A First Comparison of GALACTIC and Paspailleur

Egor Dudyrev[1,*,†], Christophe Demko[2,†], Miguel Couceiro[3,†], Mehdi Kaytoue[4,†], Karell Bertet[2,†], Amedeo Napoli[1,†] and Sergei O. Kuznetsov[5,†]

[1]*Univ. Lorraine, CNRS, Loria, F-54000 Nancy, France*
[2]*Laboratory L3I, La Rochelle University, F-17000 La Rochelle, France*
[3]*Técnico, Univ. Lisboa, INESC-ID, 1000-029 Lisboa, Portugal*
[4]*Univ. Lyon, INSA Lyon, CNRS, LIRIS UMR 5205, F-69621 Lyon, France*
[5]*HSE University, 20 Myasnitskaya St, Moscow, 101000, Russian Federation*

Abstract

This short paper presents a comparative analysis of GALACTIC and Paspailleur, both systems providing two different views about the implementation of the pattern structure formalism. The two approaches are detailed, where GALACTIC is based on so-called "predicate-based patterns" while Paspailleur relies on "atomic patterns". A primary comparison discusses (i) the formal definition of the two approaches and their positioning with respect to plain FCA, (ii) the way of exploring the concept lattice, and (iii) the way of generating new patterns from former ones. Finally two tables summarises this practical and primary comparison.

Keywords

Pattern Structures, Formal Concept Analysis, Implementation, Pattern Mining

1. Introduction

Formal concept analysis (FCA) was originally introduced as a mathematical formalism interesting for its specific properties but also for its capabilities in analysing data such as binary tables and questionnaires, and in producing a concept lattice for visualising rather small data. There were also philosophical considerations about the way concepts could be interpreted with respect to the reality related to the considered data tables or contexts.

From a more practical point of view, there were also discussions about the ways of extending the application of FCA to non-binary contexts, giving birth in particular to conceptual scaling [1]. Besides scaling, one of the first systematic and complete study about the crucial question of extension to complex data was provided in the seminal paper [2] introducing "pattern structures". Moreover, in the same years, there were also some other noticeable attempts for dealing with more complex data, including Logical Concept Analysis [3] and Fuzzy FCA [4]. Actually, the need for dealing with non-binary contexts was mainly guided by applications, trying to push FCA out of the binary case and also out of the theoretical aspects.

In the early 2000's, pattern structures were popularised in various studies more or less related to data mining and knowledge discovery needs. The objectives were to deal with complex data such as numbers, graphs, and then with natural language processing, information retrieval, linked data and ontologies in connection with relational concept analysis, functional dependencies...

In this paper, we propose a first comparison about two main and different approaches materialising the ideas on pattern structures, namely, predicate-based approach in GALACTIC and atomic patterns in Paspailleur. The comparison of both approaches is based on important directions such as the way of

The Third International Joint Conference on Conceptual Knowledge Structures (CONCEPTS), August 31 – September 4, 2026, Montpellier, France

*Corresponding author.
†These authors contributed equally.
egor.dudyrev@loria.fr (E. Dudyrev)
0000-0002-2144-3308 (E. Dudyrev); 0000-0003-0740-8466 (C. Demko); 0000-0003-2316-7623 (M. Couceiro); 0000-0002-1569-5242 (M. Kaytoue); 0000-0002-9741-4570 (K. Bertet); 0000-0001-5236-9561 (A. Napoli); 0000-0003-3284-9001 (S. O. Kuznetsov)

exploring the search space and the construction of new patterns. We study similarities and differences for understanding the capabilities of both systems dealing with applications involving complex data. Finally we analyse some ideas about pattern structures, that can guide our understanding and our usage of pattern structures in particular settings. Besides a short review about applications based on pattern structures in [5], the present study brings a brief but new point of view about theoretical and practical ways of considering and using pattern structures.

2. The On-Going Work on Pattern Structures

2.1. GALACTIC and NextPriorityConcept algorithm

GALACTIC introduces *generic descriptions of data using predicates*, adapted to different data types, either numerical, binary, categorical, or more complex. Predicates may be expressions such as "is less/greater than x" for numerical data and "match the subsequence x" for sequences. Such a predicate is applied to a data instance –or an object in a data table– and returns true of false depending on the pair object-attribute value that is considered. Since a concept covers a set of instances, its intent consists of the predicates, determined by the data type, that return true for these instances. This enables the "fusion of predicates" applying to various data types and therefore the processing of heterogeneous data. In addition, predicates are computed on the fly to each concept, starting from the top concept.

GALACTIC also introduces *generic exploration strategies*, where a strategy is based on "selectors" related to predicates. Then a selector is applied to a set of objects and returns true for a subset of these objects, thus identifying candidate instances of a subconcept. For example, the "mean strategy" is based on selectors such as "is greater/less than mean" splitting a numerical extent into two subsets. Several strategies are available, the naive one considering all possible object subsets while more specific strategies may be designed to reduce the number of subsets. Moreover, "filters" or "meta-strategies" can then be applied to concepts based on constraints such as, e.g., number of object subgroups, concept frequency w.r.t. a given threshold, or entropy-based measures for supervised data.

The NextPriorityConcept algorithm [6] computes a concept lattice from complex and heterogeneous data, using user-defined descriptions, strategies and meta-strategies. It is a top-down algorithm, that starts from an initial concept containing all data and the related predicates obtained from their descriptions based on the data type. The next-level concepts are generated using strategies and may be filtered via meta-strategies. Descriptive predicates are computed for each specific object subgroup and the process is repeated. Maintaining a Galois connection between the data and the descriptive predicates ensures that a lattice is obtained. In the predicate-based formalism [7], the descriptive predicates related to a concept are defined as the boundaries of the generalised convex hull induced by the instances in its extent, thereby preserving this connection. From a practical point of view, the GALACTIC platform[1] is based on a core containing a generic implementation of the NextPriorityConcept algorithm, complemented by a plugin mechanism making easy the integration of new features working on descriptions, predicates, and strategies, independently of the core. Presently it provides descriptions and strategies for several datatypes such as numbers, nominals, strings, sequential and temporal data [8].

2.2. Paspailleur

Paspailleur [9] is a Python package for pattern mining with Pattern Structures whose development started in 2023 in the team Orpailleur ("gold-miner" in French) in Nancy, France. The package is primarily designed for treating numerical and categorical data –popular in tabular supervised machine learning– and finding frequent patterns, as well as implication bases of frequent patterns. This predisposition for frequent patterns suggested the adaptation of the traditional meet-based bottom-up approach for working with Pattern Structures [2] for a new top-down approach. Eventually, the design and the implementation of Paspailleur lead to the creation of the framework of Atomic Patterns [10].

[1] https://thegalactic.org

Table 1
Software similarities between GALACTIC and Paspailleur.

	Associated approach	Current version	Mines	Supported data types
GALACTIC	Predicate patterns [6]	v 0.4	Concepts, Implications, Subgroups	Numerical, Category Sets, Item Sets, Sequence Sets, Polygons, Heterogeneous by Cartesian product
Paspailleur	Atomic patterns [10]	v 0.1	Concepts, Implications, Subgroups	Numerical, Category Sets, Item Sets, Sequence Sets, Heterogeneous by Cartesian product

Briefly, the idea of atomic patterns takes a set of patterns defined as a meet-semilattice $(\mathbb{D}, \sqcap)$ in [2], and completes it with join operation $\sqcup$ resulting in a pattern lattice $(\mathbb{D}, \sqcap, \sqcup)$. Here, meet operation $\sqcap$ takes two patterns and outputs their maximal common subpattern, i.e., a pattern which is less precise and thus more frequent, and join operation $\sqcup$ takes two patterns and outputs their minimal common superpattern, i.e., a pattern that is more precise and thus less frequent. The availability of join operation $\sqcup$ in the lattice $(\mathbb{D}, \sqcap, \sqcup)$ implies that some elements of the lattice can be represented as a join of some other elements, while this cannot be the case for some others, i.e., the join-irreducible elements. We call these join-irreducible elements of the lattice of patterns the *atomic patterns*. We like to imagine every pattern as "a molecule" that, although an important substance on its own, can be split into a set of smaller "atoms". More formally, let $(\mathbb{D}, \sqcap, \sqcup)$ be a lattice of patterns, then the set of atomic patterns $\mathbb{A}$ is constituted as follows:

$$\begin{aligned} \mathbb{A} &= \{d \in \mathbb{D} \mid d \text{ is } \sqcup\text{-irreducible}\} \\ &= \{d \in \mathbb{D} \mid \forall d_1, d_2 \in \mathbb{D}, (d_1 \sqcup d_2 = d) \implies (d_1 = d \text{ or } d_2 = d\}. \end{aligned} \tag{1}$$

It was shown in [10] that atomic patterns $\mathbb{A}$ are similar to attributes M of a formal context (G, M, I). Indeed, in the formal context, the description space is the Boolean lattice $(2^M, \cap, \cup)$, where every attribute m corresponds to a join-irreducible description $\{m\}$. What distinguishes the set of attributes M and the set of atomic patterns $\mathbb{A}$ is that the set of atomic patterns can be ordered with respect to the pattern order defined as $\sqsubseteq_{\mathbb{A}}$: $\forall a, b \in \mathbb{A}, a \sqsubseteq_{\mathbb{A}} b \iff a \sqcup b = b$.

In one hand, pattern structures deal with pattern lattice $(\mathbb{D}, \sqsubseteq)$ with relatively small set $\mathbb{D}$ equipped with potentially intractable operation $\sqsubseteq$. In the other hand, Formal Concept Analysis treats the Boolean lattice of descriptions $(2^M, \subseteq)$ with massive set 2^M equipped with computationally efficient subset operation $\subseteq$. Then the atomic pattern approach takes the best of the two worlds and relies on the lattice of order ideals of atomic patterns $(\mathcal{I}(\mathbb{A}, \sqsubseteq_{\mathbb{A}}), \subseteq)$. From the theoretical standpoint, this suggests replacing the lattice of patterns with its down-set Birkhoff completion [11]. Overall, atomic pattern approach makes Pattern Structures look similar to the classical Formal Concept Analysis, but more computationally efficient.

3. Comparison and perspectives

The two software solutions for Pattern Structures, GALACTIC[2] and Paspailleur[3], were developed by two distinct research groups and thus resulted in two different approaches: one uses predicates and exploration strategies while the other relies on atomic patterns. The two approaches started with the same intention to apply Pattern Structure formalism to the analysis of large heterogeneous tabular data. Interestingly, they reached similar objectives, both allowing to mine pattern concepts, implications, and subgroups for similar data types such as numerical and categorical data. One can observe the similarities and minor differences between GALACTIC and Paspailleur from software viewpoint in

[2]https://www.thegalactic.org/
[3]https://smartfca.github.io/paspailleur/

Table 2
Exploration differences between Predicate and Atomic patterns.

	Concept exploration direction	Attribute generation
Formal Concept Analysis [1]	Top-down and Bottom-up	Predefined
Original Pattern Structures [2]	Bottom-up	Dynamic
Predicate patterns [6]	Top-down	Dynamic
Atomic Patterns [10]	Top-down and Bottom-up	Predefined

Table 1. However, a careful look at the theoretical aspects of Predicate and Atomic patterns reveals two important differences in how they navigate the set of patterns.

3.1. The direction of concept exploration

The first difference is concerned with the direction of concept exploration. Recall that in FCA one can explore the concept lattice both in "top-down" and in "bottom-up" direction. Then, if needed, one can start with the empty set of attributes that selects the most frequent (so, the top) concept and gradually discover more and more precise concepts by adding more and more attributes, until reaching the least frequent (so, bottom) concept with the maximal intent. Dually, one can start with the set of all attributes that selects the least frequent (so, the bottom) concept and gradually discover more and more frequent concepts by removing more and more attributes, until reaching the most frequent (so, the top) concept. The first direction suits frequent concept mining, while the latter might work better for mining rare concepts. The original publication on Pattern Structures [2] describes the set of patterns as a meet-semilattice where the meet operation on patterns generalises the intersection operation on formal intents. Thus, the original Pattern Structures formulation does not suit well the task of frequent pattern mining: one has to meet too many rare patterns before arriving at the frequent ones.

Predicate patterns, as implemented in GALACTIC, solve this problem by focusing on top-down exploration, starting with an empty set of predicates and gradually discovering more precise predicates and more precise pattern concepts. Atomic patterns, as implemented in Paspailleur, solve this problem too by reducing a pattern structure to a formal context with a partially ordered set of attributes called atomic patterns. In this way, atomic pattern approach supports both top-down and bottom-up exploration of concepts. This makes atomic pattern approach more general than the predicate pattern approach.

3.2. Attribute generation

A second difference is in concern with the generation of attributes and patterns. Recall that in FCA it is assumed that the set of attributes M and the corresponding lattice $(2^M, \subseteq)$ is fixed before the analysis. However, when facing large complex heterogeneous data, such set of attributes may contain tens of thousands of elements, many of which might not be relevant to the analysis. The original Pattern Structures solve the problem by starting with a set of object descriptions. During the analysis, one selects a subset of objects, computes the meet of these object descriptions and discovers a new pattern. Predicate patterns as implemented in GALACTIC follow a similar approach, starting with the set of predicates describing the whole dataset and covering all the objects. Then, during the analysis, the use of strategies guides the selection of a subset of objects, allowing the system to compute a set of descriptive predicates for this subset and thus to discover a new pattern.

By contrast, Atomic pattern approach as implemented in Paspailleur breaks this tradition. It requires the generation of all atomic patterns before the start of the analysis. Then, during the analysis, it selects a subset of atomic patterns, computes their join and discovers a new pattern. Overall, both FCA and Atomic patterns approach require a predefined fixed set of attributes and atomic patterns, while both the original Pattern Structures and Predicate patterns can start with "virtually nothing", i.e., an empty set of patterns, and generate new patterns on-the-fly. This makes predicate patterns approach more general than the atomic pattern approach.

3.3. Perspectives

We highlight the stated differences in Table 2. One can check that there is no clear "winner" between the two approaches. Predicate patterns approach offers dynamic attribute –or, better to say, predicate– generation that makes it more suitable for interactive analysis of big data. Atomic pattern approach is closer to FCA as it supports both top-down and bottom-up exploration of concepts.

The theoretical differences in Table 2 combined with practical similarities in Table 1 give hope for a successful cross-pollination of two approaches. Indeed, the framework of Pattern Structures, although proposed in 2001, is still in the process of active development but not yet materialised in a book similar to the seminal book of Formal Concept Analysis [1]). It is possible that, in the future, the atomic pattern approach could support dynamic generation of atomic patterns, while predicate patterns could support a bottom-up concept exploration. A deeper analysis and comparison of the two approaches should still be provided.

Acknowledgments

All authors but the last one are carrying out this research work as part of the French ANR-21-CE23-0023 SmartFCA Research Project. The work by S.O. Kuznetsov in preparing this article was supported by the Basic Research Program of HSE University, Moscow.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] B. Ganter, R. Wille, Formal Concept Analysis, Mathematical foundations, Springer Verlag, Berlin, 1999.

[2] B. Ganter, S. O. Kuznetsov, Pattern Structures and Their Projections, in: H. S. Delugach, G. Stumme (Eds.), Proceedings of 9th ICCS, Lecture Notes in Computer Science 2120, Springer, 2001, pp. 129–142.

[3] S. Ferré, O. Ridoux, A Logical Generalization of Formal Concept Analysis, in: B. Ganter, G. W. Mineau (Eds.), Proceeedings of ICCS, Lecture Notes in Computer Science 1867, Springer, 2000, pp. 371–384.

[4] R. Belohlávek, V. Vychodil, What is a fuzzy concept lattice?, in: Proceedings of CLA, volume 162 of *CEUR Workshop Proceedings 162*, 2005.

[5] M. Kaytoue, V. Codocedo, A. Buzmakov, J. Baixeries, S. O. Kuznetsov, A. Napoli, Pattern Structures and Concept Lattices for Data Mining and Knowledge Processing, in: Proceedings of ECML-PKDD, Lecture Notes in Computer Science 9286, Springer, 2015, pp. 227–231.

[6] Ch. Demko, K. Bertet, C. Faucher, J.-F. Viaud, S. O. Kuznetsov, NextPriorityConcept: A New and Generic Algorithm Computing Concepts from Complex and Heterogeneous Data, Theoretical Computer Science 845 (2020) 1–20.

[7] Ch. Demko, K. Bertet, J.-F. Viaud, C. Faucher, D. Mondou, Description lattices of generalised convex hulls, International Journal of Approximate Reasoning (2024) 109269.

[8] S. E. Boukhetta, C. Demko, K. Bertet, J. Richard, C. Cayèré, Temporal sequence mining using FCA and GALACTIC, in: T. Braun, M. Gehrke, T. Hanika, N. Hernandez (Eds.), Proceedings of ICCS, volume 12879, Springer, 2021, pp. 185–199.

[9] E. Dudyrev, Paspailleur, 2025. URL: https://smartfca.github.io/paspailleur. doi:10.5281/zenodo.15639146.

[10] E. Dudyrev, M. Couceiro, M. Kaytoue, S. O. Kuznetsov, A. Napoli, Atomic Patterns for Efficient Computation with Pattern Structures, in: Proceedings of CONCEPTS, Lecture Notes in Computer Science 15941, Springer, 2025, pp. 178–194.

[11] M. Abdulla, J. Hirth, G. Stumme, The birkhoff completion of finite lattices, in: I. P. Cabrera, S. Ferré, S. Obiedkov (Eds.), Conceptual Knowledge Structures, Springer Nature Switzerland, Cham, 2024, pp. 20–35.