Efficiently Computing a Linear Extension of the Sub-hierarchy of a Concept Lattice

Anne Berry[1], Marianne Huchard[2], Ross M. McConnell[3], Alain Sigayret[1], and Jeremy P. Spinrad[4]

[1] LIMOS (Laboratoire d'Informatique de Modélisation et d'Optimisation des Systèmes), CNRS UMR 6158, Université Clermont-Ferrand II, Ensemble scientifique des Cézeaux, 63177 Aubière Cedex, France
{berry, sigayret}@isima.fr
[2] LIRMM (Laboratoire d'Informatique de Robotique et de Micro-électronnique de Montpellier), CNRS UMR 5506, Université Montpellier II, 161 rue Ada, 34392 Montpellier cedex 5, France
huchard@lirmm.fr
[3] EECS Department, Vanderbilt University, Nashville, TN 37235 USA
spin@vuse.vanderbilt.edu
[4] Computer Science Department, Colorado State University, Fort Collins, CO 80523-1873 USA
rmm@cs.colostate.edu

Abstract. Galois sub-hierarchies have been introduced as an interesting polynomial-size sub-order of a concept lattice, with useful applications. We present an algorithm which, given a context, efficiently computes an ordered partition which corresponds to a linear extension of this sub-hierarchy.

1 Introduction

Formal Concept Analysis (FCA) aims at mining concepts in a set of entities described by properties, with many applications in a broad spectrum of research fields including knowledge representation, data mining, machine learning, software engineering or databases. Concepts are organized in concept (Galois) lattices where the partial order emphasizes the degree of generalization of concepts and helps to visually apprehend sets of shared properties as well as groups of objects which have similarities.

The main drawback of concept lattices is that the number of concepts may be very large, or even exponential in the size of the relation.

One of the options for dealing with this problem is to use a polynomial-size representation of the lattice which preserves the most pertinent information. A way of doing this is to restrict the lattice to the concepts which introduce a new object or property, leading to two similar structures called the 'Galois sub-hierarchy' (GSH) and the 'Attribute Object Concept poset' (AOC-poset). GSH has been introduced in the software engineering field by Godin and Mili in 1993

B. Ganter and R. Godin (Eds.): ICFCA 2005, LNAI 3403, pp. 208–222, 2005.

([11]) for class hierarchy reconstruction and successfully applied in later research works ([12, 20, 14, 6]).

Recent work has shown interest of GSH in an extension of FCA to Relational Concept Analysis (RCA); RCA has been tested to identify abstractions in UML (Unified Modeling Language, see [19]) diagrams allowing to improve such diagrams in a way that had not been explored before ([7]). AOC-poset has been used in applications of FCA to non-monotonic reasoning and domain theory ([13]) and to produce classifications from linguistic data ([17, 16]). Considering AOC-poset or GSH is interesting from two points of view, namely the algorithmic and the conceptual (human perception), because the structure which is used has only a restricted number of elements.

Several algorithms have been proposed to construct the Galois sub-hierarchy, either incrementally or globally. Incremental algorithm Ares [8] and IsGood [10] add a new object given with its property set in an already constructed GSH. The best worst-case complexity is in $O(k^3\, n^2)$ for Ares (in $O(k^4\, n^2)$ for IsGood) where k is the maximal size of a property set and n is the number of elements in the initial GSH. Note that best pratical results are nevertheless obtained by IsGood. The global algorithm Ceres ([15]) computes the elements of the GSH as well as the order and has worst case complexity in $O(|O|\,(|O| + |P|)^2)$.

In this paper, we present an algorithm which outputs the elements of the GSH in a special order, compatible with a linear extension of the GSH: roughly speaking, we decompose each element of the GSH into an extent (which is the set of objects of this element) and an intent (which is the set of properties of this element), and we output a list of subsets of objects and properties, such that if E_1 is a predecessor of E_2 in the GSH, then both the intent and the extent of E_1 are listed before the intent and extent of E_2 in our output ordering.

To do this, we use a partition refinement technique, inspired by work done in Graph Theory, which can easily be implemented to run in linear time. We used this in previous works to improve Bordat's concept generation algorithm (see [4]), in order to rapidly group together the objects (or, dually, the properties) which are similar. Partition refinement has also been used to reorder a matrix ([18]).

One of the interesting new points of the algorithm presented here is that it uses the objects and properties at the same time, instead of just the objects or just the properties.

The other interesting development is that the orderings on the properties and objects created by our algorithm define a new representation of the input relation, essentially by re-ordering its rows and columns, creating a zone of zeroes in its lower right-hand corner.

The paper is organized as follows: in Section 2, we give a few necessary notations, and present a running example which we will use throughout the paper to illustrate our work. Section 3 presents some results from previous papers, and explains the general algorithmic process which is used. Section 4 gives the algorithm, as well as some interesting properties of the output. The algorithm is proved in Section 5.