J is an extent of $\mathbf{K}$

(III) n has an equivalent attribute or is reducible in $\mathbf{K}|\mathbf{C}$

(IV) $\mathbf{K}$ is dense in $\mathbf{K}|\mathbf{C}$

APPROBATIO (I)⇒(II) When n is redundant in $\mathbf{K}|\mathbf{C}$, then both contexts $\mathbf{K}|\mathbf{C}$ and $\mathbf{K}$ have the same set of extents. Thus $n^{(I\dot\cup J)} = n^J$ must clearly be an extent of $\mathbf{K}$.

(I)⇐(II) Let n^J be an extent of $\mathbf{K}$, *i.e.* $n^J = D^I$ for a suitable $D \subseteq M$. Each extent of $\mathbf{K}$ is obviously an extent of $\mathbf{K}|\mathbf{C}$, as $B^I = B^{(I\dot\cup J)}$ holds for all $B \subseteq M$ by Lemma: Common Rows and Common Columns in Apposition Context 2.11. Now let A be an extent of $\mathbf{K}|\mathbf{C}$, *i.e.* there is an attribute set $B \subseteq M \dot\cup \{n\}$ with $A = B^{(I\dot\cup J)}$. In case $n \notin B$ it clearly follows that $B^{(I\dot\cup J)} = B^I$. Otherwise $B^{(I\dot\cup J)} = (B \setminus \{n\})^I \cap n^J = B^I \cap D^I = (B \cup D)^I$. In both cases A is already an extent of $\mathbf{K}$. In summary both contexts have exactly the same set of extents.

(II)⇒(III) If the new attribute extent n^J is already an extent of $\mathbf{K}$, then there is an attribute set $B \subseteq M$ with

$$n^{(I\dot\cup J)} = n^J = B^I = \bigcap_{m\in B} m^I = \bigcap_{m\in B} m^{(I\dot\cup J)}.$$

Either an attribute $m \in B$ exists such that $n^{(I\dot\cup J)} = m^{(I\dot\cup J)}$, *i.e.* n and m are equivalent in $\mathbf{K}|\mathbf{C}$, or $n^{(I\dot\cup J)} \neq m^{(I\dot\cup J)}$ holds for all attributes $m \in B$, and as $m \in B = A^I \Leftrightarrow n^{(I\dot\cup J)} = n^J = A \subseteq m^I = m^{(I\dot\cup J)}$

$$n^{(I\dot\cup J)} = \bigcap_{m\in B} m^{(I\dot\cup J)} = \bigcap_{\substack{m\in M\\ n^{(I\dot\cup J)}\subset m^{(I\dot\cup J)}}} m^{(I\dot\cup J)},$$

i.e. n is reducible (by B) in $\mathbf{K}|\mathbf{C}$.

(II)⇐(III) If n has an equivalent attribute $m \in M$, then $n^J = n^{(I\dot\cup J)} = m^{(I\dot\cup J)} = m^I$ hold, so n^J is obviously an extent of $\mathbf{K}$. Otherwise n is reducible in $\mathbf{K}|\mathbf{C}$, *i.e.* the appropriate attribute concept $\mu n = (n^{(I\dot\cup J)}, n^{(I\dot\cup J)(I\dot\cup J)})$ is $\bigwedge$-reducible in the concept lattice $\mathfrak{B}(\mathbf{K}|\mathbf{C})$, *i.e.* it is the infimum of all proper super concepts of $\mathbf{K}|\mathbf{C}$. As all attribute concepts of $\mathbf{K}|\mathbf{C}$ make up a $\bigwedge$-dense set in $\mathfrak{B}(\mathbf{K}|\mathbf{C})$ it holds that

$$\mu n = \bigwedge_{\substack{m\in M\\ \mu n<\mu m}} \mu m.$$

By Lemma: Rows and Columns in Apposition Context 2.10 this attribute concept μn has the form $(n^J, n^{J(I\dot\cup J)})$ and each attribute concept μm has the form $(m^I, m^{I(I\dot\cup J)})$ for all other attributes $m \in M$. Thus by looking on the extents

$$n^J = \bigcap_{\substack{m\in M\\ n^J\subset m^I}} m^I = (\bigcup_{\substack{m\in M\\ n^J\subset m^I}} \{m\})^I = \left\{m \in M \,\middle|\, n^J \subset m^I\right\}^I.$$

So n^J is an extent of $\mathbf{K}$.

(III)⇒(IV) If n has an equivalent attribute or is reducible in $\mathbf{K}|\mathbf{C}$, then there is an attribute $m \in M$ with $\mu n = \mu m$ or there is a set of attributes $B \subseteq M$ with $\mu n = \bigwedge \mu B$. So μM must be dense in $\mathbf{K}|\mathbf{C}$.

■ (I)⇐(IV) This is Theorema: Embedding into Apposition Lattice 2.12 with $N = \{n\}$.

Now the concepts of $\mathbf{K}|\mathbf{C}$ are constructed from those of $\mathbf{K}$ by Theorema: Embedding into Apposition Lattice 2.12 and Theorema: Nested Concept Lattice 2.13. If n is redundant, then both concept lattices are isomorphic by the mapping

$$\phi\colon \begin{aligned} \mathfrak{B}(\mathbf{K}) &\hookrightarrow\!\!\!\!\to \mathfrak{B}(\mathbf{K}|\mathbf{C}) \\ (A,B) &\mapsto (A, B \dot\cup A^J). \end{aligned}$$

Lemma 3.3 states, that $A^J = \{n\}$, iff $A \subseteq n^J$, and $A^J = \emptyset$ otherwise. Thereby

$$\phi(A,B) = \begin{cases} (A,B) & (A \not\subseteq n^J) \\ (A, B \mathbin{\dot\cup} \{n\}) & (A \subseteq n^J) \end{cases}$$

holds. A further refinement is given in Corollarium: Concept Transition from **K** to **K**|**C** and vice versa 3.8 in the next subsection. If the new attribute n is not redundant, then both concept lattices are not isomorphic and $\mathfrak{B}(\mathbf{K})$ can only be embedded in $\mathfrak{B}(\mathbf{K}|\mathbf{C})$. To be able to construct all concepts of the apposition $\mathbf{K}|\mathbf{C}$, consider the bijective mapping

$$\psi^{-1}\colon \begin{array}{c} \mathfrak{B}(\mathbf{K}) \boxtimes \mathfrak{B}(\mathbf{C}) \hookrightarrow\!\!\!\!\rightarrow \mathfrak{B}(\mathbf{K}|\mathbf{C}) \\ ((A,B),(C,D)) \mapsto (A \cap C, B \mathbin{\dot\cup} D) \end{array}$$

from Theorema: Nested Concept Lattice 2.13. With a closer look on the concept lattice $\mathfrak{B}(\mathbf{C})$ it is clear that it can have at most two concepts. The bottom concept is $\bot = (\emptyset^{JJ}, \emptyset^J) = (\{n\}^J, \{n\})$ and the top concept is $\top = (\emptyset^J, \emptyset^{JJ}) = (G, G^J)$. Both concepts are equal, if $n^J = G$, *i.e.* when $J = G \times \{n\}$ and the context is full of crosses. If n is not redundant then it cannot be reducible and thus in this case the context cannot be full and both concepts $\top$ and $\bot$ are distinct. Thereby it can be concluded that the following equations hold:

$$\psi^{-1}((A,B),\top) = (A \cap G, B \mathbin{\dot\cup} \emptyset) = (A,B)$$

and

$$\psi^{-1}((A,B),\bot) = (A \cap n^J, B \mathbin{\dot\cup} \{n\}) = \begin{cases} (A, B \mathbin{\dot\cup} \{n\}) & (A \subseteq n^J) \\ (A \cap n^J, B \mathbin{\dot\cup} \{n\}) & (A \not\subseteq n^J). \end{cases}$$

When looking at definition of the nested concept lattice one must check that each pair $((A,B),\top)$ is in $\mathfrak{B}(\mathbf{K}) \boxtimes \mathfrak{B}(\mathbf{C})$.

$$\begin{aligned} & ((A,B),\top) \in \mathfrak{B}(\mathbf{K}) \boxtimes \mathfrak{B}(\mathbf{C}) \\ \Leftrightarrow & (A \cap G, B \mathbin{\dot\cup} \emptyset) = (A,B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \\ \Leftrightarrow & \underbrace{\underbrace{A^{(I \dot\cup J)}}_{\substack{=A^I \dot\cup A^J \\ =B \dot\cup A^J}} = B}_{\substack{\Leftrightarrow A^J = \emptyset \\ \Leftrightarrow n \notin A^J \\ \Leftrightarrow A \not\subseteq n^J}} \text{ and } \underbrace{B^{(I \dot\cup J)}}_{=B^I = A} = A \\ \Leftrightarrow & A \not\subseteq n^J \end{aligned}$$

Analogously it has to be ensured that each pair $((A,B),\bot)$ is an element of the nested concept lattice $\mathfrak{B}(\mathbf{K}) \boxtimes \mathfrak{B}(\mathbf{C})$ as well.

$$\begin{aligned} & ((A,B),\bot) \in \mathfrak{B}(\mathbf{K}) \boxtimes \mathfrak{B}(\mathbf{C}) \\ \Leftrightarrow & (A \cap n^J, B \mathbin{\dot\cup} \{n\}) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \\ \Leftrightarrow & \underbrace{\underbrace{(A \cap n^J)^{(I \dot\cup J)}}_{=(A \cap n^J)^I \dot\cup (A \cap n^J)^J} = B \mathbin{\dot\cup} \{n\}}_{\Leftrightarrow (A \cap n^J)^I = B \text{ and } \underbrace{(A \cap n^J)^J = \{n\}}_{\substack{\Leftrightarrow n \in (A \cap n^J)^J \\ \Leftrightarrow A \cap n^J \subseteq n^J}}} \text{ and } \underbrace{(B \mathbin{\dot\cup} \{n\})^{(I \dot\cup J)}}_{\substack{=B^I \cap n^J \\ =A \cap n^J}} = A \cap n^J \\ \Leftrightarrow & A \subseteq n^J \text{ or } (A \not\subseteq n^J \text{ and } (A \cap n^J)^I = B) \end{aligned}$$

In summary the following equation hold for the mapping ψ^{-1}.

$$\psi^{-1}\colon \begin{array}{ll} ((A,B),\top) \mapsto (A,B) & (A \not\subseteq n^J) \\ ((A,B),\bot) \mapsto \begin{cases} (A, B \mathbin{\dot\cup} \{n\}) \\ (A \cap n^J, B \mathbin{\dot\cup} \{n\}) \end{cases} & \begin{array}{l} (A \subseteq n^J) \\ (A \not\subseteq n^J \text{ and } (A \cap n^J)^I = B) \end{array} \end{array}$$

The concept (A,B) is always one of **K** and so this gives a strong advice how the concepts in **K**|**C** can be computed from those of **K** that are already known. Please note the high similarity to the equation for ϕ in the redundant case. For more details please have a look at the following subsection, especially at Corollarium: Concept Transition from **K** to **K**|**C** and vice versa 3.8.

3.2 **Corollarium: Structural Changes for Apposition Lattice**

If n is redundant, then the old lattice and the new lattice are isomorphic

$$\mathfrak{B}(\mathbf{K}|\mathbf{C}) \cong \mathfrak{B}(\mathbf{K}),$$

else the new lattice can be embedded into a doubling of the old lattice

$$\mathfrak{B}(\mathbf{K}|\mathbf{C}) \lesssim \mathfrak{B}(\mathbf{K}) \times \mathbf{2}.$$

3.1.1 Updating the Concepts

This following lemma gives some first observations for $\mathbf{K}|\mathbf{C}$.

3.3 **Lemma**

(I) For all object sets $A \subseteq G$ it holds

$$A \subseteq n^J \Leftrightarrow A^J = \{n\} \text{ and } A \not\subseteq n^J \Leftrightarrow A^J = \emptyset.$$

(II) For a concept (A, B) of $\mathbf{K}|\mathbf{C}$ it holds

$$n \in B \Leftrightarrow A \subseteq n^J.$$

Approbatio (I) For each object set $A \subseteq G$ always $A^J \subseteq \{n\}$ hold, and so $n \in A^J$ is also equivalent to $A^J = \{n\}$. The second equivalence follows by contraposition, and $A^J \neq \{n\}$ is equivalent to $A^J = \emptyset$ as $A^J \subseteq \{n\}$ always hold.

(II) Let (A, B) be a concept of $\mathbf{K}|\mathbf{C}$, *i.e.*

$$A = B^{I\dot\cup J} = (B \setminus \{n\})^I \cap (B \cap \{n\})^J \text{ and } B = A^{I\dot\cup J} = A^I \dot\cup A^J$$

hold by Lemma: Common Rows and Common Columns in Apposition Context 2.11, thereby the intent B is a disjoint union of the subset A^I of M and the set A^J either containing n or not. Thus $n \in B$, iff $n \in A^J$. With one of the galois properties this is equivalent to $A \subseteq n^J$. ■

Concept Transition from K to K|C

The bijective mapping ψ^{-1} from Theorema: Nested Concept Lattice 2.13 state that there are three special kinds of concepts in $\mathbf{K}$ and thus also in $\mathbf{K}|\mathbf{C}$. First, the so called old concepts of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$ whose extent is no subset of the new attribute extent n^J. The set of all these old concepts is denoted by

$$\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K}) := \left\{(A, B) \in \mathfrak{B}(\mathbf{K}) \,\middle|\, A \not\subseteq n^J\right\}$$

and the old concepts are mapped to concepts of $\mathbf{K}|\mathbf{C}$ via

$$\mathsf{old}_{\mathbf{C}} \colon \begin{array}{l} \mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K}) \hookrightarrow \mathfrak{B}(\mathbf{K}|\mathbf{C}) \\ (A, B) \mapsto \psi^{-1}((A, B), \top) = (A, B) \end{array}$$

Second, the so called varying concepts with an extent contained in the new attribute extent n^J. All these varying concepts make up the set

$$\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) := \left\{(A, B) \in \mathfrak{B}(\mathbf{K}) \,\middle|\, A \subseteq n^J\right\}$$

and are mapped to concepts of $\mathbf{K}|\mathbf{C}$ by means of the mapping

$$\mathsf{var}_{\mathbf{C}} \colon \begin{array}{l} \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) \hookrightarrow \mathfrak{B}(\mathbf{K}|\mathbf{C}) \\ (A, B) \mapsto \psi^{-1}((A, B), \bot) = (A, B \dot\cup \{n\}) \end{array}$$

Third, the so called GENERATING concepts (A, B) from $\mathbf{K}$ *w.r.t.* $\mathbf{C}$ whose extent is not contained in n^J and furthermore fulfill the constraint $(A \cap n^J)^I = B$. The set of all these generating concepts is denoted by

$$\begin{aligned}\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) := &\left\{(A, B) \in \mathfrak{B}(\mathbf{K}) \,\middle|\, A \not\subseteq n^J \text{ and } (A \cap n^J)^I = B\right\}\\ = &\left\{(A, B) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K}) \,\middle|\, (A \cap n^J)^I = B\right\},\end{aligned}$$

and are used to construct some concepts of $\mathbf{K}|\mathbf{C}$ via

$$\mathsf{new}_{\mathbf{C}}\colon \begin{array}{l}\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \hookrightarrow \mathfrak{B}(\mathbf{K}|\mathbf{C})\\ (A, B) \mapsto \psi^{-1}((A, B), \bot) = (A \cap n^J, B \,\dot\cup\, \{n\}).\end{array}$$

As ψ^{-1} is bijective, the three above defined maps are also injective. Furthermore a partition of $\mathfrak{B}(\mathbf{K}|\mathbf{C})$ is obtained by the three images

$$\begin{aligned}\mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) &:= \psi^{-1}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K}) \times \{\top\}\right) = \mathsf{old}_{\mathbf{C}}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})\right)\\ \mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) &:= \psi^{-1}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) \times \{\bot\}\right) = \mathsf{var}_{\mathbf{C}}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})\right)\\ \mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}) &:= \psi^{-1}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \times \{\bot\}\right) = \mathsf{new}_{\mathbf{C}}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})\right)\end{aligned}$$

and the three above mentioned maps $\mathsf{old}_{\mathbf{C}}$, $\mathsf{var}_{\mathbf{C}}$ and $\mathsf{new}_{\mathbf{C}}$ are then surjections onto these disjoint subsets of $\mathfrak{B}(\mathbf{K}|\mathbf{C})$. The elements of $\mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C})$ are called OLD concepts of $\mathbf{K}|\mathbf{C}$ *w.r.t.* $\mathbf{C}$, these of $\mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C})$ VARIED and these of $\mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C})$ NEW concepts of $\mathbf{K}|\mathbf{C}$ *w.r.t.* $\mathbf{C}$.
For a redundant attribute n the old concept lattice $\mathfrak{B}(\mathbf{K})$ and the new concept lattice $\mathfrak{B}(\mathbf{K}|\mathbf{C})$ are isomorphic via the mapping ϕ from Theorema: Embedding into Apposition Lattice 2.12, so the concepts of $\mathbf{K}|\mathbf{C}$ must be fully determined by the old and varying concepts in $\mathbf{K}$. In other words: The set of generating concepts of $\mathbf{K}$ must be empty, and no new concepts exist in $\mathbf{K}|\mathbf{C}$.

Corollarium: Concept Update 3.4

The formal concepts of $\mathbf{K}|\mathbf{C}$ can be computed from those of $\mathbf{K}$ by means of the three bijections $\mathsf{old}_{\mathbf{C}}$, $\mathsf{var}_{\mathbf{C}}$ and $\mathsf{new}_{\mathbf{C}}$. If the new attribute n is redundant, then there are no generating concepts in $\mathbf{K}$ and no new concepts in $\mathbf{K}|\mathbf{C}$.

Concept Transition from K|C to K

For an inversion of the concept transition, *i.e.* when removing the attribute n from $\mathbf{K}|\mathbf{C}$, explicit descriptions of the sets $\mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C})$, $\mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C})$ and $\mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C})$ as well as the inversions of the maps $\mathsf{old}_{\mathbf{C}}$, $\mathsf{var}_{\mathbf{C}}$ and $\mathsf{new}_{\mathbf{C}}$ are required. The inverse maps can be determined by means of the inverse of ψ^{-1} from Theorema: Nested Concept Lattice 2.13, *viz.*

$$\psi\colon \begin{array}{l}\underline{\mathfrak{B}}(\mathbf{K}|\mathbf{C}) \hookrightarrow\!\!\!\!\rightarrow \underline{\mathfrak{B}}(\mathbf{K}) \boxtimes \underline{\mathfrak{B}}(\mathbf{C})\\ (A, B) \mapsto ((A^{II}, A^I), (A^{JJ}, A^J)) = (((B \cap M)^I, B \cap M), ((B \cap N)^J, B \cap N)),\end{array}$$

and as only the first coordinate in $\mathfrak{B}(\mathbf{K})$ is of interest, they must be given by $(A, B) \mapsto (A^{II}, A^I) = ((B \cap M)^I, B \cap M) = ((B \setminus \{n\})^I, B \setminus \{n\})$.
First, the old concepts are analyzed. Let (A, B) be an old concept of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$. Then the extent A is no subset of n^J, and surely $(A, B) = \mathsf{old}_{\mathbf{C}}(A, B)$ does not contain n in its intent.

Lemma 3.5

A concept (A, B) of $\mathbf{K}|\mathbf{C}$ with $n \notin B$ is always a concept of $\mathbf{K}$ as well, and $A \not\subseteq n^J$ holds.

APPROBATIO From $n \notin B$ it follows $A \not\subseteq n^J$ and $A^J = \emptyset$. So $B = A^I \,\dot\cup\, \emptyset = A^I$. From $n \notin B$ we further deduce $B \cap \{n\} = \emptyset$ and $B \setminus \{n\} = B$, and it follows $A = B^I \cap \emptyset^J = B^I \cap G = B^I$. Thus (A, B) is a concept of $\mathbf{K}$ too. ∎

The preceding lemma states that $\mathsf{old}_{\mathbf{C}}$ is surjective onto $\{(A, B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \mid n \notin B\}$, and thus

$$\mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) = \{(A, B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \mid n \notin B\}$$

hold. The inverse map is clearly given by

$$\mathrm{old}_{\mathbf{C}}^{-1} \colon \begin{array}{l} \mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) \hookrightarrow\!\!\!\rightarrow \mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K}) \\ \quad (A,B) \mapsto (A^{II}, A^{I}) = (A,B). \end{array}$$

Second, the varying concepts are investigated. Let (A,B) be any varying concept of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$, then $A \subseteq n^J$ hold. The corresponding concept in $\mathbf{K}|\mathbf{C}$ is given by $\mathsf{var}_{\mathbf{C}}(A,B) = (A, B \dot\cup \{n\})$. Obviously it contains the new attribute n in its intent, and $((B \dot\cup \{n\}) \setminus \{n\})^I = B^I = A$ hold. This yields $\mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) \subseteq \{(A,B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \mid n \in B \text{ and } (B \setminus \{n\})^I = A\}$ as a first explicit description.

3.6 **Lemma**

For a concept (A,B) of $\mathbf{K}|\mathbf{C}$ with $n \in B$ and $(B \setminus \{n\})^I = A$, the biset

$$(C,D) := (A, B \setminus \{n\})$$

is always a concept of $\mathbf{K}$ with $C \subseteq n^J$, such that $(A,B) = \mathsf{var}_{\mathbf{C}}(C,D)$.

APPROBATIO The biset (C,D) is indeed a concept of $\mathbf{K}|\mathbf{C}$ as $C^I = A^I = A^{(I \dot\cup J)} \cap M = B \setminus \{n\} = D$ and $D^I = (B \setminus \{n\})^I = A = C$ hold by Lemma: Common Rows and Common Columns in Apposition Context 2.11 and Lemma 3.3. Also $n \in B$ implies $C = A \subseteq n^J$ as desired. ■

As a corollary $\mathsf{var}_{\mathbf{C}}$ is surjective onto $\{(A,B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \mid n \in B \text{ and } (B \setminus \{n\})^I = A\}$, and this yields

$$\mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) = \left\{(A,B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \,\middle|\, n \in B \text{ and } (B \setminus \{n\})^I = A\right\}.$$

The inverse map is obviously determined by

$$\mathsf{var}_{\mathbf{C}}^{-1} \colon \begin{array}{l} \mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) \hookrightarrow\!\!\!\rightarrow \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) \\ \quad (A,B) \mapsto ((B \setminus \{n\})^I, B \setminus \{n\}) = (A, B \setminus \{n\}). \end{array}$$

Third, the generating and new concepts are examined. Let (A,B) be a generating concept of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$, *i.e.* $A \not\subseteq n^J$ and $(A \cap n^J)^I = B$ hold. Then (A,B) is embedded in the concept set of the new context $\mathbf{K}|\mathbf{C}$ by $\mathsf{new}_{\mathbf{C}}(A,B) = (A \cap n^J, B \dot\cup \{n\})$. Obviously n is an element of $B \dot\cup \{n\}$, and $((B \dot\cup \{n\}) \setminus \{n\})^I = B^I = A \neq A \cap n^J$ hold since $A \not\subseteq n^J$. Thus for the set of new concepts of $\mathbf{K}|\mathbf{C}$ the inclusion $\mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}) \subseteq \{(A,B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \mid n \in B \text{ and } (B \setminus \{n\})^I \neq A\}$ hold.

3.7 **Lemma**

For a concept (A,B) of $\mathbf{K}|\mathbf{C}$ with $n \in B$ and $(B \setminus \{n\})^I \neq A$, the biset

$$(C,D) := ((B \setminus \{n\})^I, B \setminus \{n\})$$

is always a concept of $\mathbf{K}$ with $C \not\subseteq n^J$ and $(C \cap n^J)^I = D$, such that $(A,B) = \mathsf{new}_{\mathbf{C}}(C,D)$.

APPROBATIO Trivially $D^I = (B \setminus \{n\})^I = C$ holds. Also

$$C^I = (B \setminus \{n\})^{II} = (A^{(I \dot\cup J)} \cap M)^{II} = A^{III} = A^I = A^{(I \dot\cup J)} \cap M = B \setminus \{n\} = D$$

is true, so (C,D) is indeed a formal concept of $\mathbf{K}$. Furthermore

$$C \cap n^J = (B \setminus \{n\})^I \cap n^J = B^{(I \dot\cup J)} = A \neq (B \setminus \{n\})^I = C$$

hold, which implies $C \not\subseteq n^J$, and furthermore $(C \cap n^J)^I = A^I = A^{(I \dot\cup J)} \cap M = B \setminus \{n\} = D$ is true. ■

As a conclusion $\mathsf{new}_{\mathbf{C}}$ is a surjection onto $\{(A,B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \mid n \in B \text{ and } (B \setminus \{n\})^I \neq A\}$ and so

$$\mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}) = \left\{(A,B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \,\middle|\, n \in B \text{ and } (B \setminus \{n\})^I \neq A\right\}.$$

The inverse map is given by

$$\mathsf{gen}_{\mathbf{C}} := \mathsf{new}_{\mathbf{C}}^{-1} \colon \begin{array}{l} \mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}) \hookrightarrow\!\!\!\rightarrow \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \\ \quad (A,B) \mapsto ((B \setminus \{n\})^I, B \setminus \{n\}). \end{array}$$

Corollarium: Concept Transition from K to K|C and vice versa 3.8

The concepts of $\mathbf{K}|\mathbf{C}$ can be computed from those of $\mathbf{K}$ and vice versa:

(I) Each old concept of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$ is an old concept of $\mathbf{K}|\mathbf{C}$ as well, and conversely every old concept of $\mathbf{K}|\mathbf{C}$ is an old concept of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$, by the bijection

$$\mathsf{old}_{\mathbf{C}}\colon \quad \begin{array}{c} \mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K}) \hookrightarrow\!\!\!\rightarrow \mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) \\ (A,B) \mapsto (A,B) \\ (A,B) \mapsfrom (A,B) \end{array}$$

with $\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K}) = \{(A,B) \in \mathfrak{B}(\mathbf{K}) \mid A \not\subseteq n^J\}$ and $\mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) = \{(A,B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \mid n \notin B\}$.

(II) Every varying concept of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$ is mapped to a varied concept of $\mathbf{K}|\mathbf{C}$ by adding the new attribute n to the intent, and reversely each varied concept of $\mathbf{K}|\mathbf{C}$ become a varying concept of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$ be removing n from its intent. This is due to the bijection

$$\mathsf{var}_{\mathbf{C}}\colon \quad \begin{array}{c} \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) \hookrightarrow\!\!\!\rightarrow \mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) \\ (A,B) \mapsto (A, B \mathbin{\dot\cup} \{n\}) \\ (A, B \setminus \{n\}) \mapsfrom (A,B) \end{array}$$

with its domain $\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) = \{(A,B) \in \mathfrak{B}(\mathbf{K}) \mid A \subseteq n^J\}$ and its range $\mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) = \{(A,B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \mid n \in B$ and $(B \setminus \{n\}$

(III) Each new concept of $\mathbf{K}|\mathbf{C}$ can be constructed from a unique generating concept of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$ by intersecting the extent with the new attribute extent n^J and adding the new attribute n to the intent. Conversely the generator in $\mathbf{K}$ *w.r.t.* $\mathbf{C}$ for a new concept in $\mathbf{K}|\mathbf{C}$ can be computed by removing the attribute n from its intent and choosing the corresponding extent by means of the old incidence I. For the transition from $\mathbf{K}|\mathbf{C}$ to $\mathbf{K}$ these new concepts will rather be removed than determining their generators. The map

$$\mathsf{new}_{\mathbf{C}}\colon \quad \begin{array}{c} \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \hookrightarrow\!\!\!\rightarrow \mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}) \\ (A,B) \mapsto (A \cap n^J, B \mathbin{\dot\cup} \{n\}) \\ ((B \setminus \{n\})^I, B \setminus \{n\}) \mapsfrom (A,B) \end{array}$$

is a bijection, with its domain $\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) = \{(A,B) \in \mathfrak{B}(\mathbf{K}) \mid A \not\subseteq n^J$ and $(A \cap n^J)^I = B\}$ and range $\mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}) = \{(A,B) \in \mathfrak{B}(\mathbf{K}|\mathbf{C}) \mid n \in B$ and $(B \setminus \{n\})^I \neq A\}$.

3.1.2 Structural Remarks

3.9 **Theorema: Structure of Old Concepts and Varying Concepts**

(I) The set $\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})$ of old concepts is a $\vee$-closed order-filter in $\mathfrak{B}(\mathbf{K})$.

(II) The set $\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$ of varying concepts is a $\wedge$-closed order-ideal in $\mathfrak{B}(\mathbf{K})$.

APPROBATIO ($\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})$ ORDER-FILTER) Let (A,B) be an old concept of $\mathbf{K}$ *w.r.t.* the new attribute n and (C,D) a concept of $\mathbf{K}$ that is greater than (A,B). This means $n^J \not\supseteq A \subseteq C$ and thus $n^J \not\supseteq C$ holds, *i.e.* (C,D) is also an old concept.

($\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})$ $\bigvee$-CLOSED) Let (A,B) and (C,D) be two old concepts of $\mathbf{K}$ *w.r.t.* the new attribute n. Then it holds $A \not\subseteq n^J$ and $C \not\subseteq n^J$, thus the union $A \cup C$ cannot be a subset of the extent n^J, and so $(A \cup C)^{II} \not\subseteq n^J$. Eventually $(A,B) \vee (C,D) = ((A \cup C)^{II}, B \cap D)$ is also an old concept.

($\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$ ORDER-IDEAL) Let (A,B) be a varying concept of $\mathbf{K}$ *w.r.t.* the new attribute n and (C,D) a concept of $\mathbf{K}$ that is smaller than (A,B). This means $n^J \supseteq A \supseteq C$ and thus $n^J \supseteq C$ holds, *i.e.* (C,D) is also a varying concept.

($\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$ $\bigwedge$-CLOSED) Let (A,B) and (C,D) be two varying concepts of $\mathbf{K}$ *w.r.t.* the new attribute n. Then it holds $A \subseteq n^J$ and $C \subseteq n^J$, thus the intersection $A \cap C$ is a subset of the extent n^J. Eventually $(A,B) \wedge (C,D) =$ $(A \cap C, (B \cup D)^{II})$ is also an varying concept. ■

In the ongoing section there are some concept lattices drawn. Their nodes and edges can have special forms: Each generator node is highlighted with a pentagon, each new node is marked with a star and each varying or varied node is tagged with a cloud.

3.10 **Exemplum: Counterexamples**

($\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})$ NOT $\bigwedge$-CLOSED) The set $\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})$ of all old concepts of a context $\mathbf{K} = \mathbf{K}$ *w.r.t.* a new attribute n is in general not closed under arbitrary infima. To understand this, please have a look at the following minimal example: The concept lattice is a diamond with four elements, *i.e.* a lattice generated by two distinct uncomparable elements. The new attribute 3 with its extent $3^J = \emptyset$ encounters the context. As a preresult the bottom element is a varying node because its concept is $(\emptyset, \{1,2\})$ and its extent $\emptyset$ is a subset of the attribute extent 3^J. All remaining nodes are old nodes as their concepts contains at least one object and thus their extents cannot be a subset of $\emptyset$. Eventually none of the old nodes is a generator node, since none of them is able to fulfill the condition $B = (A \cap 3^J)^I$ from Corollarium: Concept Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ and vice versa 3.8. This is due to $A \cap 3^J = \emptyset$ and $\emptyset^I = \{1,2\}$, and none of the old concepts has this intent $\{1,2\}$.

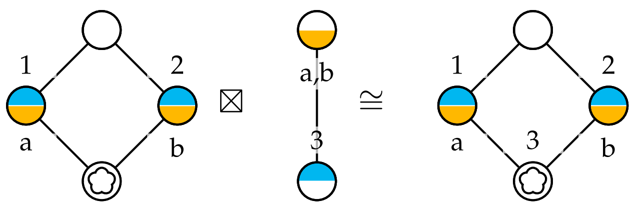
	1	2
a	×	
b		×

$|$

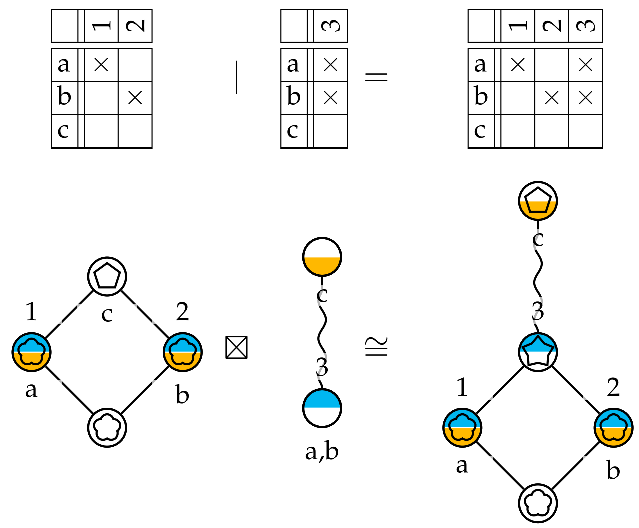
	3
a	
b	

$=$

	1	2	3
a	×		
b		×	



($\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$ NOT $\bigvee$-CLOSED) The set $\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$ of all varying nodes of $\mathbf{K} = \mathbf{K}$ *w.r.t.* a new attribute n is generally not $\bigvee$-closed. A similar example to the preceding example is chosen, but is modified to have the top node as a generator and the other nodes as varying nodes. The two maximal varying nodes do not have a varying node as their supremum.



Theorema: Structure of Generator Concepts 3.11

The set $\mathfrak{B}^{\mathbf{C}}_{\mathrm{gen}}(\mathbf{K})$ of generator concepts is $\vee$-closed in $\mathfrak{B}(\mathbf{K})$.

APPROBATIO Let (A_1, B_1) and (A_2, B_2) be two generator concepts of $\mathbf{K}$ and their supremum is $((A_1 \cup A_2)^{II}, B_1 \cap B_2)$. As each generator is also an old concept and the set of old concepts is closed under suprema, the supremum $((A_1 \cup A_2)^{II}, B_1 \cap B_2)$ must be an old concept. It remains to show, that $B_1 \cap B_2 = ((A_1 \cup A_2)^{II} \cap n^J)^I$.

($\subseteq$) Assume $B_1 \cap B_2$ is no subset of $((A_1 \cup A_2)^{II} \cap n^J)^I$. Then a attribute $m \in M$ must exist, such that $m \in B_1 \cap B_2$ and $m \notin ((A_1 \cup A_2)^{II} \cap n^J)^I$. From the second condition we get

$$m \notin ((A_1 \cup A_2)^{II} \cap n^J)^I \supseteq (A_1 \cup A_2)^{III} \cup n^{JI} \supseteq (A_1 \cup A_2)^{III} = (A_1 \cup A_2)^I = A_1^I \cap A_2^I = B_1 \cap B_2$$

in contradiction to the first condition. Thus $B_1 \cap B_2$ must be subset of $((A_1 \cup A_2)^{II} \cap n^J)^I$.

($\supseteq$) As (A_1, B_1) and (A_2, B_2) are generators, we have $B_1 = (A_1 \cap n^J)^I$ and $B_2 = (A_2 \cap n^J)^I$. Further it holds

$$\begin{aligned} B_1 \cap B_2 &= (A_1 \cap n^J)^I \cap (A_2 \cap n^J)^I \\ &= ((A_1 \cap n^J) \cup (A_2 \cap n^J))^I \\ &= (\underbrace{(A_1 \cup A_2)}_{\subseteq (A_1 \cup A_2)^{II}} \cap n^J)^I \\ &\supseteq ((A_1 \cup A_2)^{II} \cap n^J)^I. \end{aligned}$$

■

Corollarium: Largest Generator Concept 3.12

(I) $\mathfrak{B}^{\mathbf{C}}_{\mathrm{gen}}(\mathbf{K}) \cup \{\perp\}$ is a $\bigvee$-subsemilattice of $\mathfrak{B}(\mathbf{K})$.

(II) If the new attribute n is not redundant, there is always a largest generator concept

$$\top^{\mathbf{C}}_{\mathrm{gen}} := \bigvee \mathfrak{B}^{\mathbf{C}}_{\mathrm{gen}}(\mathbf{K}) = (n^{JII}, n^{JI}).$$

(III) The new generated concept $\mathsf{new}_{\mathbf{C}}(\top^{\mathbf{C}}_{\mathrm{gen}})$ then equals the attribute concept $\mu(n)$ of $\mathbf{K}|\mathbf{C}$ and has the attribute label n.

APPROBATIO (I) A $\bigvee$-subsemilattice is a subset U of a complete lattice, such that for each subset $X \subseteq U$ the supremum $\bigvee X$ is in U as well. In case of a finite lattice (like here) each *big supremum* $\bigvee$ can be expressed by means of the *small supremum* $\vee$ via

$$\bigvee\{x_1,\ldots,x_n\} = x_1 \vee (x_2 \vee (\cdots \vee (x_{n-1} \vee x_n))\ldots).$$

Thus only the empty supremum $\bigvee\emptyset = \bot$ is missing in $\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$ to form a $\bigvee$-subsemilattice.

(II),(III) When n is non-redundant, the new attribute extent n^J is no extent of $\mathbf{K}$. Hence no old or varying concept with n^J as its extent exists. Thus, the set of generators cannot be empty, as there must be a generator concept of the new attribute concept $\mu(n)$ in $\mathbf{K}|\mathbf{C}$. Then by the preceding Theorema: Structure of Generator Concepts 3.11 there must be a largest generator concept, namely $\top^{\mathbf{C}}_{\mathsf{gen}} := \bigvee\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$. By Lemma: Common Rows and Common Columns in Apposition Context 2.11 the new attribute concept can be displayed as

$$\mu(n) = (n^{(I\dot\cup J)}, n^{(I\dot\cup J)(I\dot\cup J)}) = (n^J, n^{J(I\dot\cup J)}) = (n^J, n^{JI}\dot\cup\{n\}) \in \mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C})$$

Thus a generator concept $(A,B) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$ exist such that $\mu n = \mathsf{new}_{\mathbf{C}}(A,B) = (A\cap n^J, B\dot\cup\{n\})$. So $n^J = A\cap n^J$ and $n^{JI}\dot\cup\{n\} = B\dot\cup\{n\}$ must hold. Thus, it follows that $n^J \subset A$ and $n^{JI} = B$, and

$$(A,B) = (n^{JII}, n^{JI}).$$

If there were any other concept $(C,D) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$ with $n^J \subset C$, then $C\cap n^J = n^J = A\cap n^J$ and thus $\mathsf{new}_{\mathbf{C}}(A,B) = \mathsf{new}_{\mathbf{C}}(C,D)$ would hold. This yields $(A,B) = (C,D)$, contradiction! Furthermore, if (C,D) were a generator superconcept of (A,B), then $n^J \subset A \subset C$ holds. Thereby $n^J \subset C$ yields a contradiction! Thus (A,B) is uniquelly determined by $n^J \subset A$ and $n^{JI} = B$, and furthermore (A,B) is a maximal generator. As $\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$ is $\vee$-closed, (A,B) must be the greatest generator and thus equal $\bigvee\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$. ■

In the ongoing section some further counterexamples on the generator set are given; $\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$ neither has to be $\bigwedge$-closed, nor an order ideal in $\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})$, nor convex.

3.13 **Exemplum: Counterexamples**

($\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$ NOT $\bigwedge$-CLOSED) The set $\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$ consisting of all generator concepts of $\mathbf{K}$ *w.r.t.* a new attribute n is not closed under arbitrary infima. Again a minimal example is chosen: A concept lattice with four elements (as seen on the left) and a new attribute 3 with its extent $3^J = \{a,b\}$ (in the middle). As preresults in the left old concept lattice $\underline{\mathfrak{B}}(\mathbf{K})$ the bottom concept $(\emptyset,\{1,2\})$ is varying since the empty set is enclosed in every set. The other three concepts are generators, as their extents are no subset of the new attribute extent 3^J and thus must be old nodes, and furthermore fulfill the generator condition $B = (A\cap 3^J)^I$ from **??** ??.**??**: The top concept is $(\{a,b,c,d\},\emptyset)$ and

$$(\{a,b,c,d\}\cap 3^J)^I = (\{a,b,c,d\}\cap\{a,b\})^I = \{a,b\}^I = \emptyset$$

holds, the left concept is $(\{a,c\},\{1\})$ and

$$(\{a,c\}\cap 3^J)^I = (\{a,c\}\cap\{a,b\})^I = \{a\}^I = \{1\}$$

holds and finally the right concept is $(\{b,d\},\{2\})$ and fulfills the generator condition

$$(\{b,d\}\cap 3^J)^I = (\{b,d\}\cap\{a,b\})^I = \{b\}^I = \{2\}.$$

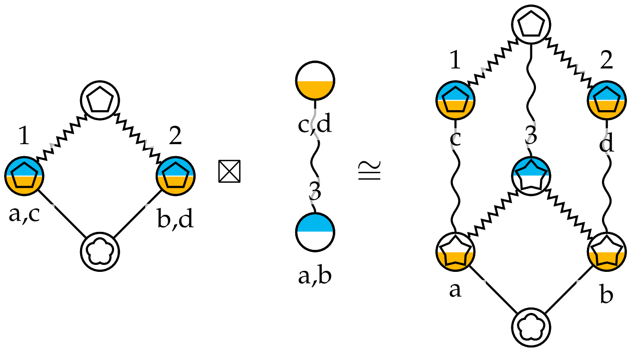
	1	2
a	×	
b		×
c	×	
d		×

$|$

	3
a	×
b	×
c	
d	

$=$

	1	2	3
a	×		×
b		×	×
c	×		
d		×	



($\mathfrak{B}^{\mathbf{C}}_{\text{gen}}(\mathbf{K})$ NOT ORDER IDEAL IN $\mathfrak{B}^{\mathbf{C}}_{\text{old}}(\mathbf{K})$) The set $\mathfrak{B}^{\mathbf{C}}_{\text{gen}}(\mathbf{K})$ of generators generally does not form an order ideal within the set $\mathfrak{B}^{\mathbf{C}}_{\text{old}}(\mathbf{K})$ of old concepts. A minimal counterexample can again be obtained from the "diamond concept lattice" as already seen in previous examples. The appropriate context is changed in a way to have the top node as a generator, the bottom node as a varying concept and the left and right concept as old ones. This is of course done *w.r.t.* the new attribute 3 with its attribute extent $3^{J} = \{c\}$. In the modified concept lattice on the right an edge between a new node and a varied one must be added while the corresponding generator was not neighboring the varying node.

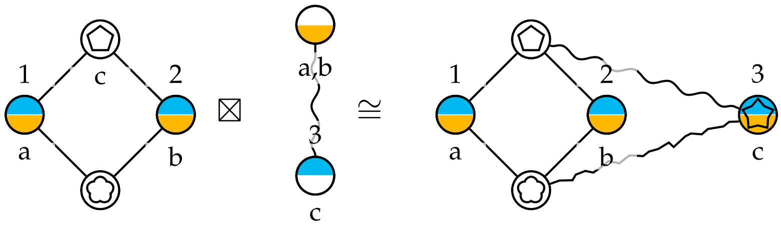
	1	2
a	×	
b		×
c		

$|$

	3
a	
b	
c	×

$=$

	1	2	3
a	×		
b		×	
c			×



($\mathfrak{B}^{\mathbf{C}}_{\text{gen}}(\mathbf{K})$ NOT CONVEX) The set $\mathfrak{B}^{\mathbf{C}}_{\text{gen}}(\mathbf{K})$ of generators is not convex in general. A counterexample is the three-element chain concept lattice on the left and the new attribute 3 with extent $3^{J} = \{c\}$. Applying Corollarium: Concept Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ and vice versa 3.8 and Corollarium: Concept Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ and vice versa 3.8 yields the top and bottom concept as generators and the inner concept as an old non-generating concept. The resulting concept lattice on the right is isomorphic to $\mathbf{N}_5$. Again we enconter a special case for the new neighborhood: Both new concepts are neighboring even though their generators are not. But these generators do not have any other generators between them, thus there cannot be any other new concept between the two new concepts. A more sophisticated answer gives Theorema: Neighborhood Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ 3.17.

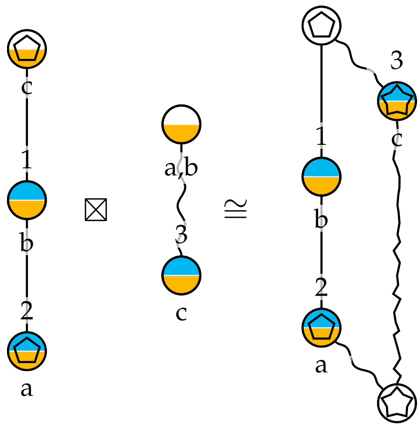
	1	2
a	×	×
b	×	
c		

$|$

	3
a	
b	
c	×

$=$

	1	2	3
a	×	×	
b	×		
c			×

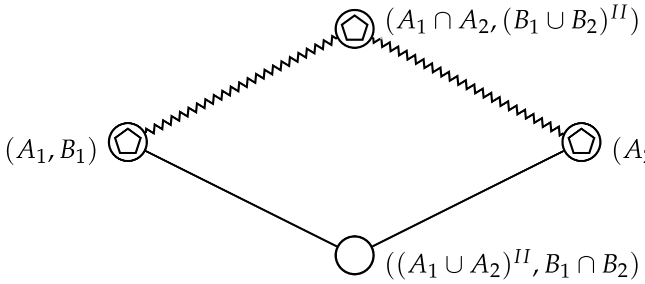


We have seen that a infimum of generators is not always a generator too. But a slightly weaker characterization of a generator infimum can be given:

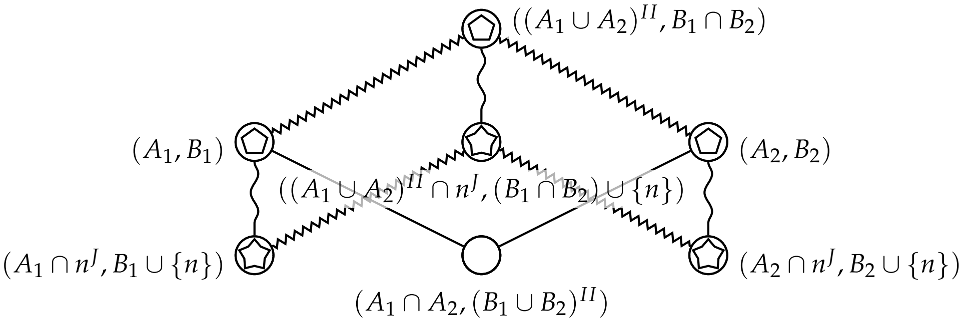
3.14 **Lemma: Infima of Generator Concepts**

The infimum, *i.e.* the greatest common subconcept, of a set of generators is always a generator too or a varying concept.

APPROBATIO Consider two uncomparable generator concepts (A_1, B_1) and (A_2, B_2), such that their infimum $(A_1, B_1) \wedge (A_2, B_2) = (A_1 \cap A_2, (B_1 \cup B_2)^{II})$ is an old concept, but not a generating concept. At first this implies that the infimum cannot be one of the two generator concepts.



Now a new irreducible attribute n is added and the diagram changes to the following structure:



As you can see, the resulting diagram does not form a lattice, since the new concepts $(A_1 \cap n^J, B_1 \cup \{n\})$ and $(A_2 \cap n^J, B_2 \cup \{n\})$ do not have any infimum. To see this consider the three possible cases for their infimum:

$((A_1 \cap n^J, B_1 \cup \{n\}) \wedge (A_2 \cap n^J, B_2 \cup \{n\}) = (A_1 \cap n^J, B_1 \cup \{n\}))$ This case condition is logically equivalent to the condition $(A_1 \cap n^J, B_1 \cup \{n\}) \leq (A_2 \cap n^J, B_2 \cup \{n\})$. Then the intents are comparable and $B_1 \dot{\cup} \{n\} \supseteq B_2 \dot{\cup} \{n\}$ holds - but this implies $B_1 \supseteq B_2$ in contradiction to the precondition that the generators (A_1, B_1) and (A_2, B_2) are not comparable. So this case cannot occur.

$((A_1 \cap n^J, B_1 \cup \{n\}) \wedge (A_2 \cap n^J, B_2 \cup \{n\}) = (A_2 \cap n^J, B_2 \cup \{n\}))$ analog to the first case.

$((A_1 \cap n^J, B_1 \cup \{n\}) \wedge (A_2 \cap n^J, B_2 \cup \{n\}) = (A_1 \cap A_2, (B_1 \cup B_2)^{II}))$ The case condition implies the equality of the extents $(A_1 \cap n^J) \cap (A_2 \cap n^J) = A_1 \cap A_2 \cap n^J$ and $A_1 \cap A_2$. This means the extent $A_1 \cap A_2$ must be a subset of the attribute extent n^J, but this is a conflict to the premise that the generator infimum $(A_1, B_1) \wedge (A_2, B_2) = (A_1 \cap A_2, (B_1 \cup B_2)^{II})$ is an old concept, *i.e.* $A_1 \cap A_2 \not\subseteq n^J$ holds. Eventually this case is also impossible.

None of these cases can occur, so the preconditions cannot occur. As a conseqence every infimum of generator concepts must also be a generating concept or a varying concept. ∎

3.1.3 Updating the Order

$\mathfrak{B}_{o\neg g}(\mathbf{K}|\mathbf{C})$ denote the set of all old concepts of $\mathbf{K}|\mathbf{C}$ that are no generator, *i.e.* $\mathfrak{B}_{o\neg g}(\mathbf{K}|\mathbf{C}) := \mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) \setminus \mathfrak{B}_{\mathsf{gen}}(\mathbf{K}|\mathbf{C})$. Furthermore $\mathfrak{B}_{\mathsf{gen}}(\mathbf{K}|\mathbf{C})$ is simply the image of all generators $\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$ under $\mathsf{old}_{\mathbf{C}}$, *i.e.* $\mathfrak{B}_{\mathsf{gen}}(\mathbf{K}|\mathbf{C}) := \mathsf{old}_{\mathbf{C}}(\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}))$. As the map $\mathsf{old}_{\mathbf{C}}$ does not change anything in the extent and intent, one does not have to distinct between old concepts of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$ and old concepts of $\mathbf{K}|\mathbf{C}$. So they are simply called old concepts. The same hold for the generator concepts.

Theorema: Order Transition from K to K|C 3.15

The order relation of $\underline{\mathfrak{B}}(\mathbf{K}|\mathbf{C})$ is divided into eight parts:

	$\mathfrak{B}_{\mathsf{old}}(\mathbf{K}\vert\mathbf{C})$	$\mathfrak{B}_{\mathsf{new}}(\mathbf{K}\vert\mathbf{C})$	$\mathfrak{B}_{\mathsf{var}}(\mathbf{K}\vert\mathbf{C})$
$\mathfrak{B}_{\mathsf{old}}(\mathbf{K}\vert\mathbf{C})$	(I)	(II)	
$\mathfrak{B}_{\mathsf{new}}(\mathbf{K}\vert\mathbf{C})$	(III)	(IV)	(V)
$\mathfrak{B}_{\mathsf{var}}(\mathbf{K}\vert\mathbf{C})$	(VI)	(VII)	(VIII)

Then the following statements characterize the order relation completely.

(I) Two old concepts are comparable in $\mathfrak{B}(\mathbf{K}|\mathbf{C})$, iff they are comparable in $\mathfrak{B}(\mathbf{K})$:

$$\underset{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})}{\forall} \mathsf{old}_{\mathbf{C}}(A,B) < \mathsf{old}_{\mathbf{C}}(C,D) \Leftrightarrow (A,B) < (C,D)$$

In other words, $\mathsf{old}_{\mathbf{C}}$ is order-preserving and order-reflecting.

(II) No old concept is smaller than any new or varied concept.

(III) A new concept is smaller than an old concept, iff its generator concept is smaller than or equals the old concept:

$$\underset{\substack{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})\\(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})}}{\forall} \mathsf{new}_{\mathbf{C}}(A,B) < \mathsf{old}_{\mathbf{C}}(C,D) \Leftrightarrow (A,B) \leq (C,D)$$

(IV) Two new concepts are comparable, iff their generator concepts are comparable:

$$\underset{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})}{\forall} \mathsf{new}_{\mathbf{C}}(A,B) < \mathsf{new}_{\mathbf{C}}(C,D) \Leftrightarrow (A,B) < (C,D)$$

In other words, $\mathsf{new}_{\mathbf{C}}$ preserves and reflects order.

(V) A new concept is smaller than a varied concept, iff the generator is smaller than the varying concept:

$$\underset{\substack{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})\\(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})}}{\forall} \mathsf{new}_{\mathbf{C}}(A,B) < \mathsf{var}_{\mathbf{C}}(C,D) \Leftrightarrow (A,B) < (C,D)$$

(VI) A varied concept is smaller than an old concept, iff its corresponding varying concept is smaller than the old concept:

$$\forall_{\substack{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})\\(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})}} \mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{old}_{\mathbf{C}}(C,D) \Leftrightarrow (A,B) < (C,D)$$

(VII) A varied concept is smaller than a new concept, iff the varying concept is smaller than the generator:

$$\forall_{\substack{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})\\(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})}} \mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{new}_{\mathbf{C}}(C,D) \Leftrightarrow (A,B) < (C,D)$$

(VIII) Two varied concepts are comparable, iff the appropriate varying concepts are comparable:

$$\forall_{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})} \mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{var}_{\mathbf{C}}(C,D) \Leftrightarrow (A,B) < (C,D)$$

In other words, $\mathsf{var}_{\mathbf{C}}$ preserves and reflects order.

APPROBATIO (I),(VI),(VIII) As $\mathsf{old}_{\mathbf{C}}$ and $\mathsf{var}_{\mathbf{C}}$ do not change the extent of any concept and the concept order is defined by means of extent inclusion, they must obviously be order-preserving and order-reflecting. This also implies (VI).

(II) If an old concept (A,B) would be smaller than any new or varied concept (C,D), it would hold that $B \supset D \ni n$ and this is a contradiction, since no old concept has the attribute n in its intent. Hence no old concept can be smaller than any new or modified concept.

(III) Let (A,B) be a generator concept and (C,D) an old concept. Then $n \notin D$ yields

$$\mathsf{new}_{\mathbf{C}}(A,B) < \mathsf{old}_{\mathbf{C}}(C,D) \Leftrightarrow B \,\dot\cup\, \{n\} \supset D \Leftrightarrow B \supseteq D \Leftrightarrow (A,B) \leq (C,D).$$

(IV),(V),(VII) Let (A,B) and (C,D) be two generator concepts, or a varying concept and a generator concept. As $n \notin B, D$, it then holds that

$$\left.\begin{array}{r}\mathsf{new}_{\mathbf{C}}(A,B) < \mathsf{new}_{\mathbf{C}}(C,D)\\ \text{or } \mathsf{new}_{\mathbf{C}}(A,B) < \mathsf{var}_{\mathbf{C}}(C,D)\\ \text{or } \mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{new}_{\mathbf{C}}(C,D)\end{array}\right\} \Leftrightarrow B \,\dot\cup\, \{n\} \supset D \,\dot\cup\, \{n\} \Leftrightarrow B \supset D \Leftrightarrow (A,B) < (C,D).$$

■

As both maps $\mathsf{old}_{\mathbf{C}}$ and $\mathsf{new}_{\mathbf{C}}$ are order-isomorphisms, as proven above, one can easily gain the order of $\mathbf{K}$ from those of $\mathbf{K}|\mathbf{C}$.

3.16 **Corollarium: Order Transition from K|C to K**

The order relation of $\mathfrak{B}(\mathbf{K}|\mathbf{C})$ is divided into eight parts:

	$\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})$	$\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$
$\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})$	(I)	(II)
$\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$	(III)	(IV)

Then the following statements characterize the order relation completely.

(I) Two old concepts are comparable in $\mathfrak{B}(\mathbf{K}|\mathbf{C})$, iff they are comparable in $\mathfrak{B}(\mathbf{K})$:

$$\forall_{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})} (A,B) < (C,D) \Leftrightarrow \mathsf{old}_{\mathbf{C}}(A,B) < \mathsf{old}_{\mathbf{C}}(C,D)$$

(II) No old concept is smaller than any varying concept.

(III) A varying concept is smaller than an old concept, iff its corresponding varied concept is smaller than the old concept:

$$\forall_{\substack{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})\\(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})}} (A,B) < (C,D) \Leftrightarrow \mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{old}_{\mathbf{C}}(C,D)$$

(IV) Two varying concepts are comparable, iff the appropriate varyied concepts are comparable:

$$\forall_{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})} (A,B) < (C,D) \Leftrightarrow \mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{var}_{\mathbf{C}}(C,D)$$

3.1.4 Updating the Neighborhood

A first clue on how to update the neighborhood is given by the cover relation on a cartesian product *w.r.t.* coordinate-wise order. It is well known (?), that $(p_1,q_1) \prec (p_2,q_2)$ hold in a cartesian product of two ordered sets, iff either $p_1 \prec p_2$ and $q_1 = q_2$, or $p_1 = p_2$ and $q_1 \prec q_2$ hold.
As the nested concept lattice $\underline{\mathfrak{B}}(\mathbf{K}) \boxtimes \underline{\mathfrak{B}}(\mathbf{C})$ from Theorema: Nested Concept Lattice 2.13 has the inherited coordinate-wise order from the cartesian product $\underline{\mathfrak{B}}(\mathbf{K}) \times \underline{\mathfrak{B}}(\mathbf{C})$, some fragments of the cover relation in the nested product can already be read off the neighborhood within the cartesian product. To be more specific, whenever (α,β) covers (γ,δ) in $\mathfrak{B}(\mathbf{K}) \times \mathfrak{B}(\mathbf{C})$, then (α,β) covers (γ,δ) in $\mathfrak{B}(\mathbf{K}) \boxtimes \mathfrak{B}(\mathbf{C})$ as well. First, $((A,B),\bot) \prec ((A,B),\top)$ holds for all suitable concepts (A,B) of $\mathbf{K}$. Recall the partition of $\mathfrak{B}(\mathbf{K}|\mathbf{C})$, or of $\mathfrak{B}(\mathbf{K}) \boxtimes \mathfrak{B}(\mathbf{C})$ respectivelly, that was constructed in Concept Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ 3.3.1.1

$$\begin{aligned}
\mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) &:= \psi^{-1}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K}) \times \{\top\}\right) = \mathsf{old}_{\mathbf{C}}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})\right)\\
\mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) &:= \psi^{-1}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) \times \{\bot\}\right) = \mathsf{var}_{\mathbf{C}}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})\right)\\
\mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}) &:= \psi^{-1}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \times \{\bot\}\right) = \mathsf{new}_{\mathbf{C}}\left(\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})\right)
\end{aligned}$$

The only concepts of $\mathbf{K}$, which can occur both with $\top$ and $\bot$ in the nested lattice, are the generator concepts. Via the isomorphism ψ from Theorema: Nested Concept Lattice 2.13, this yields

$$\forall_{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})} \mathsf{gen}_{\mathbf{C}}(A,B) \prec \mathsf{old}_{\mathbf{C}}(A,B).$$

Also $((A,B),X) \prec ((C,D),X)$ hold for all suitable concepts $(A,B) \prec (C,D)$ of $\mathbf{K}$ and $X \in \{\bot,\top\} = \mathfrak{B}(\mathbf{C})$. Thereby the following statements can be infered:

$$\begin{aligned}
&\forall_{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})} (A,B) \prec (C,D) \Rightarrow \mathsf{old}_{\mathbf{C}}(A,B) \prec \mathsf{old}_{\mathbf{C}}(C,D)\\
&\forall_{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})} (A,B) \prec (C,D) \Rightarrow \mathsf{var}_{\mathbf{C}}(A,B) \prec \mathsf{var}_{\mathbf{C}}(C,D)\\
&\forall_{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})} (A,B) \prec (C,D) \Rightarrow \mathsf{new}_{\mathbf{C}}(A,B) \prec \mathsf{new}_{\mathbf{C}}(C,D)\\
&\forall_{\substack{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})\\(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})}} (A,B) \prec (C,D) \Rightarrow \mathsf{var}_{\mathbf{C}}(A,B) \prec \mathsf{new}_{\mathbf{C}}(C,D)
\end{aligned}$$

Nevertheless these observations do not fully determine the neighborhood of $\underline{\mathfrak{B}}(\mathbf{K}|\mathbf{C})$ by means of $\underline{\mathfrak{B}}(\mathbf{K})$. A more sophisticated and complete characterization is given in the next theorem.

Theorema: Neighborhood Transition from K to K|C 3.17

The neighborhood relation of $\mathfrak{B}(\mathbf{K}|\mathbf{C})$ is divided into ten parts:

	$\mathfrak{B}_{\mathsf{o}\neg\mathsf{g}}(\mathbf{K}\vert\mathbf{C})$	$\mathfrak{B}_{\mathsf{gen}}(\mathbf{K}\vert\mathbf{C})$	$\mathfrak{B}_{\mathsf{new}}(\mathbf{K}\vert\mathbf{C})$	$\mathfrak{B}_{\mathsf{var}}(\mathbf{K}\vert\mathbf{C})$
$\mathfrak{B}_{\mathsf{o}\neg\mathsf{g}}(\mathbf{K}\vert\mathbf{C})$, $\mathfrak{B}_{\mathsf{gen}}(\mathbf{K}\vert\mathbf{C})$	(I)		(II)	
$\mathfrak{B}_{\mathsf{new}}(\mathbf{K}\vert\mathbf{C})$	(III)	(IV)	(V)	(VI)
$\mathfrak{B}_{\mathsf{var}}(\mathbf{K}\vert\mathbf{C})$	(VII)	(VIII)	(IX)	(X)

Then the following statements characterize the neighborhood relation completely.

(I) Two old concepts are neighboring in $\mathfrak{B}(\mathbf{K}|\mathbf{C})$, iff they are neighboring in $\mathfrak{B}(\mathbf{K})$:

$$\underset{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})}{\forall} \mathsf{old}_{\mathbf{C}}(A,B) \prec \mathsf{old}_{\mathbf{C}}(C,D) \Leftrightarrow (A,B) \prec (C,D)$$

In other words, $\mathsf{old}_{\mathbf{C}}$ is neighborhood-preserving and neighborhood-reflecting.

(II) No old concept is a lower neighbor of any new or varied concept.

(III) No new concept is a lower neighbor of any old non-generator concept.

(IV) Each new concept is a lower neighbor of its appropriate generator concept and moreover has no other generator concepts as upper neighbors:

$$\underset{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})}{\forall} \mathsf{new}_{\mathbf{C}}(A,B) \prec \mathsf{old}_{\mathbf{C}}(A,B)$$

(V) Two new concepts are neighboring, iff their appropriate generator concepts are comparable and no other generator concept lies between them:

$$\underset{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})}{\forall} \mathsf{new}_{\mathbf{C}}(A,B) \prec \mathsf{new}_{\mathbf{C}}(C,D) \Leftrightarrow \begin{cases} (A,B) < (C,D) \text{ and} \\ \underset{(X,Y)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})}{\nexists} (A,B) < (X,Y) < (C,D) \end{cases}$$

In other words, $\mathsf{new}_{\mathbf{C}}$ preserves neighborhood.

(VI) No new concept is a lower neighbor of any varied concept.

(VII) A varied concept is a lower neighbor of a old non-generator concept, iff the corresponding varying concept and the old non-generator concept are neighboring:

$$\underset{\substack{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})\\(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{o}\neg\mathsf{g}}(\mathbf{K})}}{\forall} \mathsf{var}_{\mathbf{C}}(A,B) \prec \mathsf{old}_{\mathbf{C}}(C,D) \Leftrightarrow (A,B) \prec (C,D)$$

(VIII) No varied concept is a lower neighbor of any generator concept.

(IX) A varied concept is a lower neighbor of a new concept, iff the corresponding varying concept and the generator concept are comparable and furthermore no other varying or generator concept is between them:

$$\underset{\substack{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})\\(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})}}{\forall} \mathsf{var}_{\mathbf{C}}(A,B) \prec \mathsf{new}_{\mathbf{C}}(C,D) \Leftrightarrow \begin{cases} (A,B) < (C,D) \text{ and} \\ \underset{(X,Y)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})\dot\cup\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})}{\nexists} (A,B) < (X,Y) < (C,D) \end{cases}$$

(X) Two varied concepts are neighboring, iff the appropriate varying concepts are neighboring:

$$\underset{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})}{\forall} \mathsf{var}_{\mathbf{C}}(A,B) \prec \mathsf{var}_{\mathbf{C}}(C,D) \Leftrightarrow (A,B) \prec (C,D)$$

In other words, $\mathsf{var}_{\mathbf{C}}$ preserves and reflects neighborhood.

APPROBATIO (I) Let (A, B) and (C, D) be two old concepts of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$. Then their corresponding concepts of $\mathbf{K}|\mathbf{C}$ are $\mathsf{old}_{\mathbf{C}}(A, B) = (A, B)$ and $\mathsf{old}_{\mathbf{C}}(C, D) = (C, D)$ according to Corollarium: Concept Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ and vice versa 3.8.
Let (A, B) be a lower neighbor of (C, D) in $\mathfrak{B}(\mathbf{K})$. At first this yields $(A, B) < (C, D)$ or rather $A \subset C$, and thereby $\mathsf{old}_{\mathbf{C}}(A, B)$ is a subconcept of $\mathsf{old}_{\mathbf{C}}(C, D)$ in $\mathfrak{B}(\mathbf{K}|\mathbf{C})$. If $\mathsf{old}_{\mathbf{C}}(A, B) \not\prec \mathsf{old}_{\mathbf{C}}(C, D)$ in $\mathfrak{B}(\mathbf{K}|\mathbf{C})$, *i.e.* there were a concept $(X, Y) \in \mathfrak{B}(\mathbf{K}|\mathbf{C})$ being between $\mathsf{old}_{\mathbf{C}}(A, B)$ and $\mathsf{old}_{\mathbf{C}}(C, D)$, then $B \supset Y \supset D$ would hold for the intents. As n cannot be an element of B, also $n \notin Y$ must hold, thus (X, Y) would surely be an old concept. It follows that $\mathsf{old}_{\mathbf{C}}^{-1}(X, Y) = (X, Y)$ were also a concept of $\mathbf{K}$, which were between (A, B) and (C, D). This is a contradiction to $(A, B) \prec (C, D)$. The embedded concepts $\mathsf{old}_{\mathbf{C}}(A, B)$ and $\mathsf{old}_{\mathbf{C}}(C, D)$ must thus be neighboring in $\mathfrak{B}(\mathbf{K}|\mathbf{C})$.
Conversely, let $\mathsf{old}_{\mathbf{C}}(A, B) \prec \mathsf{old}_{\mathbf{C}}(C, D)$ in $\mathfrak{B}(\mathbf{K}|\mathbf{C})$, *i.e.* no other concept of $\mathbf{K}|\mathbf{C}$ lies between them. Then their corresponding concepts (A, B) and (C, D) from $\mathbf{K}$ must be comparable, as $\mathsf{old}_{\mathbf{C}}(A, B) < \mathsf{old}_{\mathbf{C}}(C, D)$ and thus $A \subset C$ hold. If there were any other concept (X, Y) of $\mathbf{K}$ between (A, B) and (C, D), this must be an old concept as its extent cannot be a subset of the new attribute extent n^J, since $A \subset X \subset C$ and $A \not\subseteq n^J$ hold. This leads to a contradiction as well, as then $\mathsf{old}_{\mathbf{C}}(X, Y)$ would be between $\mathsf{old}_{\mathbf{C}}(A, B)$ and $\mathsf{old}_{\mathbf{C}}(C, D)$. Finally (C, D) covers (A, B) in the concept lattice of $\mathbf{K}$.

(II) If an old concept (A, B) would be smaller than any new or varied concept (C, D) of $\mathbf{K}|\mathbf{C}$, it would hold that $B \supset D \ni n$ and this is a contradiction, since no old concept has the attribute n in its intent. Hence no old concept can be a lower neighbor of any new or varied concept of $\mathbf{K}|\mathbf{C}$.

(III) Let (A, B) be a new concept of $\mathbf{K}|\mathbf{C}$, *i.e.* there is a generator concept (U, V). Let furthermore (C, D) be an old concept, that is no generator, and (A, B) is a lower neighbor of (C, D). Then we have $(A, B) = (U \cap n^J, V \dot\cup \{n\}) \prec (C, D)$ and thus $V \dot\cup \{n\} \supset D \not\ni n$. This leads to $V \dot\cup \{n\} \supset V \supseteq D$, so we would have $(A, B) = (U \cap n^J, V \dot\cup \{n\}) < (U, V) \leq (C, D)$. The generator concept (U, V) cannot equal the non-generator concept (C, D), hence $(A, B) < (U, V) < (C, D)$. This is a contradiction to $(A, B) \prec (C, D)$. So no new concept of $\mathbf{K}|\mathbf{C}$ can be covered by an old non-generator concept.

(IV) Let (A, B) be a new concept of $\mathbf{K}|\mathbf{C}$ with its generator concept (U, V), and (C, D) a generator concept that covers (A, B). Then it holds $(A, B) < (U, V) \leq (C, D)$ by the same arguments as in (III). When looking at the intents $B = V \dot\cup \{n\}$ and V, it is obvious that there cannot be any other intent in between, thus the new concept (A, B) must be a lower neighbor of its generator concept (C, D). Furthermore the generator concepts (U, V) and (C, D) must then be equal, since $(A, B) \prec (C, D)$.

(V) Let (A, B) and (C, D) be two new concepts of $\mathbf{K}|\mathbf{C}$ with their generating concepts (S, T) and (U, V), *i.e.*

$$(A, B) = \mathsf{new}_{\mathbf{C}}(S, T) = (S \cap n^J, T \dot\cup \{n\})$$
$$(C, D) = \mathsf{new}_{\mathbf{C}}(U, V) = (U \cap n^J, V \dot\cup \{n\}).$$

First, let $(A, B) \prec (C, D)$. We then know that each of these new concepts is a lower neighbor of their appropriate generators, *i.e.* $(A, B) \prec (S, T)$ and $(C, D) \prec (U, V)$. Hence $B = T \dot\cup \{n\} \supset V \dot\cup \{n\} = D$, or equally $T \supset V$, or $(S, T) < (U, V)$ respectivelly. If there were any other generator concept (X, Y) between (S, T) and (U, V), then the corresponding new concept $\mathsf{new}_{\mathbf{C}}(X, Y)$ would be between (A, B) and (C, D) as $T \supset Y \supset V$ yields $B \supset Y \dot\cup \{n\} \supset D$. Contradiction! These two generators (S, T) and (U, V) must thus be neighboring within the set of generators.
The other way around: Let the two generator concepts (S, T) and (U, V) be neighboring in the set of all generators. Then $T \supset V$ hold for the intents and there is no other generator-intent between T and V. This surely implies $B \supset D$ by adding n and no other generator-intent Y exist such that $B \supset Y \dot\cup \{n\} \supset D$ hold. This leads to $(A, B) < (C, D)$, and no other new concepts lies between them. Parts (III) to (VI) state that only generator concepts or new concepts can cover a new concept, and part (II) state that no generator concept can be a lower neighbor of a new concept, hence only new concepts can be between two new concepts. The concepts (A, B) and (C, D) are thus neighboring.

(VI) Let (A, B) be a new concept of $\mathbf{K}|\mathbf{C}$ with its generator (S, T), and let (C, D) be a varied concept of $\mathbf{K}|\mathbf{C}$ with its corresponding varying concept (U, V) of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$, such that $(A, B) \prec (C, D)$ hold. From these preconditions it follows that $B = T \dot\cup \{n\} \supset V \dot\cup \{n\} = D$, or equally $T \supset V$ and so $(S, T) < (U, V)$, but this implies $S \subset U \subseteq n^J$ as U is varying. Clearly this leads to a contradiction, since $S \not\subseteq n^J$ hold, because (S, T) is old. So no new concept of $\mathbf{K}|\mathbf{C}$ can be covered by a varied concept of $\mathbf{K}|\mathbf{C}$.

(VII) Let (A, B) be a varying concept of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$, *i.e.* $\mathsf{var}_{\mathbf{C}}(A, B) = (A, B \dot\cup \{n\})$, and (C, D) an old non-generator concept, *i.e.* $\mathsf{old}_{\mathbf{C}}(C, D) = (C, D)$.

First, $\mathsf{var}_{\mathbf{C}}(A,B) \prec \mathsf{old}_{\mathbf{C}}(C,D)$ yields $A \subset C$ and thereby (A,B) is a subconcept of (C,D). If there were any other concept $(X,Y) \in \mathfrak{B}(\mathbf{K})$ such that $(A,B) < (X,Y) < (C,D)$, then in case of an old concept $\mathsf{old}_{\mathbf{C}}(X,Y)$ would be between $\mathsf{var}_{\mathbf{C}}(A,B)$ and $\mathsf{old}_{\mathbf{C}}(C,D)$, and in case of a varying concept respectivelly $\mathsf{var}_{\mathbf{C}}(X,Y)$ would be between them. Both cases contradict $\mathsf{var}_{\mathbf{C}}(A,B) \prec \mathsf{old}_{\mathbf{C}}(C,D)$. So (C,D) covers (A,B).
Conversely, let (A,B) be a lower neighbor of (C,D). Then clearly $\mathsf{var}_{\mathbf{C}}(A,B)$ is a subconcept of $\mathsf{old}_{\mathbf{C}}(C,D)$. When a concept $(X,Y) \in \mathfrak{B}(\mathbf{K}|\mathbf{C})$ exists that is between $\mathsf{var}_{\mathbf{C}}(A,B)$ and $\mathsf{old}_{\mathbf{C}}(C,D)$, then obviously $A \subset X \subset C$ holds for the extents. In case of an old or varied concept this leads to a contradiction since the concept $\mathsf{old}_{\mathbf{C}}^{-1}(X,Y)$ or $\mathsf{var}_{\mathbf{C}}^{-1}(X,Y)$ respectivelly would be between (A,B) and (C,D). In case of a new concept there is a generator concept (U,V) such that $(X,Y) = \mathsf{new}_{\mathbf{C}}(U,V) = (U \cap n^J, V \dot\cup \{n\})$. Then $B \dot\cup \{n\} \supset V \dot\cup \{n\} = Y \supset D \not\ni n$ implies $B \supset V \supseteq D$ and thus $(A,B) < (U,V) \leq (C,D)$. By $(A,B) \prec (C,D)$ it follows that (U,V) equals (C,D), but this is a contradiction since no generator concept can equal any non-generator concept. In summary $\mathsf{var}_{\mathbf{C}}(A,B)$ is a lower neighbor of $\mathsf{old}_{\mathbf{C}}(C,D)$.

(VIII) Whenever a generator (C,D) covers a varied concept (A,B) of $\mathbf{K}|\mathbf{C}$, the intent B contains the intent D. Then $n \in B \supset D \not\ni n$ yields $B \supseteq D \dot\cup \{n\} \supset D$. Hence $(A,B) \leq \mathsf{new}_{\mathbf{C}}(C,D) = (C \cap n^J, D \dot\cup \{n\}) < (C,D)$, and by $(A,B) \prec (C,D)$ the varied concept (A,B) must equal the new concept $\mathsf{new}_{\mathbf{C}}(C,D)$. Clearly this is a contradiction. No varied concept can thus have a generator concept as an upper neighbor.

(IX) Suppose a varying concept (A,B) of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$ and a generator concept (C,D) are given. Then $\mathsf{var}_{\mathbf{C}}(A,B) = (A, B \dot\cup \{n\})$ and $\mathsf{new}_{\mathbf{C}}(C,D) = (C \cap n^J, D \dot\cup \{n\})$ hold.
At first let $\mathsf{var}_{\mathbf{C}}(A,B) \prec \mathsf{new}_{\mathbf{C}}(C,D)$, then $A \subset C \cap n^J \subset C$ hold for the extents, hence (A,B) is a subconcept of (C,D). Suppose there is a generator concept (X,Y) between them, then $A \subset X \subset C$ and thereby $A = A \cap n^J \subseteq X \cap n^J \subseteq C \cap n^J$ holds. This means $\mathsf{var}_{\mathbf{C}}(A,B) \leq \mathsf{new}_{\mathbf{C}}(X,Y) \leq \mathsf{new}_{\mathbf{C}}(C,D)$ and the new concepts $\mathsf{new}_{\mathbf{C}}(X,Y)$ and $\mathsf{new}_{\mathbf{C}}(C,D)$ must thus be equal. As $\mathsf{new}_{\mathbf{C}}$ is a bijection, this implies the equality of (X,Y) and (C,D). Obviously this contradicts $X \subset C$. Now let (X,Y) be a varying concept of $\mathbf{K}$ *w.r.t.* $\mathbf{C}$ between them, then $A \subset X \subset C$ holds as well. Hence $A = A \cap n^J \subset X = X \cap n^J \subseteq C \cap n^J$ or rather $\mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{var}_{\mathbf{C}}(X,Y) \leq \mathsf{new}_{\mathbf{C}}(C,D)$ holds. As $\mathsf{new}_{\mathbf{C}}(C,D)$ covers $\mathsf{var}_{\mathbf{C}}(A,B)$, the varied concept $\mathsf{var}_{\mathbf{C}}(X,Y)$ must equal the new concept $\mathsf{new}_{\mathbf{C}}(C,D)$. Contradiction! In summary, $(A,B) < (C,D)$ and there is no generator concept or varying concept between them.
Conversely, let (A,B) be a proper subconcept of (C,D), such that no generator concept or varying concept exist between them. Then $A \subset C$ holds for the extents and intersecting with the new attribute extent n^J yields $A = A \cap n^J \subseteq C \cap n^J$. Hence $\mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{new}_{\mathbf{C}}(C,D)$ holds. Suppose they would not be neighboring, *i.e.* there is any concept of $\mathbf{K}|\mathbf{C}$ between them. When $\mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{old}_{\mathbf{C}}(X,Y) < \mathsf{new}_{\mathbf{C}}(C,D)$ holds for an old concept $(X,Y) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})$, then $B \dot\cup \{n\} \supset Y \supset D \dot\cup \{n\}$ holds for the intents. This leads to a contradiction as $n \notin Y$. When $\mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{var}_{\mathbf{C}}(X,Y) < \mathsf{new}_{\mathbf{C}}(C,D)$ holds for a varying concept $(X,Y) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$, then $B \dot\cup \{n\} \supset Y \dot\cup \{n\} \supset D \dot\cup \{n\}$ holds for the intents, and thus also $B \supset Y \supset D$, *i.e.* $(A,B) < (X,Y) < (C,D)$. This is a contradiction. When $\mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{new}_{\mathbf{C}}(X,Y) < \mathsf{new}_{\mathbf{C}}(C,D)$ holds for a generator concept $(X,Y) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$, then analogously $(A,B) < (X,Y) < (C,D)$ yields a contradiction. In summary, $\mathsf{var}_{\mathbf{C}}(A,B)$ must be a lower neighbor of $\mathsf{new}_{\mathbf{C}}(C,D)$.

(X) Let $(A,B), (C,D) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$ be two varying concepts with $(A,B) \prec (C,D)$. Then surely $\mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{var}_{\mathbf{C}}(C,D)$ holds, as can be seen on the unchanging extents. If there were a concept (X,Y) of $\mathbf{K}|\mathbf{C}$ such that $\mathsf{var}_{\mathbf{C}}(A,B) < (X,Y) < \mathsf{var}_{\mathbf{C}}(C,D)$, then $B \dot\cup \{n\} \supset Y \supset D \dot\cup \{n\}$ holds, *i.e.* $n \in Y$ and (X,Y) must thus be a new or varied concept. If it were a varied concept, then $(A,B) < \mathsf{var}_{\mathbf{C}}^{-1}(X,Y) < (C,D)$ yields a contradiction. If it were a new concept, then analogously $(A,B) < \mathsf{new}_{\mathbf{C}}^{-1}(X,Y) < (C,D)$ is a contradiction. Eventually (X,Y) cannot be new or varied, and thereby such a concept cannot exist. This means $\mathsf{var}_{\mathbf{C}}(A,B)$ and $\mathsf{var}_{\mathbf{C}}(C,D)$ are neighboring.
For the other way around, let $(A,B) \prec (C,D)$ be varying concepts of $\mathbf{K}|\mathbf{C}$. Then $B \supset D$ holds and so $B \setminus \{n\} \supset D \setminus \{n\}$, which means $\mathsf{var}_{\mathbf{C}}^{-1}(A,B) < \mathsf{var}_{\mathbf{C}}^{-1}(C,D)$. If a concept (X,Y) of $\mathbf{K}$ would exist such that $\mathsf{var}_{\mathbf{C}}^{-1}(A,B) < (X,Y) < \mathsf{var}_{\mathbf{C}}^{-1}(C,D)$, then $A \supset X \supset C$ holds. This implies $X \subseteq n^J$ as $X \subset C \subseteq n^J$. So (X,Y) must be a varying concept too, and $(A,B) < \mathsf{var}_{\mathbf{C}}(X,Y) < (C,D)$ would hold, in contradiction to $(A,B) \prec (C,D)$. ∎

As an easy corollary we are now also able to describe the neighborhood of $\underline{\mathfrak{B}}(\mathbf{K})$ by means of the neighborhood of $\underline{\mathfrak{B}}(\mathbf{K}|\mathbf{C})$. This is summarized as follows.

3.18 **Corollarium: Neighborhood Transition from $\mathbf{K}|\mathbf{C}$ to $\mathbf{K}$**

The neighborhood relation of $\underline{\mathfrak{B}}(\mathbf{K})$ is divided into five parts:

<table>
<tr><td></td><td>$\mathfrak{B}^{\mathbf{C}}_{\mathsf{o}\neg\mathsf{g}}(\mathbf{K})$</td><td>$\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$</td><td>$\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$</td></tr>
<tr><td>$\mathfrak{B}^{\mathbf{C}}_{\mathsf{o}\neg\mathsf{g}}(\mathbf{K})$</td><td colspan="2" rowspan="2">(I)</td><td rowspan="2">(II)</td></tr>
<tr><td>$\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$</td></tr>
<tr><td>$\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$</td><td>(III)</td><td>(IV)</td><td>(V)</td></tr>
</table>

Then the following statements characterize the neighborhood relation completely.

(I) Two old concepts are neighboring in $\mathfrak{B}(\mathbf{K})$, iff they are neighboring in $\mathfrak{B}(\mathbf{K}|\mathbf{C})$:

$$\forall_{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})} (A,B) \prec (C,D) \Leftrightarrow \mathsf{old}_{\mathbf{C}}(A,B) \prec \mathsf{old}_{\mathbf{C}}(C,D)$$

(II) No old concept is a lower neighbor of any varying concept.

(III) A varying concept is a lower neighbor of an old non-generator concept, iff their corresponding concepts of $\mathbf{K}|\mathbf{C}$ are neighboring:

$$\forall_{\substack{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})\\(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{o}\neg\mathsf{g}}(\mathbf{K})}} (A,B) \prec (C,D) \Leftrightarrow \mathsf{var}_{\mathbf{C}}(A,B) \prec \mathsf{old}_{\mathbf{C}}(C,D)$$

(IV) A varying concept is a lower neighbor of a generator concept, iff the corresponding varied concept and new concept are neighboring and furthermore no other old non-generator concept is between the varied concept and the generator concept:

$$\forall_{\substack{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})\\(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})}} (A,B) \prec (C,D) \Leftrightarrow \begin{cases} \mathsf{var}_{\mathbf{C}}(A,B) \prec \mathsf{new}_{\mathbf{C}}(C,D) \text{ and} \\ \nexists_{(X,Y)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{o}\neg\mathsf{g}}(\mathbf{K})} \mathsf{var}_{\mathbf{C}}(A,B) < \mathsf{old}_{\mathbf{C}}(X,Y) < \mathsf{old}_{\mathbf{C}}(C,D) \end{cases}$$

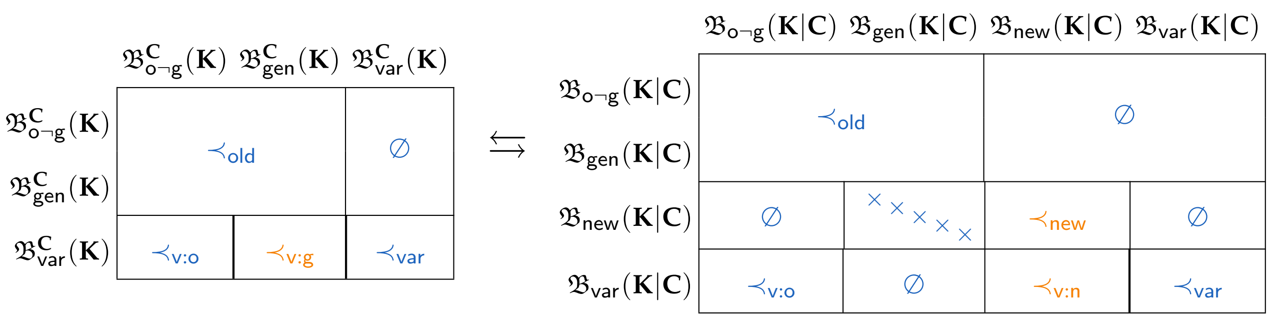
(V) Two varying concepts are neighboring, iff the appropriate varied concepts are neighboring:

$$\forall_{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})} (A,B) \prec (C,D) \Leftrightarrow \mathsf{var}_{\mathbf{C}}(A,B) \prec \mathsf{var}_{\mathbf{C}}(C,D)$$

Now we are able to completely describe the neighborhood relation of $\mathbf{K}|\mathbf{C}$ by means of the cover relation of $\mathbf{K}$ and vice versa. Especially when thinking of cover relations as binary relations encoded by matrices via the isomorphism

$$\chi\colon \begin{array}{l} \wp(X\times Y) \hookrightarrow\!\!\!\!\rightarrow \mathbf{2}^{X\times Y} \\ \qquad R \mapsto \chi_R\colon \begin{array}{l} X\times Y \to \mathbf{2} \\ (x,y) \mapsto \begin{cases} 1 & \text{if } x\,R\,y \\ 0 & \text{if } x \not R\, y \end{cases} \end{array} \\ f^{-1}(1) \leftarrow\!\mapsto f, \end{array}$$

the cover relations can be determined from each other by simply copying some parts, deleting some parts, and computing few parts. For this purpose the cover relation of $\underline{\mathfrak{B}}(\mathbf{K})$ and also the cover relation of $\underline{\mathfrak{B}}(\mathbf{K}|\mathbf{C})$ are split up in components



and the unknown parts can be computed via

$$\underset{\substack{(A,B)\in\mathfrak{B}_{\text{var}}(\mathbf{K}|\mathbf{C})\\(C,D)\in\mathfrak{B}_{\text{new}}(\mathbf{K}|\mathbf{C})}}{\forall} \text{var}_{\mathbf{C}}^{-1}(A,B) \prec_{\text{v:g}} \text{gen}_{\mathbf{C}}(C,D) \Leftrightarrow \begin{cases} (A,B) \prec_{\text{v:n}} (C,D) \text{ and} \\ \underset{(X,Y)\in\mathfrak{B}_{\text{o}\neg\text{g}}(\mathbf{K}|\mathbf{C})}{\nexists} (A,B) < (X,Y) < \text{old}_{\mathbf{C}}\text{gen}_{\mathbf{C}}(C,D) \end{cases}$$

$$\underset{(A,B),(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\text{gen}}(\mathbf{K})}{\forall} \text{new}_{\mathbf{C}}(A,B) \prec_{\text{new}} \text{new}_{\mathbf{C}}(C,D) \Leftrightarrow \begin{cases} (A,B) < (C,D) \text{ and} \\ \underset{(X,Y)\in\mathfrak{B}^{\mathbf{C}}_{\text{gen}}(\mathbf{K})}{\nexists} (A,B) < (X,Y) < (C,D) \end{cases}$$

$$\underset{\substack{(A,B)\in\mathfrak{B}^{\mathbf{C}}_{\text{var}}(\mathbf{K})\\(C,D)\in\mathfrak{B}^{\mathbf{C}}_{\text{gen}}(\mathbf{K})}}{\forall} \text{var}_{\mathbf{C}}(A,B) \prec_{\text{v:n}} \text{new}_{\mathbf{C}}(C,D) \Leftrightarrow \begin{cases} (A,B) < (C,D) \text{ and} \\ \underset{(X,Y)\in\mathfrak{B}^{\mathbf{C}}_{\text{gen}}(\mathbf{K})\dot\cup\mathfrak{B}^{\mathbf{C}}_{\text{var}}(\mathbf{K})}{\nexists} (A,B) < (X,Y) < (C,D) \end{cases}$$

3.1.5 Updating the Concept Labels

In this section the connection between attribute and object concepts of $\mathbf{K}$ and those of $\mathbf{K}|\mathbf{C}$ are investigated. At first, these special concepts of $\mathbf{K}|\mathbf{C}$ can be expressed by means of the next lemma.

Lemma: Object Concepts and Attribute Concepts of $\mathbf{K}|\mathbf{C}$ 3.19

The attribute concepts of $\mathbf{K}|\mathbf{C}$ can be described via

$$\mu_{\mathbf{K}|\mathbf{C}}(m) = \begin{cases} (m^I, m^{II}) & \text{if } m^I \not\subset n^J \\ (m^I, m^{II} \dot\cup \{n\}) & \text{if } m^I \subset n^J \end{cases}$$

$$\mu_{\mathbf{K}|\mathbf{C}}(n) = (n^J, n^{JJ} \dot\cup \{n\}).$$

The object concepts of $\mathbf{K}|\mathbf{C}$ are given by

$$\gamma_{\mathbf{K}|\mathbf{C}}(g) = \begin{cases} (g^{II}, g^I) & \text{if } g \notin n^J \\ (g^{II} \cap n^J, g^I \dot\cup \{n\}) & \text{if } g \in n^J. \end{cases}$$

APPROBATIO Some simple manipulations by means of Lemma: Common Rows and Common Columns in Apposition Context 2.11 yields

$$\mu_{\mathbf{K}|\mathbf{C}}(m) = (m^{(I\dot\cup J)}, m^{(I\dot\cup J)(I\dot\cup J)}) = (m^I, m^{I(I\dot\cup J)}) = (m^I, m^{II} \dot\cup m^{IJ})$$

$$= \begin{cases} (m^I, m^{II}) = \mu_{\mathbf{K}}(m) & \text{if } n \notin m^{IJ} \text{ iff } m^I \not\subset n^J \\ (m^I, m^{II} \dot\cup \{n\}) & \text{if } n \in m^{IJ} \text{ iff } m^I \subset n^J \end{cases}$$

$$\mu_{\mathbf{K}|\mathbf{C}}(n) = (n^{(I\dot\cup J)}, n^{(I\dot\cup J)}) = (n^J, n^{J(I\dot\cup J)}) = (n^J, n^{JI} \dot\cup \{n\})$$

$$\gamma_{\mathbf{K}|\mathbf{C}}(g) = (g^{(I\dot\cup J)(I\dot\cup J)}, g^{(I\dot\cup J)}) = ((g^I \dot\cup g^J)^{(I\dot\cup J)}, g^I \dot\cup g^J) = (g^{II} \cap g^{JJ}, g^I \dot\cup g^J)$$

$$= \begin{cases} (g^{II} \cap \emptyset^J, g^I) = (g^{II}, g^I) = \gamma_{\mathbf{K}}(g) & \text{if } n \notin g^J \text{ iff } g \not{J}\, n \text{ iff } g \notin n^J \\ (g^{II} \cap n^J, g^I \dot\cup \{n\}) & \text{if } n \in g^J \text{ iff } g\, J\, n \text{ iff } g \in n^J \end{cases}$$

■

If $\gamma_{\mathbf{K}}(g)$ is a varying concept, *i.e.* $g^{II} \subseteq n^J$, then clearly $g \in n^J$ holds. Hence, $\gamma_{\mathbf{K}|\mathbf{C}}(g) = \text{var}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$ holds. When $g \in n^J$, then $\gamma_{\mathbf{K}|\mathbf{C}}(g) = (g^{II} \cap n^J, g^I \dot\cup \{n\})$ must be a new or varied concept since its intent contains

n. Furthermore, when $\gamma_{\mathbf{K}}(g)$ is then an old concept, *i.e.* $g^{II} \not\subseteq n^J$, then $((g^I \dot\cup \{n\}) \setminus \{n\})^I = g^{II} \neq g^{II} \cap n^J$ holds, hence it must be a new concept with its generator $\mathsf{gen}_{\mathbf{C}}(\gamma_{\mathbf{K}|\mathbf{C}}(g)) = (g^{II}, g^I) = \gamma_{\mathbf{K}}(g)$. Thus $g \not{J} n$ is always true when $\gamma_{\mathbf{K}}(g)$ is an old non-generator concept. In summary the following corollary is gained.

3.20 **Corollarium: Transition of Object Concepts and Attribute Concepts**

The object and attribute concepts of $\mathbf{K}|\mathbf{C}$ can be determined from those of $\mathbf{K}$ by the following equations:

$$\forall_{m \in M} \mu_{\mathbf{K}|\mathbf{C}}(m) = \begin{cases} \mathsf{old}_{\mathbf{C}}(\mu_{\mathbf{K}}(m)) & \text{if } \mu_{\mathbf{K}}(m) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K}) \\ \mathsf{var}_{\mathbf{C}}(\mu_{\mathbf{K}}(m)) & \text{if } \mu_{\mathbf{K}}(m) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) \end{cases}$$

$$\forall_{g \in G} \gamma_{\mathbf{K}|\mathbf{C}}(g) = \begin{cases} \mathsf{old}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g)) & \text{if } \gamma_{\mathbf{K}}(g) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{o}\neg\mathsf{g}}(\mathbf{K}) \text{ or } (\gamma_{\mathbf{K}}(g) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \text{ and } g \notin n^J \\ \mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g)) & \text{if } \gamma_{\mathbf{K}}(g) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \text{ and } g \in n^J \\ \mathsf{var}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g)) & \text{if } \gamma_{\mathbf{K}}(g) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) \end{cases}$$

When n is not redundant, then by Corollarium: Largest Generator Concept 3.12 it follows that

$$\mu_{\mathbf{K}|\mathbf{C}}(n) = \mathsf{new}_{\mathbf{C}}(\top^{\mathbf{C}}_{\mathsf{gen}}),$$

otherwise $\mu_{\mathbf{K}|\mathbf{C}}(n) = \mathsf{var}_{\mathbf{C}}(\mu_{\mathbf{K}}(m))$ holds when $m^I = n^J$, or more generally

$$\mu_{\mathbf{K}|\mathbf{C}}(n) = \mathsf{var}_{\mathbf{C}}(n^J, n^{JI})$$

Conversely, due to the bijectivity of the maps $\mathsf{old}_{\mathbf{C}}$, $\mathsf{var}_{\mathbf{C}}$ and $\mathsf{new}_{\mathbf{C}}$, the object and attribute concepts of $\mathbf{K}$ can be computed from those of $\mathbf{K}|\mathbf{C}$ by the following equations:

$$\forall_{m \in M} \mu_{\mathbf{K}}(m) = \begin{cases} \mathsf{old}^{-1}_{\mathbf{C}}(\mu_{\mathbf{K}|\mathbf{C}}(m)) & \text{if } \mu_{\mathbf{K}|\mathbf{C}}(m) \in \mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) \\ \mathsf{var}^{-1}_{\mathbf{C}}(\mu_{\mathbf{K}|\mathbf{C}}(m)) & \text{if } \mu_{\mathbf{K}|\mathbf{C}}(m) \in \mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) \end{cases}$$

$$\forall_{g \in G} \gamma_{\mathbf{K}}(g) = \begin{cases} \mathsf{old}^{-1}_{\mathbf{C}}(\gamma_{\mathbf{K}|\mathbf{C}}(g)) & \text{if } \gamma_{\mathbf{K}|\mathbf{C}}(g) \in \mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) \\ \mathsf{gen}_{\mathbf{C}}(\gamma_{\mathbf{K}|\mathbf{C}}(g)) & \text{if } \gamma_{\mathbf{K}|\mathbf{C}}(g) \in \mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}) \\ \mathsf{var}^{-1}_{\mathbf{C}}(\gamma_{\mathbf{K}|\mathbf{C}}(g)) & \text{if } \gamma_{\mathbf{K}|\mathbf{C}}(g) \in \mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) \end{cases}$$

Now we are able to update the concept labels for the transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ and vice versa.
If n is redundant, then add n to the attribute labels of the concept node whose extent equals the attribute extent n^J. The object labels and all other attribute labels does not change.
Otherwise when n is not redundant, then add n to the attribute labels of the new concept node that is generated by the greatest generator concept node $\top^{\mathbf{C}}_{\mathsf{gen}}$. No other attribute labels change. The object labels of a generating concept node are distributed between the embedded old generator concept node and the generated new concept node. The object labels contained in the new attribute extent n^J are precisely the object labels of the new concept node, and the remaining object labels are precisely the object labels of the old generator concept node.
According to Corollarium: Concept Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ and vice versa 3.8 the concept node sets $\mathfrak{N}(\mathbf{K})$ and $\mathfrak{N}(\mathbf{K}|\mathbf{C})$ are subdivided into the old concept nodes $\mathfrak{N}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})$ and $\mathfrak{N}_{\mathsf{old}}(\mathbf{K}|\mathbf{C})$, the varying concept nodes $\mathfrak{N}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$ and the varied concept nodes $\mathfrak{N}_{\mathsf{var}}(\mathbf{K}|\mathbf{C})$, and finally the generator concept nodes $\mathfrak{N}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$ and the new concept nodes $\mathfrak{N}_{\mathsf{new}}(\mathbf{K}|\mathbf{C})$.
Summing up, the following isomorphism are given for the concept node transitions from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ and vice versa, by extending the bijections $\mathsf{old}_{\mathbf{C}}$, $\mathsf{var}_{\mathbf{C}}$ and $\mathsf{new}_{\mathbf{C}}$ from Corollarium: Concept Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ and vice versa 3.8.

$$\mathsf{old}_{\mathbf{C}}: \quad \begin{aligned} &\mathfrak{N}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K}) \hookrightarrow\!\!\!\!\!\rightarrow \mathfrak{N}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) \\ &(A, B, A_\lambda, B_\lambda) \mapsto \begin{cases} (A, B, A_\lambda, B_\lambda) & \text{if } (A, B) \notin \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \\ (A, B, A_\lambda \setminus n^J, B_\lambda) & \text{if } (A, B) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \end{cases} \\ &\left.\begin{matrix} (A, B) \notin \mathfrak{B}_{\mathsf{gen}}(\mathbf{K}|\mathbf{C}) \text{ fi} & (A, B, A_\lambda, B_\lambda) \\ \begin{matrix} (A, B) \in \mathfrak{B}_{\mathsf{gen}}(\mathbf{K}|\mathbf{C}) \\ \text{and } (C, D) = \mathsf{new}_{\mathbf{C}}(\mathsf{old}^{-1}_{\mathbf{C}}(A, B)) \end{matrix} \text{ fi} & (A, B, A_\lambda \dot\cup C_\lambda, B_\lambda) \end{matrix}\right\} \leftarrow\!\shortmid (A, B, A_\lambda, B_\lambda) \end{aligned}$$

$$\mathrm{var}_{\mathbf{C}}\colon \begin{array}{rcl} \mathfrak{N}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) & \hookrightarrow\!\!\!\!\rightarrow & \mathfrak{N}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) \\ (A,B,A_\lambda,B_\lambda) & \mapsto & (A,B\,\dot{\cup}\,\{n\},A_\lambda,B_\lambda) \\ (A,B\setminus\{n\},A_\lambda,B_\lambda) & \leftarrow\!\shortmid & (A,B,A_\lambda,B_\lambda) \end{array}$$

$$\mathrm{new}_{\mathbf{C}}\colon \begin{array}{l} \mathfrak{N}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \hookrightarrow\!\!\!\!\rightarrow \mathfrak{N}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}) \\ (A,B,A_\lambda,B_\lambda) \mapsto \begin{cases} (A\cap n^J, B\,\dot{\cup}\,\{n\}, A_\lambda\cap n^J, B_\lambda) & \text{if } n^J \not\subset A \text{ iff } (A,B)\neq \top^{\mathbf{C}}_{\mathsf{gen}} \\ (n^J, B\,\dot{\cup}\,\{n\}, A_\lambda\cap n^J, B_\lambda\,\dot{\cup}\,\{n\}) & \text{if } n^J \subset A \text{ iff } (A,B) = \top^{\mathbf{C}}_{\mathsf{gen}} \end{cases} \end{array}$$

Although not needed for practical purposes, also the generator concept nodes can be computed from the new concept nodes as follows.

$$\mathrm{gen}_{\mathbf{C}}\colon \begin{array}{l} \mathfrak{N}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}) \hookrightarrow\!\!\!\!\rightarrow \mathfrak{N}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \\ (A,B,A_\lambda,B_\lambda) \mapsto \begin{cases} ((B\setminus\{n\})^I, B\setminus\{n\}, A_\lambda\,\dot{\cup}\,C_\lambda, B_\lambda) & \text{if } \begin{array}{l}(C,D)=\mathrm{gen}_{\mathbf{C}}(A,B) \\ \text{and } n^J\not\subset C \text{ (iff } (C,D)\neq\top^{\mathbf{C}}_{\mathsf{gen}})\end{array} \\ ((B\setminus\{n\})^I, B\setminus\{n\}, A_\lambda\,\dot{\cup}\,C_\lambda, B_\lambda\setminus\{n\}) & \text{if } \begin{array}{l}(C,D)=\mathrm{gen}_{\mathbf{C}}(A,B) \\ \text{and } n^J\subset C \text{ (iff } (C,D)=\top^{\mathbf{C}}_{\mathsf{gen}})\end{array} \end{cases} \end{array}$$

3.1.6 Updating the Reducibility

When n is redundant, then there is no change in the set of the irreducible attributes. Possibly a previously irreducible attribute m in $\mathbf{K}$ can become reducible in $\mathbf{K}|\mathbf{C}$. This can only happen for attributes $m \in M$, whose attribute extent m^I is a superset of n^J. A (previously irreducible) attribute $m \in M$ is reducible in $\mathbf{K}|\mathbf{C}$, iff

$$m^I = n^J \cap \bigcap_{\substack{o\in M \\ o^I \supset m^I}} o^I$$

hold. A more sophisticated answer gives the next theorem. Also the irreducible attributes can be detected in the concept lattice. An attribute m is $\mathbf{K}$-irreducible, iff its attribute concept $\mu_{\mathbf{K}}(m)$ has exactly one upper neighbor $(\mu_{\mathbf{K}}(m))^*$. Then it holds that

$$\begin{aligned}(\mu_{\mathbf{K}}(m))^* &= \bigwedge_{\substack{(A,B)\in\mathfrak{B}(\mathbf{K}) \\ \mu_{\mathbf{K}}(m)<(A,B)}} (A,B) = \left(\bigcap_{\substack{A\in\mathrm{Ext}(\mathbf{K}) \\ m^I\subset A}} A,\dots\right) = \left(\bigcap_{\substack{n\in M \\ m^I\subset n^I}} n^I,\dots\right) \\ &= \left(\left\{n\in M \,\middle|\, m^I\subset n^I\right\}^I,\dots\right) = \left(\left(m^{II}\setminus\left\{n\in M\,\middle|\, n^I = m^I\right\}\right)^I,\dots\right),\end{aligned}$$

since $m^I \subset n^I \Leftrightarrow n \in m^{II}$ and $n^I \neq m^I$.

Theorema: Attribute Reducibility Update 3.21

(I) Each $\mathbf{K}$-reducible attribute is also $\mathbf{K}|\mathbf{C}$-reducible. A $\mathbf{K}$-irreducible attribute m is $\mathbf{K}|\mathbf{C}$-reducible, iff $\mu_{\mathbf{K}}(m) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$ and $(\mu_{\mathbf{K}}(m))^* \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{o}\neg\mathsf{g}}(\mathbf{K})$, and furthermore at least one superconcept of $(\mu_{\mathbf{K}}(m))^*$ is a generator concept.

(II) Each $\mathbf{K}|\mathbf{C}$-irreducible attribute from $\mathbf{K}$ is also $\mathbf{K}$-irreducible. A $\mathbf{K}|\mathbf{C}$-reducible attribute $m \in M$ is $\mathbf{K}$-irreducible, iff $\mu_{\mathbf{K}|\mathbf{C}}(m) \in \mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C})$ has exactly one old upper neighbor ω and overthis only new upper neighbors, whose generators are superconcepts of ω.

APPROBATIO (I) First, if m is a $\mathbf{K}$-reducible attribute, then the attribute extent m^I can be obtained by an intersection of attribute extents $\bigcap_{m\in B} m^I$ with $m \notin B$. Obviously then also

$$m^{(I\dot{\cup}J)} = m^I = \bigcap_{m\in B} m^I = \bigcap_{m\in B} m^{(I\dot{\cup}J)}$$

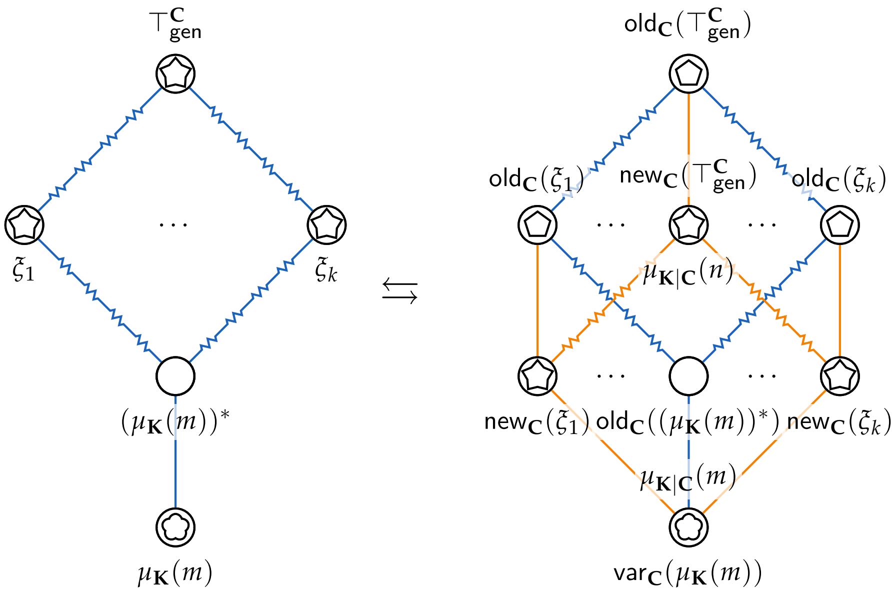
holds, hence m is $\mathbf{K}|\mathbf{C}$-reducible. Second, let m be a $\mathbf{K}$-irreducible attribute.

(⇒) Suppose m is $\mathbf{K}|\mathbf{C}$-reducible. If $\mu_{\mathbf{K}}(m)$ were an old concept, then $\mu_{\mathbf{K}|\mathbf{C}}(m) = \mathsf{old}_{\mathbf{C}}(\mu_{\mathbf{K}}(m))$ and the set of upper neighbors does not change according to Theorema: Neighborhood Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ 3.17. Thus, the irreducibility of m in $\mathbf{K}$ implies the irreducibility of m in $\mathbf{K}|\mathbf{C}$. Contradiction! Hence, the attribute concept $\mu_{\mathbf{K}}(m)$ must be varying. By Corollarium: Concept Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ and vice versa 3.8, there are no other old or varied upper neighbors of $\mu_{\mathbf{K}|\mathbf{C}}(m)$. If $(\mu_{\mathbf{K}}(m))^*$ would be a varying or generating concept, then

$$\mu_{\mathbf{K}|\mathbf{C}}(m) = \mathsf{var}_{\mathbf{C}}(\mu_{\mathbf{K}}(m)) \prec \begin{cases} \mathsf{var}_{\mathbf{C}}((\mu_{\mathbf{K}}(m))^*) & \text{if } (\mu_{\mathbf{K}}(m))^* \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) \\ \mathsf{new}_{\mathbf{C}}((\mu_{\mathbf{K}}(m))^*) & \text{if } (\mu_{\mathbf{K}}(m))^* \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \end{cases}$$

holds. Let $(A,B) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$ with $(A,B) \neq (\mu_{\mathbf{K}}(m))^*$, such that $\mathsf{new}_{\mathbf{C}}(A,B)$ covers $\mu_{\mathbf{K}|\mathbf{C}}(m)$, then $\mu_{\mathbf{K}}(m)$ must be a lower neighbor of (A,B) and there is no varying or generating concept between them. So $\mu_{\mathbf{K}}(m) \prec (\mu_{\mathbf{K}}(m))^* < (A,B)$ must hold, but this is a contradiction. In summary, $\mathsf{var}_{\mathbf{C}}((\mu_{\mathbf{K}}(m))^*)$ or $\mathsf{new}_{\mathbf{C}}((\mu_{\mathbf{K}}(m))^*)$ respectivelly must be the unique upper neighbor of $\mu_{\mathbf{K}|\mathbf{C}}(m)$, and m would be $\mathbf{K}|\mathbf{C}$-irreducible. Contradiction! Hence $(\mu_{\mathbf{K}}(m))^*$ must be an old non-generator concept. Finally if there were no generating superconcept above $(\mu_{\mathbf{K}}(m))^*$, then $\mathsf{old}_{\mathbf{C}}((\mu_{\mathbf{K}}(m))^*)$ were the only upper neighbor of $\mu_{\mathbf{K}|\mathbf{C}}(m)$, *i.e.* m would be $\mathbf{K}|\mathbf{C}$-irreducible. Contradiction!

(⇐) Suppose the attribute concept $\mu_{\mathbf{K}}(m)$ is a varying concept and its unique upper neighbor $(\mu_{\mathbf{K}}(m))^*$ is an old non-generator concept that has at least one generator superconcept. Denote the minimal ones of these generator superconcepts by $\xi_1, \xi_2, \ldots, \xi_k$. Then the following structure on the left side can be found within the concept lattice of $\mathbf{K}$. Neighboring concept nodes are connected by straight line segments and comparable concepts are connected by zig zag line segments.



Then according to Theorema: Neighborhood Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ 3.17 (IX) the new concepts $\mathsf{new}_{\mathbf{C}}(\xi_1)$, …, $\mathsf{new}_{\mathbf{C}}(\xi_k)$ must cover the varied attribute concept $\mathsf{var}_{\mathbf{C}}(\mu_{\mathbf{K}}(m))$. This is due to the fact, that no varying concept can be greater than an old concept, and the generators $\xi_1, \ldots, \xi_k$ are minimal. Furthermore $(\mu_{\mathbf{K}}(m))^*$ is the unique upper neighbor of $\mu_{\mathbf{K}}(m)$, hence there cannot be any varying or generating concept between $\mu_{\mathbf{K}}(m)$ and each ξ_j. In summary, the transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ changes the concept lattice structure as displayed in the right diagram. Obviously $\mu_{\mathbf{K}|\mathbf{C}}(m) = \mathsf{var}_{\mathbf{C}}(\mu_{\mathbf{K}}(m))$ has more than one upper neighbor, hence m is $\mathbf{K}|\mathbf{C}$-reducible.

(II) Let first $m \in M$ be a $\mathbf{K}|\mathbf{C}$-irreducible attribute. Then m must also be $\mathbf{K}$-irreducible, as otherwise m were $\mathbf{K}|\mathbf{C}$-irreducible by (I). Second, let $m \in M$ be $\mathbf{K}|\mathbf{C}$-reducible attribute.

(⇒) Suppose m is $\mathbf{K}$-irreducible. Then $\mu_{\mathbf{K}|\mathbf{C}}(m)$ must be a varied concept. Otherwise $\mu_{\mathbf{K}}(m) = \mathsf{old}^{-1}_{\mathbf{C}}(\mu_{\mathbf{K}|\mathbf{C}}(m))$ were an old concept and this is a contradiction to (I). If $\mu_{\mathbf{K}|\mathbf{C}}(m)$ had more than one old (and thus non-generating) upper neighbor in $\mathfrak{B}(\mathbf{K}|\mathbf{C})$, then the according old concepts in $\mathfrak{B}(\mathbf{K})$ would cover $\mu_{\mathbf{K}}(m)$. This is a contradiction to the $\mathbf{K}$-irreducibility of m. So $\mu_{\mathbf{K}|\mathbf{C}}(m)$ has exactly one old upper neighbor $\omega \in \mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C})$, all other upper neighbors must be varied or new concepts. If a varied concept covers $\mu_{\mathbf{K}|\mathbf{C}}(m)$, then its appropriate varying concept covers $\mu_{\mathbf{K}}(m)$ as well. Again, this is a contradiction to the **K**-irreducibility. So all other upper neighbors must be new concepts. If there were any new concept $\nu \in \mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C})$ whose generator ξ is not a superconcept of ω, then $\mu_{\mathbf{K}}(m)$ would be covered by $\mathsf{old}_{\mathbf{C}}^{-1}(\xi)$. Then $\mu_{\mathbf{K}}(m)$ had at least two upper neighbors and this contradicts the **K**-irreducibility.

($\Leftarrow$) Suppose $\mu_{\mathbf{K}|\mathbf{C}}(m)$ varies and has exactly one upper neighbor ω and overthis only new upper neighbors $\nu_1, \ldots, \nu_k$, whose generators are greater than ω. Then choose $\xi_j := \mathsf{gen}_{\mathbf{C}}(\nu_j)$ and the same structure as in the right diagram above occurs, and by Corollarium: Neighborhood Transition from $\mathbf{K}|\mathbf{C}$ to $\mathbf{K}$ 3.18 $\mathsf{old}_{\mathbf{C}}^{-1}(\omega) = (\mu_{\mathbf{K}}(m))^*$ must be the unique upper neighbor of $\mu_{\mathbf{K}}(m)$. This means m is **K**-irreducible. ■

Updating the reducibility of the objects is not as easy as for the attributes. For both directions of transition, previously reducible objects can become irreducible, and also previously irreducible objects can become reducible.

Theorema: Object Reducibility Update from K to K|C 3.22

(I) Let g be **K**-reducible. Then g is $\mathbf{K}|\mathbf{C}$-irreducible, iff one of the following statements hold:

(A) $\gamma_{\mathbf{K}}(g)$ is a generator concept with $g \notin n^J$, and all lower neighbors are varying concepts.

(B) $\gamma_{\mathbf{K}}(g)$ is a generator concept with $g \in n^J$, and exactly one lower neighbor is a varying or generator concept.

(II) Let g be **K**-irreducible. Then g is $\mathbf{K}|\mathbf{C}$-irreducible too, iff one of the following statements hold:

(A) $\gamma_{\mathbf{K}}(g)$ is an old non-generator concept.

(B) $\gamma_{\mathbf{K}}(g)$ is a varying concept.

(C) $\gamma_{\mathbf{K}}(g)$ is a generator concept, and its unique lower neighbor is a varying concept.

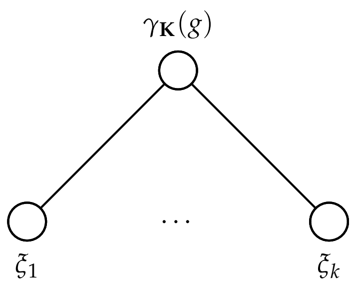
(D) $\gamma_{\mathbf{K}}(g)$ is a generator concept, and its unique lower neighbor is an old concept, and $g \in n^J$, and there is either exactly one varying concept β, such that

$$\beta < \gamma_{\mathbf{K}}(g) \text{ and } \nexists_{\alpha \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \dot\cup \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})} \beta < \alpha < \gamma_{\mathbf{K}}(g),$$

or exactly one generator concept ν, such that

$$\nu < \gamma_{\mathbf{K}}(g) \text{ and } \nexists_{\alpha \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})} \nu < \alpha < \gamma_{\mathbf{K}}(g).$$

APPROBATIO (I) Let g be **K**-reducible, *i.e.* $\gamma_{\mathbf{K}}(g)$ has as least two lower neighbors, denoted by $\xi_1, \ldots, \xi_k$.

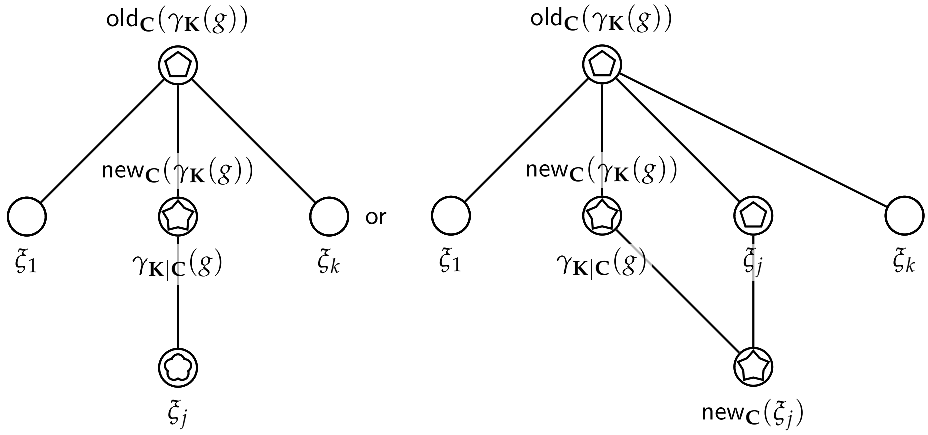


If $\gamma_{\mathbf{K}}(g)$ is a varying or an old non-generator concept, then according to Theorema: Neighborhood Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ 3.17 no structural changes in the set of lower neighbors occur, and thus g must also be reducible in $\mathbf{K}|\mathbf{C}$. (Please remind, that then $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ equals $\mathsf{var}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$ or $\mathsf{old}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$, respectively.)

Now let $\gamma_{\mathbf{K}}(g) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$. First, suppose $g \notin n^J$. Then Corollarium: Transition of Object Concepts and Attribute Concepts 3.20 yields $\gamma_{\mathbf{K}|\mathbf{C}}(g) = \mathsf{old}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$. If $\xi_j \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K})$, then $\mathsf{old}_{\mathbf{C}}(\xi_j) \prec \gamma_{\mathbf{K}|\mathbf{C}}(g)$. Otherwise, if $\xi_j \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})$, then

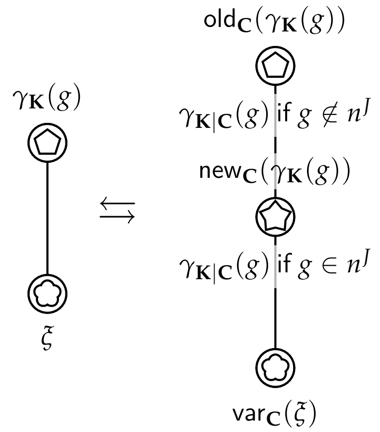
$$\mathsf{var}_{\mathbf{C}}(\xi_j) \prec \mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g)) \prec \gamma_{\mathbf{K}|\mathbf{C}}(g).$$

Thus, if all ξ_j are varying, then $\mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$ is the unique lower neighbor of $\gamma_{\mathbf{K}|\mathbf{C}}(g)$, *i.e.* g is irreducible in $\mathbf{K}|\mathbf{C}$. Obviously $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ must have more than one lower neighbor in all other cases (when at least one ξ_j is old), *i.e.* g is then $\mathbf{K}|\mathbf{C}$-reducible as well.
Second, suppose $\gamma_{\mathbf{K}}(g) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$ and $g \in n^J$. Then Corollarium: Transition of Object Concepts and Attribute Concepts 3.20 yields $\gamma_{\mathbf{K}|\mathbf{C}}(g) = \mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$. Furthermore, $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ has exactly one lower neighbor (hence g is $\mathbf{K}|\mathbf{C}$-irreducible), if exactly one ξ_j is varying or a generator.



Clearly g must be reducible in $\mathbf{K}|\mathbf{C}$ in all other cases.

(II) Let g be $\mathbf{K}$-irreducible. Then $\gamma_{\mathbf{K}}(g)$ has exactly one lower neighbor $\xi := (\gamma_{\mathbf{K}}(g))^*$. If $\gamma_{\mathbf{K}}(g)$ is a varying or an old non-generator concept, then again no structural changes occur in the set of lower neighbors. Hence g is also irreducible in $\mathbf{K}|\mathbf{C}$ then.
So let $\gamma_{\mathbf{K}}(g) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})$. First, suppose ξ is varying. The local changes in neighborhood structure are depicted below.

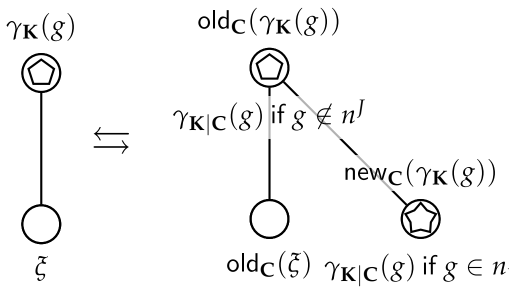


Clearly, when $g \notin n^J$, then $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ has exactly one lower neighbor $\mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$, *i.e.* g is $\mathbf{K}|\mathbf{C}$-irreducible. Otherwise, if $g \in n^J$, the object concept of g in $\mathbf{K}|\mathbf{C}$ is $\mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$. Then the varied concept $\mathsf{var}_{\mathbf{C}}(\xi)$ must be the unique lower neighbor of the new object concept, since:

- There cannot be any generator under a varying concept, thus $\gamma_{\mathbf{K}}(g)$ must be a minimal generator. Hence, no new lower neighbors of $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ exist.

- If there were any other varied concept ξ_1 below $\gamma_{\mathbf{K}|\mathbf{C}}(g)$, *i.e.* $\xi_1 \prec \gamma_{\mathbf{K}|\mathbf{C}}(g)$, then $\mathsf{var}_{\mathbf{C}}^{-1}(\xi_1) < \gamma_{\mathbf{K}}(g)$ and there is no generator or varying concept between them. Also this implies $\mathsf{var}_{\mathbf{C}}^{-1}(\xi_1) < \xi \prec \gamma_{\mathbf{K}}(g)$ since ξ is the only lower neighbor, but this is a contradiction as ξ is varying.

In summary, g is $\mathbf{K}|\mathbf{C}$-irreducible too, when $\gamma_{\mathbf{K}}(g)$ is a generator concept and its unique lower neighbor is varying.
Second, suppose ξ is old, then the neighborhood changes locally as shown below.

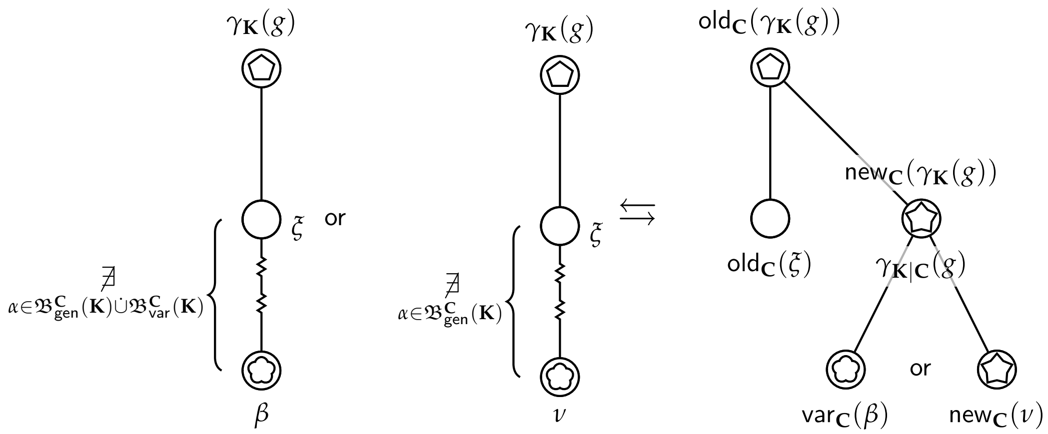


Obviously, g is $\mathbf{K}|\mathbf{C}$-reducible for $g \notin n^J$. Otherwise, when $g \in n^J$, one has to look on the lower neighbors of the new concept $\gamma_{\mathbf{K}|\mathbf{C}}(g) = \mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$. At first it can only exist new or varied lower neighbors of $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ and according to Theorema: Neighborhood Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ 3.17 it holds

$$\mathsf{new}_{\mathbf{C}}(\nu) \prec \gamma_{\mathbf{K}|\mathbf{C}}(g) \Leftrightarrow \nu < \gamma_{\mathbf{K}}(g) \text{ and } \nexists_{\alpha \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K})} \nu < \alpha < \gamma_{\mathbf{K}}(g)$$

$$\text{and } \mathsf{var}_{\mathbf{C}}(\beta) \prec \gamma_{\mathbf{K}|\mathbf{C}}(g) \Leftrightarrow \beta < \gamma_{\mathbf{K}}(g) \text{ and } \nexists_{\alpha \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{gen}}(\mathbf{K}) \dot\cup \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K})} \beta < \alpha < \gamma_{\mathbf{K}}(g).$$

So, when speaking graphically, we gather the following situations depicted below.



g is $\mathbf{K}|\mathbf{C}$-irreducible, iff the new object concept $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ has exactly one lower neighbor, *i.e.* iff there is only one β or ν as above. In all other cases g must be reducible in $\mathbf{K}|\mathbf{C}$. ∎

Theorema: Object Reducibility Update from K|C to K 3.23

(I) Let g be $\mathbf{K}|\mathbf{C}$-reducible. Then g is $\mathbf{K}$-irreducible, iff one of the following statements hold:

(A) $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a generator concept, and $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ has exactly one old lower neighbor β, and furthermore for each varying lower neighbor ξ of $\mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$ it holds that $\xi < \beta$.

(B) $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a new concept, and the appropriate generator above $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ has exactly one old lower neighbor β, and furthermore for each varying lower neighbor ξ of $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ it holds that $\xi < \beta$

(C) $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a new concept, and the appropriate generator above $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ has no old lower neighbor, and furthermore there is exactly one varying lower neighbor of $\gamma_{\mathbf{K}|\mathbf{C}}(g)$.

(II) Let g be $\mathbf{K}|\mathbf{C}$-irreducible. Then g is also $\mathbf{K}$-irreducible, iff one of the following statements hold:

(A) $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is an old non-generator concept.

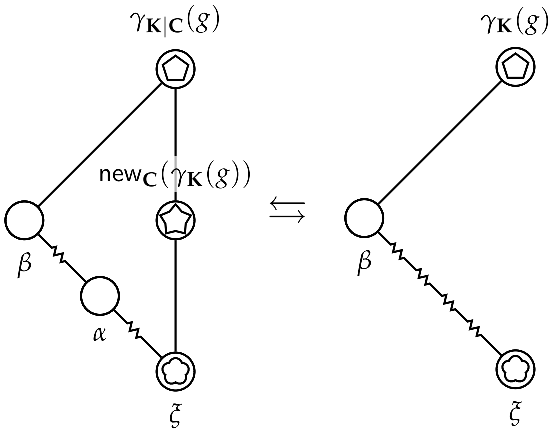
(B) $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a varying concept.

(C) $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a generator concept, and its unique lower neighbor has exactly one lower neighbor.

(D) $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a new concept, and its generator has no old lower neighbor.

(E) $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a new concept, and its generator has exactly one old lower neighbor β, such that the unique lower neighbor $\alpha \prec \gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a subconcept of β.

APPROBATIO (I) Let g be $\mathbf{K}|\mathbf{C}$-reducible. The statements (A) and (B) hold, since there are no structural changes in the set of lower neighbors of each old non-generator or varied concept. Now let $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ be a generator concept. Then $\gamma_{\mathbf{K}}(g) = \mathsf{old}_{\mathbf{C}}^{-1}(\gamma_{\mathbf{K}|\mathbf{C}}(g))$ and $g \notin n^J$ hold. $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ must have at least one old lower neighbor besides $\mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$. Obviously then g is $\mathbf{K}$-irreducible, when $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ has exactly one old lower neighbor β, and furthermore for each varying lower neighbor ξ of $\mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$ there is an old non-generator concept α, such that $\xi < \alpha < \gamma_{\mathbf{K}|\mathbf{C}}(g)$, *i.e.* $\xi < \alpha \leq \beta$ and thus $\xi < \beta$, holds. This can be seen with the following picture.



Finally suppose $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a new concept. Then $\gamma_{\mathbf{K}}(g) = \mathsf{gen}_{\mathbf{C}}(\gamma_{\mathbf{K}|\mathbf{C}}(g))$ and $g \in n^J$ hold. By the same arguments, we conclude that $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ can only have exactly one lower neighbor, when the appropriate generator above $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ has exactly one old lower neighbor β, and furthermore for each varying lower neighbor ξ of $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ it holds that $\xi < \beta$, or when the appropriate generator above $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ has no old lower neighbor, and furthermore there is exactly one varying lower neighbor ξ of $\gamma_{\mathbf{K}|\mathbf{C}}(g)$.

(II) Let g be $\mathbf{K}|\mathbf{C}$-irreducible. The statements (A) and (B) hold, since there are no structural changes in the set of lower neighbors of each old non-generator or varied concept.
When $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a generator concept, then it can only have $\mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$ as its unique lower neighbor. Clearly then g is $\mathbf{K}$-irreducible, when $\mathsf{new}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g))$ has exactly one (varying) lower neighbor.
Last but not least, suppose $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a new concept with its unique lower neighbor α. If its generator $\mathsf{old}_{\mathbf{C}}(\mathsf{gen}_{\mathbf{C}}(\gamma_{\mathbf{K}|\mathbf{C}}(g)))$ has more than one old lower neighbor in the concept lattice of $\mathbf{K}|\mathbf{C}$, then also $\gamma_{\mathbf{K}}(g) = \mathsf{gen}_{\mathbf{C}}(\gamma_{\mathbf{K}|\mathbf{C}}(g))$ would have more than one lower neighbor and g would thus be $\mathbf{K}$-reducible. So suppose that there is no old lower neighbor, then g must be $\mathbf{K}$-irreducible, as then $\mathsf{var}_{\mathbf{C}}^{-1}(\alpha)$ is the unique lower neighbor of $\gamma_{\mathbf{K}}(g)$. If there is exactly one old lower neighbor β, then g is $\mathbf{K}$-irreducible, if α is a subconcept of β. ■

3.1.7 Updating the Arrows

3.24 **Lemma: Up Arrows in K and K|C**

For an object $g \in G$ and an attribute $m \in M$ the following statement hold:

$$g \nearrow_{\mathbf{K}|\mathbf{C}} m \Leftrightarrow \begin{cases} g \nearrow_{\mathbf{K}} m & \text{if } m^I \not\subset n^J \\ g \nearrow_{\mathbf{K}} m \text{ and } g \in n^J & \text{if } m^I \subset n^J \end{cases}$$

APPROBATIO This follows easily from

$$g \nearrow_{\mathbf{K}|\mathbf{C}} m \Leftrightarrow \underbrace{g \notin m^I \text{ and } \forall_{p \in M} (m^I \subset p^I \Rightarrow g \in p^I)}_{\Leftrightarrow g \nearrow_{\mathbf{K}} m} \text{ and } (m^I \subset n^J \Rightarrow g \in n^J).$$ ■

We conclude, that up arrows do not change in columns m^I which are no subset of the new column n^J.

Theorema: Up Arrow Transition 3.25

Let $g \in G$ and $m \in M$ such that $m^I \subset n^J$.

(I) For the arrow transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ it holds:

$$g \nearrow_{\mathbf{K}|\mathbf{C}} m \Leftrightarrow g \nearrow_{\mathbf{K}} m \text{ and } g \in n^J.$$

(II) For the arrow transition from $\mathbf{K}|\mathbf{C}$ to $\mathbf{K}$ it holds:
If $g \in n^J$, then

$$g \nearrow_{\mathbf{K}} m \Leftrightarrow g \nearrow_{\mathbf{K}|\mathbf{C}} m.$$

If $g \notin n^J$, then $g \nearrow_{\mathbf{K}} m$ holds, iff one of the following statements hold:

(A) m is $\mathbf{K}|\mathbf{C}$-reducible, and $\mu_{\mathbf{K}|\mathbf{C}}(m) \in \mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C})$ has exactly one old upper neighbor ω and overthis only new upper neighbors, whose generators are superconcepts of ω, and furthermore $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a subconcept of ω.

(B) m is $\mathbf{K}|\mathbf{C}$-irreducible, $((\mu_{\mathbf{K}|\mathbf{C}}(m))^* \in \mathfrak{B}_{\mathsf{new}}(\mathbf{K}|\mathbf{C})$ and $\gamma_{\mathbf{K}|\mathbf{C}}(g) \in \mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C})$ is a subconcept of the generator $\mathsf{old}_{\mathbf{C}}(\mathsf{gen}_{\mathbf{C}}((\mu_{\mathbf{K}|\mathbf{C}}(m))^*))$.

APPROBATIO (I) by the previous lemma.

(II) In case $g \in n^J$ this follows from the preceding lemma as well. Suppose $g \notin n^J$. Then Corollarium: Transition of Object Concepts and Attribute Concepts 3.20 yields

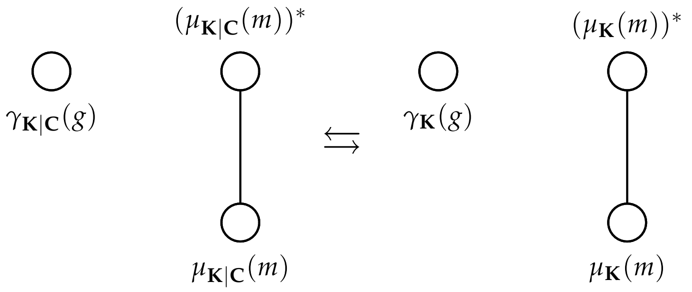
$$\gamma_{\mathbf{K}|\mathbf{C}}(g) = \begin{cases} \mathsf{old}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g)) & \text{if } \gamma_{\mathbf{K}}(g) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{old}}(\mathbf{K}) \\ \mathsf{var}_{\mathbf{C}}(\gamma_{\mathbf{K}}(g)) & \text{if } \gamma_{\mathbf{K}}(g) \in \mathfrak{B}^{\mathbf{C}}_{\mathsf{var}}(\mathbf{K}) \end{cases}.$$

(A) Let m be $\mathbf{K}|\mathbf{C}$-reducible. $g \nearrow_{\mathbf{K}} m$ can only hold, when m is irreducible in $\mathbf{K}$, *i.e.* when $\mu_{\mathbf{K}|\mathbf{C}}(m) \in \mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C})$ has exactly one old upper neighbor ω and overthis only new upper neighbors, whose generators are superconcepts of ω, according to Theorema: Attribute Reducibility Update 3.21. Then $\mathsf{old}_{\mathbf{C}}^{-1}(\omega)$ is the unique upper neighbor of $\mu_{\mathbf{K}}(m)$. Furthermore, $\gamma_{\mathbf{K}|\mathbf{C}}(g) \leq \omega$ holds, iff $\gamma_{\mathbf{K}}(g) \leq (\mu_{\mathbf{K}}(m))^*$, *i.e.* iff $g \nearrow_{\mathbf{K}} m$.

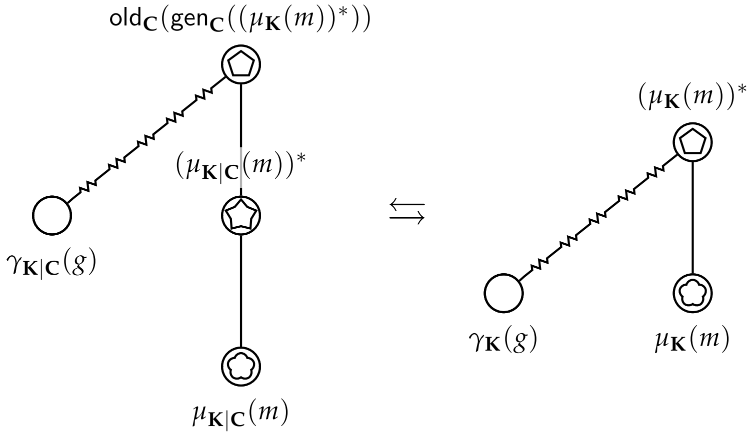
(B) When m is $\mathbf{K}|\mathbf{C}$-irreducible, then m is also $\mathbf{K}$-irreducible by Theorema: Attribute Reducibility Update 3.21. Furthermore, $g \notin n^J$ implies $g \not\nearrow_{\mathbf{K}|\mathbf{C}} m$, *i.e.* $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is no subconcept of $(\mu_{\mathbf{K}|\mathbf{C}}(m))^*$.
If $(\mu_{\mathbf{K}|\mathbf{C}}(m))^*$ is an old concept, then $\mathsf{old}_{\mathbf{C}}^{-1}((\mu_{\mathbf{K}|\mathbf{C}}(m))^*)$ is the unique upper neighbor of

$$\mu_{\mathbf{K}}(m) = \begin{cases} \mathsf{old}_{\mathbf{C}}^{-1}(\mu_{\mathbf{K}|\mathbf{C}}(m)) & \text{if } \mu_{\mathbf{K}|\mathbf{C}}(m) \in \mathfrak{B}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) \\ \mathsf{var}_{\mathbf{C}}^{-1}(\mu_{\mathbf{K}|\mathbf{C}}(m)) & \text{if } \mu_{\mathbf{K}|\mathbf{C}}(m) \in \mathfrak{B}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) \end{cases}.$$

Then $\gamma_{\mathbf{K}}(g)$ is a subconcept of $(\mu_{\mathbf{K}}(m))^*$, iff $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a subconcept of $(\mu_{\mathbf{K}|\mathbf{C}}(m))^*$. As this cannot occur according to the preconditions, $g \not\nearrow_{\mathbf{K}} m$ must hold. If $(\mu_{\mathbf{K}|\mathbf{C}}(m))^*$ is a varied concept, then $\mathsf{var}_{\mathbf{C}}^{-1}((\mu_{\mathbf{K}|\mathbf{C}}(m))^*)$ is the unique upper neighbor of $\mu_{\mathbf{K}}(m) = \mathsf{var}_{\mathbf{C}}^{-1}(\mu_{\mathbf{K}|\mathbf{C}}(m))$. Then $\gamma_{\mathbf{K}}(g)$ is smaller than $(\mu_{\mathbf{K}}(m))^*$, iff $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is a subconcept of $(\mu_{\mathbf{K}|\mathbf{C}}(m))^*$. Thus, $g \not\nearrow_{\mathbf{K}} m$ as well in this case.



If the unique upper neighbor $(\mu_{\mathbf{K}|\mathbf{C}}(m))^*$ is a new concept, then according to Corollarium: Neighborhood Transition from $\mathbf{K}|\mathbf{C}$ to $\mathbf{K}$ 3.18 $\mathsf{gen}_{\mathbf{C}}((\mu_{\mathbf{K}|\mathbf{C}}(m))^*)$ must be the unique upper neighbor of $\mu_{\mathbf{K}}(m) = \mathsf{var}_{\mathbf{C}}^{-1}(\mu_{\mathbf{K}|\mathbf{C}}(m))$. Furthermore $\gamma_{\mathbf{K}}(g)$ can only be a subconcept of $(\mu_{\mathbf{K}}(m))^*$, if it is an old concept and a subconcept of the generator. (If $\gamma_{\mathbf{K}}(g)$ would be varying and smaller than the generator, $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ must be smaller than the new generated concept as well, in contradiction to the preconditions.)



In summary, $g \nearrow_{\mathbf{K}} m$ holds in this case, iff $\gamma_{\mathbf{K}|\mathbf{C}}(g)$ is an old concept and smaller than the generator of the upper neighbor of $\mu_{\mathbf{K}|\mathbf{C}}(m)$. ∎

Due to a lack of time, the arrow transition from $\swarrow_{\mathbf{K}}$ to $\swarrow_{\mathbf{K}|\mathbf{C}}$ and vice versa remains an open problem in this document.

3.1.8 Updating the Seed Vectors

Diagram Transition from K to K|C

If the new attribute n is not redundant in $\mathbf{K}|\mathbf{C}$, a new seed vector for n has to be chosen. Before this, check if some of the old attributes of $\mathbf{K}$ become reducible in the updated context $\mathbf{K}|\mathbf{C}$ by means of Theorema: Attribute Reducibility Update 3.21 and set the appropriate seed vectors to the null vector. Then choose a new seed vector σ for n. This can be done by applying a quality metric, calculating an appropriate heatmap for the new attribute concept $\gamma_{\mathbf{K}|\mathbf{C}}(n)$ and selecting a best position. In the end, calculate the positions of the new concepts by shifting the generator positions by σ and modify the positions of the varied concepts by shifting their positions by σ.

Diagram Transformation from K|C to K

When the removed attribute n was not redundant in $\mathbf{K}|\mathbf{C}$, its appropriate seed vector has to be removed from the seed map. Also, some of the remaining previously reducible attributes in M can become irreducible in $\mathbf{K}$ by Theorema: Attribute Reducibility Update 3.21, and a new seed vector has to be introduced for them.

3.1.9 Complete iFox Algorithm

The following pseudocode algorithm describes the update process for attribute additive labeled concept diagrams. The algorithm is called iFox.

Adding a new column C to a context K

For a formal context $\mathbf{K}$ and a new column $\mathbf{C}$ the addition of $\mathbf{C}$ to $\mathbf{K}$ can be done in several steps: Firstly, determine the partition of the set of all formal concepts into the old concepts, varying concepts and generator concepts. Then update the formal concepts according to the mapping $\mathsf{old}_{\mathbf{C}}$, $\mathsf{new}_{\mathbf{C}}$ and $\mathsf{var}_{\mathbf{C}}$ as defined in Corollarium: Concept Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ and vice versa 3.8 or Corollarium: Concept Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ and vice versa 3.8 respectively. Also, the labels of the formal concepts are updated. This again is no expensive operation, as only one concept node can get a new attribute label (namely the supremum of all generators, if there are any), and object labels are pushed downwards at the border between generating and non-generating formal concept nodes.

Input: $\mathfrak{N}(\mathbf{K}), \mathbf{C}$

```
for (N = (A, B, A_λ, B_λ) ∈ 𝔑(K))
    if (A ⊈ n^J)
        if ((A ∩ n^J)^I = B)
            𝔑^C_gen(K) ← ∪ {N}
        else
            𝔑^C_o¬g(K) ← ∪ {N}
        end if
    else
        𝔑^C_var(K) ← ∪ {N}
    end if
end for

for (N = (A, B, A_λ, B_λ) ∈ 𝔑^C_gen(K))
    𝔑(K|C) ← ∪ {(A, B, A_λ \ n^J, B_λ)}
    if (n^J ⊂ A)
        𝔑(K|C) ← ∪ {(n^J, B ∪̇ {n}, A_λ ∩ n^J, B_λ ∪̇ {n})}
    else
        𝔑(K|C) ← ∪ {(A ∩ n^J, B ∪̇ {n}, A_λ ∩ n^J, B_λ)}
    end if
end for

for (N ∈ 𝔑^C_o¬g(K))
    𝔑(K|C) ← ∪ {N}
end for

for (N = (A, B, A_λ, B_λ) ∈ 𝔑^C_var(K))
    𝔑(K|C) ← ∪ {(A, B ∪̇ {n}, A_λ, B_λ)}
end for
```

Output: $\mathfrak{N}(\mathbf{K}|\mathbf{C})$

The first for loop is the complexest part. The for loop runs $|\mathfrak{B}(\mathbf{K})|$ times. The evaluation of the condition in the first if statement needs at most $|G|$ operations. Finally, the evaluation of the condition in the second if statement needs at most $|G| \cdot |M|$ operations for the computation of $(A \cap n^J)^I$ and at most $2 \cdot |M|$ operations for the equality check. In summary at most $|\mathfrak{B}(\mathbf{K})| \cdot (|G| + (|G| \cdot |M| + 2 \cdot |M|))$ operations are necessary, hence the worst case time complexity is

$$\mathcal{O}(|\mathfrak{B}(\mathbf{K})| \cdot (|G| + (|G| \cdot |M| + 2 \cdot |M|))) = \mathcal{O}(|\mathfrak{B}(\mathbf{K})| \cdot |G| \cdot |M|).$$

In the second step update the neighborhood relation for the updated set of formal concepts, as described in Theorema: Neighborhood Transition from $\mathbf{K}$ to $\mathbf{K}|\mathbf{C}$ 3.17. There are subrelations (or better said: "blocks" in the binary matrix describing the neighborhood relation) that does not change, some subrelation blocks can be copied onto a new subrelation, and other not very difficult and cost-intensive operations. In the last step the positions of the formal concepts are updated by introducing a new seed vector for the new attribute n, if it is irreducible *w.r.t.* M, and furthermore for an layout optimization also seed vectors are removed whose attributes are reducible *w.r.t.* n.

Removing a column C from a context K|C

The update process is done in a similar way as for adding **C**. First, determine the partition of the concept node set of $\mathbf{K}|\mathbf{C}$ into the old, varied and new ones. By means of them then calculate the concept node set of $\mathbf{K}$.

Input: $\mathfrak{N}(\mathbf{K}|\mathbf{C})$

for $(N = (A, B, A_\lambda, B_\lambda) \in \mathfrak{N}(\mathbf{K}|\mathbf{C}))$
    if $(n \notin B)$
        $\mathfrak{N}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) \leftarrow \cup \{N\}$
    else
        if $((B \setminus \{n\})^I = A)$
            $\mathfrak{N}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}) \leftarrow \cup \{N\}$
        else
            $\mathfrak{N}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}) \leftarrow \cup \{N\}$
        end if
    end if
end for

for $(N_1 \in \mathfrak{N}_{\mathsf{new}}(\mathbf{K}|\mathbf{C}))$
    for $(N_2 \succ N_1)$
        if $(N_2 \in \mathfrak{N}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}))$
            $\mathfrak{N}_{\mathsf{gen}}(\mathbf{K}|\mathbf{C}) \leftarrow \cup \{N_2\}$
            $\mathsf{new}_{\mathbf{C}}(N_2) \leftarrow N_1$
        end if

    end for

end for

$\mathfrak{N}_{\mathsf{o}\neg\mathsf{g}}(\mathbf{K}|\mathbf{C}) \leftarrow \mathfrak{N}_{\mathsf{old}}(\mathbf{K}|\mathbf{C}) \setminus \mathfrak{N}_{\mathsf{gen}}(\mathbf{K}|\mathbf{C})$

for $(N \in \mathfrak{N}_{\mathsf{o}\neg\mathsf{g}}(\mathbf{K}|\mathbf{C}))$
    $\mathfrak{N}(\mathbf{K}) \leftarrow \cup \{N\}$
end for

for $(N = (A, B, A_\lambda, B_\lambda) \in \mathfrak{N}_{\mathsf{gen}}(\mathbf{K}|\mathbf{C}))$
    $(C, D, C_\lambda, D_\lambda) \leftarrow \mathsf{new}_{\mathbf{C}}(N)$
    $\mathfrak{N}(\mathbf{K}) \leftarrow \cup \{(A, B, A_\lambda \dot{\cup} C_\lambda, B_\lambda)\}$
end for

for $(N = (A, B, A_\lambda, B_\lambda) \in \mathfrak{N}_{\mathsf{var}}(\mathbf{K}|\mathbf{C}))$
    $\mathfrak{N}(\mathbf{K}) \leftarrow \cup \{(A, B \setminus \{n\}, A_\lambda, B_\lambda)\}$
end for

Output: $\mathfrak{N}(\mathbf{K})$

By similar arguments as above, the worst case time complexity for the first for is $\mathcal{O}(|\mathfrak{B}(\mathbf{K})| \cdot |G| \cdot |M|)$. The second for loop for the determination of the generator concepts cycles at most $|\mathfrak{B}(\mathbf{K})| \cdot |\mathfrak{B}(\mathbf{K})|$ times, so in summary, the worst case time complexity is

$$\mathcal{O}(|\mathfrak{B}(\mathbf{K})| \cdot (|\mathfrak{B}(\mathbf{K})| + |G| \cdot |M|)).$$

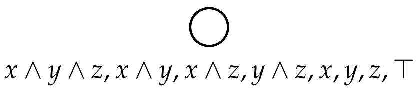
Second, update the neighborhood matrix by removing all columns for new concepts and update one block, *viz.* determine which generators cover varying concept nodes. Finally update the seed map, if **C** was not redundant. Possibly then some remaining attributes become irreducible, see Theorema: Attribute Reducibility Update 3.21 for details. For these newly irreducible attributes add an appropriate seed vector, that can be found by a search for a best point within a heatmap *w.r.t.* some chosen quality metric.

3.1.10 An Example: Stepwise Construction of FCD(3)

Consider the context of the free distributive lattice FCD(3) with three generators x, y, z:

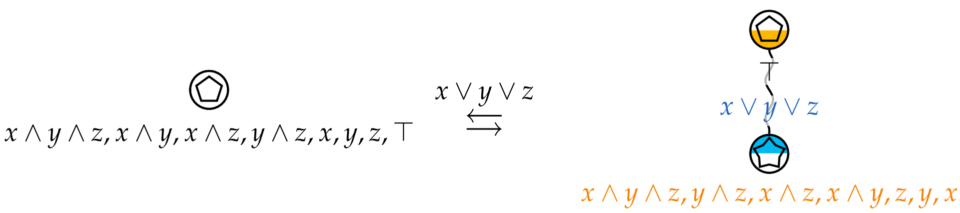
	$x \vee y \vee z$	$x \vee y$	$x \vee z$	$y \vee z$	x	y	z	$\top$
$x \wedge y \wedge z$	×	×	×	×	×	×	×	
$y \wedge z$	×	×	×	×		×	×	
$x \wedge z$	×	×	×	×	×		×	
$x \wedge y$	×	×	×	×	×	×		
z	×		×	×			×	
y	×	×		×		×		
x	×	×	×		×			
$\top$								

With the preceding algorithm its concept lattice is constructed stepwise. At first we add all objects to an initially empty context. (As we do not have an update algorithm for new objects at the moment. This is a future task and can be derived from the update algorithm for new attributes due to the duality of objects and attributes in concept lattices.) The resulting context diagram only has one node:

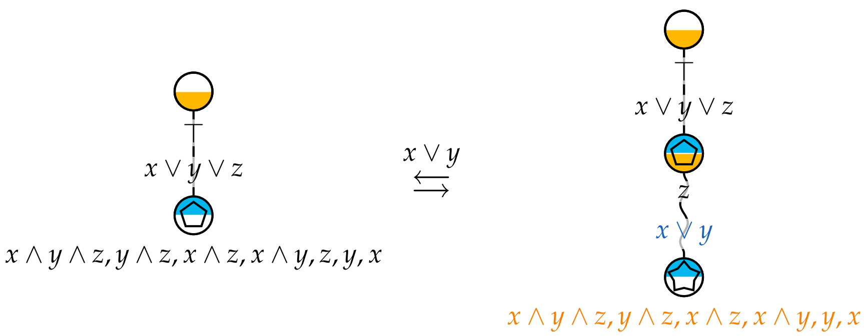


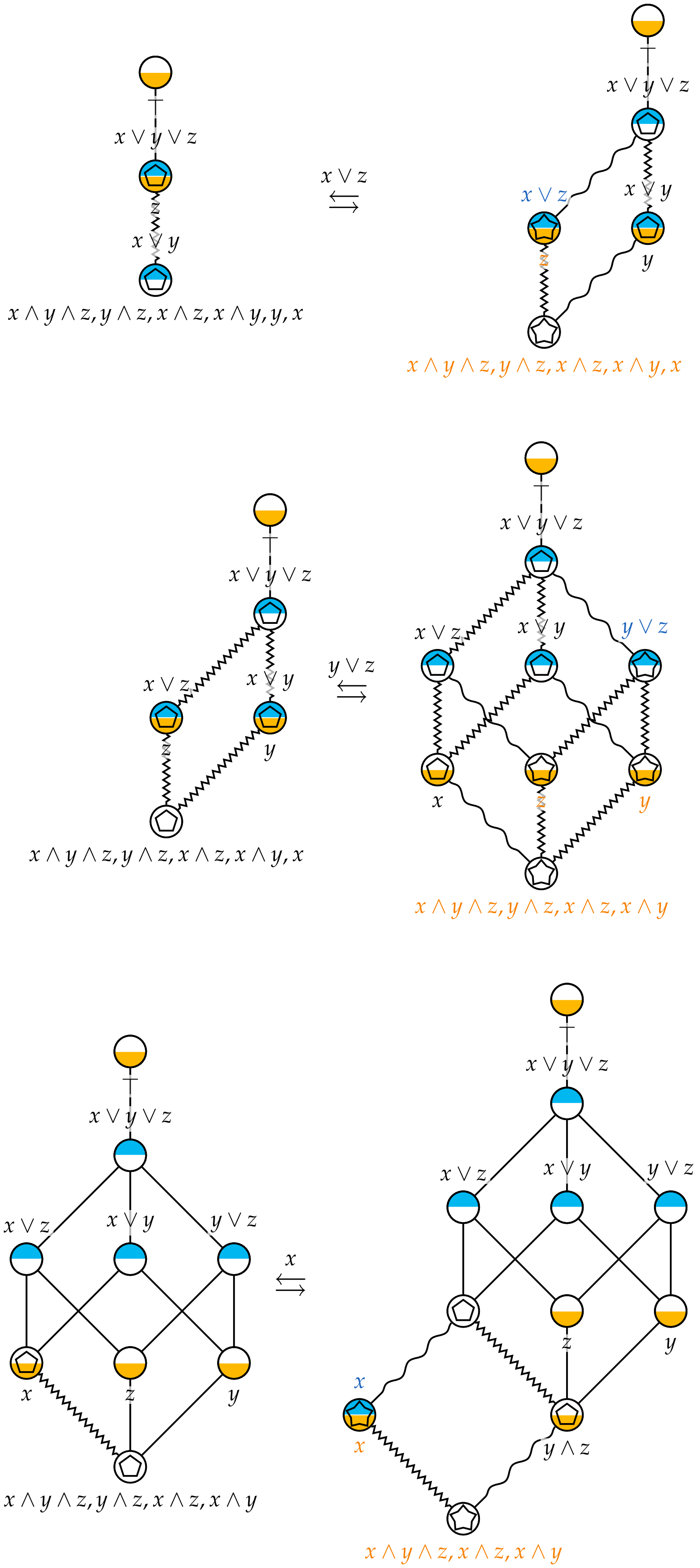
In the ongoing text the attributes are added to the context one after another. Please remark that generating nodes are always tagged by a pentagon , modified nodes are marked with a cloud , new nodes are highlighted with a star and old non-generator nodes are not tagged . Furthermore objects which change their positions are colored in red and analogously new or moved attributes are hued with blue.

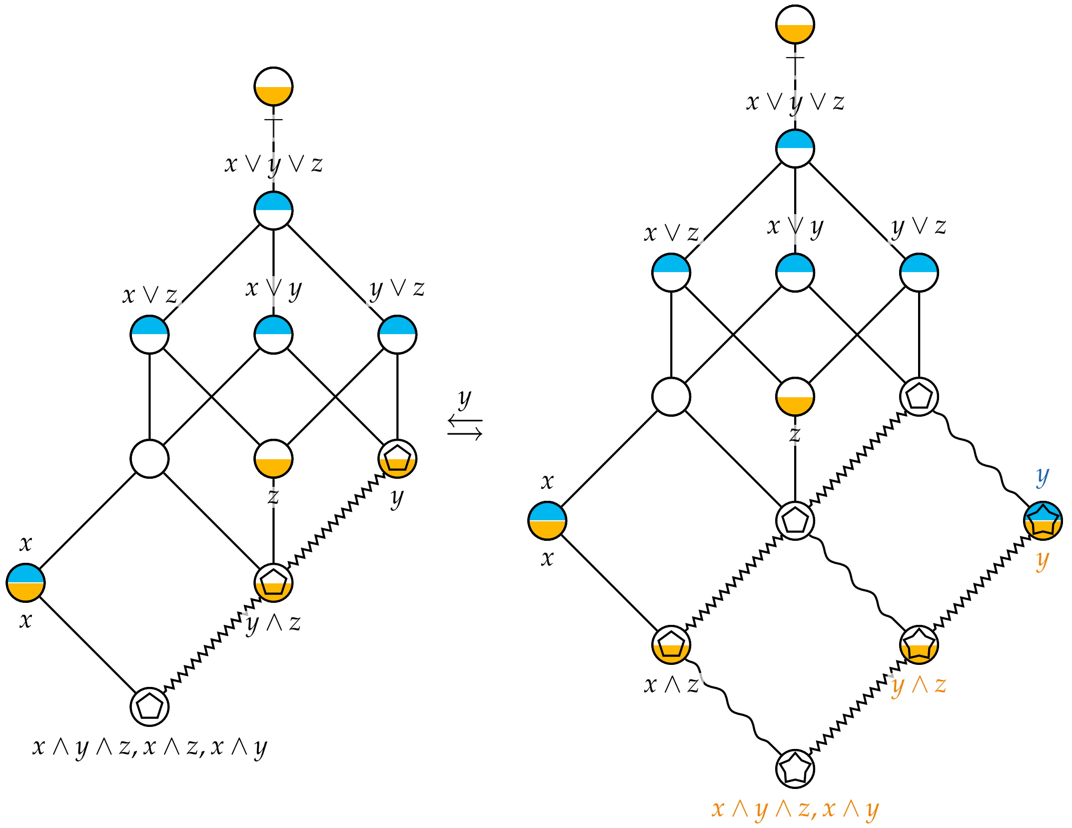
Now the first attribute $x \vee y \vee z$ is added to the context. Its extent consists of the objects $x \wedge y \wedge z, x \wedge y, x \wedge z, y \wedge z, x, y$ and z. As the single node contains all objects, and the new attribute extent not, it must be an old node. Furthermore as actually there are no attributes in the context the intent is empty, so the closure of the intersection of its extent and the new attribute extent must also be empty. This means they are equal and so the single node is a generator.



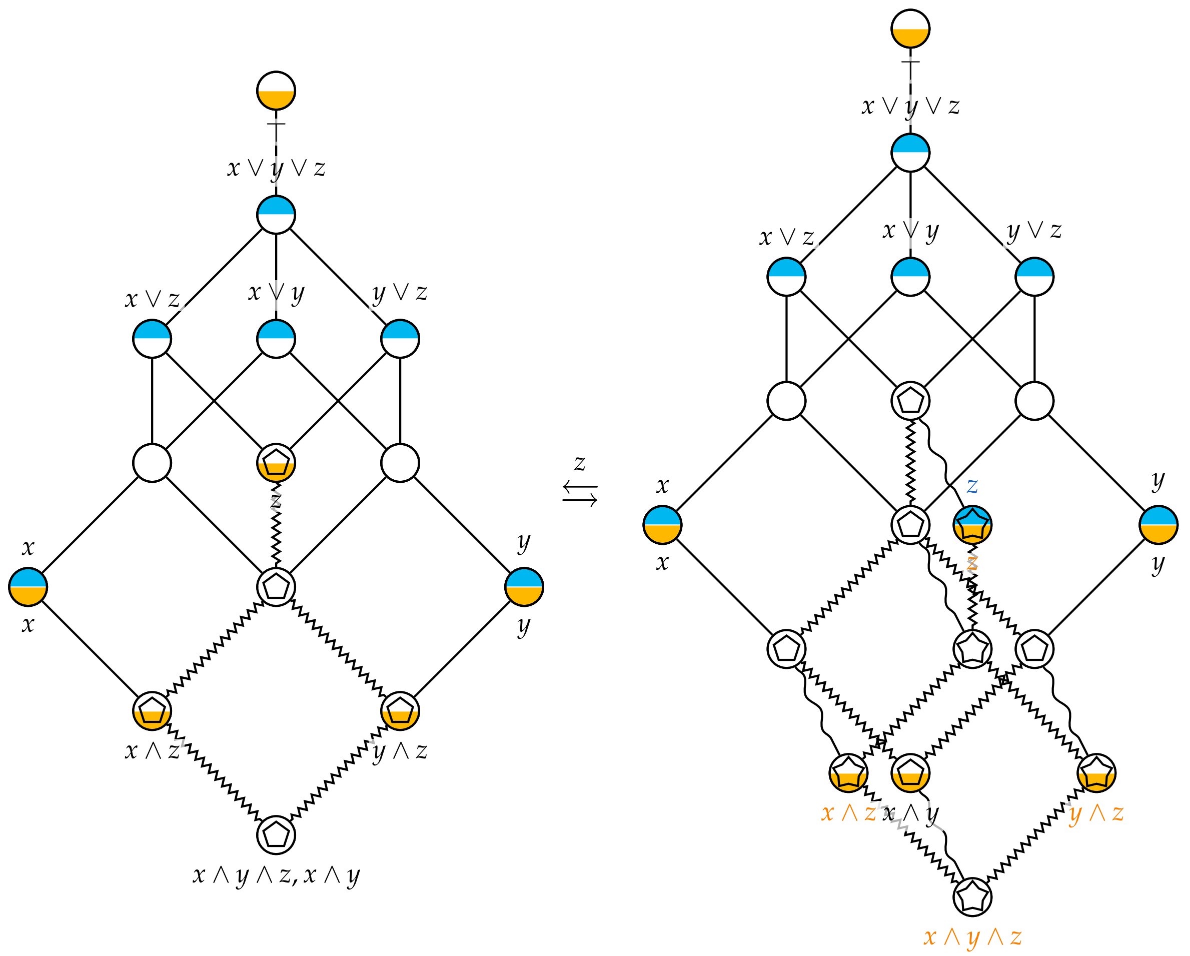
Next, the attribute $x \vee y$ is added.



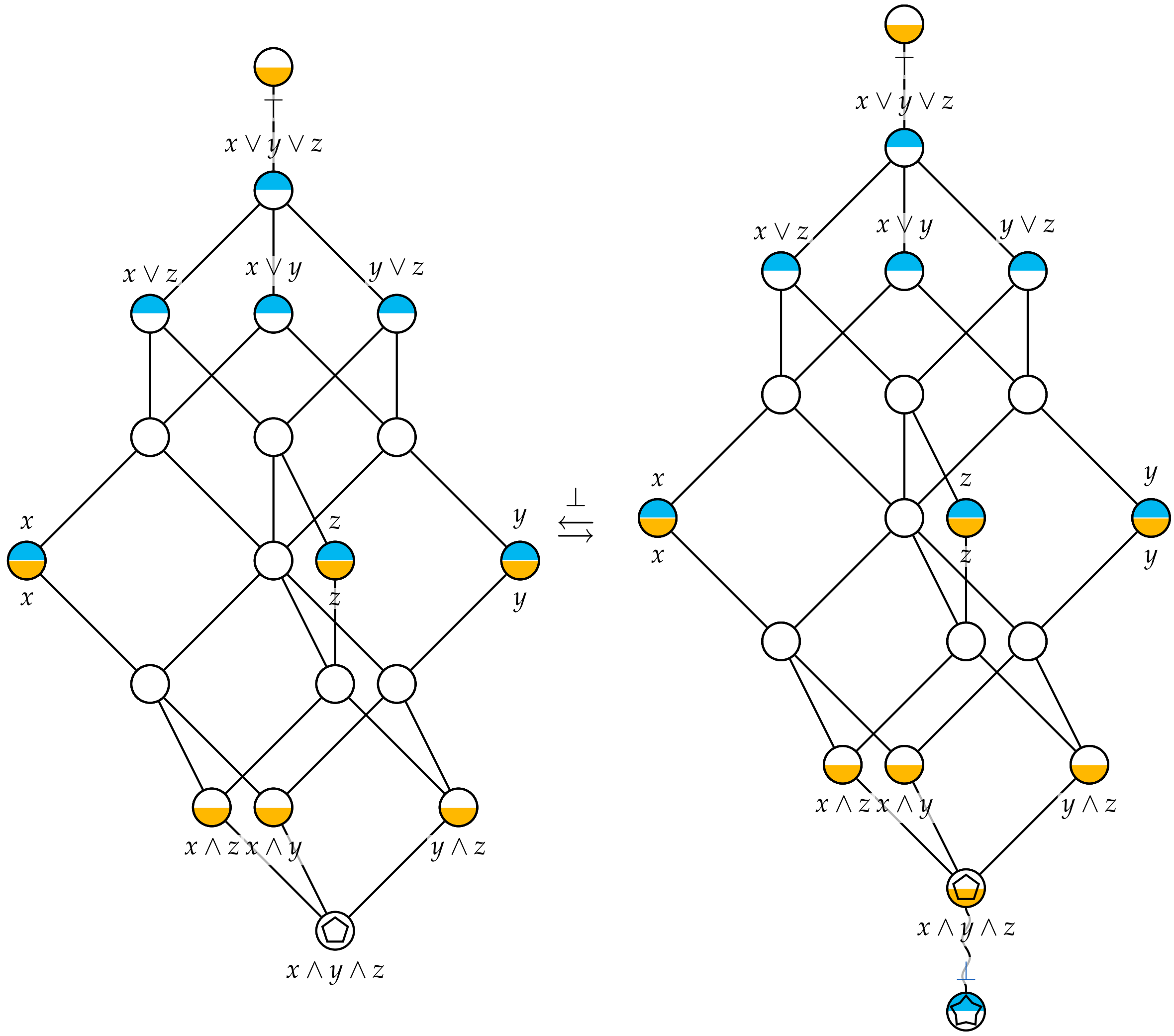

$x \vee y \vee z$
z
$x \vee y$
$x \wedge y \wedge z, y \wedge z, x \wedge z, x \wedge y, y, x$
$x \vee z$
$x \vee y \vee z$
$x \vee z$
$x \vee y$
y
$x \wedge y \wedge z, y \wedge z, x \wedge z, x \wedge y, x$
$x \vee y \vee z$
$x \vee z$
$x \vee y$
y
$x \wedge y \wedge z, y \wedge z, x \wedge z, x \wedge y, x$
$y \vee z$
$x \vee y \vee z$
$x \vee z$
$x \vee y$
$y \vee z$
x
z
y
$x \wedge y \wedge z, y \wedge z, x \wedge z, x \wedge y$
$x \vee y \vee z$
$x \vee z$
$x \vee y$
$y \vee z$
x
z
y
$x \wedge y \wedge z, y \wedge z, x \wedge z, x \wedge y$
x
$x \vee y \vee z$
$x \vee z$
$x \vee y$
$y \vee z$
z
y
x
x
$y \wedge z$
$x \wedge y \wedge z, x \wedge z, x \wedge y$



$x \vee y \vee z$
$x \vee z$
$x \vee y$
$y \vee z$
z
y
x
x
$y \wedge z$
$x \wedge y \wedge z, x \wedge z, x \wedge y$
$\overset{y}{\leftrightarrows}$
$x \vee y \vee z$
$x \vee z$
$x \vee y$
$y \vee z$
z
x
y
x
y
$x \wedge z$
$y \wedge z$
$x \wedge y \wedge z, x \wedge y$



$x \vee y \vee z$
$x \vee z$
$x \vee y$
$y \vee z$
z
x
y
x
y
$x \wedge z$
$y \wedge z$
$x \wedge y \wedge z, x \wedge y$
$\overset{z}{\leftrightarrows}$
$x \vee y \vee z$
$x \vee z$
$x \vee y$
$y \vee z$
x
z
y
x
z
y
$x \wedge z$
$x \wedge y$
$y \wedge z$
$x \wedge y \wedge z$



3.2 Setting & Deleting a single cross

Another question that occurs when working with a formal context is how the concept lattice and the corresponding line diagram change when a single cross is added or removed to its incidence. In other words, how can a line diagram of $\mathfrak{B}(G, M, I \dot\cup \{(g, m)\})$ be computed from an existing line diagram of $\mathfrak{B}(\mathbf{K})$? Anyway, the intents of $(G, M, I \dot\cup \{(g, m)\})$ are always of the form A^I or $A^I \dot\cup \{m\}$ for a suitable object set $A \subseteq G$. This can be seen as follows:

$$\begin{aligned} A^{I \dot\cup \{(g,m)\}} &= \left\{ n \in M \,\middle|\, \underset{h \in A}{\forall}\, hIn \text{ or } h\,\{(g,m)\}\,n \right\} \\ &= (A \setminus \{g\})^I \cap (\{g\}^I \cup \{m\}) \\ &= A^I \cup ((A \setminus \{g\})^I \cap \{m\}) \\ &= \begin{cases} A^I \dot\cup \{m\} & \text{if } m \in (A \setminus \{g\})^I \text{ and } m \notin A^I \\ A^I & \text{if } m \notin (A \setminus \{g\})^I \text{ or } m \in A^I \end{cases} \end{aligned}$$

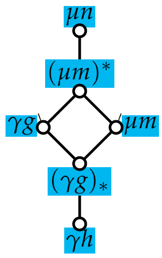
holds for every object set $A \subseteq G$. Dually each extent of $(G, M, I \dot\cup \{(g, m)\})$ has the form B^I or $B^I \dot\cup \{g\}$ for a particular attribute set $B \subseteq M$. This is due to

$$B^{I \dot\cup \{(g,m)\}} = \begin{cases} B^I \dot\cup \{g\} & \text{if } g \in (B \setminus \{m\})^I \text{ and } g \notin B^I \\ B^I & \text{if } g \notin (B \setminus \{m\})^I \text{ or } g \in B^I \end{cases}$$

for arbitrary attribute sets $B \subseteq M$. In (GW99) there is a first clue on page 128: The number of concepts can increase or decrease; an estimate by SKORSKY states

$$\tfrac{1}{2} \cdot |\mathfrak{B}(\mathbf{K})| \leq |\mathfrak{B}(G, M, I \cup \{(g, m)\})| \leq \tfrac{3}{2} \cdot |\mathfrak{B}(\mathbf{K})|.$$

The concept lattice grows when neither $g \swarrow m$ nor $g \nearrow m$ hold, as then I is a closed subrelation of $I \cup \{(g,m)\}$ and $\mathfrak{B}(\mathbf{K})$ is thus a complete sublattice of $\mathfrak{B}(G, M, I \cup \{(g,m)\})$. So one just must determine the concepts of $(G, M, I \dot\cup \{(g,m)\})$ that are not already concepts of $\mathbf{K}$. This can be done by computing all intents of $(G, M, I \dot\cup \{(g,m)\})$ which have the form $A^I \dot\cup \{m\}$.
The concept lattice shrinks when both $g \swarrow m$ and $g \nearrow m$ hold. Then g does not have m, but g is contained in the extent of every proper superconcept of μm and dually m is contained in the intent of every proper subconcept of γg. In other words, the object concept γg is no sub concept of the attribute concept μm and for all concepts $(A, B) \in \mathfrak{B}(\mathbf{K})$ with $\mu m < (A, B) < \gamma g$ it holds that $\gamma g \leq (A, B) \leq \mu m$. Clearly this is a contradiction to $\gamma g \not\leq \mu m$ and thereby either μm must be a lower neighbor of γg or μm cannot be a subconcept of γg. So either γg covers μm or both concepts are uncomparable. If they are neighboring then they are simply merged in the transition from $\mathfrak{B}(\mathbf{K})$ to $\mathfrak{B}(G, M, I \dot\cup \{(g,m)\})$. This can be seen as follows: From $(m^I, m^{II}) \prec (g^{II}, g^I)$ it follows that $g^{II} \cap m^I = m^I$. Furthermore



$$m^{I\dot\cup\{(g,m)\}} = m^I \dot\cup \{g\} = (g^{II} \cap m^I) \dot\cup \{g\} = g^{(I\dot\cup\{(g,m)\})(I\dot\cup\{(g,m)\})}$$

and so both the object concept of g and the attribute concept of m equal in $(G, M, I \dot\cup \{(g,m)\})$. No other concepts are affected.
The other cases remain for future work. For practical applications when visual animated transitions are desired one can use the complete algorithm from the preceding section to firstly remove the concerning attribute column, secondly modify it and thirdly add it again to the underlying formal context. Also one could calculate a transition algorithm by means of composing the preceding algorithm.

4 Iterative Exploration of Concept Lattices

The common way of visualizing a formal context is done by computing all formal concepts, constructing the neighborhood relation of the formal concepts and then layouting the given structures. Unfortunately this leads to hardly readable diagrams for datasets of at least a medium size (more than 50 formal concepts). Thus a technique (or better many techniques) for a interactive exploration of a formal concept lattice is neccessary.

4.1 Iceberg Lattices

A simple way to achieve an interactive navigation through a formal concept lattice could be done by displaying the top-most formal concept (G, G^I) and then enable the user to show the lower neighbors of a formal concept by clicking on a already displayed formal concept. As a consequence the user firstly sees the largest formal concept, i.e. the formal concept with all objects from the context and then the user can go down in the concept lattice to see smaller formal concepts with less objects but more attributes involved. Such a technique already exists and is called *Iceberg-Lattices*, please see (Stu02).

4.2 Alpha Iceberg Lattices

There is a way to generalize these Iceberg-Lattices to have not only a down-oriented view from the top-most concept and its lower neighbors, but also from intermediate formal concepts that rise from a given classification of the objects into groups. Initially each group induces a formal concept node in the line diagram and then the user can go down from these *grouped* formal concepts. This can be seen as a multiple Iceberg-view: from each grouped formal concept only some of child formal concepts are displayed, that fulfill certain criteria. See a formalization in the appendix or in (VS05).

4.3 Partly selections

Another well know technique for a better readable view on a concept lattice of a context (G, M, I) is *Nesting*. This is done by partitioning the attribute set into two classes M_1 and M_2, the first one contains the attributes for the *outer diagram* and the second one holds the attributes for the *inner diagram*. The two resulting subcontexts $(G, M_1, I \cap G \times M_1)$ and $(G, M_2, I \cap G \times M_2)$ are then used to construct the nested diagram in the following way: Initially compute and layout the outer concept lattice $\underline{\mathfrak{B}}(G, M_1, I \cap G \times M_1)$ and the inner concept lattice $\underline{\mathfrak{B}}(G, M_2, I \cap G \times M_2)$. Then draw the outer concept lattice as a directed graph and nest the inner concept lattice in each node of the outer diagram. Some pairs of an outer concept node and an inner concept nodes then describe formal concepts of the whole context (G, M, I), namely those outer concepts (A_1, B_1) and inner concepts (A_2, B_2) for which $(A_1 \cap A_2, B_1 \cup B_2)$ is a formal concept of (G, M, I). Thus the inner nodes (A_2, B_2) within an outer node (A_1, B_1) are so called *realized* inner concepts, iff $(A_1 \cap A_2, B_1 \cup B_2)$ is a formal concept of (G, M, I). As a result the number of edges is reduced and thus can lead to a better readability of the concept lattice diagram. A disadvantage of the *Nesting* approach is the fact that the user can hardly gain comparing information about objects that are in different outer nodes. So why not glue some interesting inner nodes to the outer nodes?
Suppose a context (G, M, I) is given and the user initially selects some attributes $M_0 \subseteq M$ for a first view on the context data. When the user sees some interesting objects $G_1 \subseteq A_1 \subseteq G$ that label a concept node (A_0, B_0) of $\underline{\mathfrak{B}}(G, M_0, I \cap G \times M_0)$, and he wants to have further information on that objects by involving some of the remaining attributes $M_1 \subseteq M \setminus M_0$, then he can click on the concept node (A_0, B_0) that is labeled with the

interesting objects G_1 and then the initial concept lattice is expanded below that concept node by glueing another concept lattice $\mathfrak{B}(G_1, M_1, I \cap G_1 \times M_1)$ below. Please have a look at the next section 4.3.1 for a sophisticated example. Then the user can decide to retain the newly attached concept nodes in the diagram or he can decide to remove these attached nodes if the shown information is not helpful and to hold the diagram complexity as low as possible. In the ongoing exploration of the concept lattice the user can select other interesting objects $G_2 \subseteq G$ by clicking other nodes and choosing appropriate attributes $M_2 \subseteq M \setminus M_0$. If there are some other attached nodes in the diagram already for the selections $\{(G_i, M_i)\}_{i=1}^{k}$ he can decide to merge the attached nodes, if they share selected attributes. This is done by computing the formal concept lattice of a subincidence of (G, M, I), namely $(\bigcup_{i=1}^{k+1} G_i, \bigcup_{i=1}^{k+1} M_i, \bigcup_{i=1}^{k+1} (I \cap G_i \times M_i))$ and glueing the appropriate concept lattice on the main concept lattice $\mathfrak{B}(G, M_0, I \cap G \times M_0)$.
This technique can be used to search for an unknown object within a context by an iterative exploration of the concept lattice by expanding nodes with interesting objects with new attributes. Take as an example a formal context about cars. Various cars are chosen as objects and appropriate attributes describe the cars, e.g. firstly show the concept lattice for attributes describing the amount of the buying price. Then athe user can select a node with affordable cars and can further structure the cars in that node by other attributes, e.g. maintainance costs. If he has found a good selection then he goes to a car dealer and looks for that car. If he could not make a decision yet, the remaining cars in a node can be further zoomed in by involving additional attributes like number of doors or size of the trunk etc. This expansion is repeated until a selection of good size is reached.

4.3.1 Example with EMAGE data

From the EMAGE data set a formal context was extracted. Its objects are various EMAP identifiers (acronyms for tissues of a mouse embryo at specific theiler stages) and its attributes are the BMP genes *bmp2*, *bmp3*, *bmp4*, *bmp5*, *bmp6*, *bmp7* and *bmp10* (*bmp* abbreviates *bone morphogenetic protein*). The resulting (reduced) context and its (full) concept lattice is shown in figure 4.1. See also (AM11) for details on the EMAGE data. In the nesting

	bmp2	bmp3	bmp4	bmp5	bmp6	bmp7	bmp10
emap167	×	⤢	×	×	⤢	×	⤢
emap234	×	↗	⤢	×	↗	⤢	↗
emap267	×	↗	⤢	⤢	↗	×	↗
emap315	⤢	↗	⤢	×	↗	×	↗
emap1259	↗	↗	×	↗	×	⤢	↗
emap1685	⤢	⤢	×	⤢	×	×	⤢
emap2322	↗	↗	⤢	↗	×	↗	↗
emap2454	⤢	⤢	⤢	⤢	⤢	⤢	×
emap7503	×	×	×	⤢	⤢	⤢	⤢
emap7843	⤢	×	⤢	×	⤢	⤢	⤢
emap8114	⤢	×	×	↗	↗	↗	↗
emap8226	×	×	⤢	↗	↗	↗	↗

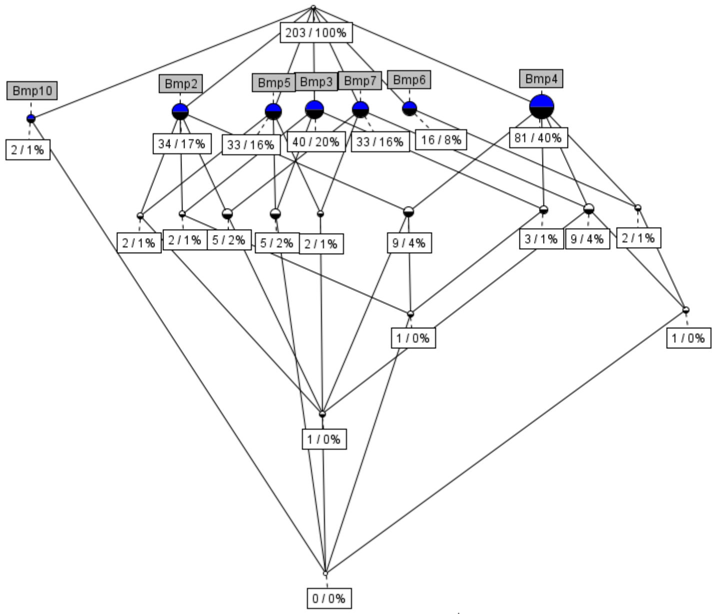


Figure 4.1: Formal Concept Lattice describing BMP genes occuring in tissues (EMAP ids)

approach the attribute set is partitioned into two subsets M_1 and M_2. We choose $M_1 := \{bmp2, bmp3, bmp7\}$ and $M_2 := \{bmp4, bmp5, bmp6, bmp10\}$. The resulting outer concept lattice $\mathfrak{B}(G, M_1, I \cap G \times M_1)$ is shown on the left in figure 4.2 and the appropriate inner concept lattice $\mathfrak{B}(G, M_2, I \cap G \times M_2)$ is displayed on the right side in figure 4.2. When some further information about the objects in the outer node labeled with attribute *bmp3* is required, the user can zoom into that node and see the inner diagram, that further structures the objects with the remaining objects from M_2. There are only four realized concepts within the inner diagram of the attribute concept node for *bmp3*: the top and bottom node and the attribute nodes for *bmp4* and *bmp5*.
The readabily is not very high with the nesting approach, especially when trying to compare objects from different outer nodes.

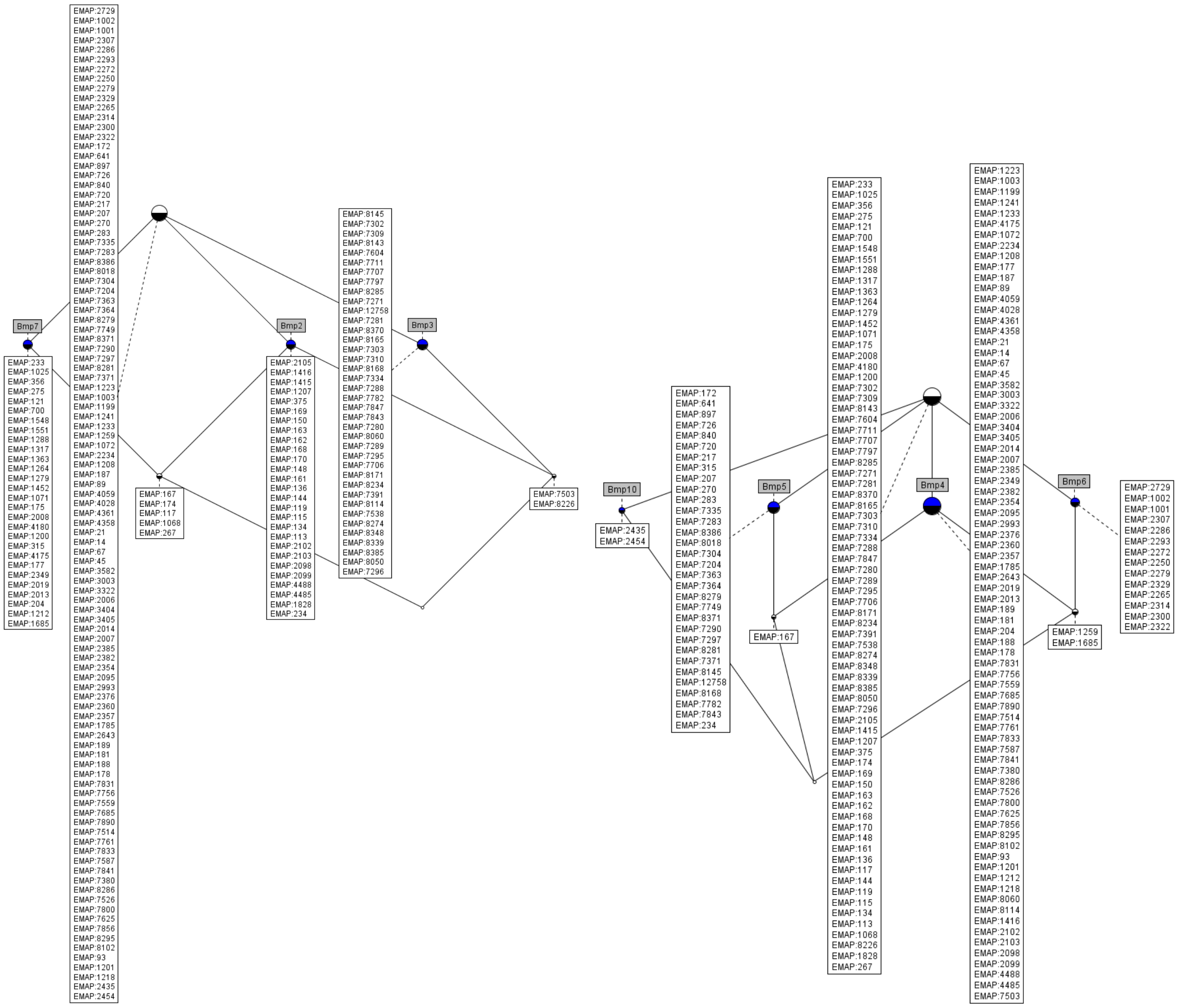


Figure 4.2: Outer and inner concept lattice for EMAGE context

Initially the user selects some attributes for a first view on the context data. For the selected subcontext the appropriate concept lattice is visualized, see figure 4.3. In the nesting approach this would be the outer concept lattice. To improve readability of the shown concept lattices, the bottom concept is hidden when it does not contain any information, i.e. it has no objects in its extent and is no attribute concept. Dually the top concept is hidden, if it has no attributes in its intent and is no object concept.

	bmp2	**bmp3**	bmp4	bmp5	bmp6	**bmp7**	bmp10
emap167	×		×	×		×	
emap234	×			×			
emap267	×					×	
emap315				×		×	
emap1259			×		×		
emap1685			×		×	×	
emap2322					×		
emap2454							×
emap7503	×	×	×				
emap7843		×		×			
emap8114		×	×				
emap8226	×	×					

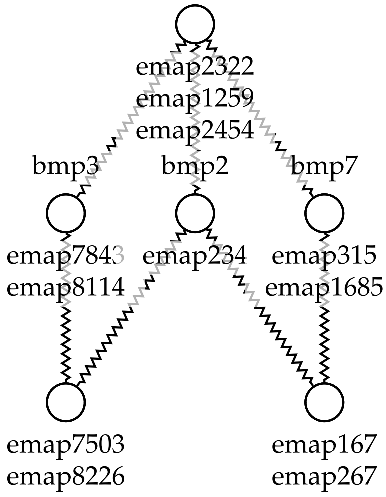


Figure 4.3: Initial view on the concept lattice diagram for attribute selection $M_0 := \{bmp2, bmp3, bmp7\}$

When looking at this initial view the user can see that the *bmp3* was detected in tissues *emap7843* and *emap8114*. (Of course *bmp3* was also detected in *emap7503* and *8226*.) Now the user is further interested in these two objects at the *bmp3* node and clicks on it. Then the remaining attributes *bmp4*, *bmp5*, *bmp6* and *bmp10* are used to give further information on these selected objects. This is done by glueing the concept lattice of the selection $(\{emap7843, emap8114\}, \{bmp4, bmp5, bmp6, bmp10\})$ below the selected *bmp3* node, as in figure 4.4.

	bmp2	bmp3	**bmp4**	**bmp5**	**bmp6**	bmp7	**bmp10**
emap167	×		×	×		×	
emap234	×			×			
emap267	×					×	
emap315				×		×	
emap1259			×		×		
emap1685			×		×	×	
emap2322					×		
emap2454							×
emap7503	×	×	×				
emap7843		×		×			
emap8114		×	×				
emap8226	×	×					

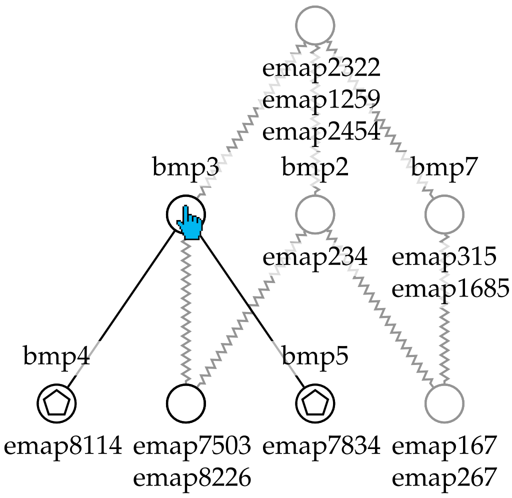


Figure 4.4: Expanded view for objects $emap7843$ and $emap8114$ for all remaining attributes $bmp4$, $bmp5$, $bmp6$ and $bmp10$.

Now the affixed concept lattice could either be removed or retained. The affixed concept lattice is equivalent to the (realized) inner lattice within a concept of the outer lattice in nested diagrams, but in this approach it is glued in the outer lattice and not displayed within a node of the outer lattice. So to continue this example, let the affixed concept lattice retain in the diagram. In the next step the user is interested in the objects *emap7503* and *emap8226* from the concept below, so the user clicks on the corresponding node. This again results in a further concept lattice that is glued below the selected node, as shown in figure 4.5.
The actual diagram state in figure 4.5 contains the main concept lattice and two affixed concept lattices for

	bmp2	bmp3	**bmp4**	**bmp5**	**bmp6**	bmp7	**bmp10**
emap167	×		×	×		×	
emap234	×			×			
emap267	×					×	
emap315				×		×	
emap1259			×		×		
emap1685			×		×	×	
emap2322					×		
emap2454							×
emap7503	×	×	×				
emap7843		×		×			
emap8114		×	×				
emap8226	×	×					

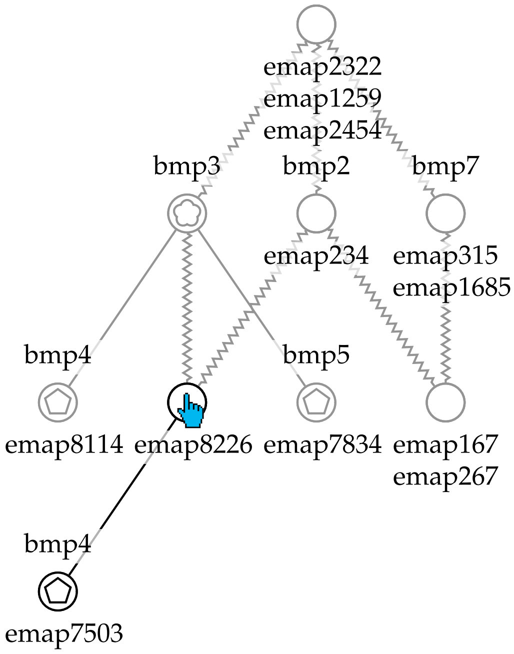


Figure 4.5: Expanded view for objects $emap8226$ and $emap7503$ structured with all remaining attributes $bmp4$, $bmp5$, $bmp6$ and $bmp10$. The preceding expansion from figure 4.4 retains.

some objects. The user could now decide to merge the affixed concept lattices, if the shown pieces of information are useful for his or her needs. Please have look at the next figure 4.6 to see the results.

	bmp2	bmp3	**bmp4**	**bmp5**	**bmp6**	bmp7	**bmp10**
emap167	×		×	×		×	
emap234	×			×			
emap267	×					×	
emap315				×		×	
emap1259			×		×		
emap1685			×		×	×	
emap2322					×		
emap2454							×
emap7503	×	×	×				
emap7843		×		×			
emap8114		×	×				
emap8226	×	×					

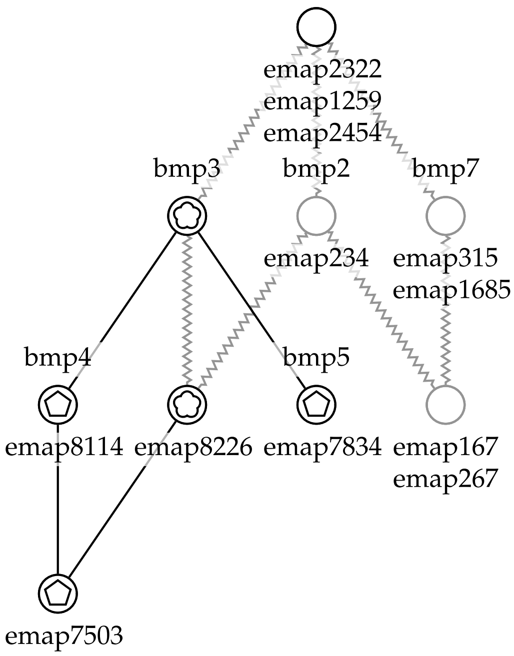


Figure 4.6: Merge of the two preceding expansions from figures 4.4 and 4.5.

Continuing the workflow of clicking, expand and merging, the following diagram states arise. Clicked nodes are always marked with a hand, already expanded nodes of the main lattice with a cloud, and affixed nodes with a pentagon.

	bmp2	bmp3	**bmp4**	**bmp5**	**bmp6**	bmp7	**bmp10**
emap167	×		×	×		×	
emap234	×			×			
emap267	×					×	
emap315				×		×	
emap1259			×		×		
emap1685			×		×	×	
emap2322					×		
emap2454							×
emap7503	×	×	×				
emap7843		×		×			
emap8114		×	×				
emap8226	×	×					

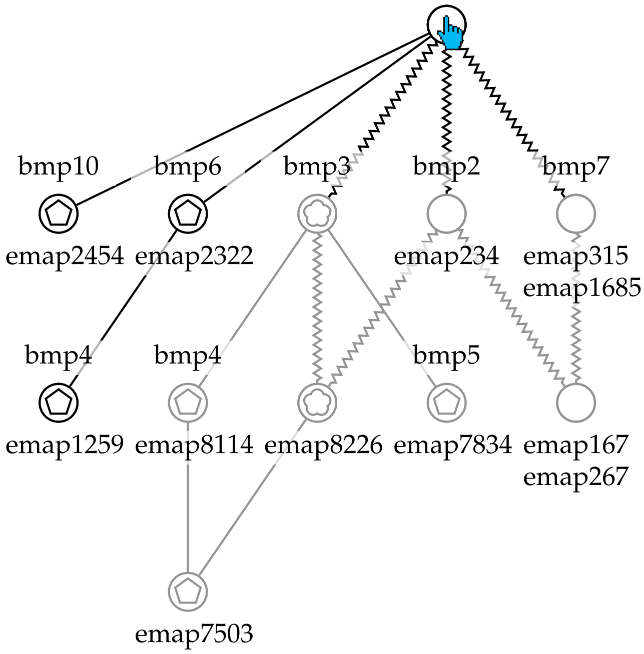


Figure 4.7: Expansion of the top concept node: objects *emap*1259, *emap*2322 and *emap*2454 are further structured using all remaining attributes *bmp*4, *bmp*5, *bmp*6 and *bmp*10. As then the top concept node is no longer labeled by any objects, it is not drawn.

	bmp2	bmp3	**bmp4**	**bmp5**	**bmp6**	bmp7	**bmp10**
emap167	×		×	×		×	
emap234	×			×			
emap267	×					×	
emap315				×		×	
emap1259			×		×		
emap1685			×		×	×	
emap2322					×		
emap2454							×
emap7503	×	×	×				
emap7843		×		×			
emap8114		×	×				
emap8226	×	×					

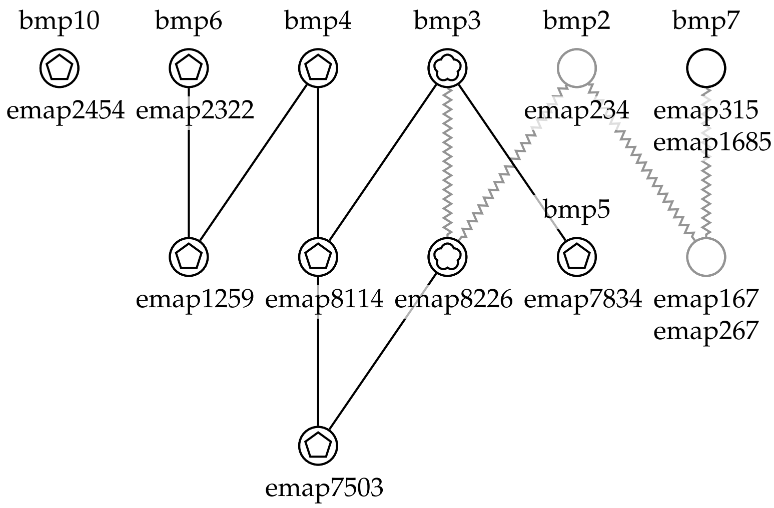


Figure 4.8: Merge of all preceding expansions. Top concept node contains no information and is omitted.

	bmp2	bmp3	**bmp4**	**bmp5**	**bmp6**	bmp7	**bmp10**
emap167	×		×	×		×	
emap234	×			×			
emap267	×					×	
emap315				×		×	
emap1259			×		×		
emap1685			×		×	×	
emap2322					×		
emap2454							×
emap7503	×	×	×				
emap7843		×		×			
emap8114		×	×				
emap8226	×	×					

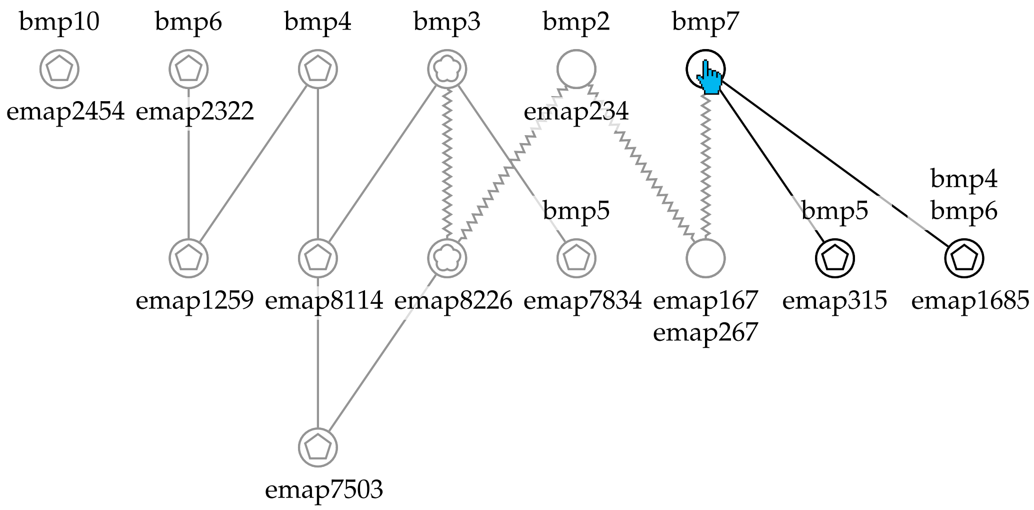


Figure 4.9: Expansion of the *bmp7* node with all left attributes from $M \setminus M_0$.

	bmp2	bmp3	**bmp4**	**bmp5**	**bmp6**	bmp7	**bmp10**
emap167	×		×	×		×	
emap234	×			×			
emap267	×					×	
emap315				×		×	
emap1259			×		×		
emap1685			×		×	×	
emap2322					×		
emap2454							×
emap7503	×	×	×				
emap7843		×		×			
emap8114		×	×				
emap8226	×	×					

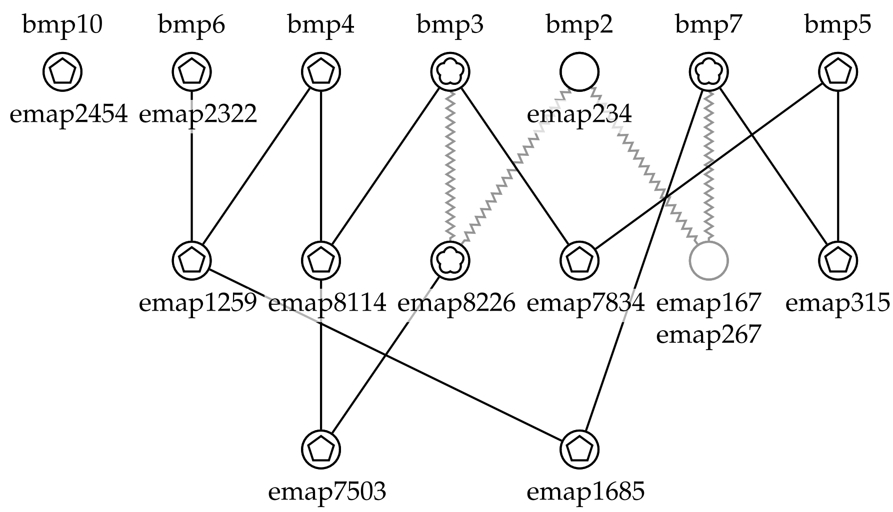


Figure 4.10: Merge of all preceding expansions.

4.4 Overview on Pruning & Interaction Techniques

There are various strategies for pruning a formal concept lattice, to gain a clearer structure of the conceptual data or to emphasize on interesting parts. The figure 4.11 gives an overview.

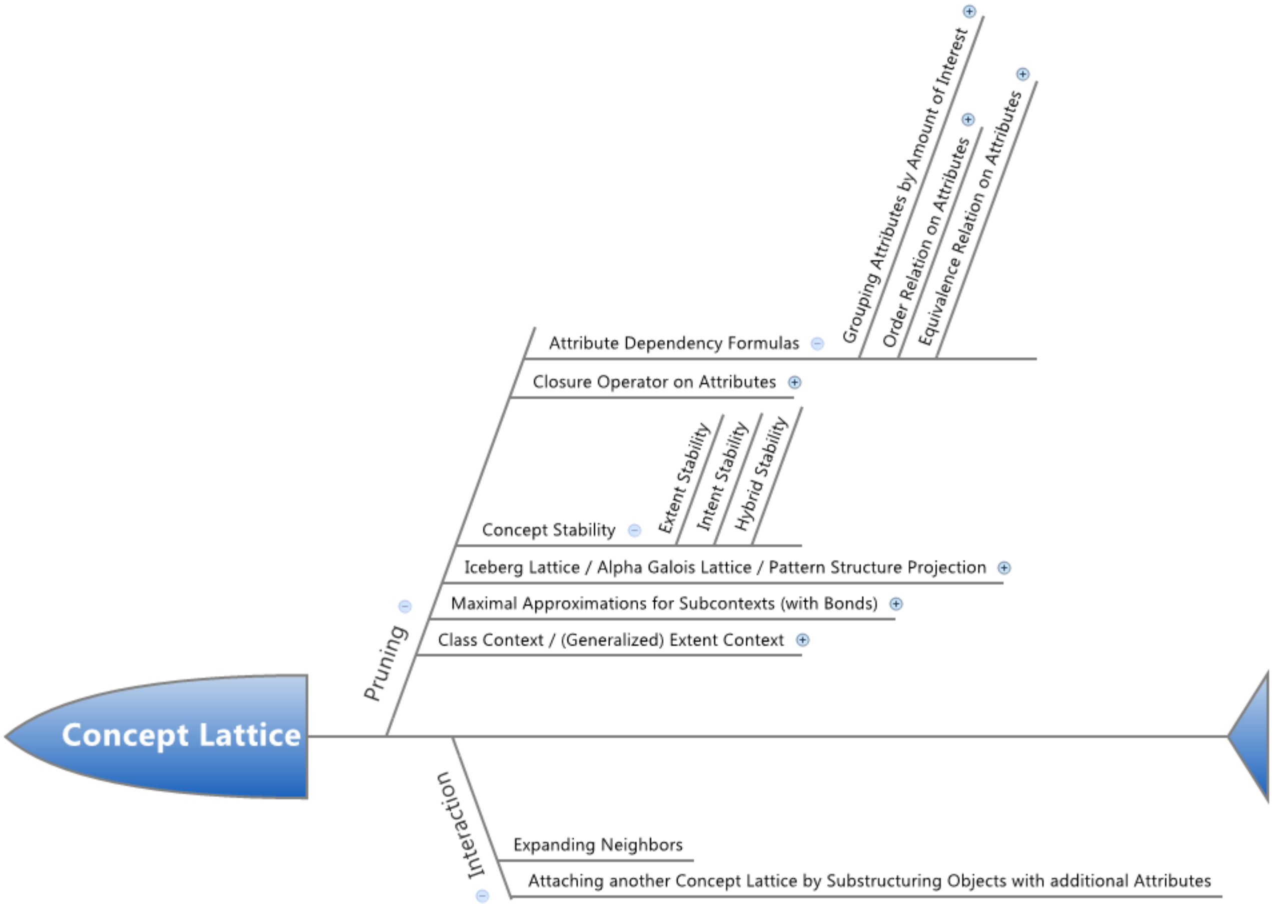


Figure 4.11: Overview on various pruning and interaction techniques on formal concept lattices

Part II

Implementation Details

5 Requirement Analysis

5.1 Introduction

In this document we analyze requirements for general graph visualizations. Then we particularize and constrain them to lattices as a special form of graphs. The requirements are classified according to different kinds, e.g. human interaction, visual and technical details.
The second section introduces and describes the visualization requirements for graphs on the user-level and the third section presents the requirements on the low-level. In the fourth section these user-level and low-level requirements will be mapped to each other and then analyzed with FCA methods. The last section lists special requirements on lattices.
First of all, we want to give a quote by E. H. Gombrich (Gom77) which describes the intent of visualization in an abstract way:

> *Everything points to the conclusion that the phrase 'the language of art' is more than a loose metaphor, that even to describe the visible world in images we need a developed system of schemata.*

There are two refinements by B. SHNEIDERMAN and D. KEIM for visualizing large amounts of data:

- **Visual Information Seeking Mantra** (Shn96)

  Overview first, zoom and filter, then details on demand (in a loop)

- **Visual Analytics Mantra** (Kei05)

  Analyse first, show the important, zoom, filter and analyse further, details on demand

In general we can say, that visualizations must not be static, but have to interact dynamically with the user. This includes that the user must tell the visualization what data or details should be displayed and that the visualization reacts then. So there are a number of requirements to visualization frameworks for graphs, and also for lattices as a special kind of graphs.
Within this document an ontology is introduced. Figure 5.1 shows the terminological box with the modelled classes and their connecting relations. The next two sections describe the instances of User-Level-Requirements and Low-Level-Requirements respectively.

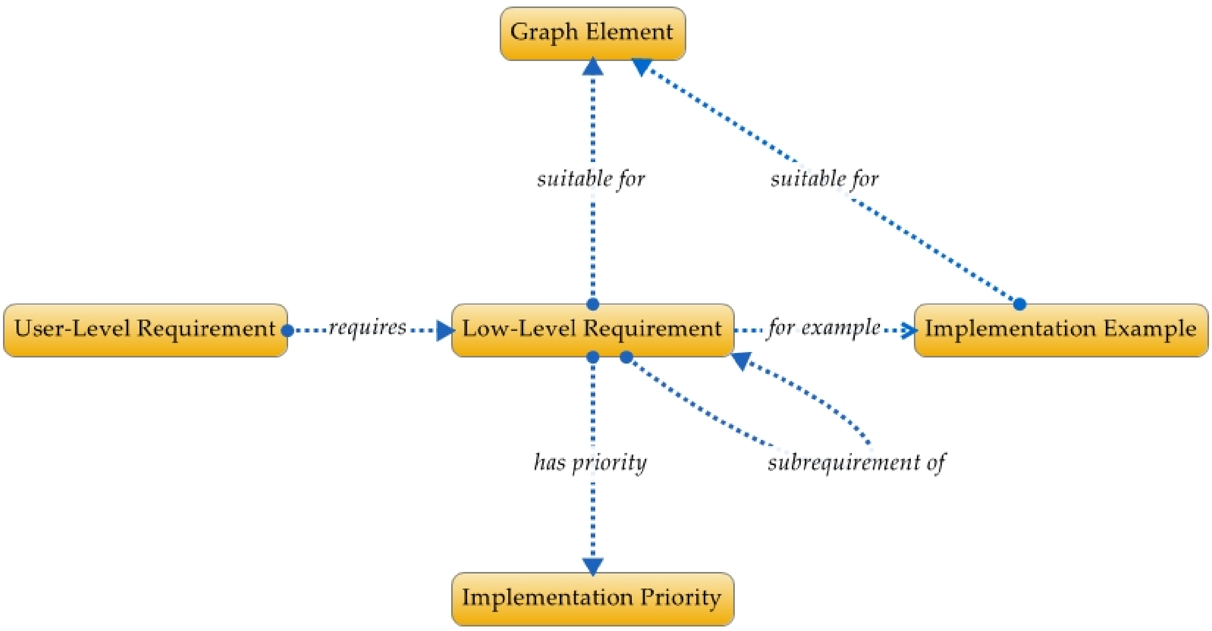


Figure 5.1: TBox of Requirements Ontology

5.2 User-Level Requirements for Graphs

From (Dau11), (YKSJ07) and (Cui) we get a general overview of human interaction techniques available for arbitrary visualizations and especially for graphs:

- **Select:** mark something as interesting
- **Explore:** show me something else
- **Reconfigure:** show me a different arrangement
- **Encode:** show me a different representation
- **Abstract/Elaborate:** show me more or less details
- **Filter:** show me something conditionally

- **Connect:** show me related items
- **Animate:** show transitions

In the ongoing section we explain each of the mentioned interaction techniques and in the next section the necessary low-level requirements to a graph visualization framework get summed up.

5.2.1 Select

A user selects by marking items that he is interested in. So when speaking about graphs, nodes and egdes should have different colors, shapes, size, transparency. A selection must be possible for a number of items, e.g. marking three nodes by hand, or marking one node by hand and adjacent nodes and edges automatically. Please note, that the Select requirement differs from the Filter requirement, as a Select just emphasizes some graph elements which the user is interested in, and a Filter would just retain some elements of interest while removing the remaining element.

> *Select interaction techniques provide users with the ability to mark (a) data item(s) of interest to keep track of it. When too many data items are presented on a view, or when representations are changed, it is difficult for users to follow items of interest. By making items of interest visually distinctive, users can easily keep track of them even in a large data set and/or with changes in representations. [...] Interestingly, Select interaction techniques seem to work as a preceding action to subsequent operations.* (YKSJ07)

5.2.2 Explore

A user wants to gain insight in the data displayed, so he wants the visualization to show different data that corresponds in some extent to the actual data displayed. This is called Exploration. So to enable the user to explore the data some low-level techniques are needed. The user must tell the visualization what he wants to do next and therefore context menus, popup menus, hovers or simply the ability to click on items of interest are needed. The exploration can be seen as a sub-technique of selection, or as a compound technique of user selection and visualization reaction.
When speaking about concept lattices, an Exploration can be realized by clicking on a lattice node (formal concept) and brushing the principal ideal and filter of this element, i.e. marking all elements above (being superconcepts or generalizations) and all elements below (being subconcepts or specializations).

> *Explore interaction techniques enable users to examine a different subset of data cases. When users view data using an Infovis system, they often can only see a limited number of data items at a time [...] Infovis system users typically examine a subset of the data to gain understanding and insight, and then they move on to view some other data. Explore interactions do not necessarily make complete changes in the data being viewed, however. More frequently, some new data items enter the view as others are removed.* (YKSJ07)

5.2.3 Reconfigure

A reconfiguration of a visualization is not a change of the visualization type, but a rearrangement of the items displayed. In a line diagram users may want to exchange the axes, or in a graph diagram users may wish to shift nodes for a better view on the graph or to emphasize a particular node.
So when speaking about graphs, it must be possible to shift nodes, to zoom out for a better overview, to zoom in for more details or to shift the panel to see different areas of the graph (panning).

> *Reconfigure interaction techniques provide users with different perspectives onto the data set by changing the spatial arrangement of representations. One of the essential purposes of Infovis is to reveal hidden characteristics of data and the relationships between them. A good static representation often serves this purpose, but a single representation rarely provides sufficient perspectives. Thus, many Infovis tools incorporate Reconfigure interaction techniques that allow users to change the way data items are arranged or the alignment of data items in order to provide different perspectives on the data set.* (YKSJ07)

5.2.4 Encode

Encoding enables the user to get insights in the data from different perspectives and is a technique quite similar to Reconfiguration, but it does not effect only some items of the visualization but all items by changing the visualization type. For example, switching from a pie chart to a line diagram, or changing the layout of a graph. So a visualization framework should have an option to switch the visualization type on the fly, without changing the presented data.
For graphs different layout algorithms must be available, e.g. force-driven, circular, hyperbolic, or special tree layouts, e.g. rooted tree, radial layout, balloon layout, treemap, sunburst diagram, cone tree, etc.
For an extension towards lattice visualizations more specific layout algorithms are needed, first of all the attribute-additive layout which produces a good readability, chain decomposition or also a derived force driven layout. Combinations of algorithms can also be useful, since all layout algorithms have some particular weaknesses.

> *Encode techniques enable users to alter the fundamental visual representation of the data including visual appearance (e.g., color, size, and shape) of each data element. [...] Simply changing how the data is represented (e.g., changing a pie chart to a histogram) is an example of Encode. By changing a type of representation, users expect to uncover new aspects of relationship. [...] Another widely used technique of Encode is the set of interaction techniques that alter the color encoding of a data set.* (YKSJ07)

5.2.5 Abstract/Elaborate

Abstraction and Elaboration is a kind of data zooming. The user may want to have some details of shown data (i.e. go to a deeper level, drill-down), or he may want to get a more generous representation (i.e. go to a higher level, drill-up). The drill-down can be realized in different ways, like splitting up data items into components, expanding adjacent nodes of a selected node in a graph, or just zoom in. On the other side, the drill-up can be summation of data items to a compound item in various ways, going to a common node of some selected nodes in a graph, or just zoom out.

> *Abstract/Elaborate interaction techniques provide users with the ability to adjust the level of abstraction of a data representation. These types of interactions allow users to alter the representation from an overview down to details of individual data cases and often many levels in-between. The user's intent correspondingly varies between seeking more of a broad, contextual view of the data to examining the individual attributes of a data case or cases. [...] An exemplary interaction technique in this category is any technique from the set of details-on-demand operations. For example, the drill-down operation in a treemap visualization. [...] Another very common but slightly complex example of Abstract/Elaborate techniques is zooming.* (YKSJ07)

5.2.6 Filter

Filtering is also closely related to Selection, but they differ in detail. The user want to focus on particular data items and restrict the visualization to some selected data. The restriction can be told to the framework by user selection or giving threshold values etc, and then can be displayed by hiding items, applying colors, transparency effects or resizing. This means that only the selected items remain visible and all others are hidden. This enables the user to focus further evaluation on some items of interest and to avoid overcrowding the visualization.
In the graph use case a user may select several nodes and the graph framework marks them red, let the other not selected nodes and adjacent edges fade away, or just make them transparent.

> *Filter interaction techniques enable users to change the set of data items being presented based on some specific conditions. In this type of interaction, users specify a range or condition, so that only data items meeting those criteria are presented. Data items outside of the range or not satisfying the condition are hidden from the display or shown differently, but the actual data usually remain unchanged so that whenever users reset the criteria, the hidden or differently shown data items can be recovered.* (YKSJ07)

5.2.7 Connect

The Connect requirement can be seen from at least two different perspectives. Firstly, imagine we have a graph with a node the user is interested in. For this node additional currently hidden information in form of adjacent nodes and edges can be displayed. The node gets connected with more data then. Secondly we could have two visualizations beneath each other and both visualizations contain information about the same set of entities (or at least share some entities). The user interaction in the one visualization (e.g. select, filter, abstract, elaborate, etc.) must be reflected in the other visualization as well. So the user can gain further information about one entity from two different visual views. This is also called *Brushing* or *Linking*.

> *Connect refers to interaction techniques that are used to (1) highlight associations and relationships between data items that are already represented and (2) show hidden data items that are relevant to a specified item.*
> *When multiple views are used to show different representations of the same data set [. . .] it may be difficult to identify the corresponding item for a data case in other view(s). To alleviate this difficulty, the brushing technique is used to highlight the representation of a selected data item in the other views being displayed. Connect interactions can apply to situations involving a single view as well. For example, in Vizster, hovering a mouse cursor over a node highlights directly connected nodes (friends) or neighbors of directly connected nodes (friends of friends).*
> *Connect interaction techniques also reveal related data items which are originally not shown. In Vizster, double clicking a node causes expansion of the node, so that the related nodes for the focus node (the person) are added.* (YKSJ07)

5.2.8 Animate

Animation is not a real interaction possibility, but a useful technique to visualize the transition between two states of the visualization when the user interacts with it. With animation the user gains insight about how the data items are transformed from one situation to another situation. Also, animations generate a new dimension to display data, namely the time dimension. Since animations consume time, they must not be too long. On the other hand they should not be too short, as then the user is not able to see it exactly. Animations should be connected with the other user interactions, like Explore, Reconfigure or Encode.

> *Animation is an unique advantage of computerized information visualization technique over other paper-based visualization techniques. It has become a very important feature in helping users understand the data sets, because it implicitly employs time as an extra dimension to facilitate data exploration. [. . .]*
> *Animation is not a standalone techniques. In fact, all the techniques described above can be combined with animation to improve their abilities. [. . .]*
> *Although animation is aesthetically good from a lot of points of view, time probably is the weakness of this technique. Animation consumes time, so there is clearly a tradeoff in how long the animation should take. Fast animation may confuse users and makes it hard to notice the connections. On the other hand, if the transition takes too long, the users' time will be wasted. [. . .] To achieve smoothness of movement, 10 frames per second are generally considered the minimum required frame rate.* (Cui)

5.3 Low-Level Requirements for Graphs

In this section I want to give a general overview on requirements to a graph visualization on the low-level, i.e. from the software implementation point of view.
A graph can be splitted in its components Panel, Node(s), Edge(s), Interface and Algorithm(s). Each of these components have various requirements, some overlapping each other. For a collapsed overview of the requirements on the low-level please have a look at figure 5.2. The tree is constructed from the ontology by putting the graph elements on the first (inner) layer, then putting the low-level requirements without any super-requirements on the second layer, then their sub-requirements on the next layer and so on until no sub-requirements are left. Finally the implementation examples are on the last layer. The connections between the single layers result from the connecting relations (or their triples more exactly). For example the triples

```
ro:Style ro:suitable-for ro:Node
ro:Shape ro:subrequirement-of ro:Style
ro:Shape ro:for-example ro:n-Polygon
```

are displayed as path

```
Node→Style→Shape→n-Polygon
```

within the tree. The graph elements are displayed as boxed nodes. Low-Level-Requirements are displayed with regular font. Implementation Examples are drawn as italics.

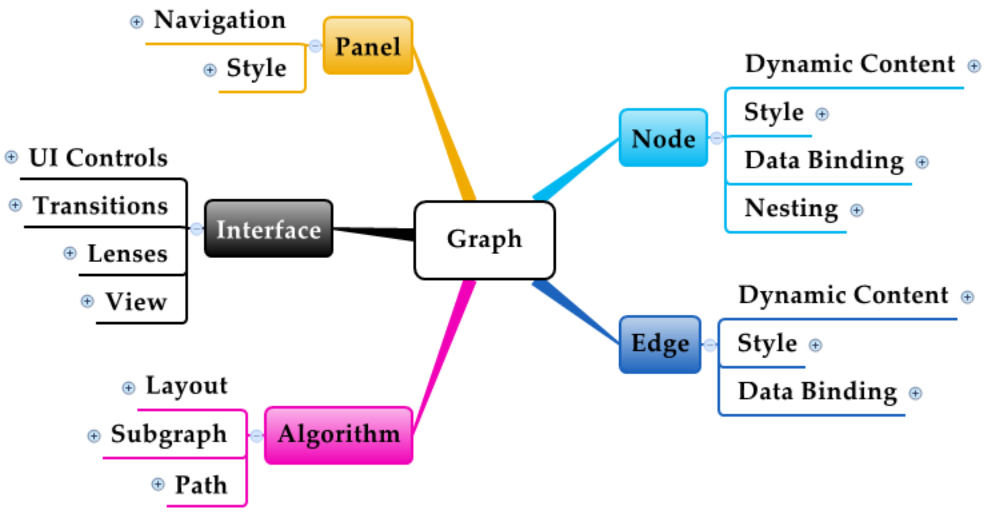


Figure 5.2: Collapsed summary of graph requirements

Graphs in general are not bound in their size or shape. So a continuous zoom is necessary to have an overview on the one side, and to get detailed views of substructures within the graph. Therefore vectorgraphics (SVG) is the first choice for a graph visualization framework. To visualize different types of data in a graph it is necessary to color nodes or edges, to set different line widths, have transparency effects and so on. Such techniques are related to the highlight and filter requirements of human interaction. To provide real interaction possibilities the user has to tell the graph what he wants to see, so context menus must be available for nodes and edges.

- **Zoom**: magnify or shrink a portion of a visualization
- **Panning**: go to currently hidden area
- **Rotate**: change the perspective in three dimensions
- **Add/Remove/Edit**: add/remove/edit nodes and edges
- **Group/Cluster**: group several nodes to one node
- **Color**: color nodes and edges
- **Size**: various sized nodes and edges
- **Transparency**: transparency effects on nodes and edges
- **Shape**: different shapes for nodes and edges
- **Highlight/Animation**: emphasize or brush nodes and edges
- **Picture**: pictures as or in nodes
- **Displacement**: slightly adjusting the edges to curves
- **Data Binding**: binding data to the graph elements, maintaining synchronization of data
- **Nesting**: nest graphs or other visualizations in a node
- **UI Controls**: various user interface components, e.g. buttons, check boxes, text fields etc.
- **Transitions**: have smooth animations when graph state changes
- **Lenses**: tools for enlarging portions of visualizations (movable), e.g. fisheye lense, magnifier
- **View**: 2D or 3D

- **Layout**: various types of layout algorithms
- **Dynamic Layout**: slightly rearrange items after adding or removing nodes and edges
- **Subgraph**: various algorithms for subgraph constructions, e.g. spanning trees, concave/convexe hulls etc.
- **Path**: various algorithms for path computations, e.g. shortest path, max-flow/min-cut etc.
- **Reaction**: react on user interactions

In the ongoing section some of the requirements (including implementation examples) on each graph element are presented.

5.3.1 Panel

The Panel is the base layer of a graph, i.e. the component the nodes and edges are drawn in. Mostly it would simply be a rectangle containing the nodes and edges, as computer monitors are rectangular. At this point we reach the first limitation as huge graphs cannot be appropriately displayed at once - so we need navigational techniques like zoom and panning to at least enable the user to see parts of the graph in a readable size. Also for a considerable integration of a graph and lattice visualization to other tools and products some styling possibilities must be available.

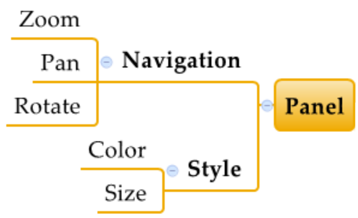


Figure 5.3: Summary of panel requirements

5.3.2 Node and Edge

Since nodes and edges are the "substance" of each graph, they need to have possibilities for dynamically adding, removing, editing and moving them (on the panel). Also nodes and edges should be stylable in color, size, shape, transparency, etc. Nodes can also be visualized by a picture. Edges should be displaceble to gain a more clear arrangement. There are at least three techniques for edge displacement: confluent drawing (see figure 5.5), edge clustering (see figure 5.6) and edge bundling (see figure 5.7). Some animation features like enlarge and reduce, fade in and fade out, intertia on movement, or flash, glow and pulse are strongly needed to give feedback to the user.
To provide live interactable and up-to-date graphs, it must be possible to bind data to nodes and edges. This permits a live and dynamic data preview by styling nodes and edges according to their inherited or connected data, for example coloring all nodes representing ontology classes appropriate to their number of ontology individuals. A second use case: Visualizing the nodes as data points within other visualization types, like line diagrams, scatterplots, histograms, parallel coordinate plots or other. For example if the data represented by a node includes two values on ordinal dimensions, these two dimensions are chosen as the axes of a diagram and the graph's nodes are taken as the points within the diagram while positioning them to these two chosen values. When thinking of multi-layered graphs, i.e. each node is itself a graph, a nesting technique must be available, which means drawing a graph within a node of another graph. A quite similar option is to nest visualization of other types, e.g. a pie chart or a sunburst diagram, to display data represented by a node or connected to a node.

5.3.3 Interface

The Interface connects the user and the graph visualization. It should give possibilities for user interaction with UI controls like menus, triggers, input/output fields etc. Also transitions must be available to enable the user to see changes of the content; it should under all circumstances be avoided that a minor change in the displayed data results in a major change of the visual representation. When showing huge graphs it is hardly possible to show the whole graph in well-arranged way. So to disburden the user and to make a quick

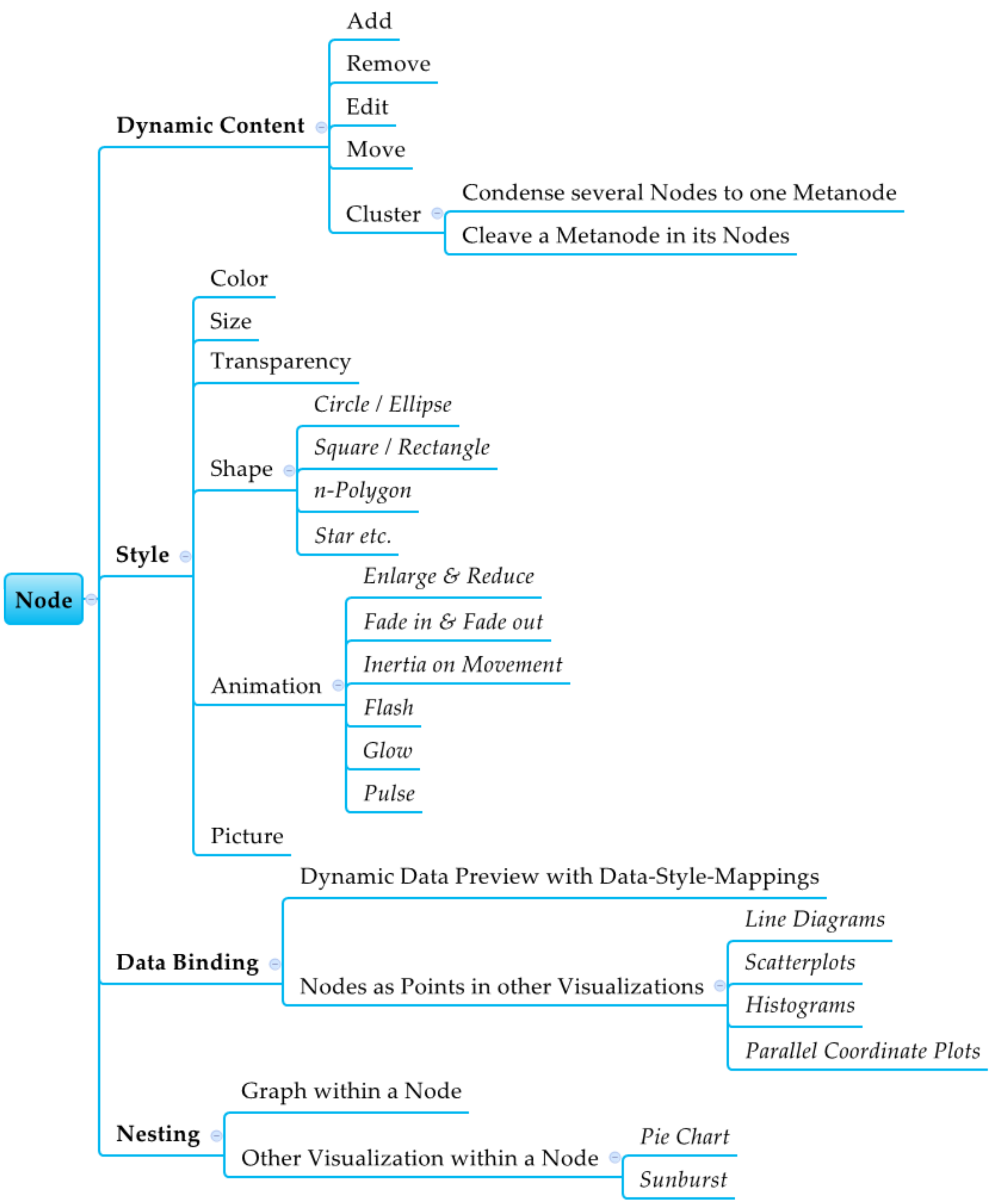


Figure 5.4: Summary of node requirements

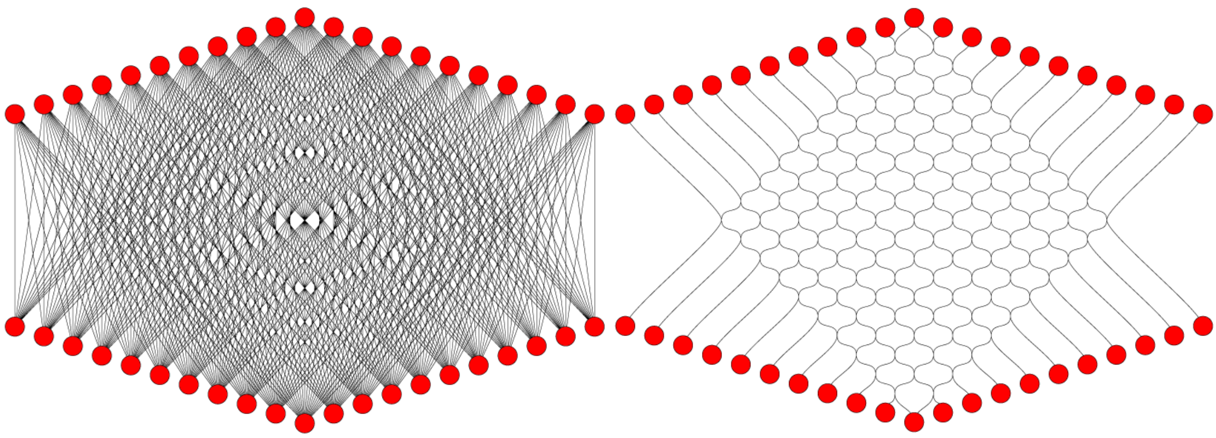


Figure 5.5: Edge displacement by confluent drawing

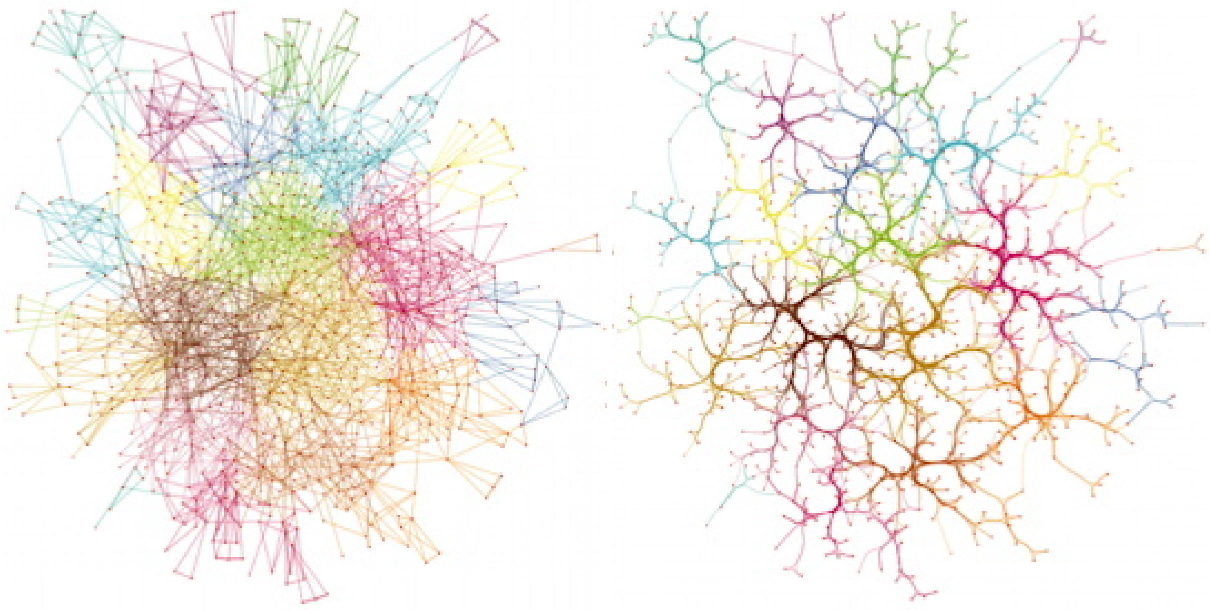


Figure 5.6: Edge displacement by clustering

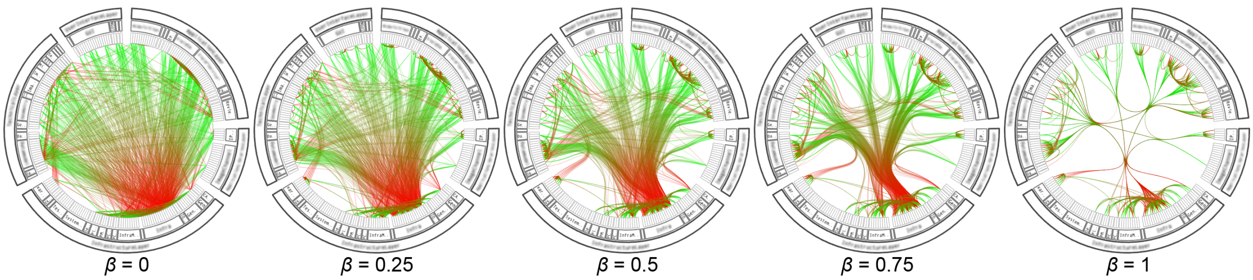


Figure 5.7: Edge displacement by bundling

detail zoom possible, lenses (e.g. fisheye view, simple magnifier) should be implemented. Last but not least a three-dimensional view of a graph could enlarge its clarity.

5.3.4 Algorithm

Algorithms are the hidden core. At first, layout algorithms are indispensable for graph visualizations. There are different types for general graphs, tree graphs and lattice graphs.

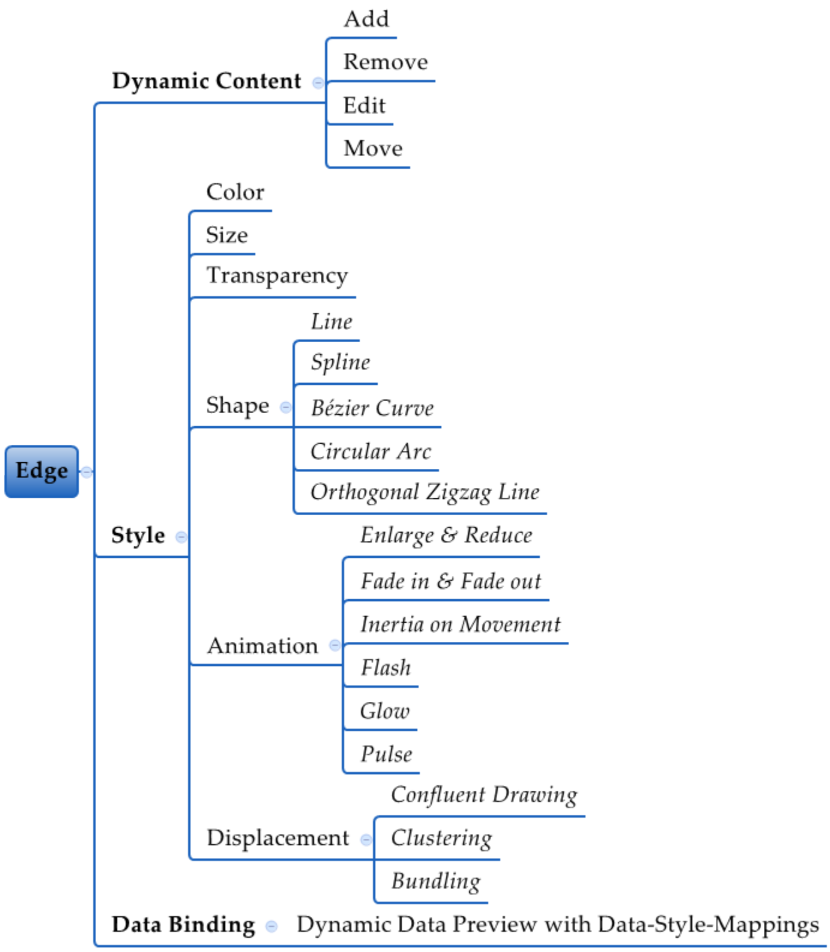


Figure 5.8: Summary of edge requirements

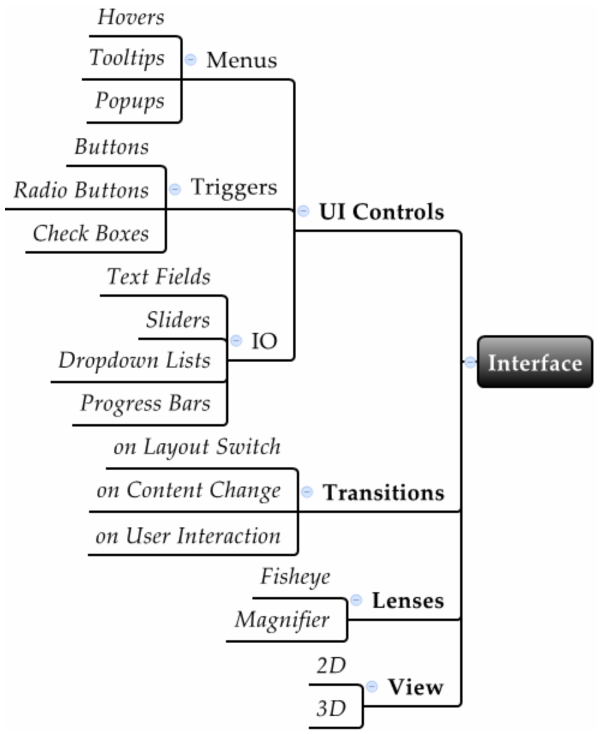


Figure 5.9: Summary of interface requirements

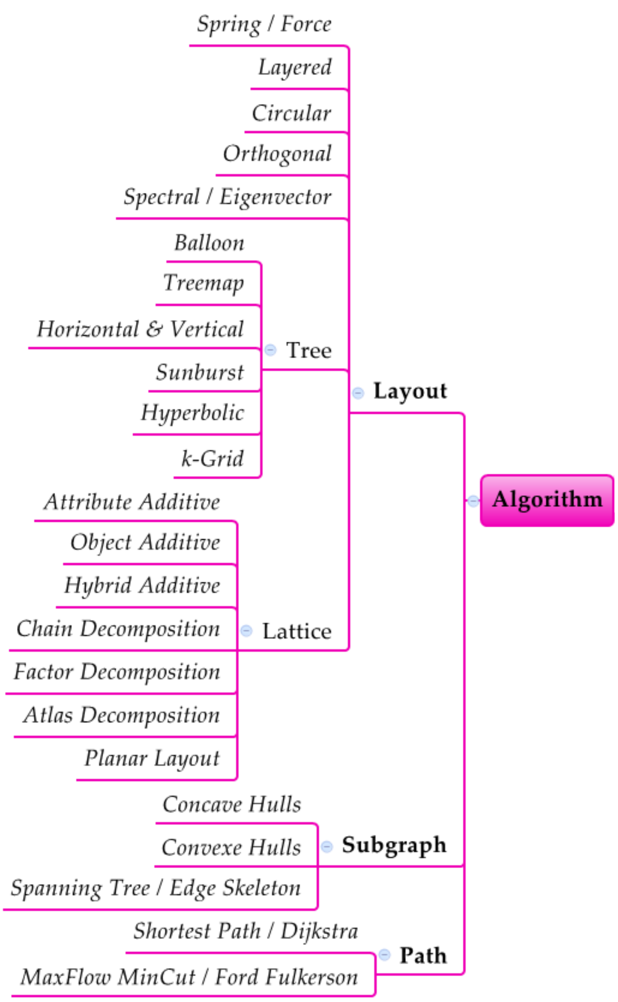


Figure 5.10: Summary of algorithm requirements

5.4 Mapping of Low-Level Requirements to User-Level Requirements

Now the assertional box of the requirements ontology is populated by the preceding sections. As you can see in the figures 5.11 and 5.12, you see nothing in the ontology visualizations by common graph layout algorithms. To overcome this, some formal contexts have been constructed from the ontology by SPARQL queries. These queries have exactly two bound variables and the first one yields the objects and the second variable yields the attributes of the formal context. Whenever a pair fits the query, then the incidence cross is set in the constructed context. Please have a look at figures 5.13, 5.14, 5.15 and 5.16 for expressive examples.

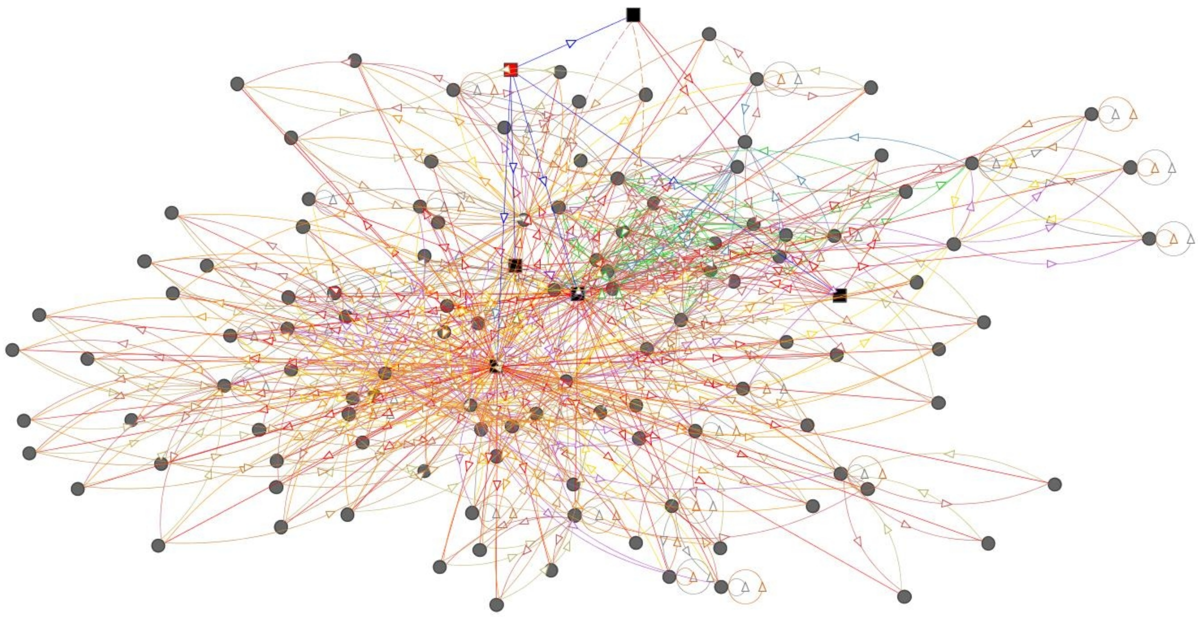


Figure 5.11: ABox of Requirements Ontology (spring layout)

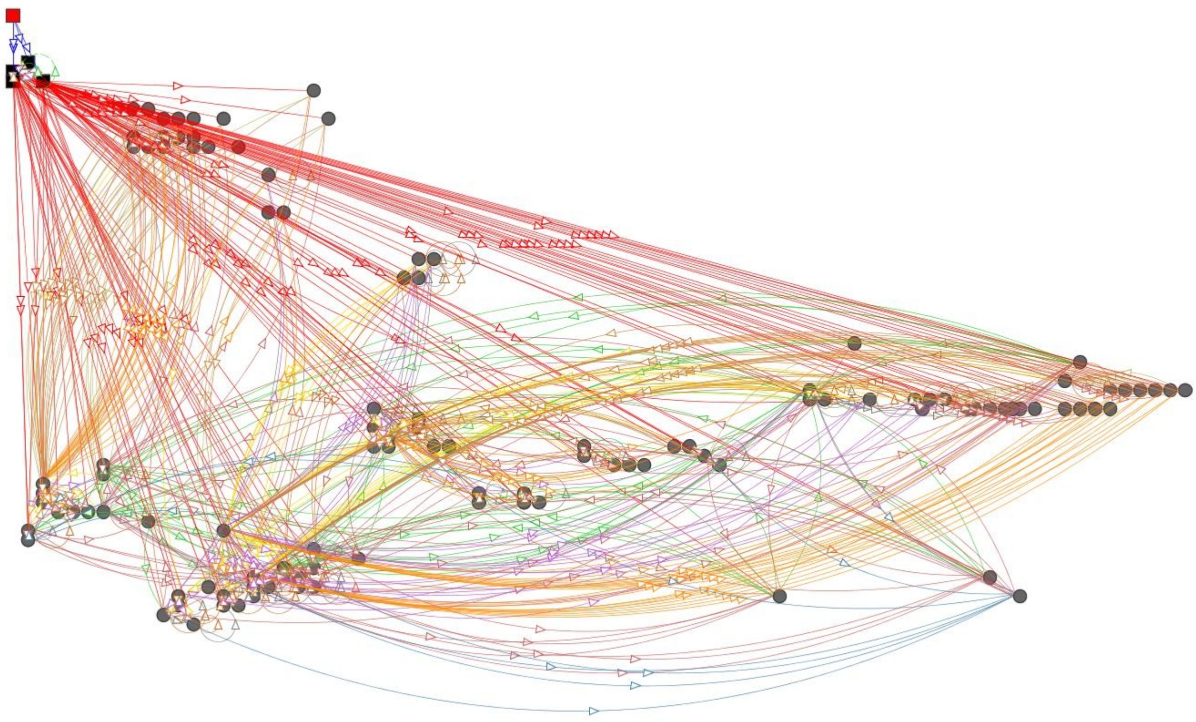


Figure 5.12: ABox of Requirements Ontology (tree layout)

```
PREFIX ro:<http://www.research.sap.com/requirements-ontology#>
SELECT ?ulr ?llr
WHERE {
  ?ulr ro:requires ?llr .
}
```

	dynamic-content	path	layout	nesting	style	transitions	data-binding	ui-controls	subgraph	navigation
connect	×	×	×	×	×	×	×	×	⤢	
reconfigure	×	↗	×	×	×	×	×	×	↗	
filter	×	×	×	×	×	×	×	×	×	⤢
abstract-elaborate	×	⤢	×	×	×	×	×	×	×	
encode	×	↗	×	×	×	×	×	×	↗	
explore	⤢	×	×	×	×	×	×	×	×	×
animate	×	↗	×	⤢	×	×	×	×	↗	
select		×	↙	↙	×	↙	↙	×	×	×

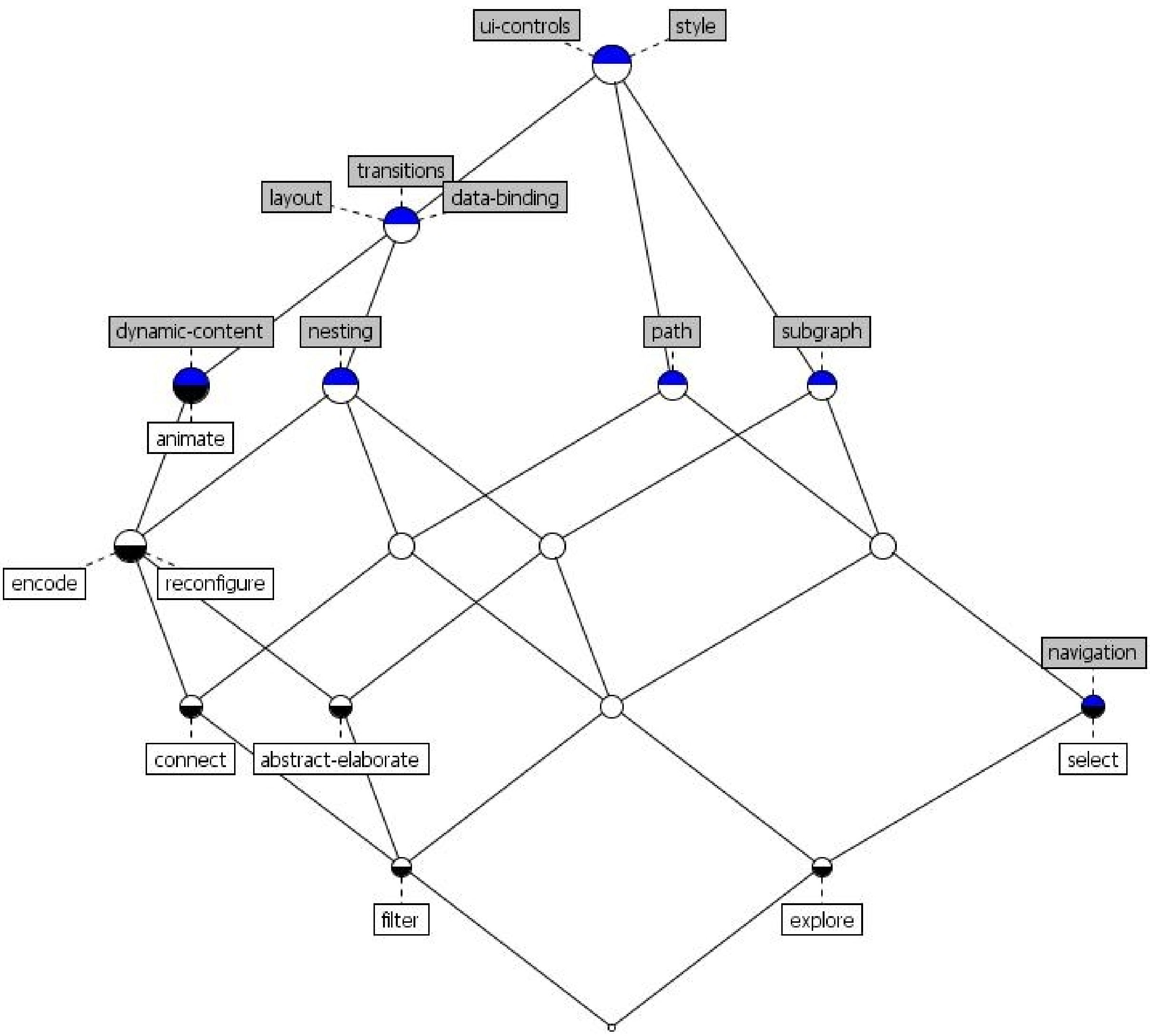


Figure 5.13: Mapping between Top-Low-Level and User-Level Requirements

```
PREFIX ro:<http://www.research.sap.com/requirements-ontology#>
SELECT ?ge ?llr
WHERE {
  ?ge ro:hasRequirement ?llr .
  FILTER NOT EXISTS {
    ?sllr ro:subrequirementOf ?llr
    FILTER (?sllr != ?llr)
  }
}
```

	dynamic-content	style	data-binding	nesting	navigation	path	layout	subgraph	view	lenses	transitions	ui-controls	picture
node	×	×	×	×	⤢	↙	↙	↙	↙	↙	↙	↙	×
panel	↙	×	↙	↙	×	↙	↙	↙	↙	↙	↙	↙	↙
edge	×	×	×	↙	↗								↙
algorithm	↙	⤢	↙	↙	↙	×	×	×	↙	↙	↙	↙	↙
interface	↙	⤢	↙	↙	↙	↙	↙	↙	×	×	×	×	↙

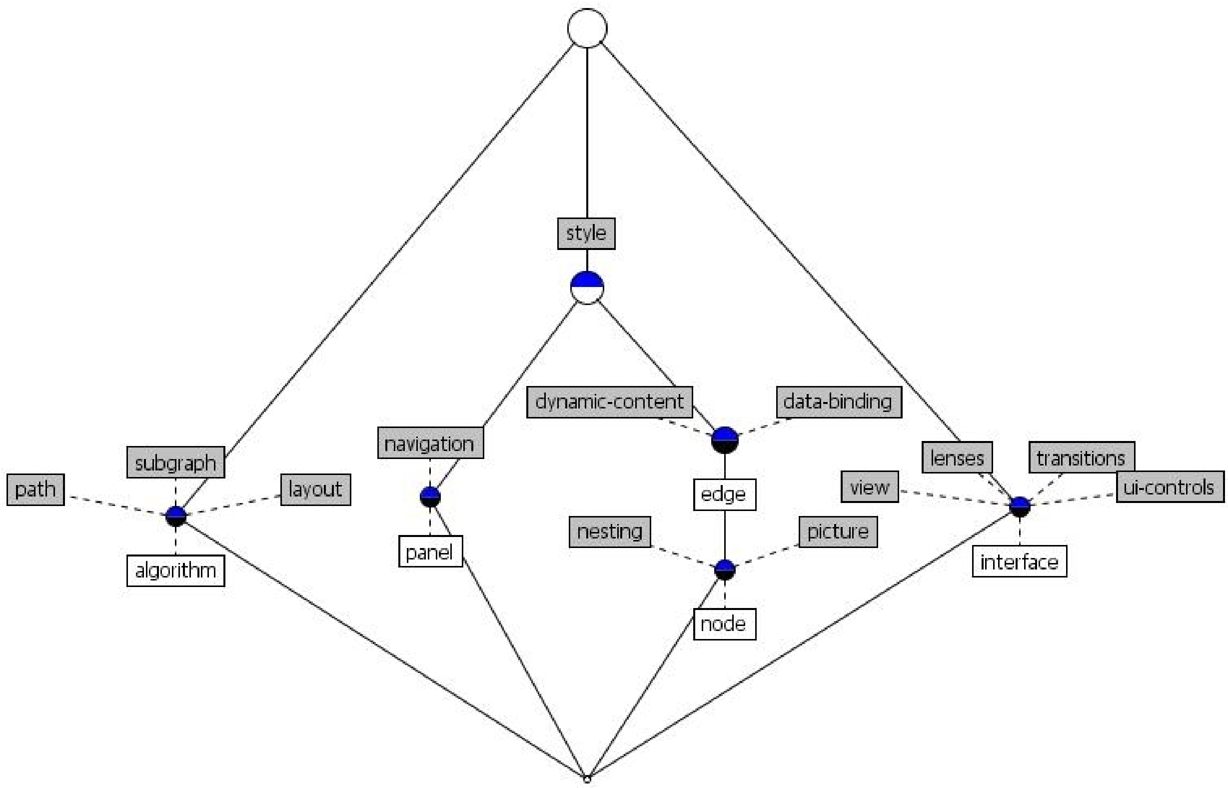


Figure 5.14: Mapping between top-low-level requirements and graph elements

```
PREFIX ro:<http://www.research.sap.com/requirements-ontology#>
SELECT ?ip ?llr
WHERE {
  ?ip ro:prioritizes ?llr .
}
```

	dynamic-content	layout	navigation	style	transitions	data-binding	ui-controls	path	view	lenses	nesting	subgraph
necessary	×	×	×	×	×	×	×	✓	✓	✓	✓	✓
optional	✓	✓	✓	✓	✓	✓	✓	×	×	×	×	×

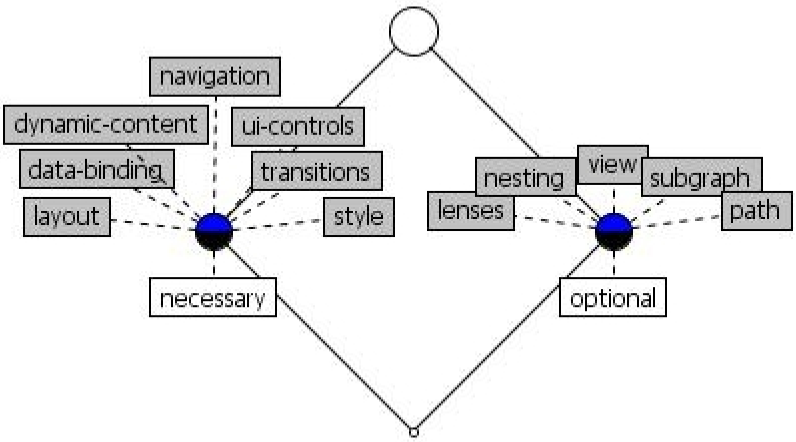


Figure 5.15: Mapping between top-low-level requirements and implementation priorities

```
PREFIX ro:<http://www.research.sap.com/requirements-ontology#>
SELECT ?ge ?llr
WHERE {
  ?ge ro:hasRequirement ?llr .
  FILTER (
    ?ge != ro:algorithm
    && ?ge != ro:interface
  )
}
```

	picture	dynamic-content	size	add	animation	nodes-as-points-in-other-visualizations	remove	color	transparency	style	data-binding	move	shape	graph-within-node	cleave-a-metanode-in-its-nodes	edit	dynamic-data-preview-with-data-style-mappings	nesting	condense-several-nodes-to-one-metanode	other-visualization-within-node	cluster	rotate	pan	zoom	navigation	displacement
node	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	↙	↙	↙	↙	⤢
panel	↙	↙	×	↙	↙	↙	↙	×	↙	×	↙	↙	↙	↙	↙	↙	↙	↙	↙	↙	↙	×	×	×	×	↙
edge	↙	×	×	×	×	↙	×	×	×	×	×	×	×	↙	↙	×	×	↙	↙	↙	↙	↙	↙	↙	↙	×

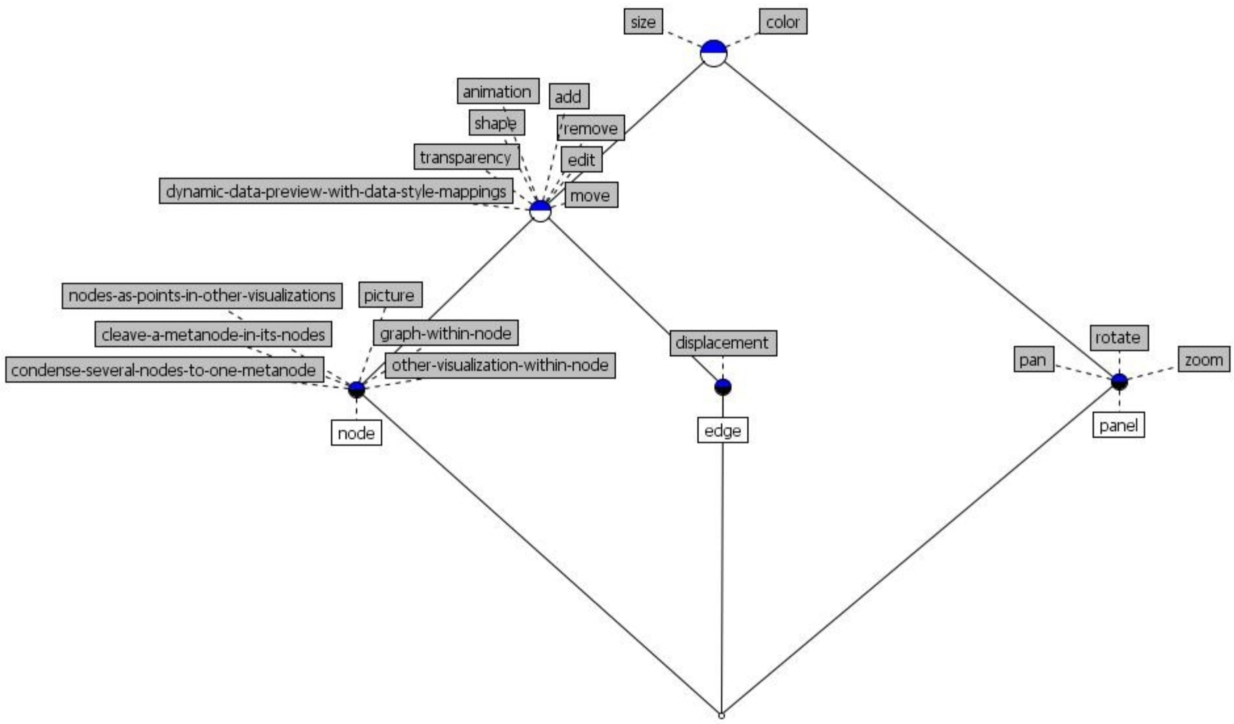


Figure 5.16: Mapping between low-level requirements and some graph elements

5.5 Specific Visualization Requirements for Lattices

In this section, we describe some additional requirements which apply to so-called concept lattices (i.e. lattices gained from formal contexts). This section is an important part of this document as concept lattices are used as a main visualization means in CUBIST, and might further be used in other contexts as well.
Lattices are a special kind of ordered sets (i.e. sets with a reflexive, antisymmetric and transitive binary relation), which impose additional restrictions. The nodes in concept lattices contain additional information, which in turn enable specific layout algorithms, see 5.5.6.
Readers who are only interested in graph-visualization as such can safely skip this chapter.

5.5.1 Lattice Zoom/Recursive Lattices/Partly Nested Lattices

A useful requirement for drawing lattices is the ability to split up one node into several nodes, that themselves make up a lattice. Then it would be necessary to shift nodes in the filter upwards and nodes in the ideal downwards to retain the order structure of the lattice.

5.5.2 Planarity

Graph diagrams and lattices as special graphs are much better readable, when the number of intersecting edges is minimal. Sometimes it is even possible to visualize lattices without any edge crossings, these lattices are called *planar*. Please see figure 5.17 for an example. There exist some mathematical research papers and a dissertation from DR. CHRISTIAN ZSCHALIG regarding this, please see (Zsc05) or (Zsc07). In his work he invents and presents a way to construct planar diagrams of lattices, if possible. A lattice can be drawn planar, if and only if its order dimension is at most two, i.e. if it can be embedded in a grid of two chains.

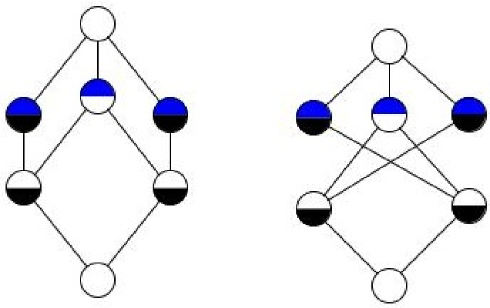


Figure 5.17: Both pictures show the same lattice but with different layouts. The left one is planar while the right one is not.

5.5.3 Labels

Formal concepts or lattice elements in general should have a name to be identifiable by the user, and an internal id to be identifiable by the computer.

5.5.4 Selection of Ideals, Filters and Intervalls

As a subrequirement of selection we need the ability to select ideals, filters and intervalls in lattices. An ideal is simply the set of all elements under a certain lattice element, dually a filter consists of all elements above a certain element. Intervalls are the intersection of both an ideal and a filter and just represents the set of all lattice elements between two given elements.

5.5.5 Restricted Moving of Elements

Since the elements of a lattice are ordered, it must be avoided that a lattice element can be moved over any of its upper neighbors und dually under any of its lower neighbors. This ensures the original order in the changed diagram. As a possible solution for unlimited moving it is possible to move also the corresponding ideal and filter when moving an element. This would ensure the original order, too. Both possibilities are shown in figure 5.18.

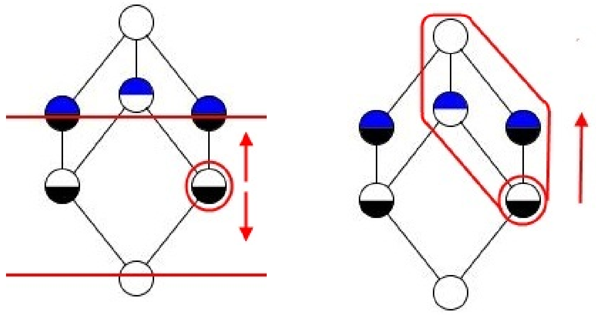


Figure 5.18: Left: The red marked node can only be move within the two red lines. Right: The red marked node can be moved upwards arbitrarily when the corresponding filter (red bounded) is also moved upwards.

5.5.6 Layout Algorithms

It has to be possible to implement and use different layout algorithms for displaying a lattice. There exists a number of layout algorithms, e.g. additive layout, force driven layout, chain decomposition layout, hybrid layouts, etc.

5.5.7 Additional Feature: Three Dimensions and Rotation

An extra feature is the visualization in three dimensions, with the ability to rotate the diagram. Then there were less edge crossings as in an diagram projected in the plane. It would be possible to extend the planarity of ZSCHALIG to three-dimensionality.

5.5.8 Additional Feature: Nesting

Nesting enables diagrams to have another visualization in a node. For some cases it could provide a higher information density, but in other cases it is possibly lowering the readability. So its usage should be well considered. In figure 5.19 there is a sample for nesting diagrams.

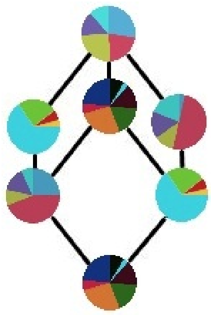


Figure 5.19: A lattice nested with a pie chart in each node.

6.1 Architecture

Within the framework of this diploma thesis, a program package for computional formal concept analysis was written. The used programming languange is Java, and it was developed in Eclipse with Maven build tool. After a refactoring, there were about 10000 lines of code. The program module is called FCAFOX. Its features are as follows:

- At first, a set list structure was implemented. It is simply a crossover between a set and a list, *i.e.* it implements both the set and the list interfaces. Each element of a set list can be contained at most once, and furthermore has an index, *i.e.* there is an enumeration function that maps each element injectively to a natural number. See figure 6.1 for a class diagram.

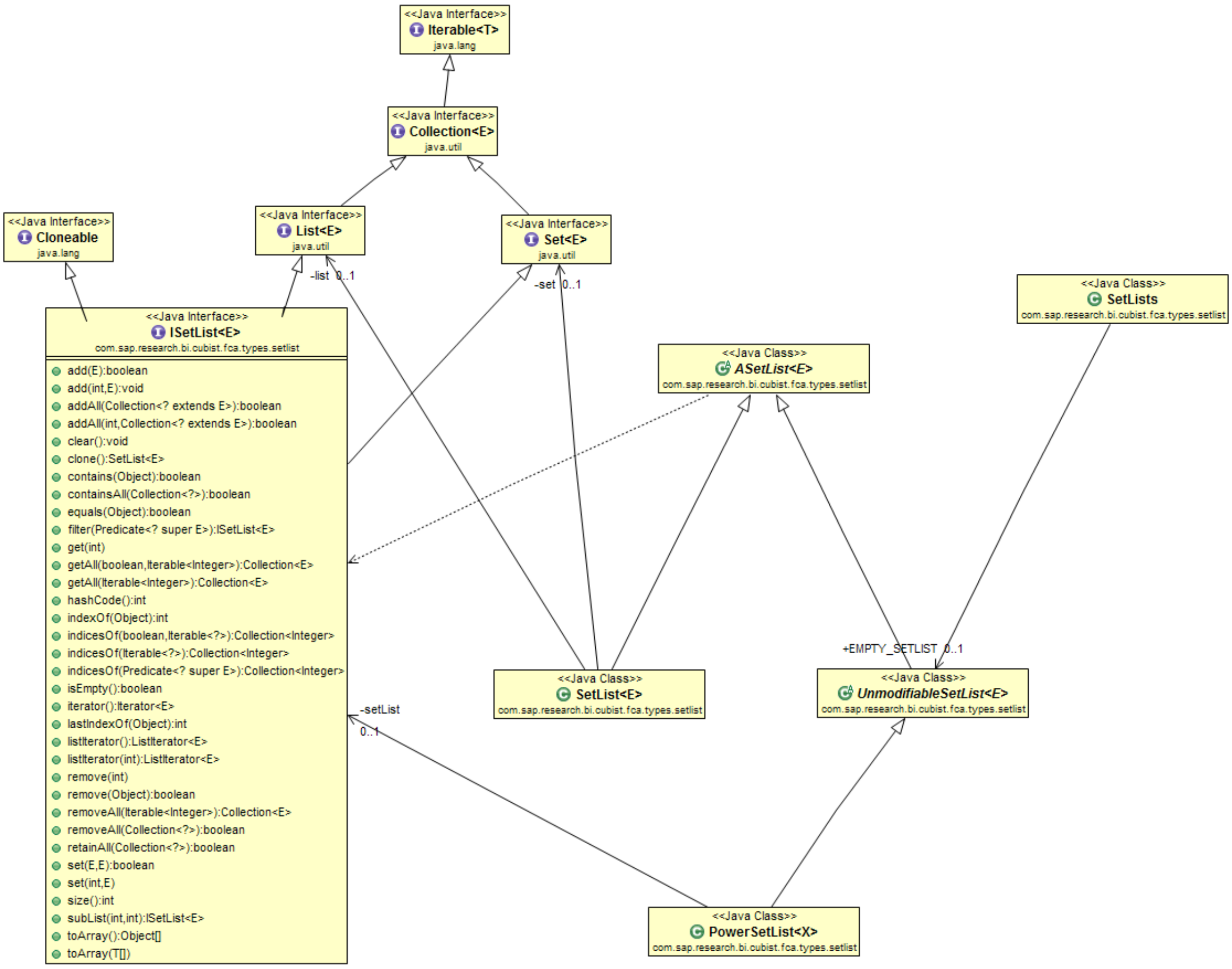


Figure 6.1: SetLists class diagram

- Second, an incidence structure was modelled. Each instanciated incidence internally consists of a boolean matrix and two set lists. The first set lists is used as domain and the other as codomain, and changes in domain or codomain are reflected in the matrix, *e.g.* removing a domain element also removes the appropriate row from the matrix. An incidence behaves like a set of pairs, *i.e.* one can do set like operators on them, *e.g.* union, intersection, difference, *etc.* Furthermore, one can grab a row or column from an incidence, which behaves like a set then and changes are reflected in the matrix as well. Please note, one can only remove or add an element from or to an row/column, when the element is already contained in the domain/codomain. See figure 6.2 for a class diagram.

- Formal contexts are modelled as a subtype of incidence, with some additional operations.

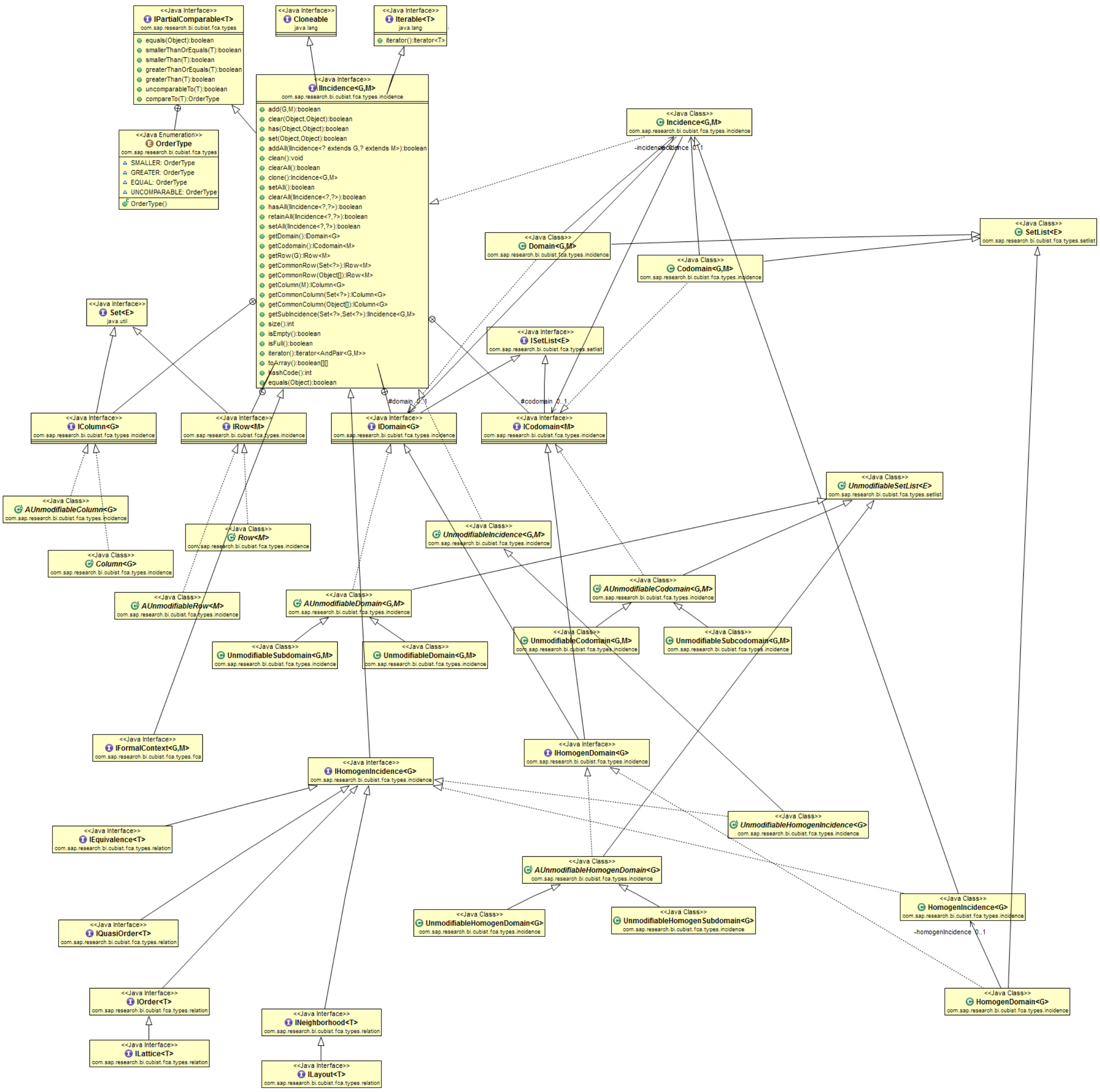


Figure 6.2: Incidences class diagram

- It is both possible to work with formal contexts in a set-like fashion directly (with higher operational costs for index lookups) or to work with indices as long as possible and perform the index lookup in the end.
- Cleaning and reducing of formal contexts are computed by lazy evaluation. Also the arrow contexts including the double arrows can be computed.
- An event system has been implemented, see figure 6.3. A component can be registered at an incidence and is then informed about changes in the incidence, *e.g.* new elements in domain or codomain, changes in rows or columns, *etc.*
- There are different interfaces for so called processors, that can be registered at a formal context. Possible processor types are concept processors, neighborhood processors, layout processors, importer and exporter. Of course it can be extend if necessary.
- Implemented Processors: my IFOX concept/neighborhood/layout processor, NextClosure concept processor, experimental fault tolerance concept processors, BorderAlg neighborhood processor, iPred neighborhood processor, additive layout processor (with seeds selection by chain decomposition, subdirect decomposition, context rearrangement/shaking, spectral decomposition), projective layout processor, SPARQL

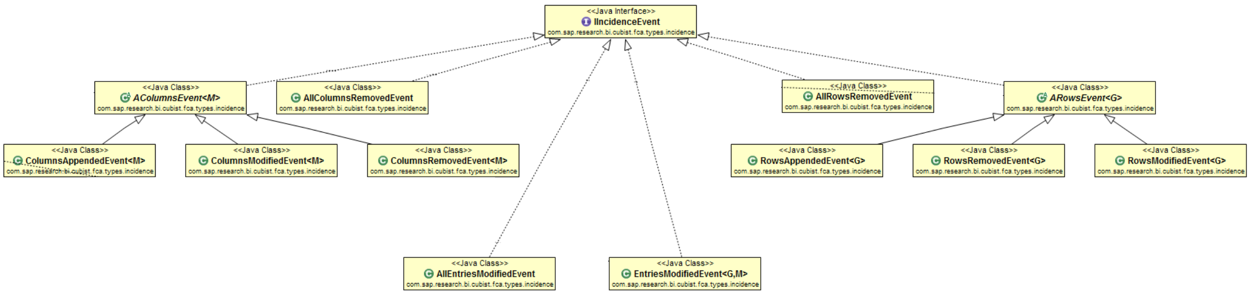


Figure 6.3: Events class diagram

importer and exporter (Dau12), Burmeister format importer and export, LaTeX exporter (GANTER's FCA package, my TiKZ concept lattice package)

- Live heatmap visualization on node movement. A combination of the conflict avoidance parameter metric (Gan) and edge crossing metric is used.

- The module is integrated into the CUBIST prototype and connected to a SVG graph panel via server/-client architecture.

- Additional features: scale generators, order interface, equivalence interface, lattice interface, . . .

A Appendix

A.1 Synonym Lexicon

Mathematical Term	Data Mining Term
context	binary database, transactional database
object	transaction, row
attribute	item, column
pair of sets	bi-set
preconcept	rectangle
set of attributes	itemset, attribute pattern
set of objects	object pattern
implication	association rule

A.2 Galois Connections & Galois Lattices

The first section defines a general structure called *galois connection* and shows a way to obtain a lattice from an arbitrary galois connection. Furthermore a general possibility to modify a galois connection by arbitrary kernel operators is introduced. A *Galois connection* is a special correspondence between two ordered sets. Galois connections generalize the Galois theory by Évariste Galois (France, 1811 – 1832), that investigate correspondences between subgroups and subfields. Galois connections are rather weak compared to dual order isomorphisms, but every Galois connection induces a dual order isomorphism of certain kernel systems within the involved ordered sets.

Definitio: Galois Connection A.1

A GALOIS CONNECTION between two ordered sets $(P, \leq)$ and $(Q, \leq)$ is a pair (ϕ, ψ) of mappings $\phi\colon P \to Q$ and $\psi\colon Q \to P$ such that the following conditions hold:

(I) ϕ and ψ are antimonotonic[a]

(II) $\psi \circ \phi$ and $\phi \circ \psi$ are extensive[b]

The two mappings are called DUALLY ADJOINT to each other and we write

$$(P, \leq) \overset{(\phi,\psi)}{\bullet\!\!-\!\!\bullet} (Q, \leq).$$

A GALOIS CONNECTION between two arbitrary sets X and Y is a galois connection (ϕ, ψ) between the two powersets $\wp X$ and $\wp Y$ canonically ordered by subset inclusion $\subseteq$, *i.e.*

$$(\wp X, \subseteq) \overset{(\phi,\psi)}{\bullet\!\!-\!\!\bullet} (\wp Y, \subseteq).$$

[a] *i.e.* $p_1 \leq p_2$ implies $\phi p_1 \geq \phi p_2$ for all $p_1, p_2 \in P$, and analogously for ψ

[b] *i.e.* $p \leq \psi\phi p$ holds for all $p \in P$, and analogously for $\phi \circ \psi$

A pair (ϕ, ψ) of mappings as above is a galois connection iff

$$p \leq \psi q \Leftrightarrow q \leq \phi p$$

holds for all $p \in P$ and $q \in Q$.
For any galois connection (ϕ, ψ) it holds that $\phi \circ \psi \circ \phi = \phi$ and $\psi \circ \phi \circ \psi = \psi$.
Galois connections are in a strong correspondence to closure operators. Each closure operator ϕ in an ordered set $(P, \leq)$ induces a galois connection $(P, \leq) \overset{(\phi,\mathrm{id})}{\bullet\!\!-\!\!\bullet} (P, \geq)$. Dually each galois connection (ϕ, ψ) between two ordered sets $(P, \leq)$ and $(Q, \leq)$ induces two closure operators $\psi \circ \psi$ and $\phi \circ \psi$.

A.2 **Exemplum**

(I) For a binary relation $I \subseteq G \times M$ or for a formal context (G, M, I) respectively we get a galois connection (ϕ_I, ψ_I) between $(\wp G, \subseteq)$ and $(\wp M, \subseteq)$ with

$$\phi_I \colon \begin{array}{l} \wp G \to \wp M \\ A \mapsto A^I := \left\{ m \in M \,\middle|\, \forall_{g \in A}\, gIm \right\} \end{array}$$

$$\psi_I \colon \begin{array}{l} \wp M \to \wp G \\ B \mapsto B^I := \left\{ g \in G \,\middle|\, \forall_{m \in B}\, gIm \right\} \end{array}$$

Please have a look on the Theorema: Contexts & Galois Connections 2.2 for further details.

(II) For a pattern structure $(G, (D, \sqcap), \delta)$ a galois connection $(\phi_\delta, \psi_\delta)$ between $(\wp G, \subseteq)$ and $(D, \sqsubseteq)$ is obtained by

$$\phi_\delta \colon \begin{array}{l} \wp G \to D \\ A \mapsto A^\square := \bigsqcap \delta A = \bigsqcap_{g \in A} \delta g \end{array}$$

$$\psi_\delta \colon \begin{array}{l} D \to \wp G \\ d \mapsto d^\square := \{ g \in G \mid d \sqsubseteq \delta g \} \end{array}$$

Both mappings are obviously antimonotonic[a] and their compositions are extensive[b].

[a] For object sets $A_1 \subseteq A_2 \subseteq G$ we can easily see that $A_1^\square = \bigsqcap \delta A_1 \sqsupseteq \bigsqcap \delta A_2 = A_2^\square$, as $\delta A_1 \subseteq \delta A_2$ holds. For patterns $d_1 \sqsubseteq d_2$ we have $d_1^\square = \{g \in G \mid d_1 \sqsubseteq \delta g\} \supseteq \{g \in G \mid d_2 \sqsubseteq \delta g\} = d_2^\square$, as $d_2 \sqsubseteq \delta g$ always implies $d_1 \sqsubseteq \delta g$.
[b] For each object set $A \subseteq G$ it follows that $A \subseteq \{g \in G \mid \bigsqcap \delta A \sqsubseteq \delta g\}$, as $\bigsqcap \delta A \sqsubseteq \delta g$ holds for all objects $g \in A$. For each pattern $d \in D$ it trivially holds that $d \sqsubseteq \bigsqcap \delta \{g \in G \mid d \sqsubseteq \delta g\} = \bigsqcap_{d \sqsubseteq \delta g} \delta g$.

There are various other examples for galois connections, that occur in many mathematical fields. From each galois connection an ordered set or even a complete lattice can be found within the cartesian product of the basic sets.

A.3 **Theorema: Galois Lattice**

Let (ϕ, ψ) be a galois connection between two posets $(P, \leq)$ and $(Q, \leq)$. Define $(\mathcal{G}(\phi, \psi), \leq)$ with

$$\mathcal{G}(\phi, \psi) = \{(p, q) \in P \times Q \mid \phi p = q, \psi q = p\}$$

and $(p_1, q_1) \leq (p_2, q_2)$ iff $p_1 \leq p_2$. Then $(\mathcal{G}(\phi, \psi), \leq)$ is a poset. If further $(P, \leq)$ and $(Q, \leq)$ are complete lattices, then $(\mathcal{G}(\phi, \psi), \leq)$ is a complete lattice and the infima and suprema are given by

$$\bigwedge_{t \in T} (p_t, q_t) = \left(\bigwedge_{t \in T} p_t, \phi\psi \bigvee_{t \in T} q_t \right)$$

$$\bigvee_{t\in T}(p_t,q_t)=\left(\psi\phi\bigvee_{t\in T}p_t,\bigwedge_{t\in T}q_t\right)$$

Theorema A.4

For a galois connection $(P,\leq)\overset{(\phi,\psi)}{\bullet\!\!-\!\!\bullet}(Q,\leq)$, a kernel operator (projection) α on $(P,\leq)$ and another kernel operator (projection) β on $(Q,\leq)$ it holds:

(I) A new galois connection $(\alpha P,\leq)\overset{(\beta\circ\phi,\alpha\circ\psi)}{\bullet\!\!-\!\!\bullet}(\beta Q,\leq)$ is obtained.

(II) Furthermore, there is a order-preserving epimorphism of $\mathcal{G}(\phi,\psi)$ on $\mathcal{G}(\beta\circ\phi,\alpha\circ\psi)$ given by

$$\iota\colon\begin{array}{c}\mathcal{G}(\phi,\psi)\twoheadrightarrow\mathcal{G}(\beta\circ\phi,\alpha\circ\psi)\\ (p,q)\mapsto(\alpha p,\beta\phi\alpha p)\end{array}$$

(III) Dually, there is another order-preserving epimorphism of $\mathcal{G}(\phi,\psi)$ on $\mathcal{G}(\beta\circ\phi,\alpha\circ\psi)$ given by

$$\kappa\colon\begin{array}{c}\mathcal{G}(\phi,\psi)\twoheadrightarrow\mathcal{G}(\beta\circ\phi,\alpha\circ\psi)\\ (p,q)\mapsto(\alpha\psi\beta q,\beta q)\end{array}$$

$$\begin{tikzcd}
(P,\leq) \arrow[r, dash, "{(\phi,\psi)}"] \arrow[d, "\alpha"] & (Q,\leq) \arrow[d, "\beta"] \\
(\alpha P,\leq) \arrow[r, dash, "{(\beta\circ\phi,\alpha\circ\psi)}"] & (\beta Q,\leq)
\end{tikzcd}$$

APPROBATIO (I) Let $p\in\alpha P$, i.e. $\alpha p=p$, and $q\in\beta Q$, i.e. $\beta q=q$. From $p\leq\alpha\psi q$ follows $p\leq\psi q$ as α is intensive. Since (ϕ,ψ) is a galois connection, this implies $q\leq\phi p$. Further it follows $q=\beta q\leq\beta\phi p$ because β is monotone. The other way round follows dually.

(II) At first we show that the images of ι are really elements of $\mathcal{G}(\beta\circ\phi,\alpha\circ\psi)$. So let $(p,q)\in\mathcal{G}(\phi,\psi)$, then $\iota(p,q)=(\alpha p,\beta\phi\alpha p)\in\mathcal{G}(\beta\circ\phi,\alpha\circ\psi)$ holds as

$$\alpha\psi\beta\phi\alpha p=\alpha\psi\beta\phi\alpha\psi q=\alpha\psi q=\alpha p.$$

Now we show that ι preserves order. For $(p_1,q_1),(p_2,q_2)\in\mathcal{G}(\phi,\psi)$ we have

$$\begin{aligned}&(p_1,q_1)\leq(p_2,q_2)\\ \Rightarrow\,&p_1\leq p_2\\ \Rightarrow\,&\alpha p_1\leq\alpha p_2\\ \Rightarrow\,&(\alpha p_1,\beta\phi\alpha p_1)\leq(\alpha p_2,\beta\phi\alpha p_2)\\ \Rightarrow\,&\iota(p_1,q_1)\leq\iota(p_2,q_2).\end{aligned}$$

For an element $(p,q)\in\mathcal{G}(\beta\circ\phi,\alpha\circ\psi)$ we have obviously $(\psi q,\phi\psi q)\in\mathcal{G}(\phi,\psi)$ and

$$\iota(\psi q,\phi\psi q)=(\alpha\psi q,\beta\phi\alpha\psi q)=(p,\beta\phi p)=(p,q),$$

and thus ι is an epimorphism.

(III) analogously. ∎

A.5 **Exemplum: Generalized Iceberg Lattices a.k.a. Alpha Galois Lattices**

We investigate the galois connection $(\wp G, \subseteq) \overset{(\phi_I,\psi_I)}{\bullet\!\!-\!\!\bullet} (\wp M, \subseteq)$ induced by a formal context (G, M, I). Let $\mathcal{C} = \{C_i\}_{i=0}^n \subseteq \wp\wp(G)$ be partition of G. Then form the kernel system $\mathcal{K}_\mathcal{C}$ generated by $\mathcal{C}$, *i.e.* close it under arbitrary joins:

$$\mathcal{K}_\mathcal{C} = \langle \mathcal{C} \rangle_\bigcup = \left\{ \bigcup \mathcal{X} \,\middle|\, \mathcal{X} \subseteq \mathcal{C} \right\}$$

This leads to a kernel operator

$$\alpha_\mathcal{C} : \begin{array}{l} \wp G \to \wp G \\ A \mapsto \bigcup\limits_{\substack{K \in \mathcal{K}_\mathcal{C} \\ K \subseteq A}} K = \bigcup\limits_{\substack{\mathcal{X} \subseteq \mathcal{C} \\ \bigcup \mathcal{X} \subseteq A}} \bigcup \mathcal{X} = \bigcup\limits_{\substack{C \in \mathcal{C} \\ C \subseteq A}} C \end{array}$$

and we modify it to

$$\alpha_{\mathcal{C},\delta} : \begin{array}{l} \wp G \to \wp G \\ A \mapsto \bigcup\limits_{\substack{C \in \mathcal{C} \\ \frac{|A \cap C|}{|C|} \geq \delta}} (A \cap C) \end{array}$$

Then the galois lattice $\mathfrak{B}_{\mathcal{C},\delta}(G, M, I) := \mathcal{G}(\phi_I, \alpha_{\mathcal{C},\delta} \circ \psi_I) = \left\{ (A, B) \in \wp(G) \times \wp(M) \mid A^I = B \text{ and } \alpha_{\mathcal{C},\delta}(B)^I = A \right\}$ is called ALPHA ICEBERG LATTICE *w.r.t.* $\mathcal{C}$ and δ. Furthermore then a natural order epimorphism from $\mathfrak{B}(G, M, I)$ onto $\mathfrak{B}_{\mathcal{C},\delta}(G, M, I)$ given by $(A, B) \mapsto (\alpha_{\mathcal{C},\delta}(A), \alpha_{\mathcal{C},\delta}(A)^I)$. See (VS05) for further details.

A.3 Fault Tolerance Extensions to Formal Concept Analysis

A pair $(A, B) \in \wp(G) \times \wp(M)$ is a formal concept of (G, M, I), iff the following conditions hold:

1. $\forall_{g \in A} \left| g^{\not I} \cap A \right| = 0$ and dually $\forall_{m \in B} \left| m^{\not I} \cap B \right| = 0$
2. $\forall_{h \in \complement A} \forall_{g \in A} \left| h^{\not I} \cap A \right| > 0$ and dually $\forall_{n \in \complement B} \forall_{m \in B} \left| n^{\not I} \cap B \right| > 0$

A.6 **Definitio: support, frequency, γ-frequent**

For an object set $X \subseteq G$ or an attribute set $X \subseteq M$ respectively the SUPPORT and FREQUENCY are defined as

$$\text{supp}(X) := X^I \quad \text{and} \quad \text{freq}(X) := |\text{supp}(B)|.$$

For $B \subseteq M$ and $\text{freq}(B) \geq \gamma$ we say that B is γ-FREQUENT.

A.7 **Definitio: association rule, frequency, confidence**

A pair (B, D) is called ASSOCIATION RULE iff $B, D \subseteq M$, $D \neq \emptyset$ and $B \cap D = \emptyset$. An association rule (B, D) is usually written as $B \Rightarrow D$ and its FREQUENCY and CONFIDENCE respectively are

$$\text{freq}(B \Rightarrow D) := \text{freq}(B \cup D) \quad \text{and} \quad \text{conf}(B \Rightarrow D) := \frac{\text{freq}(B \Rightarrow D)}{\text{freq}(B)}.$$

Let $B \Rightarrow D$ be an association rule. Then $\text{conf}(B \Rightarrow D) = 1$ holds iff $\text{supp}(B) \subseteq \text{supp}(D)$ (safe rule). Also, $\text{conf}(B \Rightarrow D) = 0$ holds iff $\text{supp}(B) \cap \text{supp}(D) = \emptyset$ (impossible rule).

Definitio: violation, strength, δ-strong A.8

For an association rule $B \Rightarrow D$ the VIOLATION and STRENGTH respectively are defined as

$$\mathrm{viol}(B \Rightarrow D) := \mathrm{supp}(B) \setminus \mathrm{supp}(D) \quad \text{and} \quad \mathrm{stre}(B \Rightarrow D) := |\mathrm{viol}(B \Rightarrow D)|.$$

An association rule $B \Rightarrow D$ is δ-STRONG iff $\mathrm{stre}(B \Rightarrow D) \leq \delta$ holds, i.e. the rule is not violated in more than δ objects. Then we write $B \Rightarrow^{\delta} D$.

For an association rule $B \Rightarrow D$ holds

$$\mathrm{stre}(B \Rightarrow D) = \mathrm{freq}(B) - \mathrm{freq}(B \cup D).$$

Definitio: δ-free set, δ-closure, δ-biset A.9

An attribute set $B \subseteq M$ is δ-FREE iff there is no δ-strong association rule $B \Rightarrow^{\delta} m$ such that $m \notin B$. The δ-CLOSURE of a δ-free set $B \subseteq M$ is the maximal superset $\mathrm{clos}_{\delta}(B) \supseteq B$ such that $B \Rightarrow m$ is a δ-strong association rule for all attributes $m \in \mathrm{clos}_{\delta}(B) \setminus B$. Each δ-free set $B \subseteq M$ induces a δ-BISET $(\mathrm{supp}(B), \mathrm{clos}_{\delta}(B))$.

A δ-biset is always a union of concepts.

Bibliography

[AGU72] AHO, A. V. ; GAREY, M. R. ; ULLMAN, J. D.: The Transitive Reduction of a Directed Graph. In: *SIAM Journal on Computing* 1 (1972), Nr. 2, S. 131

[Ale] Aletheia - Semantische Föderation umfassender Produktinformationen. In: *Research Project* http://www.aletheia-projekt.de

[AM11] ANDREWS, Simon ; MCLEOD, Kenneth: Gene Co-Expression in Mouse Embryo Tissues. (2011)

[And09] ANDREWS, Simon: In-Close, a Fast Algorithm for Computing Formal Concepts. (2009)

[And11] ANDREWS, Simon: In-Close2, a High Performance Formal Concept Miner. (2011)

[AO10a] ANDREWS, Simon ; ORPHANIDES, Constantinos: FcaBedrock, a Formal Context Creator. (2010)

[AO10b] ANDREWS, Simon ; ORPHANIDES, Constantinos: Knowledge discovery through creating formal contexts. (2010)

[AP11] ANDREWS, Simon ; POLOVINA, Simon: A Mapping from Conceptual Graphs to Formal Concept Analysis. (2011)

[BC03] BECKER, Peter ; COLE, Richard: Querying and analysing document collections with Formal Concept Analysis. (2003)

[Bec] BECKER, Peter: Multi-dimensional Representations of Conceptual Hierarchies.

[BRB05] BESSON, Jérémy ; ROBARDET, Céline ; BOULICAUT, Jean-François: Mining Formal Concepts with a Bounded Number of Exceptions from Transactional Data. (2005), Nr. LNCS 3377

[BRB06] BESSON, Jérémy ; ROBARDET, Céline ; BOULICAUT, Jean-François: Mining a New Fault-Tolerant Pattern Type as an Alternative to Formal Concept Discovery. (2006), Nr. LNAI 4068

[BS05] BĚLOHLÁVEK, Radim ; SKLENÁŘ, Vladimír: Formal Concept Analysis Constrained by Attribute-Dependency Formulas. (2005), Nr. LNAI 3403

[BSVG09] BAIXERIES, Jaume ; SZATHMARY, Laszlo ; VALTCHEV, Petko ; GODIN, Robert: Yet a Faster Algorithm for Building the Hasse Diagram of a Concept Lattice. (2009)

[BSZ] BĚLOHLÁVEK, Radim ; SKLENÁŘ, Vladimír ; ZACPAL, Jiří: Concept lattices constrained by systems of partitions.

[BSZ04a] BĚLOHLÁVEK, Radim ; SKLENÁŘ, Vladimír ; ZACPAL, Jiří: Concept Lattices Constrained by Equivalence Relations. (2004)

[BSZ04b] BĚLOHLÁVEK, Radim ; SKLENÁŘ, Vladimír ; ZACPAL, Jiří: Formal concept analysis with hierarchically ordered attributes. In: *International Journal of General Systems* 33 (2004), Nr. 4, S. 383–394

[BT11] BALCAZAR, Jose L. ; TIRNAUCA, Cristina: Border Algorithms for Computing Hasse Diagrams of Arbitrary Lattices. (2011)

[BV06] BĚLOHLÁVEK, Radim ; VYCHODIL, Vilém: Formal Concept Analysis with Constraints by Closure Operators. (2006), Nr. LNAI 4068

[CDE06] COLE, Richard ; DUCROU, Jon ; EKLUND, Peter W.: Automated Layout of Small Lattices Using Layer Diagrams. (2006), Nr. LNAI 3874

[Cola] COLE, Richard: Automated Layout of Concept Lattice Using Layer Diagrams and Additive Diagrams.

[Colb] COLE, Richard: Automated Layout of Concept Lattices Using Force Directed Placement and Genetic Algorithms.

[Cub] Cubist - Your Business Intelligence. In: *Research Project* http://www.cubist-project.eu/

[Cui] CUI, Weiwei: A Survey on Graph Visualizations.

[Dau11] DAU, Frithjof: Graph Viz: State of the Art. (2011)

[Dau12] DAU, Frithjof: Towards Scalingless Generation of Formal Contexts from an Ontology in a Triple Store. (2012)

[DFT] DIATTA, Jean ; FENO, Daniel R. ; TOTOHASINA, A.: Galois Lattices and Bases for MGK-valid Association Rules.

[DP02] DAVEY, Brian A. ; PRIESTLEY, Hilary A.: *Lattices and Order*. 2. Cambridge University Press, 2002. – ISBN 9780521784511

[DS11] DAU, Frithjof ; SERTKAYA, Baris: An Extension of ToscanaJ for FCA-based Data Analysis over Triple Stores. (2011)

[Duq10] DUQUENNE, Vincent: Lattice Drawings and Morphisms. (2010)

[DV] DIAS, Sergio M. ; VIEIRA, Newton J.: Reducing the Size of Concept Lattices: The JBOS Approach.

[EDB04] EKLUND, Peter W. ; DUCROU, Jon ; BRAWN, Peter: Concept Lattices for Information Visualization: Can Novices Read Line-Diagrams? (2004), Nr. LNAI 2961

[EKMS] ERNÉ, M. ; KOSLOWSKI, J. ; MELTON, A. ; STRECKER, G. E.: A Primer on Galois Connections.

[EV10] EKLUND, Peter W. ; VILLERD, Jean: A Survey of Hybrid Representations of Concept Lattices in Conceptual Knowledge Processing. (2010)

[Fre04] FREESE, Ralph: Automated Lattice Drawing. (2004), Nr. LNAI 2961

[Ful55] FULKERSON, D. R.: Note on Dilworth's Decomposition Theorem for Partially Ordered Sets. (1955)

[Gan] GANTER, Bernhard: Conflict Avoidance in additive order diagrams.

[Gan07] GANTER, Bernhard: Relational Galois Connections. (2007), Nr. LNAI 4390

[GG12] GANTER, Bernhard ; GLODEANU, Cynthia V.: Ordinal factor analysis. (2012)

[GK01] GANTER, Bernhard ; KUZNETSOV, Sergei O.: Pattern Structures and Their Projections. (2001)

[GM94] GODIN, Robert ; MISSAOUI, Rokia: An incremental concept formation approach for learning from databases. (1994)

[GMA95] GODIN, Robert ; MISSAOUI, Rokia ; ALAOUI, Hassan: Incremental concept formation algorithms based on Galois (concept) lattices. (1995)

[Gom77] GOMBRICH, E. H.: *Art and Illusion: A Study in the Psychology of Pictorial Representation*. Fifth. Phaidon Press, 1977 http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0714842087. – ISBN 0714842087

[GR04] GODSIL, Chris ; ROYLE, Gordon: *Algebraic graph theory*. [Nachdr.]. New York and NY : Springer, 2004. – ISBN 9780387952208

[Grä06] GRÄTZER, George A.: *The congruences of a finite lattice: A proof-by-picture approach*. Boston and Basel : Birkhäuser, 2006. – ISBN 978–0–8176–3224–3

[Grä08] GRÄTZER, George A.: *Universal algebra*. 2. New York : Springer, 2008. – ISBN 0387774874

[Grä11] GRÄTZER, George A.: *Lattice theory: Foundation*. Basel and New York : Birkhauser and Springer Basel, 2011. – ISBN 3034800177

[GS02] GANTER, Bernhard ; STUMME, Gerd: Formal Concept Analysis: Methods and Applications in Computer Science. (2002)

[GW99] GANTER, Bernhard ; WILLE, Rudolf: *Formal concept analysis: Mathematical foundations*. Berlin : Springer-Verlag, 1999. – ISBN 9783540627715

[GYNS04] GASMI, Gh ; YAHIA, S. B. ; NGUIFO, E. M. ; SLIMANI, Y.: A new Informative Generic Base of Association Rules. (2004)

[HP06] HANNAN, Tim ; POGEL, Alex: Spring-Based Lattice Drawing Highlighting Conceptual Similarity. (2006), Nr. LNAI 3874

[HSWW00] HERETH, Joachim ; STUMME, Gerd ; WILLE, Rudolf ; WILLE, Uta: Conceptual Knowledge Discovery and Data Analysis. (2000), Nr. LNAI 1867

[Kei05] KEIM, Daniel: Scaling Visual Analytics to Very Large Data Sets. In: *Workshop on Visual Analytics* (2005)

[KM06] KRÖTZSCH, Markus ; MALIK, Grit: The Tensor Product as a Lattice of Regular Galois Connections. (2006), Nr. LNAI 3874

[KOR] KUZNETSOV, Sergei O. ; OBIEDKOV, Sergei ; ROTH, Camille: Reducing the Representation Complexity of Lattice-Based Taxonomies.

[Kuz] KUZNETSOV, Sergei O.: On Stability of a Formal Concept.

[LS05] LAKHAL, Lotfi ; STUMME, Gerd: Efficient Mining of Association Rules Based on Formal Concept Analysis. (2005), Nr. LNAI 3626

[Lux91] LUXENBURGER, Michael: Implications partielles dans un contexte. (1991)

[Mar79] MARKOWSKI, George: The Representation of Posets and Lattices by Sets. (1979)

[ME08] MARTIN, Ben ; EKLUND, Peter W.: From Concepts to Concept Lattice A Border Algorithm for Making Covers Explicit. (2008)

[Mes10] MESCHKE, Christian: Approximations in Concept Lattices. (2010)

[Mor07] MORI, Hideo: Functorial Properties of Formal Concept Analysis. (2007), Nr. LNAI 4604

[ORCR] OBIEDKOV, Sergei ; ROMASHKIN, Nikita ; COINTET, Jean-Philippe ; ROTH, Camille: Filtering lattice-based taxonomies with concept stability: A case study in the analysis of political texts.

[Out08] OUTRATA, Jan: Drawing lattices with a geometric heuristic. (2008)

[PB05] PENSA, Ruggero G. ; BOULICAUT, Jean-François: Towards Fault-Tolerant Formal Concept Analysis. (2005)

[Pre00] PREDIGER, Susanne: Formal Concept Analysis for General Objects. (2000)

[Pri] PRISS, Uta: Lattice-based Information Retrieval.

[PRSV] PERNELLE, Nathalie ; ROUSSET, Marie-Christine ; SOLDANO, Henry ; VENTOS, Véronique: ZooM: a nested Galois lattices-based system for conceptual clustering.

[Riv87] RIVAL, Ivan: Algorithms and Order. (1987)

[RL] RICORDEAU, Marc ; LIQUIERE, Michel: Policies Generalization in Reinforcement Learning using Galois Partitions Lattices.

[ROK] ROTH, Camille ; OBIEDKOV, Sergei ; KOURIE, Derrick G.: Towards Concise Representation for Taxonomies of Epistemic Communities.

[Ros04] ROSS, Elissa: Spectral Graph Drawing: A Survey. (2004)

[SAP] SAP AG. http://www.sap.com

[Sch] SCHULZ, Hans-Jörg: treevis.net - A Visual Bibliography of Tree Visualization 2.0 *beta*.

[Shn96] SHNEIDERMAN, Ben: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: *VL '96: Proceedings of the 1996 IEEE Symposium on Visual Languages* (1996), S. 336

[Sko] SKORSKY, Martin: How to draw a Concept Lattice with Parallelograms.

[STBL] STUMME, Gerd ; TAOUIL, Rafik ; BASTIDE, Yves ; LAKHAL, Lotfi: Conceptual Clustering with Iceberg Concept Lattices.

[Stu02] STUMME, Gerd: *Computing Iceberg Concept Lattices with TITANIC*. 2002

[Stu04] STUMME, Gerd: Iceberg Query Lattices for Datalog. (2004), Nr. LNAI 3127

[SV11] SOLDANO, Henry ; VENTOS, Véronique: Abstract Concept Lattices. (2011)

[SVCF] SOLDANO, Henry ; VENTOS, Véronique ; CHAMPESME, Marc ; FORGE, David: Incremental construction of Alpha lattices and association rules.

[SW] STUMME, Gerd ; WILLE, Rudolf: A geometrical heuristic for drawing concept lattices.

[Tam] TAMASSIA, Roberto (Hrsg.): *Handbook of Graph Drawing and Visualization*. CRC Press

[TPBL00] TAOUIL, Rafik ; PASQUIER, Nicolas ; BASTIDE, Yves ; LAKHAL, Lotfi: Mining bases for association rules using closed sets. (2000)

[TUD] Technische Universität Dresden, Institut für Algebra. http://tu-dresden.de/die_tu_dresden/fakultaeten/fakultaet_mathematik_und_naturwissenschaften/fachrichtung_mathematik/institute/algebra

[Ume] UMEYAMA, Shinji: An eigendecomposition approach to weighted graph matching problems.

[VS05] VENTOS, Véronique ; SOLDANO, Henry: Alpha Galois Lattices: An Overview. (2005), Nr. LNAI 3403

[VSL] VENTOS, Véronique ; SOLDANO, Henry ; LAMADON, Thibaut: Alpha Galois Lattices.

[Wat11] WATMOUGH, Martin: Evaluation of an Approach for Teaching Formal Concept Analysis. (2011)

[Wil] WILLE, Rudolf: Lattices in Data Analysis: How to draw them with a Computer.

[Wil92] WILD, Marcel: Cover-preserving order embeddings into Boolean lattices. (1992)

[Xie06] XIE, Zhipeng: Phi-Generalized Concept Lattice Models: From Power Concepts to Formal Concepts, and then to Robust Concepts. (2006)

[YKSJ07] YI, Ji S. ; KANG, Youn ah ; STASKO, John T. ; JACKO, Julie A.: Toward a Deeper Understanding of the Role of Interaction in Information Visualization. In: *IEEE Trans. Vis. Comput. Graph* 13 (2007), Nr. 6, 1224–1231. http://dblp.uni-trier.de/db/journals/tvcg/tvcg13.html#YiKSJ07

[YTK+] YEVTUSHENKO, Serhiy ; TANE, Julian ; KAISER, Tim B. ; OBIEDKOV, Sergei ; HERETH, Joachim ; REPPE, Heiko: ConExp - The Concept Explorer. http://conexp.sourceforge.net

[Zsc05] ZSCHALIG, Christian: Planarity of Lattices – An approach based on attribute additivity. In: *Lecture Notes in Computer Science* 3403 (2005), S. 391–402

[Zsc07] ZSCHALIG, Christian: An FDP-Algorithm for Drawing Lattices. In: *CEUR Workshop Proceedings* 331 (2007)

ERKLÄRUNG

Hiermit erkläre ich, dass ich die am heutigen Tag eingereichte Diplomarbeit zum Thema "Graphische Darstellung begrifflicher Daten mit Methoden der formalen Begriffsanalyse" unter Betreuung von Prof. Dr. Bernhard Ganter und Dr. Frithjof Dau selbstständig erarbeitet, verfasst und Zitate kenntlich gemacht habe. Andere als die angegebenen Hilfsmittel wurden von mir nicht benutzt.

Datum Unterschrift