Performances of Galois Sub-hierarchy-building Algorithms

Gabriela Arévalo[1,3], Anne Berry[2], Marianne Huchard[1], Guillaume Perrot[1], and Alain Sigayret[2]

[1] LIRMM - CNRS UMR 5506 - Université de Montpellier II - Montpellier (France)
{huchard, arevalo, perrot}@lirmm.fr
[2] LIMOS - CNRS UMR 6158 - Univ. Blaise Pascal - Clermont-Ferrand II (France)
{berry, sigayret}@isima.fr
[3] LIFIA - Facultad de Informática (UNLP) - La Plata (Argentina)
garevalo@sol.info.unlp.edu.ar

Abstract. The Galois Sub-hierarchy (GSH) is a polynomial-size representation of a concept lattice which has been applied to several fields, such as software engineering and linguistics.

In this paper, we analyze the performances, in terms of computation time, of three GSH-building algorithms with very different algorithmic strategies: Ares, Ceres and Pluton. We use Java and C++ as implementation languages and Galicia as our development platform.

Our results show that implementations in C++ are significantly faster, and that in most cases Pluton is the best algorithm.

Keywords: Galois Sub-hierarchy, AOC-Poset, Performance Analysis.

1 Introduction

Formal concept analysis (FCA) has been used in a broad spectrum of research fields, such as knowledge representation, data mining, machine learning, software engineering and databases. The main drawback of concept lattices is that the number of concepts may be of much larger size than the relation (or even exponential in the size of the relation). It is therefore feasible, when this problem is encountered, to use a polynomial-size representation of the lattice while preserving the most relevant information.

One of the approaches, which has proved useful in practice, is to restrict the lattice to the concepts which introduce a new object or a new property. This idea is the basis for two very similar structures called the *Galois Sub-hierarchy (GSH)* and the *Attribute Object Concept poset (AOC-poset)*. The Galois Sub-hierarchy has been introduced in the software engineering field by Godin et al. [GM93] for class hierarchy reconstruction and successfully applied in later research work [GMM+98], [AYLCB96], [HDL00], [DHL+02]. The AOC-poset has been used in applications of FCA to non-monotonic reasoning and domain theory [Hit04] and to produce classifications from linguistic data [OP02], [Pet01].

S.O. Kuznetsov and S. Schmidt (Eds.): ICFCA 2007, LNAI 4390, pp. 166–180, 2007.

These structures are interesting not only as a feasible alternative to oversized concept lattices, but also as a conceptual improvement, as human perception of a problem is enhanced by an easy visualization of a restricted number of elements.

As the size of the input may still be large, naturally it is important to have efficient Galois Sub-hierarchy-building algorithms to work with. There are several efficient Galois Sub-hierarchy-building algorithms, with very different algorithmic strategies, and with theoretical worst-time complexity analyses which are difficult to compare. Kuznetsov et al. [KO02] propose a rather extensive implementative comparative analysis of lattice-building algorithms, but to our knowledge the only existing work on comparing algorithms related to GSH-building algorithms is proposed by Godin et al. [GC99], comparing Ares and ISGOOD, which is restrictive, as it builds only the attribute elements of the Galois Sub-hierarchy.

In this paper we address the issue of comparing the execution times of the three main Galois Sub-hierarchy-building algorithms: Ares, Ceres and Pluton, in order to determine which algorithm can be recommended to a user and in which case. This choice is meaningful because these three algorithms are used as tools with a strong user-based interaction, where the response time is a very important factor. The performance factors we tested are the density of the relation and the number of objects and attributes.

The paper is structured as follows: Section 2 introduces the main terminology of Galois Sub-hierarchy. Section 3 explains how the three Galois Sub-hierarchy-building algorithms work. Section 4 details the experimental approach which we used. Section 5 presents our evaluation of the results. We conclude in Section 6.

2 Notations and Definitions

In this section, we introduce the main terminology necessary to understanding how the Galois Sub-hierarchy algorithms work. We do not explain in detail the basics of FCA features but focus more on Galois Sub-hierarchy definitions. We refer the reader to Ganter et al. [GW99] for a complete introduction to partial orders and lattices.

In FCA, a formal context is a triple $\mathbb{K} = (G, M, I)$ where G and M are sets (objects and attributes respectively) and I is a binary relation, i.e., $I \subseteq G \times M$. Figure 1(left) shows context $\mathbb{K} = (\{1, 2, 3, 4, 5, 6\}, \{a, b, c, d, e, f, g, h\}, I)$.

For a set $A \subseteq G$ of objects, we define $A' := \{m \in M | gIm \text{ for all } g \in A\}$ (the set of attributes common to the objects in A). Correspondingly, for a set $B \subseteq M$, we define $B' := \{g \in G | gIm \text{ for all } m \in B\}$ (the set of objects which have all attributes in B). Then, a formal concept of the context (G, M, I) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. A is called the *extent* and B the *intent* of the concept (A, B). $\underline{\mathfrak{B}}(G, M, I)$ denotes the set of all concepts of the context (G, M, I). Figure 1 (right) shows the concept lattice corresponding to our example.

The concepts $C^O = \{\gamma o = (o'', o') | o \in G\}$ are called the *object concepts* of o, and the concepts $C^A = \{\mu a = (a', a'') | a \in A\}$ are called the *attribute concepts*.