RCA-Seq: an Original Approach for Enhancing the Analysis of Sequential Data Based on Hierarchies of Multilevel Closed Partially-Ordered Patterns

Cristina Nica, Agnès Braud, Florence Le Ber

RCA-Seq: an Original Approach for Enhancing the Analysis of Sequential Data Based on Hierarchies of Multilevel Closed Partially-Ordered Patterns

Cristina Nica[a,b,c], Agnès Braud[a,b], Florence Le Ber[a,b,c,*]

[a] *Université de Strasbourg, ICube UMR 7357, F-67400 Illkirch-Graffenstaden, France*
[b] *CNRS, ICube UMR 7357, F-67400 Illkirch-Graffenstaden, France*
[c] *ENGEES, ICube UMR 7357, F-67000 Strasbourg, France*

Abstract

Methods for analysing sequential data generally produce a huge number of sequential patterns that have then to be evaluated and interpreted by domain experts. To diminish this number and thus the difficulty of the interpretation task, methods that directly extract a more compact representation of sequential patterns, namely closed partially-ordered patterns (CPO-patterns), were introduced. In spite of the fewer number of obtained CPO-patterns, their analysis is still a challenging task for experts since they are unorganised and besides, do not provide a global view of the discovered regularities. To address these problems, we present and formalise an original approach within the framework of Relational Concept Analysis (RCA), referred to as RCA-Seq, that focuses on facilitating the interpretation task of experts. The hierarchical RCA result allows to directly obtain and organize the relationships between the extracted CPO-patterns. Moreover, a generalisation order on items is also revealed, and multilevel CPO-patterns are obtained. Therefore, a hierarchy of such CPO-patterns guides the interpretation task, helps experts in better understanding the extracted patterns, and minimises the chance of overlooking interesting CPO-patterns. RCA-Seq is compared with another approach that relies on pattern structures. In addition, we highlight the adaptability of RCA-Seq by integrating a user-defined tax-

*Corresponding author
Email addresses: nica.cristina87@gmail.com (Cristina Nica), agnes.braud@unistra.fr (Agnès Braud), florence.leber@engees.unistra.fr (Florence Le Ber)

onomy over the items, and by considering user-specified constraints on the order relations on itemsets.

Keywords: Closed Partially-Ordered Patterns, Hierarchy of Multilevel Patterns, Formal Concept Analysis, Relational Concept Analysis, Sequential Data Analysis

---

1. Introduction

Formal Concept Analysis (FCA, [1]) is a mathematical formalism which allows to analyse binary or more complex data by grouping objects with shared properties in formal concepts. These concepts are organised in a hierarchical structure called a concept lattice. Relational Concept Analysis (RCA, [2]) is an extension of FCA designed to deal with relational data, and that results in a family of interrelated lattices, one lattice for each category of objects. It has been used with success in various domain, e.g. the fuzzy semantic annotation of web resources [3], the analysis and re-engineering of software models [4], or semantic wikis [5], or the extraction of rules from large datasets [6]. Nevertheless, RCA, as FCA, has some major challenges, one of them being the complexity of its result, since the number of concepts grows exponentially with the size of the dataset. However, several measures have been proposed to help the selection of relevant concepts (e.g. [7]). Furthermore, regarding RCA, users have to manually navigate the interrelated lattices in order to highlight the relationships between different categories of objects, which can be a complex task when there are several lattices. To deal with this problem, we proposed in a previous work [8] to synthesise the navigation paths into closed partially-ordered patterns (CPO-patterns [9]), i.e. directed acyclic graphs where vertices are labelled with information extracted from the concepts out of the family of lattices.

Besides, sequential pattern mining is an active research domain whose aim is to find regularities in sequential data that can be assessed and interpreted by experts [10]. Various algorithms have therefore been proposed [11] and many of them focus on efficiently extracting concise representations of sequential patterns (e.g. closed sequential patterns [12]). To obtain a more compact set of such sequential patterns, efficient algorithms for directly mining CPO-patterns were introduced in [13, 14]. Precisely, a CPO-pattern summarises a set of closed sequential patterns that coexist in the same analysed sequences, and it has a graphical representation that facilitates the

interpretation step. However, regardless of the fewer number of obtained CPO-patterns vs. sequential patterns, there are still some limitations as follows:

1. the interpretation task remains difficult since these CPO-patterns are unorganised; thus, experts should manually figure out how these CPO-patterns relate to each other;
2. experts do not have a global view of the discovered CPO-patterns; thus, pertinent CPO-patterns can be overlooked;
3. some interesting CPO-patterns cannot be found since they cannot be inferred. For example, one cannot infer the regularity *broccoli before citrus* from *broccoli before lemon*, and *broccoli before orange* without employing generalisation knowledge.

The approach we propose here is a contribution to both fields of RCA and sequential pattern analysis. The whole process is described in Fig. 1 and it is a twofold approach: (sequential data) exploration and (pattern) extraction. Firstly, RCA is applied to a *relational context family* (i.e. the RCA input) that encodes the analysed sequential data in order to obtain a *family of concept lattices* (i.e. the RCA result). Secondly, the interrelated concepts from the RCA result, are navigated to directly extract a CPO-pattern for each concept of a chosen lattice (called *main lattice*). The obtained CPO-patterns are automatically organised thanks to the generalisation order that exists between the associated concepts. Moreover, RCA reveals a generalisation order on items, and thus *multilevel* [15] CPO-patterns are obtained that provide a global view of the regularities hidden in sequential data. This generalisation order on items combined with the generalisation order on the CPO-pattern structures provides more instructive results. Finally, the hierarchy guides the experts in interpreting the obtained patterns, and provides a quick way to navigate to interesting CPO-patterns.

In this paper we contribute on:

1. formalising an RCA-based approach, referred to as RCA-Seq, which relies on the principles given by [8], and presenting an algorithm CPOHrchy that directly extracts multilevel CPO-patterns by navigating the RCA result;
2. providing a complexity analysis of RCA-Seq based on the complexity of RCA;

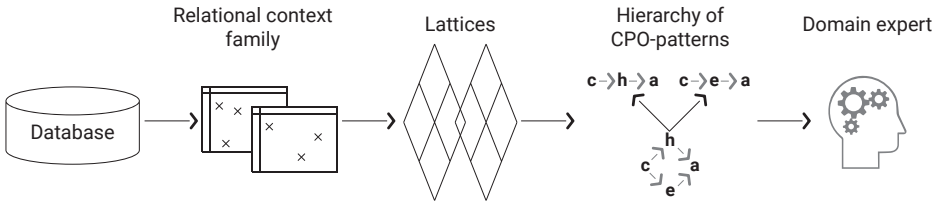


Figure 1: RCA-based approach schema (taken from [8])

3. comparing RCA-Seq with a pattern-structure approach [16] and with a sequential data mining approach [14];
4. showing through an illustrative example that the structure of a hierarchy of multilevel CPO-patterns can be exploited to facilitate sequential data analysis;
5. extending RCA-Seq to integrate a user-defined taxonomy, and to extract CPO-patterns with user-specified constraints on the order relations on itemsets.

The paper is structured as follows. Section 2 gives the theoretical background of our work while Sect. 3 is an overview of the related work. Section 4 introduces a running example and details how to process it by using RCA. Section 5 focuses on interesting properties of the RCA result, and Sect. 6 defines and illustrates our proposal for automatically extracting hierarchies of multilevel CPO-patterns. In Sect. 7 we compare RCA-Seq with a pattern structure approach and show how it can be extended. Finally, we conclude the paper and give some perspective of our work in Sect. 8.

2. Preliminaries

Our approach relies on both sequential pattern analysis and FCA domains. In this section, we recall their definitions and principles.

2.1. Sequences, Sequential Patterns and PO-Patterns

Let $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$ be a set of *items*. An *itemset* $\mathfrak{I} = \{I_{j_1}, \cdots, I_{j_k}\}$, where $I_{j_i} \in \mathcal{I}$, is a non empty set of items. Let $\mathcal{J} = 2^{\mathcal{I}} - \emptyset$ be the set of all itemsets built from $\mathcal{I}$. A *sequence*, denoted by $S = \langle \mathfrak{I}_1 \mathfrak{I}_2 \cdots \mathfrak{I}_p \rangle$, is an ordered list of itemsets $\mathfrak{I}_1, \ldots, \mathfrak{I}_p \in \mathcal{J}$. The order on the itemsets of S is denoted by $\leqslant_S$. Thus, for any two itemsets $\mathfrak{I}_\alpha$ and $\mathfrak{I}_\beta$ in S it is possible

to determine if $\mathfrak{I}_\alpha \leqslant_S \mathfrak{I}_\beta$ or $\mathfrak{I}_\beta \leqslant_S \mathfrak{I}_\alpha$. The order $\leqslant_S$ can be a temporal relation (e.g. {New Year's Eve 2015} *is preceded by* {New Year's Eve 2013}), a topological relation (e.g. {Polygon1} *contains* {Polygon2}), a directional relation (e.g. {Canada, US} *is north of* {Mexico}) or any other order in which related itemsets follow each other.

A sequence $S = \langle \mathfrak{I}_1 \mathfrak{I}_2 \cdots \mathfrak{I}_p \rangle$ is a *subsequence* of another sequence $S' = \langle \mathfrak{I}'_1 \mathfrak{I}'_2 \cdots \mathfrak{I}'_q \rangle$, denoted by $S \preceq_s S'$, if $p \leq q$ and if there are integers $j_1 < j_2 < \cdots < j_k < \cdots < j_p$ such that $\mathfrak{I}_1 \subseteq \mathfrak{I}'_{j_1}, \mathfrak{I}_2 \subseteq \mathfrak{I}'_{j_2}, \ldots, \mathfrak{I}_p \subseteq \mathfrak{I}'_{j_p}$.

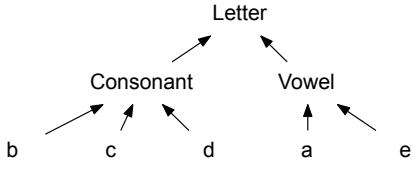
A small example of a sequence database $\mathcal{D}_\mathcal{S}$, i.e. a set of sequences, that contains three sequences built from the set of items $\mathcal{I} = \{a, b, c, d\}$ is shown in Fig. 2(a). For example, $S_1 = \langle \{a\}\{b, c\}\{d\} \rangle$ is a sequence with three itemsets $\{a\}$, $\{b, c\}$ and $\{d\}$. A sequence $S' = \langle \{a\}\{c\} \rangle$ is a subsequence of S_1, $S' \preceq_s S_1$, since $\{a\} \subseteq \{a\}$, $\{c\} \subseteq \{b, c\}$, and the order on itemsets is preserved.

Let us now consider a *partial order* on the items, denoted by $(\mathcal{I}, \leqslant)$. An itemset can thus contain items of various levels. Then the set inclusion on $\mathcal{J}$ is redefined as follows: $\mathfrak{I} \subseteq \mathfrak{I}'$ if $\forall I_j \in \mathfrak{I}, \exists I_{j'} \in \mathfrak{I}', I_{j'} \leqslant I_j$ and $\forall I_l \neq I_j$, $\exists I_{l'} \neq I_{j'}$ such that $I_{l'} \leqslant I_l$. The order on sequences is defined as previously.

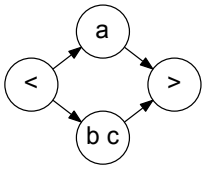
To illustrate this, let us consider the set of items $\mathcal{I}_1 = \{a, b, c, d, e,$ *Consonant*, *Vowel*, *Letter*$\}$ and the partial order $(\mathcal{I}_1, \leqslant)$ depicted in Fig. 2(b), where an edge represents the binary relation *is-a*, denoted by $\leqslant$. For example, $a \leqslant$ *Vowel* designates that letter "a" is a vowel. If we have two itemsets $\{a, b, c\}$ and $\{a,$ *Consonant*$\}$, then $\{a,$ *Consonant*$\} \subseteq \{a, b, c\}$ since $a \leqslant a$ and $b \leqslant$ *Consonant* (or $c \leqslant$ *Consonant*).

Seq Id	Sequence
S_1	$\langle \{a\}\{b, c\}\{d\} \rangle$
S_2	$\langle \{a, b, c\} \rangle$
S_3	$\langle \{a\}\{b, c\}\{a\} \rangle$

(a) $\mathcal{D}_\mathcal{S}$



(b) $(\mathcal{I}_1, \leqslant)$



(c) $\mathcal{G}_1$

Figure 2: (a) a sequence database $\mathcal{D}_\mathcal{S}$; (b) a partial order on the set of items $\mathcal{I}_1$; (c) a PO-pattern associated with the set of sequences $\{S_1, S_2, S_3\} \in \mathcal{D}_\mathcal{S}$

Sequential patterns were introduced by [10] as frequent subsequences found in a sequence database. A subsequence is associated with a support, i.e. the number of sequences containing this subsequence. Formally,

the support of a subsequence M in a sequence dataset $\mathcal{D}_\mathcal{S}$ is defined as $Support(M) = |\{S \in \mathcal{D}_\mathcal{S} | M \preceq_s S\}|$. Given a user-specified minimum support θ, the M subsequence is θ-frequent, if $Support(M) \geq \theta$. A θ-frequent subsequence is called a sequential pattern in the following. For instance, in Fig. 2(a) four sequential patterns are found for $\theta = 3$, namely $M_1 = \langle\{a\}\rangle$, $M_2 = \langle\{b\}\rangle$, $M_3 = \langle\{c\}\rangle$, and $M_4 = \langle\{b, c\}\rangle$.

Partially-ordered patterns (PO-patterns) were introduced by [9], to synthesise sets of sequential patterns. Formally, a PO-pattern is a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, l)$. $\mathcal{V}$ is the set of vertices, $\mathcal{E}$ is the set of directed edges such that $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and l is the labelling function mapping each vertex to an itemset. This structure allows to define a strict partial order on vertices: let $u \neq v$, $u < v$ if and only if there is a directed path from vertex u to vertex v. Each path of the graph represents a sequential pattern, and the set of paths in $\mathcal{G}$ is denoted by $\mathcal{P}_\mathcal{G}$. A PO-pattern is associated with a set of sequences $\mathcal{S}_\mathcal{G}$ where each sequence contains all paths within $\mathcal{P}_\mathcal{G}$. The support of a PO-pattern is defined as $Support(\mathcal{G}) = |\mathcal{S}_\mathcal{G}| = |\{S \in \mathcal{D}_\mathcal{S} | \forall M \in \mathcal{P}_\mathcal{G}, M \preceq_s S\}|$. Furthermore, let $\mathcal{G}$ and $\mathcal{G}'$ be two PO-patterns with $\mathcal{P}_\mathcal{G}$ and $\mathcal{P}_{\mathcal{G}'}$ their sets of paths. $\mathcal{G}'$ is a sub PO-pattern of $\mathcal{G}$, denoted by $\mathcal{G}' \preceq_g \mathcal{G}$, if $\forall M' \in \mathcal{P}_{\mathcal{G}'}, \exists M \in \mathcal{P}_\mathcal{G}$ such that $M' \preceq_s M$. A PO-pattern $\mathcal{G}$ is *closed*, referred to as *CPO-pattern*, if there exists no PO-pattern $\mathcal{G}'$ such that $\mathcal{G} \prec_g \mathcal{G}'$ with $\mathcal{S}_\mathcal{G} = \mathcal{S}_{\mathcal{G}'}$. Figure 2(c) shows the CPO-pattern $\mathcal{G}_1$ that synthesises the aforementioned M_1, M_2, M_3, and M_4 sequential patterns ($M_1 = \langle\{a\}\rangle$ is the upper path, M_4 is the lower one, M_2 and M_3 are covered by M_4) that coexist exactly in the same sequences S_1, S_2, and S_3.

2.2. FCA

FCA considers an object-attribute context as input, and builds from it a concept lattice used to analyse the objects and their attributes. Concisely, an *object-attribute context* K is a 3-tuple (G, M, I), where G is a set of objects, M is a set of attributes, and $I \subseteq G \times M$ is an incidence relation that specifies which objects have which attributes. A *formal concept* C derived from K is a pair (X, Y) where $X = Y'$ and $Y = X'$ with $Y' = \{g \in G | \forall m \in Y, (g, m) \in I\}$ and $X' = \{m \in M | \forall g \in X, (g, m) \in I\}$. X and Y are respectively the *extent* and the *intent* of concept C. Let $\mathcal{C}_K$ be the set of all formal concepts that can be built on K, and $C_1 = (X_1, Y_1)$ and $C_2 = (X_2, Y_2)$ be two concepts from $\mathcal{C}_K$. The concept *generalisation order* $\preceq_K$ is defined by $C_1 \preceq_K C_2$ if and only if $X_1 \subseteq X_2$ ($\Leftrightarrow Y_2 \subseteq Y_1$). $\mathcal{L}_K$=($\mathcal{C}_K$, $\preceq_K$) is the *concept lattice*

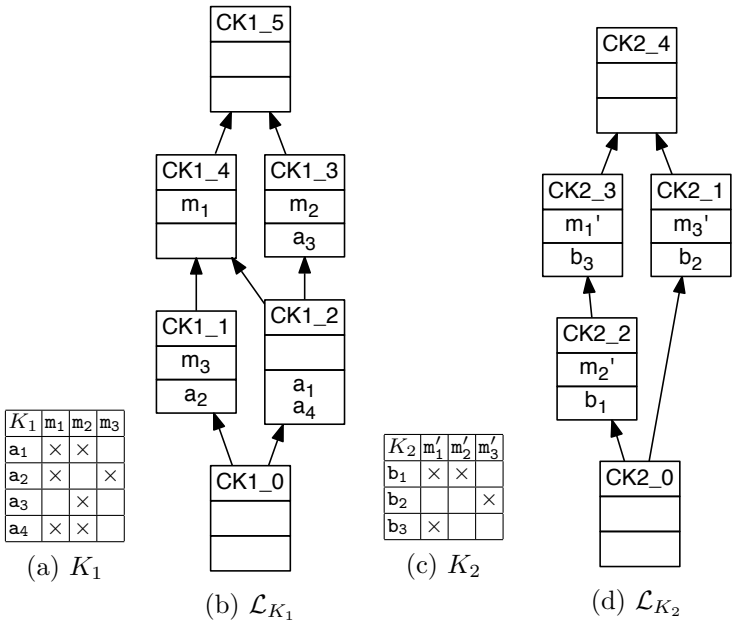
built from K. We denote by $\top(\mathcal{L}_K)$ the concept from $\mathcal{L}_K$ whose extent has all the objects, and by $\bot(\mathcal{L}_K)$ the concept from $\mathcal{L}_K$ whose intent has all the attributes. The support of a concept $C = (X, Y)$ is defined as the cardinality of X. A concept is θ-frequent if $|X| \geq \theta$, where θ is user-defined minimum support. The set of all frequent concepts of $\mathcal{C}_K$ is called the *iceberg concept lattice* of the context [17].

Two object-attribute contexts $K_1 = (G_1, M_1, I_1)$ and $K_2 = (G_2, M_2, I_2)$ represented as cross-tables are shown in Fig. 3(a) and 3(c), respectively. For instance, in cross-table K_1 the rows are the objects $G_1 = \{\mathtt{a_1}, \mathtt{a_2}, \mathtt{a_3}, \mathtt{a_4}\}$, the columns are the attributes $M_1 = \{\mathtt{m_1}, \mathtt{m_2}, \mathtt{m_3}\}$, and a cross in a cell identified by a pair $(\mathtt{a_i}, \mathtt{m_j}) \in I_1$ signifies that object $\mathtt{a_i} \in G_1$ has attribute $\mathtt{m_j} \in M_1$. Figures 3(b) and 3(d) illustrate the concept lattices $\mathcal{L}_{K_1}$ and $\mathcal{L}_{K_2}$ built respectively from K_1 and K_2. Each concept is represented by a box structured from top to bottom as follows: concept name, simplified intent, and simplified extent. The arrows represent the generalisation order on concepts, e.g. in Fig. 3(d) $\mathtt{CK2\_2} \preceq_{K_2} \mathtt{CK2\_3}$. The representation of the lattice is simplified as every attribute/object is top-down/bottom-up inherited. Thus an attribute/object is shown only in the highest/lowest concept where it appears. For example, in Fig. 3(b), concept `CK1_3` has the extent $\{\mathtt{a_1}, \mathtt{a_3}, \mathtt{a_4}\}$ where the objects $\mathtt{a_1}$ and $\mathtt{a_4}$ are inherited from concept `CK1_2`. The intent of `CK1_2` is $\{\mathtt{m_1}, \mathtt{m_2}\}$ where attributes $\mathtt{m_1}$ and $\mathtt{m_2}$ are inherited from `CK1_4` and `CK1_3`, respectively. `CK1_1` has the extent $\{\mathtt{a_2}\}$, and thus its support is $|\{\mathtt{a_2}\}| = 1$. If a minimum support $\theta = 2$ is defined, then `CK1_1` is not considered.

2.3. RCA

RCA extends the purpose of FCA to relational data. Compared to the data used in FCA, relational data represent several categories of objects rather than one category, and capture relations between these objects. The power of relational approaches lies in their ability to handle all this information. To handle relational data in RCA, two types of contexts are employed: object-attribute contexts describe the objects as in FCA, and *object-object* contexts encode binary relations between objects. For instance, by considering the K_1 and K_2 object-attribute contexts from Fig. 3 we can build the R_1 object-object context that defines the relation $r_1 \subseteq G_1 \times G_2$ between the objects of G_1 and G_2.

Object-attribute and object-object contexts are gathered in a so-called *Relational Context Family* (RCF) that is the input of RCA. An RCF is a pair



K_1	$\mathtt{m}_1$	$\mathtt{m}_2$	$\mathtt{m}_3$
$\mathtt{a}_1$	×	×	
$\mathtt{a}_2$	×		×
$\mathtt{a}_3$		×	
$\mathtt{a}_4$	×	×	

(a) K_1

(b) $\mathcal{L}_{K_1}$

K_2	$\mathtt{m}'_1$	$\mathtt{m}'_2$	$\mathtt{m}'_3$
$\mathtt{b}_1$	×	×	
$\mathtt{b}_2$			×
$\mathtt{b}_3$	×		

(c) K_2

(d) $\mathcal{L}_{K_2}$

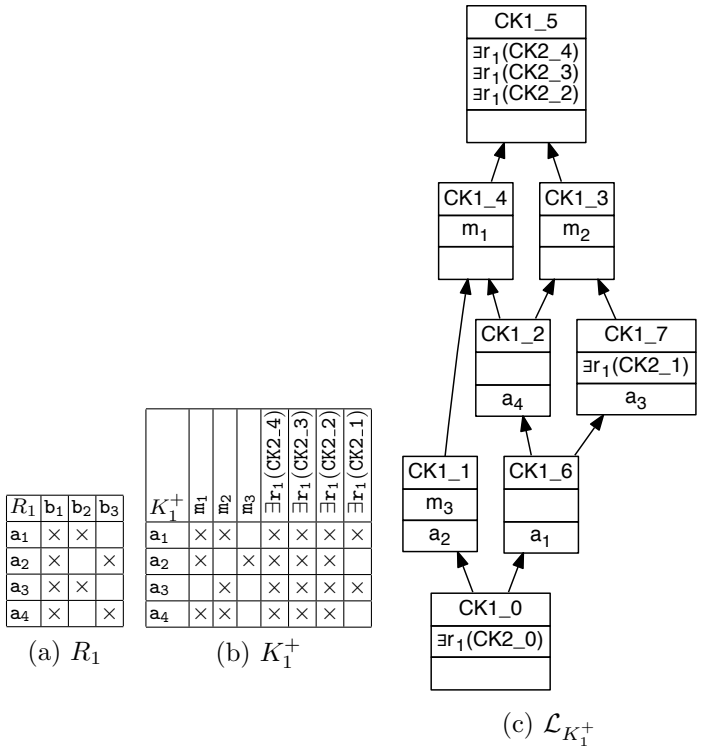
Figure 3: Object-attribute contexts and their concept lattices

$(\mathcal{K}, \mathcal{R})$, where $\mathcal{K}$ is a set of object-attribute contexts and $\mathcal{R}$ is a set of object-object contexts. $\mathcal{K}$ contains n object-attribute contexts $K_i = (G_i, M_i, I_i)\,, i \in \{1, ..., n\}$. $\mathcal{R}$ contains m object-object contexts $R_j = (G_k, G_l, r_j)\,, j \in \{1, ..., m\}$, where $r_j \subseteq G_k \times G_l$ is a binary relation with $k, l \in \{1, ..., n\}$, $G_k = dom(r_j)$ is the domain of r_j, and $G_l = ran(r_j)$ is the range of r_j. For example, the RCF $(\{K_1, K_2\}, \{R_1\})$ can be built from Figs. 3a, 3c and 4a.

RCA then relies on a *relational scaling mechanism* which aims at capturing in object-attribute contexts the relational information encoded in object-object contexts. It is used to transform a relation r_j into a set of *relational attributes* that extends the object-attribute context describing the set of objects $dom(r_j)$. Here, we focus on a specific type of relational scaling, namely the *existential scaling*. A relational attribute $\exists r_j(C)$, where $\exists$ is the existential quantifier and $C = (X, Y)$ is a concept whose extent contains objects from $ran(r_j)$, is owned by an object $g \in dom(r_j)$ if $r_j(g) \cap X \neq \emptyset$.

The RCA process consists in applying FCA first on each object-attribute context of an RCF, and then iteratively on each object-attribute context extended by the relational attributes created by using the learnt concepts from the previous step. The RCA result is obtained when the families of

lattices of two consecutive steps are isomorphic or in an equivalent manner, the object-attribute contexts are unchanged.



R_1	b_1	b_2	b_3
a_1	×	×	
a_2	×		×
a_3	×	×	
a_4	×		×

(a) R_1

K_1^+	m_1	m_2	m_3	$\exists r_1$(CK2_4)	$\exists r_1$(CK2_3)	$\exists r_1$(CK2_2)	$\exists r_1$(CK2_1)
a_1	×	×		×	×	×	×
a_2	×		×	×	×	×	
a_3		×		×	×	×	×
a_4	×	×		×	×	×	

(b) K_1^+

(c) $\mathcal{L}_{K_1^+}$

Figure 4: (a) R_1 object-object context that defines the binary relation r_1; (b) K_1^+ is the extended object-attribute context obtained after relational scaling by using existential quantifier; (c) $\mathcal{L}_{K_1^+}$ concept lattice of K_1^+

For our example, $\mathcal{L}_{K_1}$ (Fig. 3(b)) and $\mathcal{L}_{K_2}$ (Fig. 3(d)) are respectively the initial lattices for the G_1 and G_2 sets of objects. K_1 is extended with relational attributes built by using concepts of $\mathcal{L}_{K_2}$, and thus the extended object-attribute context K_1^+ (Fig. 4(b)) is obtained. For instance, object $\mathtt{a_3}$ has the relational attribute $\exists r_1(\mathtt{CK2\_1})$ since the extent of $\mathtt{CK2\_1}$ contains $\mathtt{b_2}$ and $(\mathtt{a_3}, \mathtt{b_2}) \in r_1$. The RCA result consists of $\mathcal{L}_{K_1^+}$ (Fig. 4(c)) and $\mathcal{L}_{K_2}$ concept lattices. There is no other iteration since $\mathcal{L}_{K_2}$ has no new learnt concept. The RCA result is navigated following the concepts used to build relational attributes, e.g. $\exists r_1(\mathtt{CK2\_1})$ of the $\mathtt{CK1\_7}$ intent (Fig. 4(c)) allows us to navigate from $\mathcal{L}_{K_1^+}$ lattice to $\mathtt{CK2\_1}$ concept in $\mathcal{L}_{K_2}$ lattice.

3. Related Work

As far as we know, our work is the first attempt to explore sequential data by means of RCA. Nevertheless, there are various FCA approaches dealing with sequential data [22, 23, 9, 24]. For example, in [22] Temporal Concept Analysis (TCA) is introduced to explore objects described by dates and states. These data are merged into a single context, and temporal relations between the derived concepts are actually revealed by manually analysing the dates in the concepts. [25] used TCA to analyse sequential data about crime suspects. In contrast, RCA-Seq considers the temporal relations between dates as object-object relations, and it automatically reveals the temporal links between concepts of different lattices through the relational scaling mechanism. Authors in [24] propose to use pattern structures [26] to build a concept lattice on complex sequential data about care trajectories. The pattern structure is $(P, (S, \sqcap), \delta)$, where P is the set of patients, S is a set of sequences and their subsequences, and $\sqcap$ is the set intersection. Each patient of P is described by a sequence (and its subsequences) through δ relation.This approach is deepened in [16], where object descriptions are organised into a semi-lattice of closed sets of closed subsequences, which are built based on the corresponding CPO-patterns extracted from a sequence dataset (see a comparison with our approach in Sect. 7.1). A similar approach is used for analysing demographic sequences in [27].

Besides, sequential pattern mining area proposes various methods to explore sequential data [11]. Sequential patterns [10] have been used for different purposes, e.g. classification [28] and prediction [29]. Most of the existing works focus on mining efficiently sequential patterns [30, 31]. To decrease the number of generated patterns and to have acceptable performance measures, many studies concentrate on finding more concise representations of sequential patterns such as closed sequential patterns [12], and maximal sequential patterns [32]. Regardless of the fewer number of generated concise patterns, the interpretation task is still difficult since there are closed sequential patterns that occur in the same set of sequences from the analysed database. Thus it is not an easy task for experts to gather these patterns. To this end, CPO-patterns are proposed in [9] where the generated closed sequential patterns are post-processed. To our knowledge, there are two algorithms [13] and [14] that directly extract CPO-patterns, but with no generalisation on items.

FCA can classify and filter complex data (e.g. graphs and sequences)

based on its extensions, such as Logical Concept Analysis [33], pattern structures [26], and Graph-FCA [34]. Lattices of closed partial orders were introduced in [35], where a concept is defined as follows: *a formal concept is a pair* (S, G_p) *where* G_p *is a closed partial order for the set of sequences* S*, and the set of sequences* S *is closed for the partial order* G_p. In addition, Cellier et al. explain in [36] how Logical Concept Analysis can be used to organise already extracted patterns into a concept lattice. Similarly, in [37] sequential patterns are mined using M3SP [38] algorithm from patient trajectory data. Then, a hierarchy of such patterns is built based on FCA by considering patients as objects, and sequential patterns as attributes. Following the same idea, the set of CPO-patterns obtained by using [13] or [14] can be organised into a lattice. The context would be (S, P, I) with S the set of sequences and P the set of CPO-patterns, and $I(s, p)$ if $s \in S$ is in the support of $p \in P$. By construction, to each subset of sequences is associated a unique CPO-pattern. Thus a concept is made of a CPO-pattern (intent) and the corresponding maximal subset of sequences (extent). If a user-defined minimum support is used, an iceberg lattice can be built as well. The resulting concept lattice can be compared to the hierarchy of CPO-patterns built in RCA-Seq. Let us however notice that first, in our approach, the CPO-patterns are extracted and organised into a hierarchy directly from the RCA result. Secondly, a partial order on items is generated, and thus multilevel CPO-patterns are obtained rather than those extracted in [13, 14]. Such results can be related to [15], where generalised sequential patterns are extracted based on a user-given taxonomy. In contrast, RCA allows to discover a taxonomy over the items.

Recently, [39] has proposed to help the analysis of the RCA result by extracting so-called concept graphs. There is no main lattice and both relational (inter-lattices) and hierarchical (intra-lattices) links are included within the graphs. Nevertheless, it gives the same results as those presented in [8] when applied to sequential data.

4. Relational Analysis of Sequential Data

In this section, we introduce a small example inspired by the football team sport and show how it is modelled and processed by RCA.

4.1. Running Example

Patterns hidden in sequential football data about players (e.g. midfielders and goalkeepers), and their training session histories can provide valuable knowledge for coaches. Here, we propose to study the football drills (e.g. dribbling and pitch vision) that often constitute the criteria on which coaches decide the squad roles of players for the upcoming match, namely starters, substitutes, and reserves. Football drills are devised to develop the skills of players during their daily training sessions. There is a player evaluation prior to each match when coaches assign the squad roles to the football players. A *football player sequence* consists in a chronologically ordered set of training sessions for a player, and a corresponding evaluation which ends the sequence. A training session represents an *observation itemset* of football drills, while the player evaluation represents a *target itemset*; here this last itemset has only one item, specifically the squad role assigned to the player. Each observation or target itemset is associated with a pair (Player,Date) that uniquely identifies the itemset, where Date is the time of the Player trainig session or evaluation and is formatted as Day/Month.

Table 1 illustrates a small example of sequential football data collected during a year, where we focus on the starter squad role. Here, we study only the training sessions of midfielders prior to their evaluations. There can be several sequences of the same midfielder. The football sequences are built from three items as follows: two football drills `PASSING` and `VISION`, and a squad role `STARTER`. The football drills are rated as `good` or `excellent`, and the starter squad role is evaluated as `important`. Thus, in this example we deal with *qualitative sequential data*. For example, the last three rows from Tab. 1 constitute the sequence $\langle\{\texttt{PASSING}_{\texttt{excellent}}\}\{\texttt{PASSING}_{\texttt{good}}, \texttt{VISION}_{\texttt{excellent}}\}\{\texttt{STARTER}_{\texttt{important}}\}\rangle$ of midfielder P3. The first training – identified by the pair (P3,11/04) – was on April 11$^{\text{th}}$ when the passing skills of midfielder P3 were excellent. The second training was on April 12$^{\text{th}}$ when the passing and pitch vision skills of the same midfielder were respectively good and excellent. Then, on April 13$^{\text{th}}$, midfielder P3 was assigned as an important starter for the upcoming match.

Exploiting the relational character of our sequential football data, we propose to model our data given in Tab. 1 as shown in Fig. 5. There are four rectangles, one for each set of objects we manipulate as follows: squad roles (SR), football drills (FD), player evaluations (PE), and training sessions (TS).

Player	Date	Training Session		Evaluation
		PASSING	VISION	STARTER
P1	25/09	–	excellent	–
	26/09	good	–	–
	27/09	good	excellent	–
	28/09	–	–	important
	17/11	excellent	–	–
	18/11	–	–	important
P2	13/05	excellent	–	–
	14/05	good	excellent	–
	15/05	–	–	important
	08/02	good	–	–
	09/02	–	–	important
P3	11/04	excellent	–	–
	12/04	good	excellent	–
	13/04	–	–	important

Table 1: Small illustrative example of sequential football data

The squad roles are linked to player evaluations by a qualitative binary relation *has squad role important*. Similarly, training sessions are linked to football drills by the qualitative relation *has drill*, which can be rated either as *good* or *excellent*. The name 'qualitative' relation is used to refer to a relation that includes an evaluation (good, important, etc.). Player evaluations/training sessions and training sessions are linked by a temporal binary relation *is preceded by* that associates a player evaluation/training session to a training session if the player evaluation/training session is preceded in time by this training session. There is no temporal binary relation between player evaluations since our aim is to study the football drills in order to help coaches to identify the important starters for the upcoming match.

4.2. Applying RCA on Sequential Data

RCA processing firstly requires to encode the sequential football data (Tab. 1) into a relational context family. This is done in Tab. 2 according to the data model depicted in Fig. 5. The cross-tables `KFD` (football drills), `KPE` (player evaluations), and `KTS` (training sessions) represent object-attribute contexts. As explained before the set of football drills contains two objects `PASSING` and `VISION`. The set of player evaluations/training sessions contains pairs (Player,Date) (e.g. the (P1,28/09) pair in `KPE`). Since the set of squad roles contains only one object `STARTER` there is no object-attribute context

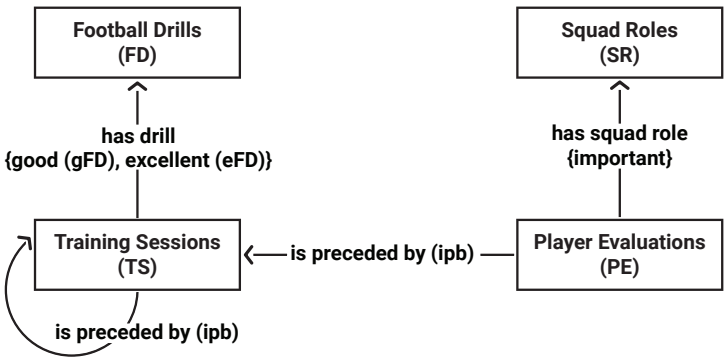


Figure 5: The modelling of the sequential football data from Tab. 1

of squad roles. KTS and KPE cross-tables have no column since a training session (resp. player evaluation) is described only by using qualitative relations. RPE-ipb-TS (player evaluation *is preceded by* training session) and RTS-ipb-TS (training session *is preceded by* training session) cross-tables represent temporal object-object contexts since both define temporal relations. RgFD (training session *has good* football drill) and ReFD (training session *has excellent* football drill) cross-tables represent qualitative object-object contexts since both define qualitative relations.

Secondly, RCA is applied (by using RCAExplore[1] tool) on the aforementioned relational context family and the family of concept lattices given in Fig. 6 is obtained after four iterations. There is a concept lattice for each object-attribute context as follows: $\mathcal{L}_{\texttt{KPE}}$ (player evaluations), $\mathcal{L}_{\texttt{KFD}}$ (football drills), and $\mathcal{L}_{\texttt{KTS}}$ (training sessions). The obtained lattices contain *temporal* and/or *qualitative* relational attributes. For instance, the relational attribute $\exists$ReFD(CKFD_1) of concept CKTS_5 intent in $\mathcal{L}_{\texttt{KTS}}$ lattice is a qualitative one since it introduces the qualitative relation *has drill excellent*, and allows us to navigate from $\mathcal{L}_{\texttt{KTS}}$ to $\mathcal{L}_{\texttt{KFD}}$. Similarly, the relational attribute $\exists$RPE-ipb-TS(CKTS_9) of concept CKPE_2 intent in $\mathcal{L}_{\texttt{KPE}}$ is a temporal one since it introduces the temporal relation *is preceded by*, and allows us to navigate from $\mathcal{L}_{\texttt{KPE}}$ to $\mathcal{L}_{\texttt{KTS}}$.

[1] http://dataqual.engees.unistra.fr/logiciels/rcaExplore tool computes an (iceberg) concept lattice by using an attribute-incremental version of AddIntent algorithm [18], referred to as AddExtent

KFD	PASSING	VISION
PASSING	×	
VISION		×

KPE
(P1,28/09)
(P1,18/11)
(P2,15/05)
(P2,09/02)
(P3,13/04)

KTS
(P1,25/09)
(P1,26/09)
(P1,27/09)
(P1,17/11)
(P2,13/05)
(P2,14/05)
(P2,08/02)
(P3,11/04)
(P3,12/04)

RPE-ipb-TS	(P1,25/09)	(P1,26/09)	(P1,27/09)	(P1,17/11)	(P2,13/05)	(P2,14/05)	(P2,08/02)	(P3,11/04)	(P3,12/04)
(P1,28/09)	×	×	×						
(P1,18/11)				×					
(P2,15/05)					×	×			
(P2,09/02)							×		
(P3,13/04)								×	×

RTS-ipb-TS	(P1,25/09)	(P1,26/09)	(P1,27/09)	(P1,17/11)	(P2,13/05)	(P2,14/05)	(P2,08/02)	(P3,11/04)	(P3,12/04)
(P1,25/09)									
(P1,26/09)	×								
(P1,27/09)	×	×							
(P1,17/11)									
(P2,13/05)									
(P2,14/05)					×				
(P2,08/02)									
(P3,11/04)									
(P3,12/04)								×	

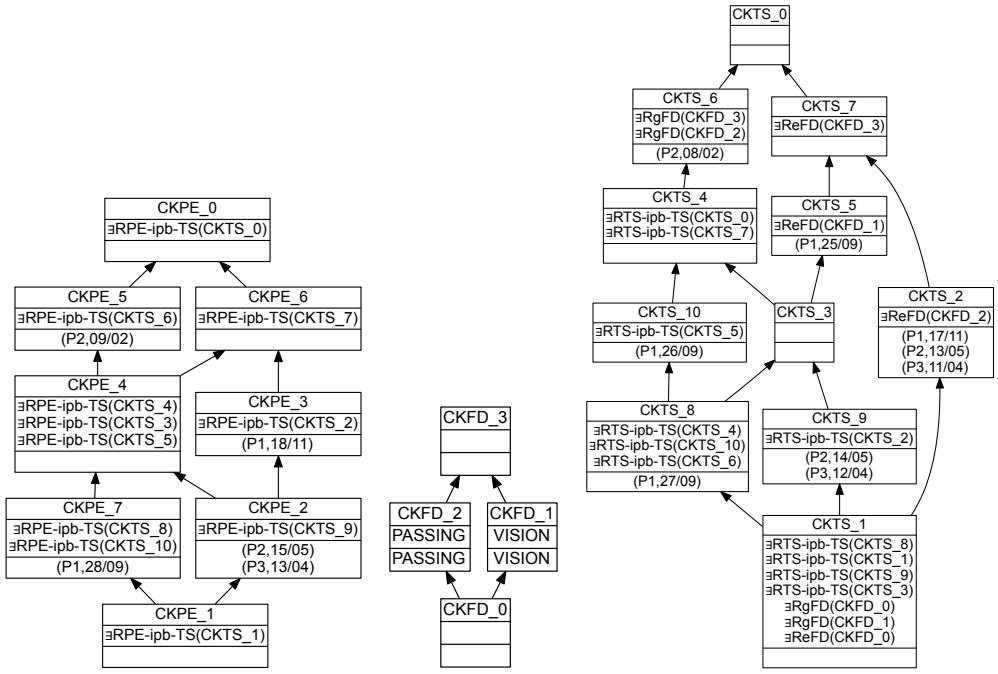
RgFD	PASSING	VISION
(P1,25/09)		
(P1,26/09)	×	
(P1,27/09)	×	
(P1,17/11)		
(P2,13/05)		
(P2,14/05)	×	
(P2,08/02)	×	
(P3,11/04)		
(P3,12/04)	×	

ReFD	PASSING	VISION
(P1,25/09)		×
(P1,26/09)		
(P1,27/09)		×
(P1,17/11)	×	
(P2,13/05)	×	
(P2,14/05)		×
(P2,08/02)		
(P3,11/04)	×	
(P3,12/04)		×

Table 2: RCA input composed of object-attribute contexts: KFD, KPE, and KTS; temporal object-object contexts: RPE-ipb-TS and RTS-ipb-TS; qualitative object-object contexts: RgFD and ReFD

5. Structure and Properties of RCA on Sequential Data

In the following we consider a database $\mathcal{D}_{\mathcal{S}}$ where sequences are made of two parts: 1) successive qualitative observations, and 2) one synthetic observation or evaluation (a target). A sequence is written as follows: $S = \langle \mathfrak{I}_{t_1} \mathfrak{I}_{t_2} \cdots \mathfrak{I}_{t_p} \mathfrak{I}_m \rangle$ where $\mathfrak{I}_{t_1}, \mathfrak{I}_{t_2}, \ldots, \mathfrak{I}_{t_p} \in \mathcal{J}_t$ and $\mathfrak{I}_m \in \mathcal{J}_m \neq \mathcal{J}_t$. Each itemset in a sequence is associated with a timestamp $t_1, t_2, \ldots t_p, t_{target}$ and a contextual information. A contextual timestamp (p, t) combines the contextual information p with a timestamp t. Such data can represent care trajectories and the final diagnoses of patients, physico-chemical measures followed by a global biological assessment of the water quality of a river [19], or football player coaching as in our running example. Itemsets are linked by the temporal relation "*is preceded by*" since the general aim is to explain the evaluation at the end of the sequence by previous observations.



(a) $\mathcal{L}_{\texttt{KPE}}$ (main lattice) (b) $\mathcal{L}_{\texttt{KFD}}$ (c) $\mathcal{L}_{\texttt{KTS}}$ (temporal lattice)

Figure 6: (a) $\mathcal{L}_{\texttt{KPE}}$ – lattice of player evaluations; (b) $\mathcal{L}_{\texttt{KFD}}$ – lattice of football drills; (c) $\mathcal{L}_{\texttt{KTS}}$ – lattice of training sessions (obtained by applying RCA to Tab. 2)

5.1. RCA on Sequential Data: Input and Result Structure

Based on the data model shown in Fig. 5, we transform the database within a relational context family with four object-attribute contexts and at least four object-object contexts. The object-attribute contexts rely on four sets of objects: G_m is the set of all contextual timestamps (p, t_{target}), representing the sequence targets. G_t is the set of all contextual timestamps (p, t) representing the observations within sequences, G_i is the set of all items from singletons of $\mathcal{J}_t$, and G_e is the set of all items from singletons of $\mathcal{J}_m$. $K_m = (G_m, M_m, I_m)$ and $K_t = (G_t, M_t, I_t)$ where $M_m = M_t = \emptyset$, while $K_i = (G_i, M_i, I_i)$ and $K_e = (G_e, M_e, I_e)$ where $M_i = G_i$ (resp. $M_e = G_e$) and I_i (resp. I_e) is the identity relation.

The object-object contexts are as follows: a first temporal relation *is preceded by*, denoted by $ipb_1 \subseteq G_m \times G_t$, that defines temporal links between target contextual timestamps (of G_m) and observation contextual timestamps; a second temporal relation *is preceded by*, denoted by $ipb_2 \subseteq G_t \times G_t$, that

defines temporal links between contextual timestamps of G_t; a qualitative relations *has item*, denoted by $hi \subseteq G_t \times G_i$, that defines itemsets in $\mathcal{J}_t$; a qualitative relation *has evaluation*, $he \subseteq G_m \times G_e$, that defines itemsets in $\mathcal{J}_m$. If items are annotated with quality levels, the hi (resp. he) relation can be repeated into various hi_q relations (resp. he_q).

Example: Table 2 represents the relational context family of the running example. Object-attribute contexts `KPE` and `KTS` correspond respectively to K_m and K_t. `KFD` corresponds to K_i. K_e (and he) are not represented since there is only one item in G_e (i.e. `STARTER`). Object-object context `RPE-ipb-TS` corresponds to ipb_1 while `RTS-ipb-TS` corresponds to ipb_2. There is two (hi_q) relations, `RgFD` and `ReFD`, according to the quality level (good or excellent) associated to the items.

The RCA result comprises four lattices, one for each set of objects, as follows: a *main lattice* $\mathcal{L}_{K_m}$, a *temporal lattice* $\mathcal{L}_{K_t}$, a *lattice of observation items* $\mathcal{L}_{K_i}$, and a *lattice of target items* $\mathcal{L}_{K_e}$. The corresponding lattices of the running example are represented respectively in Fig. 6(a) ($\mathcal{L}_{\texttt{KPE}}$), 6(c) ($\mathcal{L}_{\texttt{KTS}}$), 6(b) ($\mathcal{L}_{\texttt{KFD}}$). The lattice of squad roles is unnecessary here.

The structure of resulting lattices is described in the following. Firstly, the main lattice $\mathcal{L}_{K_m} = (\mathcal{C}_{K_m}, \preceq_{K_m})$ has concepts $C_m = (X_m, Y_m)$ such that:

- **the intent** Y_m contains temporal relational attributes of the form $\exists ipb_1(C_t)$, where $C_t \in \mathcal{L}_{K_t}$ describes objects from $ran(ipb_1) = G_t$;
  e.g. $\texttt{CKPE\_4} \in \mathcal{L}_{\texttt{KPE}}$ has attribute $\exists\texttt{RPE-ipb-TS}(\texttt{CKTS\_3})$ where $\texttt{CKTS\_3} \in \mathcal{L}_{\texttt{KTS}}$ (see Fig. 6).
- **the extent** X_m gathers all objects in G_m that respect the temporal order with at least one G_t object pointed by temporal relational attributes of Y_m;
  e.g. $\texttt{CKPE\_4} \in \mathcal{L}_{\texttt{KPE}}$ has an object $(\texttt{P1}, \texttt{28/09})$ preceded by the object $(\texttt{P1}, \texttt{27/09})$ of `CKTS_3`.

Secondly, the lattice of observation items $\mathcal{L}_{K_i} = (\mathcal{C}_{K_i}, \preceq_{K_i})$ is such that $\top(\mathcal{L}_{K_i})$ extent contains all the items, while the other concept extents contain only one item (see Fig. 6(b) for $\mathcal{L}_{\texttt{KFD}}$ as an example). The lattice $\mathcal{L}_{K_e} = (\mathcal{C}_{K_e}, \preceq_{K_e})$ has the same characteristics. Finally, the temporal lattice $\mathcal{L}_{K_t} = (\mathcal{C}_{K_t}, \preceq_{K_t})$ contains temporal concept $C_t = (X_t, Y_t)$ such that:

- **the intent** Y_t contains two types of relational attributes: temporal attributes of the form $\exists ipb_2(C'_t)$, where $C'_t \in \mathcal{L}_{K_t}$ describes objects from

$ran(ipb_2) = G_t$; qualitative relational attributes of the form $\exists hi(C_i)$, where $C_i \in \mathcal{L}_{K_i}$ describes objects from $ran(hi) = G_i$;
e.g. $\texttt{CKTS\_10} \in \mathcal{L}_{\texttt{KTS}}$ has attribute $\exists\texttt{RTS-ipb-TS}(\texttt{CKTS\_5})$ and inherits attribute $\exists\texttt{RgFD}(\texttt{CKFD\_2})$ (see Fig. 6).

– **the extent** X_t gathers all observation timestamps in G_t associated with the items revealed by qualitative relational attributes of Y_t, and that respect the temporal order with G_t objects pointed by temporal relational attributes of Y_t.
e.g. $\texttt{CKTS\_10} \in \mathcal{L}_{\texttt{KTS}}$ has two objects $(\texttt{P1}, \texttt{27/09})$ and $(\texttt{P1}, \texttt{26/09})$ which are both preceded by the object $(\texttt{P1}, \texttt{25/09})$ of $\texttt{CKTS\_5}$; furthermore, both are associated with the itemset $\{\texttt{PASSING}_{\texttt{good}}\}$ (see Tab. 1).

5.2. Properties of the RCA Result

We recall here some useful properties of the RCA result (proofs are in [8]), which rely on its aforementioned structure, and are used to help the extraction step of CPO-patterns. Briefly, sequential patterns that coexist in the same sequences in $\mathcal{D}_\mathcal{S}$ are revealed by navigating interrelated concept intents.

Property 1. Each temporal relational attribute of a main concept intent allows to extract at least one sequential pattern. In contrast, if there is no temporal relational attribute in a main concept intent, this concept represents no sequential pattern.

Suppose, $C_m \in \mathcal{C}_{K_m}$ is a main concept, and $\exists ipb_1(C_t)$ is a temporal relational attribute of its intent, where $C_t \in \mathcal{C}_{K_t}$. If C_t intent contains a qualitative relational attribute $\exists hi(C_i)$, where $C_i \in \mathcal{C}_{K_i}$, then C_t reveals an itemset of *qualitative values*; if C_t concept intent contains a temporal relational attribute $\exists ipb_2(C'_t)$, then C_t leads to another itemset in the sequential pattern, depending on C'_t intent. Therefore, the order on itemsets in the sequential pattern is revealed by temporal relational attributes. If a navigated concept intent contains no temporal relational attribute, then the extraction of the sequential pattern is finished.

Example (Fig. 7): $\texttt{CKPE\_4}$ has attribute $\exists\texttt{RPE-ipb-TS}(\texttt{CKTS\_4})$ whereas $\texttt{CKTS\_4} \in \mathcal{L}_{\texttt{KTS}}$ has attribute $\exists\texttt{RgFD}(\texttt{CKFD\_2})$ revealing the itemset $\texttt{PASSING}_{\texttt{good}}$ (see Fig. 6); furthermore $\texttt{CKTS\_4}$ has attribute $\exists\texttt{RTS-ipb-TS}(\texttt{CKTS\_7})$ leading to $\texttt{CKTS\_7}$; on the contrary $\texttt{CKTS\_5}$ has only qualitative relational attributes.

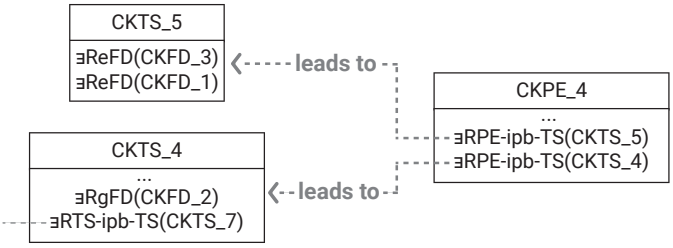


Figure 7: Two navigation paths beginning with `CKPE_4` main concept intent (cf. Fig. 6(a))

Property 2. Let $C_m = (X_m, Y_m) \in \mathcal{C}_{K_m}$ be a main concept whose intent Y_m contains at least one temporal relational attribute. Then C_m can be associated with a CPO-pattern $\mathcal{G}_m$ that summarises the set of sequential patterns derived from Y_m. The support of $\mathcal{G}_m$ is $|X_m|$.

Property 3. The set of CPO-patterns associated with $\mathcal{L}_{K_m}$ main lattice is ordered according to the inclusion on extents. This order corresponds to the subsumption on graphs $\preceq_g$ (see Sect. 2.1).

The following properties 4 and 5 are useful to remove the redundancy occurring when all relational attributes of interrelated concept intents are navigated. Basically, these two properties help us to obtain directly the minimal representations of the extracted CPO-patterns by considering only the relational attributes pointing to the most specific concepts, and by pruning temporal relational attributes that can be deduced by transitivity.

Property 4. Let $C_1 = (X_1, Y_1)$ and $C_2 = (X_2, Y_2)$ be two concepts from the same lattice $\mathcal{L}_K$ such that $C_1 \preceq_K C_2$. Let $C = (X, Y)$ be a concept whose intent has two relational attributes $\exists r(C_1)$ and $\exists r(C_2)$ (derived from the same relation r). Then $\exists r(C_1) \rightarrow \exists r(C_2)$.

Hence, the relational attributes are ordered according to the concepts they point at, and $\exists r(C_2)$ is redundant in the interpretation of C.

Property 5. Let *ipb* be a temporal relation. Let $C = (X, Y)$, $C_1 = (X_1, Y_1)$ and $C_2 = (X_2, Y_2)$ be three concepts such that $\{\exists ipb(C_1), \exists ipb(C_2)\} \subseteq Y$, and $\exists ipb(C_2) \in Y_1$. Then $\exists ipb(C_2) \in Y$ can be deduced from $\exists ipb(C_1) \in Y$.

6. Extracting Multilevel CPO-Patterns from the RCA Result

In this section we introduce an algorithm that directly generates a hierarchy of multilevel CPO-patterns. We provide a complexity analysis and show how the obtained hierarchy can be analysed and navigated with the help of our running example.

6.1. CPOHrchy Algorithm: from the RCA Result to a Hierarchy of CPO-Patterns

Since our objective is to directly obtain organised CPO-patterns, and, besides, since there is a generalisation order on the concepts, we propose to use a 3-tuple structure $P_m = (X_m, Y_m, \mathcal{G}_m)$ derived from a main concept $C_m = (X_m, Y_m)$. $\mathcal{G}_m$ is the CPO-pattern associated with C_m, and it is built from C_m and a list of nested linked concepts $C_t = (X_t, Y_t)$; from each Y_t is derived a vertex v_t.

Algorithm 1, referred to as CPOHrchy, takes as input the three lattices $\mathcal{L}_{K_m}$, $\mathcal{L}_{K_t}$ and $\mathcal{L}_{K_i}$, and its output is a lattice $\mathcal{L}^*_{K_m}$ of P_m structures – i.e. the main concepts of $\mathcal{L}_{K_m}$ are extended with the corresponding CPO-patterns. The three lattices are represented as sets of concepts, where for each concept its upper covers are known[2].

For each main concept C_m, whose intent has at least one temporal relational attribute, a list of adjacent concepts is built in a breadth-first manner based on Properties 4 and 5. The adjacent concepts are further navigated relying on the temporal relational attributes from their intents. For each navigated concept is derived a vertex labelled with an itemset (detailed below). $\perp(\mathcal{L}_{K_m})$ is not taken into consideration, since this concept is too specific and not frequent.

Algorithm 2, called SearchAdjacentConcepts, shows how to derive from temporal relational attributes of C_m intent the next concepts $\mathcal{C}_{next}$ that should be navigated by relying on Properties 4 and 5, i.e. the concepts linked to C_m by the temporal relational attributes of its intent. This algorithm is applicable to temporal concepts (in this case ipb_1 is replaced with ipb_2) as well. **Lines [2-8]:** delete all concepts in $\mathcal{C}_{next}$ that are upper covers for other concepts in $\mathcal{C}_{next}$, i.e. delete concepts that are not the most specific

[2]A concept C is an upper cover (or upper neighbour) of a concept C_1 in a lattice $\mathcal{L}_K = (\mathcal{C}_K, \preceq_K)$, if $C_1 \prec_K C$ and there is no $C_2 \in \mathcal{C}_K$ such that $C_1 \prec_K C_2 \prec_K C$. This is denoted by $C \rhd_K C_1$ [20].

ones in $\mathcal{C}_{next}$. **Lines [9-15]:** prune all concepts in $\mathcal{C}_{next}$ that can be deduced by navigating other ones in $\mathcal{C}_{next}$.

Basically, CPOHrchy algorithm stems from our RCA-based approach introduced in [8] in which we focused on extracting directly sequential patterns, and then on converting them to CPO-patterns using the merging and pruning methods presented in [14]. However, CPOHrchy extracts directly CPO-patterns. To improve its efficiency we use two optimisations:

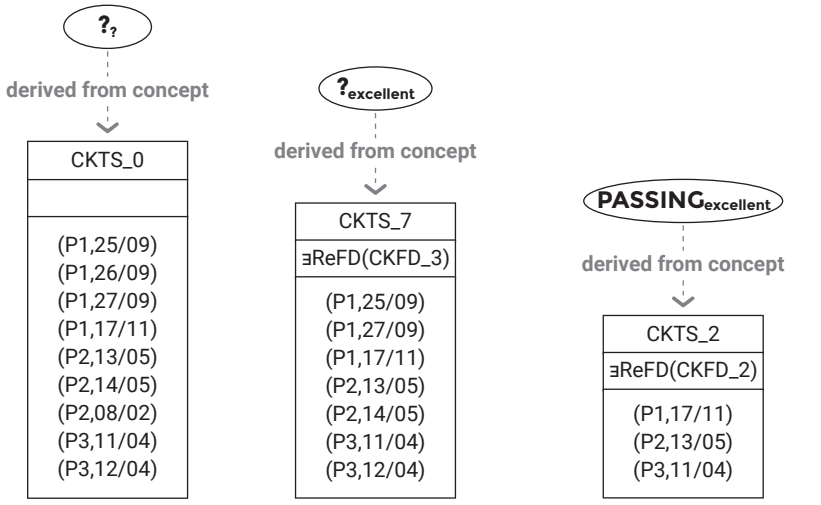
1. since a temporal concept $C_t = (X_t, Y_t)$ can be navigated several times for distinct CPO-patterns, we process C_t only at its first navigation, i.e. SearchAdjacentConcepts is applied only once and its result is saved for later use; similarly, v_t is computed and saved together with C_t;
2. since a CPO-pattern $\mathcal{G}_m$ associated with a main concept $C_m = (X_m, Y_m)$ is discovered if $Support(\mathcal{G}_m) = |X_m| \geq \theta$ (remember that θ is a user-specified minimum support for the main lattice), then all navigated temporal concepts $C_t = (X_t, Y_t)$ should have $|X_t| \geq |X_m|$. Therefore, we diminish the navigation space by defining a minimum support $\theta' = \theta$ for the temporal lattice as well.

The labelling of vertices was described in [8]. We briefly recall this step. To convert a navigated concept to a vertex labelled with an itemset, all qualitative relational attributes of this concept intent are analysed. Property 4 is applied again to consider, for the same qualitative relation, only the most specific concepts used to build the corresponding qualitative relational attributes in the concept intent. A qualitative relational attribute $\exists r(C_i)$, where C_i is a concept in the $\mathcal{L}_{K_i}$ lattice of items, can be *vague* if $C_i \equiv \top(\mathcal{L}_{K_i})$, respectively it can be *defined* if $C_i \prec_{K_i} \top(\mathcal{L}_{K_i})$.

Based on the order on items given by lattice $\mathcal{L}_{K_i}$ and on the aforementioned types of qualitative relational attributes, there are three types of items that can be derived, precisely concrete qualitative, abstract qualitative and abstract. A *concrete qualitative item*, denoted by "$item_q$", is derived from a defined qualitative relational attribute $\exists hi_q(C_i)$, with $extent(C_i) = \{item\}$ and q the item quality. An *abstract qualitative item*, denoted by "$?_q$", is derived from a vague qualitative relational attribute $\exists hi_q(\top(\mathcal{L}_{K_i}))$. An *abstract item*, denoted by "$?_?$", is obtained when the concept intent has no qualitative relational attribute.

For instance, Fig. 8(a) depicts a vertex having an abstract item that characterises all training sessions from Tab. 1 since all of them are described

by at least one football drill rated either as good or excellent. Figure 8(b) shows a vertex having an abstract qualitative item characteristic to all training sessions from extent CKTS_7 that are described by different football drills, but all rated as excellent. Figure 8(c) illustrates a vertex having a concrete qualitative item characteristic to all training sessions from extent CKTS_2 that are described by a specific football drill, namely passing, rated as excellent.



(a) abstract (b) abstract qualitative (c) concrete qualitative

Figure 8: Deriving vertices from concepts

Depending on the number of concrete and/or abstract items in a CPO-pattern (except for the target itemset, in this case), we are able to directly extract multilevel CPO-patterns, namely abstract, hybrid and concrete. An *abstract CPO-pattern* contains only abstract items, and reveals an imprecise common regularity of the analysed sequences. A *hybrid CPO-pattern* contains both abstract and concrete items, and discloses a "more or less" accurate common regularity of the analysed sequences. A *concrete CPO-pattern* contains only concrete items, and reveals an accurate common regularity of the analysed sequences.

6.2. Complexity Analysis

We present a time complexity analysis of the RCA-Seq approach that is compared with the time complexity of [14] and [8].

We first consider the method of CPO-pattern extraction described in [14]. In the following $\mathcal{I}$ is a set of items, $\mathcal{D}_{\mathcal{S}}$ is a sequence dataset built on itemsets

from $\mathcal{I}$, and l is the maximum length of the sequences in $\mathcal{D}_{\mathcal{S}}$. Let us denote by m the number of obtained CPO-patterns. [14] spans two steps, and the overall complexity in the worst-case scenario is $O(m \cdot 2 \cdot (2 \cdot |\mathcal{I}|)^l)$. If we build a hierarchy of these results in a post-processing step, and since the CPO-patterns are closed and already associated with their supporting sequences, the complexity of the added step would be 1) building the context patterns-sequences: $O(m \cdot |\mathcal{D}_{\mathcal{S}}|)$ and 2) building the lattice[3]: $O(m^2 \cdot |\mathcal{D}_{\mathcal{S}}| \cdot (m+2))$. Thus the whole [14] process complexity would be $O(m \cdot 2 \cdot (2 \cdot |\mathcal{I}|)^l + m^3 \cdot |\mathcal{D}_{\mathcal{S}}|) = O_1$.

We now consider the current RCA-Seq approach that relies on two algorithms, namely Multi-FCA (RCA process, [2]) and CPOHrchy. We focus on the worst-case scenario. Following our data model (Fig. 5), let us consider an RCA input that comprises the set of object-attribute contexts $\{K_m = (G_m, M_m, I_m), K_t = (G_t, M_t, I_t), K_i = (G_i, M_i, I_i)\}$ and the associated object-object contexts. At the end of RCA process, the obtained RCA result contains the set of concept lattices $\{\mathcal{L}_{K_m}, \mathcal{L}_{K_t}, \mathcal{L}_{K_i}\}$ built from the extended object-attribute contexts $\{K_m^+ = (G_m, M_m^+, I_m^+), K_t^+ = (G_t, M_t^+, I_t^+), K_i = (G_i, M_i, I_i)\}$. We denote by $|\mathcal{L}_{K_i}|$ the number of formal concepts of $\mathcal{L}_{K_i}$. According to [2], in the worst-case scenario the overall computation time of the considered RCA result is $O(n_c \cdot n_o \cdot (n_a + n_o))$, where $n_c = max(|\mathcal{L}_{K_m}|, |\mathcal{L}_{K_t}|, |\mathcal{L}_{K_i}|)$, $n_a = max(|M_m^+|, |M_t^+|, |M_i|)$, and $n_o = max(|G_m|, |G_t|, |G_i|)$. The completeness and correctness of RCA process is discussed in [21].

The worst-case scenario for CPOHrchy algorithm is when each main/temporal concept points to all concepts in $\mathcal{L}_{K_t}$. Let us denote $m = |\mathcal{L}_{K_m}|$ (each element of $\mathcal{L}_{K_m}$ reveals a CPO-pattern), $p = |\mathcal{L}_{K_t}|$, and q the number of all qualitative relational attributes from a temporal concept intent. First, we focus on Algorithm 2. The overall computation time is $O(p)$ since we iterate throughout p concepts pointed by the temporal relational attributes of Y_m at **Lines [1, 4, 7, 11, 14]**. The other lines are $O(1)$. Second, in Algorithm 1, **Lines [3–24]** are executed m times. **Lines [5, 7]** have the complexity $O(p)$ since $\mathcal{C}_{next}$ contains p concepts pointed by Y_m temporal relational attributes. **Lines [8–21]** are executed p times since each temporal concept of $\mathcal{L}_{K_t}$ is visited only once and the complexity of these lines is $O(p(q+p))$. Indeed, **Lines [12, 14, 16]** are $O(p)$ since $\mathcal{C}'_{next}$ has p concepts pointed by

[3]the complexity of building a lattice $\mathcal{L}$ from a context (G, M, I) is $O(|G|^2 \cdot |M| \cdot |\mathcal{L}|)$ [18]

Y_t temporal relational attributes; **Line [10]** is $O(q)$; and the other lines are $O(1)$. Since generally $q \leq p$, the computation time becomes $O(p^2)$. Therefore, in the worst-case scenario the overall computation time for CPOHrchy is $O(m \cdot p^2)$.

To sum up, the overall time complexity of RCA-Seq is $O(n_c \cdot n_o \cdot (n_a + n_o) + m \cdot p^2) = O_2$. To compare with the aforementioned complexity O_1, we consider that the sizes of $\mathcal{I}$ and $\mathcal{D}_\mathcal{S}$ – which correspond to sets of objects – are smaller than n_o, while m and p are smaller than n_c. Then, O_1 is upper bounded by $O(n_c \cdot 2 \cdot (2 \cdot n_o)^l + n_c^3 \cdot n_o)$, while O_2 is upper bounded by $O(n_c \cdot n_o \cdot (n_a + n_o) + n_c^3)$. Finally, since l is generally greater than 3, the complexity of RCA-Seq is better than the complexity of the approach described in [14] combined with a lattice-building step.

Regarding the previous approach [8], it employs the merging and pruning step explained in [14] that has in the worst-case scenario the complexity $O(m \cdot (2 \cdot |\mathcal{I}|)^l)$, where the set of items $\mathcal{I}$ corresponds to all the qualitative relational attributes used to derive the items from the discovered patterns. Thus, the complexity of [8] extraction step is $O(m \cdot (p^2 + (2 \cdot |\mathcal{I}|)^l))$.

6.3. Application to the Running Example: Extraction and Navigation of a Hierarchy of Multilevel CPO-Patterns

To illustrate RCA-Seq processing, let us consider that we want to extract the CPO-pattern associated with the main concept `CKPE_7` from the lattice of player evaluations $\mathcal{L}_{\texttt{KPE}}$ (Fig. 6(a)). Following Fig. 9, from right to left, we start by examining all temporal relational attributes from intent `CKPE_7` that are ordered according to the generalisation order $\preceq_{\texttt{KTS}}$ on the concepts used to build them. Since there is only one most specific concept, the next navigated concept is `CKTS_8` from the lattice of training sessions $\mathcal{L}_{\texttt{KTS}}$.

By analysing the concepts used to build all temporal relational attributes from intent `CKTS_8`, we notice that there are two most specific concepts `CKTS_10` and `CKTS_5`. Moreover, `CKTS_10` intent contains the ∃`RTS-ipb-TS`(`CKTS_5`) temporal relational attribute that points to `CKTS_5`, and thus the next navigated concept is only `CKTS_10` since `CKTS_5` generates redundant information. The intent of `CKTS_10` consists in three temporal relational attributes and the most specific concept used to build them is `CKTS_5`. The intent of `CKTS_5` shown in $\mathcal{L}_{\texttt{KTS}}$ (Fig. 6(c)) has no temporal relational attribute, and thus the navigation is finished. In Fig. 9, ① represents the set of navigated concepts that should be converted into vertices.

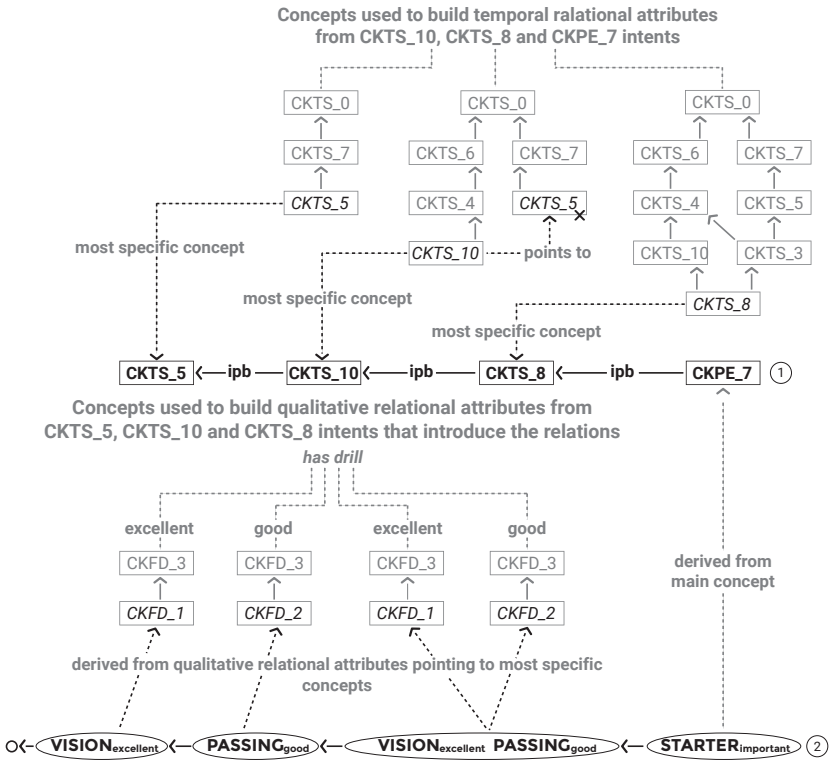


Figure 9: Extracting the CPO-pattern associated with the CKPE_7 main concept. (1) is the set of navigated concepts and (2) is the generated CPO-pattern

To this end, we analyse the qualitative relational attributes from the navigated concept intents that enable us to extract the CPO-pattern (2), denoted by $\mathcal{G}_{\texttt{CKPE\_7}}$. From right to left, the vertex labelled with 1-itemset $\{\texttt{STARTER}_{\texttt{important}}\}$ contains the default concrete qualitative item associated with CKPE_7 intent. The intent of concept CKTS_8 contains four qualitative relational attributes, $\exists\texttt{ReFD}(\texttt{CKFD\_3})$, $\exists\texttt{ReFD}(\texttt{CKFD\_1})$, $\exists\texttt{RgFD}(\texttt{CKFD\_3})$, and $\exists\texttt{RgFD}(\texttt{CKFD\_2})$, that highlight two qualitative relations *has drill excellent* and *has drill good*. The concrete qualitative item $\texttt{VISION}_{\texttt{excellent}}$ is then derived from the most specific concept used to highlight the *has drill excellent* relation, and $\texttt{PASSING}_{\texttt{good}}$ is derived from the most specific concept used to highlight the *has drill good* relation. Therefore, the itemset $\{\texttt{VISION}_{\texttt{excellent}}, \texttt{PASSING}_{\texttt{good}}\}$ is the label of the vertex derived from concept CKTS_8. Similarly, the vertex labelled with itemset $\{\texttt{PASSING}_{\texttt{good}}\}$ consists in only one concrete qualitative item derived from $\exists\texttt{RgFD}(\texttt{CKFD\_2})$ qualitative

relational attribute of CKTS_10 intent; finally the vertex labelled with itemset $\{\texttt{VISION}_{\texttt{excellent}}\}$ consists in only one concrete qualitative item derived from the $\exists\texttt{ReFD}(\texttt{CKFD\_1})$ qualitative relational attribute of intent CKTS_5. The obtained CPO-pattern can be interpreted as follows: a midfielder should be assigned as important starter when prior to the upcoming match he has an excellent vision and at least a good level of passing.

RCA-Seq results in a hierarchy of multilevel CPO-patterns as shown in Fig. 10 for our running example. CPO-pattern (a) is an abstract one; (c) and (d) are hybrid CPO-patterns; (b), (e), (f) and (g) are concrete ones.

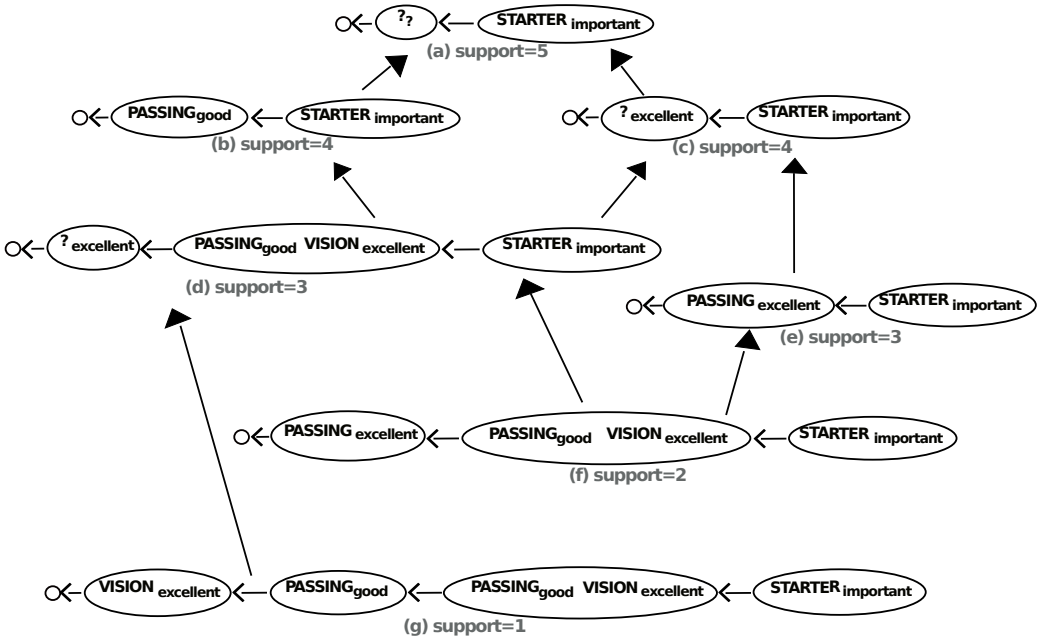


Figure 10: The hierarchy of multilevel CPO-patterns obtained by exploring with RCA-Seq the illustrative example given in Tab. 1

In short, the hierarchy emphasises two benefits of the RCA-Seq result. Firstly, the generalisation order regarding the structure of CPO-patterns. For example, the structure of (d) is more specific than the one of its ancestor CPO-patterns. Secondly, the partial order on items and the inclusion order on itemsets. For instance, (e) reveals the $\{\texttt{PASSING}_{\texttt{excellent}}\} \leftarrow \{\texttt{STARTER}_{\texttt{important}}\}$ regularity – i.e. a midfielder is selected as starter if before the upcoming match he showcases excellent passing skills – that is an accurate specialisation of the less accurate regularity $\{?_{\texttt{excellent}}\} \leftarrow \{\texttt{STARTER}_{\texttt{important}}\}$

revealed by (c) – i.e. a midfielder is selected as starter if he showcases excellent skills. Similarly, (d) discloses the $\{\texttt{PASSING}_{\texttt{good}}, \texttt{VISION}_{\texttt{excellent}}\} \leftarrow \{\texttt{STARTER}_{\texttt{important}}\}$ regularity that is a specialisation of the $\{\texttt{PASSING}_{\texttt{good}}\} \leftarrow \{\texttt{STARTER}_{\texttt{important}}\}$ one revealed by (b). In addition, the hybrid CPO-pattern (d) with $Support = 3$ can be uncovered when $\theta = 3$ even if its accurate specialisation (f) is not frequent, and thus is not extracted. Therefore, relying on these benefits, we can say that the obtained hierarchy provides a quick way to navigate to interesting CPO-patterns, and, besides it can help experts (e.g. football coaches) to better understand the analysed data.

Accordingly, a coach can navigate the hierarchy given in Fig. 10 based on the employed football style. Precisely, CPO-pattern (b) emphasises the counter-attacking style, i.e. when a midfielder possesses the ball, he is reactive and immediately tries to pass the ball to the most advanced teammate. In contrast, CPO-pattern (c) highlights the possession style, i.e. midfielders have to maintain possession throughout the game by using accurate passes and/or excellent visual acuity. As a result, the coach might navigate a different number of CPO-patterns, namely 3 descendants of (b) or 4 of (c). Furthermore, the (f) and (g) CPO-patterns can assist the coach in deciding particular positions of midfielders on the pitch. Hence, firstly, (g) might represent a defensive midfielder that among other skills needs to possess good passing skills to hold the ball under sustained pressure. Secondly, (f) might represent an attacking midfielder that has to possess superior technical abilities in terms of passing and vision to deliver passes to strikers. Following the same principles, the coach can continue the navigation being guided by the relationships between the multilevel CPO-patterns.

7. Comparing and Extending the RCA-Seq Approach

In this section, we want to compare RCA-Seq with another FCA-based method for extracting CPO-patterns. Furthermore, we want to illustrate based on the running example how to adapt our approach to generate: first, CPO-patterns that contain items from different levels of a user-defined taxonomy; second, CPO-patterns that have user-specified constraints on the order relations on itemsets.

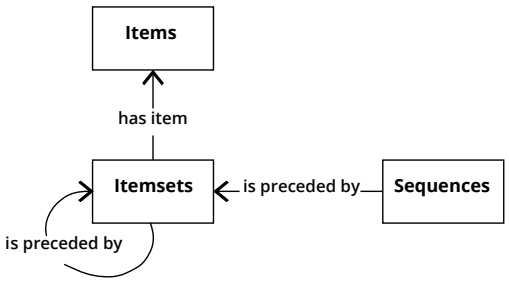
7.1. Hierarchies of Multilevel CPO-Patterns vs Sequence Pattern Concept Lattices

Here, we compare the results obtained by using two distinct approaches for analysing sequential data, namely our RCA-Seq and the one introduced in [16], which relies on pattern structures. This comparison is relevant since [16] also proposes to represent the closed subsequences that coexist in one or more sequences as a directed acyclic graph of alignments, i.e. a CPO-pattern.

We use the small sequence database $\mathcal{D}_{\mathcal{S}}$ shown in Tab. 11(a), which is taken from [16]. No threshold for the support measure is employed. Firstly, for the sake of brevity we do not explain how to apply pattern structures to $\mathcal{D}_{\mathcal{S}}$ and we rely on the generated sequence pattern concept lattice given in [16]. Secondly, we apply RCA-Seq to $\mathcal{D}_{\mathcal{S}}$ by using the data model depicted in Fig. 11(b) and by following the steps presented in Sect. 4 and 6. In short, the three rectangles in Fig. 11(b) represent: the set of sequence-building items $\{a, b, c, d, e, f, g\}$, the set of the identifiers of sequence-building itemsets and the set of the analysed sequences $\{\alpha^1, \alpha^2, \alpha^3, \alpha^4\}$.

Seq Id	Sequence
α^1	$\langle\{a\}\{a,b,c\}\{a,c\}\{d\}\{c,f\}\rangle$
α^2	$\langle\{a,d\}\{c\}\{b,c\}\{a,e\}\rangle$
α^3	$\langle\{e,f\}\{a,b\}\{d,f\}\{c\}\{b\}\rangle$
α^4	$\langle\{e\}\{g\}\{a,f\}\{c\}\{b\}\{c\}\rangle$

(a)

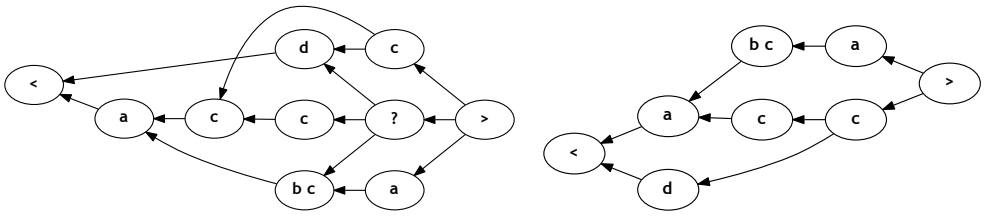


(b)

Figure 11: (a) sequence database $\mathcal{D}_{\mathcal{S}}$ taken from [16]; (b) the modelling of $\mathcal{D}_{\mathcal{S}}$ used by RCA-Seq

Both the hierarchy of multilevel CPO-patterns extracted with RCA-Seq and the sequence pattern concept lattice obtained through [16] contain 15 concepts having the same extents but different intents. For instance, in case of RCA-Seq the concept with extent $\{\alpha^1, \alpha^2\}$ is associated with the multilevel CPO-pattern $\mathcal{G}_1$ shown in Fig. 12(a), while in case of [16] the same concept extent is associated with the classical CPO-pattern $\mathcal{G}_2$ given in Fig. 12(b). It is noted that $\mathcal{G}_1$ summarises 6 closed subsequences that coexist in α^1 and α^2, whereas $\mathcal{G}_2$ summarises only 3 closed subsequences. In addition,

$\mathcal{G}_2 \prec_g \mathcal{G}_1$. Hence, $\mathcal{G}_1$ uncovers the regularities captured by $\mathcal{G}_2$ and besides, it reveals additional regularities common to α^1 and α^2, but to a lesser extent. For example, $\{d\}$ precedes $\{c\}$ in both sequences, but $\{d\}$ also precedes $\{?\}$, which indicates different items in α^1 (e.g. $\{f\}$) and α^2 (e.g. $\{b\}$ or $\{e\}$).



(a) $\mathcal{G}_1$ (b) $\mathcal{G}_2$

Figure 12: (a) multilevel CPO-pattern generated by RCA-Seq; (b) classical CPO-pattern generated by [16]

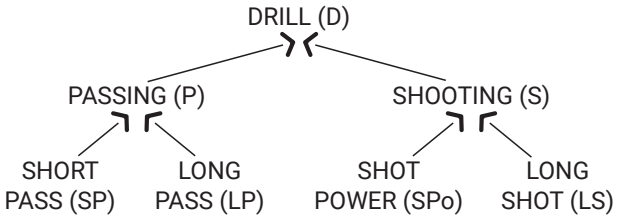
To sum up, on the one hand, RCA-Seq provides CPO-patterns that reveal really detailed overviews of the associated sequences. On the other hand, [16] generates CPO-patterns that uncover only broad overviews of the associated sequences. Furthermore, the lattice of CPO-patterns is directly built by RCA-Seq, while the approach described in [16] first extracts CPO-patterns, and then builds the lattice.

7.2. Usage of RCA-Seq *with a Taxonomy*

RCA-Seq reveals a taxonomy over the items due to the nominal scaling used to encode the set of items into the RCA input. This taxonomy has only two levels: first, the level comprising each specific item, and second, the level with the general item, i.e. the item that represents the set of items used to build the analysed sequences. Accordingly, the extracted multilevel CPO-patterns contain only items from these two levels. Srikant et al. in [15] propose to use a user-defined taxonomy over the items, in order to generate sequential patterns including items across different levels. To this end, they preprocess each sequence from the database to obtain an "extended-sequence", i.e. the ancestors (from the taxonomy) of each item are added in the sequence. Thus, their algorithm GSP explores sequences that already contain the relationships between the items and their ancestors.

In contrast, RCA-Seq can easily integrate a user-defined taxonomy in the RCA input, and, besides extracts directly organised CPO-patterns that contain items from different levels of the taxonomy without preprocessing the

analysed sequences. To illustrate this, we consider the user-defined taxonomy over football drills depicted in Fig. 13(a), and the football sequence database $\mathcal{D}_{\mathcal{S}1}$ given in Fig. 13(b). It is noted that our training sessions contain only one or more specific drills, such as `LONG PASS` (LP), `SHORT PASS` (SP), `SHOT POWER` (SPo) and `LONG SHOT` (LS).



(a) taxonomy over football drills

Player	Date	Training Session				Evaluation
		LP	SP	SPo	LS	STARTER
P1	17/09	excellent	–	good	–	–
	18/09	–	–	–	good	–
	19/09	–	excellent	–	–	–
	20/09	–	–	–	–	important
P2	15/05	–	excellent	good	–	–
	16/05	good	excellent	–	–	–
	17/05	–	–	–	–	important
P3	08/04	excellent	excellent	–	–	–
	09/04	–	–	–	–	important

(b) $\mathcal{D}_{\mathcal{S}1}$

Figure 13: (a) a user-defined taxonomy over football drills; (b) a football sequence database $\mathcal{D}_{\mathcal{S}1}$

To explore $\mathcal{D}_{\mathcal{S}1}$ sequences by using the taxonomy over football drills, we follow the steps presented in Sect. 4.2. The only difference is the mixture of ordinal and nominal scaling [1], instead of only nominal scaling, used to build the context of football drills given in Fig. 14(a). From this context is obtained the lattice of football drills $\mathcal{L}_{\texttt{KFD}}$, shown in Fig. 14(c). Note that qualitative object-object contexts encode strictly the relations between training sessions and specific drills as given in $\mathcal{D}_{\mathcal{S}1}$. For instance, we encode in Fig. 14(b) that on September 18$^{\text{th}}$ the long shot skill of midfielder P1 was good, and no extra information regarding the ancestors of long shot.

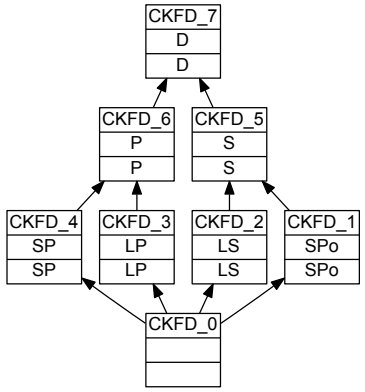
From the obtained RCA result, e.g. the CPO-pattern depicted in Fig. 15 is extracted as explained in Sec. 6. This CPO-pattern contains items from

KFD	D	P	S	LP	SP	SPo	LS
D	×						
P	×	×					
S	×		×				
LP	×	×		×			
SP	×	×			×		
SPo	×		×			×	
LS	×		×				×

(a) KFD

RgFD	D	P	S	LP	SP	SPo	LS
(P1,17/09)						×	
(P1,18/09)							×
(P1,19/09)							
(P2,15/05)						×	
(P2,16/05)				×			
(P3,08/04)							

(b) RgFD



(c) $\mathcal{L}_{\texttt{KFD}}$

Figure 14: (a) and (b) are respectively the object-attribute and the object-object contexts from the RCF built for $\mathcal{D}_{\mathcal{S}1}$ dataset (as explained in Sect. 4.2); (c) the lattice of football drills (items) built from KFD shown in (a)

different levels of the taxonomy of football drills. Indeed, during the relational scaling step, RCA reveals the relationships between training sessions and football drills across different levels of the taxonomy. The conversion of navigated concepts into vertices relies on the simplified lattice of football drills given in Fig. 14(c).

Let us note that our approach with a user-defined taxonomy can be applied to explore sequences that include items across different levels of the taxonomy, as well.

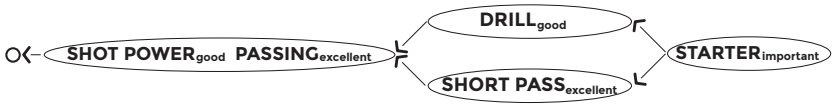


Figure 15: CPO-pattern with items across different levels of Fig. 13(a) taxonomy

7.3. Usage of RCA-Seq *with Constraints on the Order Relations on Itemsets*

In RCA-Seq approach the order on itemsets in a CPO-pattern is revealed by relational attributes from the navigated concept intents; these attributes being built using the existential scaling. For instance, using the running example, in Fig. 16 there is a temporal link between the CKPE_3 main concept and the CKTS_2 temporal concept (revealed by $\exists$RPE-ipb-TS(CKTS_2) in

CKPE_3 intent) since each player evaluation in extent CKPE_3 is preceded by at least one training session in extent CKTS_2. Therefore, the CKPE_3 extent gathers all player evaluations from the analysed data (Tab. 1) that are preceded respectively by at least one training session when the midfielder passing skills were excellent.

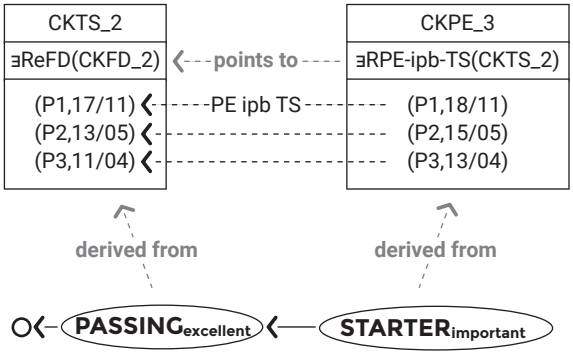


Figure 16: Temporal links between CKPE_3 main concept and CKTS_2 temporal concept from the RCA result shown in Fig. 6; the CPO-pattern is associated with concept CKPE_3

However, coaches can be interested in finding out patterns that are available for midfielders that frequently show certain skills before an upcoming match. For example, coaches can look for player evaluations that are preceded by more than 50% of the associated training sessions for which passing skills were rated excellent. Similarly, such constraints can be used on the temporal relations between training sessions. Using the RCA-SEQ approach, CPO-patterns revealing such situations can be extracted by only changing the quantifier used to introduce relations encoded in temporal object-object contexts during the iterative steps. To add the constraint *"player evaluation is preceded by more than* 50% *of the associated training sessions"*, we use the existential quantifier with a user-specified cardinality, denoted by $\exists_{>n\%}$ [2] where $n = 50$. Formally, a relational attribute $\exists_{>n\%} r(C)$, where r is a relation and $C = (X, Y)$ is a concept whose extent contains objects from $ran(r)$, describes an object $g \in dom(r)$ if $r(g) \cap X \neq \emptyset$ and $|r(g) \cap X| > \frac{n \times |r(g)|}{100}$.

To illustrate this, we apply again RCA to the RCA input depicted in Tab. 2 by changing quantifier $\exists$ to $\exists_{>50\%}$ for the RPE-ipb-TS temporal object-object context (the $\exists$ quantifier is preserved for RTS-ipb-TS). The same RCA result from Fig. 6 is obtained except for the $\mathcal{L}_{\texttt{KPE}}$ main lattice that has the new structure shown in Fig. 17. It is noted that the number of extracted CPO-patterns (main concepts) is smaller since the criterion imposed by coaches is

more restrictive. In addition, the CPO-pattern in Fig. 16 is associated with the CKPE_2 main concept in Fig. 17 lattice, and in this case there is only a player evaluation that has more than 50% of the associated training sessions for which passing skills were excellent.

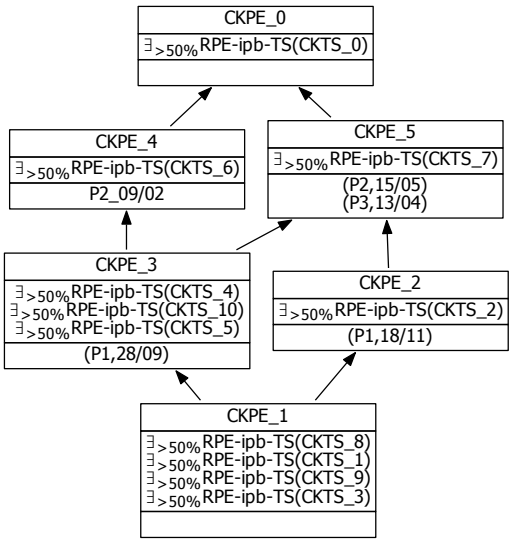


Figure 17: The $\mathcal{L}_{\texttt{KPE}}$ main lattice of player evaluations obtained by scaling the temporal links with training sessions using the $\exists_{>50\%}$ quantifier

Let us note that depending on the motivation behind the analysis, the various quantifiers presented in [2] and their variants can be applied in the same way.

8. Conclusion

In this paper, we have presented an original framework, referred to as RCA-Seq, for helping experts when exploring sequential qualitative data. This framework allows to directly extract hierarchies of multilevel CPO-patterns thanks to the structure and the properties of the RCA result.

The primary aim of our approach is to enhance the analysis of the extracted set of CPO-patterns. To this end, we benefit from the fact that some CPO-patterns are naturally sub-patterns of others, and we propose to extract hierarchies of CPO-patterns where each CPO-pattern is projected into its descendants. Consequently, when an interesting CPO-pattern is found, the analysis can continue by exploring the surrounding area in the hierarchy.

Then, we exploit the order on items revealed by RCA, and we extract multilevel CPO-patterns. Therefore, a global view of the trends of the analysed data is obtained. In addition, we show that RCA-Seq can be easily adapted to extract CPO-patterns with items across different levels of a user-defined taxonomy, and to specify constraints on the order relations on the itemsets of extracted CPO-patterns.

This work opens up interesting research directions. One of them is to improve RCA-Seq to be applicable to large volumes of sequential data (e.g. combining AOC poset [40] with RCA to cope with the "concept explosion" problem). A second direction is to push measures of interest (e.g. stability index [7] or distribution index [19]) into the exploration step to decrease the number of extracted multilevel CPO-patterns. Using or adapting other more sophisticated scores proposed in the FCA framework [41] is also an important extension to the present work. Another future research direction is to study different quantifiers that can be used during the relational scaling mechanism and the types of revealed CPO-patterns [2, 42].

Acknowledgement

We wish to thank Marianne Huchard (LIRMM, Montpellier University) and Xavier Dolques (previously ENGEES Strasbourg) who helped us in the preliminary steps of this work.

[1] B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer, 1999.

[2] M. Rouane-Hacene, M. Huchard, A. Napoli, P. Valtchev, Relational concept analysis: Mining concept lattices from multi-relational data, Annals of Mathematics and Artificial Intelligence 67 (1) (2013) 81–108.

[3] C. De Maio, G. Fenza, M. Gallo, V. Loia, S. Senatore, Formal and relational concept analysis for fuzzy-based automatic semantic annotation, Applied Intelligence 40 (1) (2014) 154–177.

[4] X. Dolques, M. Huchard, C. Nebut, P. Reitz, Fixing Generalization Defects in UML Use Case Diagrams, Fundam. Inform. 115 (4) (2012) 327–356.

[5] L. Shi, Y. Toussaint, A. Napoli, A. Blansché, Mining for reengineering: An application to semantic wikis using formal and relational concept

analysis, in: Proceedings of the 8th Extended Semantic Web Conf., ESWC 2011, Part II, Springer, 2011, pp. 421–435.

[6] X. Dolques, F. Le Ber, M. Huchard, C. Grac, Performance-friendly rules extraction in large water datasets with AOC-posets and relational concept analysis, Int. Journal of General Systems 45 (2) (2016) 187–210.

[7] A. Buzmakov, S. O. Kuznetsov, A. Napoli, Is concept stability a measure for pattern selection?, Procedia Computer Science 31 (2014) 918 – 927.

[8] C. Nica, A. Braud, X. Dolques, M. Huchard, F. Le Ber, Extracting hierarchies of closed partially-ordered patterns using relational concept analysis, in: Proceedings of the 22nd Int. Conf. on Conceptual Structures, ICCS, Springer, 2016, pp. 17–30.

[9] G. Casas-Garriga, Summarizing sequential data with closed partial orders, in: 2005 SIAM Int. Conf. on Data Mining, 2005, pp. 380–391.

[10] R. Agrawal, R. Srikant, Mining sequential patterns, in: Int. Conf. on Data Engineering, 1995, pp. 3–14.

[11] P. Fournier-Viger, J. C. Lin, R. U. Kiran, Y. S. Koh, R. Thomas, A survey of sequential pattern mining, Data Science and Pattern Recognition 1 (1) (2017) 54–77.

[12] A. Gomariz, M. Campos, R. Marin, B. Goethals, Clasp: An efficient algorithm for mining frequent closed sequences, in: Proceedings of the 17th Pacific-Asia Conf., PAKDD, Part I, Springer, 2013, pp. 50–61.

[13] J. Pei, H. Wang, J. Liu, K. Wang, J. Wang, P. S. Yu, Discovering frequent closed partial orders from strings, IEEE Transactions on Knowledge and Data Engineering 18 (11) (2006) 1467–1481.

[14] M. Fabrègue, A. Braud, S. Bringay, F. Le Ber, M. Teisseire, Mining closed partially ordered patterns, a new optimized algorithm, Know.-Based Syst. 79 (2015) 68–79.

[15] R. Srikant, R. Agrawal, Mining sequential patterns: Generalizations and performance improvements, in: Proceedings of the 5th Int. Conf. on Extending Database Technology, EDBT, Springer-Verlag, 1996, pp. 3–17.

[16] V. Codocedo, G. Bosc, M. Kaytoue, J.-F. Boulicaut, A. Napoli, A proposition for sequence mining using pattern structures, in: Proceedings of the 14th Int. Conf. on Formal Concept Analysis, ICFCA, Springer, 2017, pp. 106–121.

[17] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal, Computing iceberg concept lattices with Titanic, Data & Knowledge Engineering 42 (2) (2002) 189–222.

[18] D. van der Merwe, S. Obiedkov, D. Kourie, Addintent: A new incremental algorithm for constructing concept lattices, in: Proceedings of the 2nd Int. Conf. on Formal Concept Analysis, ICFCA, Springer, 2004, pp. 372–385.

[19] C. Nica, A. Braud, X. Dolques, M. Huchard, F. Le Ber, Exploring temporal data using relational concept analysis: An application to hydroecological data, in: Proceedings of the 13th Int. Conf. on Concept Lattices and Their Applications, CLA., CEUR-WS.org, 2016, pp. 299–311.

[20] C. Roth, S. Obiedkov, D. G. Kourie, On succinct representation of knowledge community taxonomies with formal cconcept analysis, Int. Journal of Foundations of Computer Science 19 (02) (2008) 383–404.

[21] M. Rouane-Hacene, M. Huchard, A. Napoli, P. Valtchev, Soundness and completeness of relational concept analysis, in: Proceedings of the 11th Int. Conf. on Formal Concept Analysis, ICFCA, Springer, 2013, pp. 228–243.

[22] K. E. Wolff, Temporal concept analysis, in: Proceedings of the 9th Int. Conf. on Conceptual Structures, ICCS, 2001, pp. 91–107.

[23] S. Ferré, The efficient computation of complete and concise substring scales with suffix trees, in: Proceedings of the 5th Int. Conf. on Formal Concept Analysis, ICFCA, Springer, 2007, pp. 98–113.

[24] A. Buzmakov, E. Egho, N. Jay, S. O. Kuznetsov, A. Napoli, C. Raïssi, On Mining Complex Sequential Data by means of FCA and Pattern Structures, Int. Journal of General Systems 45 (2016) 135–159.

[25] J. Poelmans, P. Elzinga, S. Viaene, G. Dedene, A Method based on Temporal Concept Analysis for Detecting and Profiling Human Trafficking Suspects, in: Artificial Intelligence and Applications, AIA, 2010, pp. 1–9.

[26] B. Ganter, S. O. Kuznetsov, Pattern structures and their projections, in: Proceedings of the 9th Int. Conf. on Conceptual Structures, ICCS, 2001, pp. 129–142.

[27] D. Gizdatullin, D. I. Ignatov, E. Mitrafanova, A. Muratova, Classification of demographic sequence basd on pattern structures and emerging patterns, in: Supplementary Proceedings of ICFCA, 2017, pp. 49–66.

[28] H. Cheng, X. Yan, J. Han, C. Hsu, Discriminative frequent pattern analysis for effective classification, in: Int. Conf. on Data Engineering, 2007, pp. 716–725.

[29] M. Wang, X. Shang, Z. Li, Sequential pattern mining for protein function prediction, in: Advanced Data Mining and Applications, ADMA, Springer, 2008, pp. 652–658.

[30] J. Chen, An updown directed acyclic graph approach for sequential pattern mining, IEEE Transactions on Knowledge and Data Engineering 22 (7) (2010) 913–928.

[31] P. Fournier-Viger, A. Gomariz, M. Campos, R. Thomas, Fast vertical mining of sequential patterns using co-occurrence information, in: Advances in Knowledge Discovery and Data Mining: Proceedings of the 18th Pacific-Asia Conf., PAKDD, Part I, Springer, 2014, pp. 40–52.

[32] C. Luo, S. M. Chung, Efficient mining of maximal sequential patterns using multiple samples, in: Proceedings of the 2005 SIAM Int. Conf. on Data Mining, 2005, pp. 415–426.

[33] S. Ferré, O. Ridoux, A logical generalization of formal concept analysis, in: Proceedings of the 8th Int. Conf. on Conceptual Structures, ICCS, Springer, 2000, pp. 371–384.

[34] S. Ferré, A proposal for extending formal concept analysis to knowledge graphs, in: Proceedings of the 13th Int. Conf. on Formal Concept Analysis, ICFCA, Springer, 2015, pp. 271–286.

[35] G. Casas-Garriga, J. L. Balcázar, Coproduct transformations on lattices of closed partial orders, in: Graph Transformations, Springer, 2004, pp. 336–351.

[36] P. Cellier, S. Ferré, M. Ducassé, T. Charnois, Partial orders and logical concept analysis to explore patterns extracted by data mining, in: Proceedings of the 19th Int. Conf. on Conceptual Structures, ICCS, Springer, 2011, pp. 77–90.

[37] E. Egho, N. Jay, C. Raïssi, A. Napoli, A FCA-based analysis of sequential care trajectories, in: Proceedings of the 8th Int. Conf. on Concept Lattices and their Applications, CLA, 2011, pp. 1–11.

[38] M. Plantevit, A. Laurent, D. Laurent, M. Teisseire, Y. W. Choong, Mining multidimensional and multilevel sequential patterns, ACM Trans. Knowl. Discov. Data 4 (1) (2010) 1–37.

[39] S. Ferré, P. Cellier, How hierarchies of concept graphs can facilitate the interpretation of RCA lattices?, in: Proceedings of the 14th Int. Conf. on Concept Lattices and Their Applications, CLA 2018, Olomouc, Czech Republic, 2018, pp. 69–80.

[40] R. Godin, H. Mili, Building and maintaining analysis-level class hierarchies using galois lattices, in: Proceedings of the 8th Annual Conf. on Object-oriented Programming Systems, Languages, and Applications, OOPSLA, ACM, 1993, pp. 394–410.

[41] S. Kuznetsov, T. Makhalova, On interestingness measures of formal concepts, Information Sciences 442-443 (2018) 202–219.

[42] A. Braud, X. Dolques, M. Huchard, F. Le Ber, Generalization effect of quantifiers in a classification based on relational concept analysis, Knowledge-Based Systems 160 (2018) 119–135.

Algorithm 1: CPOHrchy

Input : the RCA result comprises $\mathcal{L}_{K_m} = (\mathcal{C}_{K_m}, \preceq_{K_m})$, $\mathcal{L}_{K_t} = (\mathcal{C}_{K_t} \preceq_{K_t})$, and $\mathcal{L}_{K_i} = (\mathcal{C}_{K_i}, \preceq_{K_i})$

Output: the lattice $\mathcal{L}^*_{K_m}$ of P_m structures, based on $\mathcal{L}_{K_m}$ concepts and their associated CPO-patterns $\mathcal{G}_m$

$\mathcal{L}^*_{K_m} \leftarrow \emptyset$
foreach $C_m = (X_m, Y_m) \in \mathcal{C}_{K_m} \setminus \bot(\mathcal{L}_{K_m})$ **do**
  $\mathcal{G}_m \leftarrow$ initialise to $(C_m, \emptyset, \emptyset)$
  if Y_m *has temporal relational attributes* **then**
    $\mathcal{C}_{next} \leftarrow \mathsf{SearchAdjacentConcepts}(Y_m)$
    $\mathcal{G}_m \leftarrow$ initialise to $(C_m, \emptyset, \mathcal{C}_{next})$
    Queue $\leftarrow$ enqueue $\mathcal{C}_{next}$ concepts and mark them as visited
    repeat
      $C_t = (X_t, Y_t) \leftarrow$ dequeue *Queue*
      $v_t \leftarrow$ derive an itemset based on $\mathcal{L}_{K_i}$ and the qualitative relational attributes of Y_t
      if Y_t *has temporal relational attributes* **then**
        $\mathcal{C}'_{next} \leftarrow \mathsf{SearchAdjacentConcepts}(Y_t)$
        $\mathcal{G}_m \leftarrow$ add $(C_t, v_t, \mathcal{C}'_{next})$ to $\mathcal{G}_m$
        $\mathcal{C}'_{next} \leftarrow$ delete from $\mathcal{C}'_{next}$ already visited concepts
        if $\mathcal{C}'_{next}$ *is not empty* **then**
          Queue $\leftarrow$ enqueue the $\mathcal{C}'_{next}$ concepts and mark them as visited
        end
      else
        $\mathcal{G}_m \leftarrow$ add $(C_t, v_t, \emptyset)$ to $\mathcal{G}_m$
      end
    until *Queue is empty*;
  end
  $P_m \leftarrow (X_m, Y_m, \mathcal{G}_m)$
  $\mathcal{L}^*_{K_m} \leftarrow$ add P_m to $\mathcal{L}^*_{K_m}$
end
return $\mathcal{L}^*_{K_m}$

Algorithm 2: SearchAdjacentConcepts

Input : intent Y_m of a main concept C_m
Output: $\mathcal{C}_{next}$ the set of the next navigated concepts

$\mathcal{C}_{next} \leftarrow$ initialise to $\{C_t | (\exists ipb_1(C_t)) \in Y_m\}$
if $|\mathcal{C}_{next}| > 1$ **then**
    $UpperCovers \leftarrow \emptyset$
    foreach $C_t \in \mathcal{C}_{next}$ **do**
        $UpperCovers \leftarrow$ add $\{C'_t | C'_t \rhd_{K_t} C_t\}$ to $UpperCovers$
    end
    $\mathcal{C}_{next} \leftarrow \mathcal{C}_{next} \setminus UpperCovers$
end
if $|\mathcal{C}_{next}| > 1$ **then**
    $ToBeDeleted \leftarrow \emptyset$
    foreach $C_t = (X_t, Y_t) \in \mathcal{C}_{next}$ **do**
        $ToBeDeleted \leftarrow$ add $\{C'_t | (\exists ipb_2(C'_t)) \in Y_t\}$ to $ToBeDeleted$
    end
    $\mathcal{C}_{next} \leftarrow \mathcal{C}_{next} \setminus ToBeDeleted$
end
return $\mathcal{C}_{next}$