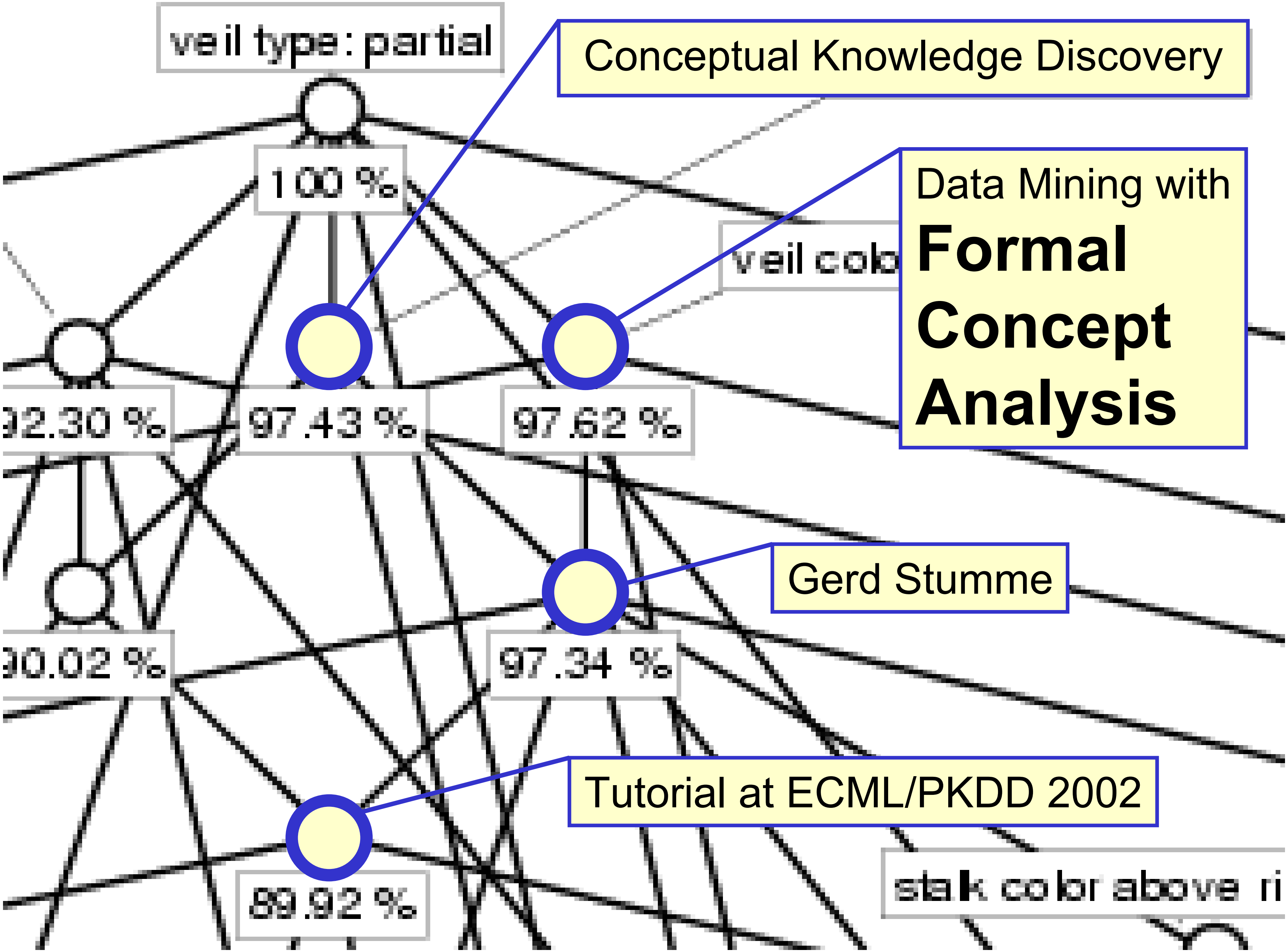


Conceptual Knowledge Discovery

Data Mining with

Formal Concept Analysis

Gerd Stumme

Tutorial at ECML/PKDD 2002

1. Introduction
2. Formal Contexts & Concept Lattices
3. Application Examples I
4. Computing Concept Lattices
5. Exercises

Coffee Break

6. Conceptual Scaling
7. Application Examples II
8. Conceptual Clustering
9. FCA-Based Mining of Association Rules
10. FCA Tools
11. Exercises

1. **Introduction**
2. Formal Contexts & Concept Lattices
3. Application Examples I
4. Computing Concept Lattices
5. Exercises
6. Conceptual Scaling
7. Application Examples II
8. Conceptual Clustering
9. FCA-Based Mining of Association Rules
10. FCA Tools
11. Exercises

Formal Concept Analysis

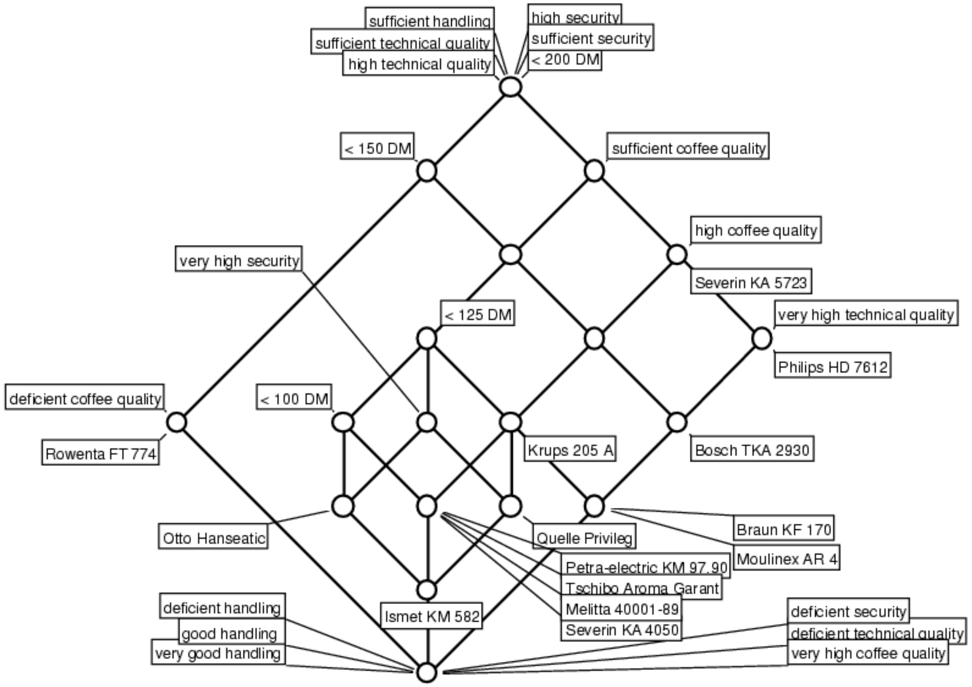
arose around 1980 in Darmstadt as a mathematical theory, which formalizes the concept of ‚concept‘.

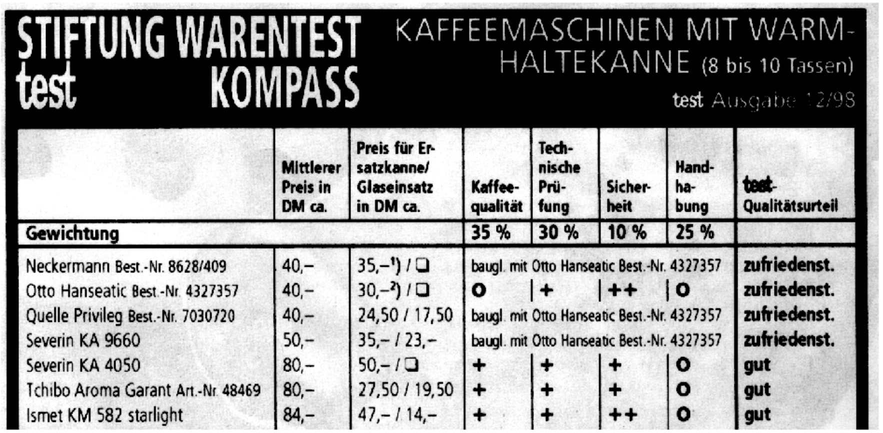
Since then, FCA has found many uses in Informatics, e.g. for

- Data Analysis,
- Knowledge Discovery,
- Software Engineering.

Based on datasets, FCA derives concept hierarchies.

FCA allows to generate and visualize the concept hierarchies.





STIFTUNG WARENTEST test KOMPASS

KAFFEEMASCHINEN MIT WARMHALTEKANNE (8 bis 10 Tassen)

test Ausgabe 12/98

	Mittlerer Preis in DM ca.	Preis für Ersatzkanne/ Glaseinsatz in DM ca.	Kaffeequalität	Technische Prüfung	Sicherheit	Handhabung	test-Qualitätsurteil
Gewichtung			35 %	30 %	10 %	25 %	
Neckermann Best.-Nr. 8628/409	40,–	35,–^{1}) / ❑	baugl. mit Otto Hanseatic Best.-Nr. 4327357				zufriedenst.
Otto Hanseatic Best.-Nr. 4327357	40,–	30,–^{2}) / ❑	o	+	++	o	zufriedenst.
Quelle Privileg Best.-Nr. 7030720	40,–	24,50 / 17,50	baugl. mit Otto Hanseatic Best.-Nr. 4327357				zufriedenst.
Severin KA 9660	50,–	35,– / 23,–	baugl. mit Otto Hanseatic Best.-Nr. 4327357				zufriedenst.
Severin KA 4050	80,–	50,– / ❑	+	+	+	o	gut
Tchibo Aroma Garant Art.-Nr. 48469	80,–	27,50 / 19,50	+	+	+	o	gut
Ismet KM 582 starlight	84,–	47,– / 14,–	+	+	++	o	gut

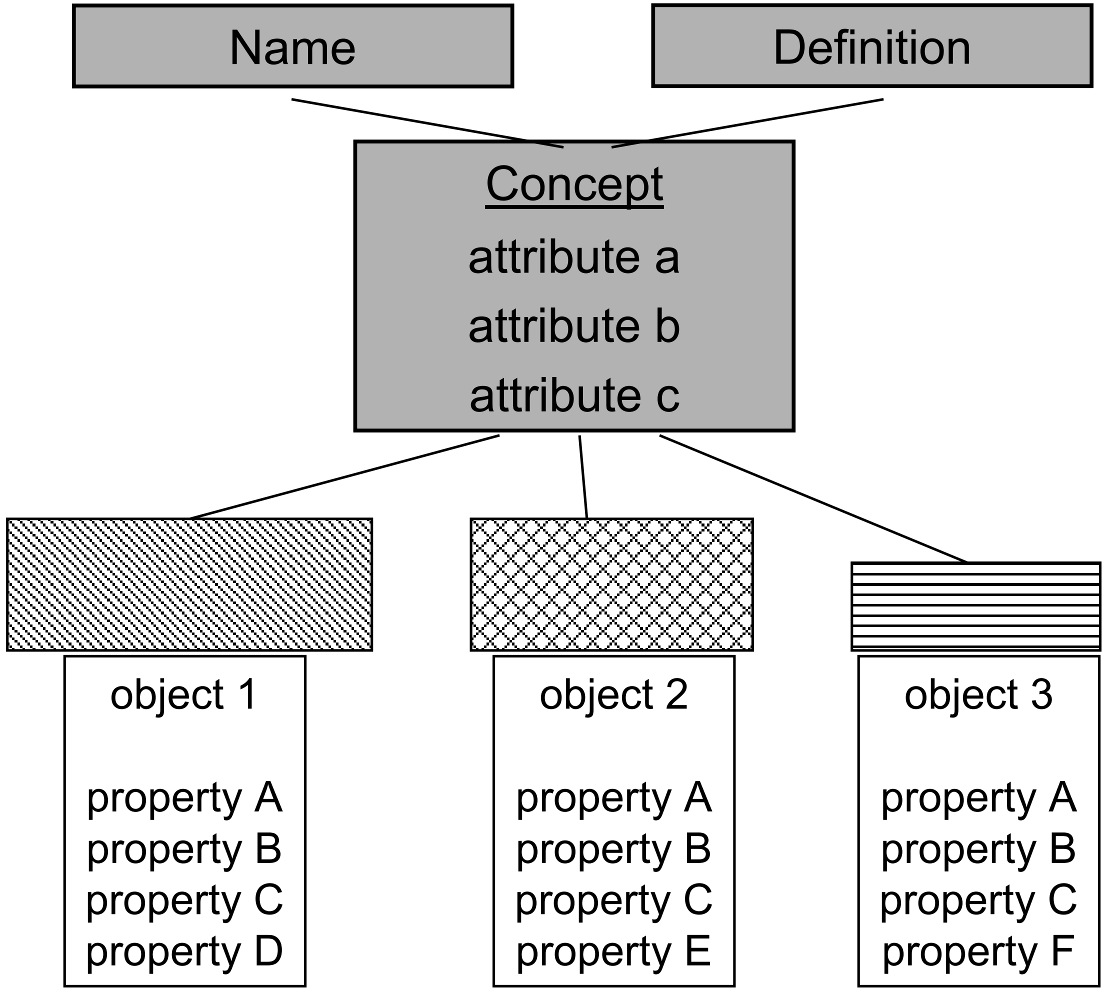
- FCA models **concepts** as **units of thought**, consisting of two parts:
  - The **extension** consists of all objects belonging to the concept.
  - The **intension** consists of all attributes common to all those objects.
- FCA is used for data analysis, information retrieval, and knowledge discovery.
- FCA can be understood as conceptual clustering method, which clusters simultanously objects and their descriptions.
- FCA can also be used for efficiently computing association rules.

ISO 704: Terminology Work: Principles and Methods

Representation level

Concept level

Object level



Name

Definition

Concept

attribute a

attribute b

attribute c

object 1

property A
property B
property C
property D

object 2

property A
property B
property C
property E

object 3

property A
property B
property C
property F

FCA focusses on the concept level, the representation level plays only a minor role.

Some **typical applications**:

- analysis of children suffering from diabetes
- IT security management system
- database marketing in a Suiss department store
- email management system
- developing qualitative theories in music estethics
- analysis of flight movements at Frankfurt airport

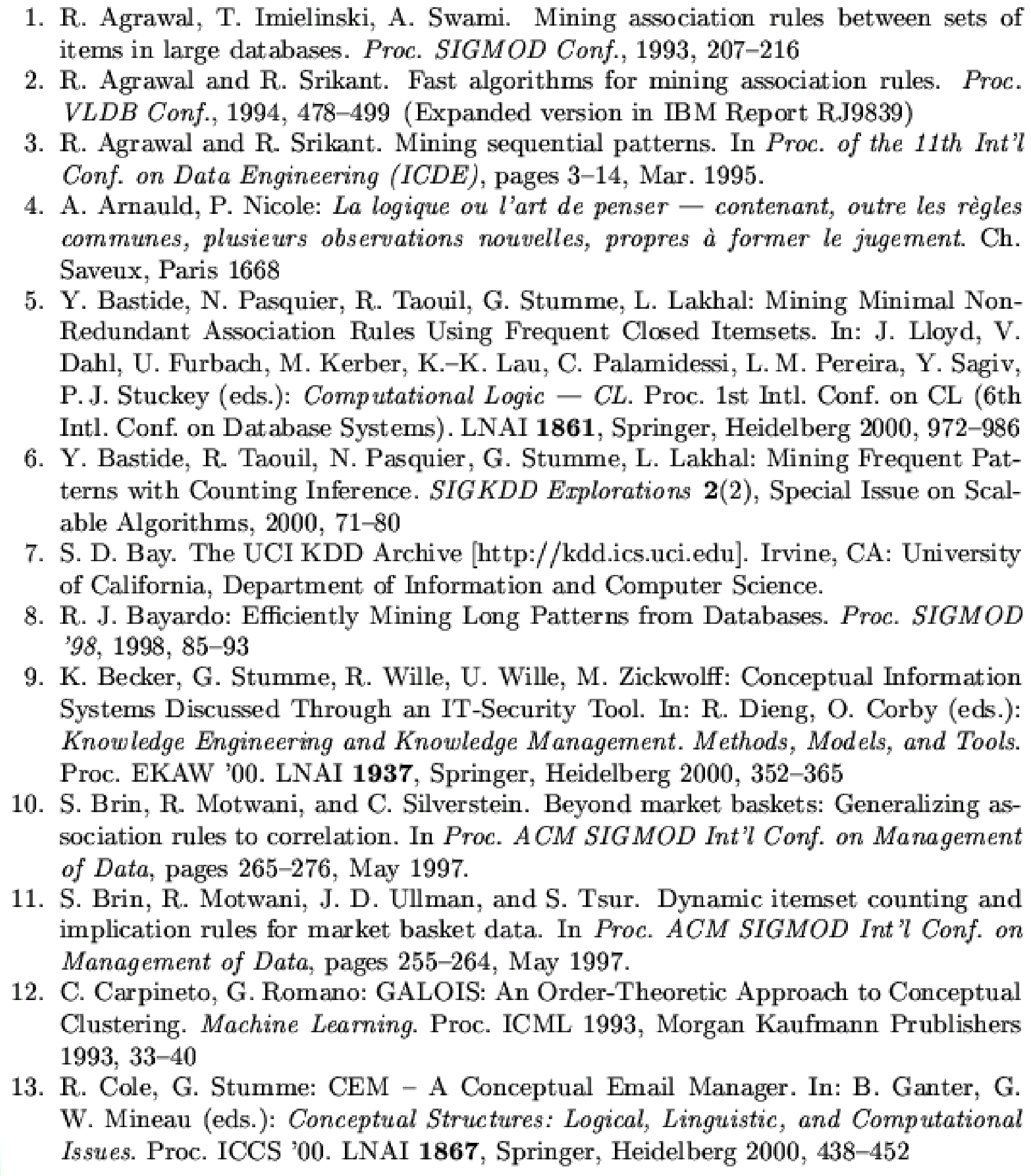
Links

- The Karlsruhe FCA page:
  km.aifb.uni-karlsruhe.de/fca
- FCA Mailing List:
  http://www.aifb.uni-karlsruhe.de/mailman/listinfo/fca-list
- The Darmstadt Research Group on FCA:
  www.mathematik.tu-darmstadt.de/ags/ag1/
- Research Center Conceptual Knowledge Processing, Darmstadt:
  http://www.fzbw.de/
- Ernst Schröder Center, Darmstadt:
  http://www.mathematik.tu-darmstadt.de/ags/esz/Welcome-en.html
- NaviCon GmbH:
  www.navicon.de
- The Dresden FCA page:
  http://www.math.tu-dresden.de/~ganter/fba.html
- Uta Priss‘ FCA page:
  http://php.indiana.edu/~upriss/fca/fca.html (will be moving)
- Michel Liquiere‘s FCA page:
  http://www.lattices.org/

Bibliography

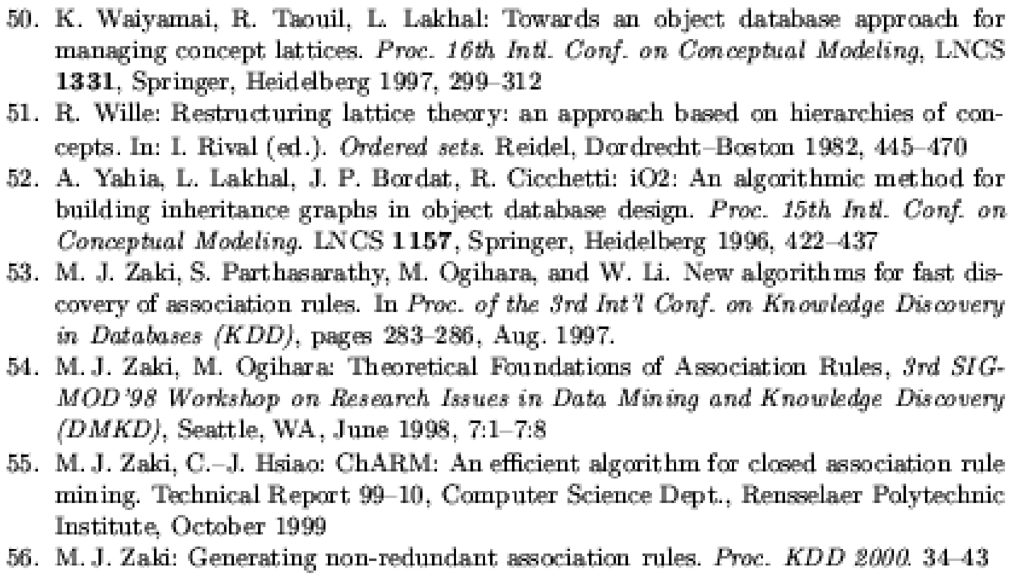
- R. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): Ordered sets. Reidel, Dordrecht-Boston, 1982, 445-470
- B. Ganter, R. Wille: *Formal Concept Analysis – Mathematical Foundations*. Springer, Heidelberg 1999 (Translation of: *Formale Begriffsanalyse – Mathematische Grundlagen*. Springer, Heidelberg 1996)
- G. Stumme, R. Wille (Eds.): Begriffliche Wissensverarbeitung - Methoden und Anwendungen. Springer 2000
- Proc. Intl. Conf. on Conceptual Structures (ICCS), Springer, Heidelberg (contain FCA topics since 1995)
- More books: http://www.math.tu-dresden.de/~ganter/FCAbooks.html
- List of publications: http://www.mathematik.tu-darmstadt.de/ags/ag1/Literatur/literatur_en.html

KDD/FCA specific bibliography



1. R. Agrawal, T. Imielinski, A. Swami. Mining association rules between sets of items in large databases. *Proc. SIGMOD Conf.*, 1993, 207–216
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. VLDB Conf.*, 1994, 478–499 (Expanded version in IBM Report RJ9839)
3. R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. of the 11th Int'l Conf. on Data Engineering (ICDE)*, pages 3–14, Mar. 1995.
4. A. Arnauld, P. Nicole: *La logique ou l'art de penser — contenant, outre les règles communes, plusieurs observations nouvelles, propres à former le jugement.* Ch. Saveux, Paris 1668
5. Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, L. Lakhal: Mining Minimal Non-Redundant Association Rules Using Frequent Closed Itemsets. In: J. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.–K. Lau, C. Palamidessi, L. M. Pereira, Y. Sagiv, P. J. Stuckey (eds.): *Computational Logic — CL.* Proc. 1st Intl. Conf. on CL (6th Intl. Conf. on Database Systems). LNAI **1861**, Springer, Heidelberg 2000, 972–986
6. Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, L. Lakhal: Mining Frequent Patterns with Counting Inference. *SIGKDD Explorations* **2**(2), Special Issue on Scalable Algorithms, 2000, 71–80
7. S. D. Bay. The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science.
8. R. J. Bayardo: Efficiently Mining Long Patterns from Databases. *Proc. SIGMOD '98*, 1998, 85–93
9. K. Becker, G. Stumme, R. Wille, U. Wille, M. Zickwolff: Conceptual Information Systems Discussed Through an IT-Security Tool. In: R. Dieng, O. Corby (eds.): *Knowledge Engineering and Knowledge Management. Methods, Models, and Tools.* Proc. EKAW '00. LNAI **1937**, Springer, Heidelberg 2000, 352–365
10. S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlation. In *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, pages 265–276, May 1997.
11. S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, pages 255–264, May 1997.
12. C. Carpineto, G. Romano: GALOIS: An Order-Theoretic Approach to Conceptual Clustering. *Machine Learning.* Proc. ICML 1993, Morgan Kaufmann Prublishers 1993, 33–40
13. R. Cole, G. Stumme: CEM – A Conceptual Email Manager. In: B. Ganter, G. W. Mineau (eds.): *Conceptual Structures: Logical, Linguistic, and Computational Issues.* Proc. ICCS '00. LNAI **1867**, Springer, Heidelberg 2000, 438–452

14. H. Dicky, C. Dony, M. Huchard, T Libourel: On automatic class insertion with overloading. *OOPSLA 1996*, 251–267
15. B. Ganter, R. Wille: *Formal Concept Analysis: Mathematical Foundations.* Springer, Heidelberg 1999
16. R. Godin, H. Mili, G. Mineau, R. Missaoui, A. Arfi, T. Chau: Design of class hierarchies based on concept (Galois) lattices. *TAPOS* **4**(2), 1998, 117–134
17. J. Han and M. Kamber. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, Sept. 2000.
18. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, pages 1–12, May 2000.
19. J. Hereth, G. Stumme, U. Wille, R. Wille: Conceptual Knowledge Discovery and Data Analysis. In: B. Ganter, G. Mineau (eds.): *Conceptual Structures: Logical, Linguistic, and Computational Structures.* Proc. ICCS 2000. LNAI **1867**, Springer, Heidelberg 2000, 421–437
20. Y. Huhtala, J. Kärkkäinen, P. Porkka, H. Toivonen: TANE: an efficient algorithm for discovering functional and approximate dependencies. *The Computer Journal* **42**(2), 1999, 100–111
21. M. Kamber, J. Han, and Y. Chiang. Metarule-guided mining of multi-dimensional association rules using data cubes. In *Proc. of the 3rd KDD Int'l Conf.*, Aug. 1997.
22. B. Lent, A. Swami, and J. Widom. Clustering association rules. In *Proc. of the 13th Int'l Conf. on Data Engineering (ICDE)*, pages 220–231, Mar. 1997.
23. D. Lin and Z. M. Kedem. Pincer-Search: A new algorithm for discovering the maximum frequent set. In *Proc. of the 6th Int'l Conf. on Extending Database Technology (EDBT)*, pages 105–119, Mar. 1998.
24. M. Luxenburger: Implications partielles dans un contexte. *Mathématiques, Informatique et Sciences Humaines* **29**(113), 1991, 35–55
25. K. Mackensen, U. Wille: Qualitative Text Analysis Supported by Conceptual Data Systems. *Quality and Quantity: Internatinal Journal of Methodology* **2**(33), 1999, 135–156
26. H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, Sept. 1997.
27. G. Mineau, G., R. Godin: Automatic Structuring of Knowledge Bases by Conceptual Clustering. *IEEE Transactions on Knowledge and Data Engineering* **7**(5), 1995, 824–829
28. M. Missikoff, M. Scholl: An algorithm for insertion into a lattice: application to type classification. *Proc. 3rd Intl. Conf. FODO 1989.* LNCS **367**, Springer, Heidelberg 1989, 64–82
29. J. S. Park, M. S. Chen, and P. S. Yu. An efficient hash based algorithm for mining association rules. In *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, pages 175–186, May 1995.
30. N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal: Pruning Closed Itemset Lattices for Association Rules. *14ièmes Journées Bases de Données Avancées (BDA'98)*, Hammamet, Tunisia, 26–30 October 1998
31. N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal: Efficient mining of association rules using closed itemset lattices. *Journal of Information Systems*, **24**(1), 1999, 25–46
32. N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal: Discovering frequent closed itemsets for association rules. *Proc. ICDT'99.* LNCS **1540**. Springer, Heidelberg 1999, 398–416
33. J. Pei, J. Han, R. Mao: CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets. *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery 2000*, 21–30
34. A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In *Proc. of the 21th Int'l Conf. on Very Large Data Bases (VLDB)*, pages 432–444, Sept. 1995.
35. P. Scheich, M. Skorsky, F. Vogt, C. Wachter, R. Wille: Conceptual Data Systems. In: O. Opitz, B. Lausen, R. Klar (eds.): *Information and Classification.* Springer, Berlin-Heidelberg 1993, 72–84
36. I. Schmitt, G. Saake: Merging inheritance hierarchies for database integration. *Proc. 3rd IFCIS Intl. Conf. on Cooperative Information Systems*, New York City, Nework, USA, August 20-22, 1998, 122–131
37. C. Silverstein, S. Brin, and R. Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2(1), Jan. 1998.
38. S. Strahringer, R. Wille: Conceptual clustering via convex-ordinal structures. In: O. Opitz, B. Lausen, R. Klar (eds.): *Information and Classification.* Springer, Berlin-Heidelberg 1993, 85–98
39. G. Stumme: *Conceptual Knowledge Discovery with Frequent Concept Lattices.* FB4-Preprint **2043**, TU Darmstadt 1999
40. G. Stumme, R. Taouil, Y. Bastide, L. Lakhal: Conceptual Clustering with Iceberg Concept Lattices. *Proc. GI–Fachgruppentreffen Maschinelles Lernen '01.* Universität Dortmund **763**, Oktober 2001
41. G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal: Fast computation of concept lattices using data mining techniques. *Proc. 7th Intl. Workshop on Knowledge Representation Meets Databases*, Berlin, 21–22. August 2000. CEUR-Workshop Proceeding. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/
42. G. Stumme, R. Taouil, Y. Bastide, N. Pasqier, L. Lakhal: Computing Iceberg Concept Lattices with Titanic. *J. on Knowledge and Data Engineering* **42**(2), 2002, 189–222
43. G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal: Intelligent Structuring and Reducing of Association Rules with Formal Concept Analysis. In: F. Baader. G. Brewker, T. Eiter (eds.): *KI 2001: Advances in Artificial Intelligence.* Proc. KI 2001. LNAI **2174**, Springer, Heidelberg 2001, 335–350
44. G. Stumme, R. Wille, U. Wille: Conceptual Knowledge Discovery in Databases Using Formal Concept Analysis Methods. In: J. M. Żytkow, M. Quafofou (eds.): *Principles of Data Mining and Knowledge Discovery.* Proc. 2nd European Symposium on PKDD'98, LNAI **1510**, Springer, Heidelberg 1998, 450–458
45. G. Stumme, R. Wille (eds.): *Begriffliche Wissensverarbeitung – Methoden und Anwendungen.* Springer, Heidelberg 2000
46. G. Stumme: Formal Concept Analysis on its Way from Mathematics to Computer Science. *Proc. 10th Intl. Conf. on Conceptual Structures (ICCS 2002).* LNCS, Springer, Heidelberg 2002
47. R. Taouil, N. Pasquier, Y. Bastide, L. Lakhal: Mining Bases for Association Rules Using Closed Sets. *Proc. 16th Intl. Conf. ICDE 2000*, San Diego, CA, US, February 2000, 307
48. H. Toivonen. Sampling large databases for association rules. In *Proc. of the 22nd Int'l Conf. on Very Large Data Bases (VLDB)*, pages 134–145, Sept. 1996.
49. F. Vogt, R. Wille: TOSCANA – A graphical tool for analyzing and exploring data. LNCS **894**, Springer, Heidelberg 1995, 226–233



50. K. Waiyamai, R. Taouil, L. Lakhal: Towards an object database approach for managing concept lattices. *Proc. 16th Intl. Conf. on Conceptual Modeling*, LNCS **1331**, Springer, Heidelberg 1997, 299–312

51. R. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.). *Ordered sets*. Reidel, Dordrecht–Boston 1982, 445–470

52. A. Yahia, L. Lakhal, J. P. Bordat, R. Cicchetti: iO2: An algorithmic method for building inheritance graphs in object database design. *Proc. 15th Intl. Conf. on Conceptual Modeling*. LNCS **1157**, Springer, Heidelberg 1996, 422–437

53. M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In *Proc. of the 3rd Int'l Conf. on Knowledge Discovery in Databases (KDD)*, pages 283–286, Aug. 1997.

54. M. J. Zaki, M. Ogihara: Theoretical Foundations of Association Rules, *3rd SIGMOD'98 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*, Seattle, WA, June 1998, 7:1–7:8

55. M. J. Zaki, C.–J. Hsiao: ChARM: An efficient algorithm for closed association rule mining. Technical Report 99–10, Computer Science Dept., Rensselaer Polytechnic Institute, October 1999

56. M. J. Zaki: Generating non-redundant association rules. *Proc. KDD 2000*. 34–43

These references are taken from

G. Stumme: Efficient Data Mining Based on Formal Concept Analysis. *Proc. 13th Intl. Conf. on Database and Expert Systems Applications (DEXA 2002).* LNCS, Springer, Heidelberg 2002 (Invited Talk, in press)

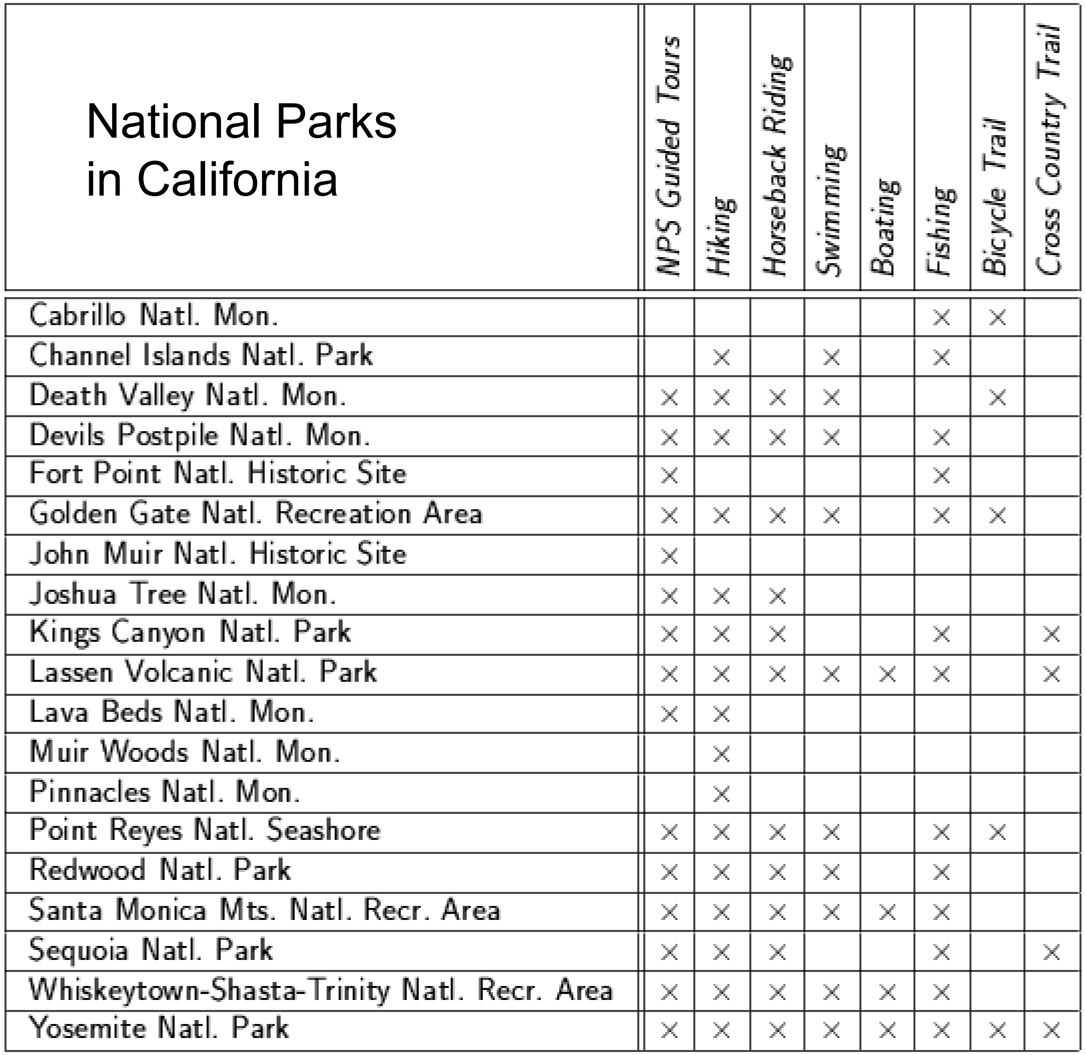
1. Introduction
2. **Formal Contexts & Concept Lattices**
3. Application Examples I
4. Computing Concept Lattices
5. Exercises
6. Conceptual Scaling
7. Application Examples II
8. Conceptual Clustering
9. FCA-Based Mining of Association Rules
10. FCA Tools
11. Exercises

Formal Concept Analysis

Def.: A **formal context** is a tripel (*G*,*M*,*I*), where

- *G* is a set of objects,
- *M* is a set of attributes
- and *I* is a relation between *G* and *M*.

- (*g*,*m*)∈*I* is read as „object *g* has attribute *m*“.



National Parks in California	NPS Guided Tours	Hiking	Horseback Riding	Swimming	Boating	Fishing	Bicycle Trail	Cross Country Trail
Cabrillo Natl. Mon.						×	×	
Channel Islands Natl. Park		×		×		×		
Death Valley Natl. Mon.	×	×	×	×			×	
Devils Postpile Natl. Mon.	×	×	×	×		×		
Fort Point Natl. Historic Site	×					×		
Golden Gate Natl. Recreation Area	×	×	×	×		×	×	
John Muir Natl. Historic Site	×							
Joshua Tree Natl. Mon.	×	×	×					
Kings Canyon Natl. Park	×	×	×			×		×
Lassen Volcanic Natl. Park	×	×	×	×	×	×		×
Lava Beds Natl. Mon.	×	×						
Muir Woods Natl. Mon.		×						
Pinnacles Natl. Mon.		×						
Point Reyes Natl. Seashore	×	×	×	×		×	×	
Redwood Natl. Park	×	×	×	×		×		
Santa Monica Mts. Natl. Recr. Area	×	×	×	×	×	×		
Sequoia Natl. Park	×	×	×			×		×
Whiskeytown-Shasta-Trinity Natl. Recr. Area	×	×	×	×	×	×		
Yosemite Natl. Park	×	×	×	×	×	×	×	×

For $A \subseteq G$, we define

$A' := \{ m \in M \mid \forall g \in A: (g,m) \in I \}$.

For $B \subseteq M$, we define dually

$B' := \{ g \in G \mid \forall m \in B: (g,m) \in I \}$.

A´

National Parks in California	NPS Guided Tours	Hiking	Horseback Riding	Swimming	Boating	Fishing	Bicycle Trail	Cross Country Trail
Cabrillo Natl. Mon.						×	×	
Channel Islands Natl. Park		×		×		×		
Death Valley Natl. Mon.	×	×	×	×			×	
Devils Postpile Natl. Mon.	×	×	×	×		×		
Fort Point Natl. Historic Site	×					×		
Golden Gate Natl. Recreation Area	×	×	×	×		×	×	
John Muir Natl. Historic Site	×							
Joshua Tree Natl. Mon.	×	×	×					
Kings Canyon Natl. Park	×	×	×			×		×
Lassen Volcanic Natl. Park	×	×	×	×	×	×		×
Lava Beds Natl. Mon.	×	×						
Muir Woods Natl. Mon.		×						
Pinnacles Natl. Mon.		×						
Point Reyes Natl. Seashore	×	×	×	×		×	×	
Redwood Natl. Park	×	×	×	×		×		
Santa Monica Mts. Natl. Recr. Area	×	×	×	×	×	×		
Sequoia Natl. Park	×	×	×			×		×
Whiskeytown-Shasta-Trinity Natl. Recr. Area	×	×	×	×	×	×		
Yosemite Natl. Park	×	×	×	×	×	×	×	×

A

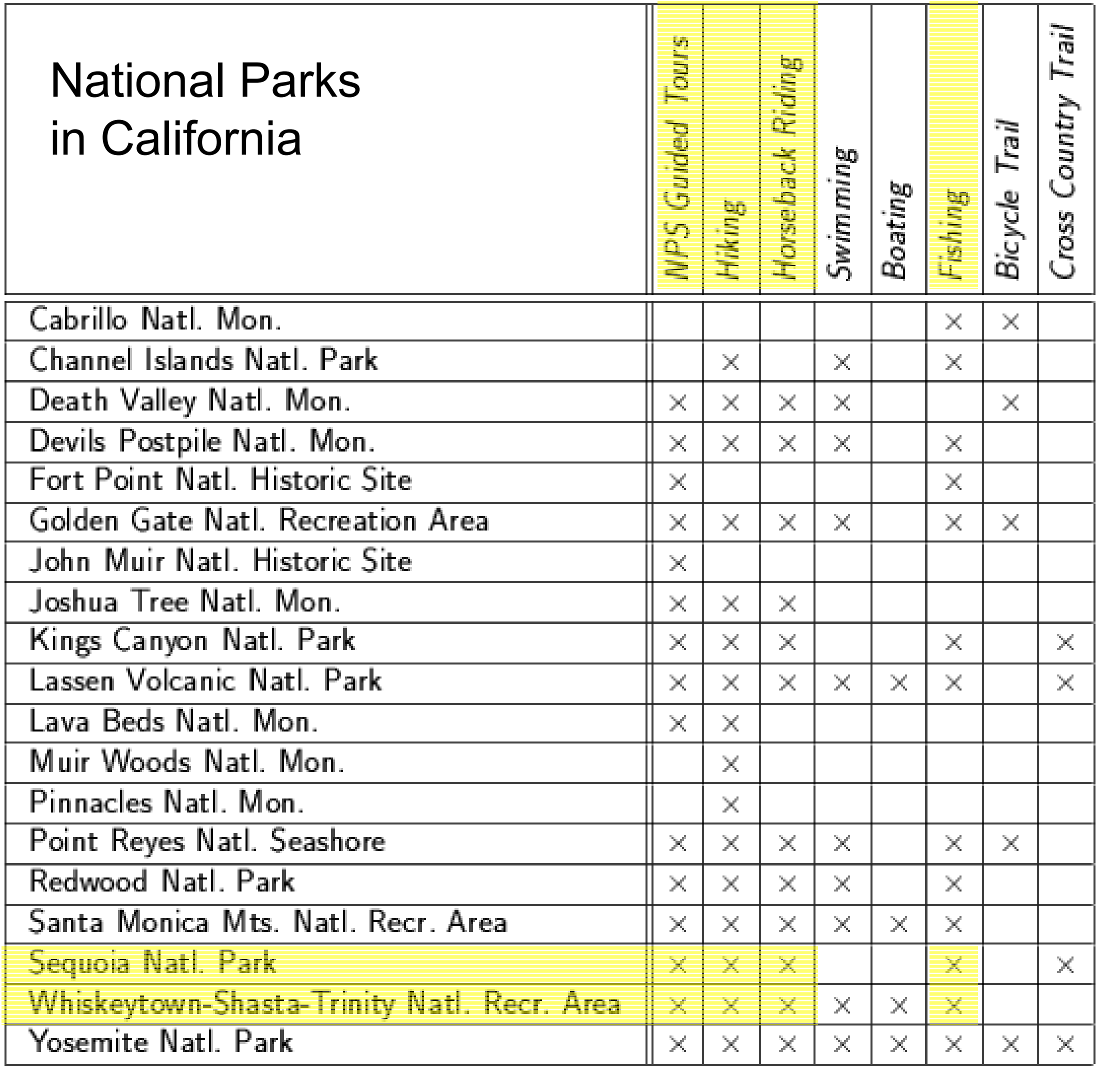
For A, A_1, $A_2 \subseteq G$ holds:

- $A_1 \subseteq A_2 \;\Rightarrow\; A'_2 \subseteq A'_1$
- $A \subseteq A''$
- $A' = A'''$

For B, B_1, $B_2 \subseteq M$ holds:

- $B_1 \subseteq B_2 \;\Rightarrow\; B'_2 \subseteq B'_1$
- $B \subseteq B''$
- $B' = B'''$

A´



National Parks in California	NPS Guided Tours	Hiking	Horseback Riding	Swimming	Boating	Fishing	Bicycle Trail	Cross Country Trail
Cabrillo Natl. Mon.						×	×	
Channel Islands Natl. Park		×		×		×		
Death Valley Natl. Mon.	×	×	×	×			×	
Devils Postpile Natl. Mon.	×	×	×	×		×		
Fort Point Natl. Historic Site	×					×		
Golden Gate Natl. Recreation Area	×	×	×	×		×	×	
John Muir Natl. Historic Site	×							
Joshua Tree Natl. Mon.	×	×	×					
Kings Canyon Natl. Park	×	×	×			×		×
Lassen Volcanic Natl. Park	×	×	×	×	×	×		×
Lava Beds Natl. Mon.	×	×						
Muir Woods Natl. Mon.		×						
Pinnacles Natl. Mon.		×						
Point Reyes Natl. Seashore	×	×	×	×		×	×	
Redwood Natl. Park	×	×	×	×		×		
Santa Monica Mts. Natl. Recr. Area	×	×	×	×	×	×		
Sequoia Natl. Park	×	×	×			×		×
Whiskeytown-Shasta-Trinity Natl. Recr. Area	×	×	×	×	×	×		
Yosemite Natl. Park	×	×	×	×	×	×	×	×

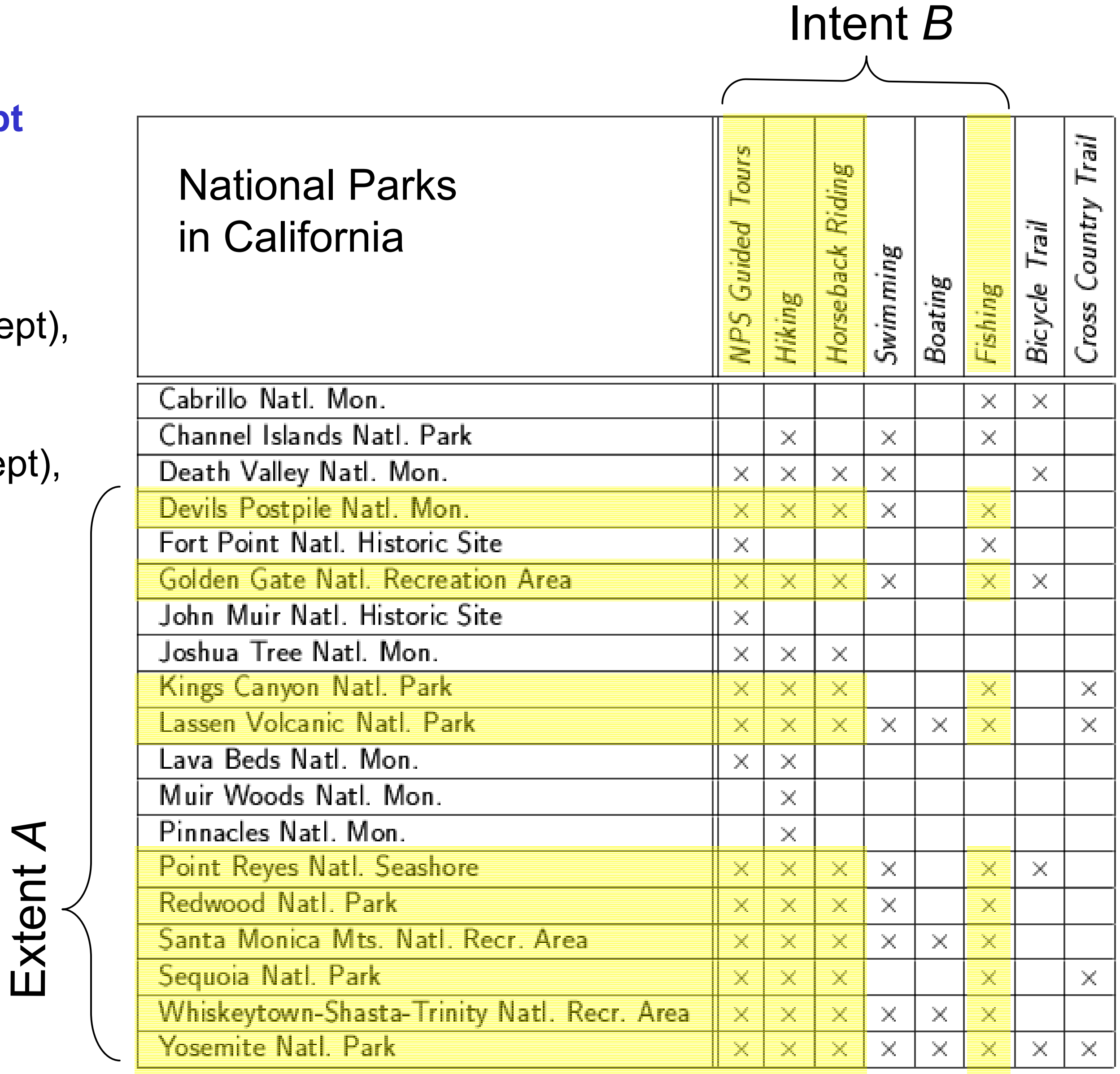
A

Def.: A **formal concept**

is a pair (*A*,*B*) where

- *A* is a set of objects (the **extent** of the concept),
- *B* is a set of attributes (the **intent** of the concept),
- A‘ = B and B‘ = A.

The last condition is equivalent *to* *A*×*B* being a maximal rectangle in the binary relation (i.e., *A* and *B* are maximal with *A*×*B* ⊆ *I*).

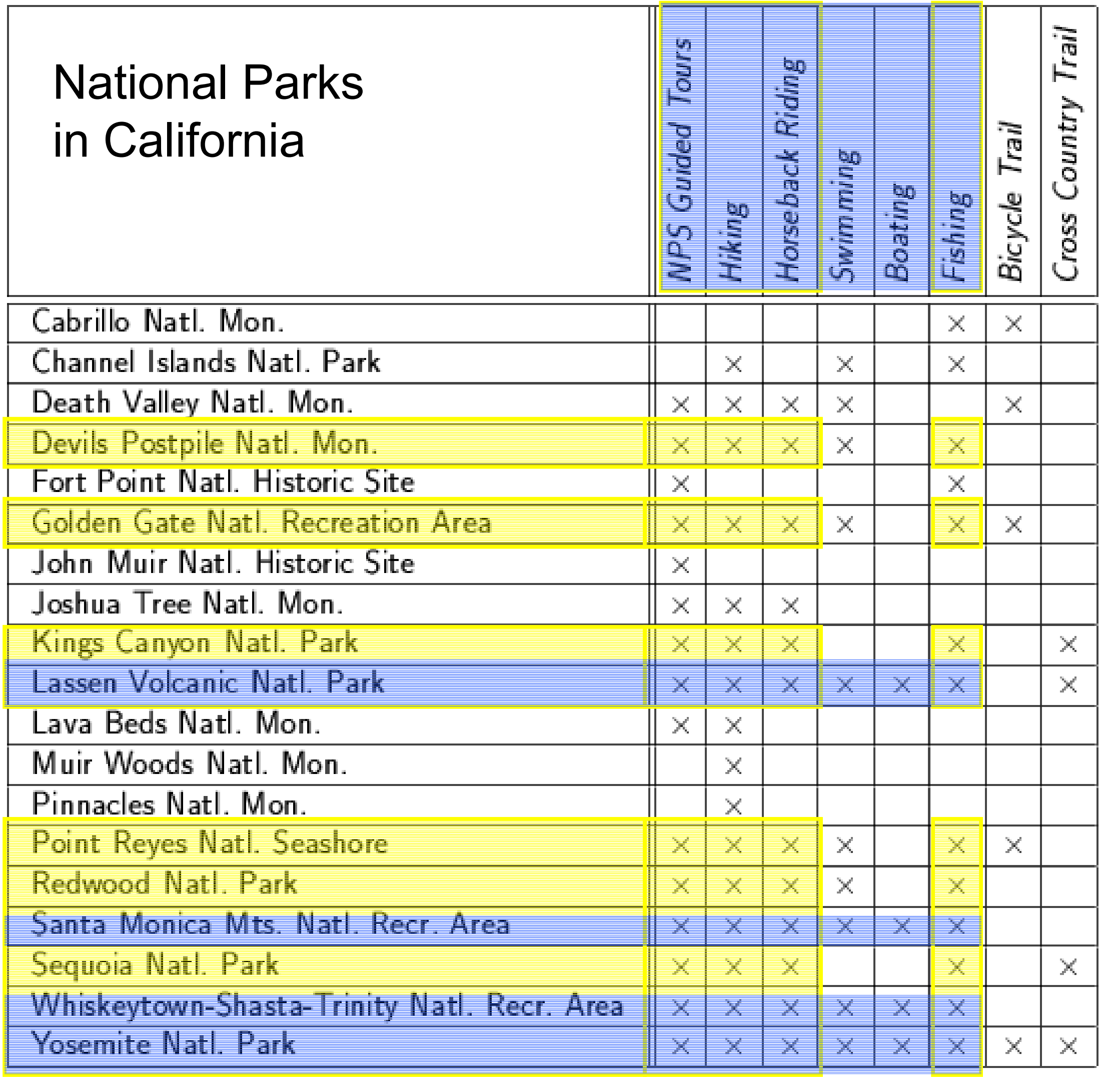

Intent *B*
Extent *A*

National Parks in California	NPS Guided Tours	Hiking	Horseback Riding	Swimming	Boating	Fishing	Bicycle Trail	Cross Country Trail
Cabrillo Natl. Mon.						×	×	
Channel Islands Natl. Park		×		×		×		
Death Valley Natl. Mon.	×	×	×	×			×	
Devils Postpile Natl. Mon.	×	×	×	×		×		
Fort Point Natl. Historic Site	×					×		
Golden Gate Natl. Recreation Area	×	×	×	×		×	×	
John Muir Natl. Historic Site	×							
Joshua Tree Natl. Mon.	×	×	×					
Kings Canyon Natl. Park	×	×	×			×		×
Lassen Volcanic Natl. Park	×	×	×	×	×	×		×
Lava Beds Natl. Mon.	×	×						
Muir Woods Natl. Mon.		×						
Pinnacles Natl. Mon.		×						
Point Reyes Natl. Seashore	×	×	×	×		×	×	
Redwood Natl. Park	×	×	×	×		×		
Santa Monica Mts. Natl. Recr. Area	×	×	×	×	×	×		
Sequoia Natl. Park	×	×	×			×		×
Whiskeytown-Shasta-Trinity Natl. Recr. Area	×	×	×	×	×	×		
Yosemite Natl. Park	×	×	×	×	×	×	×	×

Tutorial Formal Concept Analysis

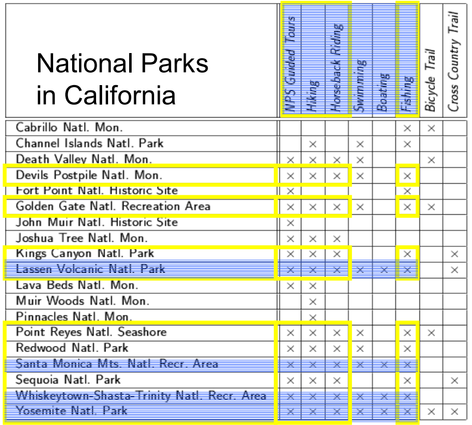
The blue concept is a **subconcept** of the yellow one, since its extent is contained in the yellow one.

(⇔ the yellow intent is contained in the blue one.)

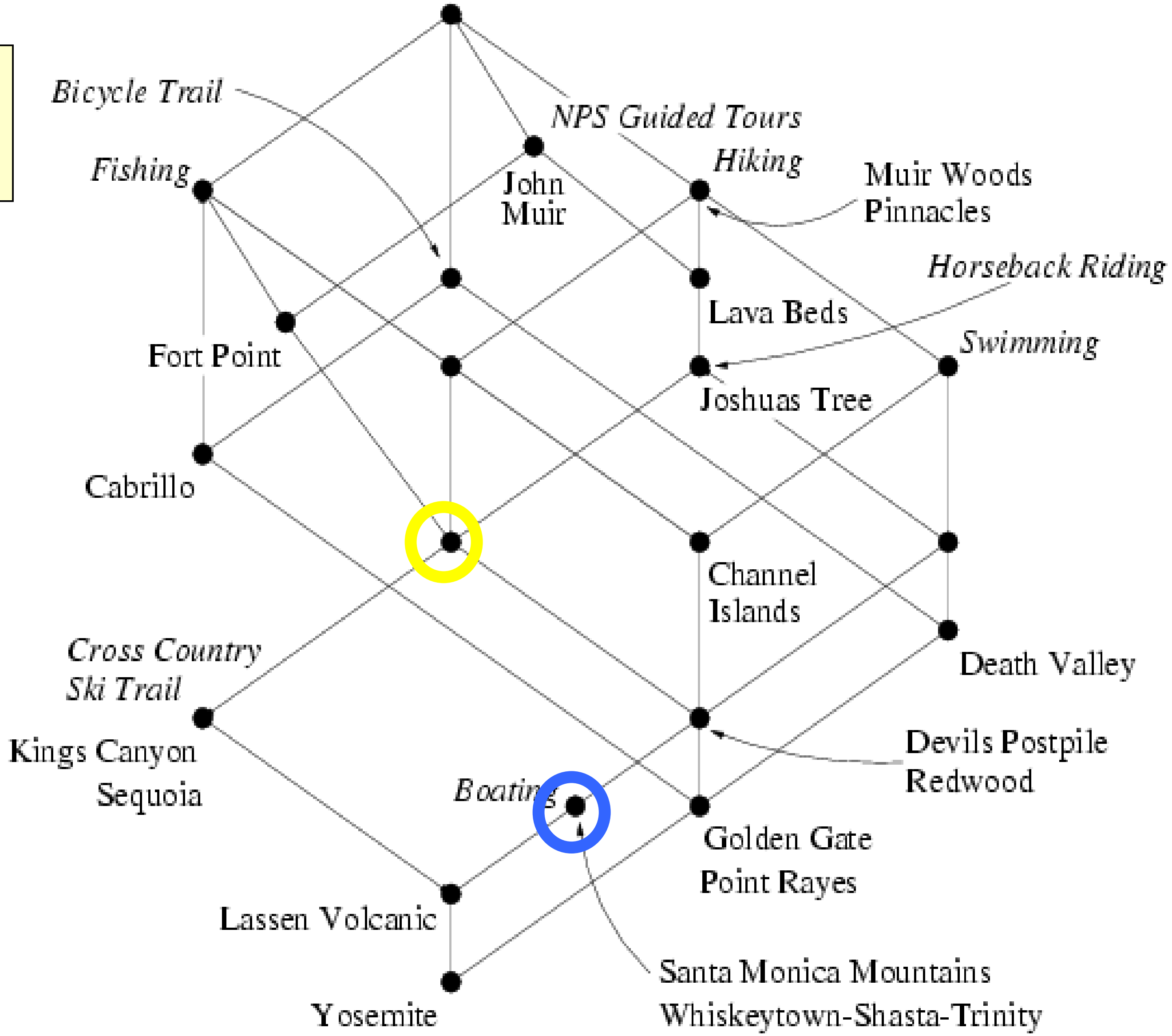


National Parks in California	NPS Guided Tours	Hiking	Horseback Riding	Swimming	Boating	Fishing	Bicycle Trail	Cross Country Trail
Cabrillo Natl. Mon.						×	×	
Channel Islands Natl. Park		×		×		×		
Death Valley Natl. Mon.	×	×	×	×			×	
Devils Postpile Natl. Mon.	×	×	×	×		×		
Fort Point Natl. Historic Site	×					×		
Golden Gate Natl. Recreation Area	×	×	×	×		×	×	
John Muir Natl. Historic Site	×							
Joshua Tree Natl. Mon.	×	×	×					
Kings Canyon Natl. Park	×	×	×			×		×
Lassen Volcanic Natl. Park	×	×	×	×	×	×		×
Lava Beds Natl. Mon.	×	×						
Muir Woods Natl. Mon.		×						
Pinnacles Natl. Mon.		×						
Point Reyes Natl. Seashore	×	×	×	×		×	×	
Redwood Natl. Park	×	×	×	×		×		
Santa Monica Mts. Natl. Recr. Area	×	×	×	×	×	×		
Sequoia Natl. Park	×	×	×			×		×
Whiskeytown-Shasta-Trinity Natl. Recr. Area	×	×	×	×	×	×		
Yosemite Natl. Park	×	×	×	×	×	×	×	×

The **concept lattice** of the National Parks in California



National Parks in California	NPS Guided Tours	Hiking	Horseback Riding	Swimming	Boating	Fishing	Bicycle Trail	Cross Country Trail
Cabrillo Natl. Mon.						×	×	
Channel Islands Natl. Park		×		×		×		
Death Valley Natl. Mon.	×	×	×	×			×	
Devils Postpile Natl. Mon.	×	×	×	×		×		
Fort Point Natl. Historic Site	×					×		
Golden Gate Natl. Recreation Area	×	×	×	×		×	×	
John Muir Natl. Historic Site	×							
Joshua Tree Natl. Mon.	×	×	×					
Kings Canyon Natl. Park	×	×	×			×		×
Lassen Volcanic Natl. Park	×	×	×	×	×	×		×
Lava Beds Natl. Mon.	×	×						
Muir Woods Natl. Mon.		×						
Pinnacles Natl. Mon.		×						
Point Reyes Natl. Seashore	×	×	×	×		×	×	
Redwood Natl. Park	×	×	×	×		×		
Santa Monica Mts. Natl. Recr. Area	×	×	×	×	×	×		
Sequoia Natl. Park	×	×	×			×		×
Whiskeytown-Shasta-Trinity Natl. Recr. Area	×	×	×	×	×	×		
Yosemite Natl. Park	×	×	×	×	×	×	×	×



- **Def.:** The **concept lattice** [**Begriffsverband**] of a formal context (G,M,I) is the set of all formal concepts of (G,M,I), together with the partial order

$$(A_1,B_1) \leq (A_2,B_2) :\Leftrightarrow A_1 \subseteq A_2 \quad (\Leftrightarrow B_1 \supseteq B_2)\ .$$

The concept lattice is denoted by $\underline{\mathfrak{B}}(G,M,I)$.

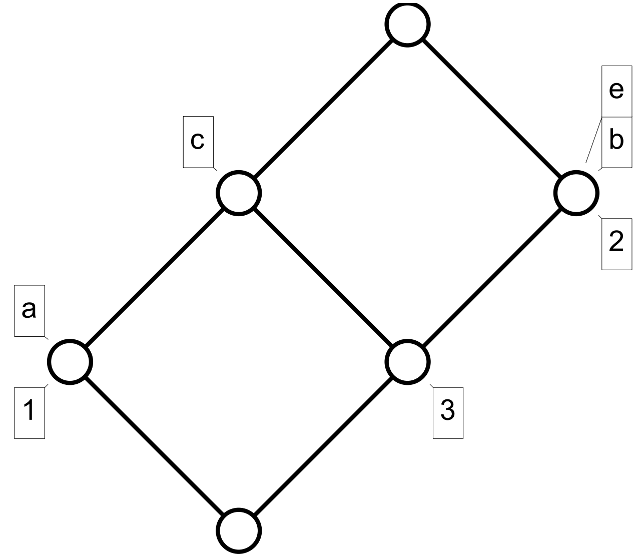
- **Theorem:** The concept lattice is a lattice, i.e. for two concepts (A_1,B_1) and (A_2,B_2), there is always
  - a greatest common subconcept: $(A_1 \cap A_2\ , (B_1 \cup B_2)'')$
  - and a least common superconcept: $((A_1 \cup A_2)'', B_1 \cap B_2)$.

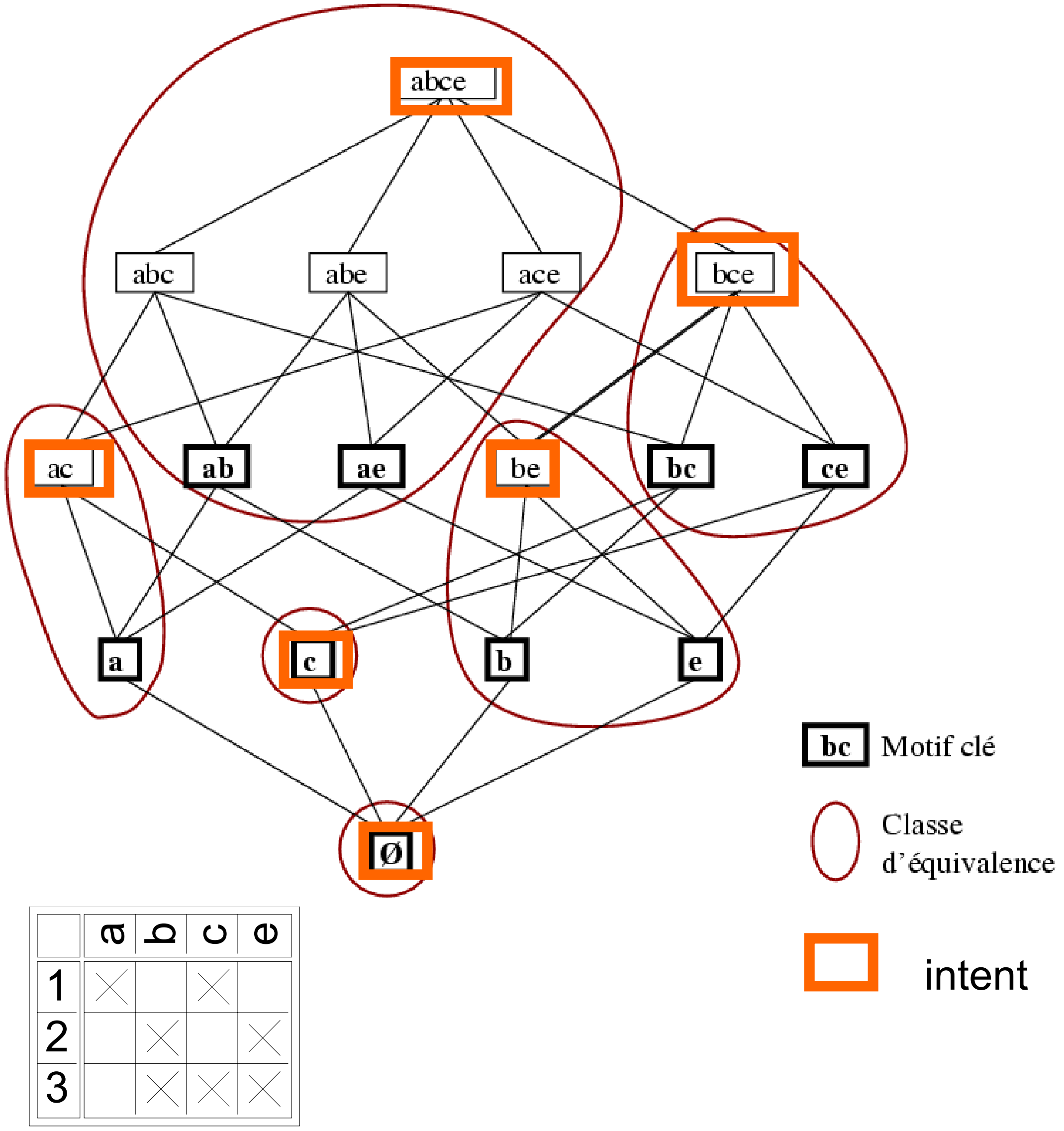
More general, it is even a complete lattice, i.e. the greatest common subconcept and the least common superconcept exist for all (finite and infinite) sets of concepts.

Corollary: The set of all concept intents of a formal context is a closure system. The corresponding closure operator is $h(X) := X$''.

In the power set of M, the concept intents are always the largest sets among those with the same closure.

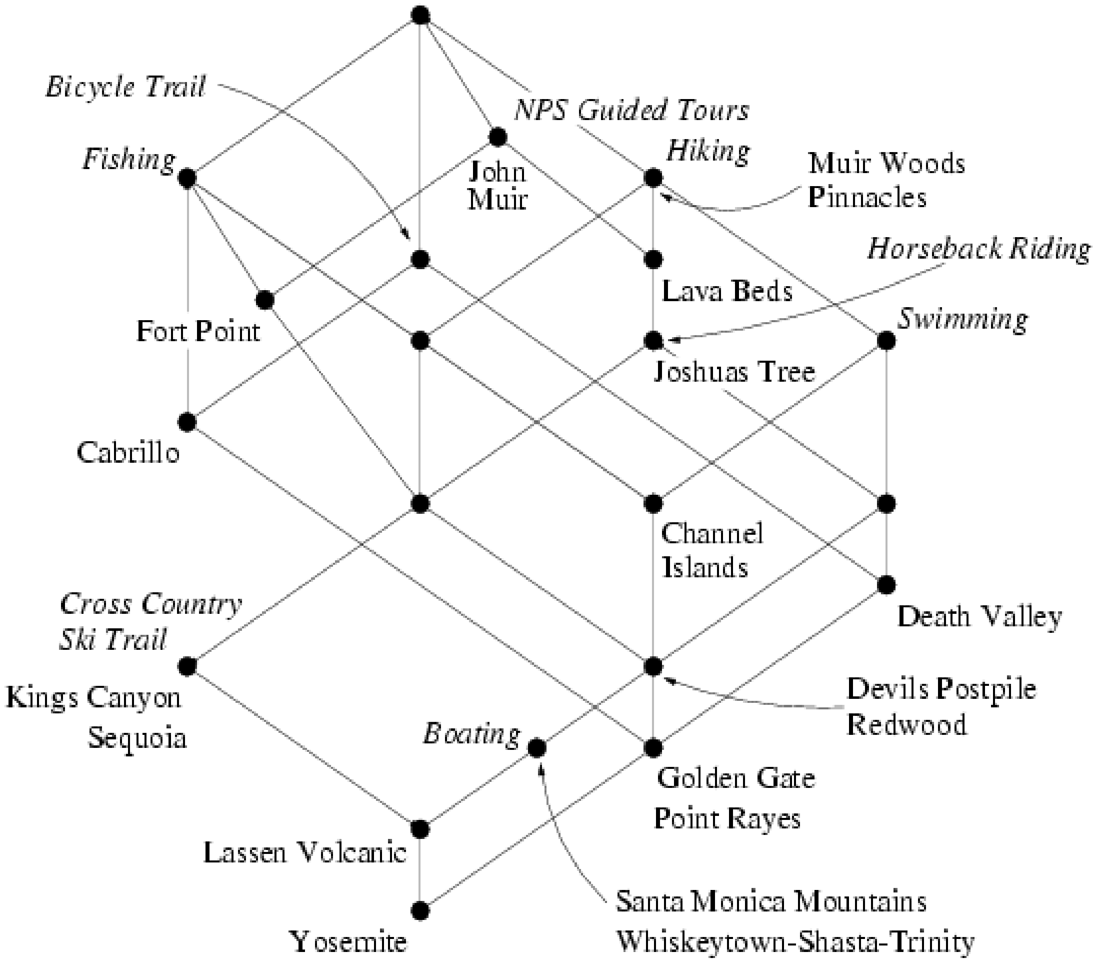
Example: h({a,b}) = h({a,b,c}) = h({a,b,c,e}) = {a,b,c,e}





	a	b	c	e
1	×		×	
2		×		×
3		×	×	×

Implications



Def.: An **implication** $X \to Y$ holds in a context, if every object having all attributes in X also has all attributes in Y.

- **Examples:**

Swimming → Hiking

Boating → Swimming, Hiking, NPS Guided Tours, Fishing

Bicycle Trail, NPS Guided Tours → Swimming, Hiking

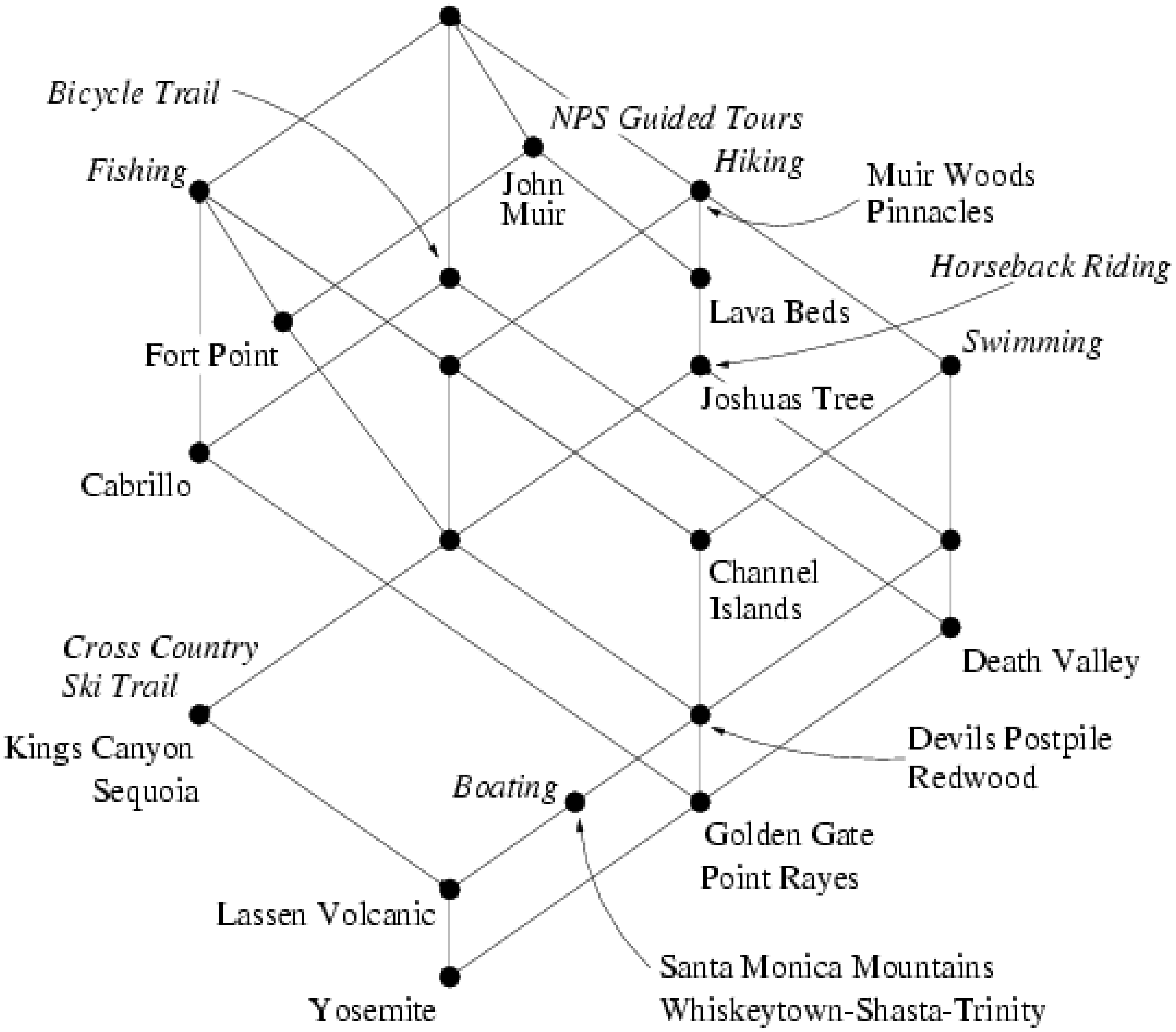
Independency

Def.: Let $X \subseteq M$. The attributes in X are **independent**, if there are no trivial dependencies between them.

Example:

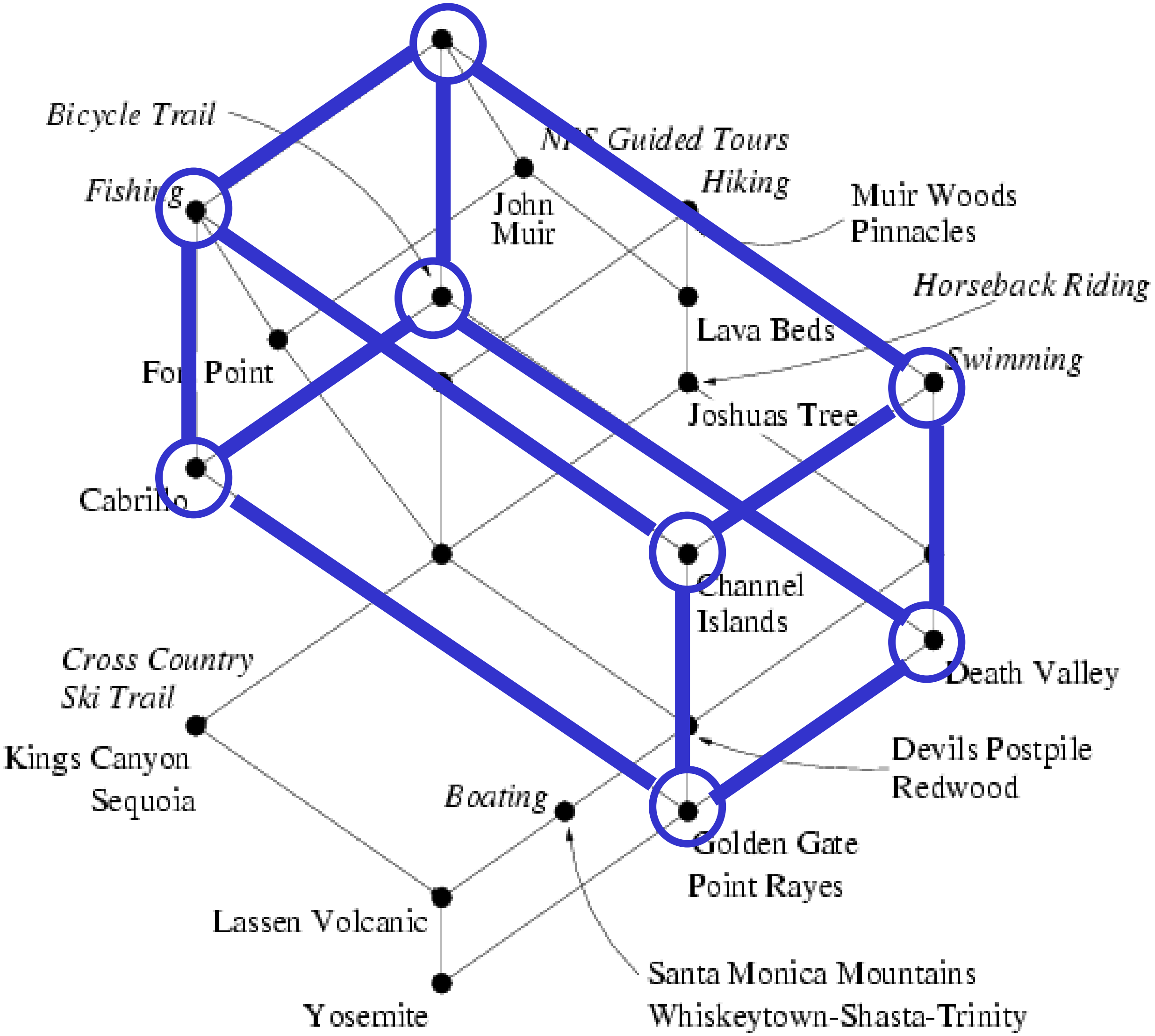
- Fishing
- Bicycle Trail
- Swimming

are independent attributes.



Independency

Lemma: Attributes are independent if they span a hyper-cube.



Example:

- Fishing
- Bicycle Trail
- Swimming

are independent attributes.

Concept Intents and Implications

Def.: A subset $T \subseteq M$ **respects** an implication $A \to B$, if $A \not\subseteq T$ or $B \subseteq T$. /

T **respects a set** $\mathcal{L}$ of implications, if T respects every single implication in $\mathcal{L}$.

Lemma: An implication $A \to B$ holds in a context iff $B \subseteq A''$. It is then respected by all concept intents.

Lemma: Is $\mathcal{L}$ a set of implications in M, then

$$\mathcal{H}(\mathcal{L}) := \{ X \subseteq M \mid X \text{ respects } \mathcal{L} \}$$

is a closure system.

The related closure operator is constructed as follows:
For a set $X \subseteq M$ let

$$X^{\mathcal{L}} := X \cup \bigcup \{ B \mid A \to B \in \mathcal{L}, A \subseteq X \}.$$

Compute $X^{\mathcal{L}}$, $X^{\mathcal{L}\mathcal{L}}$, $X^{\mathcal{L}\mathcal{L}\mathcal{L}}$,..., until a set

$$\mathcal{L}(X) := X^{\mathcal{L}\dots\mathcal{L}}$$

with $\mathcal{L}(X)^{\mathcal{L}} = \mathcal{L}(X)$ (i.e., a fix point) is reached. (for infinite contexts this may be an infinite process). $\mathcal{L}(X)$ ist then the closure of X with respect to the closure system $\mathcal{H}(\mathcal{L})$.

Def.: An implication $A \to B$ **is (semantically) entailed** from a set $\mathcal{L}$ of implications, if every subset of M respecting $\mathcal{L}$ also respects $A \to B$.

A family $\mathcal{L}$ of implications ist called **closed** if every implication entailed from $\mathcal{L}$ is already contained in $\mathcal{L}$.

Lemma: A set $\mathcal{L}$ of implications on M is closed iff the following conditions (Amstrong rules) are fulfilled for all $W, X, Y, Z \subseteq M$:

1. $X \to X \in \mathcal{L}$,
2. If $X \to \mathrm{Y} \in \mathcal{L}$, then $X \cup Z \to Y \in \mathcal{L}$,
3. If $X \to Y \in \mathcal{L}$ and $Y \cup Z \to W \in \mathcal{L}$, then $X \cup Z \to W \in \mathcal{L}$.

Def.: A set $\mathcal{L}$ of implications of a context (G, M, I) is called **complete**, if every implication of (G, M, I) is entailed from $\mathcal{L}$.
A set $\mathcal{L}$ of implications is called **non-redundant**, if no implication is entailed from the others.

Def.: $P \subseteq M$ is called **pseudo intent** of (G, M, I) if $P \neq P''$ and for every pseudo intent $Q \subseteq P$ with $Q \neq P$ holds $Q'' \subseteq P$.

Theorem: The set of implications

$$\mathcal{L} := \{ P \rightarrow P'' \mid P \text{ Pseudoinhalt} \}$$

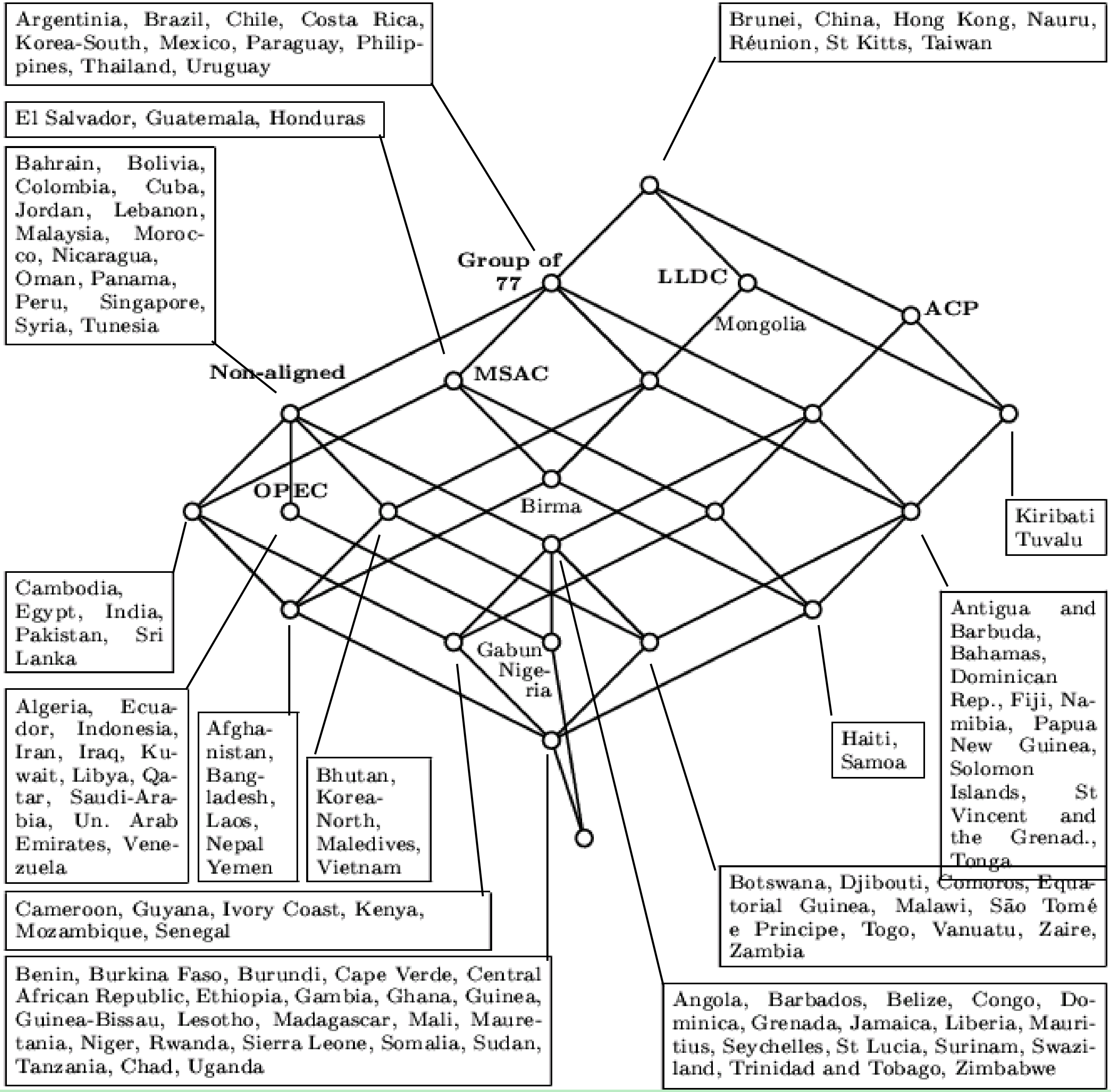
is non-redundant and complete. We call $\mathcal{L}$ **stem basis.**

Example: Membership of developing countries in supranational groups (Source: Lexikon Dritte Welt. Rowohlt-Verlag, Reinbek 1993)

Taken from: B. Ganter, R. Wille: Formal Concept Analysis - Mathematical Foundations. Springer, Heidelberg 1999

	Group of 77	Non-aligned	LLDC	MSAC	OPEC	ACP
Afghanistan	×	×	×	×		
Algeria	×	×			×	
Angola	×	×				×
Antigua and Barbuda	×					×
Argentina	×					
Bahamas	×					×
Bahrain	×	×				
Bangladesh	×	×	×	×		
Barbados	×	×				×
Belize	×	×				×
Benin	×	×	×	×		×
Bhutan	×	×	×			
Bolivia	×	×				
Botswana	×	×	×			×
Brazil	×					
Brunei						
Burkina Faso	×	×	×	×		×
Burundi	×	×	×	×		×
Cambodia	×	×		×		
Cameroon	×	×		×		×
Cape Verde	×	×	×	×		×
Central African Rep.	×	×	×	×		×
Chad	×	×	×	×		×
Chile	×					
China						
Colombia	×	×				
Comoros	×	×	×			×
Congo	×	×				×
Costa Rica	×					
Cuba	×	×				
Djibouti	×	×	×			×
Dominica	×	×				×
Dominican Rep.	×					×
Ecuador	×	×			×	
Egypt	×	×		×		
El Salvador	×			×		
Equatorial Guinea	×	×	×			×
Ethiopia	×	×	×	×		×
Fiji	×					×
Gabon	×	×			×	×
Gambia	×	×	×	×		×
Ghana	×	×	×	×		×
Grenada	×	×				×
Guatemala	×			×		
Guinea	×	×	×	×		×
Guinea-Bissau	×	×	×	×		×
Guyana	×	×		×		×
Haiti	×		×	×		×
Honduras	×			×		
Hong Kong						
India	×	×		×		
Indonesia	×	×			×	
Iran	×	×			×	
Iraq	×	×			×	
Ivory Coast	×	×		×		×
Jamaica	×	×				×
Jordan	×	×				
Kenya	×	×		×		×
Kiribati			×			×
Korea-North	×	×	×			
Korea-South	×					
Kuwait	×	×			×	
Laos	×	×	×	×		
Lebanon	×	×				
Lesotho	×	×	×	×		×
Liberia	×	×				×
Libya	×	×			×	
Madagascar	×	×	×	×		×
Malawi	×	×	×			×
Malaysia	×	×				
Maledives	×	×	×			
Mali	×	×	×	×		×
Mauretania	×	×	×	×		×
Mauritius	×	×				×
Mexico	×					
Mongolia			×			
Morocco	×	×				
Mozambique	×	×		×		×
Myanmar	×		×	×		
Namibia	×					×
Nauru						
Nepal	×	×	×	×		
Nicaragua	×	×				
Niger	×	×	×	×		×
Nigeria	×	×			×	×
Oman	×	×				
Pakistan	×	×		×		
Panama	×	×				
Papua New Guinea	×					×
Paraguay	×					
Peru	×	×				
Philippines	×					
Qatar	×	×			×	
Réunion						
Rwanda	×	×	×	×		×
Samoa	×		×	×		×
São Tomé e Principe	×	×	×			×
Saudi Arabia	×	×			×	
Senegal	×	×		×		×
Seychelles	×	×				×
Sierra Leone	×	×	×	×		×
Singapore	×	×				
Solomon Islands	×					×
Somalia	×	×	×	×		×
Sri Lanka	×	×		×		
St Kitts						
St Lucia	×	×				×
St Vincent& Grenad.	×					×
Sudan	×	×	×	×		×
Surinam	×	×				×
Swaziland	×	×				×
Syria	×	×				
Taiwan						
Tanzania	×	×	×	×		×
Thailand	×					
Togo	×	×	×			×
Tonga	×					×
Trinidad and Tobago	×	×				×
Tunisia	×	×				
Tuvalu			×			×
Uganda	×	×	×	×		×
United Arab Emirates	×	×			×	
Uruguay	×					
Vanuatu	×	×	×			×
Venezuela	×	×			×	
Vietnam	×	×	×			
Yemen	×	×	×	×		
Zaire	×	×	×			×
Zambia	×	×	×			×
Zimbabwe	×	×				×

The abbreviations stand for: LLDC := Least Developed Countries, MSAC := Most Seriously Affected Countries, OPEC := Organization of Petrol Exporting Countries, ACP := African, Caribbean and Pacific Countries.



Argentinia, Brazil, Chile, Costa Rica, Korea-South, Mexico, Paraguay, Philippines, Thailand, Uruguay

El Salvador, Guatemala, Honduras

Bahrain, Bolivia, Colombia, Cuba, Jordan, Lebanon, Malaysia, Morocco, Nicaragua, Oman, Panama, Peru, Singapore, Syria, Tunesia

Brunei, China, Hong Kong, Nauru, Réunion, St Kitts, Taiwan

Kiribati Tuvalu

Cambodia, Egypt, India, Pakistan, Sri Lanka

Algeria, Ecuador, Indonesia, Iran, Iraq, Kuwait, Libya, Qatar, Saudi-Arabia, Un. Arab Emirates, Venezuela

Afghanistan, Bangladesh, Laos, Nepal Yemen

Bhutan, Korea-North, Maledives, Vietnam

Haiti, Samoa

Antigua and Barbuda, Bahamas, Dominican Rep., Fiji, Namibia, Papua New Guinea, Solomon Islands, St Vincent and the Grenad., Tonga

Botswana, Djibouti, Comoros, Equatorial Guinea, Malawi, São Tomé e Principe, Togo, Vanuatu, Zaire, Zambia

Cameroon, Guyana, Ivory Coast, Kenya, Mozambique, Senegal

Benin, Burkina Faso, Burundi, Cape Verde, Central African Republic, Ethiopia, Gambia, Ghana, Guinea, Guinea-Bissau, Lesotho, Madagascar, Mali, Mauretania, Niger, Rwanda, Sierra Leone, Somalia, Sudan, Tanzania, Chad, Uganda

Angola, Barbados, Belize, Congo, Dominica, Grenada, Jamaica, Liberia, Mauritius, Seychelles, St Lucia, Surinam, Swaziland, Trinidad and Tobago, Zimbabwe

Stem basis of the 3rd World context:

OPEC → Group of 77, Non-Alligned

MSAC → Group of 77

Non-Alligned → Group of 77

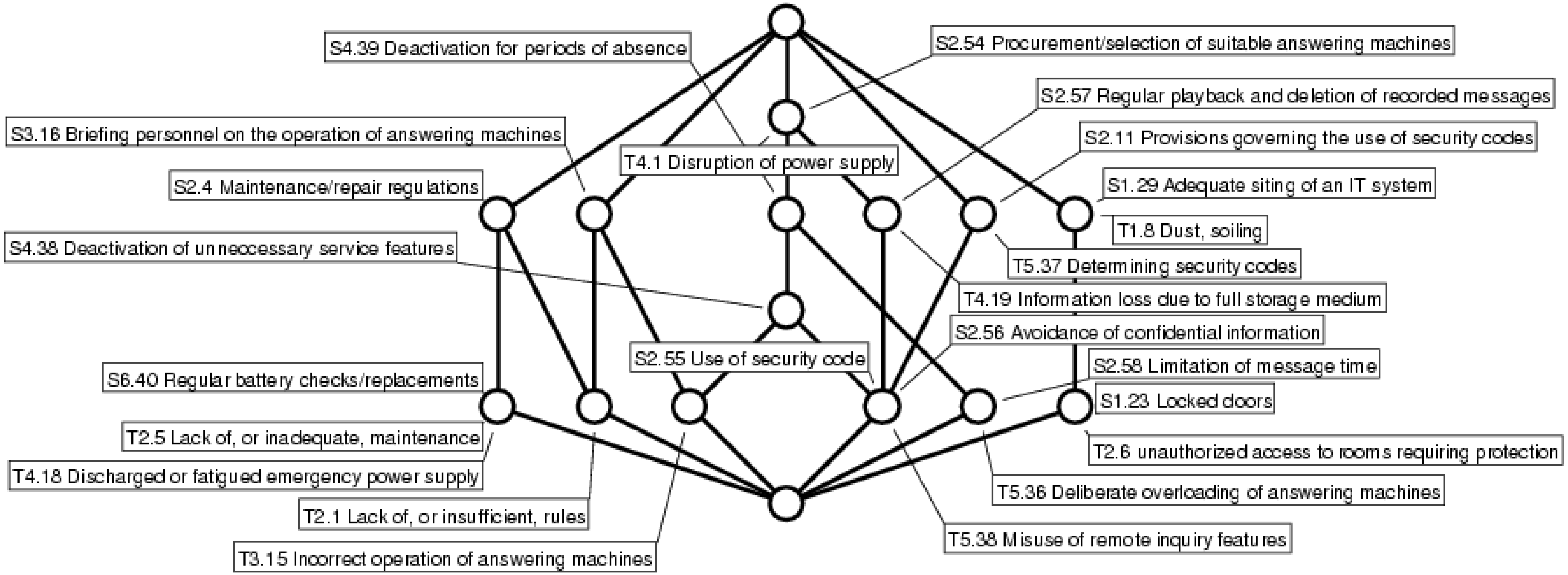
Group of 77, Non-Alligned, MSAC, OPEC → LLDC, AKP

Group of 77, Non-Alligned, LLDC, OPEC → MSAC, AKP

1. Introduction
2. Formal Contexts & Concept Lattices
3. **Application Examples I**
4. Computing Concept Lattices
5. Exercises
6. Conceptual Clustering
7. Exercises
8. FCA-Based Mining of Association Rules
9. Application Examples II

IT-Security Management

- Supports the analysis of security risks in IT units
- status quo test for establishing guidelines and checklists


S4.39 Deactivation for periods of absence
S2.54 Procurement/selection of suitable answering machines
S2.57 Regular playback and deletion of recorded messages
S3.16 Briefing personnel on the operation of answering machines
S2.11 Provisions governing the use of security codes
T4.1 Disruption of power supply
S2.4 Maintenance/repair regulations
S1.29 Adequate siting of an IT system
T1.8 Dust, soiling
S4.38 Deactivation of unneccessary service features
T5.37 Determining security codes
T4.19 Information loss due to full storage medium
S2.56 Avoidance of confidential information
S2.55 Use of security code
S6.40 Regular battery checks/replacements
S2.58 Limitation of message time
S1.23 Locked doors
T2.5 Lack of, or inadequate, maintenance
T2.6 unauthorized access to rooms requiring protection
T4.18 Discharged or fatigued emergency power supply
T5.36 Deliberate overloading of answering machines
T2.1 Lack of, or insufficient, rules
T5.38 Misuse of remote inquiry features
T3.15 Incorrect operation of answering machines

More examples ...

... on the overhead projector.

1. Introduction
2. Formal Contexts & Concept Lattices
3. Application Examples I
4. **Computing Concept Lattices**
5. Exercises
6. Conceptual Clustering
7. Exercises
8. FCA-Based Mining of Association Rules
9. Application Examples II

There exist a number of algorithms for computing concept lattices:

- Naive approach
- Intersection method
- Titanic [Stumme et al 2001]
- Next-Closure [Ganter 1984]
- and some incremental algorithms

Naive Approach

Theorem: Every concept of the context (G,M,I) is of the form (X'', X') for some $X \subseteq G$ (and of the form (Y', Y'') for at least one $Y \subseteq M$).

On the other hand, each such pair is a concept.

„Algorithm“: Determine for each subset Y of M the pair (Y', Y'').

But: Too many concepts are created too often.

Intersection Method

This method is also suitable for manual computation. [Wille 1982]

It provides the best worst-case time complexity. [Nourine, Raynoud 1999]

It uses the following theorem:

Theorem: Each intent is intersection of attribute intents. I.e., the closure system of all intents is generated by the attribute intents.

The question is which intersections of attribute intents to take.

⇘ Example „Faces“ on the Blackboard

How to compute/draw a concept lattice (manually):

- From left to right, consider all intersections of each column extent with every column extent to the left of it. If the resulting extent is not already a column, add it as column at the right end of the context. Repeat this until the last (added) column is reached.

- Add a full column, unless there is already one. (Now each column stands for one concept.)

- Draw a circle for the full column.

- Draw for each column, starting for the ones with a maximal number of crosses, a circle, and link it with a line to the circles where the column comprises the current column.

- Attach every attribute label to the circle of the corresponding column.

- Attach every object label to the circle laying exactly below the circles of the attributes in its intent.

How to check the drawing of a concept lattice:

- Is it really a lattice? (This test is usually skipped.)
- Is every concept with exactly one upper neighbor labeled by at least one attribute?
- Is every concept with exactly one lower neighbor labeled by at least one object?
- Is, for all $g \in G$ and all $m \in M$, the label of object g below the label of attribute m iff $(g,m) \in I$?

TITANIC

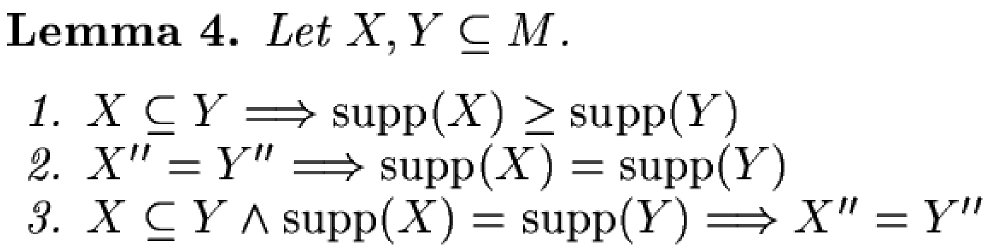
computes the closure system of all (frequent) concept intents using the *support function*:

Def.: The **support** of an attribute set (itemset) $X \subseteq M$ is given by

$$\mathrm{supp}(X) = \frac{|X'|}{|G|}$$

Only concepts with a support above a threshold minsupp $\in [0,1]$.

TITANIC makes use of some simple facts about the support function:



Lemma 4. *Let* $X, Y \subseteq M$.

1. $X \subseteq Y \Longrightarrow \mathrm{supp}(X) \geq \mathrm{supp}(Y)$
2. $X'' = Y'' \Longrightarrow \mathrm{supp}(X) = \mathrm{supp}(Y)$
3. $X \subseteq Y \wedge \mathrm{supp}(X) = \mathrm{supp}(Y) \Longrightarrow X'' = Y''$

TITANIC

tries to optimize the following three questions:

1. How can the closure of an itemset be determined based on supports only?

2. How can the closure system be computed with determining as few closures as possible?

3. How can as many supports as possible be derived from already known supports?

TITANIC

1. How can the closure of an itemset be determined based on supports only?

$$X'' = X \cup \{ x \in M \setminus X \mid supp(X) = supp(X \cup x) \}$$

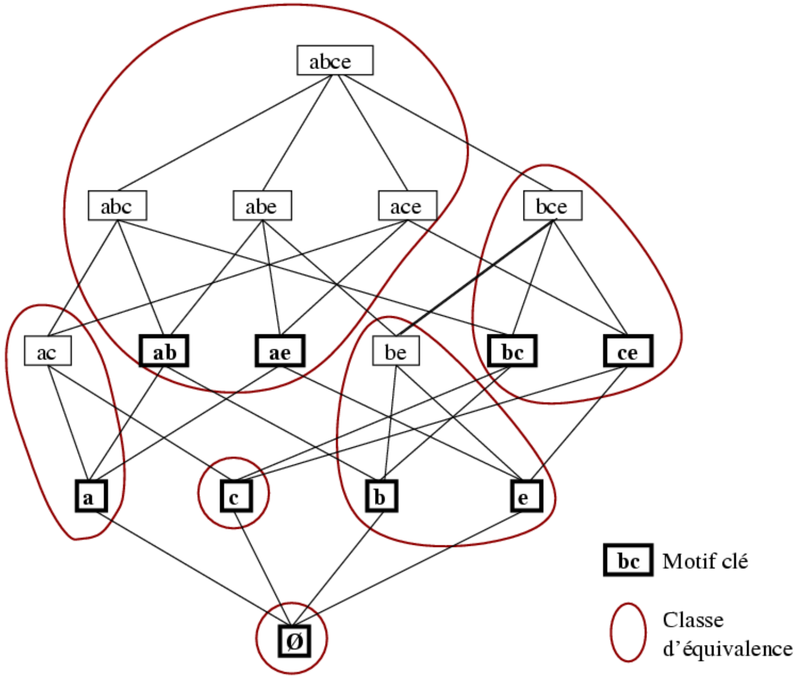
Example: { b,c }“ = { b, c, e }, since

supp({ b, c }) = 1/3

supp({ a, b, c }) = 0/3

supp({ b, c, e }) = 1/3,

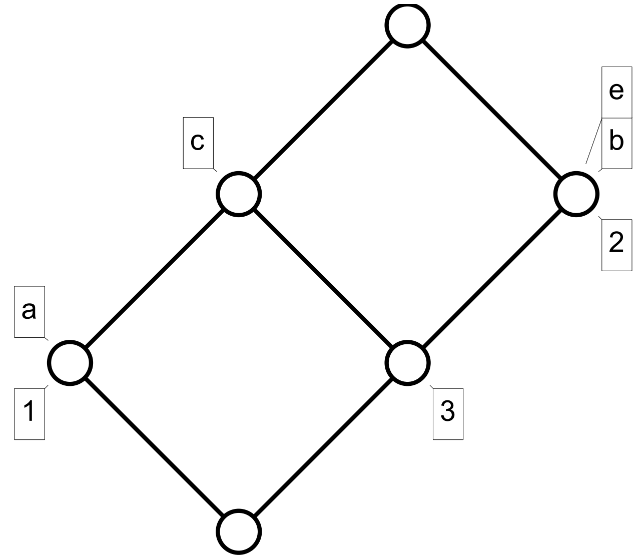
	a	b	c	e
1	×		×	
2		×		×
3		×	×	×

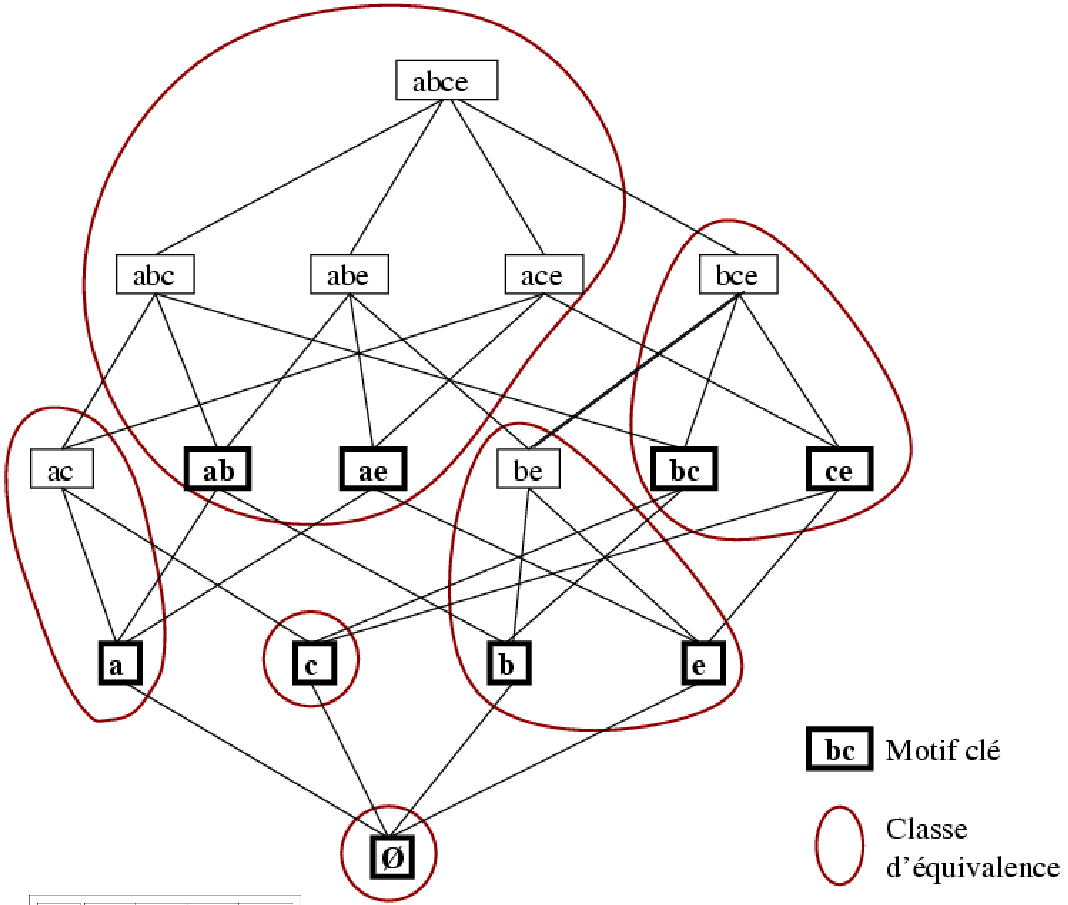


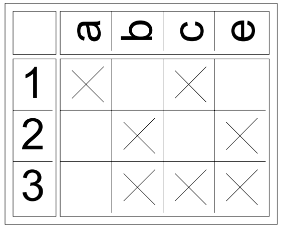
TITANIC

2. **How can the closure system be computed with determining as few closures as possible?**

- We determine only the closures of the minimal generators.







	a	b	c	e
1	×		×	
2		×		×
3		×	×	×

TITANIC

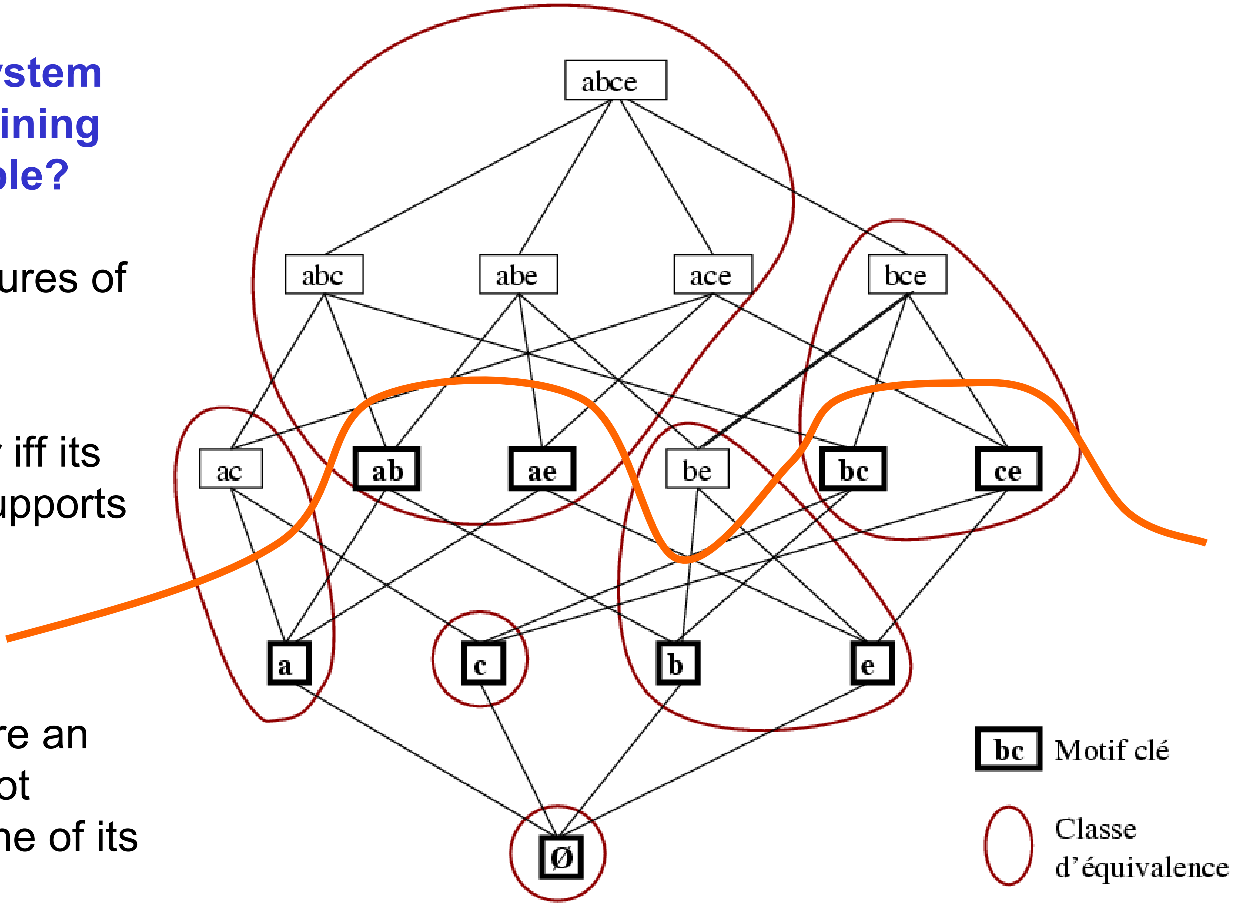
2. How can the closure system be computed with determining as few closures as possible?

We determine only the closures of the minimal generators.

- A set is minimal generator iff its support is different of the supports of all its lower covers.

- The minimal generators are an order ideal (i.e., if a set is not minimal generator, then none of its supersets is either.)

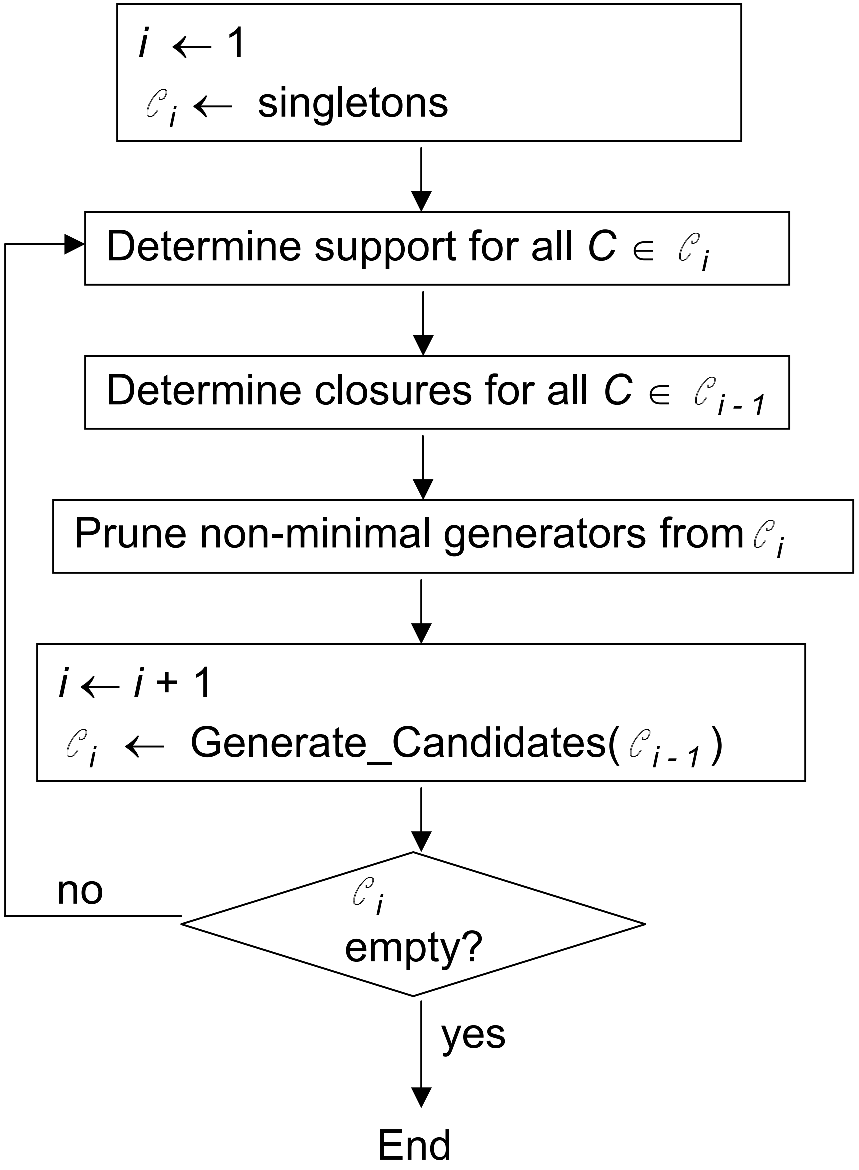
→ Apriori like approach



In the example, TITANIC needs two runs (and Apriori four).

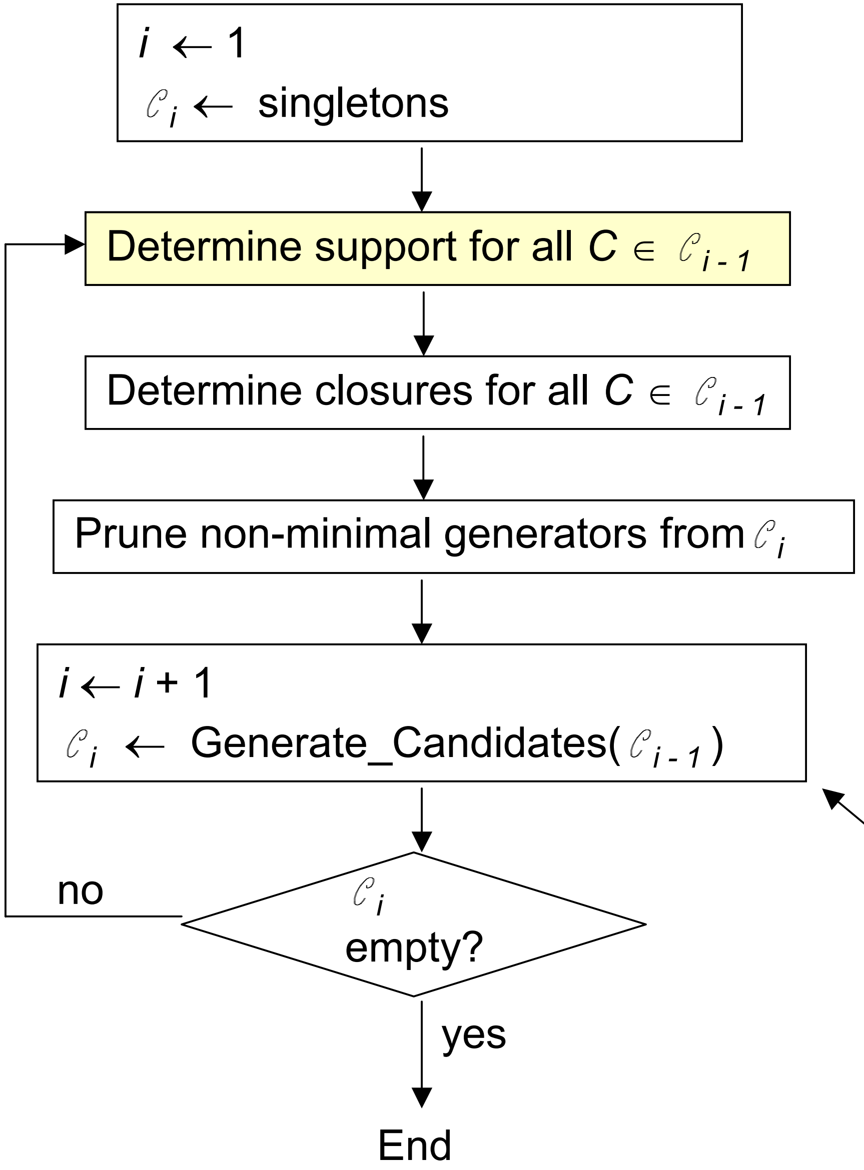
TITANIC

Apriori like approach


$i \leftarrow 1$
$\mathcal{C}_i \leftarrow$ singletons
Determine support for all $C \in \mathcal{C}_i$
Determine closures for all $C \in \mathcal{C}_{i-1}$
Prune non-minimal generators from $\mathcal{C}_i$
$i \leftarrow i + 1$
$\mathcal{C}_i \leftarrow$ Generate_Candidates($\mathcal{C}_{i-1}$)
no
$\mathcal{C}_i$
empty?
yes
End

TITANIC

compared with Apriori



$i \leftarrow 1$

$\mathcal{C}_i \leftarrow$ singletons

Determine support for all $C \in \mathcal{C}_{i-1}$

Determine closures for all $C \in \mathcal{C}_{i-1}$

Prune non-minimal generators from $\mathcal{C}_i$

$i \leftarrow i + 1$

$\mathcal{C}_i \leftarrow$ Generate_Candidates($\mathcal{C}_{i-1}$)

$\mathcal{C}_i$ empty?

End

If the support is too low **or equal to the support of a lower cover**, the candidate is pruned.

We only generate candidates for **minimal generators**.

TITANIC

1. How can the closure of an itemset be determined based on supports only?

$$X'' = X \cup \quad x \in M \setminus X \mid \mathrm{supp}(X) = \mathrm{supp}(X \cup \quad x\ )$$

2. How can the closure system be computed with determining as few closures as possible?

Approach à la Apriori

3. How can as many supports as possible be derived from already known supports?

3. How can as many supports as possible be derived from already known supports?

	a	b	c	e
1	✕		✕	
2		✕		✕
3		✕	✕	✕

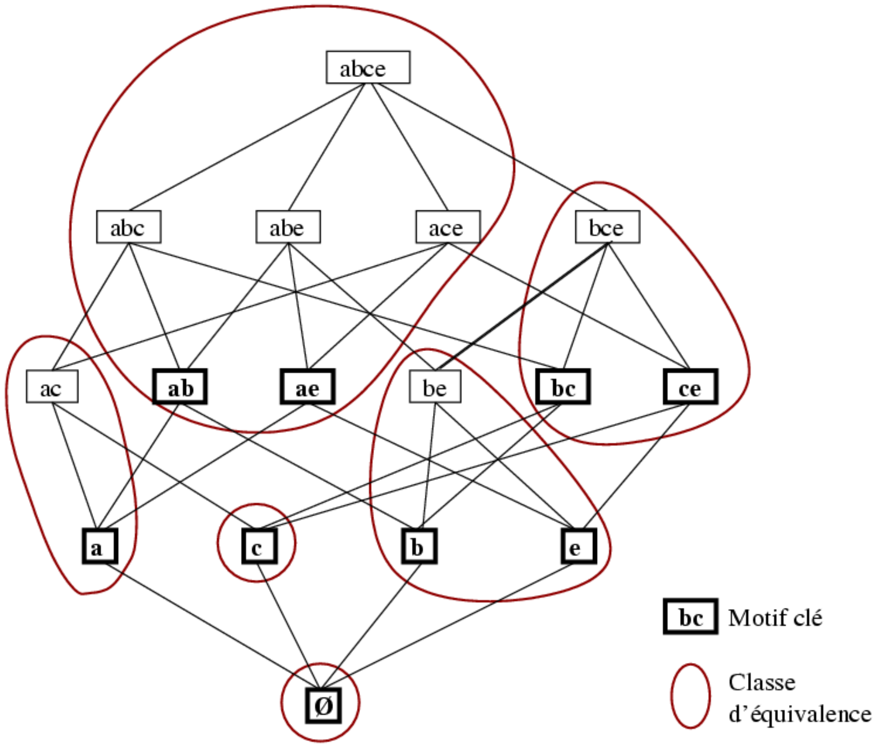
Theorem: If *X* is no minimal generator, then

supp(*X*) = min { supp(*K*) | *K* is minimal generator, $K \subseteq X$ } .

Example: supp({ a, b, c }) = min { 0/3, 1/3, 1/3, 2/3, 2/3 } = 0, since the set is no minimal generator, and since

supp({ a, b }) = 0/3, supp({ b, c }) = 1/3
supp({ a }) = 1/3, supp({ b }) = 2/3
supp({ c }) = 2/3

Remark: It is sufficient to check the largest generators *K* with $K \subseteq X$, i.e. here { a, b } and { b, c} .



TITANIC

1. How can the closure of an itemset be determined based on supports only?

$$X'' = X \cup \quad x \in M \setminus X \mid \mathrm{supp}(X) = \mathrm{supp}(X \cup \quad x\ )$$

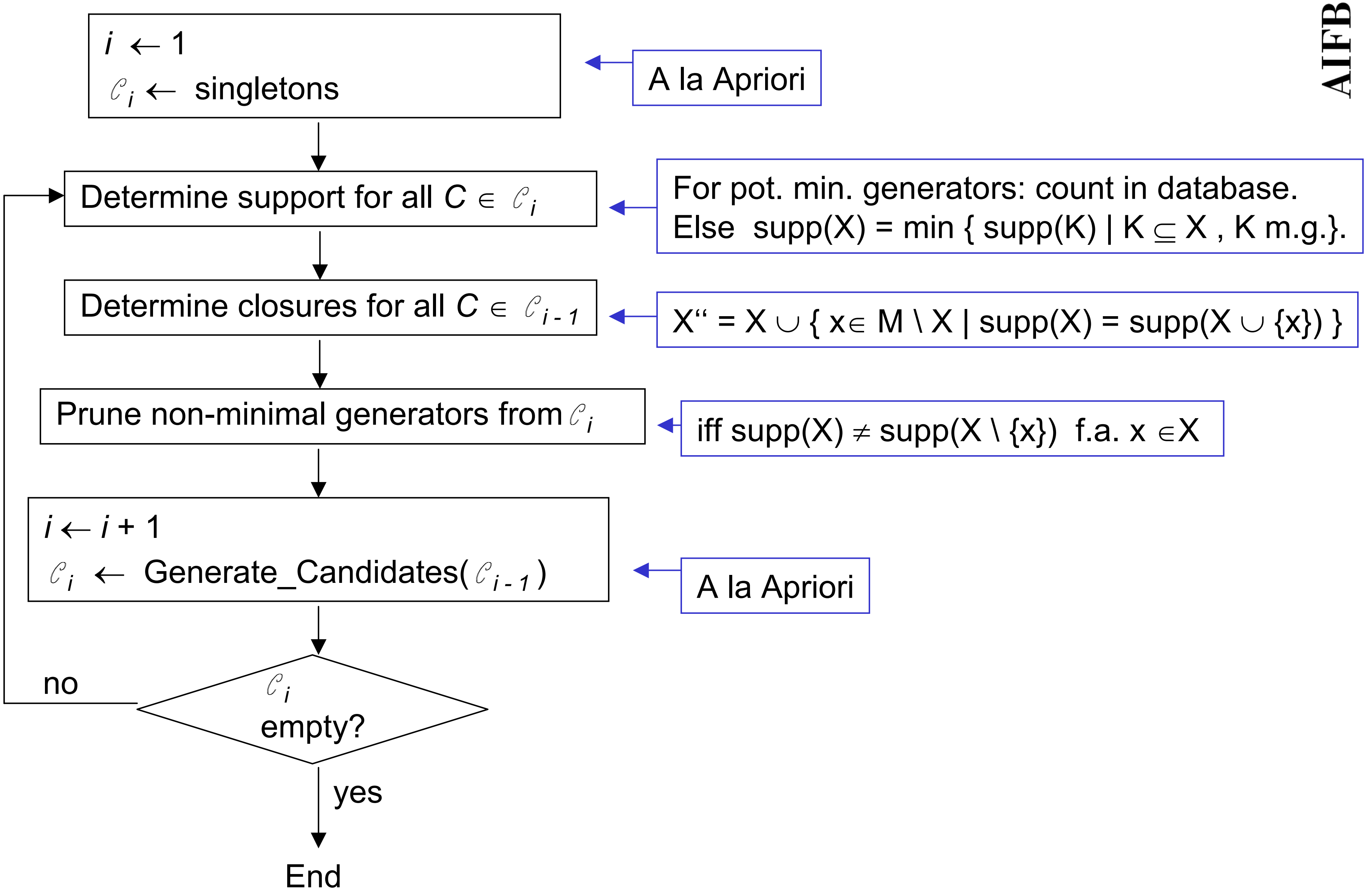
2. **How can the closure system be computed with determining as few closures as possible?**

Approach à la Apriori

3. **How can as many supports as possible be derived from already known supports?**

If *X* is no minimal generator, then

$$\mathrm{supp}(X) = \min\ \{ \mathrm{supp}(K) \mid K \text{ is minimal generator, } K \subseteq X \}\ .$$



$i \leftarrow 1$

$\mathcal{C}_i \leftarrow$ singletons

A la Apriori

Determine support for all $C \in \mathcal{C}_i$

For pot. min. generators: count in database.
Else supp(X) = min { supp(K) | K $\subseteq$ X , K m.g.}.

Determine closures for all $C \in \mathcal{C}_{i-1}$

X'' = X $\cup$ { x$\in$ M \ X | supp(X) = supp(X $\cup$ {x}) }

Prune non-minimal generators from $\mathcal{C}_i$

iff supp(X) $\neq$ supp(X \ {x}) f.a. x $\in$X

$i \leftarrow i + 1$

$\mathcal{C}_i \leftarrow$ Generate_Candidates($\mathcal{C}_{i-1}$)

A la Apriori

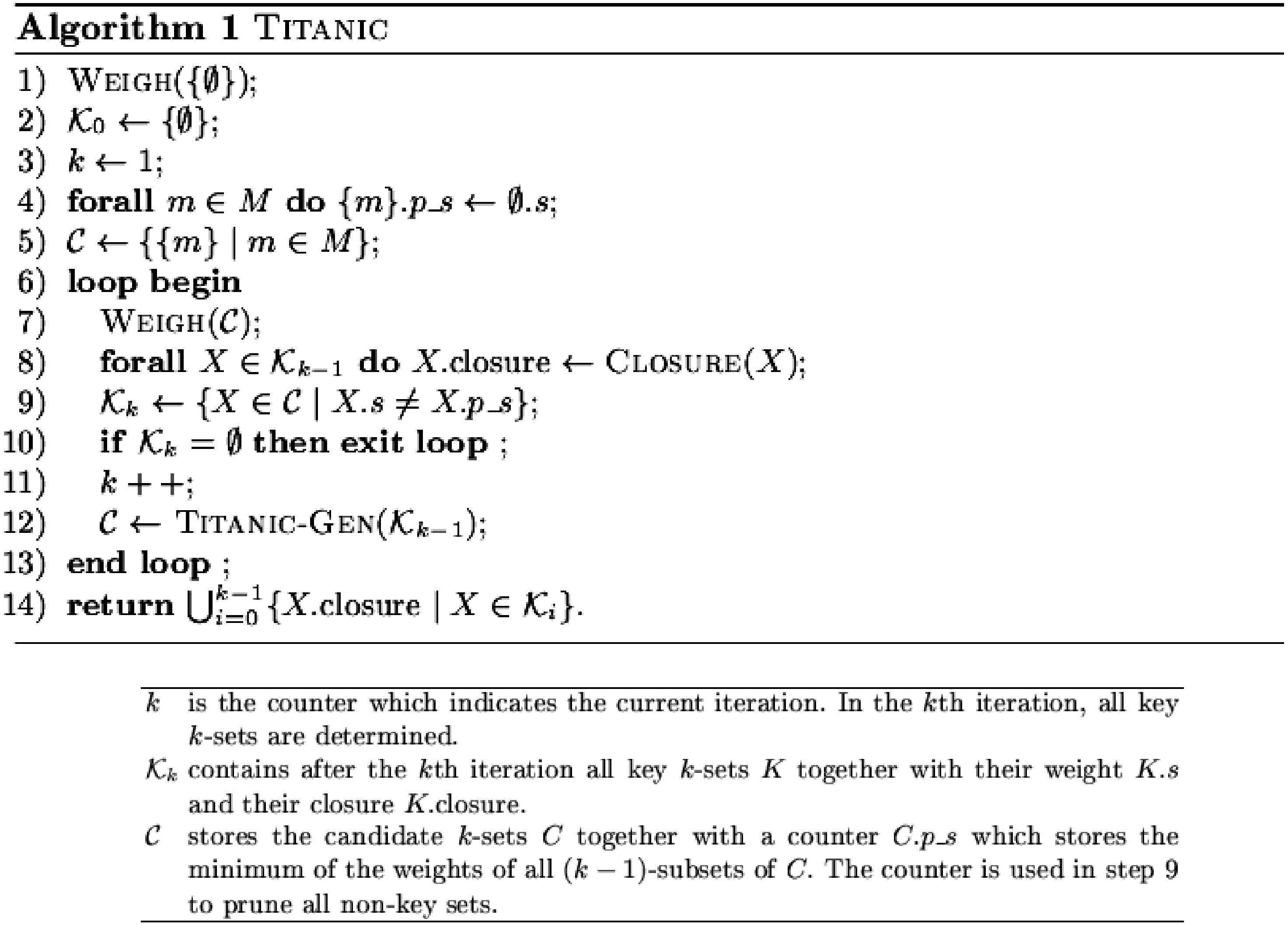
$\mathcal{C}_i$ empty?

no

yes

End

TITANIC



Algorithm 1 TITANIC

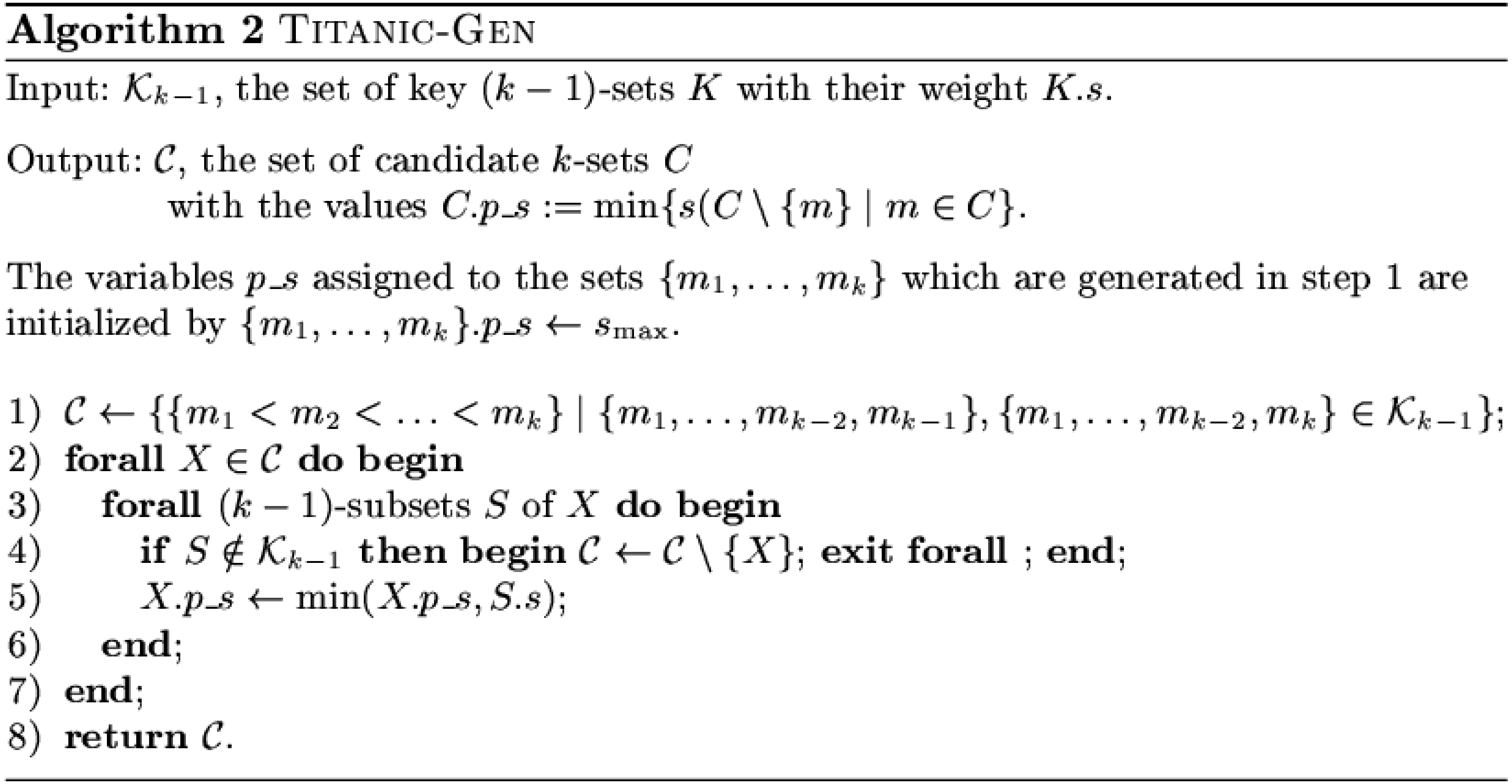
```
1)  WEIGH($\{\emptyset\}$);
2)  $\mathcal{K}_0 \leftarrow \{\emptyset\}$;
3)  $k \leftarrow 1$;
4)  forall $m \in M$ do $\{m\}.p\_s \leftarrow \emptyset.s$;
5)  $\mathcal{C} \leftarrow \{\{m\} \mid m \in M\}$;
6)  loop begin
7)    WEIGH($\mathcal{C}$);
8)    forall $X \in \mathcal{K}_{k-1}$ do $X$.closure $\leftarrow$ CLOSURE($X$);
9)    $\mathcal{K}_k \leftarrow \{X \in \mathcal{C} \mid X.s \neq X.p\_s\}$;
10)   if $\mathcal{K}_k = \emptyset$ then exit loop ;
11)   $k++$;
12)   $\mathcal{C} \leftarrow$ TITANIC-GEN($\mathcal{K}_{k-1}$);
13) end loop ;
14) return $\bigcup_{i=0}^{k-1}\{X.\text{closure} \mid X \in \mathcal{K}_i\}$.
```

k is the counter which indicates the current iteration. In the kth iteration, all key k-sets are determined.

$\mathcal{K}_k$ contains after the kth iteration all key k-sets K together with their weight $K.s$ and their closure K.closure.

$\mathcal{C}$ stores the candidate k-sets C together with a counter $C.p\_s$ which stores the minimum of the weights of all $(k-1)$-subsets of C. The counter is used in step 9 to prune all non-key sets.

TITANIC



Algorithm 2 TITANIC-GEN

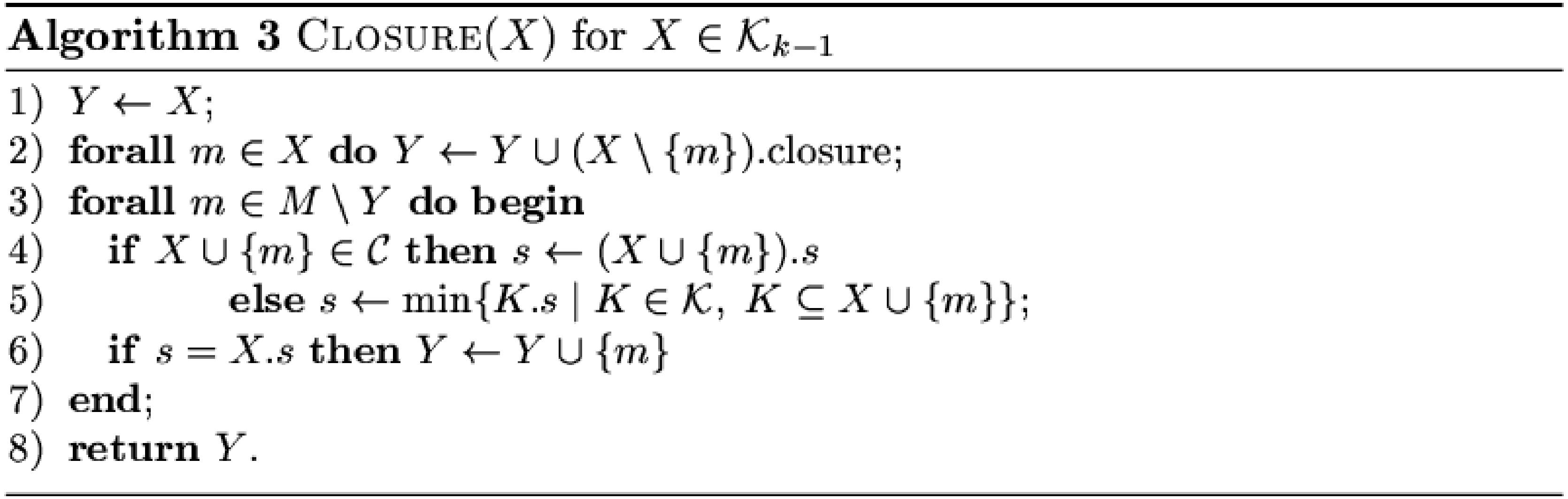
Input: $\mathcal{K}_{k-1}$, the set of key $(k-1)$-sets K with their weight $K.s$.

Output: $\mathcal{C}$, the set of candidate k-sets C
with the values $C.p\_s := \min\{s(C \setminus \{m\} \mid m \in C\}$.

The variables p_s assigned to the sets $\{m_1, \ldots, m_k\}$ which are generated in step 1 are initialized by $\{m_1, \ldots, m_k\}.p\_s \leftarrow s_{\max}$.

```
1) $\mathcal{C} \leftarrow \{\{m_1 < m_2 < \ldots < m_k\} \mid \{m_1, \ldots, m_{k-2}, m_{k-1}\}, \{m_1, \ldots, m_{k-2}, m_k\} \in \mathcal{K}_{k-1}\}$;
2) forall $X \in \mathcal{C}$ do begin
3)    forall $(k-1)$-subsets $S$ of $X$ do begin
4)       if $S \notin \mathcal{K}_{k-1}$ then begin $\mathcal{C} \leftarrow \mathcal{C} \setminus \{X\}$; exit forall ; end;
5)       $X.p\_s \leftarrow \min(X.p\_s, S.s)$;
6)    end;
7) end;
8) return $\mathcal{C}$.
```

TITANIC

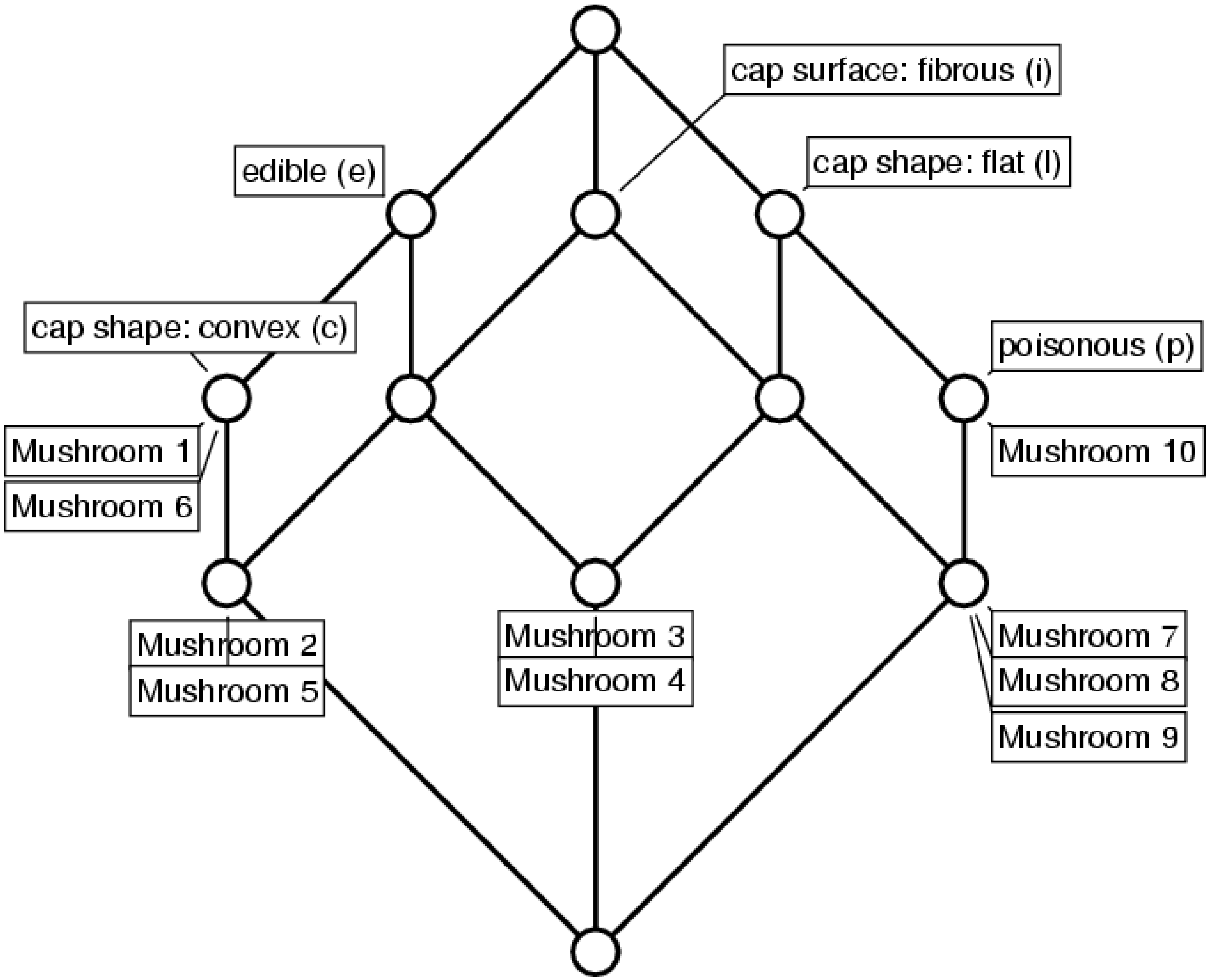


Algorithm 3 CLOSURE(X) for $X \in \mathcal{K}_{k-1}$

```
$Y \leftarrow X$;
forall $m \in X$ do $Y \leftarrow Y \cup (X \setminus \{m\})$.closure;
forall $m \in M \setminus Y$ do begin
    if $X \cup \{m\} \in \mathcal{C}$ then $s \leftarrow (X \cup \{m\}).s$
              else $s \leftarrow \min\{K.s \mid K \in \mathcal{K},\ K \subseteq X \cup \{m\}\}$;
    if $s = X.s$ then $Y \leftarrow Y \cup \{m\}$
end;
return $Y$.
```

Example of TITANIC

	edible (e)	poisonous (p)	cap shape: convex (c)	cap shape: flat (l)	cap surface: fibrous (i)
Mushroom 1	×		×		
Mushroom 2	×		×		×
Mushroom 3	×			×	×
Mushroom 4	×			×	×
Mushroom 5	×		×		×
Mushroom 6	×		×		
Mushroom 7		×		×	×
Mushroom 8		×		×	×
Mushroom 9		×		×	×
Mushroom 10		×		×	



cap surface: fibrous (i)

edible (e)

cap shape: flat (l)

cap shape: convex (c)

poisonous (p)

Mushroom 1

Mushroom 6

Mushroom 10

Mushroom 2

Mushroom 5

Mushroom 3

Mushroom 4

Mushroom 7

Mushroom 8

Mushroom 9

$k = 0$:

step 1		step 2
X	$X.s$	$X \in \mathcal{K}_k$?
$\emptyset$	1	yes

$k = 1$:

steps 4+5		step 7	step 9
X	$X.p\_s$	$X.s$	$X \in \mathcal{K}_k$?
$\{e\}$	1	6/10	yes
$\{p\}$	1	4/10	yes
$\{c\}$	1	4/10	yes
$\{l\}$	1	6/10	yes
$\{i\}$	1	7/10	yes

Step 8 returns: $\emptyset$.closure ← $\emptyset$

	edible (e)	poisonous (p)	cap shape: convex (c)	cap shape: flat (l)	cap surface: fibrous (i)
Mushroom 1	☒		☒		
Mushroom 2	☒		☒		☒
Mushroom 3	☒			☒	☒
Mushroom 4	☒			☒	☒
Mushroom 5	☒		☒		☒
Mushroom 6	☒		☒		
Mushroom 7		☒		☒	☒
Mushroom 8		☒		☒	☒
Mushroom 9		☒		☒	☒
Mushroom 10		☒		☒	

Then the algorithm repeats the loop for $k = 2, 3$, and 4:

<u>$k = 2$:</u>

step 12		step 7	step 9
X	$X.p\_s$	$X.s$	$X \in \mathcal{K}_k$?
$\{e, p\}$	4/10	0	yes
$\{e, c\}$	4/10	4/10	no
$\{e, l\}$	6/10	2/10	yes
$\{e, i\}$	6/10	4/10	yes
$\{p, c\}$	4/10	0	yes
$\{p, l\}$	4/10	4/10	no
$\{p, i\}$	4/10	3/10	yes
$\{c, l\}$	4/10	0	yes
$\{c, i\}$	4/10	2/10	yes
$\{l, i\}$	6/10	5/10	yes

Step 8 returns: $\{e\}$.closure ← $\{e\}$
$\{p\}$.closure ← $\{p, l\}$
$\{c\}$.closure ← $\{c, e\}$
$\{l\}$.closure ← $\{l\}$
$\{i\}$.closure ← $\{i\}$

<u>$k = 3$:</u>

step 12		step 7	step 9
X	$X.p\_s$	$X.s$	$X \in \mathcal{K}_k$?
$\{e, l, i\}$	2/10	2/10	no
$\{p, c, i\}$	4/10	0	yes
$\{c, l, i\}$	4/10	0	yes

Step 8 returns: $\{e, p\}$.closure ← $\{e, p, c, l, i\}$
$\{e, l\}$.closure ← $\{e, l, i\}$
$\{e, i\}$.closure ← $\{e, i\}$
$\{p, c\}$.closure ← $\{e, p, c, l, i\}$
$\{p, i\}$.closure ← $\{p, l, i\}$
$\{c, l\}$.closure ← $\{e, p, c, l, i\}$
$\{c, i\}$.closure ← $\{e, c, i\}$
$\{l, i\}$.closure ← $\{l, i\}$

	edible (e)	poisonous (p)	cap shape: convex (c)	cap shape: flat (l)	cap surface: fibrous (i)
Mushroom 1	╳		╳		
Mushroom 2	╳		╳		╳
Mushroom 3	╳			╳	╳
Mushroom 4	╳			╳	╳
Mushroom 5	╳		╳		╳
Mushroom 6	╳		╳		
Mushroom 7		╳		╳	╳
Mushroom 8		╳		╳	╳
Mushroom 9		╳		╳	╳
Mushroom 10		╳		╳	

$k = 4$:

Step 12 returns the empty set. Hence there is nothing to weigh in step 7. Step 9 sets $\mathcal{K}_4$ equal to the empty set; and in step 10, the loop is exited.

Step 8 returns: $\{p, c, i\}.\text{closure} \leftarrow \{e, p, c, l, i\}$
$\{c, l, i\}.\text{closure} \leftarrow \{e, p, c, l, i\}$

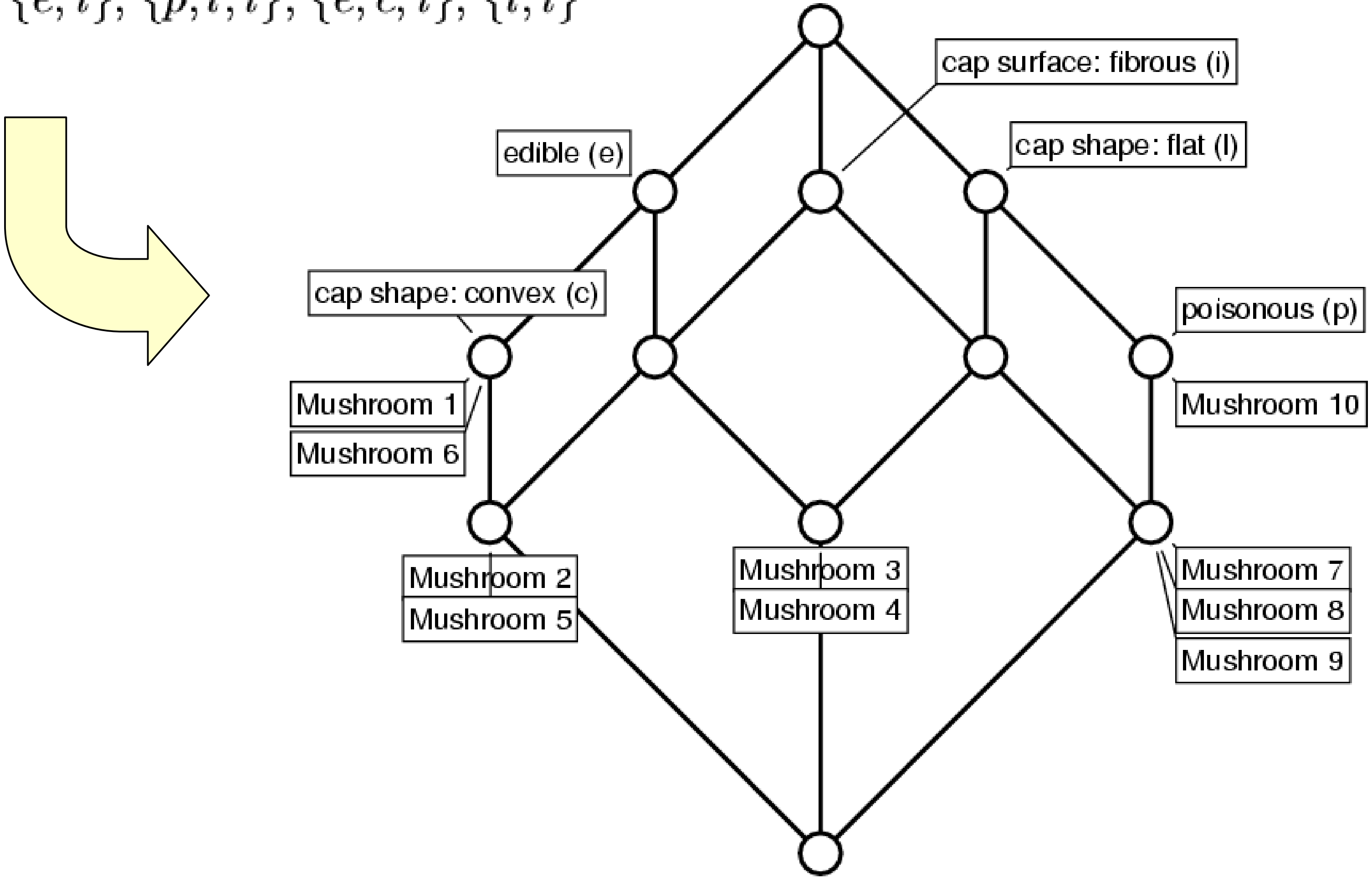
Finally the algorithm collects all concept intents (step 14):

$$\emptyset,\ \{e\},\ \{p, l\},\ \{c, e\},\ \{l\},\ \{i\},\ \{e, p, c, l, i\},$$
$$\{e, l, i\},\ \{e, i\},\ \{p, l, i\},\ \{e, c, i\},\ \{l, i\}$$

(which are exactly the intents of the concepts of the concept lattice in Figure 8). The algorithm determined the support of $5 + 10 + 3 = 18$ attribute sets in three passes of the database.

	edible (e)	poisonous (p)	cap shape: convex (c)	cap shape: flat (l)	cap surface: fibrous (i)
Mushroom 1	×		×		
Mushroom 2	×		×		×
Mushroom 3	×			×	×
Mushroom 4	×			×	×
Mushroom 5	×		×		×
Mushroom 6	×		×		
Mushroom 7		×		×	×
Mushroom 8		×		×	×
Mushroom 9		×		×	×
Mushroom 10		×		×	

$$\emptyset,\ \{e\},\ \{p,l\},\ \{c,e\},\ \{l\},\ \{i\},\ \{e,p,c,l,i\},$$
$$\{e,l,i\},\ \{e,i\},\ \{p,l,i\},\ \{e,c,i\},\ \{l,i\}$$



Next-Closure

was developed by B. Ganter (1984).

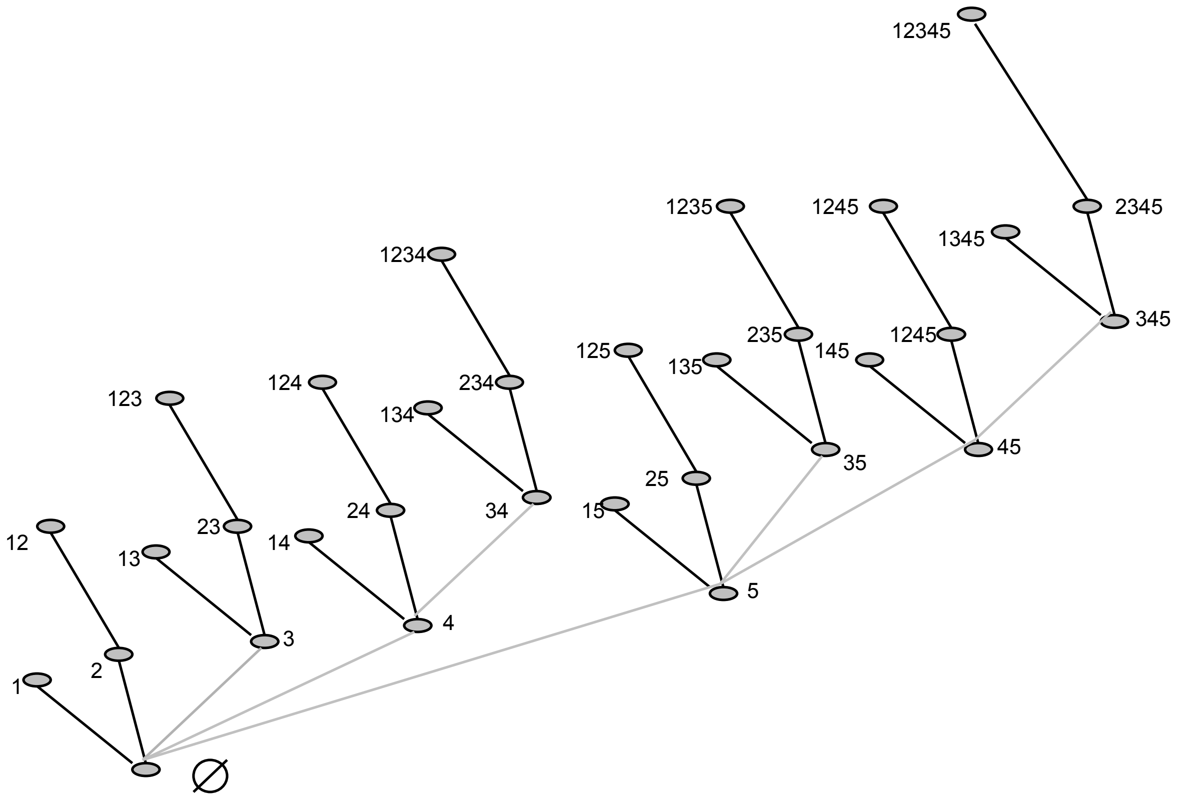
It can be used

- to determine the concept lattice or
- to determine the concept lattice together with the stem basis or
- for interactive knowledge acquisition.

It determines the concept intents in **lectical order.**

Let $M = \{1, ..., n\}$. $A \subseteq M$ is **lectically smaller** than $B \subseteq M$, if $B \neq A$ if the smallest element where A and B differ belongs to B :

$$A < B \;:\Leftrightarrow\; \exists\, i \in B \backslash A\colon A \cap \{1, 2, ..., i\text{-}1\} = B \cap \{1, 2, ..., i\text{-}1\}$$



We need the following:

$A <_i B \ :\Leftrightarrow i \in B \setminus A \ \wedge \ A \cap \{1, 2, ..., i\text{-}1\} = B \cap \{1, 2, ..., i\text{-}1\}$

$A \oplus i \ := (A \cap \{1, 2, ..., i\text{-}1\}) \cup \{i\}$

Theorem: The smallest concept intent, which according to the lectical order is larger as a given set $A \subset M$, is

$$(A \oplus i)'',$$

where i is the largest element of M with $A <_i (A \oplus i)''$.

Algorithm **Next-Closure** for determining all concept intents:

1) The lectically smallest concept intent is ∅‘‘.

2) Is *A* a concept intent, then we find the lectically next intent, by checking all attributes i ∈ *M* \ *A* , starting with the largest, und then in decreasing order, until $A <_i (A \oplus i)''$ holds. Then $(A \oplus i)''$ is the lectically next concept intent.

3) If $(A \oplus i)'' = M$, then stop, else $A \leftarrow (A \oplus i)''$ and goto 2).

	Handy (1)	Telefon (2)	Fax (3)	Fax w. n. paper (4)
Sinus 44		X		
Nokia 6110	X	X		
T-Fax 301			X	X
T-Fax 360 PC			X	

Example: on blackboard

A	i	$A \oplus i$	$(A \oplus i)$“	A $<_i$ $(A \oplus i)$“ ?	new concept intent

TITANIC vs. Next-Closure

- Next-Closure needs almost no memory.
- Next-Closure can exploit known symmetries between attributes.
- Next-Closure can be used for knowledge acquisition.
- TITANIC has far better performance, especially on large data sets.

1. Introduction
2. Formal Contexts & Concept Lattices
3. Application Examples I
4. Computing Concept Lattices
5. **Exercises**
6. Conceptual Scaling
7. Application Examples II
8. Conceptual Clustering
9. FCA-Based Mining of Association Rules
10. FCA Tools
11. Exercises

Exercise: Compute the concept lattices of the following formal contexts

	round	polygonal	rectangular
speed limit sign	×		
one way sign		×	×
stop sign		×	

	young	medium	old
y	×		
m		×	
o			×

	male	female
m	×	
f		×

	blue	red	yellow
orange		×	×
green	×		×
violet	×	×	

	edible (e)	poisonous (p)	cap shape: convex (c)	cap shape: flat (l)	cap surface: fibrous (i)
Mushroom 1	×		×		
Mushroom 2	×		×		×
Mushroom 3	×			×	×
Mushroom 4	×			×	×
Mushroom 5	×		×		×
Mushroom 6	×		×		
Mushroom 7		×		×	×
Mushroom 8		×		×	×
Mushroom 9		×		×	×
Mushroom 10		×		×	

Star Alliance Partners	Latin America	Europe	Canada	Asia Pacific	Middle East	Africa	Mexico	Caribbean	United States
Air Canada	×	×	×	×	×		×	×	×
Air New Zealand		×		×					×
All Nippon Airways		×		×					×
Ansett Australia				×					
The Austrian Airlines Group		×	×	×	×	×			×
British Midland		×							
Lufthansa	×	×	×	×	×	×	×		×
Mexicana	×		×				×	×	×
Scandinavian Airlines	×	×		×		×			×
Singapore Airlines		×	×	×	×	×			×
Thai Airways International	×	×		×				×	×
United Airlines	×	×	×	×			×	×	×
VARIG	×	×		×		×	×		×

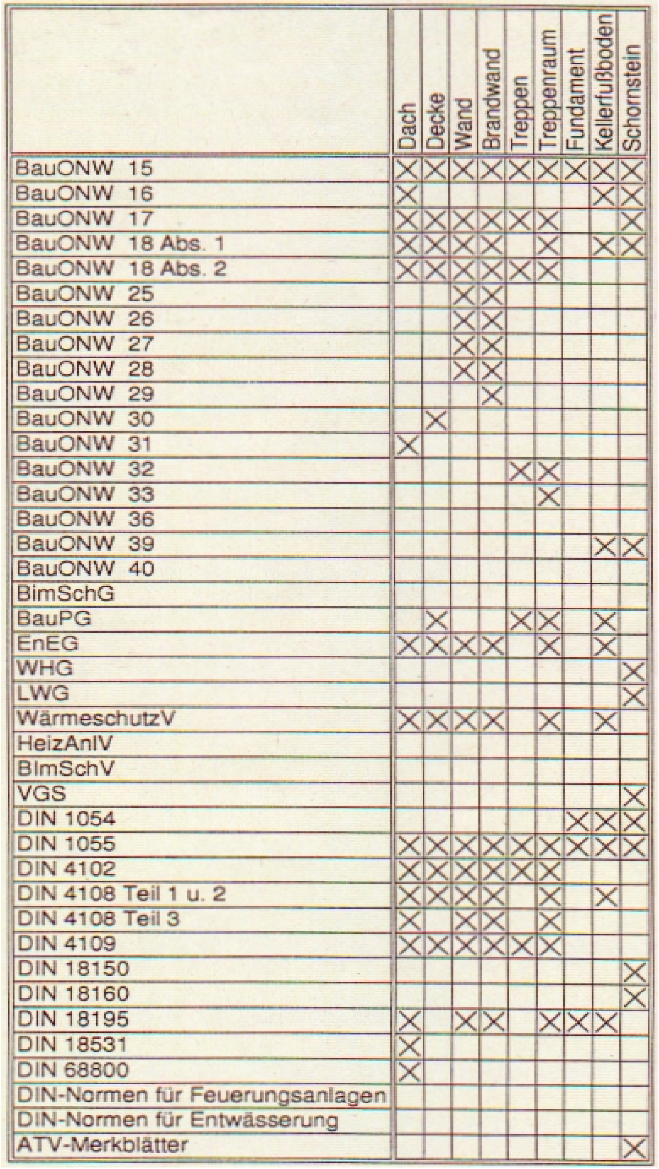
1. Introduction
2. Formal Contexts & Concept Lattices
3. Application Examples I
4. Computing Concept Lattices
5. Exercises
6. **Conceptual Scaling**
7. Application Examples II
8. Conceptual Clustering
9. FCA-Based Mining of Association Rules
10. FCA Tools
11. Exercises

Problem: Concept lattices can grow exponential in the size of the context.

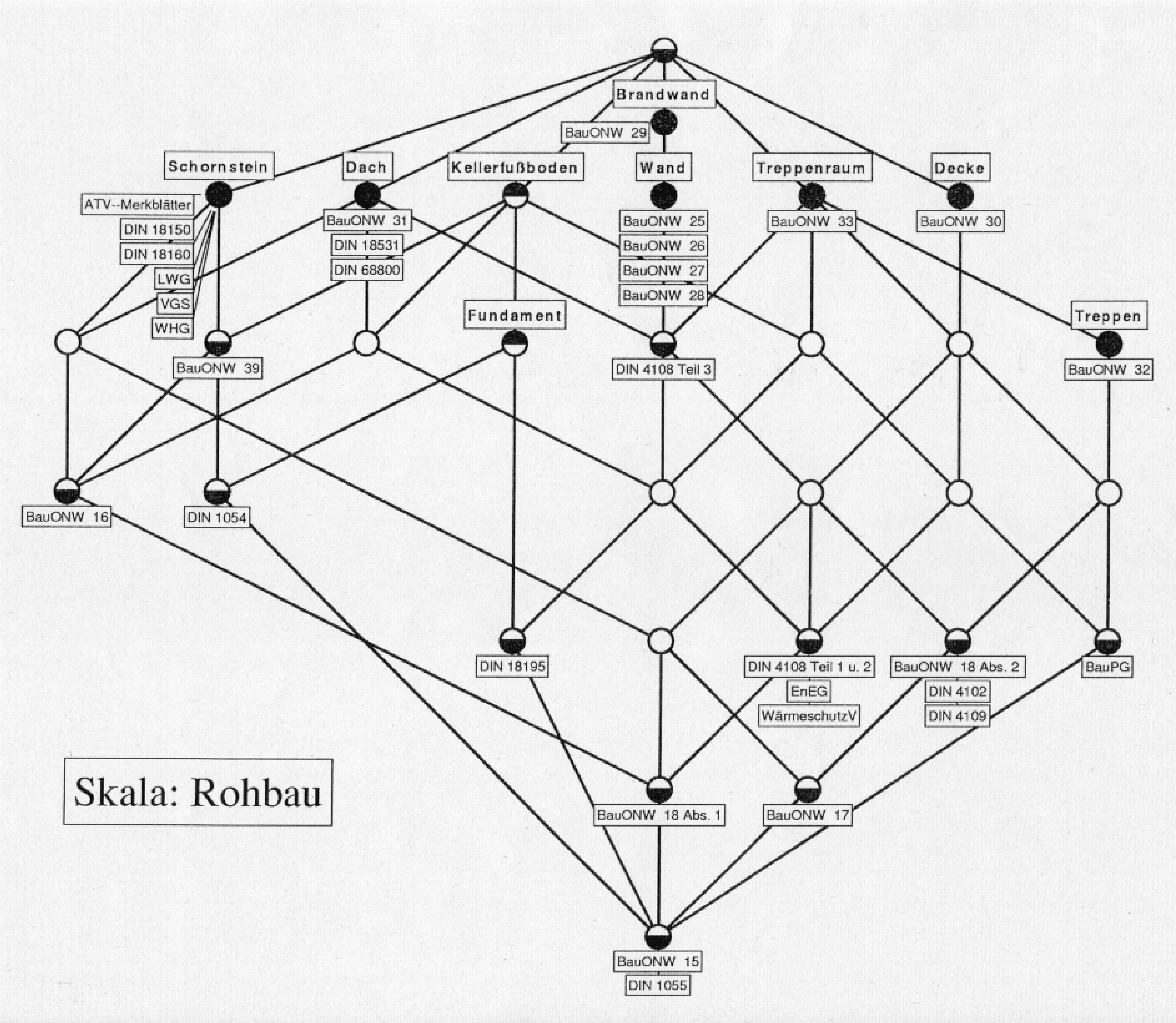
Answer:

- One method for reducing the complexity of the diagram is conceptual scaling.
- The idea is to consider only few attributes at a time.
- If combinations are of interest, they can be put together again.

Example: Civil Engineering regulations in Nordrhein-Westfalen



	Dach	Decke	Wand	Brandwand	Treppen	Treppenraum	Fundament	Kellerfußboden	Schornstein
BauONW 15	X	X	X	X	X	X	X	X	X
BauONW 16	X							X	X
BauONW 17	X	X	X	X	X	X			X
BauONW 18 Abs. 1	X	X	X	X		X		X	X
BauONW 18 Abs. 2	X	X	X	X	X	X			
BauONW 25			X	X					
BauONW 26			X	X					
BauONW 27			X	X					
BauONW 28			X	X					
BauONW 29				X					
BauONW 30		X							
BauONW 31	X								
BauONW 32					X	X			
BauONW 33						X			
BauONW 36									
BauONW 39								X	X
BauONW 40									
BimSchG									
BauPG		X			X	X		X	
EnEG	X	X	X	X		X		X	
WHG									X
LWG									X
WärmeschutzV	X	X	X	X		X		X	
HeizAnlV									
BImSchV									
VGS									X
DIN 1054							X	X	X
DIN 1055	X	X	X	X	X	X	X	X	X
DIN 4102	X	X	X	X	X	X			
DIN 4108 Teil 1 u. 2	X	X	X	X		X		X	
DIN 4108 Teil 3	X		X	X		X			
DIN 4109	X	X	X	X	X	X			
DIN 18150									X
DIN 18160									X
DIN 18195	X		X	X		X	X	X	
DIN 18531	X								
DIN 68800	X								
DIN-Normen für Feuerungsanlagen									
DIN-Normen für Entwässerung									
ATV-Merkblätter									X

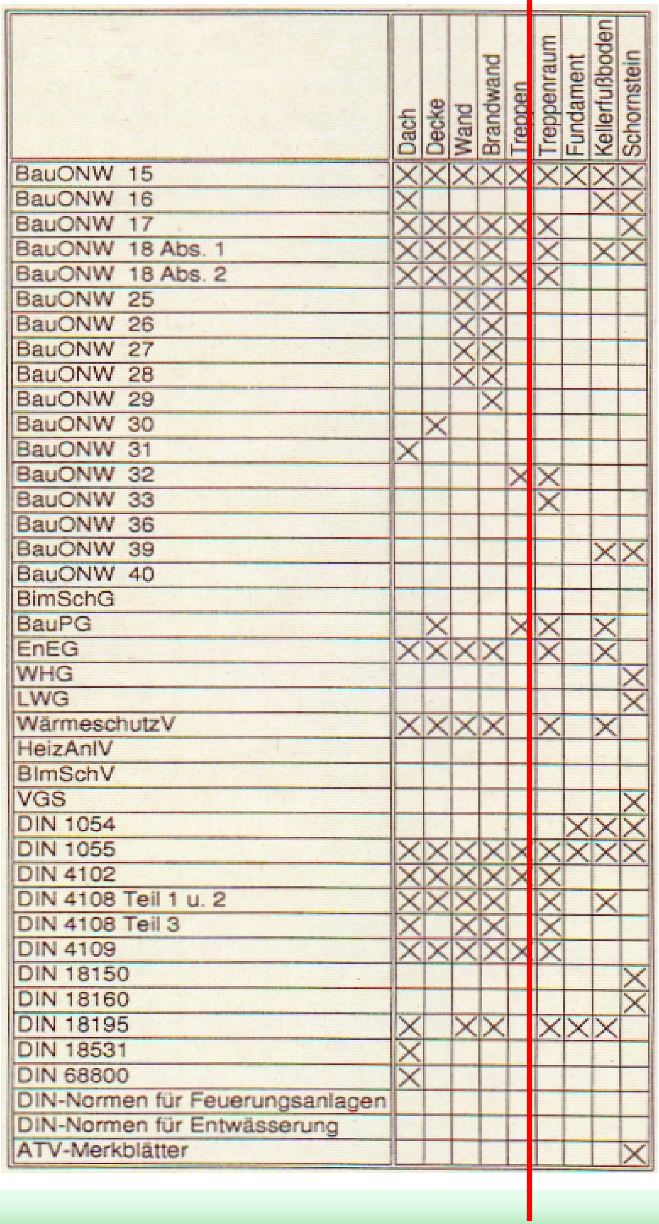

Brandwand
BauONW 29
Schornstein
Dach
Kellerfußboden
Wand
Treppenraum
Decke
ATV--Merkblätter
DIN 18150
DIN 18160
LWG
VGS
WHG
BauONW 31
DIN 18531
DIN 68800
BauONW 25
BauONW 26
BauONW 27
BauONW 28
BauONW 33
BauONW 30
Fundament
Treppen
BauONW 39
DIN 4108 Teil 3
BauONW 32
BauONW 16
DIN 1054
DIN 18195
DIN 4108 Teil 1 u. 2
EnEG
WärmeschutzV
BauONW 18 Abs. 2
DIN 4102
DIN 4109
BauPG
Skala: Rohbau
BauONW 18 Abs. 1
BauONW 17
BauONW 15
DIN 1055

Problem: Concept lattices can grow exponential in the size of the context.

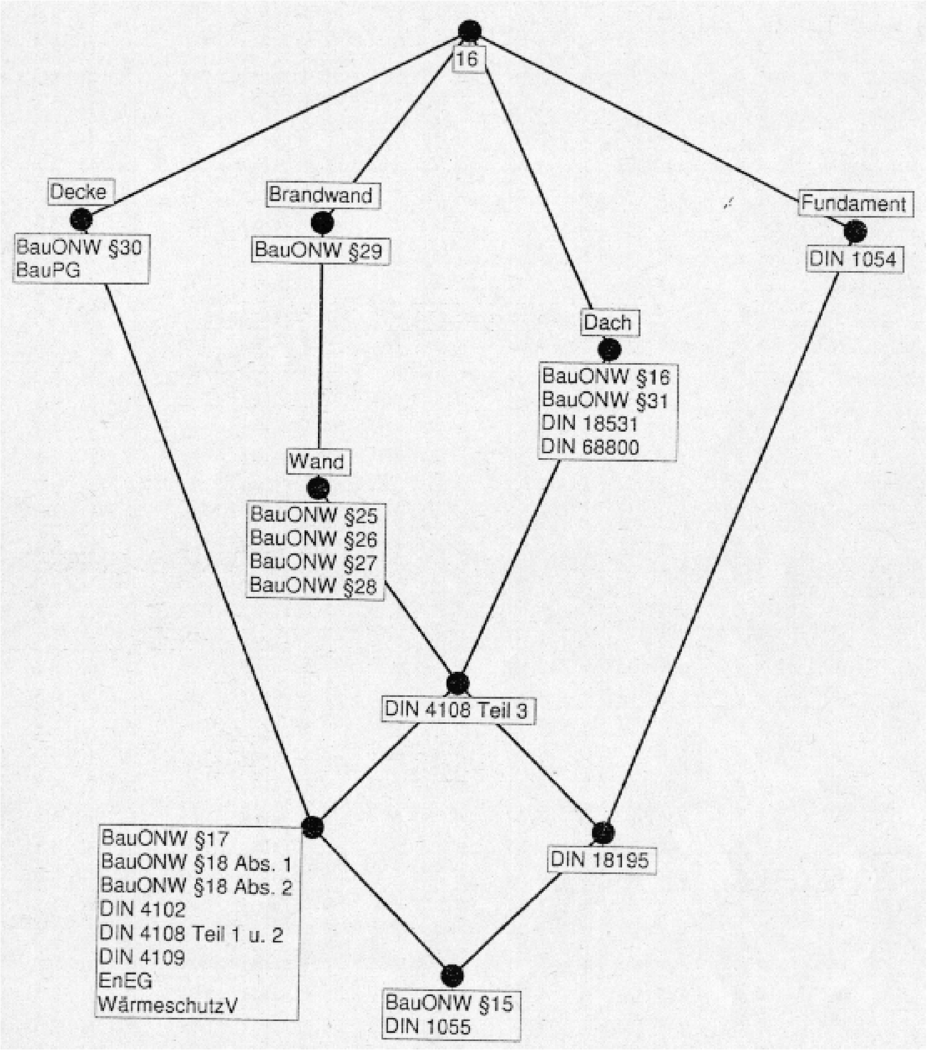
Answer:

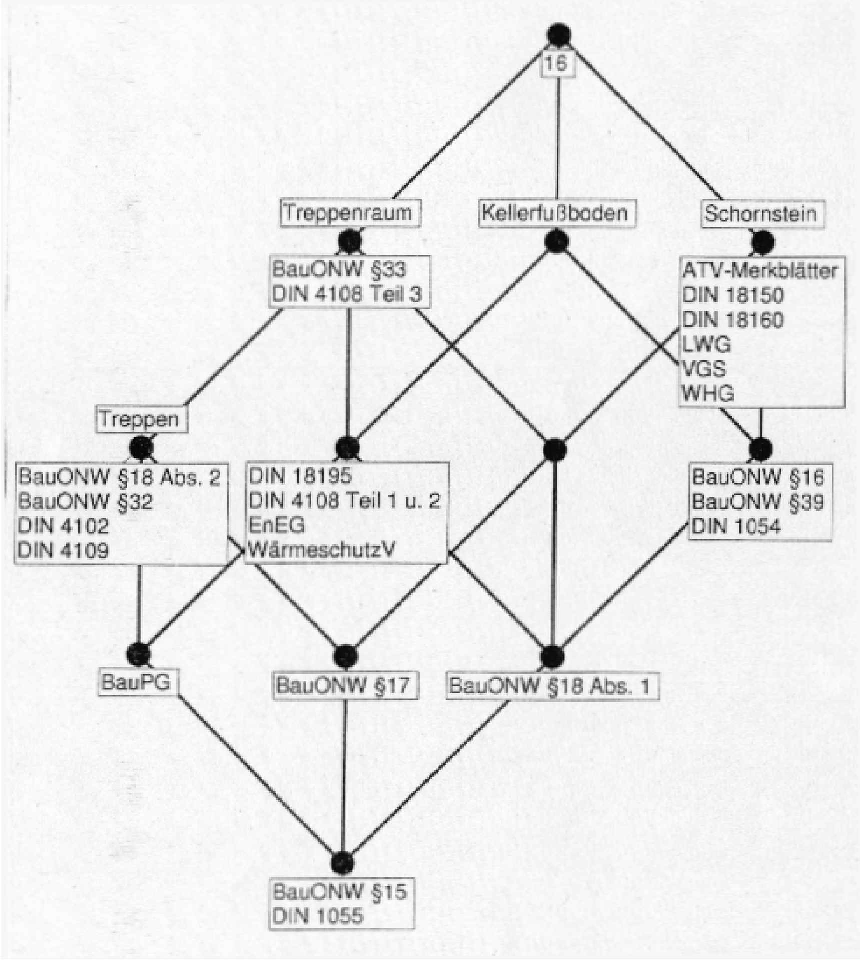
- One method for reducing the complexity of the diagram is conceptual scaling.
- The idea is to consider only few attributes at a time.
- If combinations are of interest, they can be put together again.

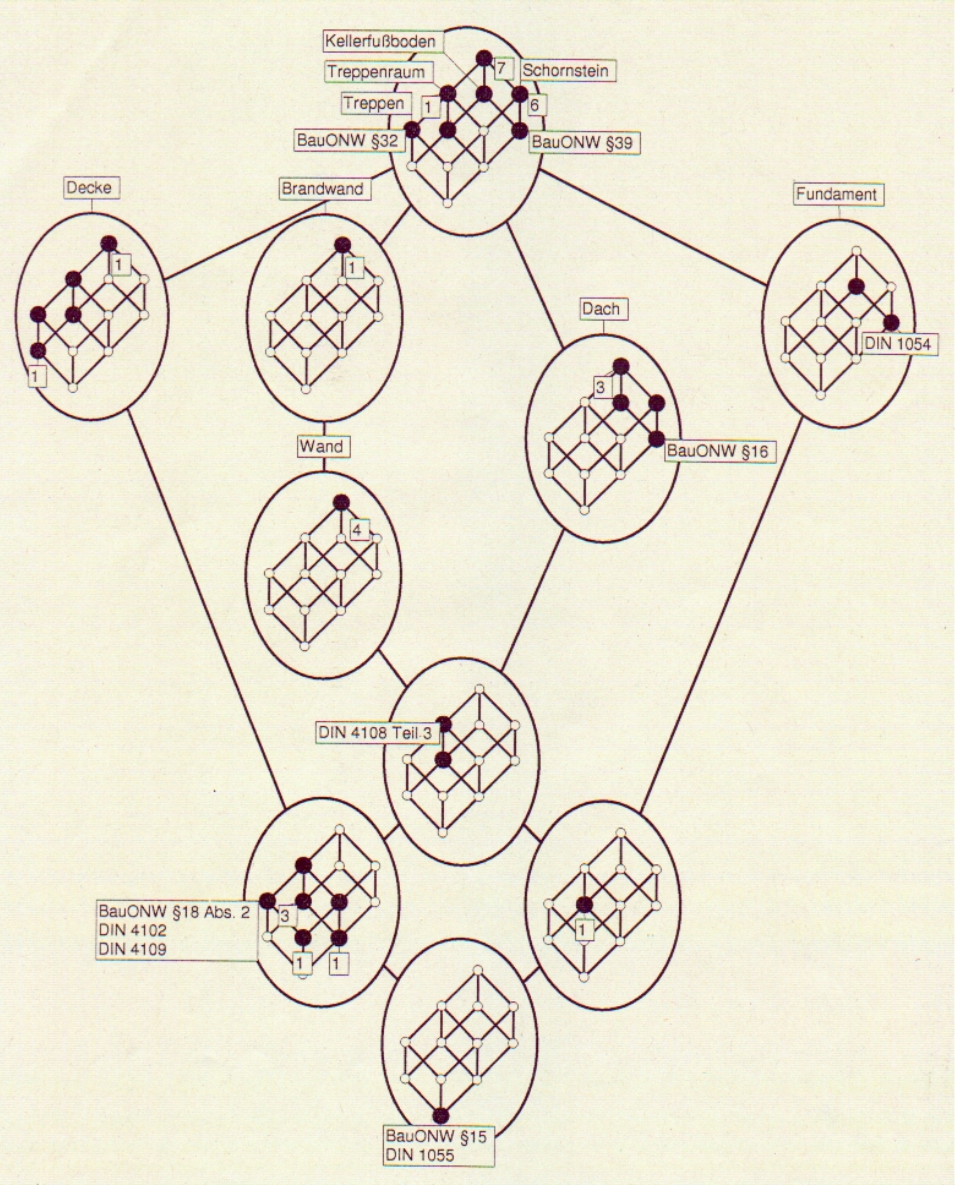
Example: Civil Engineering regulations in Nordrhein-Westfalen



	Dach	Decke	Wand	Brandwand	Treppen	Treppenraum	Fundament	Kellerfußboden	Schornstein
BauONW 15	X	X	X	X	X	X	X	X	X
BauONW 16	X							X	X
BauONW 17	X	X	X	X	X	X			X
BauONW 18 Abs. 1	X	X	X	X		X		X	X
BauONW 18 Abs. 2	X	X	X	X	X	X			
BauONW 25			X	X					
BauONW 26			X	X					
BauONW 27			X	X					
BauONW 28			X	X					
BauONW 29				X					
BauONW 30		X							
BauONW 31	X								
BauONW 32					X	X			
BauONW 33						X			
BauONW 36									
BauONW 39								X	X
BauONW 40									
BimSchG									
BauPG		X			X	X		X	
EnEG	X	X	X	X		X		X	
WHG									X
LWG									X
WärmeschutzV	X	X	X	X		X		X	
HeizAnlV									
BImSchV									
VGS									X
DIN 1054							X	X	X
DIN 1055	X	X	X	X	X	X	X	X	X
DIN 4102	X	X	X	X	X	X			
DIN 4108 Teil 1 u. 2	X	X	X	X		X		X	
DIN 4108 Teil 3	X		X	X		X			
DIN 4109	X	X	X	X	X	X			
DIN 18150									X
DIN 18160									X
DIN 18195	X		X	X		X	X	X	
DIN 18531	X								
DIN 68800	X								
DIN-Normen für Feuerungsanlagen									
DIN-Normen für Entwässerung									
ATV-Merkblätter									X


16
Decke
BauONW §30
BauPG
Brandwand
BauONW §29
Fundament
DIN 1054
Dach
BauONW §16
BauONW §31
DIN 18531
DIN 68800
Wand
BauONW §25
BauONW §26
BauONW §27
BauONW §28
DIN 4108 Teil 3
BauONW §17
BauONW §18 Abs. 1
BauONW §18 Abs. 2
DIN 4102
DIN 4108 Teil 1 u. 2
DIN 4109
EnEG
WärmeschutzV
DIN 18195
BauONW §15
DIN 1055


16
Treppenraum
Kellerfußboden
Schornstein
BauONW §33
DIN 4108 Teil 3
ATV-Merkblätter
DIN 18150
DIN 18160
LWG
VGS
WHG
Treppen
BauONW §18 Abs. 2
BauONW §32
DIN 4102
DIN 4109
DIN 18195
DIN 4108 Teil 1 u. 2
EnEG
WärmeschutzV
BauONW §16
BauONW §39
DIN 1054
BauPG
BauONW §17
BauONW §18 Abs. 1
BauONW §15
DIN 1055


Kellerfußboden
Treppenraum
Treppen
BauONW §32
7
Schornstein
1
6
BauONW §39
Decke
Brandwand
Fundament
1
1
1
Dach
3
DIN 1054
BauONW §16
Wand
4
DIN 4108 Teil 3
BauONW §18 Abs. 2
DIN 4102
DIN 4109
3
1
1
1
BauONW §15
DIN 1055

Many-valued Contexts and Conceptual Scaling

In general, attributes may not only be properties which are or are not related to an object, but they may allow for different values. We call such attributes, as e.g. „color“, „sexe“, „weight“, **many-valued attributes**.

Def.: A **many-valued context** (G, M, W, I) consists of sets G, M, and W and ternary relation I between G, M and W (i.e. I$\subseteq$ G x M x W), where the following holds:

$$(g, m, w) \in I \text{ and } (g, m, v) \in I \text{ imply } w = v.$$

The elements of G are called **objects**, the elements of M **(many-valued) attributes** and the elements of W **attribute values**.

(g, m, w) $\in$ I is read as „attribute m has value w for object g“.

Many-valued attributes can be considered as partial mappings from G to W, hence we note m(g) = w instead of (g, m, w) $\in$ I.

Example.: This many-valued context lists different drive concepts for cars:.

Conventional
Front-Wheel
Rear-Wheel
Mid-Engine
All-Wheel

	De	Dl	R	S	E	C	M
Conventional	poor	good	good	understeering	good	medium	excellent
Front-wheel	good	poor	excellent	understeering	excellent	very low	good
Rear-wheel	excellent	excellent	very poor	oversteering	poor	low	very poor
Mid-engine	excellent	excellent	good	neutral	very poor	low	very poor
All-wheel	excellent	excellent	good	understeering/neutral	good	high	poor

In: Antriebskonzept für Personenkraftwagen. Quelle: Schlag nach! 100 000 Tatsachen aus allen Wissenschaftsgebieten. BI-Verlag Mannheim, 1982

How to derive concepts from many-valued contexts?

- The many-valued context is transformed by **conceptual scaling** (as described below) to a one-valued context, for which one then can compute formal concepts.

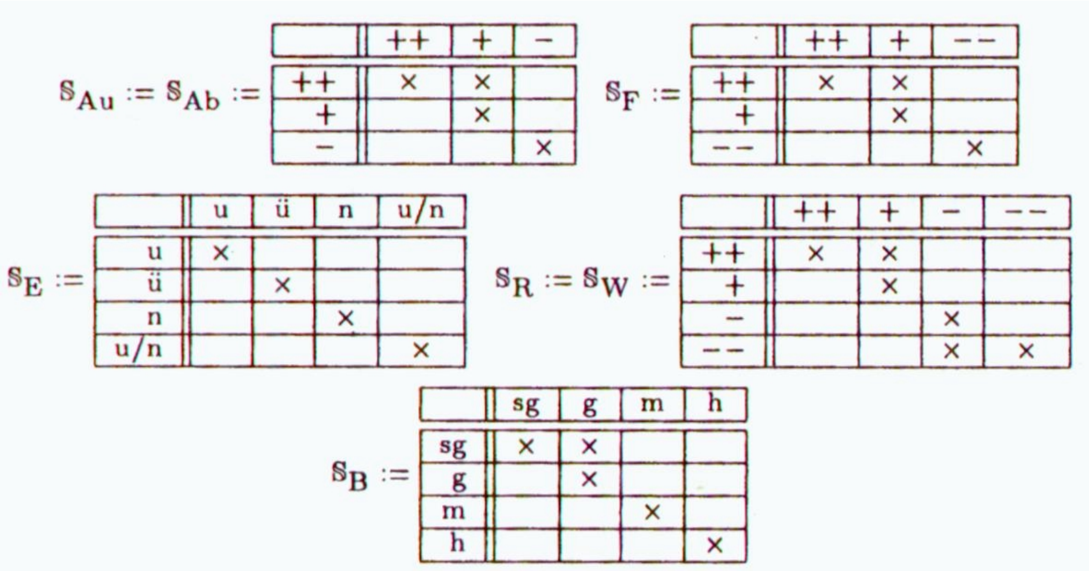
- Conceptual Scaling involves the human expert, as s/he has several choices how to interpret the data:

- For scaling, each attribute of the many-valued context is represented by a formal context, called **conceptual scale**.

Def.: A **(conceptual) scale** for attribute m of the many-valued context is a (one-valued) context $S_m := (G_m, M_m, I_m)$ with $m(G) \subseteq G_m$. The attributes of a scale are called **scale values**, the attributes **scale attributes**.

Plain Scaling

Using the following conceptual scales, we obtain the *derived context* on the following slide:



$\mathbb{S}_{Au} := \mathbb{S}_{Ab} :=$

	++	+	−
++	×	×	
+		×	
−			×

$\mathbb{S}_F :=$

	++	+	−−
++	×	×	
+		×	
−−			×

$\mathbb{S}_E :=$

	u	ü	n	u/n
u	×			
ü		×		
n			×	
u/n				×

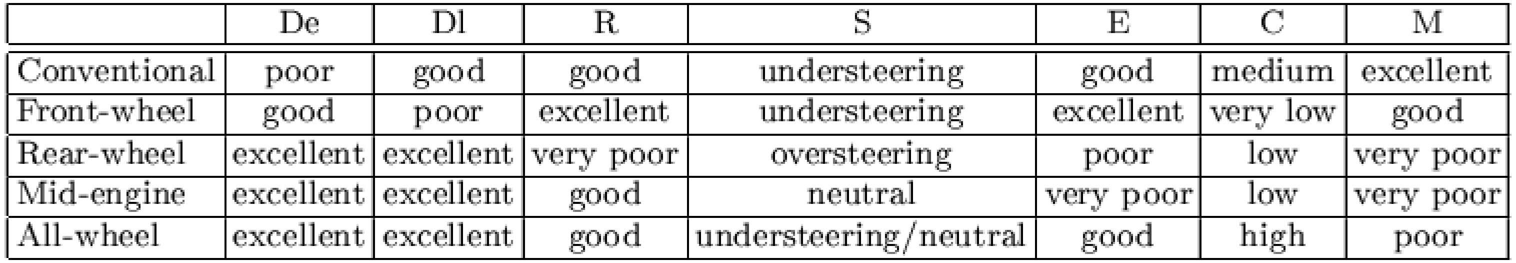
$\mathbb{S}_R := \mathbb{S}_W :=$

	++	+	−	−−
++	×	×		
+		×		
−			×	
−−			×	×

$\mathbb{S}_B :=$

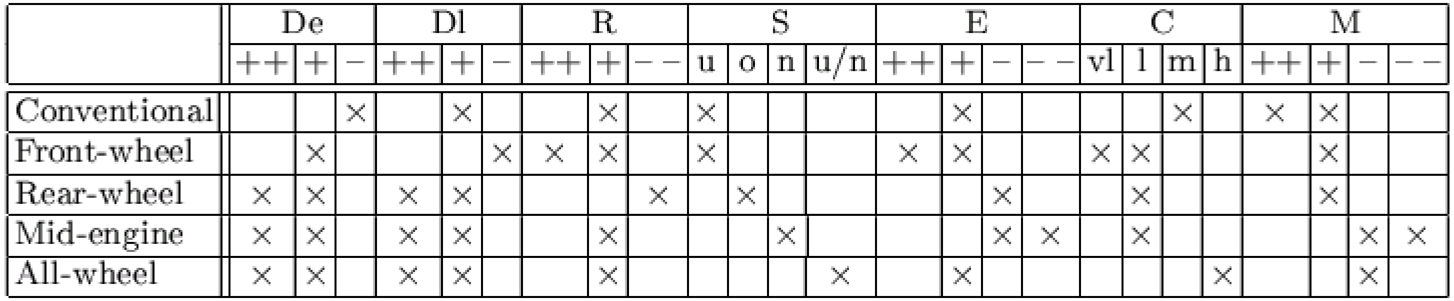
	sg	g	m	h
sg	×	×		
g		×		
m			×	
h				×

Tutorial Formal Concept Analysis



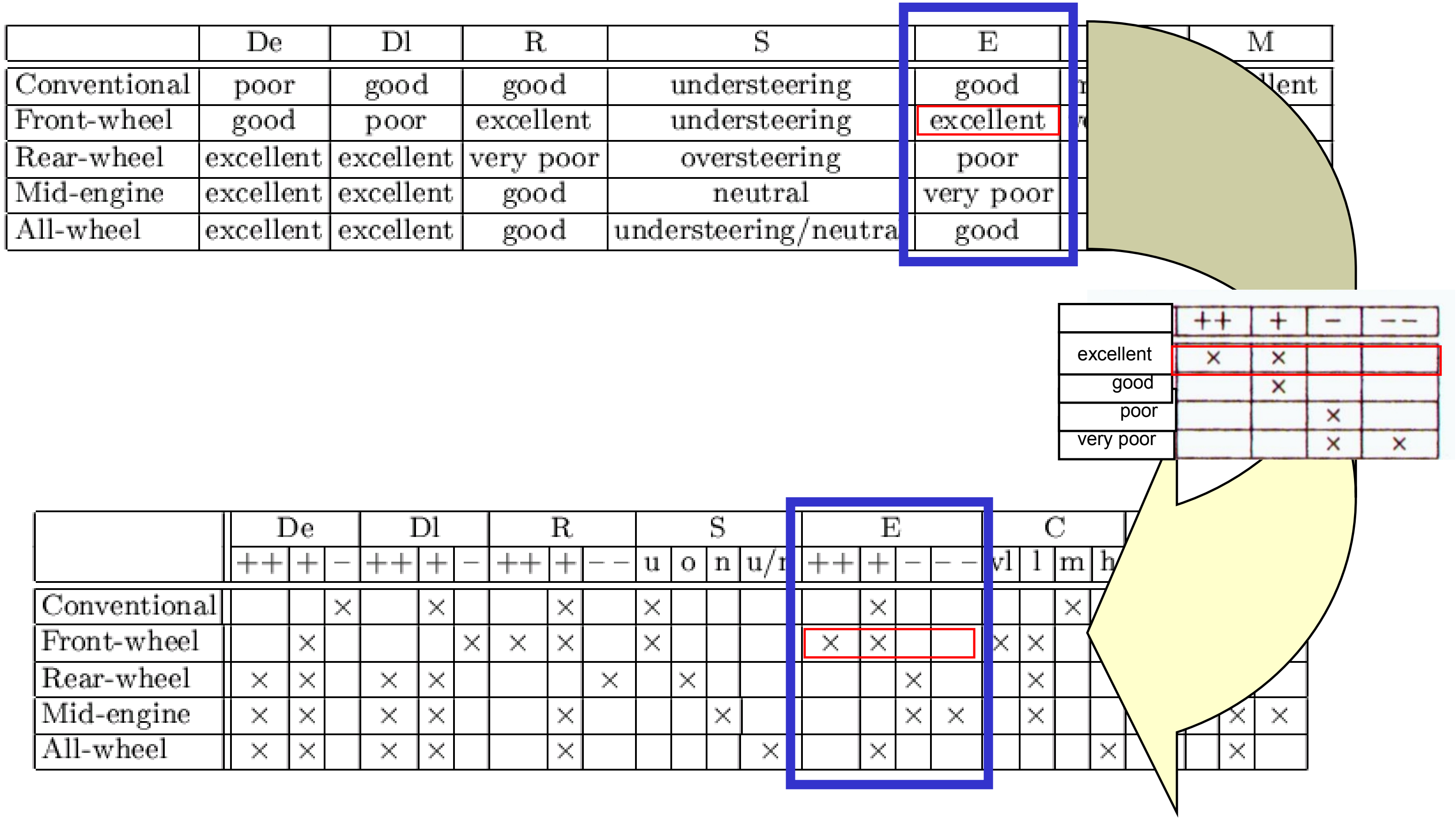
	De	Dl	R	S	E	C	M
Conventional	poor	good	good	understeering	good	medium	excellent
Front-wheel	good	poor	excellent	understeering	excellent	very low	good
Rear-wheel	excellent	excellent	very poor	oversteering	poor	low	very poor
Mid-engine	excellent	excellent	good	neutral	very poor	low	very poor
All-wheel	excellent	excellent	good	understeering/neutral	good	high	poor

From the many-valued context at the top, we obtain the following *derived context*:



	De			Dl			R			S				E				C				M			
	++	+	−	++	+	−	++	+	− −	u	o	n	u/n	++	+	−	− −	vl	l	m	h	++	+	−	− −
Conventional			×		×			×		×					×					×		×	×		
Front-wheel		×				×	×	×		×				×	×			×	×				×		
Rear-wheel	×	×		×	×				×		×					×			×				×		
Mid-engine	×	×		×	×			×				×				×	×		×					×	×
All-wheel	×	×		×	×			×					×		×						×			×	

De := drive efficiency empty; Dl := drive efficiency loaded; R := road holding/handling properties;
S := self-steering effect; E := economy of space; C := cost of construction; M := maintainability
++ := excellent; + := good; − := poor; − − := very poor; u := understeering;
o := oversteering; n := neutral; vl := very low; l := low, m := medium; h := high.



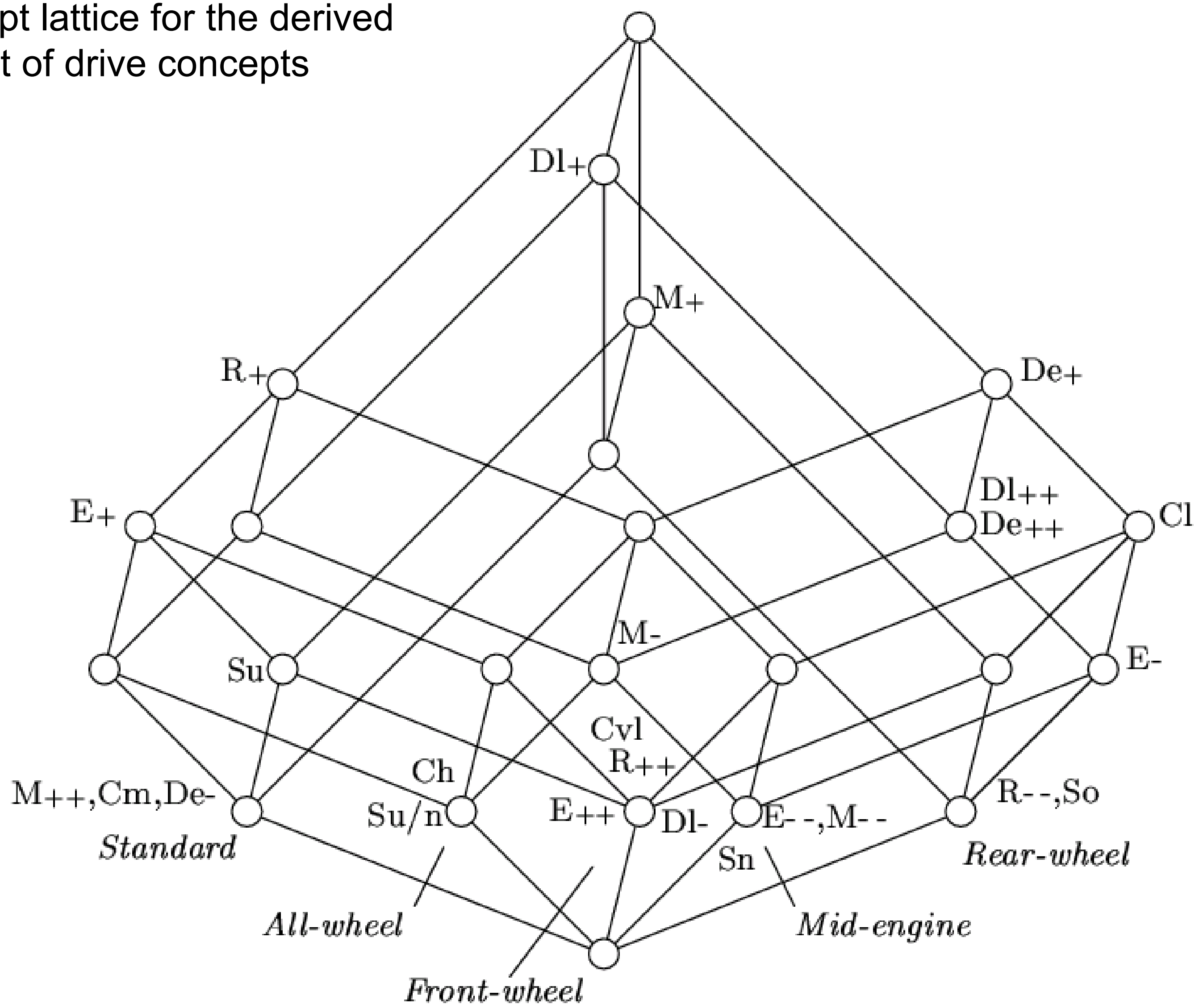
	De	Dl	R	S	E		M
Conventional	poor	good	good	understeering	good	m	llent
Front-wheel	good	poor	excellent	understeering	excellent	v	
Rear-wheel	excellent	excellent	very poor	oversteering	poor		
Mid-engine	excellent	excellent	good	neutral	very poor		
All-wheel	excellent	excellent	good	understeering/neutra	good		

	++	+	−	− −
excellent	×	×		
good		×		
poor			×	
very poor			×	×

	De			Dl			R			S				E				C						
	++	+	−	++	+	−	++	+	− −	u	o	n	u/r	++	+	−	− −	vl	l	m	h			
Conventional			×		×			×		×					×					×				
Front-wheel		×				×	×	×		×				×	×			×	×					
Rear-wheel	×	×		×	×				×		×					×			×					
Mid-engine	×	×		×	×			×				×				×	×		×				×	×
All-wheel	×	×		×	×			×					×		×						×		×	

De := drive efficiency empty; Dl := drive efficiency loaded; R := road holding/handling properties; S := self-steering effect; E := economy of space; C := cost of construction; M := maintainability
++ := excellent; + := good; − := poor; − − := very poor; u := understeering;
o := oversteering; n := neutral; vl := very low; l := low, m := medium; h := high.

Concept lattice for the derived context of drive concepts



Plain Scaling

As in the example above one obtains from a many-valued context (G, M, W, I) and the conceptual scales S_m, $m \in M$, the **derived context** as follows:

The object set G remains unchanged. Every many-valued attribute m is replaced by the scale attributes of the scale $S_{m.}$ Every attribute value m(g) is replaced by the corresponding row of the scale context S_m.

The formal definition is on the next slide.

Def.: For a many-valued context (G, M, W, I) and scale contexts S_m, $m \in M$, the **derived context** is (G, N, J) with

$$N := \bigcup_{m \in M} M_m,$$

and

$$gJ(m, n) :\Leftrightarrow (\, m(g) = w \text{ and } wI_m n) \, .$$

(M_m stands for $\{ m \} \times M_m$ in order to distinguish attribute values of different many-valued attributes.)

Any context can be a scale, there is no formal difference. However, we will call only those contexts ‚scales' which have a clear conceptual structure. Some very simple contexts are often used as scales:

Def.: Elementary Scales

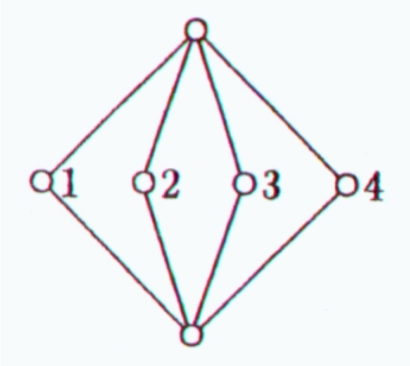
We use the abbreviation **n** := { 1, ..., n }

Nominal scales $\mathbf{N}_n$:= (**n**, **n**, =)

are used for scaling attributes whose values exclude each other. (E.g., an attribute having the values masculine, feminine, neuter will be scaled nominally.) Then the concept extents are a **partition** of the object set.

	1	2	3	4
1	x			
2		x		
3			x	
4				x

Die nominal scale $\mathbf{N}_4$

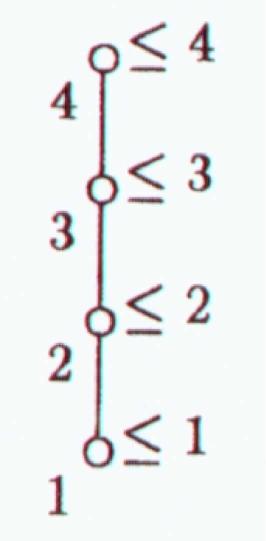


Ordinal scales $\mathbf{O_n := (n, n, \leq)}$

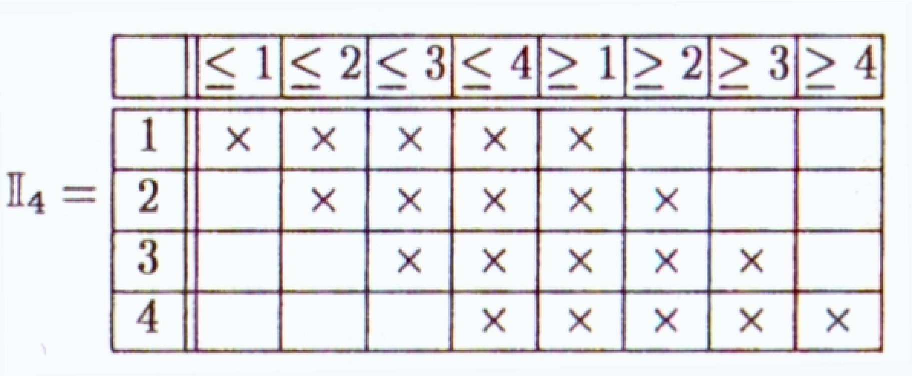
are used for attributes with ordered values, where each value implies the smaller values. (E.g., loud, very loud, extremely loud.) The result is a chain of concept extents which can be interpreted as **ranking**.

	1	2	3	4
1	x	x	x	x
2		x	x	x
3			x	x
4				x

The ordinal scale O_4

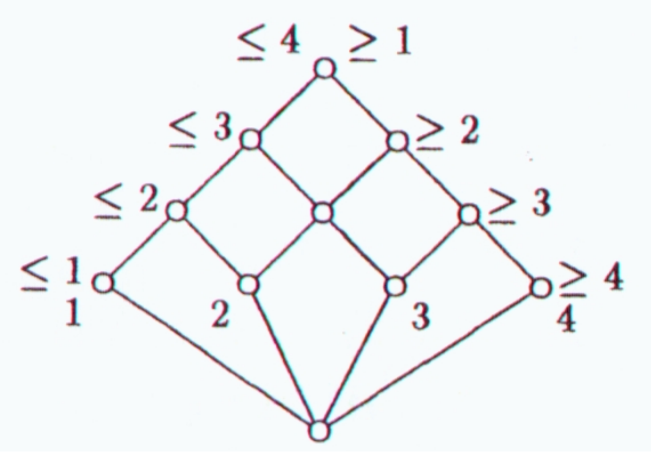


Interordinal scales $I_n := (n, n, \leq) \mid (n, n, \geq)$
are e.g. used in questionaries where one can select values on a scale like *activ-passiv* or *agree-disagree*. The concept intents are exactly the **intervals** of scale values - this reflects conceptually the **between relation**.



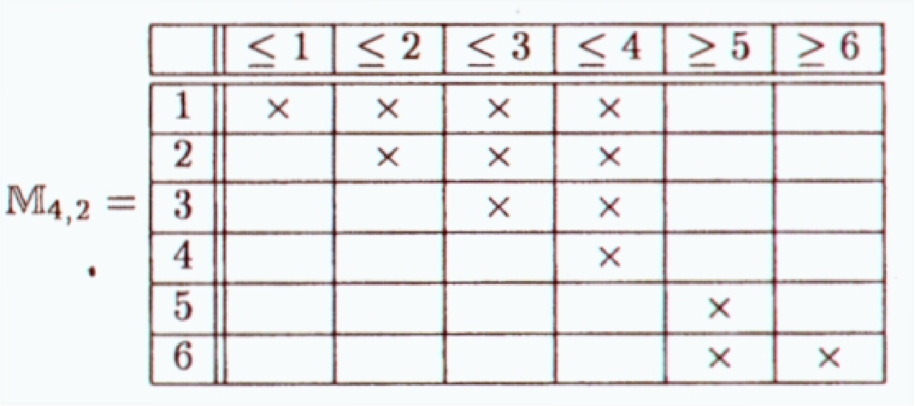
$\mathbb{I}_4 =$

	≤ 1	≤ 2	≤ 3	≤ 4	≥ 1	≥ 2	≥ 3	≥ 4
1	×	×	×	×	×			
2		×	×	×	×	×		
3			×	×	×	×	×	
4				×	×	×	×	×



Often these attributes can also be scaled bi-ordinally:

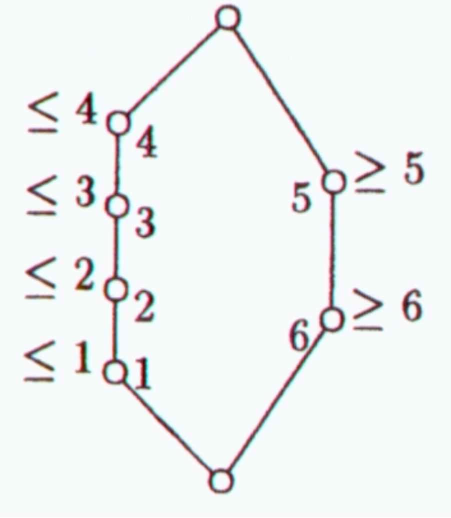
Biordinal scales $\mathbf{M}_{n,m} := (n, n, \leq) \cup (m, m, \geq)$
are used when the objects are assigned to one of two poles, and this with a different degree. (E.g., *very silent, silent, loud, very loud*: loud and silent exlude each other, very loud implies loud, and very silent implies silent.) The result is a **partition with ranking.**



$\mathbb{M}_{4,2} =$

	≤ 1	≤ 2	≤ 3	≤ 4	≥ 5	≥ 6
1	×	×	×	×		
2		×	×	×		
3			×	×		
4				×		
5					×	
6					×	×

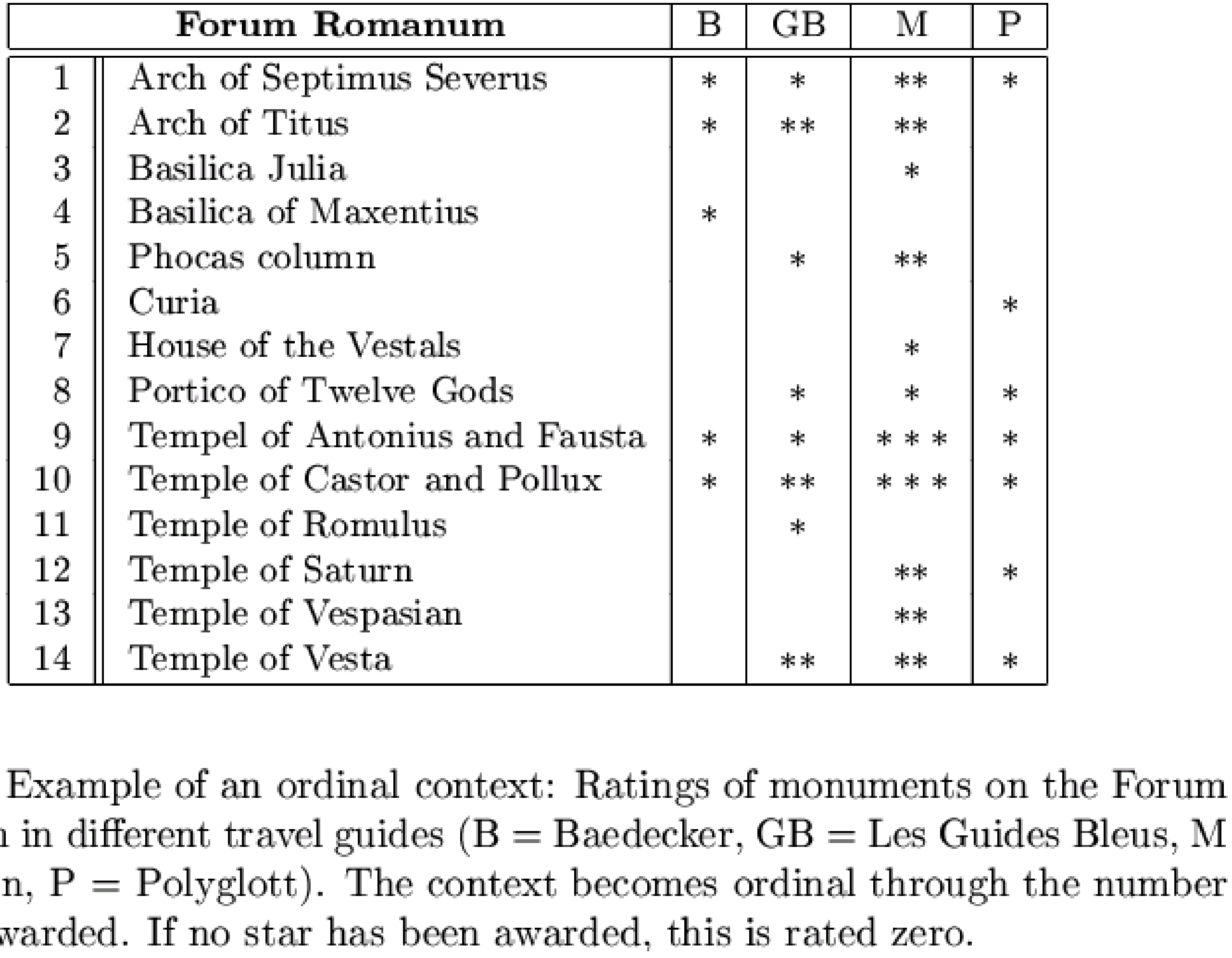
The bi-ordinal scale $O_{4,2}$ (e.g. for German school grades)



The **dichotomic scale** D := (0, 1 , 0,1 , =)
is a special case, since it is isomorphic to the scales N_2 und $M_{1,1}$ and closely related to I_2 . It is used most often for scaling **yes-no**.

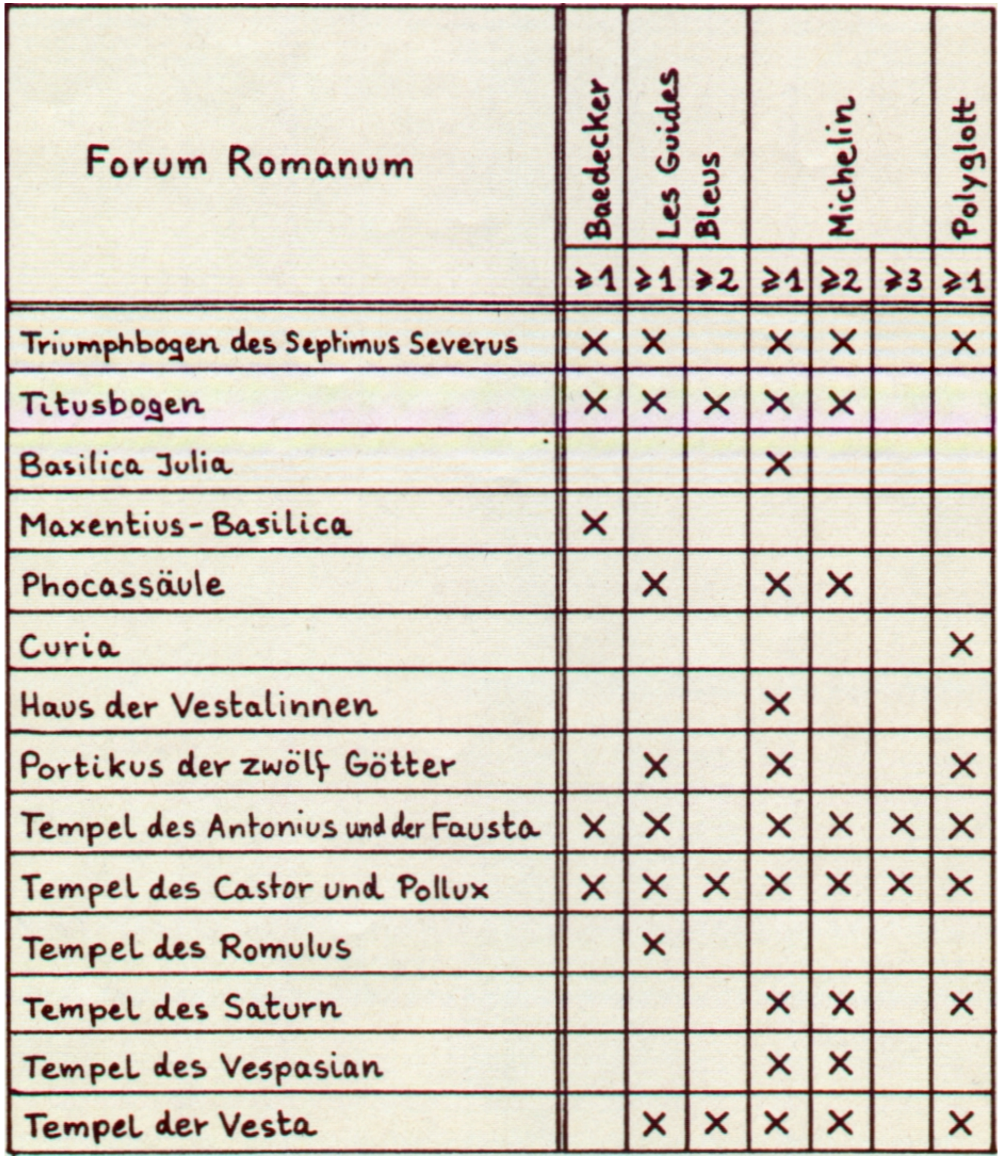
	0	1
0	x	
1		x

Example: This context will be scaled ordinally.



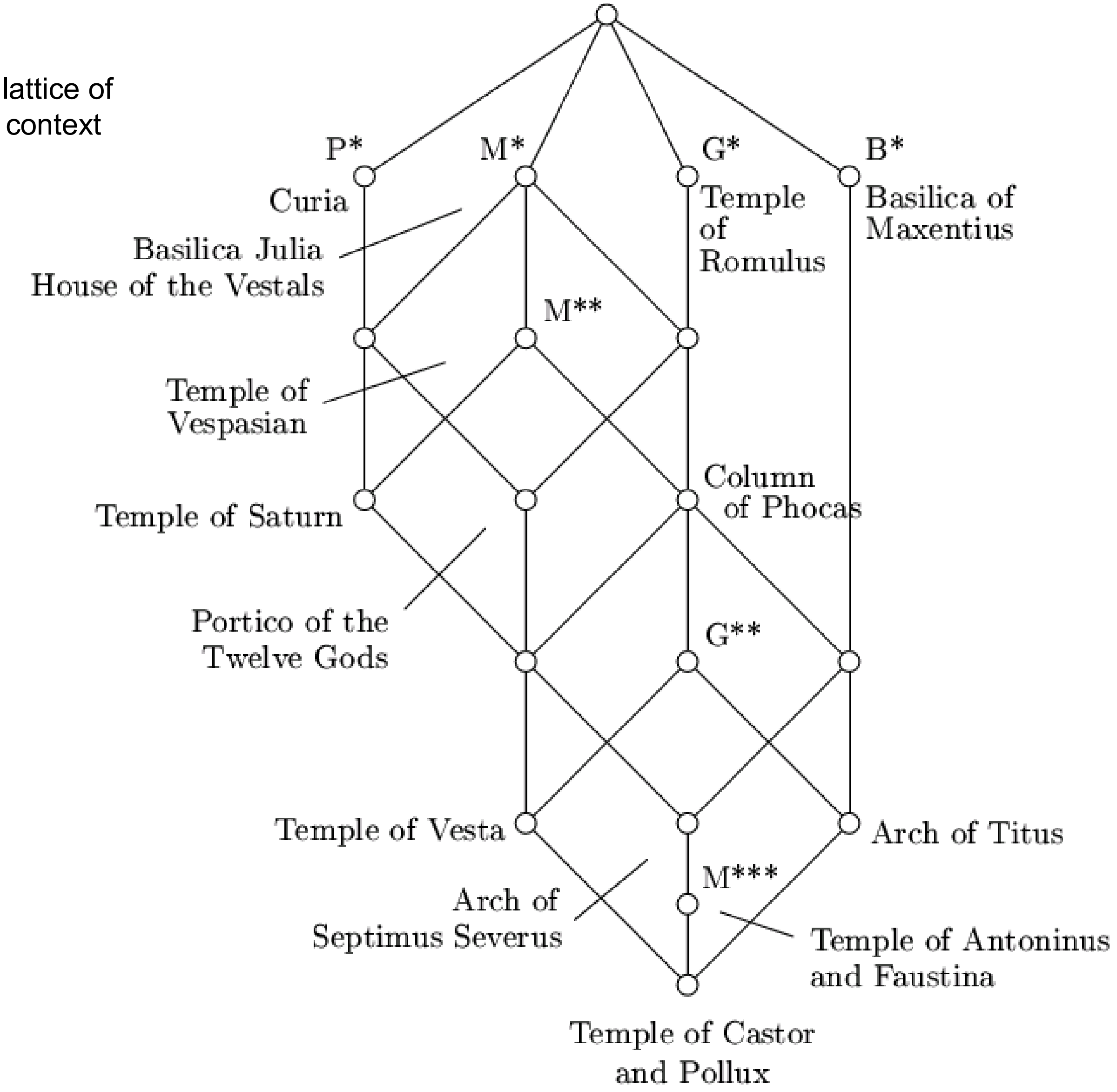
Forum Romanum		B	GB	M	P
1	Arch of Septimus Severus	*	*	**	*
2	Arch of Titus	*	**	**	
3	Basilica Julia			*	
4	Basilica of Maxentius	*			
5	Phocas column		*	**	
6	Curia				*
7	House of the Vestals			*	
8	Portico of Twelve Gods		*	*	*
9	Tempel of Antonius and Fausta	*	*	* * *	*
10	Temple of Castor and Pollux	*	**	* * *	*
11	Temple of Romulus		*		
12	Temple of Saturn			**	*
13	Temple of Vespasian			**	
14	Temple of Vesta		**	**	*

Figure 4: Example of an ordinal context: Ratings of monuments on the Forum Romanum in different travel guides (B = Baedecker, GB = Les Guides Bleus, M = Michelin, P = Polyglott). The context becomes ordinal through the number of stars awarded. If no star has been awarded, this is rated zero.



Forum Romanum	Baedecker	Les Guides Bleus		Michelin			Polyglott
	≥1	≥1	≥2	≥1	≥2	≥3	≥1
Triumphbogen des Septimus Severus	×	×		×	×		×
Titusbogen	×	×	×	×	×		
Basilica Julia				×			
Maxentius-Basilica	×						
Phocassäule		×		×	×		
Curia							×
Haus der Vestalinnen				×			
Portikus der zwölf Götter		×		×			×
Tempel des Antonius und der Fausta	×	×		×	×	×	×
Tempel des Castor und Pollux	×	×	×	×	×	×	×
Tempel des Romulus		×					
Tempel des Saturn				×	×		×
Tempel des Vespasian				×	×		
Tempel der Vesta		×	×	×	×		×

The concept lattice of
the derived context



Additive Line Diagrams

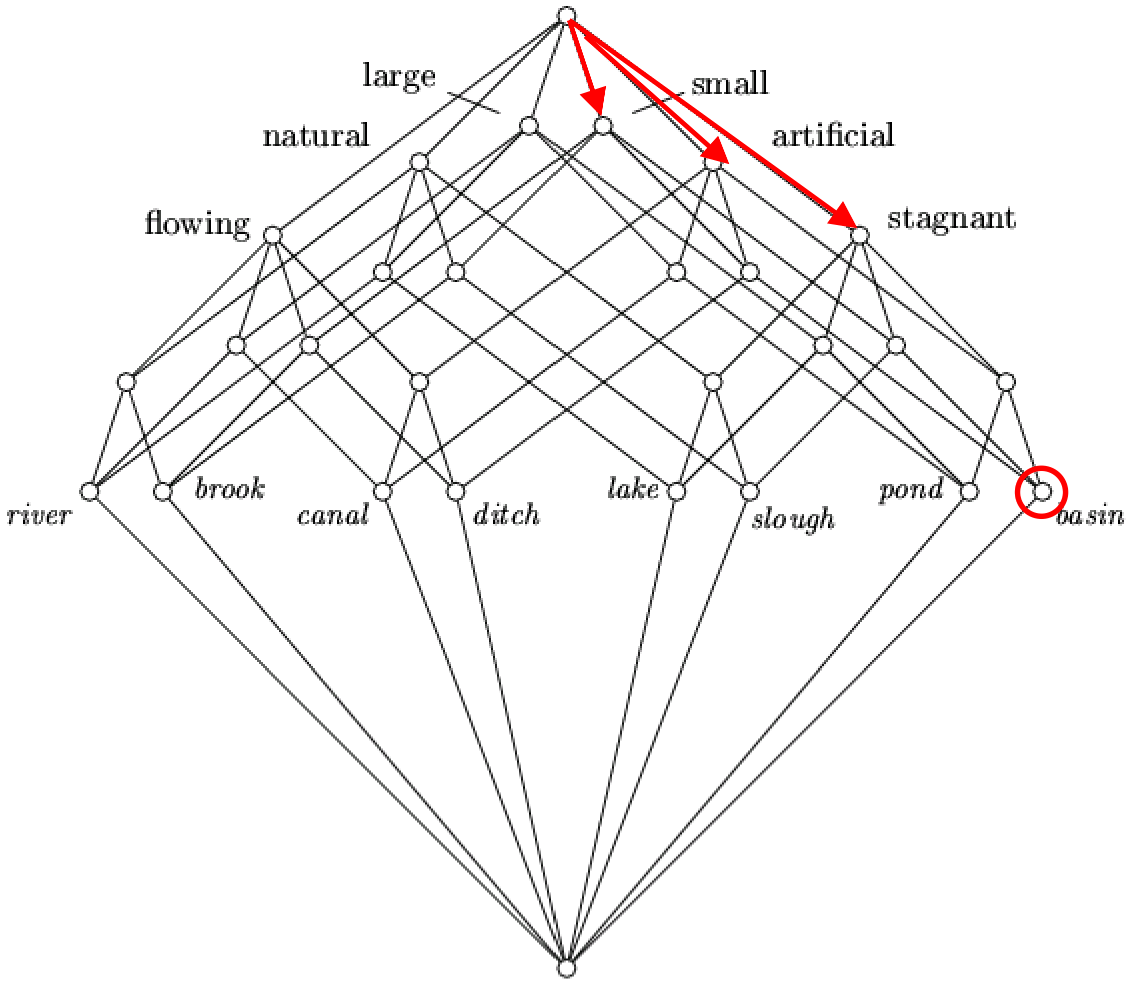


Figure 2: An additive line diagram of the concept lattice of a *lexical field* “*waters*”. The set representation is based on the irreducible attributes, i.e. the positioning of the attribute concepts determines that of all remaining concepts. If we interpret the line segments between the unit element and the attribute concepts as vectors, we obtain the position of an arbitrary concept by the sum of the vectors belonging to attributes of its concept intent starting from the unit

und in Figure ??.

Additive Line Diagrams

Def.: An attribute $m \in M$ is **irreducible**, if there are no other attributes m_1, $m_2 \in M$ with $m_1 \neq m \neq m_2$ and $m_1' \cap m_2' = m'$. The set of all irreducible attributes is denoted by M_{irr}

We define the mapping irr : $\underline{\boldsymbol{B}}(G,M,I) \to P(M_{ir})$ by

$$irr(A,B) := \{ m \in B \mid m \text{ irreducible} \}$$

Let vec : $M_{irr} \to \mathbf{R} \times \mathbf{R}_{<0.}$

Then pos : $\underline{\boldsymbol{B}}(G,M,I) \to \mathbf{R}^2$ with $pos(A,B) := \Sigma_{x\in\ irr(A,B)}$ vec(m) is an **additive line diagram** of the concept lattice $\underline{\boldsymbol{B}}(G,M,I)$.

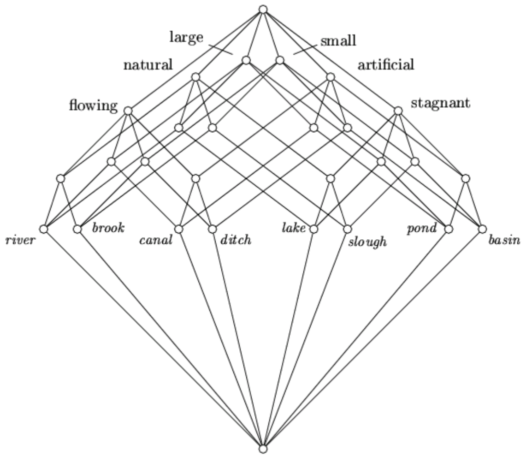
Nested Line Diagrams

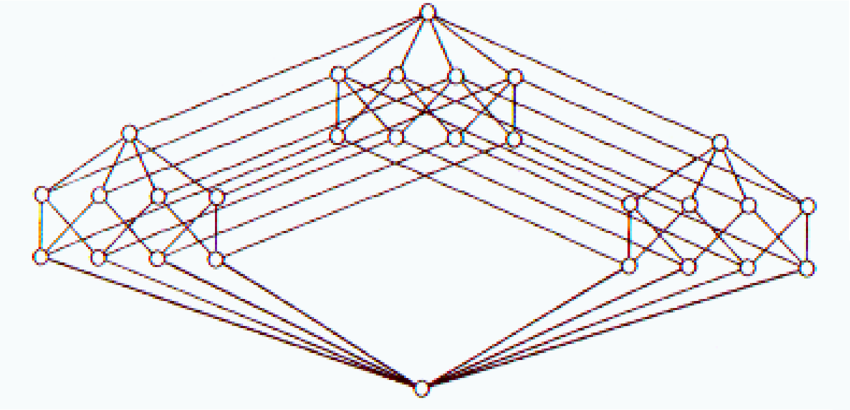
Nested line diagrams are used for

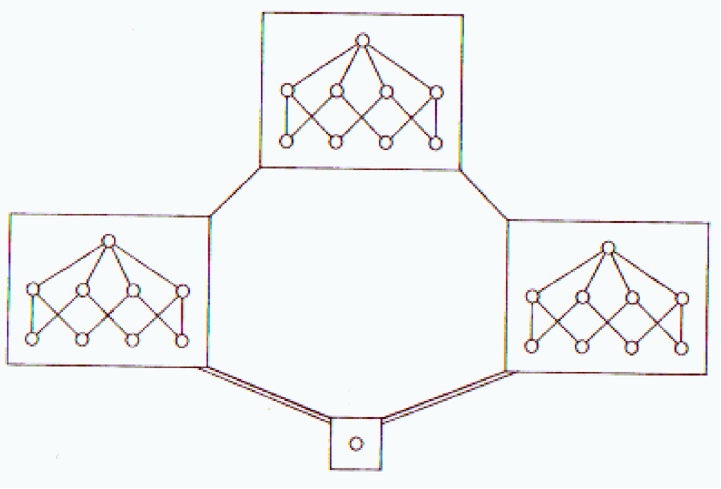
- for visualizing larger concept lattices
- for emphasizing sub-structures and regularities.
- for combining conceptual scales on-line.

The basic idea is to „summarize“ parallel lines and display it as just one line.

Nested Line Diagrams







These line diagrams all show the same concept lattice.

Nested Line Diagrams

A nested line diagram consists thus of an outer line diagram, which contains in each node inner diagrams.

In the simplest case the inner diagrams of two connected nodes of the outer diagram are congruent. The connecting line of the outer diagram indicates then that each node in an inner diagram is connected with the corresponding node in the other inner diagram.

A double line between two nodes indicates that each element within the upper node is larger than each element in the lower node.

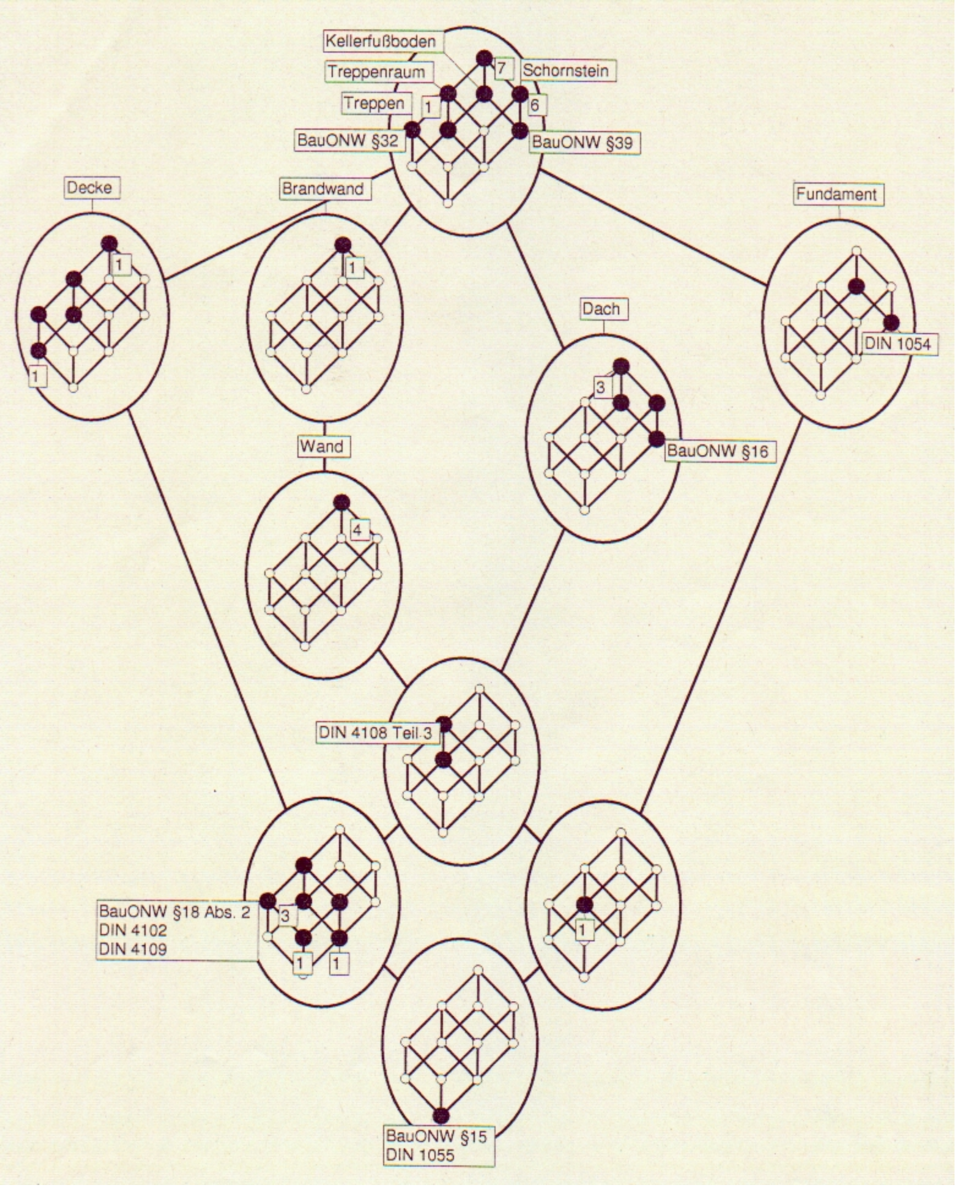
Nested Line Diagrams

We also allow that the inner diagrams are not congruent, but only substructures of congruent diagrams.

The congruent diagrams are then drawn as „background structure“, having some **unrealized concepts**.

Unrealized concepts indicate implications as we will see below.

Example for a nested line diagram with non-congruent components. (Details below)



Reading Implications in Nested Line Diagrams

- Implications within the inner scale are read from the inner diagram at the upmost concept:

{Treppen} → {Treppenraum} {stairs} → {staircase}

- Implications within the outer scale are read directly from it:

{Wand} → {Brandwand} {wall} → {firewall}

{Decke, Brandwand} → {Wand, Brandwand} {ceiling, firewall} → {wall, firewall}

{Decke, Fundament} → ? {ceiling, foundation} → ?

- Implications between inner and outer scale are indicated by non-realized concepts. The premise is the intent of the non-realized concept, and the conclusion is the intent of the largest realized subconcept:

{Decke, Kellerfußboden} → {Treppenraum} {ceiling, cave floor} → {staircase}

{Treppenraum, Schornstein} → {Decke, Wand, Brandwand, Dach}

{staircase, chimney} → {ceiling, wall, firewall, roof}

{Wand, Dach, Schornstein} → ? {wall, roof, chimney} → ?

Construction of Nested Line Diagrams

Def.: The least common superconcept of two concepts c_1 und c_2 is called **Supremum of c_1 and c_2** (denoted **$c_1 \vee c_2$**). The greatest common subconcept of c_1 and c_2 is the **Infimum von c_1 und c_2** (denoted **$c_1 \wedge c_2$**).
A mapping $f : V \to W$ between two lattices V and W is called **supremum-preserving**, if $f(x \vee y) = f(x) \vee f(y)$.

Remark.: If a mapping preserves suprema, it also preserves the partial order, since

$$x \leq y \quad \Leftrightarrow \quad x \vee y = y \quad \Rightarrow \quad f(x) \vee f(y) = f(x \vee y) = f(y) \quad \Leftrightarrow \quad f(x) \leq f(y)$$

Theorem: Let (G, M, I) be a context and $M = M_1 \cup M_2$. The mapping

$$(A, B) \to (((B \cap M_1)', B \cap M_1), ((B \cap M_2)', B \cap M_2))$$

is an supremum-preserving embedding of $\underline{\mathrm{B}}$(G, M, I) in the direct product

$$\underline{\mathrm{B}}(G, M_1, I \cap G \text{ x } M_1) \quad \underline{\mathrm{B}}(G, M_2, I \cap G \text{ x } M_2).$$

Construction of Nested Line Diagrams

- For constructing nested line diagrams, one first splits the attribute set: $M = M_1 \cup M_2$.
- The sets need not be disjoint, it is more important that they are grouped meaningfully.
- For many-valued contexts, the sets M_1 and M_2 the attribute sets of the conceptual scales.

- One draws the concept lattices of the smaller contexts

$$K_i := (G, M_i, I \cap G \times M_i),\ i \in\ 1, 2\ ,$$

and labels them with the objects and attributes as usual.

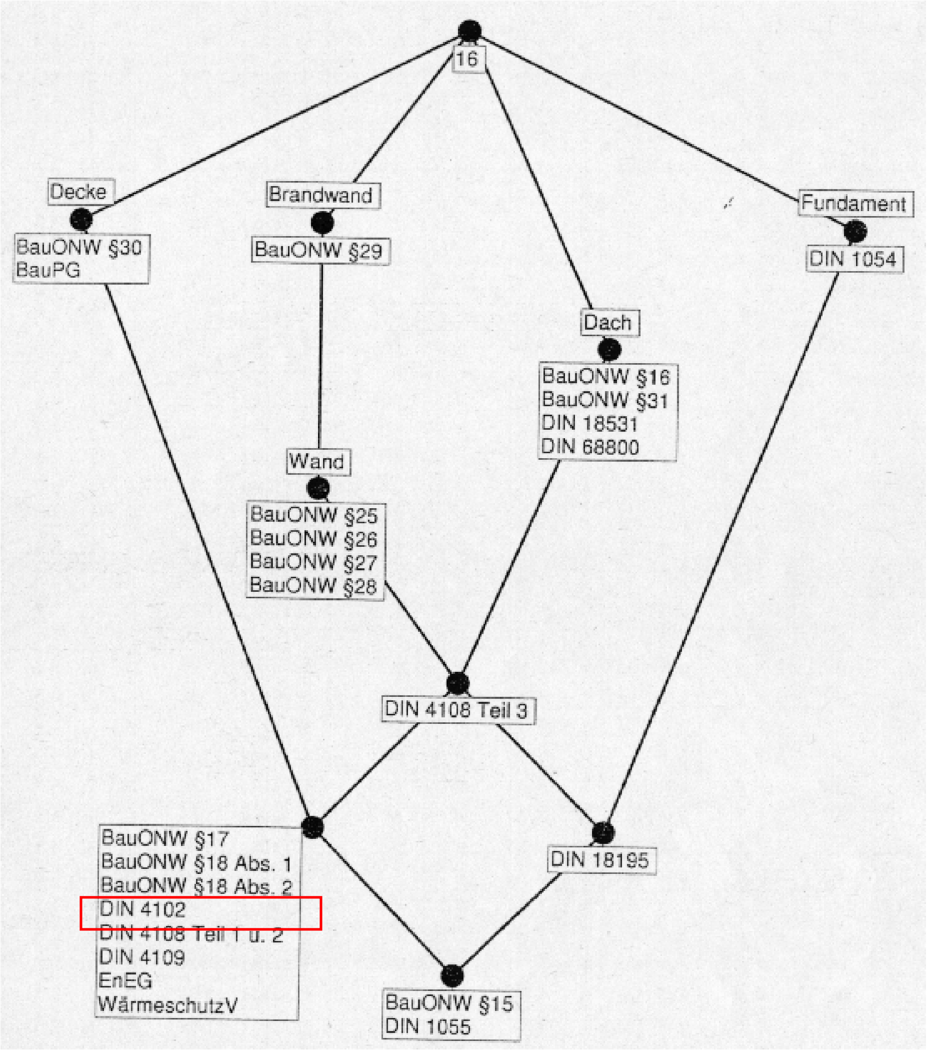
- Then the direct product of $B(K_1)$ and $B(K_2)$ is drawn. Draw a large diagram for $B(K_1)$, where the nodes are large ellipses, in which diagrams of $B(K_2)$ are drawn.

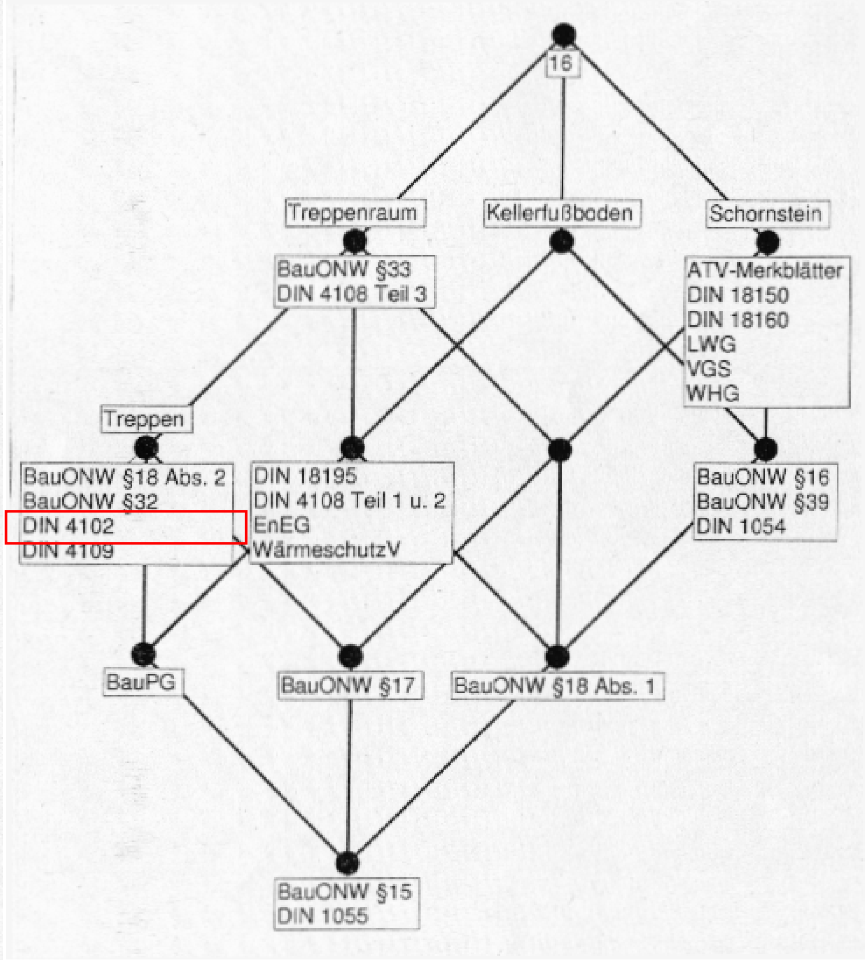
The concept lattice B(G, M, I) is embedded in this direct product as a $\bigvee$-semi-lattice (according to the previous theorem).

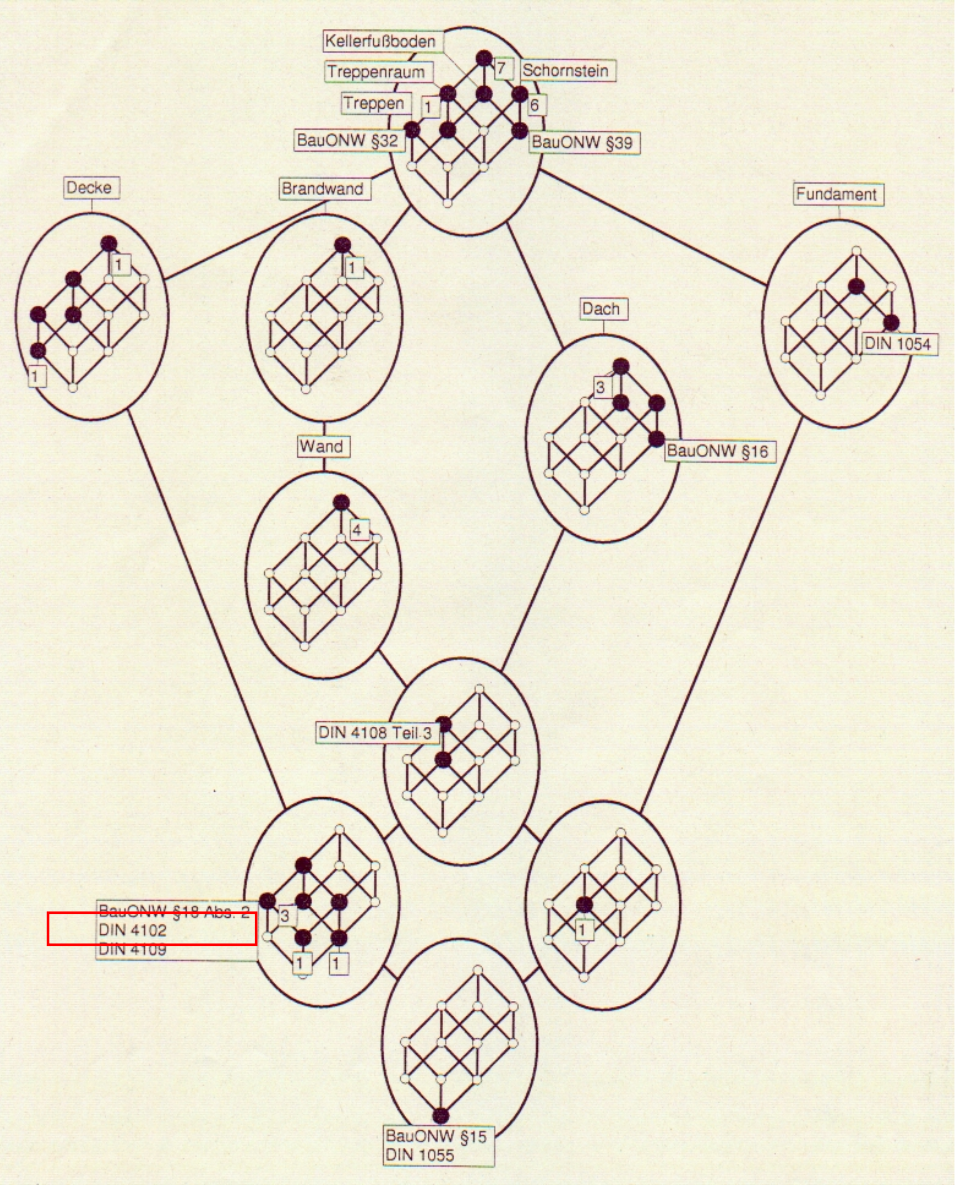
Mark it as follows:

- Put all object labels to the right positions.
- Compute all suprema, and mark them as realized concepts.

Tutorial Formal Concept Analysis

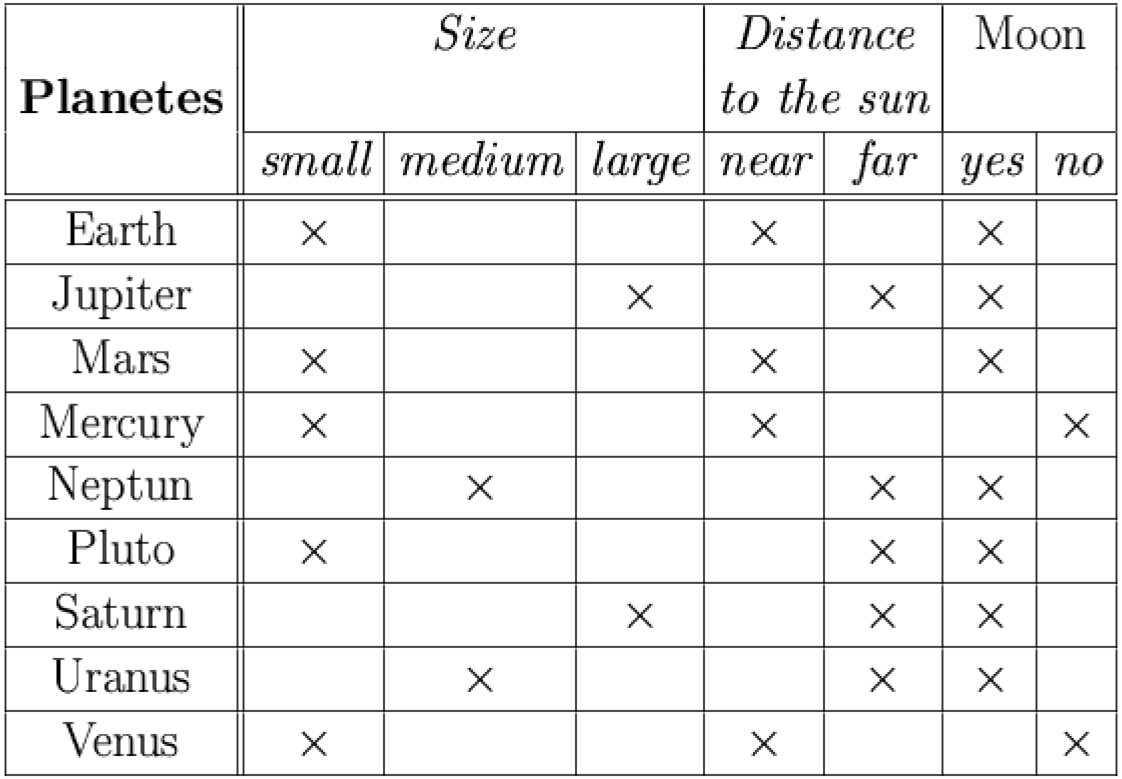





Kellerfußboden
Treppenraum
Treppen
BauONW §32
7
Schornstein
1
6
BauONW §39
Decke
1
1
Brandwand
1
Fundament
DIN 1054
Dach
3
BauONW §16
Wand
4
DIN 4108 Teil 3
BauONW §18 Abs. 2
DIN 4102
DIN 4109
3
1
1
1
BauONW §15
DIN 1055

Example on

Blackboard



Planetes	*Size*			*Distance to the sun*		Moon	
	small	*medium*	*large*	*near*	*far*	*yes*	*no*
Earth	×			×		×	
Jupiter			×		×	×	
Mars	×			×		×	
Mercury	×			×			×
Neptun		×			×	×	
Pluto	×				×	×	
Saturn			×		×	×	
Uranus		×			×	×	
Venus	×			×			×

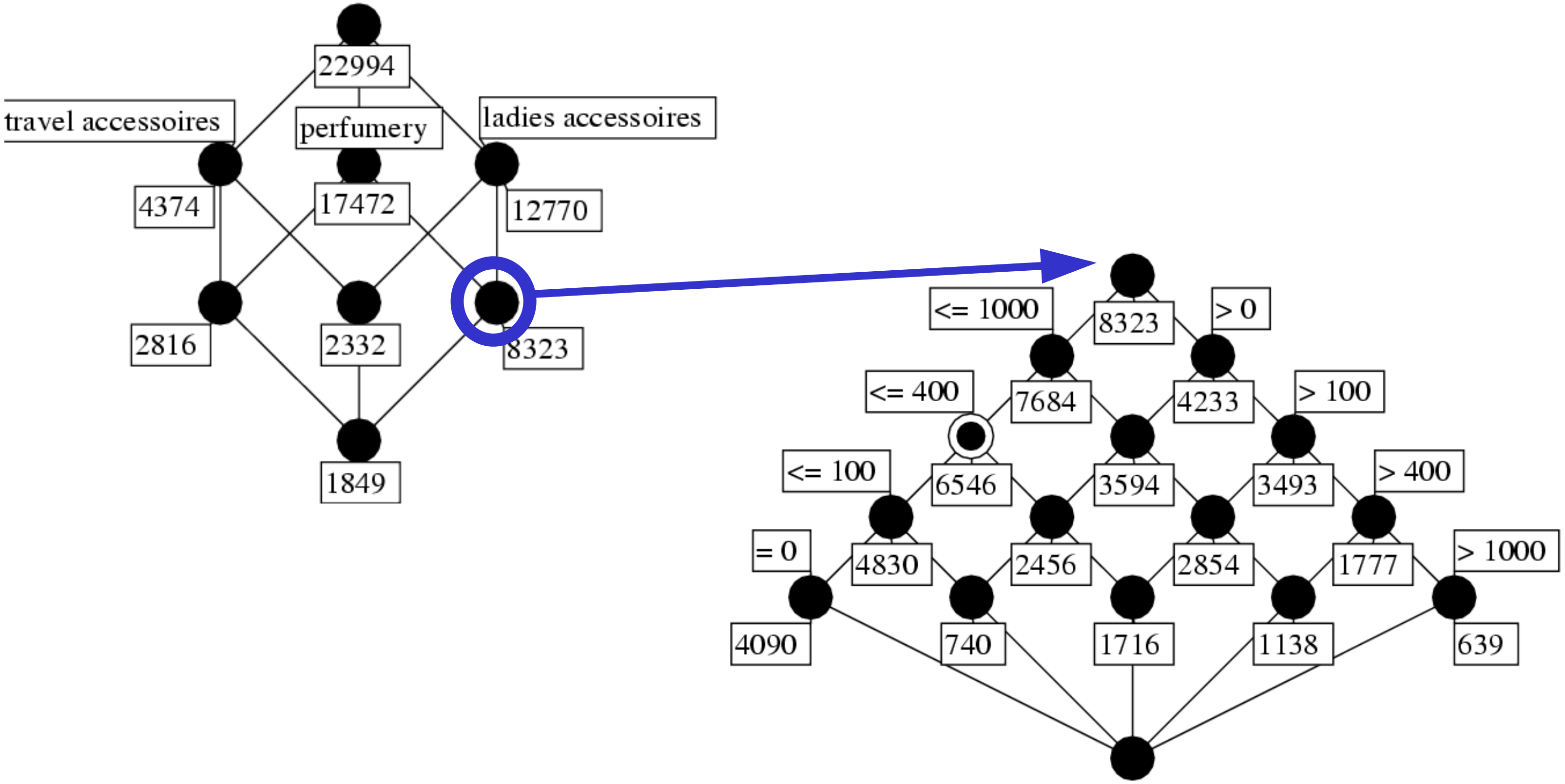
1. Introduction
2. Formal Contexts & Concept Lattices
3. Application Examples I
4. Computing Concept Lattices
5. Exercises
6. Conceptual Scaling
7. **Application Examples II**
8. Conceptual Clustering
9. FCA-Based Mining of Association Rules
10. FCA Tools
11. Exercises

Application Examples

- Database Marketing at Jelmoli AG, Zürich
- Analysis of flight movements at Frankfurt Airport
- Information Retrieval at the library of the Center for Interdisciplinary Technology Research (ZIT), TU Darmstadt
- Analysis of children suffering from diabetes, McGill Hospital Montréal
- Conceptual Email Management

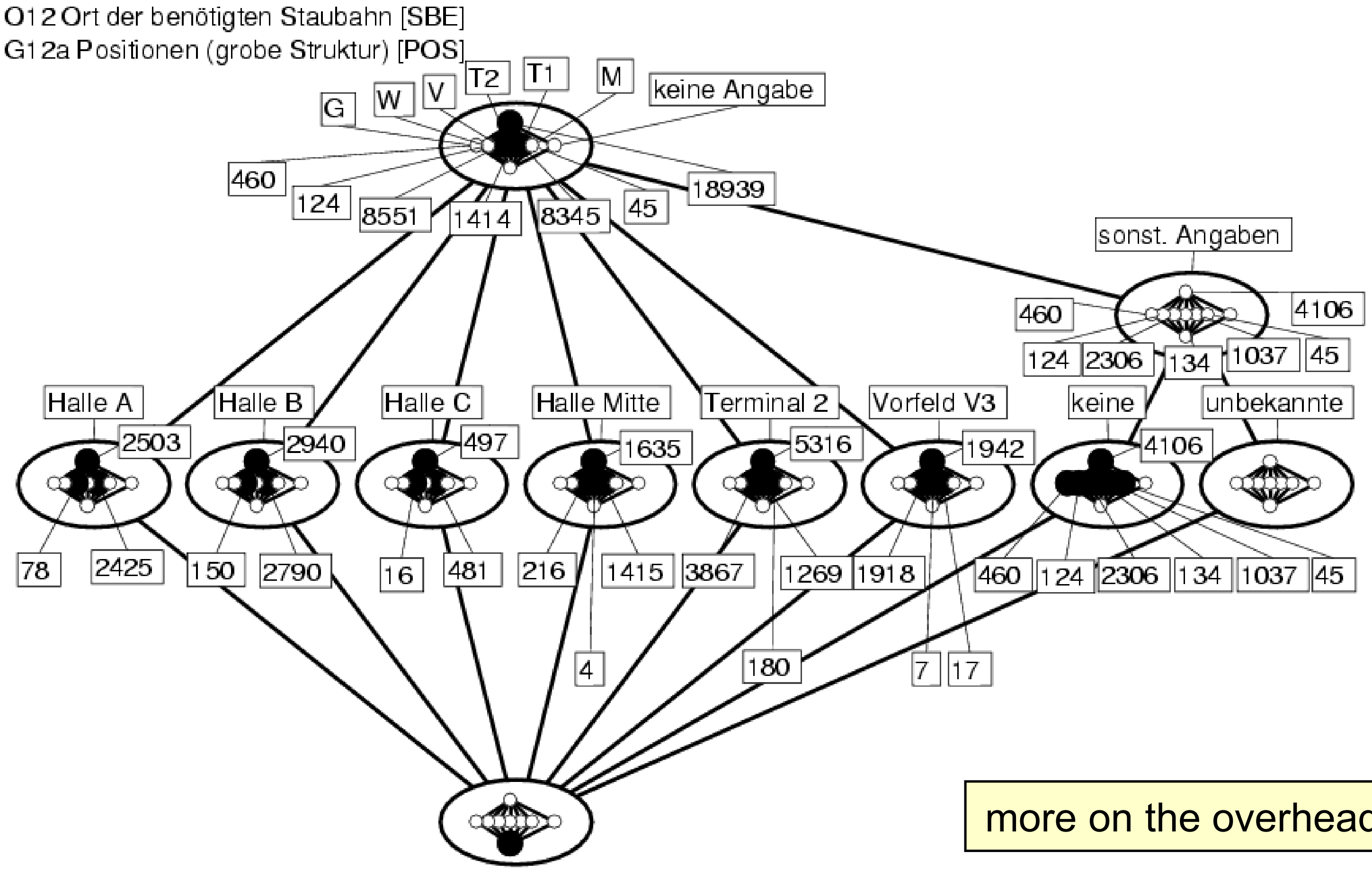
Database Marketing at Jelmoli AG, Zürich

- Analysis of the user behavior of customers using the Shopping Bonus Card
- Supporting of Cross-Selling via Direct Mailing



Analysis of flight movements at Frankfurt Airport

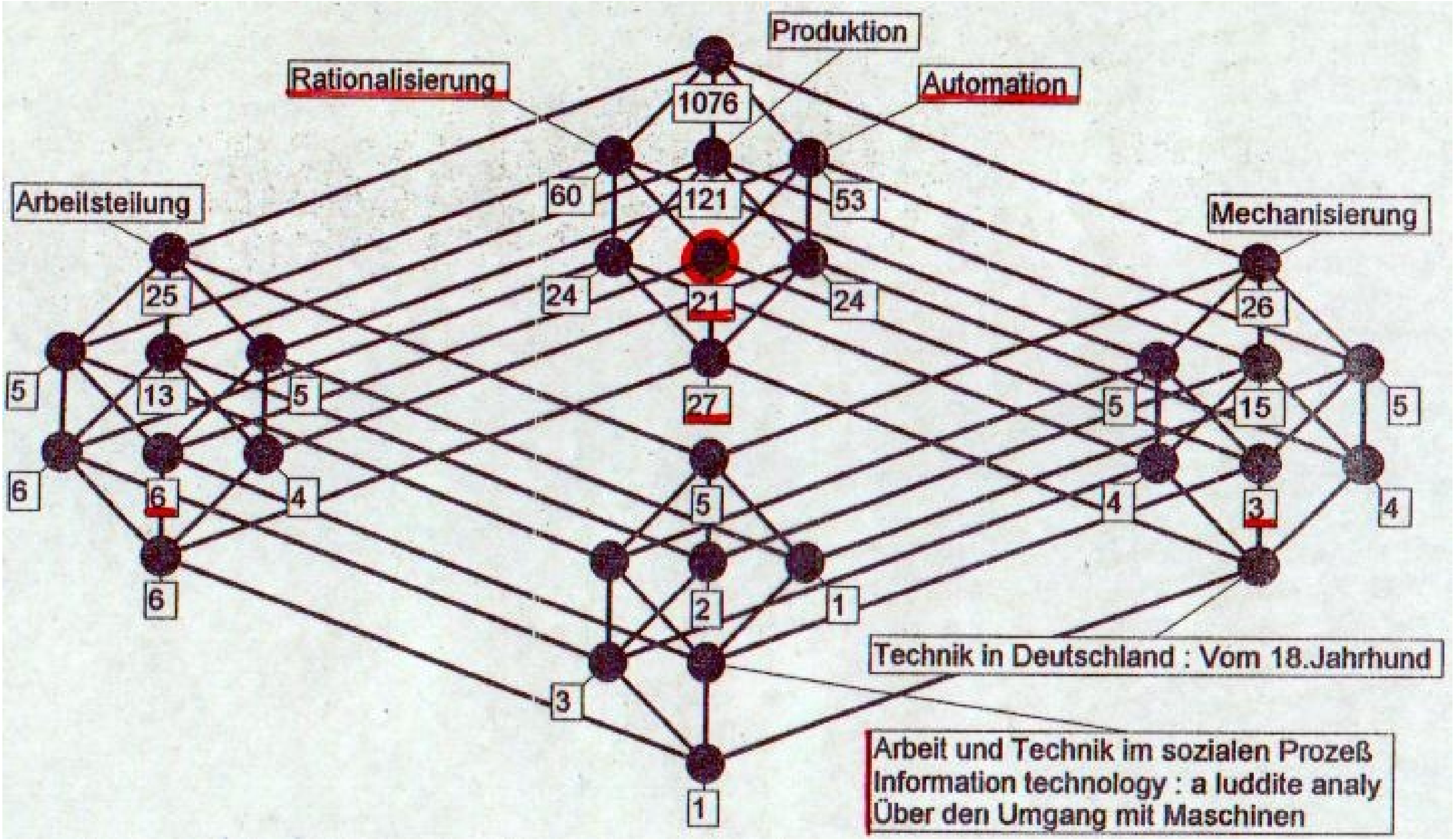
- Ermöglichen von Ad-hoc-Anfragen an die Datenbank
- Visualisierung von Zusammenhängen



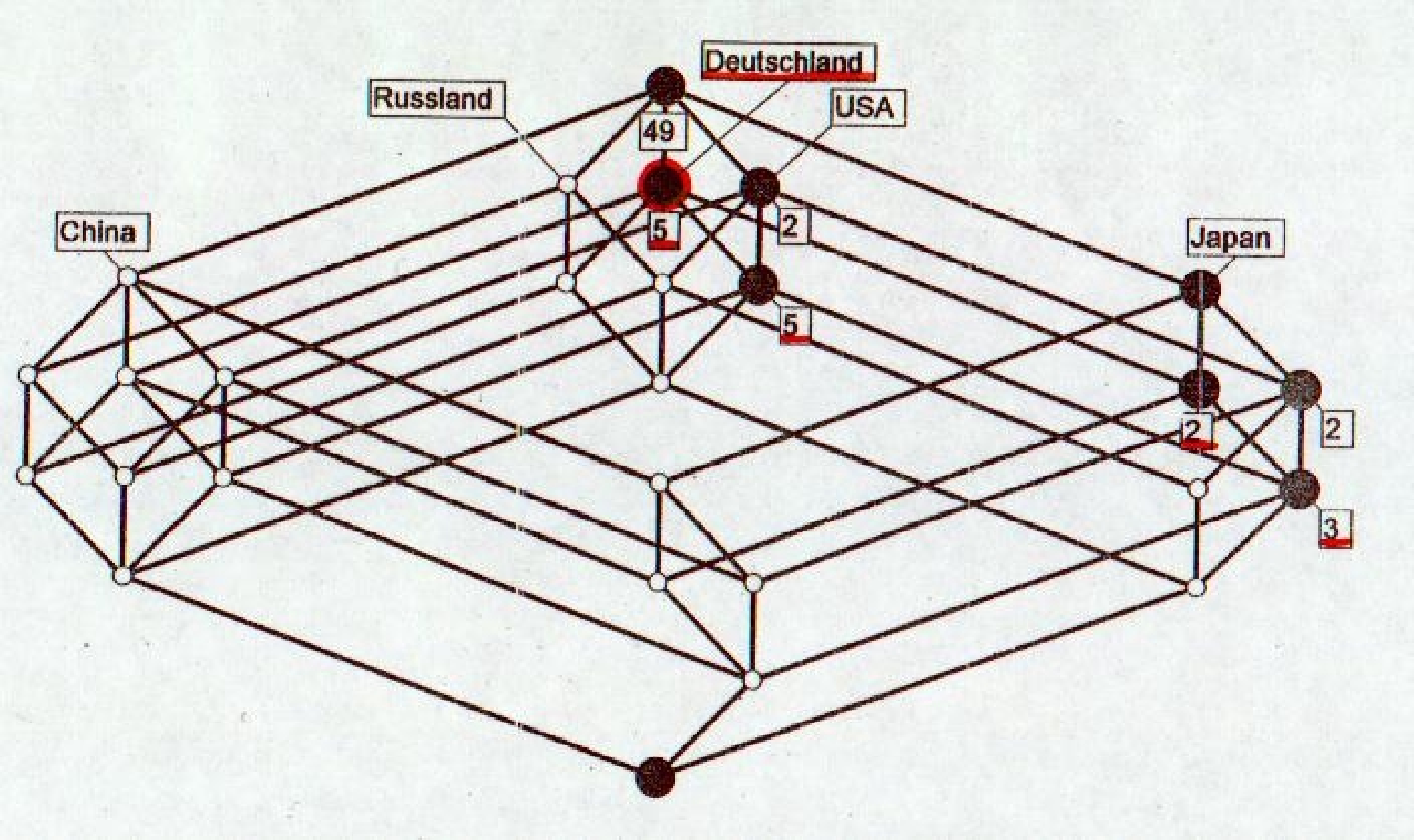
more on the overhead slides

Information Retrieval at the ZIT library, TU Darmstadt

Example: Search for older literature about automation in the most important industrial countries

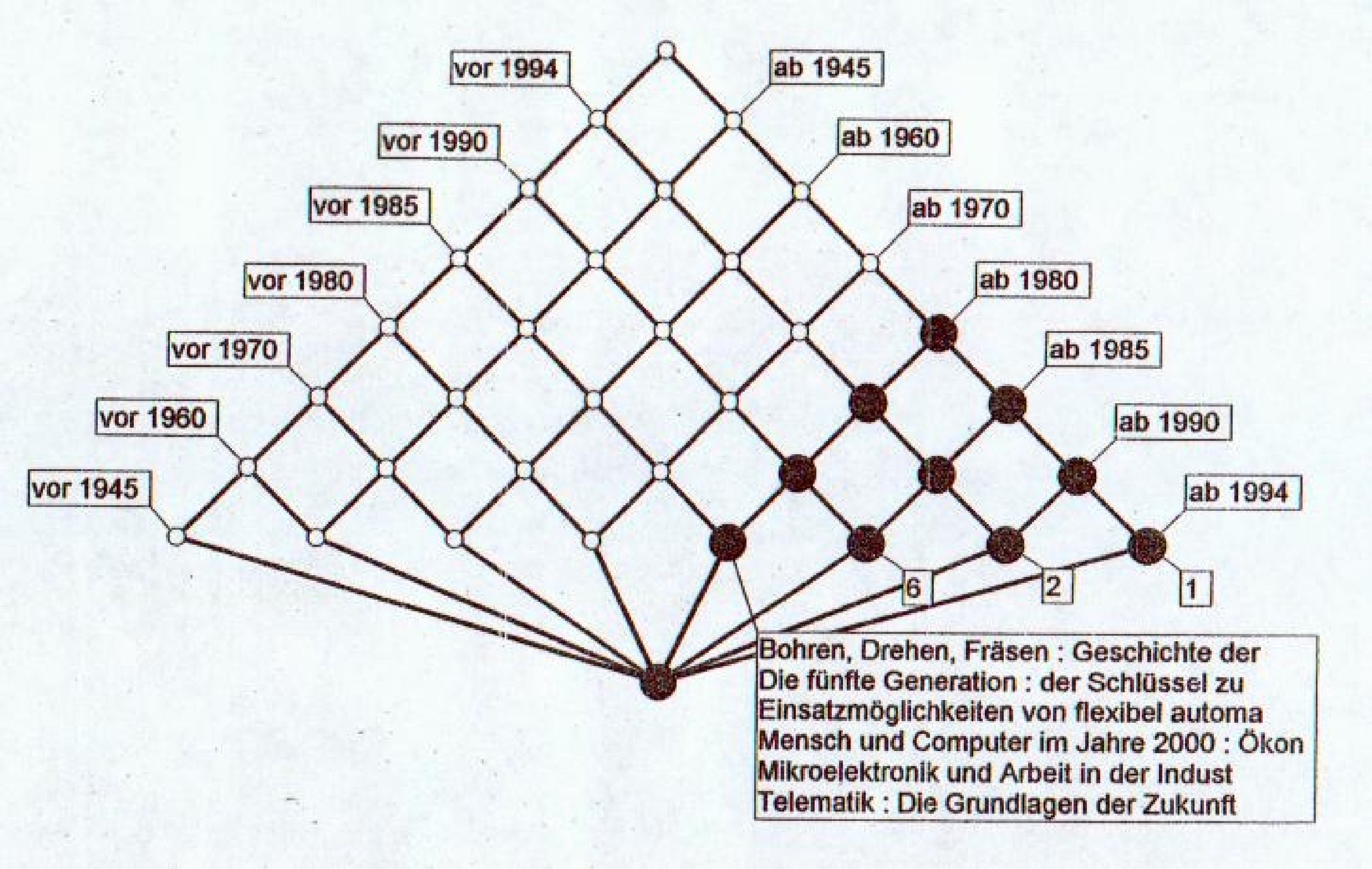

Produktion
Rationalisierung
Automation
1076
Arbeitsteilung
60
121
53
Mechanisierung
25
24
21
24
26
5
13
5
27
5
15
5
6
6
4
5
4
3
4
6
2
1
Technik in Deutschland : Vom 18.Jahrhund
3
Arbeit und Technik im sozialen Prozeß
Information technology : a luddite analy
Über den Umgang mit Maschinen
1

Scale Change of Production



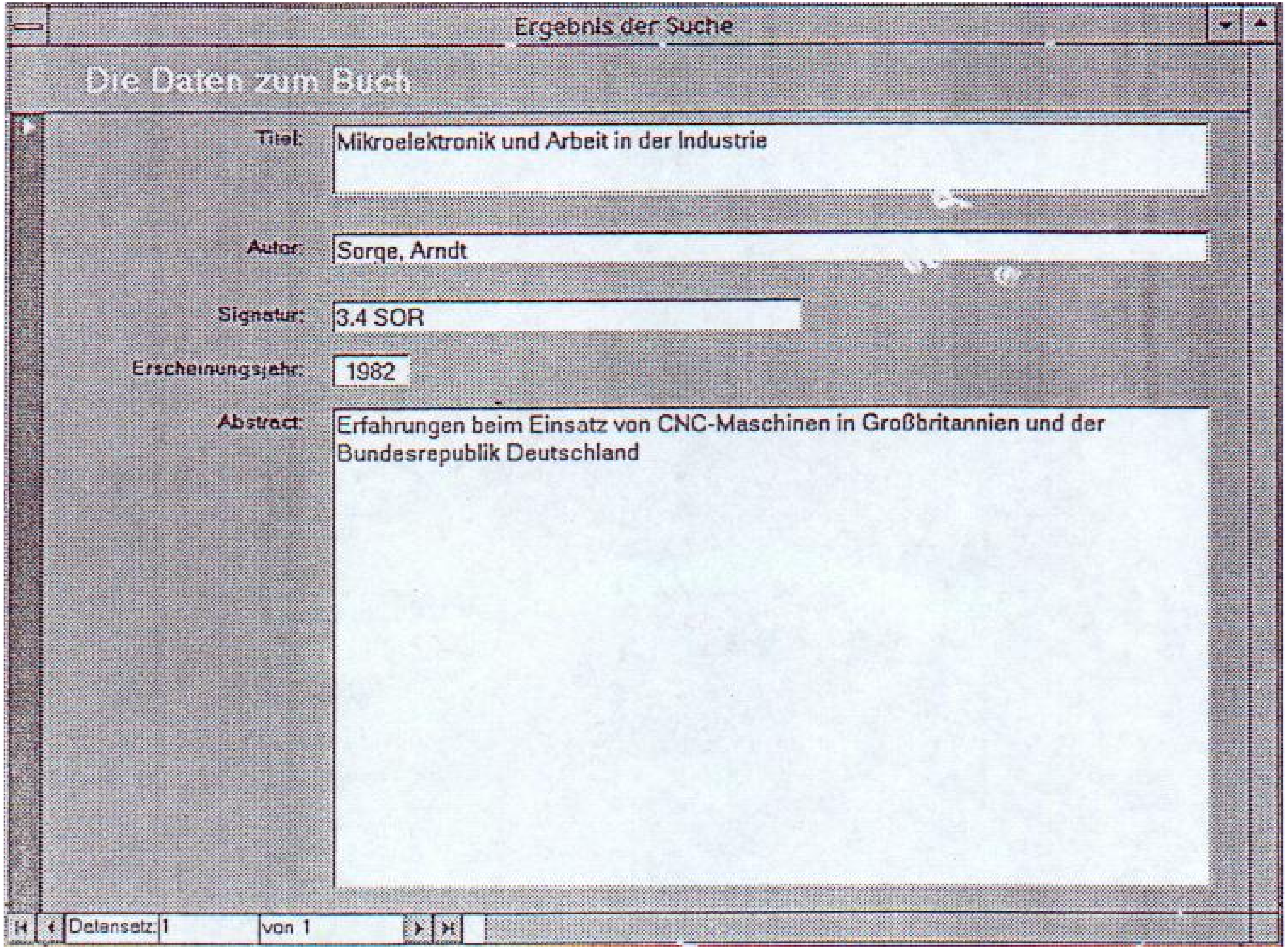
Scale Important Industrial Countries

(restricted to books with the catchwords Automation and Rationalisierung


vor 1994
ab 1945
vor 1990
ab 1960
vor 1985
ab 1970
vor 1980
ab 1980
vor 1970
ab 1985
vor 1960
ab 1990
vor 1945
ab 1994
6
2
1
Bohren, Drehen, Fräsen : Geschichte der
Die fünfte Generation : der Schlüssel zu
Einsatzmöglichkeiten von flexibel automa
Mensch und Computer im Jahre 2000 : Ökon
Mikroelektronik und Arbeit in der Indust
Telematik : Die Grundlagen der Zukunft

Scale Publishing Year

restricted to books with the catchwords Deutschland, Automation and Rationalisierung)



Titel: Mikroelektronik und Arbeit in der Industrie

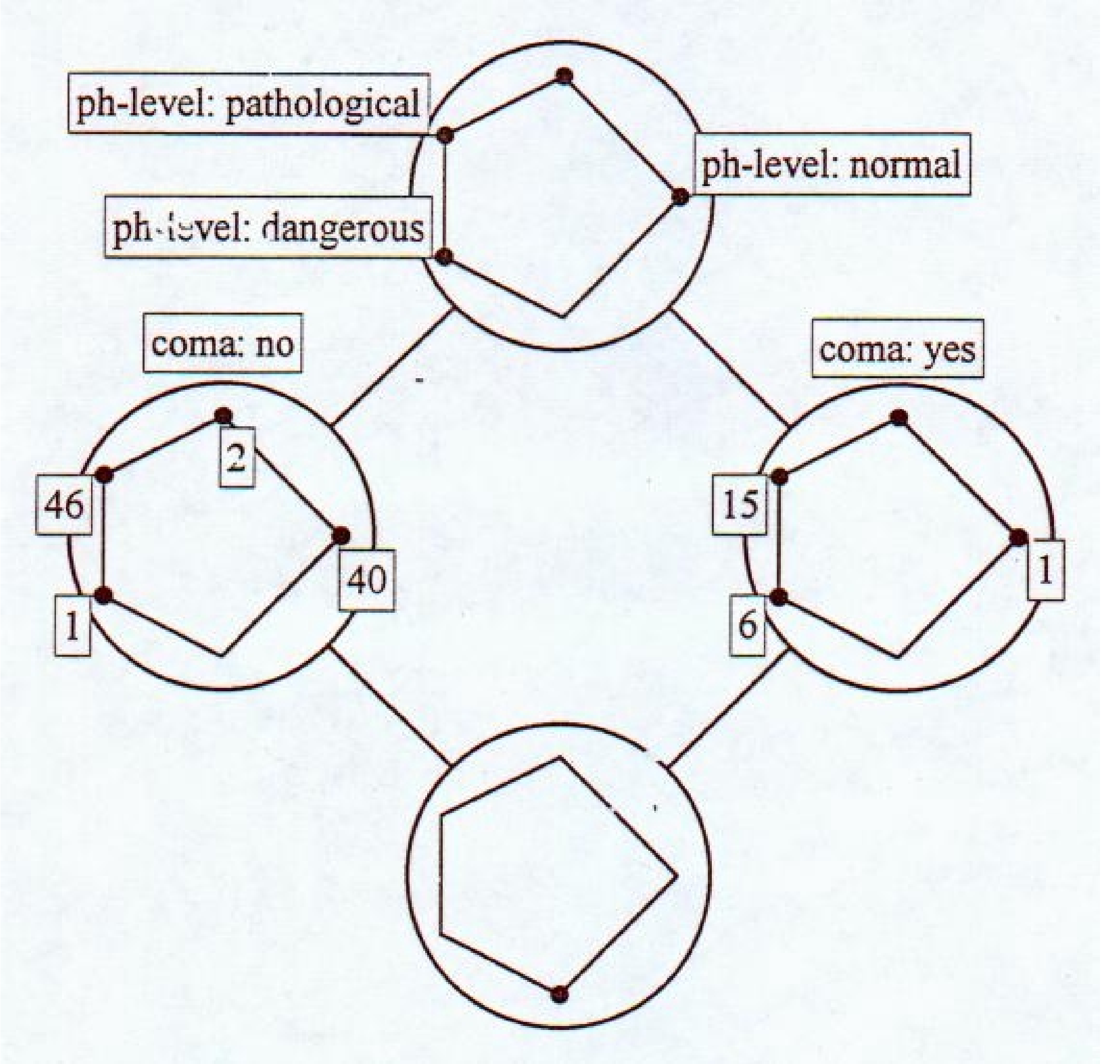
Autor: Sorge, Arndt

Signatur: 3.4 SOR

Erscheinungsjahr: 1982

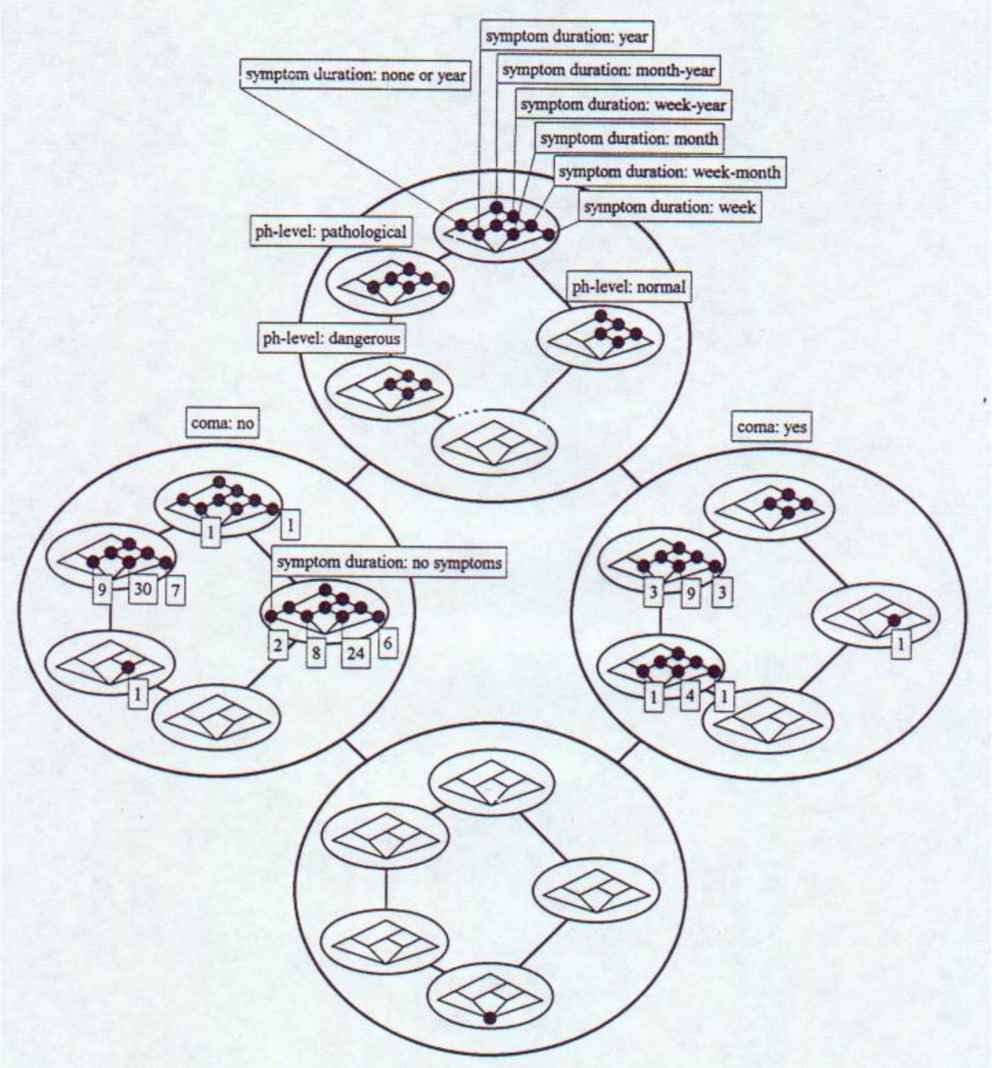
Abstract: Erfahrungen beim Einsatz von CNC-Maschinen in Großbritannien und der Bundesrepublik Deutschland

Analysis of children suffering from diabetes, McGill Hospital Montréal



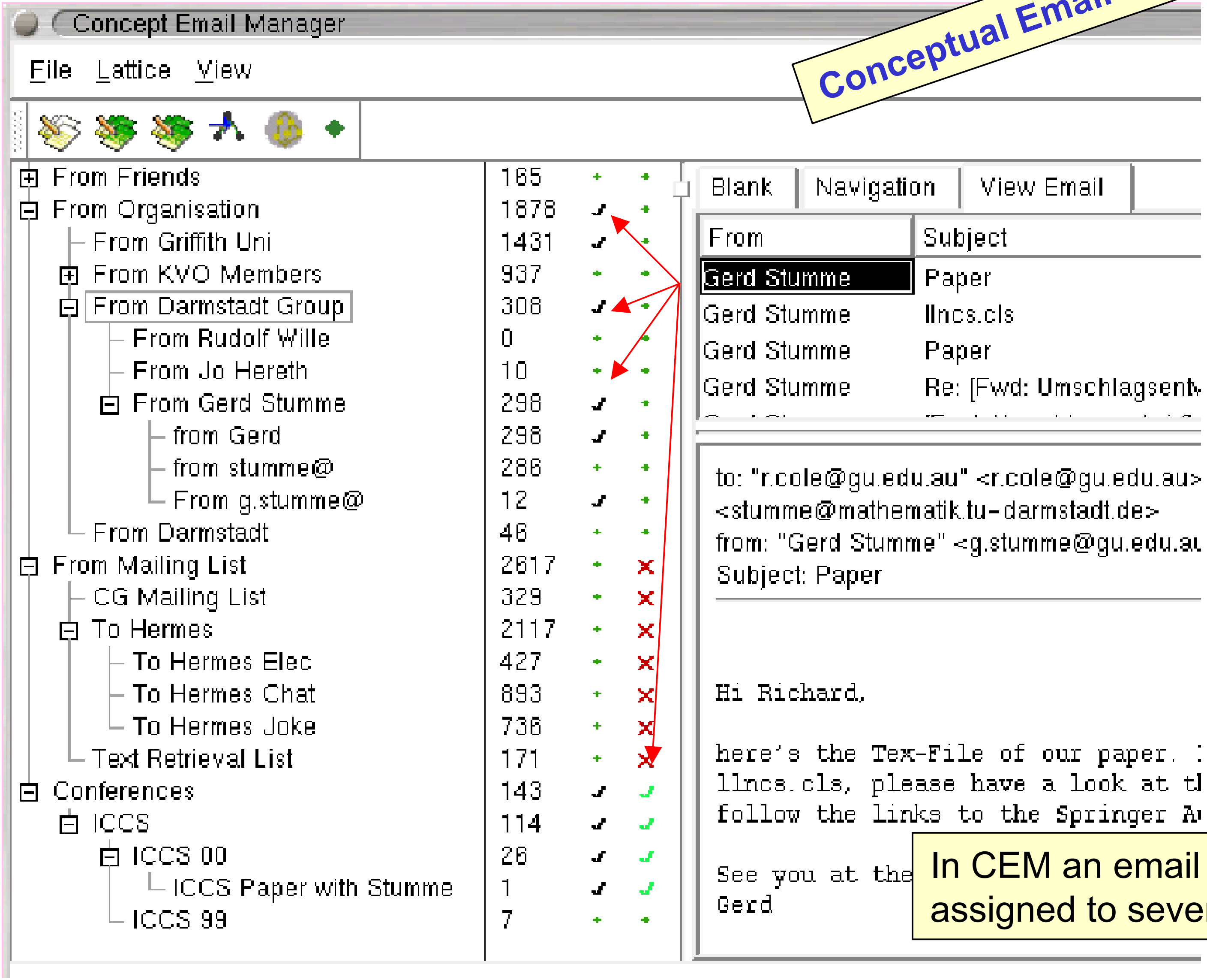
Scales Coma and pH-level of the blood

Analysis of children suffering from diabetes, McGill Hospital Montréal



Scales Coma, pH-Wert of the blood and Symptom Duration

Conceptual Email Manager



⊞ From Friends	165	·	·
⊟ From Organisation	1878	✓	·
From Griffith Uni	1431	✓	·
⊞ From KVO Members	937	·	·
⊟ From Darmstadt Group	308	✓	·
From Rudolf Wille	0	·	·
From Jo Hereth	10	·	·
⊟ From Gerd Stumme	298	✓	·
from Gerd	298	✓	·
from stumme@	286	·	·
From g.stumme@	12	✓	·
From Darmstadt	46	·	·
⊟ From Mailing List	2617	·	✗
CG Mailing List	329	·	✗
⊟ To Hermes	2117	·	✗
To Hermes Elec	427	·	✗
To Hermes Chat	893	·	✗
To Hermes Joke	736	·	✗
Text Retrieval List	171	·	✗
⊟ Conferences	143	✓	✓
⊟ ICCS	114	✓	✓
⊟ ICCS 00	26	✓	✓
ICCS Paper with Stumme	1	✓	✓
ICCS 99	7	·	·

From	Subject
Gerd Stumme	Paper
Gerd Stumme	llncs.cls
Gerd Stumme	Paper
Gerd Stumme	Re: [Fwd: Umschlagsenl

to: "r.cole@gu.edu.au" <r.cole@gu.edu.au>
<stumme@mathematik.tu-darmstadt.de>
from: "Gerd Stumme" <g.stumme@gu.edu.au
Subject: Paper

Hi Richard,

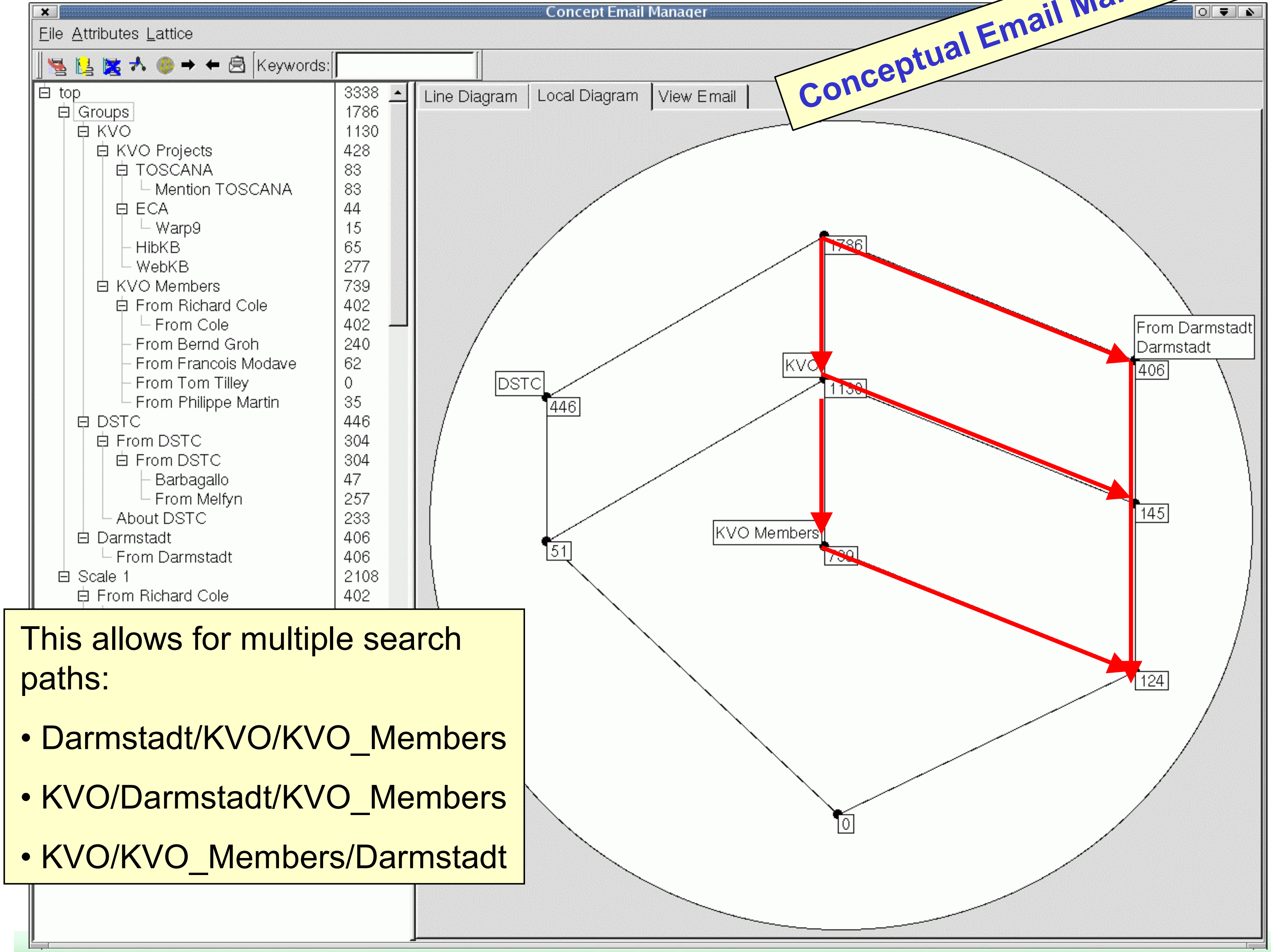
here's the Tex-File of our paper.
llncs.cls, please have a look at tl
follow the links to the Springer A

See you at the
Gerd

In CEM an email can be assigned to several „folders".

Conceptual Email Manager

Conceptual Email Manager

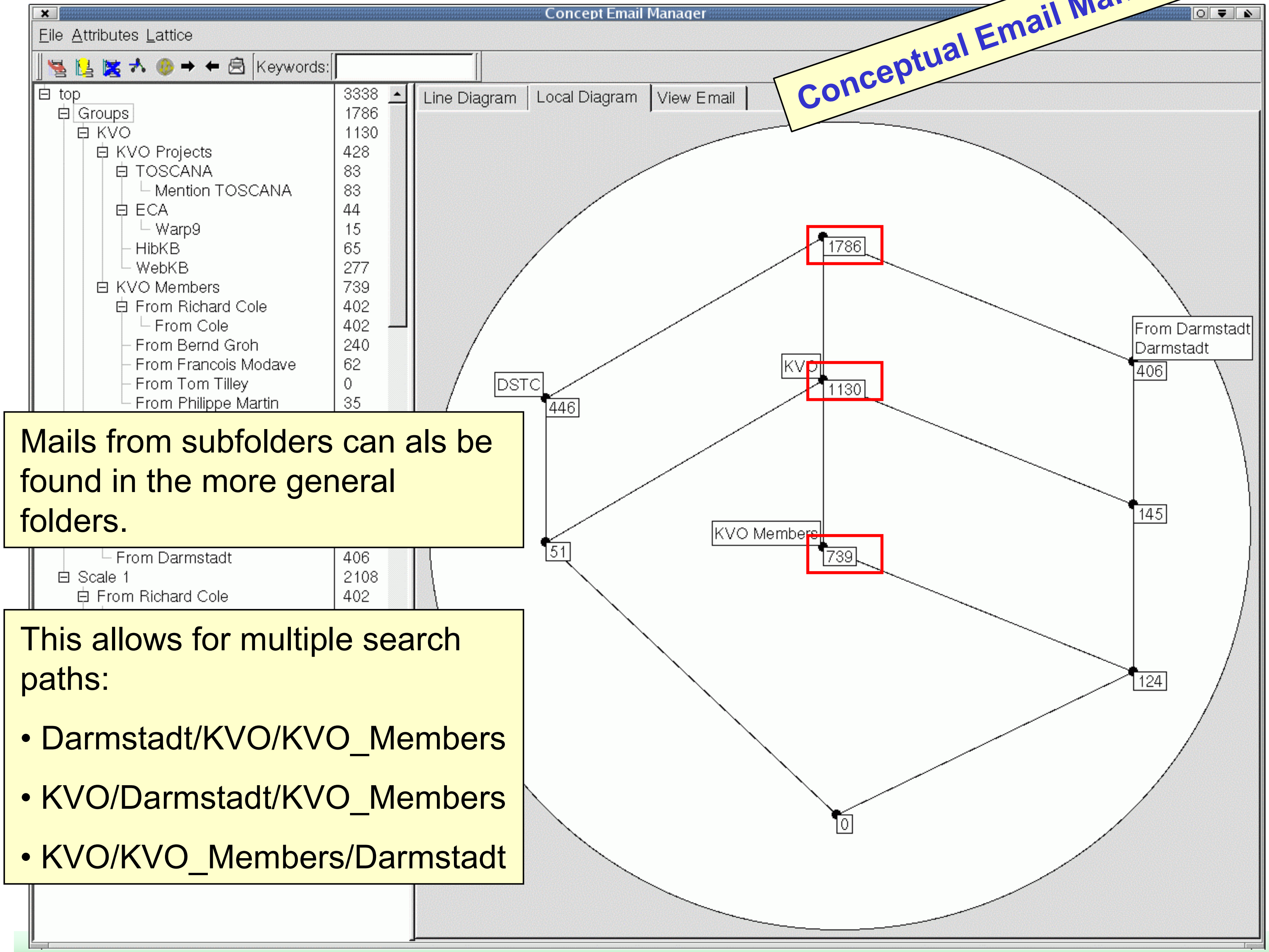


top	3338
Groups	1786
KVO	1130
KVO Projects	428
TOSCANA	83
Mention TOSCANA	83
ECA	44
Warp9	15
HibKB	65
WebKB	277
KVO Members	739
From Richard Cole	402
From Cole	402
From Bernd Groh	240
From Francois Modave	62
From Tom Tilley	0
From Philippe Martin	35
DSTC	446
From DSTC	304
From DSTC	304
Barbagallo	47
From Melfyn	257
About DSTC	233
Darmstadt	406
From Darmstadt	406
Scale 1	2108
From Richard Cole	402

This allows for multiple search paths:

- Darmstadt/KVO/KVO_Members
- KVO/Darmstadt/KVO_Members
- KVO/KVO_Members/Darmstadt

Conceptual Email Manager



top	3338
Groups	1786
KVO	1130
KVO Projects	428
TOSCANA	83
Mention TOSCANA	83
ECA	44
Warp9	15
HibKB	65
WebKB	277
KVO Members	739
From Richard Cole	402
From Cole	402
From Bernd Groh	240
From Francois Modave	62
From Tom Tilley	0
From Philippe Martin	35
From Darmstadt	406
Scale 1	2108
From Richard Cole	402

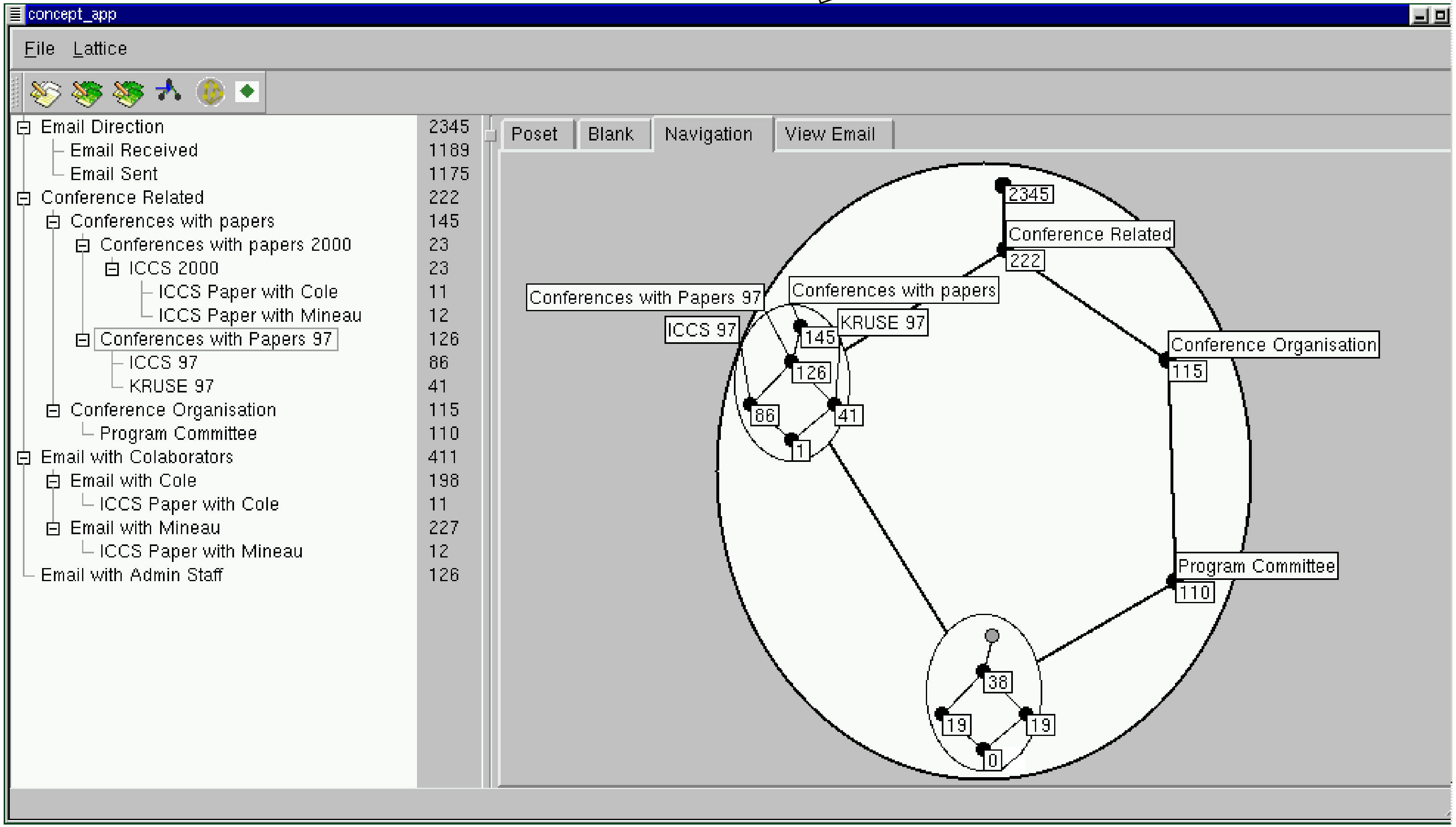
Mails from subfolders can als be found in the more general folders.

This allows for multiple search paths:

- Darmstadt/KVO/KVO_Members
- KVO/Darmstadt/KVO_Members
- KVO/KVO_Members/Darmstadt

Nested line diagrams allow the combination of views.

Conceptual Email Manager



⊟ Email Direction	2345
Email Received	1189
Email Sent	1175
⊟ Conference Related	222
⊟ Conferences with papers	145
⊟ Conferences with papers 2000	23
⊟ ICCS 2000	23
ICCS Paper with Cole	11
ICCS Paper with Mineau	12
⊟ Conferences with Papers 97	126
ICCS 97	86
KRUSE 97	41
⊟ Conference Organisation	115
Program Committee	110
⊟ Email with Colaborators	411
⊟ Email with Cole	198
ICCS Paper with Cole	11
⊟ Email with Mineau	227
ICCS Paper with Mineau	12
Email with Admin Staff	126

1. Introduction
2. Formal Contexts & Concept Lattices
3. Application Examples I
4. Computing Concept Lattices
5. Exercises
6. Conceptual Scaling
7. Application Examples II
8. **Conceptual Clustering**
9. FCA-Based Mining of Association Rules
10. FCA Tools
11. Exercises

Conceptual Clustering

Conceptual Clustering methods are clustering methods which generate simultaneously descriptions of the clusters.

- Examples: Michalski & Stepp 1983; Lebowitz 1987; Fisher 1987; Gennari et al 1989
- **Advantages** of conceptual clustering against non-conceptual clustering:
  - A cluster is not only a set of objects, but there also exists an intensional description.
- **Disadvantages**:
  - The language used to describe the clusters restricts the type of clusters which can be built.
  - The computation has usually higher complexity.

Iceberg concept lattices only allow conjunctions of attributes as descriptions.

- **Recall:** the support of an itemset $X \subseteq M$ is given by

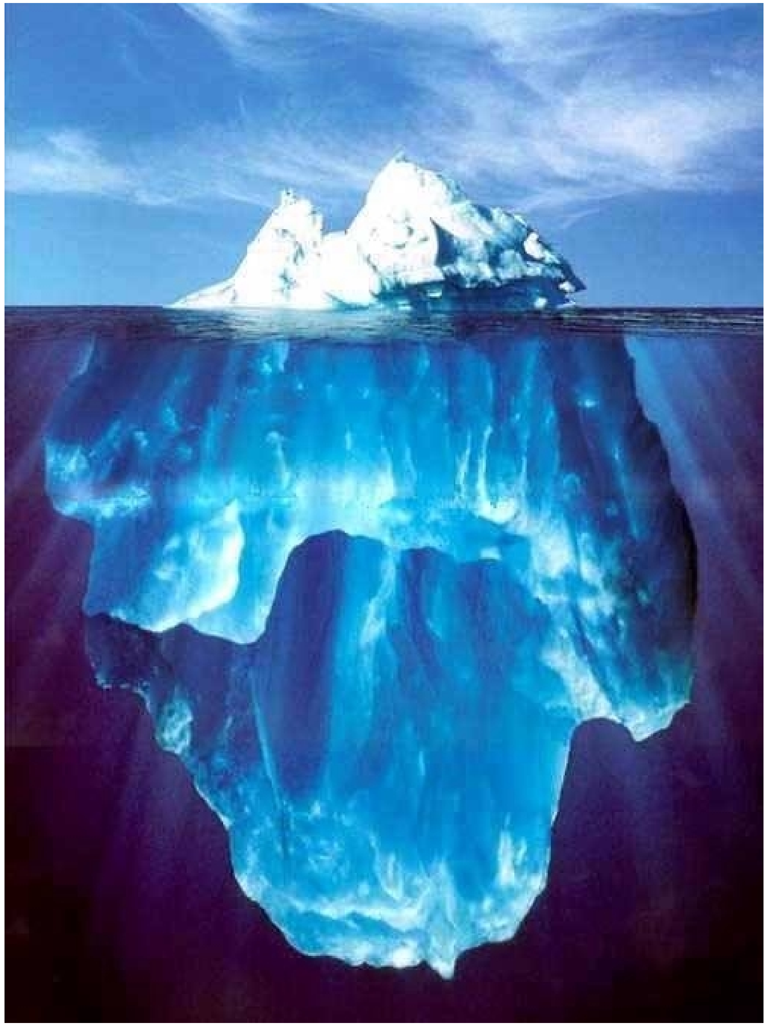
$$\mathrm{supp}(X) = \frac{|X'|}{|G|}$$

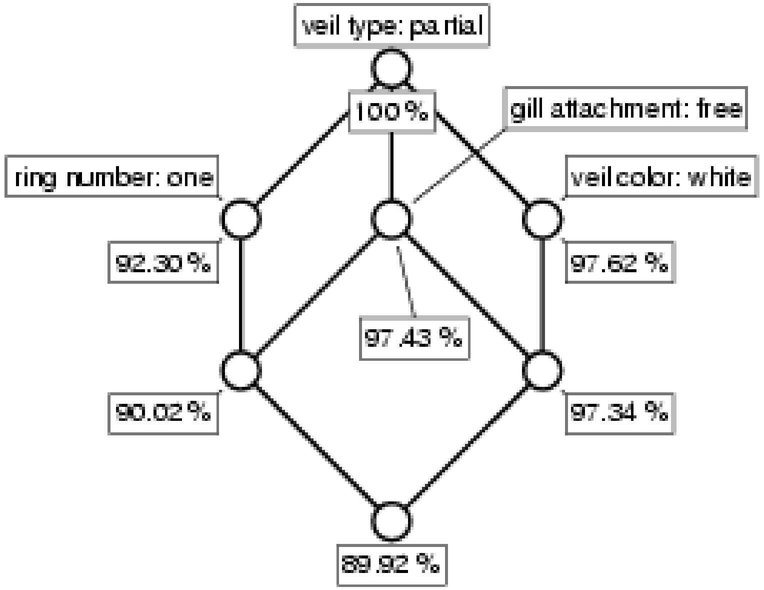
- Def.: The **iceberg concept lattice** of a formal context (G,M,I) for a given minimal support minsupp is the set

$$\{ (A,B) \in \underline{\mathbf{B}}(G,M,I) \mid \mathrm{supp}(B) \geq \mathrm{minsupp} \}$$

- It can be computed with **TITANIC.** [Stumme et al 2001]

Iceberg Concept Lattices

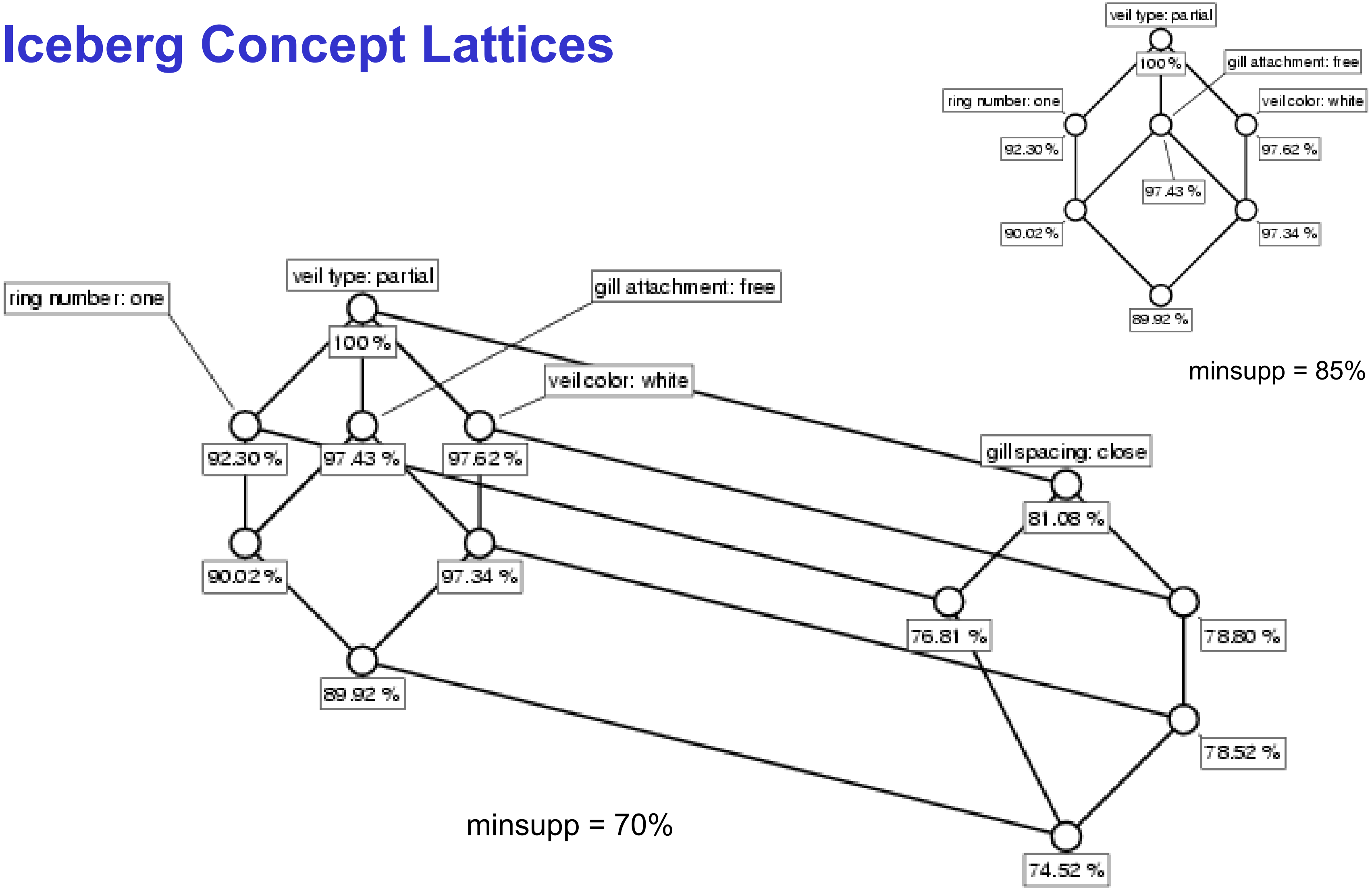




minsupp = 85%

For minsupp = 85% the seven most general of the 32.086 concepts of the Mushrooms database http:\\kdd.ics.uci.edu are shown.

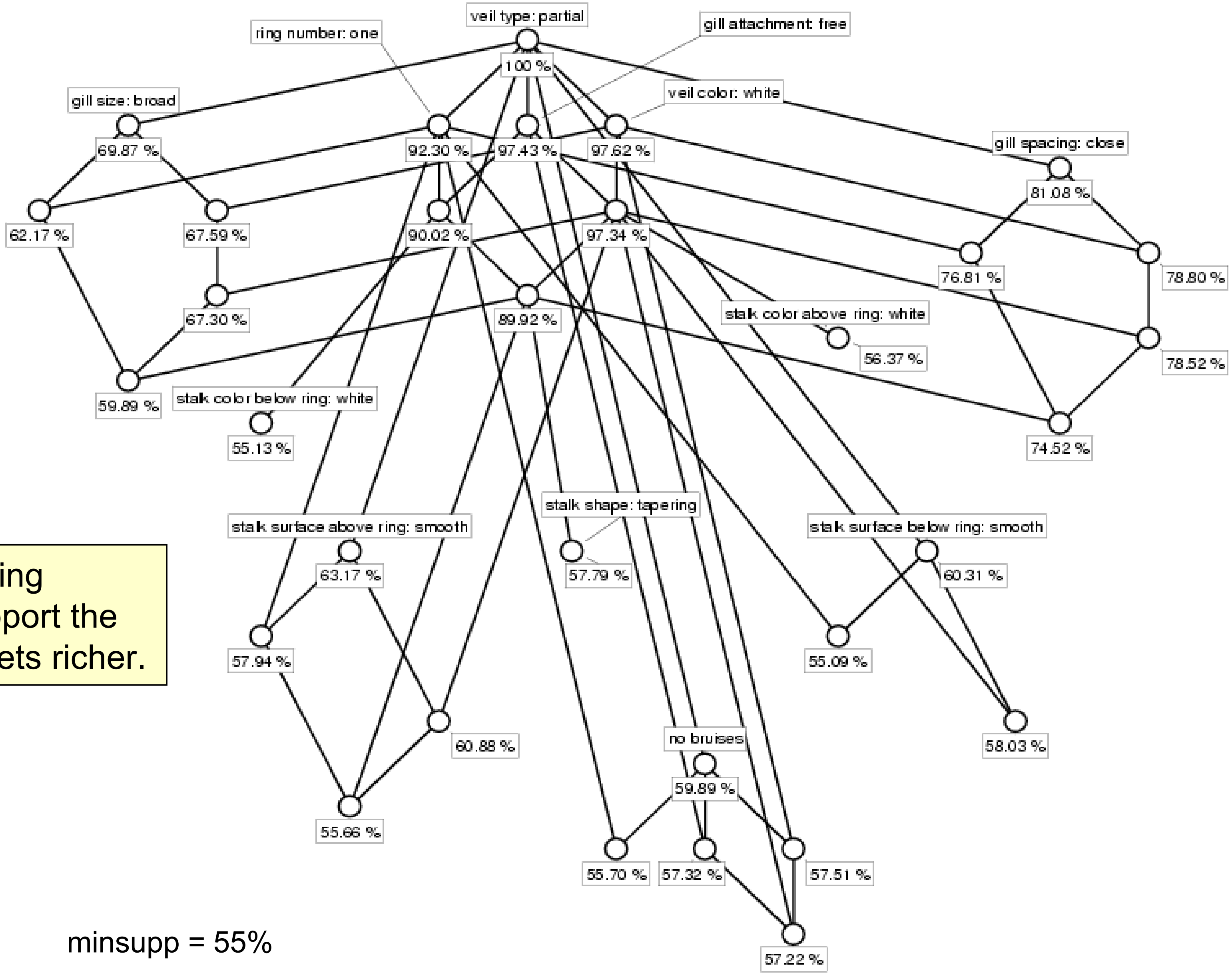
Iceberg Concept Lattices



minsupp = 85%

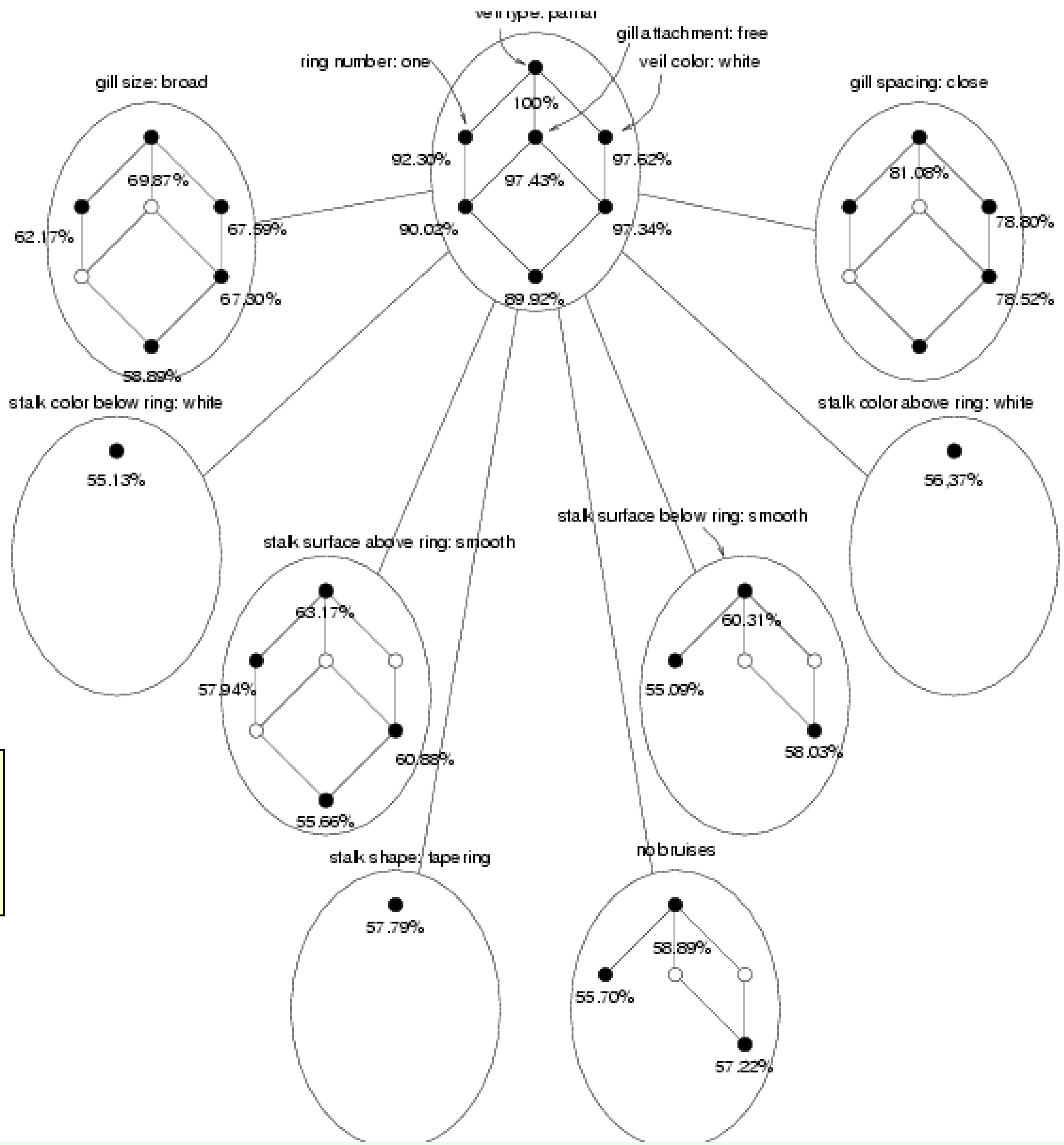
minsupp = 70%

Tutorial Formal Concept Analysis



With decreasing minimum support the information gets richer.

minsupp = 55%



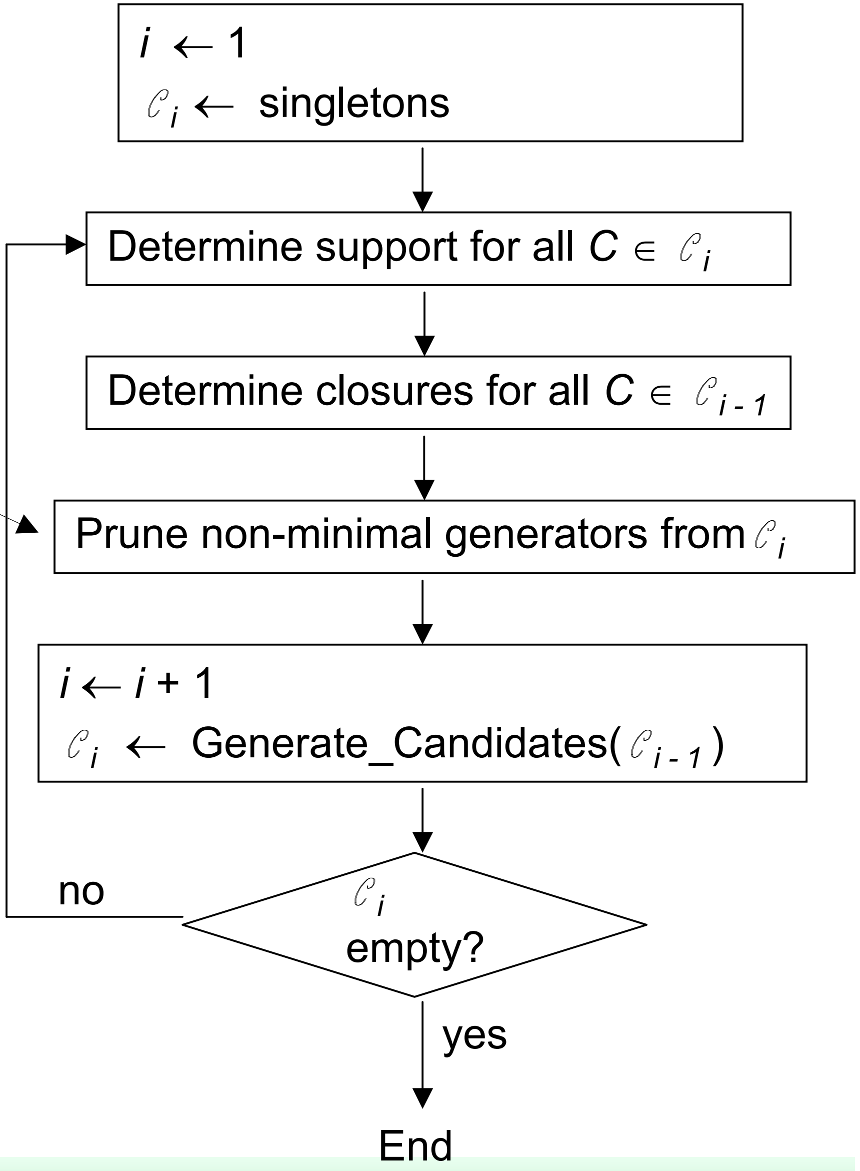
The visualization as a nested line diagram indicates implications.

TITANIC

modified for computing iceberg concept lattices

The only modification is the additional condition that

a candidate is also removed if its support is too low.



$i \leftarrow 1$

$\mathcal{C}_i \leftarrow$ singletons

Determine support for all $C \in \mathcal{C}_i$

Determine closures for all $C \in \mathcal{C}_{i-1}$

Prune non-minimal generators from $\mathcal{C}_i$

$i \leftarrow i + 1$

$\mathcal{C}_i \leftarrow$ Generate_Candidates($\mathcal{C}_{i-1}$)

$\mathcal{C}_i$ empty?

no

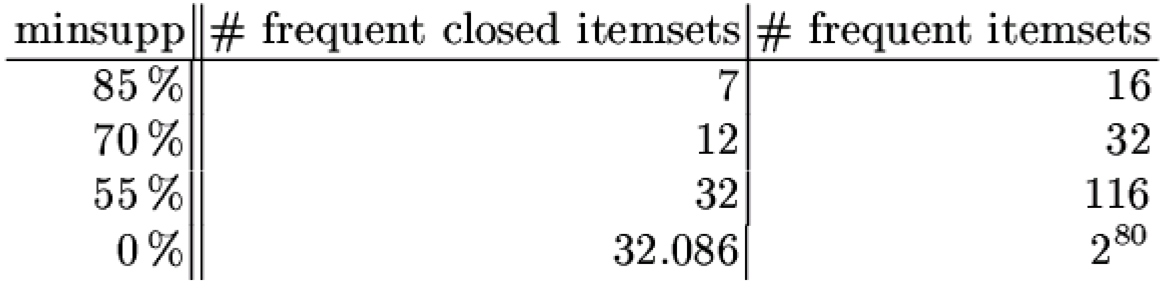
yes

End

Iceberg Concept Lattices and Frequent Itemsets

Iceberg concept lattices are a condensed representation of frequent itemsets:

supp(X) = supp(X‘‘)



minsupp	# frequent closed itemsets	# frequent itemsets
85 %	7	16
70 %	12	32
55 %	32	116
0 %	32.086	2^{80}

Differences between frequent concepts and frequent itemsets in the mushrooms database.

1. Introduction
2. Formal Contexts & Concept Lattices
3. Application Examples I
4. Computing Concept Lattices
5. Exercises
6. Conceptual Scaling
7. Application Examples II
8. Conceptual Clustering
9. **FCA-Based Mining of Association Rules**
10. FCA Tools
11. Exercises

Association Rules

{ veil color: white, gill spacing: close } → { gill attachment: free }

Support: 78,52 % Confidence: 99,6 %

The input data of association rules algorithms can be written as a formal context (*G,M,I*):

- *M* is a set of items*,*
- *G* consists of the transaction IDs,
- and the relation *I* is the list of transactions*.*

Association Rules

{ veil color: white, gill spacing: close } → { gill attachment: free }

Support: 78,52 % Confidence: 99,6 %

The **support** is the percentage of all objects having all attributes in premise and conclusion:

Def.: The support of an attribute set $X \subseteq M$ is given by

$$\text{supp}(X) = \frac{|X'|}{|G|}$$

The support of an association rule $X \rightarrow Y$ is given by supp ($X \rightarrow Y$) := supp ($X \cup Y$).

The **confidence** is the percentage of all objects fulfilling the premise among all objects fulfilling both premise and conclusion.

Def.: The confidence of a rule $X \rightarrow Y$ is given by

$$\text{conf}(X \rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}$$

Bases of Association Rules

{ veil color: white, gill spacing: close } → { gill attachment: free }

Support: 78,52 % Confidence: 99,6 %

Classical Data Mining Task: Find, for given minsupp, minconf ∈ [0,1], all rules with support and confidence above these thresholds

Our task: Find a **basis** of rules, i.e., a minimal set of rules out of which all other rules can be derived.

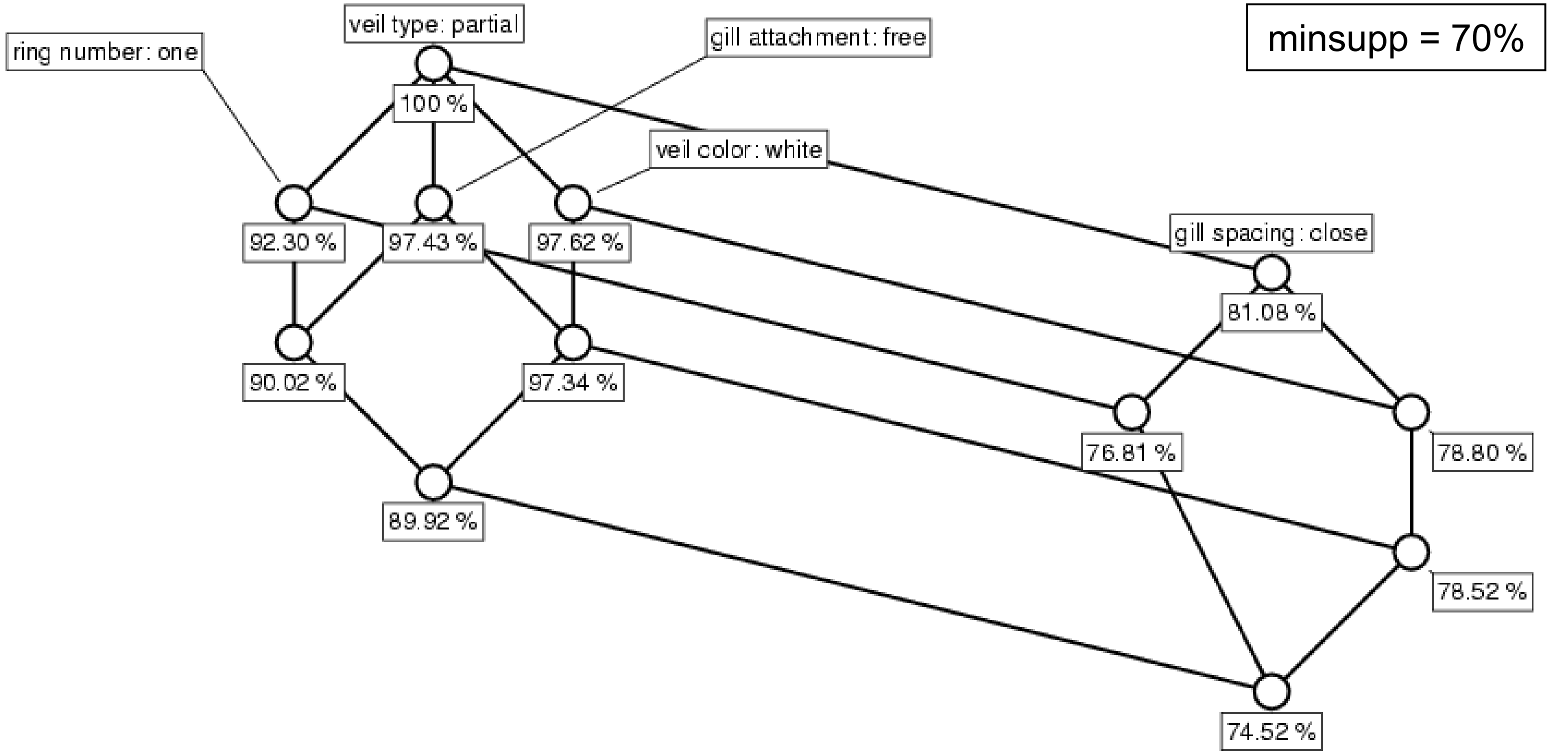
- From *B*‘ = *B*‘‘‘ follows

$$\operatorname{supp}(B) = \frac{|B'|}{|G|} = \frac{|B'''|}{|G|} = \operatorname{supp}(B'')$$

Theorem: $X \rightarrow Y$ and $X'' \rightarrow Y''$ have the same support and the same confidence.

Hence for computing association rules, it is sufficient to compute the supports of all frequent sets with *B* = *B*´´ (i.e., the intents of the iceberg concept lattice).

Advantage of the use of iceberg concept lattices (compared to frequent itemsets)



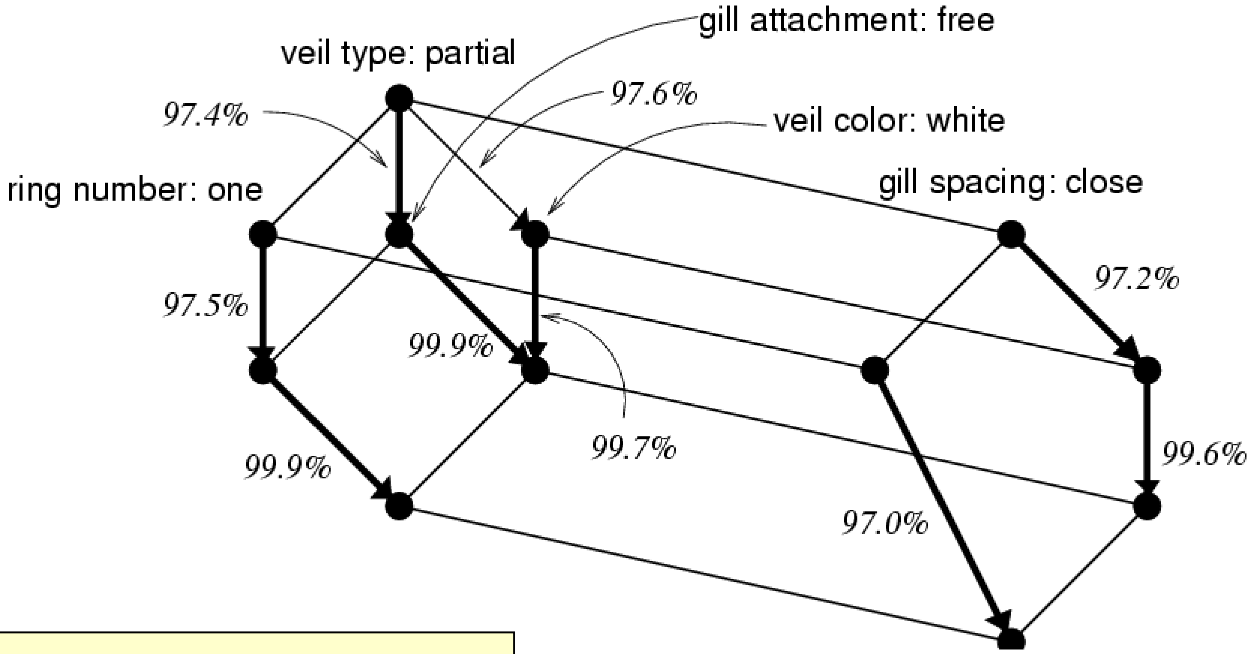
minsupp = 70%

32 frequent itemsets are represented by 12 frequent concept intents

→ **more efficient computation (e.g. TITANIC)**

→ **fewer rules (without information loss!)**

Advantage of the use of iceberg concept lattices (compared to frequent itemsets)



Association rules can be visualized in the iceberg concept lattice:

- **exact rules** ← conf = 100 %
- **approximate rules** ← conf < 100 %

Exact Association Rules

can be derived from the stem basis (Sect. 2).

In concept lattices, they can be directly read from the diagram:

- **Lemma:** An implication $X \to Y$ holds iff the largest concept which is below all concepts generated by the attributes in X is below all concepts generated by attributes in Y.

- **Examples:**
  - Swimming → Hiking

    (supp=10/19 ≈ 52.6%, conf = 100%)

  - Boating → Swimming, Hiking, NPS Guided Tours, Fishing

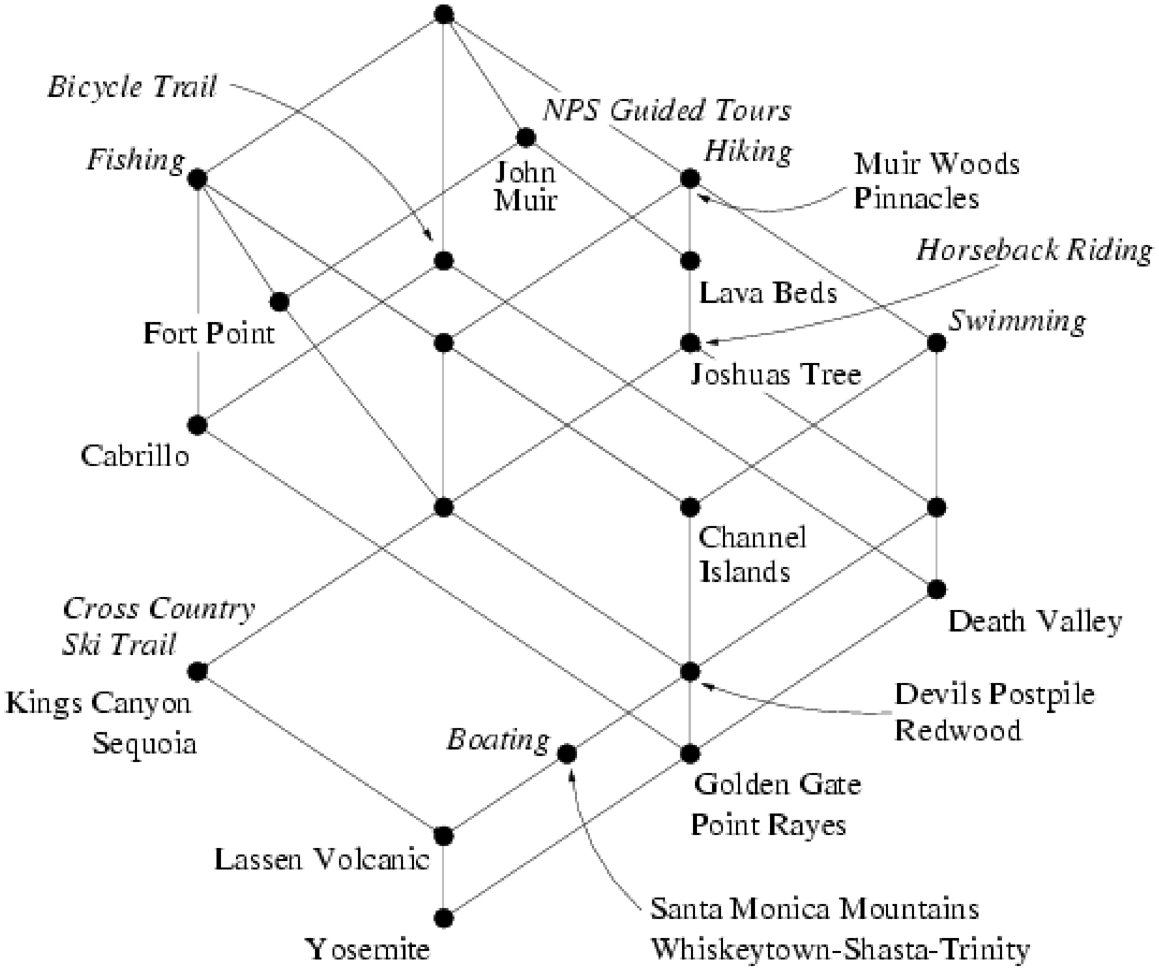
    (supp=4/19 ≈ 21.0%, conf = 100%)

  - Bicycle Trail, NPS Guided Tours → Swimming, Hiking

    (supp=4/19 ≈ 21.0%, conf = 100%)

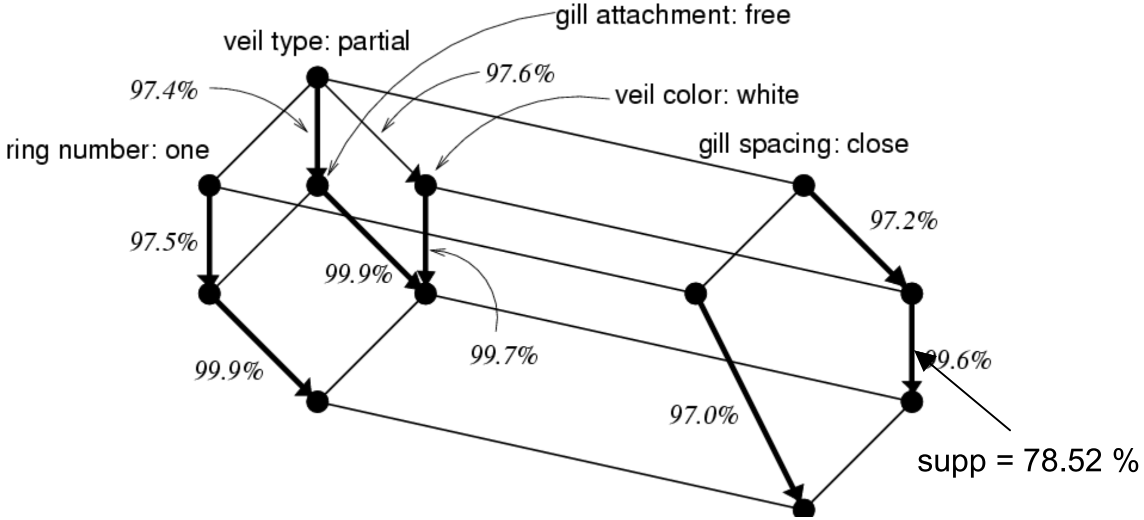
Exact Association Rules

The concept lattice of the National Parks in California



Approximate Association Rules

Def.: The **Luxenburger basis** consists of all valid association rules $X \rightarrow Y$ such that there are concepts (A_1, B_1) and (A_2, B_2) where (A_1, B_1) is a direct upper neighbor of (A_2, B_2), $X = B_1$, and $X \cup Y = B_2$.

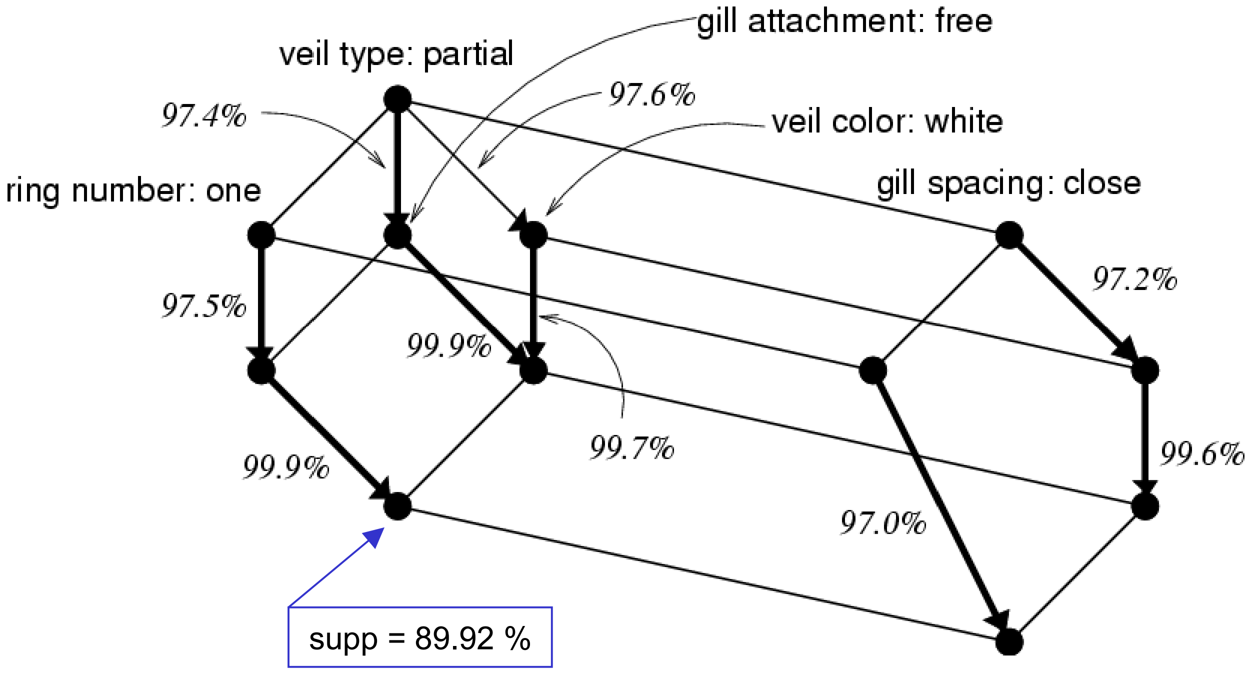


Each arrow indicates a rule of the basis, e.g. the rightmost arrow stands for
{ veil type: partial, gill spacing: close, veil color: white } → { gill attachment: free }
(conf = 99.6 %, supp = 78.52 %)

Satz: From the Luxenburger-Basis all approximate rules (incl. support und confidence) can be derived with the following rules:

- $\phi(X \to Y) = (X \to Y \setminus Z)$, für $\phi \in \{ \text{conf}, \text{supp} \}$, $Z \subseteq X$
- $\phi(X'' \to Y'') = \phi(X \to Y)$
- $\text{conf}(X \to X) = 1$
- $\text{conf}(X \to Y) = p$, $\text{conf}(Y \to Z) = q \Rightarrow \text{conf}(X \to Z) = p \cdot q$ for all frequent concept intents $X \subset Y \subset Z$.
- $\text{supp}(X \to Z) = \text{supp}(Y \to Z)$, for all $X, Y \subseteq Z$.

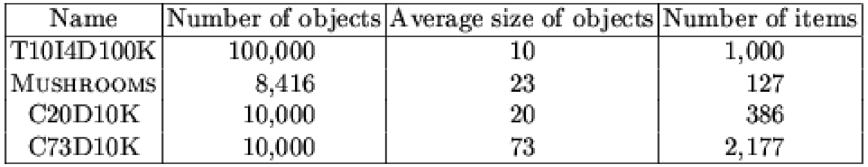
The basis is minimal with this property.



Example:

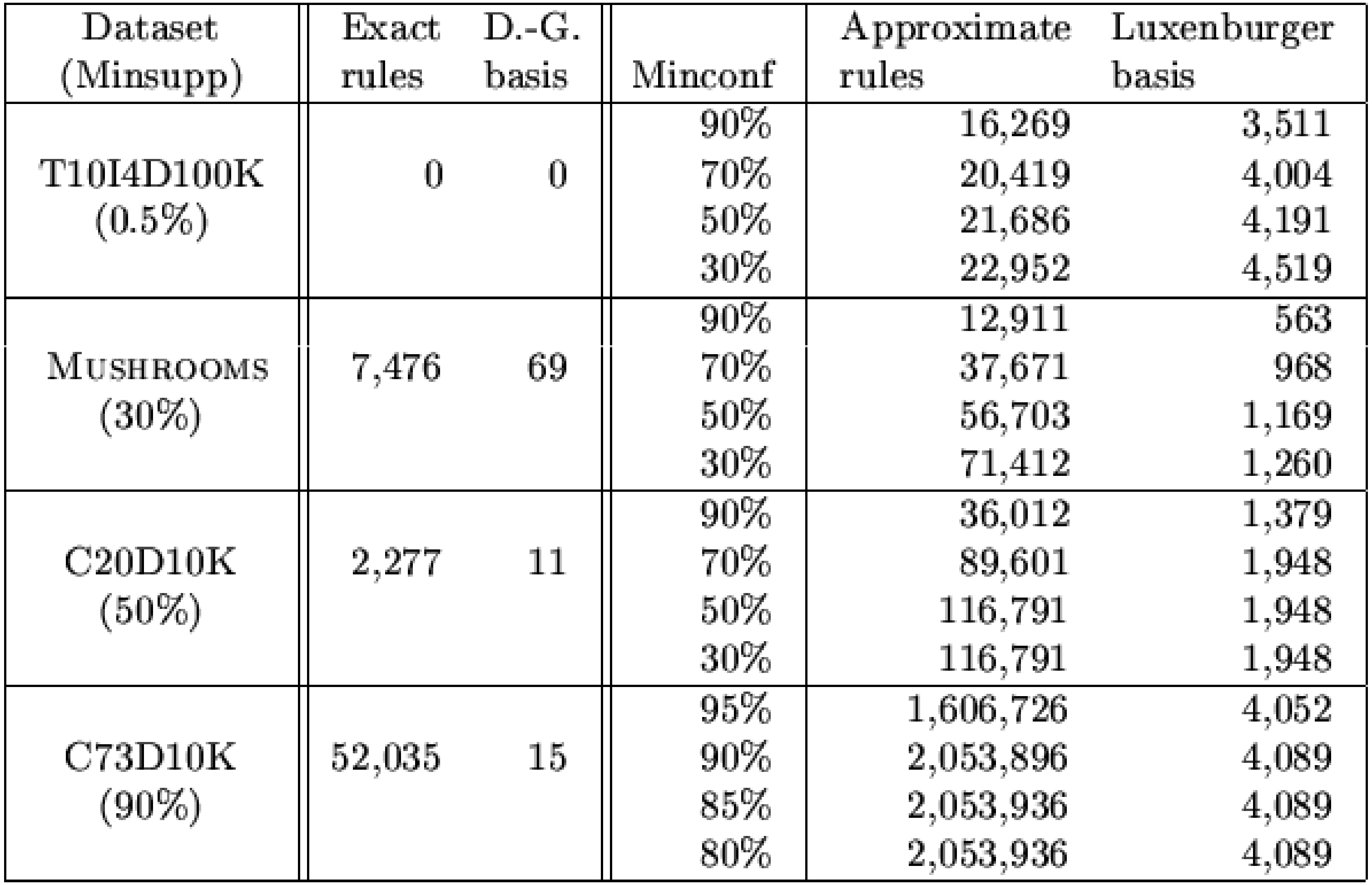
{ ring number: one } → { veil color: white }

- has support 89.92 % (the support of the largest concept having both attributes in its intent)
- and confidence 97.5 % × 99.9 % ≈ 97.4 %.



Name	Number of objects	Average size of objects	Number of items
T10I4D100K	100,000	10	1,000
MUSHROOMS	8,416	23	127
C20D10K	10,000	20	386
C73D10K	10,000	73	2,177

Some experimental results



Dataset (Minsupp)	Exact rules	D.-G. basis	Minconf	Approximate rules	Luxenburger basis
T10I4D100K (0.5%)	0	0	90%	16,269	3,511
			70%	20,419	4,004
			50%	21,686	4,191
			30%	22,952	4,519
MUSHROOMS (30%)	7,476	69	90%	12,911	563
			70%	37,671	968
			50%	56,703	1,169
			30%	71,412	1,260
C20D10K (50%)	2,277	11	90%	36,012	1,379
			70%	89,601	1,948
			50%	116,791	1,948
			30%	116,791	1,948
C73D10K (90%)	52,035	15	95%	1,606,726	4,052
			90%	2,053,896	4,089
			85%	2,053,936	4,089
			80%	2,053,936	4,089

1. Introduction
2. Formal Contexts & Concept Lattices
3. Application Examples I
4. Computing Concept Lattices
5. Exercises
6. Conceptual Scaling
7. Application Examples II
8. Conceptual Clustering
9. FCA-Based Mining of Association Rules
10. **FCA Tools**
11. Exercises

FCA Tools

- TOSCANA 2
- Anaconda
- Toscana 3
- ToscanaJ
- Cernato
- ConImp
- ConExp

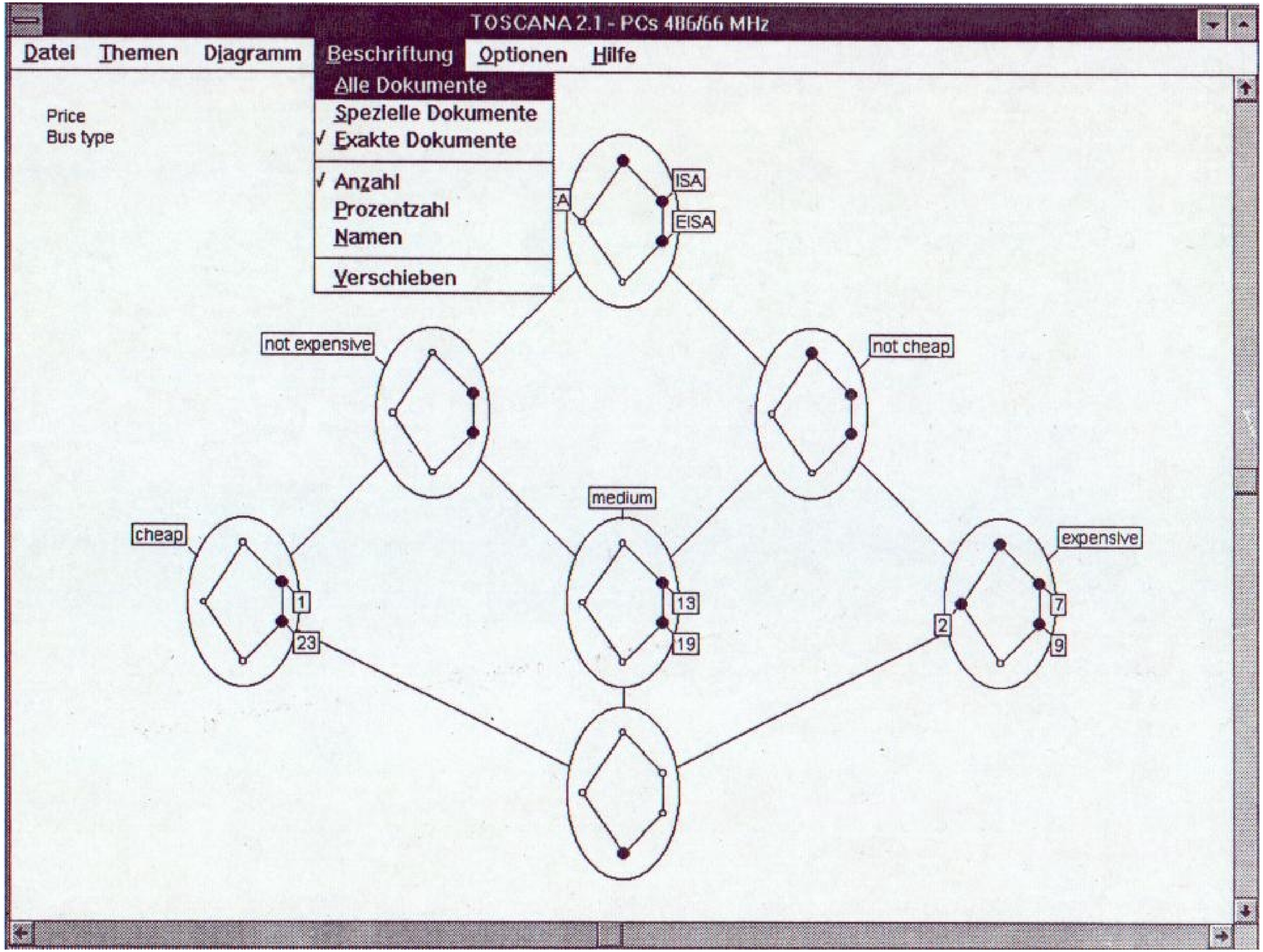
For an overview see at

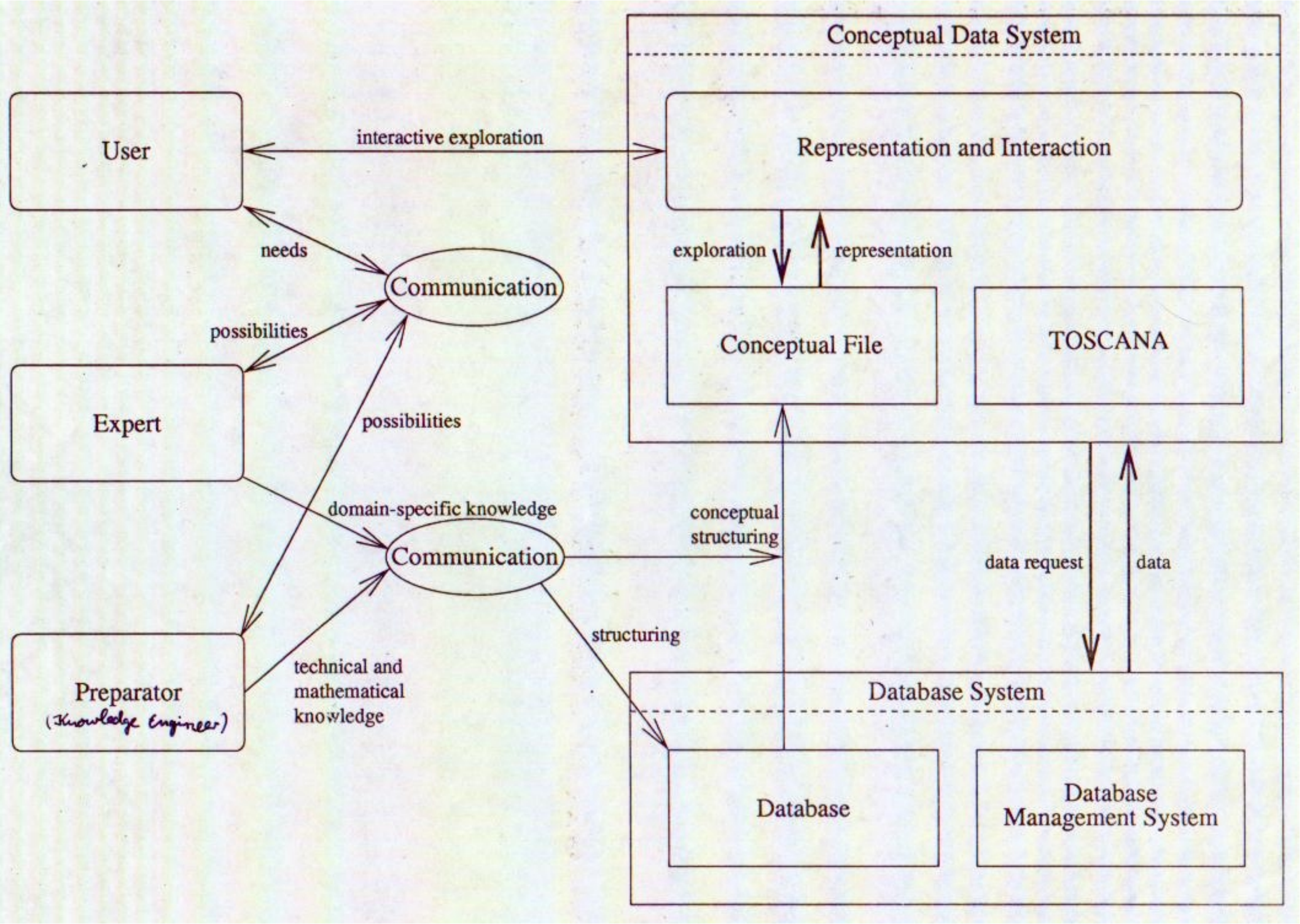
http://www.mathematik.tu-darmstadt.de/~plueschke/fcatools/programs.html

FCA Tools

- **TOSCANA 2**
- Anaconda
- Toscana 3
- ToscanaJ
- Cernato
- ConImp
- ConExp

- visualizes nested line diagrams
- accesses all ODBC databases (where the context is stored)
- the conceptual scales are stored in a proprietary format .csc
- the scales have to be prepared in advance, using Anaconda
- is part of the Navicon Decision Tool Suite (together with Anaconda and Cernato)
- available from Navicon (research licence): www.navicon.de

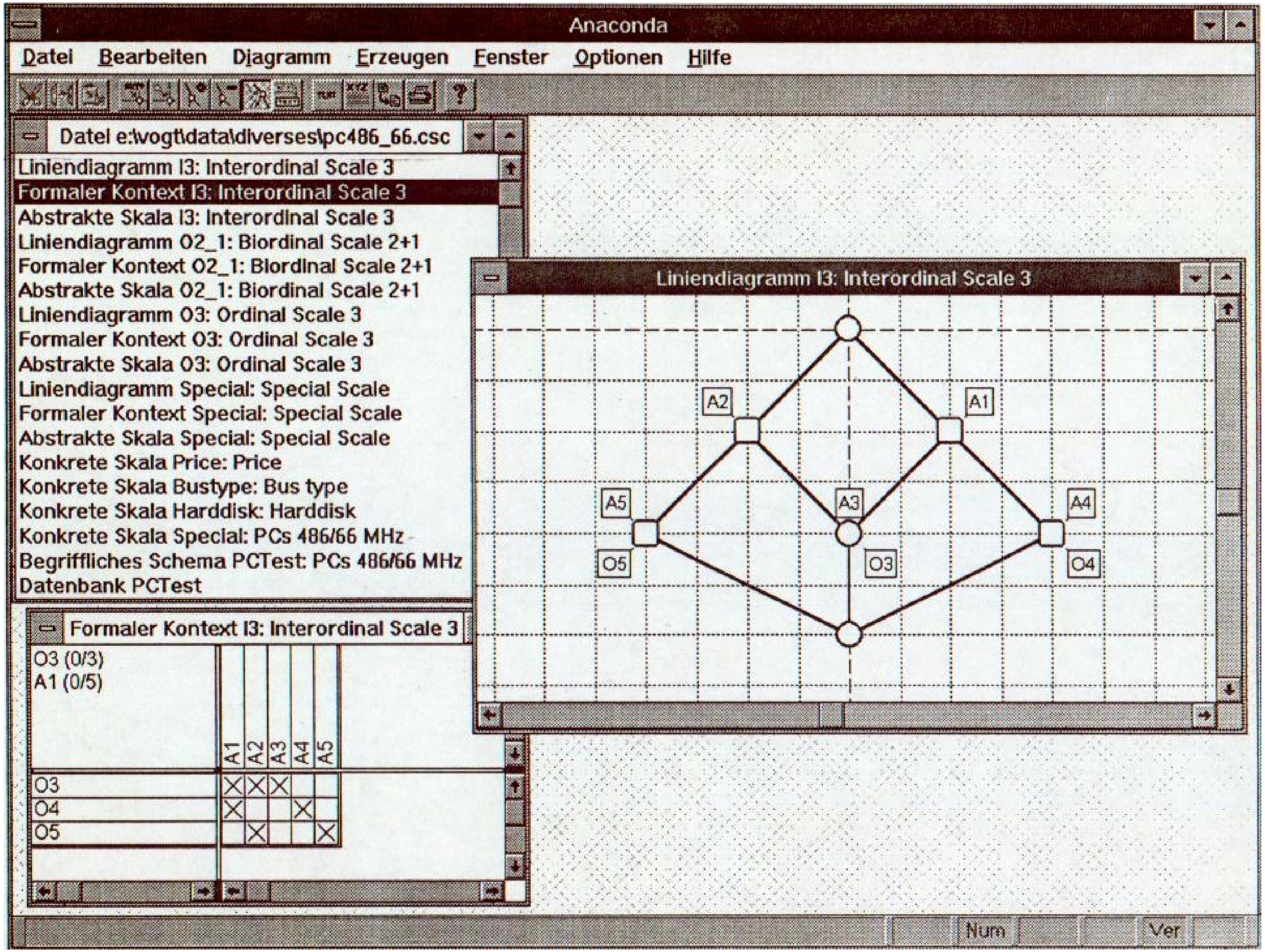

TOSCANA 2.1 - PCs 486/66 MHz
Datei
Themen
Diagramm
Beschriftung
Optionen
Hilfe
Alle Dokumente
Spezielle Dokumente
√ Exakte Dokumente
√ Anzahl
Prozentzahl
Namen
Verschieben
Price
Bus type
ISA
EISA
A
not expensive
not cheap
medium
cheap
expensive
1
23
13
19
2
7
9


Conceptual Data System
User
interactive exploration
Representation and Interaction
needs
exploration
representation
Communication
possibilities
Conceptual File
TOSCANA
Expert
possibilities
domain-specific knowledge
conceptual
structuring
Communication
data request
data
structuring
Preparator
(Knowledge Engineer)
technical and
mathematical
knowledge
Database System
Database
Database
Management System

FCA Tools

- TOSCANA 2
- **Anaconda**
- Toscana 3
- ToscanaJ
- Cernato
- ConImp
- ConExp

- is the preparation tool for TOSCANA applications
- allows easy editing of formal contexts and concept lattices
- computes a concept lattice out of a context and provides an initial layout
- stores its data in the format .csc
- is part of the Navicon Decision Tool Suite (together with TOSCANA 2 and Cernato)
- available from Navicon (research licence): www.navicon.de



Datei e:\vogt\data\diverses\pc486_66.csc

Liniendiagramm I3: Interordinal Scale 3
Formaler Kontext I3: Interordinal Scale 3
Abstrakte Skala I3: Interordinal Scale 3
Liniendiagramm O2_1: Biordinal Scale 2+1
Formaler Kontext O2_1: Biordinal Scale 2+1
Abstrakte Skala O2_1: Biordinal Scale 2+1
Liniendiagramm O3: Ordinal Scale 3
Formaler Kontext O3: Ordinal Scale 3
Abstrakte Skala O3: Ordinal Scale 3
Liniendiagramm Special: Special Scale
Formaler Kontext Special: Special Scale
Abstrakte Skala Special: Special Scale
Konkrete Skala Price: Price
Konkrete Skala Bustype: Bus type
Konkrete Skala Harddisk: Harddisk
Konkrete Skala Special: PCs 486/66 MHz
Begriffliches Schema PCTest: PCs 486/66 MHz
Datenbank PCTest

Liniendiagramm I3: Interordinal Scale 3

Formaler Kontext I3: Interordinal Scale 3

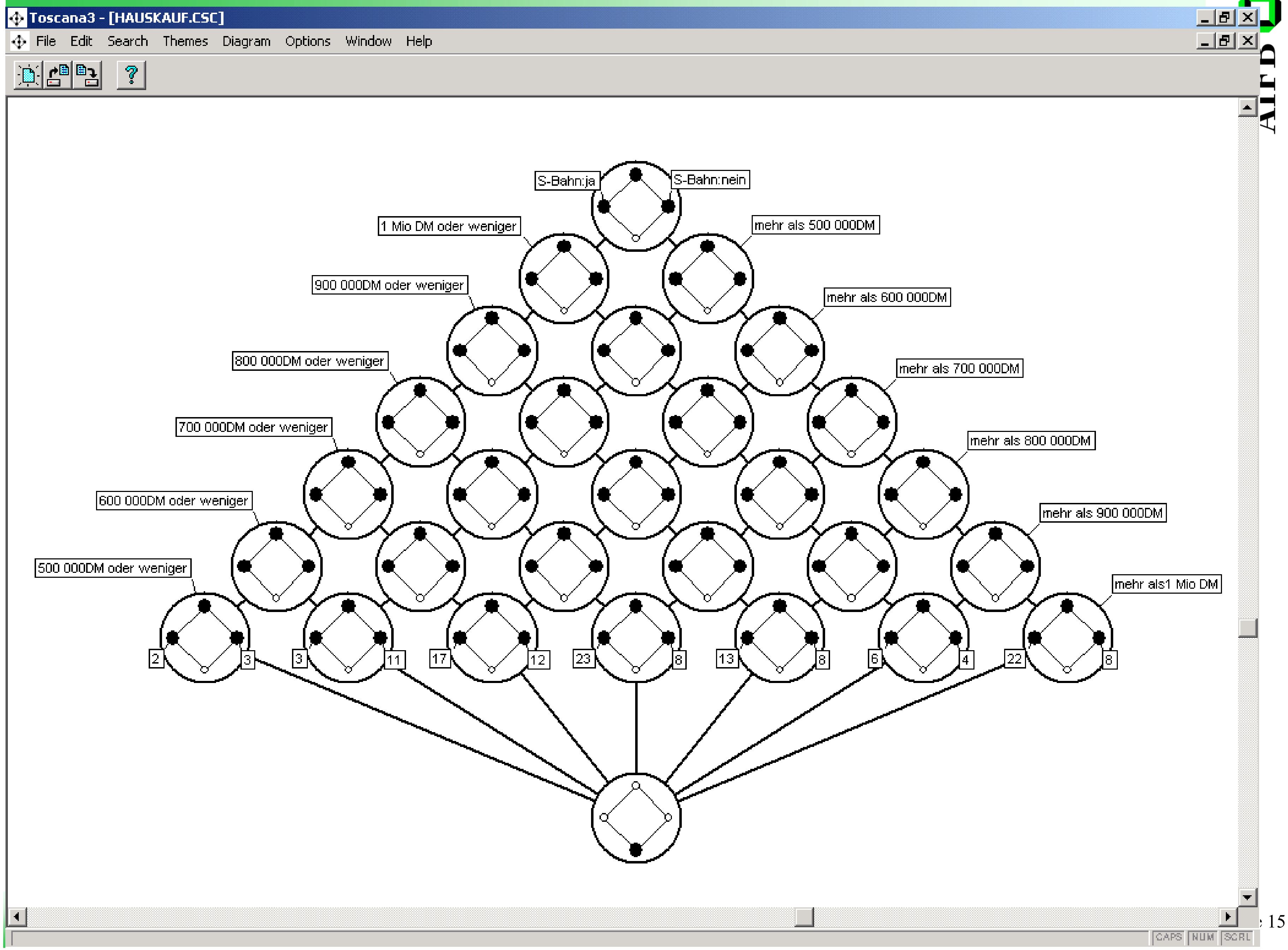
O3 (0/3)
A1 (0/5)

	A1	A2	A3	A4	A5
O3	X	X	X		
O4	X			X	
O5		X			X

FCA Tools

- TOSCANA 2
- Anaconda
- **Toscana 3**
- ToscanaJ
- Cernato
- ConImp
- ConExp

- is a C++ re-implementation of TOSCANA
- accesses all ODBC databases (where the context is stored)
- stores its data in the format .csc
- the preparation software AnacondaJ is under development
- is available from the author: B. Groh http://www.itee.uq.edu.au/people/staffView.jsp?id=bernd

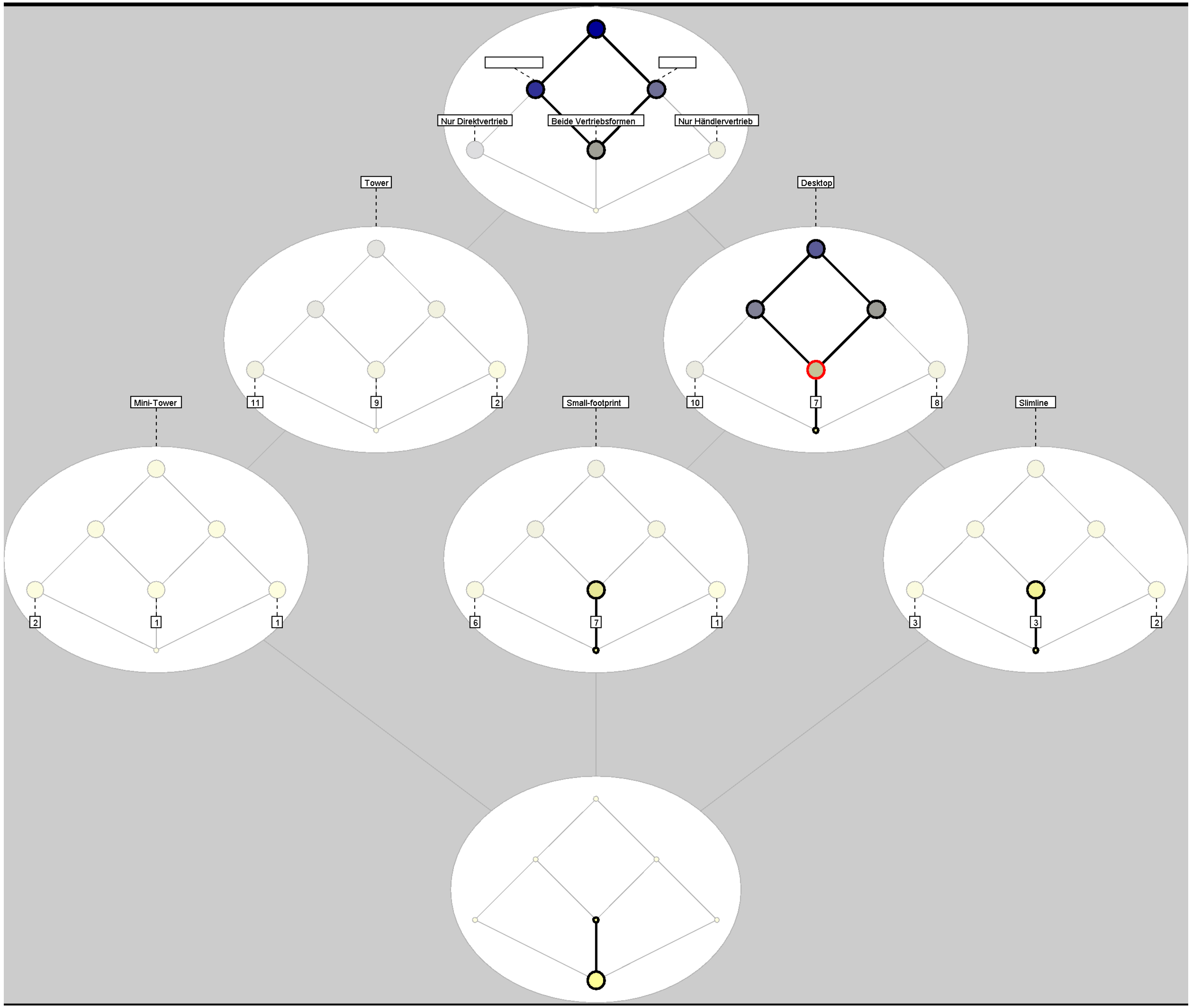


Toscana3 - [HAUSKAUF.CSC]
File
Edit
Search
Themes
Diagram
Options
Window
Help
S-Bahn:ja
S-Bahn:nein
1 Mio DM oder weniger
mehr als 500 000DM
900 000DM oder weniger
mehr als 600 000DM
800 000DM oder weniger
mehr als 700 000DM
700 000DM oder weniger
mehr als 800 000DM
600 000DM oder weniger
mehr als 900 000DM
500 000DM oder weniger
mehr als1 Mio DM
2
3
3
11
17
12
23
8
13
8
6
4
22
8
CAPS
NUM
SCRL

FCA Tools

- TOSCANA 2
- Anaconda
- Toscana 3
- **ToscanaJ**
- Cernato
- ConImp
- ConExp

- is an open source re-implementation of TOSCANA
- accesses all ODBC databases (where the context is stored)
- the conceptual scales are stored in an XML-based file format .csx
- the preparation software AnacondaJ is under development
- is downloadable from http://toscanaj.sourceforge.net/
- **Join the effort !!!**

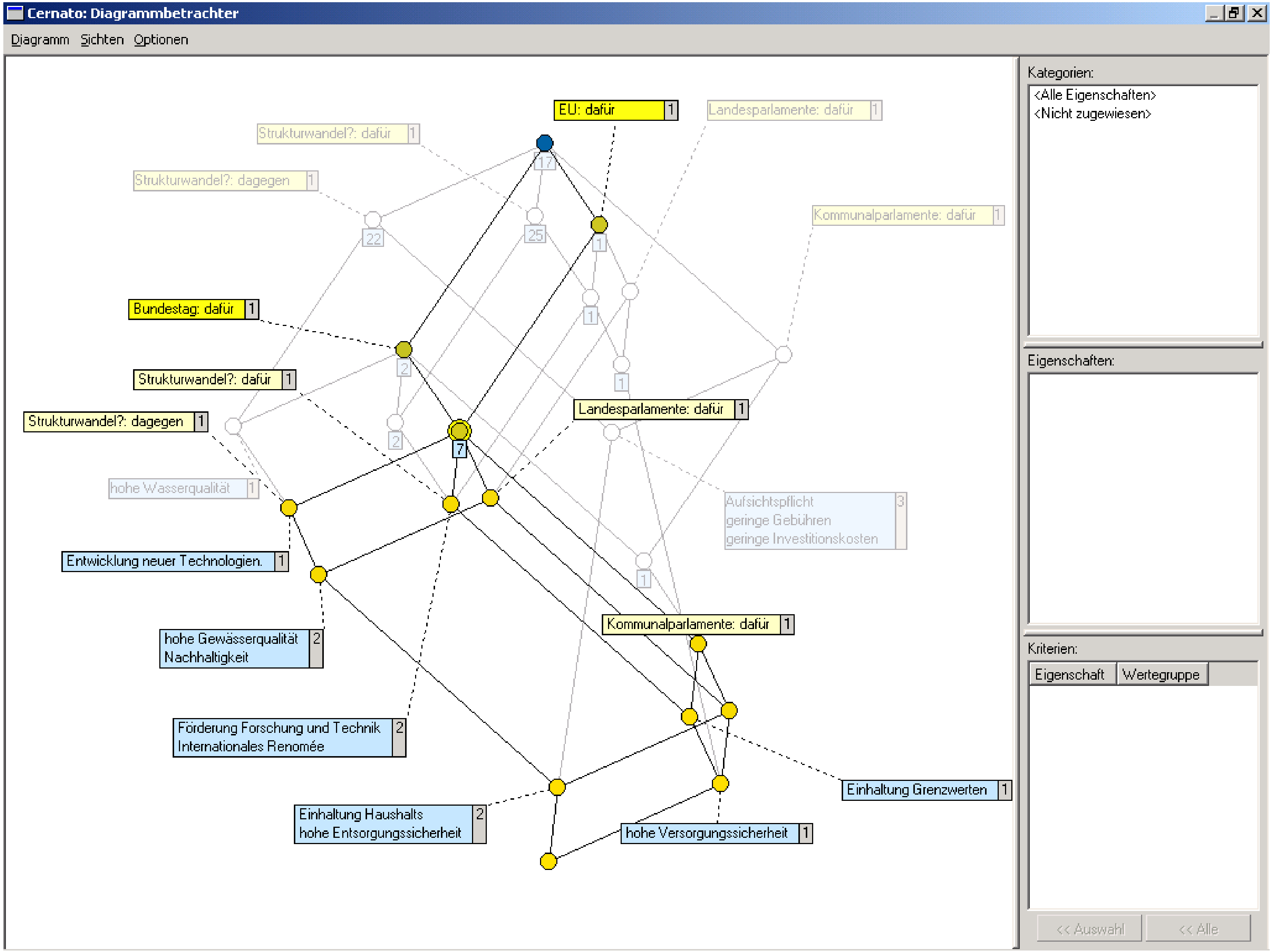

Nur Direktvertrieb
Beide Vertriebsformen
Nur Händlervertrieb
Tower
Desktop
11
9
2
10
7
8
Mini-Tower
Small-footprint
Slimline
2
1
1
6
7
1
3
3
2

FCA Tools

- TOSCANA 2
- Anaconda
- Toscana 3
- ToscanaJ
- **Cernato**
- ConImp
- ConExp

- allows ad-hoc visualization
- reads data from csv files (eg export of MS Excel)
- animates line diagrams during drawing
- is part of the Navicon Decision Tool Suite (together with Anaconda and TOSCANA 2)
- available from Navicon (research licence): www.navicon.de

Tutorial Formal Concept Analysis

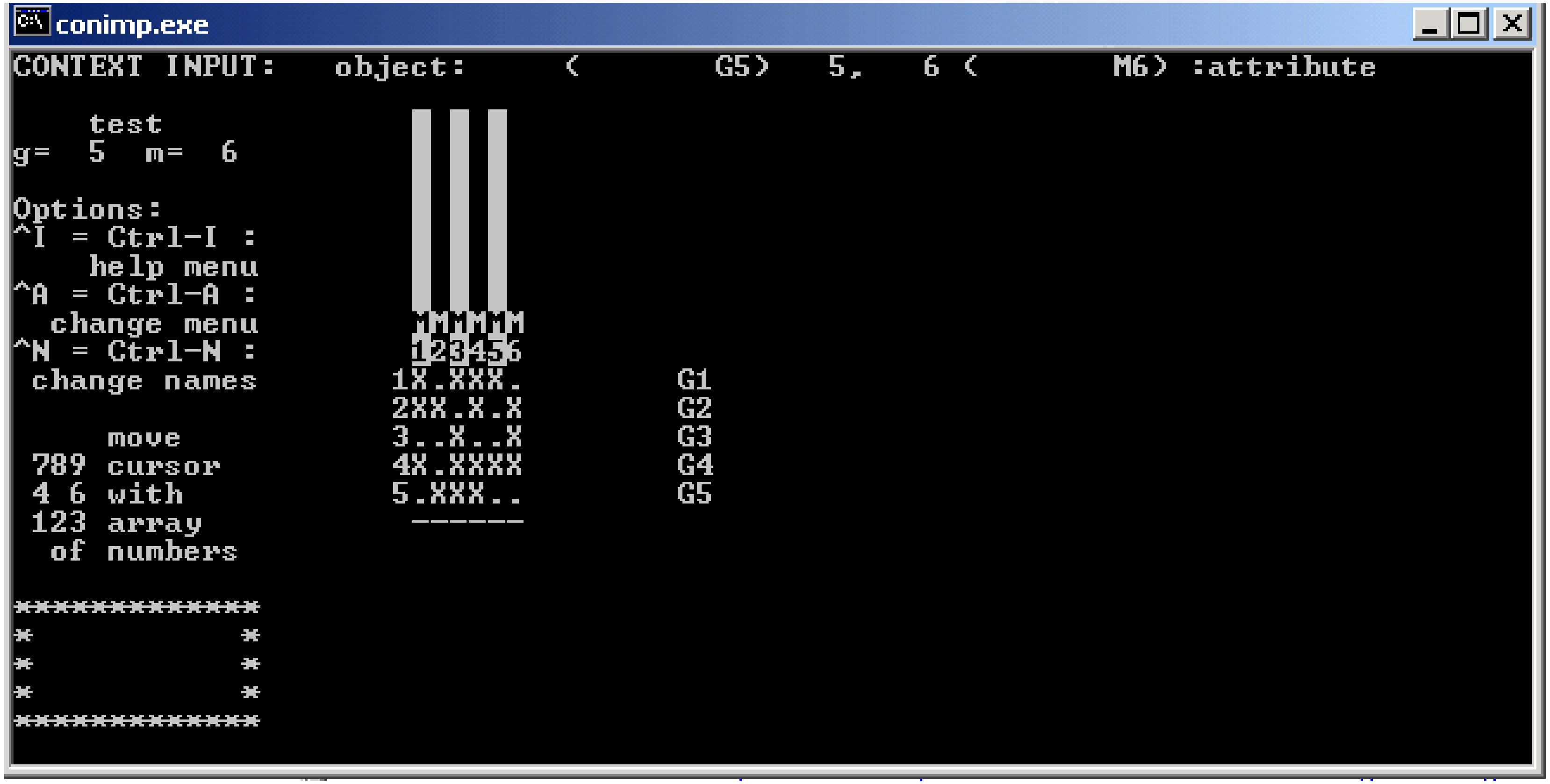


Cernato: Diagrammbetrachter
Diagramm
Sichten
Optionen
EU: dafür 1
Landesparlamente: dafür 1
Strukturwandel?: dafür 1
Strukturwandel?: dagegen 1
Kommunalparlamente: dafür 1
17
22
25
1
Bundestag: dafür 1
Strukturwandel?: dafür 1
Strukturwandel?: dagegen 1
2
7
Landesparlamente: dafür 1
hohe Wasserqualität 1
Aufsichtspflicht
geringe Gebühren
geringe Investitionskosten 3
Entwicklung neuer Technologien. 1
hohe Gewässerqualität
Nachhaltigkeit 2
Kommunalparlamente: dafür 1
Förderung Forschung und Technik
Internationales Renomée 2
Einhaltung Grenzwerten 1
Einhaltung Haushalts
hohe Entsorgungssicherheit 2
hohe Versorgungssicherheit 1
Kategorien:
<Alle Eigenschaften>
<Nicht zugewiesen>
Eigenschaften:
Kriterien:
Eigenschaft
Wertegruppe
<< Auswahl
<< Alle

FCA Tools

- TOSCANA 2
- Anaconda
- Toscana 3
- ToscanaJ
- Cernato
- **ConImp**
- ConExp

- computes implications for given contexts
- implements the knowledge acquisition technique Attribute Exploration
- is a DOS based tool
- downloadable from http://www.mathematik.tu-darmstadt.de/ags/ag1/Software/software_en.html



```
CONTEXT INPUT:   object:     (       G5)   5,   6 (      M6) :attribute

    test
g=  5  m=  6

Options:
^I = Ctrl-I :
    help menu
^A = Ctrl-A :
  change menu        MMMMMM
^N = Ctrl-N :        123456
 change names       1X.XXX.      G1
                    2XX.X.X      G2
     move           3..X..X      G3
 789 cursor         4X.XXXX      G4
 4 6 with           5.XXX..      G5
 123 array           ------
  of numbers

*************
*           *
*           *
*           *
*************
```

FCA Tools

- TOSCANA 2
- Anaconda
- Toscana 3
- ToscanaJ
- Cernato
- ConImp
- **ConExp**

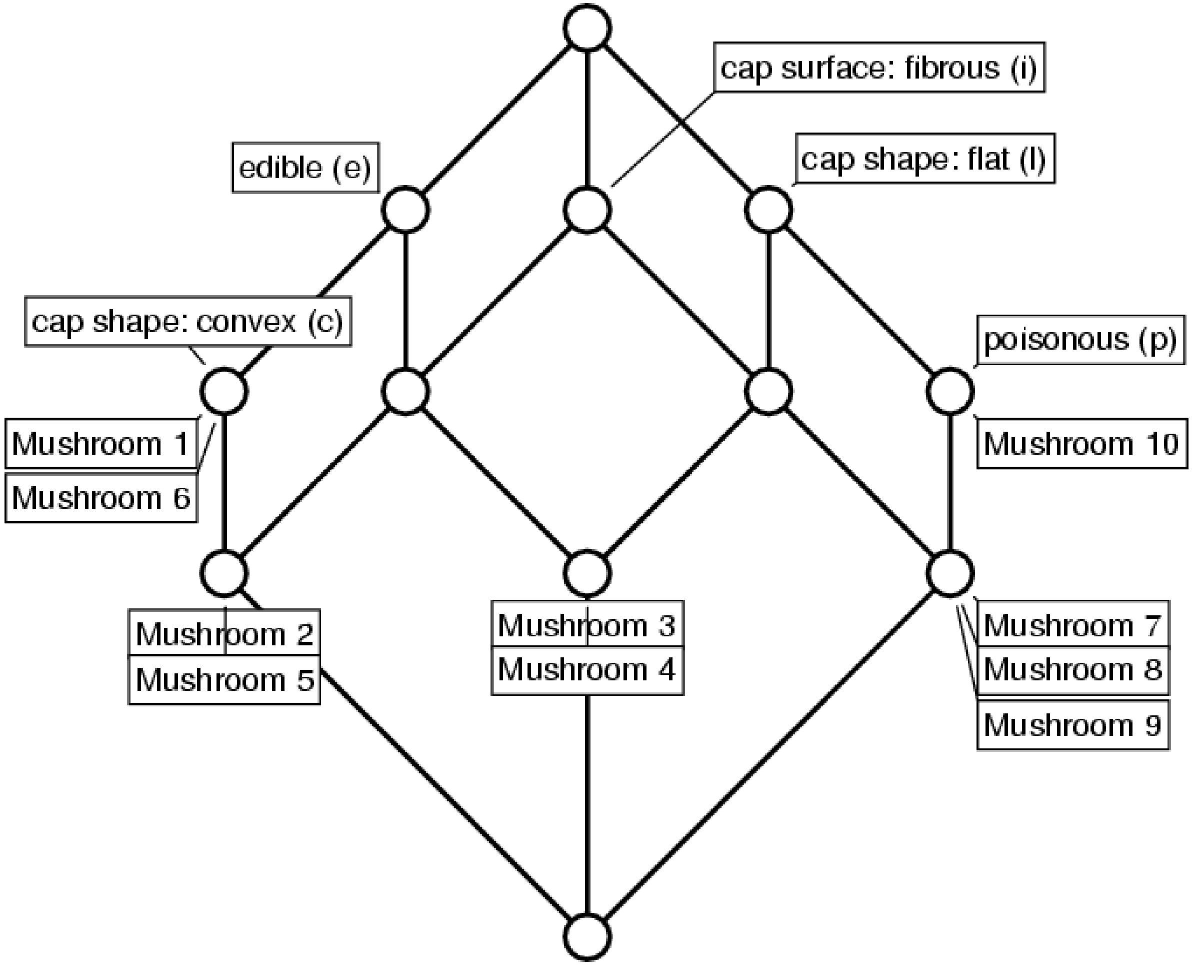
- Concept Explorer is a mixture of ConImp and Cernato.
- written by Sergey Yevtushenko in Java
- comes with its own diagram editor and also supports implications.
- is still under heavy development .
- It is planned to extend the program, so that it can be used as in exchange for Anaconda within the ToscanaJ project.
- for more information, contact the author: sergey@intellektik.informatik.tu-darmstadt.de

1. Introduction
2. Formal Contexts & Concept Lattices
3. Application Examples I
4. Computing Concept Lattices
5. Exercises
6. Conceptual Scaling
7. Application Examples II
8. Conceptual Clustering
9. FCA-Based Mining of Association Rules
10. FCA Tools
11. **Exercises**

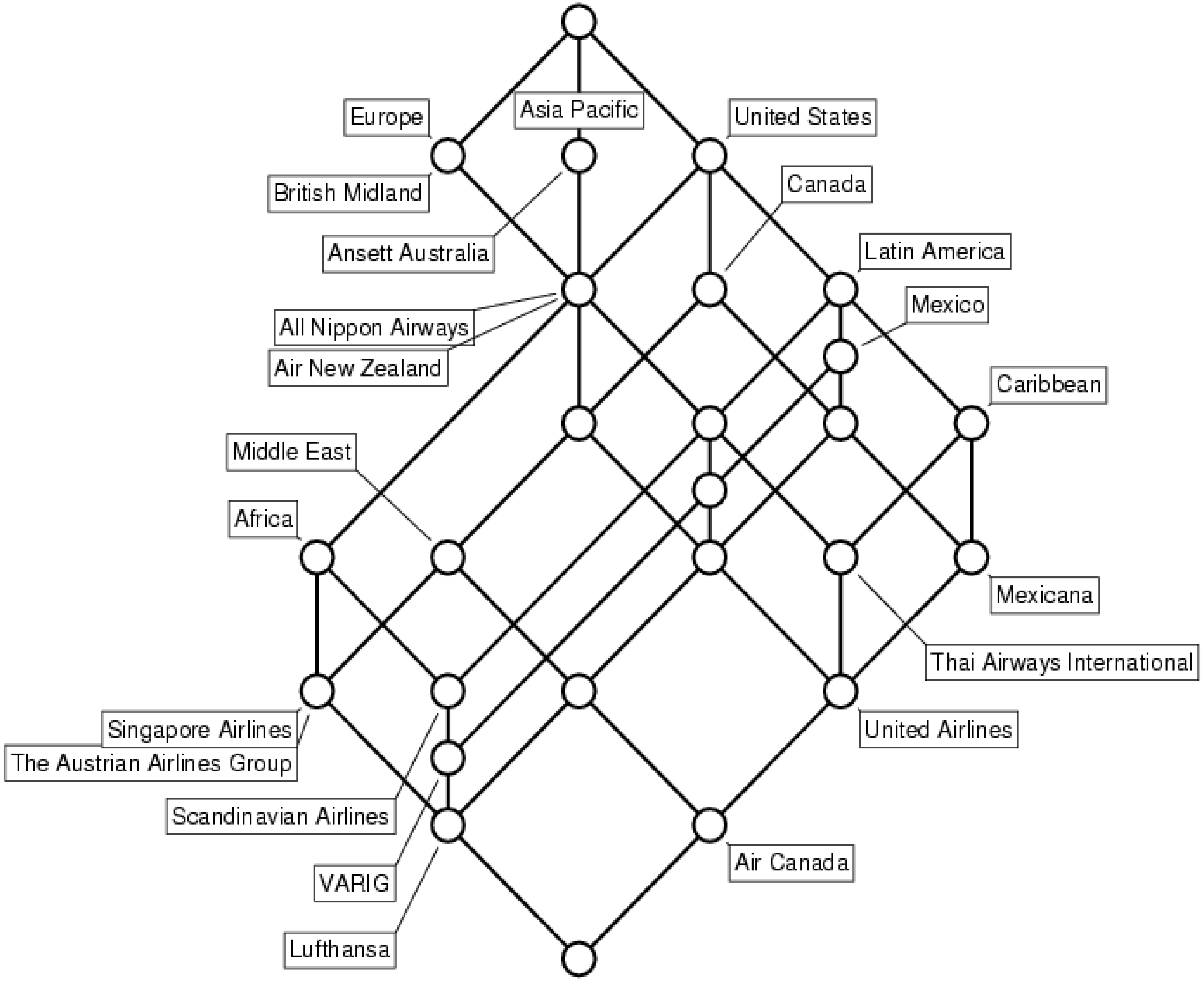
see extra exercises sheet

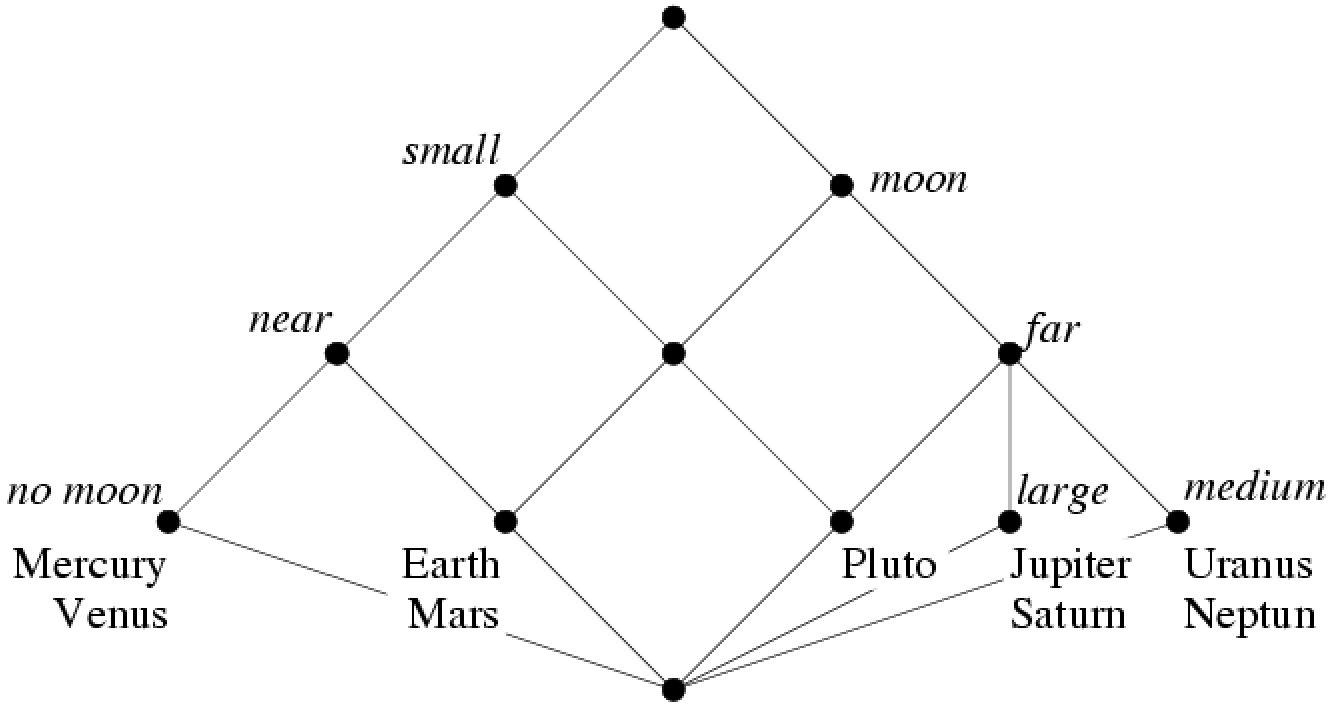
1. Introduction
2. Formal Contexts & Concept Lattices
3. Application Examples I
4. Computing Concept Lattices
5. Exercises
6. Conceptual Scaling
7. Application Examples II
8. Conceptual Clustering
9. FCA-Based Mining of Association Rules
10. FCA Tools
11. Exercises

Solutions to Sect. 5



Tutorial Formal Concept Analysis





A non-nested diagram of the context „Planets“