bases" is presented here. The full text of this communication is available at:
http://www.public.iastate.edu/~CYBERSTACKS/DPC97.htm.

This collection of papers presents different views regarding subject access and its future. All of them contribute to information science theory by helping prepare the ground for accompanying changes in modes of communicating and storing knowledge.

Maria Salet Novellino

Maria Salet Novellino, Postgraduate Program in Information Science, Rio de Janeiro, Brazil, e-mail: salet@trendnet.com.br

GANTER, Bernhard, and WILLE, Rudolf. Formal Concept Analysis : Mathematical Foundations. Berlin : Springer-Verlag, 1999. 284 p. ISBN 3-540-62771-5.

Formal concept analysis is a theory of conceptual data analysis and knowledge processing. A concept is formalized as an element of a mathematical lattice. Formal concept analysis is thus a field of applied mathematics and competes with and is related to other formalisms in the area of knowledge representation, such as the relational database model and conceptual graphs.

In formal concept analysis, a formal context consists of a set of data considered to be objects, a set of data considered to be attributes and a relation between. A context is graphically represented by a cross table that has a row for each object, a column for each attribute and a cross for each element of the relation. From a context, a set of concepts is derived. Among the concepts, a subconcept-superconcept relation holds. Together with that relation the set of concepts forms a so-called concept lattice. An advantage of this data analysis approach is that conceptual hierarchies can be graphically represented via lattices and that a large inventory of mathematical operations, such as operations for constructing, combining and decomposing contexts and lattices, can be utilized for studying data. Furthermore, dependencies and implications among attributes are visualised in the lattice modelling. Formal concept analysis can therefore incorporate reasoning procedures and be extended to a conceptual logic.

As stated in the preface, this volume covers solely the mathematical foundations of formal concept analysis. A detailed representation of applications and methods and of philosophical and other non-mathematical foundations is left for future volumes. Furthermore, conceptual logic which has been greatly elaborated in the years since the publication of the first German edition of this book in 1996 is also left for future volumes, with the exception of attribute implications and dependencies which are covered in chapter two.

This is the first comprehensive publication on the mathematical foundations of formal concept analysis. Although the book is self-contained because chapter zero explains all necessary order-theoretical mathematical terms, the book is not aimed at a general audience; readers should have a strong mathematical background to fully appreciate its content. On the other hand, readers who are not mathematicians but are already familiar with formal concept analysis may find chapters one and two, which comprise one third of the book, very helpful as a source of definitions, hints, lists of scales, and examples. The book contains an extensive bibliography and every chapter ends with a section called "Hints and References" that gives detailed information on where the results of each chapter were first published and hints for further readings.

The first chapter defines formal contexts, concepts, concept lattices, and clarification and reduction of contexts. Lattices of the reduced or clarified version of a context have the same lattice structure as the original lattice and can often be used to define or check mathematical properties of concept lattices. Another technical aid is provided by arrow relations that can be inserted into a context. Contexts and lattices in formal concept analysis can be infinitely large. Many theorems, however, hold only for subsets of contexts and lattices that have additional properties, such as doubly foundedness. Many-valued contexts are contexts that contain a third set, a set of values, such that at most one value is assigned to each object-attribute pair. Relational database tables can be interpreted as many-valued contexts. These contexts must be scaled into single-valued contexts. The last two sections of the first chapter describe the process of scaling, a list of standard scales, and simple methods for constructing contexts from other contexts.

The second chapter provides an explanation of methods for manually deriving all concepts of a context and for manually drawing easily-readable lattice diagrams using the geometrical method. Software tools for formal concept analysis exist that automate these methods. These tools are often based on an algorithm for generating concepts that produces the concepts in their lectic order and an algorithm for generating preliminary versions of diagrams that are then interactively refined. Nested line diagrams are described as a method of reducing the visual complexity of lattice diagrams. Implications among attributes lead to the notion of an attribute logic. A basis is defined for the set of attribute implications of a lattice. By means of attribute exploration, sets of implications that are complete and non-redundant can be generated even for an incomplete initial context. For this procedure a computer program questions a user whether certain implications hold or whether a counter example exists. In the case of many-valued contexts, attribute dependencies, such as functional or ordinal dependencies, are investigated.

The third chapter describes parts of contexts and lattices. In this and the next two chapters the correspondence between lattices and contexts is essential. An operation or structure of a context is always analyzed with respect to how it can be recognized in the corresponding lattice, and vice versa. Subcontexts are substructures of contexts that are created by deleting rows and columns from a context. The lattice of a compatible subcontext is isomorphic to a factor lattice of the original lattice. The process of finding compatible subcontexts corresponds to finding complete congruence relations in lattices. Complete sublattices are substructures of lattices that correspond to certain subrelations of the object-attribute relation in the context. Tolerance relations in lattices are generalized congruence relations and correspond to block relations in contexts.

The fourth and fifth chapters describe decompositions and constructions of concept lattices, which facilitate splitting a complex lattice into simpler parts and composing a complex lattice from smaller lattices. A sub-direct decomposition represents a lattice as sublattice of a direct product of lattices. Atlas decompositions are constructed in analogy to a geographical atlas: a lattice is decomposed into small lattices (similar to maps in an atlas) whose indices form another lattice (similar to an index map in an atlas). Atlas decompositions use tolerance relations and can be utilized to improve the graphical representation of lattice diagrams. Other forms of decompositions use the substitution product and the tensor product.

The construction methods in chapter five are: the subdirect product of lattices which corresponds to fusion of contexts; gluing of lattices which is similar to atlas composition and composes lattices by gluing them together along common substructures; local doubling which is based on duplicating a convex subset of a lattice; and further tensorial constructions.

Chapter 6 characterizes the mathematical properties of concept lattices, such as distributivity, modularity and dimension. Some of these characteristics are in addition to the ones in traditional lattice theory because the inventory of formal concept analysis includes objects, attributes and relations such as the arrow relations defined in chapter one. At the end of the third section the implications among the mathematical properties are conveniently visualized in a lattice diagram. The discussion of different types of lattice dimensions includes an example of an application of the set dimension. The example shows how the digits of the seven-segment display, which is the display of digits often used by digital watches and old fashioned calculators, can be represented as a union of less than seven parts.

The last, seventh chapter discusses mappings (morphisms) among contexts. Scale measures are morphisms that facilitate a comparison between scales as to whether one scale is finer or coarser than the other. This is relevant for the process of scaling many-valued contexts into single-valued contexts as described in chapter two. The book ends with a discussion of concept-analytic measurement theory which is based on a process inverse to scaling: data given in a single-valued context is explained in terms of a many-valued context with appropriate scaling.

Overall the book is impressive. It demonstrates how an abstract notion of "concept" has been developed into a branch of applied mathematics by the two authors and with the help of numerous master's and Ph.D. theses over the past 20 years. Hopefully, the other two volumes announced in the preface will follow in the future.

Uta Priss

Dr. Uta Priss, School of Library and Information Science, Indiana University Bloomington, 10th & Jordan, Bloomington, IN 47405, e-mail: upriss@indiana.edu

ORTNER, von Erich. **Methodenneutraler Fachentwurf : Zu den Grundlagen einer anwendungsorientierten Informatik** [Method-neutral Engineering requirements : A contribution to the fundamentals of an application oriented computer science]. Teubner-Reihe Wirtschaftsinformatik, herausgegeben von Dieter Ehrenberg, Dietrich Seibt, Wolffried Stucky. Leipzig: B.G. Teubner Verlagsgesellschaft 1997. 196 p. ISBN 3-8154-2602-2.

Ortner's book presents a systematic process for programming application systems in the business environment. The author concentrates on the early phase of identifying and representing the needs of users and of the environment of application, the so-called engineering requirements.

Ortner characterizes the conventional methods of representation as method-specific engineering that should be replaced by a method-neutral one. To do so, he develops a language through a reconstruction of concepts from a domain discourse. In this language, the things, phenomena, activities and characteristics of the application may be expressed. This language will be called *material*, since it consists of a dictionary with standardized terms belonging to a subject field, and so-called structural words to express relationships and characteristics of (subject)-terms relationships.

The problem with engineering requirements is twofold. On the one hand, one has to represent application