# Chapter 1: Introduction

Formal Concept Analysis (FCA) is an established area of research in the computer sciences with many areas of application especially in branches of artificial intelligence. Central to FCA is the notion of a formal concept lattice (or concept lattice for short). A concept lattice is a mathematically well-defined structure that comprises of a number of concepts describing a context. Each concept has an extent consisting of a number of objects from the context as well as the attributes that these objects have in common within the context. The concepts are ordered in a partial order and form an order-theoretic lattice. Chapter 2 defines and gives examples of concept lattices.

Concept lattices have been proposed and studied in a number of areas of application:

- Data analysis (Vogt and Wille (1995)).
- Discovery of association rules (Stumme et al. (1998, 2000), Pasquier et al. (1999)).
- Information retrieval (Godin et al. (1995a), Carpineto and Romano (1996a)).
- Exploration of attributes and attribute relationships in data (Ganter (1999), Duquenne (1999, 2001)).
- Conceptual clustering and classification (Godin et al. (1991), Carpineto and Romano (1996b)).
- Machine learning (Oosthuizen (1994b), Mephu Nguifo and Njiwoua (2001), Xie et al. (2002)).
- Computer assisted human browsing and dissemination of data (Cole and Stumme (2000), Cole and Eklund (2001)).

FCA was proposed in the early eighties by Wille (1982). Ganter and Wille (1999) now serves as the foundation of FCA. FCA builds on the work of Birkhoff (1973) and Barbut and Monjardet (1970). In FCA, the problem of generating the set of concepts of a concept lattice and then constructing the line diagram to represent the concept lattice has been well-studied (refer to Kuznetsov and Obiedkov (2002) for an overview and comparison). Chapter 3 describes the basic challenges of this problem.

In the worst case, concept lattices are however exponential in size in terms of the input context. Although, in general, natural data does not realize the worst case, in practical applications very large lattices can still result. This creates time and space performance issues for applications using lattices and therefore every effort should be made to more efficiently construct lattices. This dissertation approaches the efficiency problems from two points of view and proposes complementary solutions for each. First a new lattice construction algorithm, called AddAtom, unrelated to well known and published algorithms is proposed. The algorithm efficiently constructs lattices and is proposed as a general purpose lattice construction algorithm that outperforms other published algorithms in a wide range of contexts (although not in all types of contexts). The second proposed approach to managing the time taken to build the lattice is to construct sub-lattices instead of the complete lattice of all concepts. A generic framework and the necessary operations for building such lattices are proposed and defined. The resulting data structure is called a compressed pseudo-lattice. Not only does this approach allows for the incremental scaling of the lattice size in relation to the amount of time that an application can afford to spend on constructing lattices, but initial evidence suggest a number of other advantages, despite the removal of a large number of concepts from the lattice.

**AddAtom**

The AddAtom algorithm is explained and defined in chapter 4. When inserting a new object into a lattice $L_n$, the basic strategy of the algorithm is to start at the zero concept in $L_n$ and then recursively search its parent concepts for generator concepts in an elegant and tightly focussed search. After creating the required new concepts and arcs associated with the generator concepts and modifying the relevant concepts a new lattice $L_{n+1}$ is produced.

The algorithm is characterised by a number of features that differentiates it from some of the other construction algorithms. Firstly, it is an incremental lattice construction algorithm and therefore constructs a lattice $L_{n+1}$ from an input lattice $L_n$ by adding an additional object to the context of $L_n$. Secondly, it creates the set of concepts and the line diagram of the lattice. Thirdly, it is defined on concept lattices as well as on concept sublattices, which makes it suitable to be used in generating compressed pseudo-lattices.

Fourthly, AddAtom is defined in terms of a class of lattice operations, called the intent- and extent representative operations of a lattice. These operations elegantly define the nature of lattice construction and their use to traverse and inspect a lattice is significantly different from most other approaches that use the supremum and infimum operations. These operations are in some sense second order supremum and infimum operations and are useful in situations where, for example, the infimum or supremum of a set of attributes in a lattice is trivial (i.e. either unit or the zero concept). Throughout the dissertation it is argued that these operations are very useful in traversing and searching the lattice for areas of interest. This is underpinned by the fact that the intent- and extent representative operations are key to the definition of AddAtom as well as compressed pseudo-lattices. Chapter 6 also provides examples of how these operations may be useful in information retrieval (IR) and machine learning.

During the past few decades a number of lattice construction algorithms have been proposed and although not all the algorithms are directly comparable due to the fact that all do not produce the same outputs (e.g. some generate the line diagrams while others merely generate the set of concepts). In Chapter 5 a worst-case complexity bound of AddAtom is established as $O(\|L\|.\|O\|^2.\max(\|O'\|))$. This bound is cubic in nature relative to the lattice size. Although this bound is not of the same order of magnitude as that of the lowest known for lattice construction algorithms, this does not necessarily mean that the algorithm has a worse performance.

The results of experimental comparisons of AddAtom with other lattice construction algorithms (chapter 5) show that AddAtom is a very good lattice construction algorithm and compares well with other algorithms from an experimental point of view. It does however not perform the best across all types of contexts. In random contexts with either very low or very high densities other algorithms perform slightly better than AddAtom. In these contexts AddAtom is still the second best performer. These results are consistent with the claim that the theoretical bound is not very sharp, confirming that the algorithmic performance of AddAtom is very good and that it is a worthy candidate for use as a general purpose lattice construction algorithm in that it is efficient (compared to other algorithms) over the range of artificial and natural data sets albeit not over all types of contexts.

**Compressed pseudo-lattices**

Compressed pseudo-lattices are defined in chapter 6 and provide a formal framework within which concepts can be removed from lattices to create sublattices. The data structure, essentially a bipartite graph that incorporates an embedded sublattice, combines desirable features of concept lattices in a structure that allows for a flexible mechanism of scaling the size of the embedded sublattice, using defined operations that compress and expand it by removing or adding atoms and coatoms. A compressed pseudo-lattice essentially represents a lattice from which a number of atoms and/or coatoms have been removed. Additionally the relation of the sublattice to the context from which it was derived is preserved. An application-dependent compression strategy or criterion is required to guide this process. It is argued that the removal of concepts from a concept lattice may hold advantages over traditional approaches.

The implementation of the algorithms developed for supporting the AddAtom and compressed pseudo-lattice implementations are discussed in chapter 7. A number of implementation issues and considerations as well as the strategies to deal with these are also discussed.

Finally, chapter 8 summarise the findings of the dissertation and ends by identifying areas of further research related to AddAtom and compressed pseudo-lattices.