# On Projections of Sequential Pattern Structures (with an application on care trajectories)

Aleksey Buzmakov[1,2], Elias Egho[1], Nicolas Jay[1], Sergei O. Kuznetsov[2], Amedeo Napoli[1], and Chedy Raïssi[1]

[1] LORIA (CNRS – Inria NGE – U. de Lorraine), Vandœuvre-lès-Nancy, France
[2] National Research University Higher School of Economics, Moscow, Russia
{aleksey.buzmakov, chedy.raissi}@inria.fr,
{elias.egho, nicolas.jay, amedeo.napoli}@loria.fr, skuznetsov@hse.ru

**Abstract.** In this paper, we are interested in the analysis of sequential data and we propose an original framework based on FCA. For that, we introduce sequential pattern structures, an original specification of pattern structures for dealing with sequential data. Sequential pattern structures are given by a subsumption operation between set of sequences, based on subsequence matching. To avoid a huge number of resulting concepts, domain knowledge projections can be applied. The original definition of projections is revised in order to operate on sequential pattern structures in a meaningful way. Based on the introduced definition, several projections of sequential pattern structures involving domain or expert knowledge are defined and discussed. This projections are evaluated on a real dataset on care trajectories where every hospitalization is described by a heterogeneous tuple with different fields. The evaluation reveals interesting concepts and justify the usage of introduced projections of sequential pattern structures. This research work provides a new and efficient extension of FCA to deal with complex data, which can be an alternative to the analysis of sequential datasets.

**Keywords:** formal concept analysis, pattern structures, projections, sequential pattern structures, sequences

## Introduction

Analysis of sequential data is a challenging task. In the last two decades, the main emphasis has been on developing efficient mining algorithms with effective pattern representations for sequences of itemsets [1–4]. The traditional sequential pattern mining algorithms generate a large number of frequent sequences while a few of them are truly relevant. Moreover, in some particular cases, only sequential patterns of a certain type are of interest and should be mined first. *Are we able to develop a framework for taking into account only patterns of required types?* Furthermore, in many cases sequential data are described by sequences with complex elements, e.g. a text is a sequence of syntactic trees. To process such kind of data with existing algorithms, elements of sequences can be scaled

 Published in Manuel Ojeda-Aciego, Jan Outrata (Eds.): CLA 2013, pp. 199–210, ISBN 978–2–7466–6566–8, Laboratory L3i, University of La Rochelle, 2013. 

into itemsets as it is done in the case of multilevel multidimensional data [5]. However, in this case it is rather difficult to introduce expert requirements within a sequence, which leads to even a larger set of resulting patterns.

We approach this problem with FCA and pattern structures [6, 7]. FCA is successfully used for analysis of sequential data [8, 9]. Moreover, it allows one to use different measures of interestingness for the resulting patterns (concepts). Pattern structures allows to directly process sequential data without a scaling step. Furthermore, there are projections of pattern structures, which were introduced in order to simplify the computation of pattern lattices, by simplifying descriptions. Moreover, projections can be efficiently used as special domain knowledge requirements, allowing to reduce the number of irrelevant patterns. We generalize the original definitions of projections, in order to deal with projections respecting domain knowledge. For example, sequences of length 1 are rare useful but they cannot be excluded by the original definition of projections.

The rest of the paper is organized as follows. In Section 1 we remind FCA, pattern structures and measures of concept interestingness. Section 2 states the problem of complex sequences analysis and introduces sequential pattern structures. In Section 3, first, the generalization of projections is defined, and, second, some projections specific to sequential pattern structures are introduced and analyzed. And finally before concluding the paper, we discuss an experimental evaluation in Section 4.

## 1 FCA and Pattern Structures

FCA [6] is a mathematical formalism having many applications in data analysis. Pattern structures is a generalization of FCA for dealing with complex structures, such as sequences or graphs [7].

**Definition 1.** *A pattern structure is a triple* $(G, (D, \sqcap), \delta)$*, where* $G$ *is a set of objects,* $(D, \sqcap)$ *is a complete meet-semilattice of descriptions and* $\delta : G \to D$ *maps an object to a description.*

The lattice operation in the semilattice ($\sqcap$) corresponds to the similarity between two descriptions. Standard FCA can be presented in terms of pattern structures. In this case, $G$ is the set of objects, the semilattice of descriptions is $(\wp(M), \sqcap)$, where a description is a set of attributes, with the $\sqcap$ operation corresponding to the set intersection ($\wp(M)$ denotes the powerset of $M$). If $x = \{a, b, c\}$ and $y = \{a, c, d\}$ then $x \sqcap y = x \cap y = \{a, c\}$. The mapping $\delta : G \to \wp(M)$ is given by, $\delta(g) = \{m \in M \mid (g, m) \in I\}$, and returns the description for a given object as a set of attributes.

The Galois connection for $(G, (D, \sqcap), \delta)$ is defined as follows:

$$A^\diamond := \prod_{g \in A} \delta(g), \qquad \text{for } A \subseteq G$$

$$d^\diamond := \{g \in G \mid d \sqsubseteq \delta(g)\}, \qquad \text{for } d \in D$$

The Galois connection makes a correspondence between sets of objects and descriptions. Given a set of objects $A$, $A^\diamond$ returns the description which is common to all objects in $A$. And given a description $d$, $d^\diamond$ is the set of all objects whose description subsumes $d$. More precisely, the partial order (or the subsumption order) on $D$ ($\sqsubseteq$) is defined w.r.t. the similarity operation $\sqcap$: $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$, and $c$ is subsumed by $d$.

**Definition 2.** *A pattern concept of a pattern structure* $(G, (D, \sqcap), \delta)$ *is a pair* $(A, d)$ *where* $A \subseteq G$ *and* $d \in D$ *such that* $A^\diamond = d$ *and* $d^\diamond = A$, $A$ *is called the concept extent and* $d$ *is called the concept intent.*

As in standard FCA, a pattern concept corresponds to the maximal set of objects $A$ whose description subsumes the description $d$, where $d$ is the maximal common description for objects in $A$. The set of all concepts can be partially ordered w.r.t. partial order on extents (dually, intent patterns, i.e $\sqsubseteq$), within a concept lattice. An example of a pattern structure is given and described in the next sections. It can be noticed that Table 1 defines a pattern structure, while the corresponding lattice is depicted in Figure 1.

It is worth mentioning, that the size of the concept lattice can be exponential w.r.t. to the number of objects, and, thus, we need a special ranking method to select the most interesting concepts for further analysis. Several techniques are considered in [10], where it is shown that stability index [11] is more reliable in noisy data. Thus, we use this index in our current work.

**Definition 3.** *Given a concept* $c$*, the concept stability* $Stab(c)$ *is the percent of subsets of the concept extent (denoted* $Ext(c)$*), whose description is equal to the concept intent (denoted* $Int(c)$*).*

$$Stab(c) := \frac{|\{s \in \wp(Ext(c)) \mid s^\diamond = Int(c)\}|}{|\wp(Ext(c))|} \tag{1}$$

Stability measures how much the concept depends on the initial dataset. The larger the stability the more objects can be deleted from the context without affecting the intent of the concept, i.e. the intent of the most stable concepts are likely to be a characteristic pattern of a studied phenomena rather than an artifact of a data set.

After a brief general description of the analysis with pattern structures, the analysis of sequential data can be specified.

## 2 Sequential Pattern Structures

### 2.1 An Example of Sequential Data

Imagine that we have medical trajectories of patients, i.e. sequences of hospitalizations, where every hospitalization is described by a hospital name and a set of procedures. An example of sequential data on medical trajectories with three patients is given in Table 1. There are a set of procedures $P = \{a, b, c, d\}$, a

| Patient | Trajectory |
|---|---|
| $p^1$ | $\langle[H_1,\{a\}];[H_1,\{c,d\}];[H_1,\{a,b\}];[H_1,\{d\}]\rangle$ |
| $p^2$ | $\langle[H_2,\{c,d\}];[H_3,\{b,d\}];[H_3,\{a,d\}]\rangle$ |
| $p^3$ | $\langle[H_4,\{c,d\}];[H_4,\{b\}];[H_4,\{a\}];[H_4,\{a,d\}]\rangle$ |

Table 1: Toy sequential data on patient medical trajectories.

set of hospital names $T_H = \{H_1, H_2, H_3, H_4, CL, CH, *\}$, where hospital names are hierarchically organized (by the level of generality), $H_1$ and $H_2$ are central hospitals ($CH$) and $H_3$ and $H_4$ are clinics ($CL$), and $*$ denotes the root of this hierarchy. For the sake of simplicity, we use the $\sqcap$ operator in order to denote the least common ancestor in $T_H$, i.e. $H_1 \sqcap H_2 = CH$. Every hospitalization is described with one hospital name and may contain several procedures. The procedure order in each hospitalization is not important in our case. For example, the first hospitalization $[H_2, \{c, d\}]$ for the second patient ($p^2$) was in hospital $H_2$ and during this hospitalization the patient underwent procedures $c$ and $d$. An important task is to find the "characteristic" sequences of procedures and associated hospitals in order to improve hospitalization planning, optimize clinical processes or detect anomalies. This sequences can be found by searching for the most stable concepts in the lattice corresponding to a pattern structure.

### 2.2 Partial Order on Complex Sequences

A sequence is constituted of elements from an alphabet. The classical subsequence matching task requires no special properties of the alphabet. Several generalizations of the classical case were made by introducing a subsequence relation based on itemset alphabet [8] or on multidimensional and multilevel alphabet [5], scaled to itemset alphabet as well. Both these alphabets are certain semilattices, and, thus, we generalize the previous cases, requiring for an alphabet to form a general semilattice $(E, \sqcap_E)$[1]. Thanks to pattern structure formalism we are able to process in a unified way all types of sequential datasets with poset-shaped alphabet. However, some sequential data can have connections between elements, e.g. [12], and, thus, cannot be immediately processed by our approach.

**Definition 4.** *A sequence is an ordered list of $e \in (E, \sqcap_E)$, such that $e \neq \bot_E$.*

Here, $\forall e \in E, \bot_E = \bot_E \sqcap_E e$. The bottom element is required by the lattice structure but provide us with no useful information (it matches to any other element), thus, it is excluded from the sequences. In the same way, in mining of sequences of itemsets the empty itemset cannot be a proper element [2].

**Definition 5.** *A sequence $t = \langle t_1; ...; t_k\rangle$ is a subsequence of a sequence $s = \langle s_1; ...; s_n\rangle$, denoted $t \leq s$, iff $k \leq n$ and there exist $j_1, ..j_k$ such that $1 \leq j_1 < j_2 < ... < j_k \leq n$ and for all $i \in \{1, 2, ..., k\}$, $t_i \sqsubseteq_E s_{j_i}$ ($\Leftrightarrow t_i \sqcap_E s_{j_i} = t_i$).*

[1] In this paper we consider two semilattices, the first one is related to the characters of the alphabet, $(E, \sqcap_E)$, and the second one is related to pattern structures, $(D, \sqcap)$.

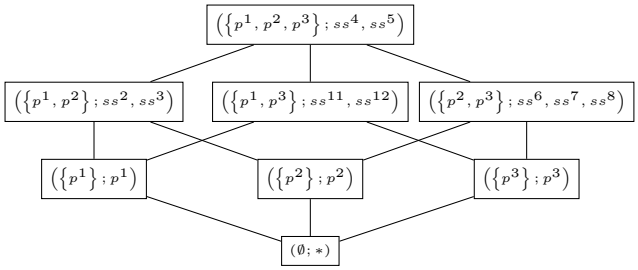



Fig. 1: The concept lattice for the pattern structure given by Table 1. Concept intents reference to sequences in Tables 1 and 2.

| | Subsequences |
|---|---|
| $ss^1$ | $\langle [CH, \{c, d\}]; [H_1, \{b\}]; [*, \{d\}] \rangle$ |
| $ss^3$ | $\langle [CH, \{\}]; [*, \{d\}]; [*, \{a\}] \rangle$ |
| $ss^5$ | $\langle [*, \{a\}] \rangle$ |
| $ss^7$ | $\langle [CL, \{d\}]; [CL, \{\}] \rangle$ |
| $ss^9$ | $\langle [CH, \{c, d\}] \rangle$ |
| $ss^{11}$ | $\langle [*, \{c, d\}]; [*, \{b\}] \rangle$ |

| | Subsequences |
|---|---|
| $ss^2$ | $\langle [CH, \{c, d\}]; [*, \{b\}]; [*, \{d\}] \rangle$ |
| $ss^4$ | $\langle [*, \{c, d\}]; [*, \{b\}] \rangle$ |
| $ss^6$ | $\langle [*, \{c, d\}]; [CL, \{b\}]; [CL, \{a\}] \rangle$ |
| $ss^8$ | $\langle [CL, \{\}]; [CL, \{a, d\}] \rangle$ |
| $ss^{10}$ | $\langle [CL, \{b\}]; [CL, \{a\}] \rangle$ |
| $ss^{12}$ | $\langle [*, \{a\}]; [*, \{d\}] \rangle$ |

Table 2: Subsequences of patient sequences in Table 1.

With complex sequences and such kind of subsequences the computational procedure can be difficult, thus, to simplify the procedure, only "contiguous" subsequences are considered, where only the order of consequent elements is taken into account, i.e. given $j_1$ in Definition 5, $j_i = j_{i-1}+1$ for all $i \in \{2, 3, ..., k\}$. Such a restriction makes sens for our data, because a hospitalization is a discrete event and it is likely that the next hospitalization has a relation with the previous one, for example, hospitalizations for treating aftereffects of chemotherapy. Below the word "subsequence" refers to "contiguous" subsequence.

*Example 1.* In the running example (Section 2.1), the alphabet is $E = T_H \times \wp(P)$ with the similarity operation $(h_1, P_1) \sqcap (h_2, P_2) = (h_1 \sqcap h_2, P_1 \cap P_2)$, where $h_1, h_2 \in T_H$ are hospitals and $P_1, P_2 \in \wp(P)$ are sets of procedures. Thus, the sequence $ss^1$ in Table 2 is a subsequence of $p^1$ in Table 1 because if we set $j_i = i + 1$ (Definition 5) then $ss^1_1 \sqsubseteq p^1_{j_1}$ ('CH' is an ancestor for $H_1$ and $\{c, d\} \subseteq \{c, d\}$), $ss^1_2 \sqsubseteq p^1_{j_2}$ (the same hospital and $\{b\} \subseteq \{b, a\}$) and $ss^1_3 \sqsubseteq p^1_{j_3}$ ('*' is an ancestor for anything and $\{d\} \subseteq \{d\}$).

### 2.3 Meet-semilattice of Sequences

Using the previous definitions, we can precisely define the sequential pattern structures that are used for representing and managing sequences. For that, we make an analogy with pattern structures for graphs where the meet-semilattice operation $\sqcap$ respects subgraph isomorphism [13]. Thus, we introduce a sequential meet-semilattice respecting subsequence relation. Let us consider $\mathfrak{S}$ as the set of all sequences based on an alphabet $(E, \sqcap_E)$. $\mathfrak{S}$ is partially ordered w.r.t. Definition 5. $(D, \sqcap)$ is a semilattice on sequences $\mathfrak{S}$, where $D \subseteq \wp(\mathfrak{S})$ such that

if $d \in D$ contains a sequence $s$ then all subsequences of $s$ should be included into $d$, $\forall s \in d, \nexists \tilde{s} \leq s : \tilde{s} \notin d$, and the similarity operation is the set intersection for two sets of sequences. Given two patterns $d_1, d_2 \in D$, the set intersection operation ensures that if a sequence $s$ belongs to $d_1 \sqcap d_2$ then any subsequence of $s$ belongs to $d_1 \sqcap d_2$ and thus $d_1 \sqcap d_2 \in D$. As the set intersection operation is idempotent, commutative and associative, $(D, \sqcap)$ is a valid semilattice.

*Example 2.* The sequential pattern structure for our example (Subsection 2.1) is $(G, (D, \sqcap), \delta)$, where $G = \{p^1, p^2, p^3\}$ is the set of patients, $(D, \sqcap)$ is the semilattice of sequential descriptions, and $\delta$ is the mapping (shown in Table 1) associating a patient in $G$ to a description in $D$. Figure 1 shows the resulting lattice of sequential pattern concepts for this particular pattern structure.

The set of all possible subsequences for a given sequence can be rather large. Thus, it is more efficient and readable to keep a pattern $d \in D$ as a set of only maximal sequences $\tilde{d}$, $\tilde{d} = \{s \in d \mid \nexists s^* \in d : s^* \geq s\}$. In the rest of the paper, every pattern is given only by the set of its maximal sequences. For example, $\{p^2\} \sqcap \{p^3\} = \{ss^6, ss^7, ss^8\}$ (see Tables 1 and 2), i.e. $\{ss^6, ss^7, ss^8\}$ is the set of all maximal sequences specifying the intersection result of two sets of sequences $\{p^2\}$ and $\{p^3\}$, in the same way $\{ss^6, ss^7, ss^8\} \sqcap \{p^1\} = \{ss^4, ss^5\}$. Note that representing a pattern by the set of all maximal sequences allows for an efficient implementation of the intersection "$\sqcap$" of two patterns. The next proposition is follows from this subsection and Definition 5.

**Proposition 1.** *Given $(G, (D, \sqcap), \delta)$ and $x, y \in D$, $x \sqsubseteq y$ if and only if $\forall s^x \in x$ there is a sequence $s^y \in y$, such that $s^x \leq s^y$.*

## 3 Projections of Sequential Pattern Structures

Pattern structures can be hard to process due to the usually large number of concepts in the concept lattice and the complexity of the involved similarity operation (make the parallel with the graph isomorphism problem). Moreover, a given pattern structure can produce a lattice with a lot of patterns which are not interesting for an expert. *Can we save computational time by avoiding the construction of unnecessary patterns?* Projections of pattern structures "simplify" to some degree the computation and allow one to work with a reduced description. In fact, projections can be considered as constraints (or filters) on patterns respecting certain mathematical properties. These mathematical properties ensure that the projection of a lattice is a lattice where projected concepts have certain correspondence to original ones. Moreover, the stability measure of projected concepts never decreases w.r.t the corresponding concepts. We introduce projections on sequential patterns, revising them from [7]. An extended definition of projections w.r.t. the definition in [7] should be provided in order to deal with interesting projections for real-life sequential datasets.

**Definition 6.** *A projection $\psi : D \to D$ is an interior operator, i.e. it is (1) monotone ($x \sqsubseteq y \Rightarrow \psi(x) \sqsubseteq \psi(y)$), (2) contractive ($\psi(x) \sqsubseteq x$) and (3) idempotent ($\psi(\psi(x)) = \psi(x)$).*

Under a projection $\psi$, a pattern structure $(G, (D, \sqcap), \delta)$ becomes the projected pattern structure $\psi((G, (D, \sqcap), \delta)) = (G, (D_\psi, \sqcap_\psi), \psi \circ \delta)$, where $D_\psi = \psi(D) = \{d \in D \mid \exists d^* \in D : \psi(d^*) = d\}$ and $\forall x, y \in D, x \sqcap_\psi y := \psi(x \sqcap y)$. Note that in [7] $\psi((G, (D, \sqcap), \delta)) = (G, (D, \sqcap), \psi \circ \delta)$. Now we should show that $(D_\psi, \sqcap_\psi)$ is a semilattice.

**Proposition 2.** *Given a semilattice $(D, \sqcap)$ and a projection $\psi$, for all $x, y \in D$ $\psi(x \sqcap y) = \psi(\psi(x) \sqcap y)$.*

*Proof.* 1. $\psi(x) \sqsubseteq x$, thus, $x, y \sqsupseteq (x \sqcap y) \sqsupseteq (\psi(x) \sqcap y) \sqsupseteq \psi(\psi(x) \sqcap y)$
2. $x \sqsubseteq y \Rightarrow \psi(x) \sqsubseteq \psi(y)$, thus, $\psi(x \sqcap y) \sqsupseteq \psi(\psi(x) \sqcap y)$
3. $\psi(x \sqcap y) \sqcap \psi(x) \sqcap y \underset{\psi(x \sqcap y) \sqsubseteq \psi(x)}{=} \psi(x \sqcap y) \sqcap y \underset{\psi(x \sqcap y) \sqsubseteq y}{=} \psi(x \sqcap y)$,
   then $(\psi(x) \sqcap y) \sqsupseteq \psi(x \sqcap y)$ and $\psi(\psi(x) \sqcap y) \sqsupseteq \psi(\psi(x \sqcap y)) = \psi(x \sqcap y)$
4. From (2) and (3) follows that $\psi(x \sqcap y) = \psi(\psi(x) \sqcap y)$.

**Corollary 1.** $X_1 \sqcap_\psi X_2 \sqcap_\psi \cdots \sqcap_\psi X_N = \psi(X_1 \sqcap X_2 \sqcap \cdots \sqcap X_N)$

**Corollary 2.** *Given a semilattice $(D, \sqcap)$ and a projection $\psi$, $(D_\psi, \sqcap_\psi)$ is a semilattice, i.e. $\sqcap_\psi$ is commutative, associative and idempotent.*

The concepts of a pattern structure and a projected pattern structure are connected with the next proposition, following from Corollary 1:

**Proposition 3.** *An extent in $\psi((G, (D, \sqcap), \delta))$ is an extent in $(G, (D, \sqcap), \delta)$. An intent in $\psi((G, (D, \sqcap), \delta))$ is of the form $\psi(d)$, where $d$ is the intent of the concept with the same extent.*

Moreover, preserving extents of some concepts, projections cannot decrease the stability of the projected concepts, i.e. if the projection preserves a stable concept, then its stability (Definition 3) can only increase.

**Proposition 4.** *Given a pattern structure $(G, (D, \sqcap), \delta)$, its concept $c$ and a projected pattern structure $(G, (D_\psi, \sqcap_\psi), \psi \circ \delta)$, and the projected concept $\tilde{c}$, if the concept extents are equal ($Ext(c) = Ext(\tilde{c})$) then $Stab(c) \leq Stab(\tilde{c})$.*

*Proof.* Concepts $c$ and $\tilde{c}$ have the same extent. Thus, according to Definition 3, in order to prove the proposition statement, it is enough to prove that for any subset $A \subseteq Ext(c)$, if $A^\diamond = Int(c)$ in the original pattern structure, then $A^\diamond = Int(\tilde{c})$ in the projected one. It can be proven from contrary.

Suppose that $\exists A \subset Ext(c)$ such that $A^\diamond = Int(c)$ in the original pattern structure and $A^\diamond \neq Int(\tilde{c})$ in the projected one. Then there is a descendant concept $\tilde{d}$ of $\tilde{c}$ in the projected pattern structure such that $A^\diamond = Int(\tilde{d})$ in the projected lattice. Then there is an original concept $d$ for the projected concept $\tilde{d}$ with the same $Ext(d)$. Then $A^\diamond \sqsupseteq Int(d) \sqsupset Int(c)$ and, so, $A^\diamond$ cannot be equal to $Int(c)$ in the original lattice. Contradiction.

No we are going to present two projections of sequential pattern structures. The first projection comes from the following observation. In many cases it may

be more interesting to analyze quite long subsequences rather than short one. This kind of projections is called *Minimal Length Projection* (MLP) and it depends on the minimal allowed length parameter $l$ for the sequences in a pattern. The corresponding function $\psi$ maps a pattern without short sequences to itself, and a sequence with short sequences to the pattern containing only long sequences, $\psi(d) = \{s \in d \mid length(s) > l\}$. Later, propositions 1 and 5 stay that MLP is coherent with Definition 6.

*Example 3.* If we prefer common subsequences of length $\geq 3$, then between $p^2$ and $p^3$ in Table 1 there is only one maximal common subsequence, $ss^6$ in Table 2, while $ss^7$ and $ss^8$ are too short to be considered. Figure 2a shows the lattice corresponding the projected pattern structure (Table 1) with patterns of length more or equal to 3.

**Proposition 5.** *MLP is a monotone, contractive and idempotent function on the semilattice* $(D, \sqcap)$.

*Proof.* The contractivity and idempotentcy are quite clear from the definition. Remains the proof for monotonicity.

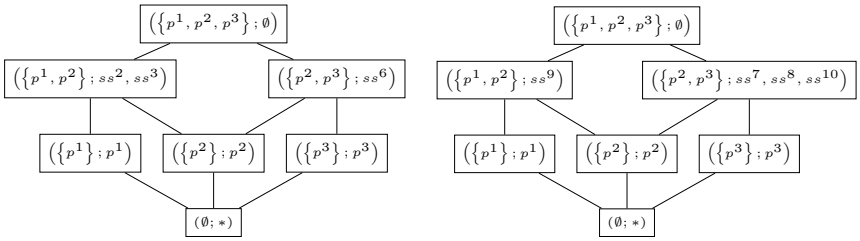
If $X \sqsubseteq Y$ where $X$ and $Y$ are sets of sequences then for every sequence $x \in X$ there is a sequence $y \in Y$ such that $x \leq y$ (Proposition 1). We should show that $\psi(X) \sqsubseteq \psi(Y)$, or in other words for every sequence $x \in \psi(X)$ there is a sequence $y \in \psi(Y)$, such that $x \leq y$. Given $x \in \psi(X)$, since $\psi(X)$ is a subset of $X$ and $X \sqsubseteq Y$, then there is a sequence $y \in Y$ such that $x \leq y$, with $|y| \geq |x| \geq l$ ($l$ is a parameter of MLP), and thus, $y \in \psi(Y)$.

The second projection of a sequential pattern structure is connected to a projection of an alphabet semilattice, $(E, \sqcap_E)$.

*Example 4.* An expert is interested in finding sequential patterns on how a patient changes hospitals, but he has little interest in procedures. Thus, any element of the alphabet lattice, containing a non-empty set of procedures can be projected to the element with the same hospital but with the empty set of procedures.

*Example 5.* An expert is interested in finding sequential patterns containing some information about the hospital in every hospitalization, and the corresponding procedures, i.e. hospital field in the patterns cannot be equal to the element "any hospital", denoted $*$, e.g., $ss^5$ is an invalid pattern, while $ss^6$ is a valid pattern in Table 2. Thus, any element of the alphabet semilattice with $*$ hospital can be projected to the $\bot_E$. Figure 2b shows the lattice corresponding to the projected pattern structure (Table 1), where projection comes from the projection of the alphabet semilattice.

Below we formally define how the alphabet projection of a sequential pattern structure should be processed. Intuitively, every sequence in a pattern should be substituted with another sequence, by applying the alphabet projection to all its elements. However, the result can be an incorrect sequence, because $\bot_E$ is forbidden to be in a sequence, thus, sequences in a pattern should be "developed" w.r.t. $\bot_E$, as it is explained below.




(a) MLP projection, $l = 3$ (b) Projection removing '*' hospitals

Fig. 2: The projected concept lattices for the pattern structure given by Table 1. Concept intents refer to the sequences in Tables 1 and 2.

**Definition 7.** *Given an alphabet $(E, \sqcap_E)$, an alphabet projection $\psi$ and a sequence $s$ based on $E$, the projection of the sequence $\psi(s)$ is the sequence $\tilde{s}$ such that, $\tilde{s}_i = \psi(s_i)$ ($s_i$ is the $i$-th element of sequence $s$).*

Here, it should be noted that $\tilde{s}$ can be incoherent with Definition 4, since it allows $\bot_E$ to be an element. For simplicity, we allow this incoherence here.

**Definition 8.** *Given an alphabet $(E, \sqcap_E)$, an alphabet projection $\psi$, and a pattern $d \in D$, an alphabet-projected pattern $\tilde{d} = \psi(d)$, is the set of sequences obtained by the following procedure. For every sequence $s \in d$, the projection of $s$ is computed (Definition 7) and, then, the projection of the sequence is substituted by the set of its maximal subsequences containing no $\bot$. All the resulting sequences constitute the set $\hat{d}$, and $\tilde{d}$ is the set of maximal sequences in $\hat{d}$.*

*Example 6.* $\{ss^6\}$ is an alphabet-projected pattern for the pattern $\{ss^{10}\}$, where alphabet lattice projection is given in Example 5.

$\{\langle [CH, \{c, d\}]\rangle\}$ is an alphabet-projected pattern for the pattern $\{ss^2\}$, where alphabet lattice projection is given by projecting every element with medical procedure $b$ to the element with the same hospital and with the same set procedures excluding $b$. The projected sequence of sequence $ss^2$ is $\langle [CH, \{c, d\}]; [*, \{\}]; [*, \{d\}]\rangle$, but $[*, \{\}] = \bot_E$, and, thus, in order to project the pattern $\{ss^2\}$ the projected sequence is substituted by its maximal subsequences, i.e.
$\psi(\{\langle [CH, \{c, d\}]; [*, \{b\}]; [*, \{d\}]\rangle\}) = \{\langle [CH, \{c, d\}]\rangle\}$.

**Proposition 6.** *Considering an alphabet $(E, \sqcap_E)$, the projection of an alphabet $\psi$, a sequential pattern structure $(G, (D, \sqcap), \delta)$, the procedure given by Definition 8 is monotone, contractive and idempotent.*

*Proof.* This procedure is idempotent, since the projection of the alphabet is idempotent. It is contractive because for a pattern $d$, for any sequences $s \in d$, the projection of the sequence $\tilde{s} = \psi(s)$ is a subsequence of $s$. In the Definition 8 the projected sequences should be substituted by its maximal subsequences in order to avoid $\bot_E$, building the sets $\{\tilde{s}^i\}$. Thus, $s$ is a supersequence for any $\tilde{s}^i$, and, so, the projected pattern $\tilde{d} = \psi(d)$ is subsumed by the pattern $d$.

Finally, we should show monotonicity. Given two patterns $x, y \in D$, such that $x \sqsubseteq y$, i.e. $\forall s^x \in x, \exists s^y \in y : s^x \leq s^y$, consider the projected sequence of $s^x$, $\psi(s^x)$. As $s^x \leq s^y$ for some $s^y$ then for some $j_0 < j_1 < j_{|s^x|}$ (see Definition 5) $s^x_i \sqsubseteq_E s^y_{j_i}$ $(i \in 1, 2, ..., |s^x|)$, then $\psi(s^x_i) \sqsubseteq_E \psi(s^y_{j_i})$ (by the monotonicity of the alphabet projection), i.e. projected sequence preserve the subsequence relation. Thus, the alphabet projection of the pattern preserve pattern subsumption relation, $\psi(x) \leq \psi(y)$ (Proposition 1), i.e. the alphabet projection is monotone.

## 4 Sequential Pattern Structure Evaluation

### 4.1 Implementation

Nearly all state-of-the-art FCA algorithms can be adapted to process pattern structures. We adapted `AddIntent` algorithm [14], as the lattice structure is important for us to calculate stability (see the algorithm for calculating stability in [15]). To compute the semilattice operation ($\sqcap$, $\sqsubseteq$) between two sets of sequences $S = \{s^1, ...s^n\}$ and $T = \{t^1, ..., t^m\}$, $S \sqcap T$ is calculated according to Section 2.3, i.e. maximal sequences among all maximal common subsequences for any pair of $s^i$ and $t^j$. To find all common subsequences of two sequences, the following observations is useful. If $ss = \langle ss_1; ...; ss_l \rangle$ is a subsequence of $s = \langle s_1; ...; s_n \rangle$ with $j^s_i = k^s + i$ (Definition 5: $k^s$ is the index difference from which $ss$ is a subsequence of $s$) and a subsequence of $t = \langle t_1; ...; t_m \rangle$ with $j^t_i = k^t + i$ (likewise), then for any index $i \in \{1, 2, ..., l\}$, $ss_i \sqsubseteq_E (s_{j^s_i} \sqcap t_{j^t_i})$. Thus, to find maximal common subsequences between $s$ and $t$, we, first, align $s$ and $t$ in all possible ways, and then for every alignment we compute the resulting intersection and keep only the maximal ones.

### 4.2 Experiments and Discussion

The experiments are carried out on an "Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz" computer with 8Gb of memory under the Ubuntu 12.04 operating system. The algorithms are not parallelized and are coded in C++.

The dataset considered here comes from a French healthcare system [16]. Each elements of a sequence has a "complex" nature. This dataset contains 2400 patients suffering from *cancer*. Every patient is described as a sequence of hospitalizations without any timestamps. The hospitalization is a tuple with three fields: (i) healthcare institution (e.g. University Hospital of Paris ($CHU_{Paris}$)), (ii) reason of the hospitalization (e.g. a cancer disease), and (iii) set of medical procedures that the patient underwent. An example of a medical trajectory of a patient is $\langle [\text{CHU}_{Paris}, \text{Cancer}, \{P_1, P_2\}]; [\text{CH}_{Nancy}, \text{Chemo}, \{\}]; [\text{CH}_{Nancy}, \text{Chemo}, \{\}] \rangle$. This sequence represents a patient trajectory with three hospitalizations. It expresses that the patient was first admitted to the University Hospital of Paris ($CHU_{Paris}$) for a cancer problem as the reason, and underwent procedures $P_1$ and $P_2$. Then he had two consequent hospitalizations in Central Hospital of Nancy ($CH_{Nancy}$) in order to do chemotherapy with no additional procedures.

For this dataset the computation of the whole lattice is infeasible. However a medical expert is not interested in all possible patterns, but rather in patterns which answer his analysis question(s). First of all, the patterns of length 1 are unlikely to be of interest for him. Thus, we use the MLP projection of length 2 or 3 taking into account the small average length of the sequences in the dataset.

For the search of patterns containing only information about reasons and medical procedures, we should project every alphabet element on the element with the same reason and the same set of procedures, but substitute hospitalization institution by the most general element in the corresponding taxonomy. Moreover, we do not want to allow reason to be empty, i.e. all such elements should be projected onto $\perp_E$. In this case computation takes 18 seconds producing a lattice with around 34700 concepts. One of the stable concepts has the following intent $\langle[Cancer, \{App.\}]; [Ch.Prep, \{\}]; [Chemo, \{\}]\rangle$, specifying that a cancer was found during the appendix removal surgery, followed by a chemotherapy. This patterns highlight a discovered fact that acute appendicitis has been shown to occur antecedent to cancer within three years because of a carcinoma in colon or rectum [17].

To find patterns revealing dependences between hospitals and reasons all the procedures should be removed from each alphabet element and elements with most general hospital and/or with most general reason should be projected to $\perp_E$. The computation of the corresponding lattice takes 10 seconds, producing around 4200 concepts. $\langle[Region\ Lorraine, Cancer]; [Clinic\ in\ Lorraine, Chemo]\rangle$ is among stable concepts which is rather interesting, because the patients detected cancer somewhere in Region A but then went to exactly the same clinic for chemotherapy. It suggests that the department can lack from clinics for chemotherapy or the quality of the clinic is high.

## Conclusion

In this paper, we present sequential pattern structures, an original specification of pattern structures able to deal with complex sequences. Projections of sequential pattern structures allow us to efficiently build concept lattices, by specifying expert demands. To be able to introduce interesting projections, their classical definition is extended. This extension allows us to introduce special projections for sequential pattern structures. The introduced projections are efficiently used for analysis of a dataset on care trajectories.

There are two main directions for future work. First, a study on properties of generalized projections within the overall framework of FCA should be carried out. Second, projections of sequential pattern structures can be deeper analyzed, for producing even more interesting and readable patterns.

**Acknowledgements:** this research received funding from the Basic Research Program at the National Research University Higher School of Economics (Russia) and from the BioIntelligence project (France).

## References


1. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: PrefixSpan Mining Sequential Patterns Efficiently by Prefix Projected Pattern Growth. In: 17th International Conference on Data Engineering. (2001) 215–226
2. Yan, X., Han, J., Afshar, R.: CloSpan: Mining Closed Sequential Patterns in Large Databases. In: Proc. of SIAM Int'l Conf. Data Mining (SDM'03). (2003) 166–177
3. Raïssi, C., Calders, T., Poncelet, P.: Mining conjunctive sequential patterns. Data Min. Knowl. Discov. **17**(1) (2008) 77–93
4. Ding, B., Lo, D., Han, J., Khoo, S.C.: Efficient Mining of Closed Repetitive Gapped Subsequences from a Sequence Database. In: Proc. of IEEE 25th International Conference on Data Engineering, IEEE (March 2009) 1024–1035
5. Plantevit, M., Laurent, A., Laurent, D., Teisseire, M., Choong, Y.W.: Mining multidimensional and multilevel sequential patterns. ACM Transactions on Knowledge Discovery from Data **4**(1) (January 2010) 1–37
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer, Secaucus, NJ, USA (1997)
7. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In Delugach, H., Stumme, G., eds.: Conceptual Structures: Broadening the Base SE - 10. Volume 2120 of LNCS. Springer Berlin Heidelberg (2001) 129–142
8. Casas-Garriga, G.: Summarizing Sequential Data with Closed Partial Orders. In: Proc. of the 5th SIAM Int'l Conf. on Data Mining (SDM'05). (2005)
9. Ferré, S.: The Efficient Computation of Complete and Concise Substring Scales with Suffix Trees. In Kuznetsov, S.O., Schmidt, S., eds.: Formal Concept Analysis SE - 7. Volume 4390 of Lecture Notes in Computer Science. Springer (2007) 98–113
10. Klimushkin, M., Obiedkov, S.A., Roth, C.: Approaches to the Selection of Relevant Concepts in the Case of Noisy Data. In: Proc. of the 8th International Conference on Formal Concept Analysis. ICFCA'10, Springer (2010) 255–266
11. Kuznetsov, S.O.: On stability of a formal concept. Annals of Mathematics and Artificial Intelligence **49**(1-4) (2007) 101–115
12. Adda, M., Valtchev, P., Missaoui, R., Djeraba, C.: A framework for mining meaningful usage patterns within a semantically enhanced web portal. In: Proceedings of the 3rd C* Conference on Computer Science and Software Engineering. C3S2E '10, New York, NY, USA, ACM (2010) 138–147
13. Kuznetsov, S.O.: Learning of Simple Conceptual Graphs from Positive and Negative Examples. In Żytkow, J., Rauch, J., eds.: Principles of Data Mining and Knowledge Discovery SE - 47. Volume 1704 of LNCS. Springer Berlin Heidelberg (1999) 384–391
14. Merwe, D.V.D., Obiedkov, S., Kourie, D.: AddIntent: A new incremental algorithm for constructing concept lattices. In Goos, G., Hartmanis, J., Leeuwen, J., Eklund, P., eds.: Concept Lattices. Volume 2961. Springer (2004) 372–385
15. Roth, C., Obiedkov, S., Kourie, D.G.: On succinct representation of knowledge community taxonomies with formal concept analysis A Formal Concept Analysis Approach in Applied Epistemology. International Journal of Foundations of Computer Science **19**(02) (April 2008) 383–404
16. Fetter, R.B., Shin, Y., Freeman, J.L., Averill, R.F., Thompson, J.D.: Case mix definition by diagnosis-related groups. Med Care **18**(2) (February 1980) 1–53
17. Arnbjörnsson, E.: Acute appendicitis as a sign of a colorectal carcinoma. Journal of Surgical Oncology **20**(1) (May 1982) 17–20