Towards a Formal Framework for Mining General Patterns from Ordered data *

G. C. Garriga

Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Jordi Girona Salgado 1-3, Barcelona
gcasas@lsi.upc.es

Abstract. In this paper we present the initial notions of a framework for mining general patterns from complex structured data. To achieve this goal, we start by focusing here on the most basic case: the mining of a plain ordered collection of data, such as sequential databases or time-series data. Our framework develops on the hypothesis of using the *closure* of the Galois connection as a way to reduce the number of the extracted general patterns to the most representative ones. So, we employ formal concept analysis to characterize the concept lattice of these ordered contexts; our main contribution is a new Galois connection that, in this work, allows to exactly derive the closed sequential patterns found by a recent algorithm (CloSpan, [9]).

1 Introduction

Mining structured data is a complex task requiring specific techniques and formalizations different from the ones commonly applied to single normalized tables. The most basic type of structured data is an ordered context, where elements follow in an specific sequential order. These elements in the sequence can come in a simple form, such as a set of items, or also have a more complex structure, such as a hierarchical organization. The kind of patterns that can be extracted from any of these contexts can also range from a plain structure (sequential subsequences matching frequently in the data) to more complex tree-like forms (such as hybrid episodes). As a starting point for the initial foundations of our framework, we will be considering first those ordered contexts whose elements present a simple structure (i.e, sequences of sets of items that can be transactions of a sequential database or also are extended into a time-series-data form). The analysis of these ordered contexts gives rise to the sequential pattern mining problem, a relevant task in Knowledge Discovery in Databases where the identification of frequenly-arising patterns or subsequences is expected to be of interest ([2,

* This work is supported in part by EU ESPRIT IST-1999-14186 (ALCOM-FT), and MCYT TIC 2002-04019-C03-01 (MOISES)

7]). This task has many applications in different fields such as DNA sequences, proteins with similar biological functions, customer shopping sequences, intrusion detection problems, analysis of alarms in a telecomunication network and so on. One problem of the sequential pattern mining task arises when using very low support threshold in the algorithms; then, the number of extracted patterns is usually too large for a thorough examination. This problem also occurs in the task of mining association rules for large masses of unordered data ([1]): an interesting solution, called *closed itemsets* ([3, 8, 10, 11]), uses the closure of the Galois connection to reduce the total number of association rules. The foundations of closed itemsets are based on the mathematical model of concept lattices ([4, 5]), that fits perfectly well in the data mining context of an unstructured binary table. The main idea of the formalization is to present a concept-lattice framework (a lattice of closed concepts where each concept contains a closed itemset) constructed by using the Galois connection primarily introduced in [4, 5]. It can be shown that this set of closed itemsets in the lattice is necessary and sufficient to capture all the information about frequent itemsets, having smaller cardinality than the set of all frequent itemsets.

In this paper, we start by developing our framework on this hypothesis of employing the closure of a Galois connection to represent a minimum cover of all the complex structured patterns that are present in our data. As we know, there is nowadays a considerably lack of theory and formal methods for the structured case, specially in terms of representing and formalizing the idea of closed structured patterns. Thus, our main goal here is evaluate the use of formal concept analysis as a formal method to characterize, for the moment, the ordered contexts (taken as the most basic case for further complex structured data) and so, construct a lattice of closed ordered concepts by defining a new Galois connection. In this work we will show how our proposed concept lattice will serve to exactly characterize the closed patterns extracted by a recent algorithm (CloSpan, [9]), and at the same time, it will be general enough for the slightly different problem of having a single event sequence [7].

1.1 Paper Organization

We divide the work in several sections: in section 2 we present the problem formulation and the preliminary notions. Section 3 explains the previous work in [9] as the only first attempt to tackle the closure of sequences (i.e, plain ordered patterns), and motivates the need of a more formal method. The use of formal concept analysis to generally characterize the ordered context is presented in section 4; this section includes three different subsections, each one of them is a necessary step to get a well-founded lattice of closed concepts with order. Section 5 shows that it is possible to derive the closed sequential patterns in [9] from our lattice model. Finally, last section addresses the conclusions and the future work to motivate that our model can be still general enough for deriving other kind of general patterns.

2 Problem Formulation and other Necessary Definitions

Let $\mathcal{I} = \{i_1, \ldots, i_n\}$ be a set of all items, also called the set of all **attributes** of the database. A subset of $\mathcal{I}$, i.e $I \subseteq \mathcal{I}$, is called an itemset. We are now dealing with the problem of having a collection of data $\mathcal{D} = \{s_1, s_2, \ldots s_n\}$, where each s_i is a **sequence**, also called an ordered transaction. It is important for our work that we model each element of the sequence, not as an item, but as an itemset. As we see, this problem definition fits perfectly well in the context of having a *sequential database*; in case we are dealing with *time-series data*, the long string of data can be divided in several sliding windows representing each one a piece of $\mathcal{D}$.

We consider a **sequence** an ordered list of itemsets. It can be represented as a list of pairs, $\langle (I_1, t_1)(I_2, t_2) \ldots (I_n, t_n) \rangle$, where each I_i are subsets of $\mathcal{I}$, and t_i are the order of occurrence of each itemset in the sequence (i.e, position indices). The set of all the possible sequences will be noted by $\mathcal{S}$. Without loss of generality we assume that the elements of a sequence are mapped to a set of contiguous integers and that the items in each itemset are sorted in certain order (such as alphabetic order). So, for short we will note sequences of itemsets $\langle (I_1, t_1)(I_2, t_2) \ldots (I_n, t_n) \rangle$ as $\langle (I_1)(I_2) \ldots (I_n) \rangle$. Some definitions on sequences are the following:

Definition 1. *A sequence $s = \langle (I_1) \ldots (I_n) \rangle$* **is subsequence** *of another sequence $s' = \langle (I'_1) \ldots (I'_m) \rangle$, i.e we note it by $s \subseteq s'$, if there exist integers $j_1 < j_2 \ldots < j_n$ such that $I_1 \subseteq I'_{j_1}, \ldots, I_n \subseteq I'_{j_n}$.*

When s is subsequence of another s', we also say that s **is contained** in s'. For example, the sequence $\langle (C)(D) \rangle$ is contained in $\langle (AC)(D)(B) \rangle$, so the first is a subsequence of the second. A sequence is maximal if it is not contained in any other sequence.

Definition 2. *The* **intersection** *of a collection of sequences $s_1, \ldots, s_n \in \mathcal{S}$, i.e, $s_1 \cap \ldots \cap s_n$, is the set of maximal subsequences contained into all the s_i.*

The intersection of a set of sequences can give rise to more than one sequence (note that this fact did not occur when intersecting a set of itemsets of a single table of binary data). For example, the intersection of $s = \langle (AD)(C)(B) \rangle$ and $s' = \langle (A)(B)(C) \rangle$ is the set of sequences $\{\langle (A)(C) \rangle, \langle (A)(B) \rangle\}$ since both are contained in s and s'. This important fact will determine the characterization of the subsequent lattice model, different from the lattice for unordered data.

The **support** of a sequence s in the ordered data $\mathcal{D}$ is the number of sequences in $\mathcal{D}$ that contain s. Usually, the data mining problem stated for this kind of ordered context is to find all the frequent sequential patterns (i.e, sequences whose support is over a user-specified value). However, the number of generated patterns grows quickly as the support threshold gets lower. An interesting solution is to consider the closure on sequential patterns to represent just the representative ones. We want to characterize here the closure in terms of Galois connection of all potential patterns in our data. For this purpose we are not considering in this paper any minimum support threshold for the concepts (or in other words, we consider that the threshold is set to zero, so that all possible patterns can be represented in the model); in a forthcoming work we will describe how to add support conditions to our framework.

3 Previous Related Work

An initial attempt to tackle the closure of sequences is recently performed by [9]. In that work the authors present an algorithm called CloSpan to find a kind of individual closed sequences in a sequential database. In particular, the authors define a set of **closed sequential patterns** to be found by CloSpan as:

$$CS \;=\; \{s \mid \nexists s' \; s.t \; s \subset s' \; and \; support(s) = support(s')\}$$

The way the algorithm works to find those sequences in CS is actually irrelevant for our purposes. We are trying to do a formalization of a framework for closed structured patterns (that will include in turn the notion of stability expressed by this set CS among others); so, the performance analysis of the algorithm is not important for our work, and we only focus on the definition given by [9] for closed sequential patterns. Actually, we can see that the intuitive notion of closure that can be recognized in the definition of CS is linked with the support of those sequences covering others in the database. We can see then that CS gathers a very similar semantic as the one stated for closed itemsets for simple binary data of a single normalized table. In particular, if we consider Γ to be the closure operator of an itemset closure system, the set of **closed itemsets** is defined as the following ([3, 8, 10, 11]):

$$\begin{aligned} CI = \{X \mid \Gamma(X) = X\} &\equiv \{X \mid \delta(\rho(X)) = X\} \equiv \\ &\{X \mid \nexists Y \; s.t \; X \subset Y \; and \; support(X) = support(Y)\} \end{aligned}$$

where Γ is the closure operator resulting from the composition of operators δ and ρ that form a Galois connection, such that, $\rho(X)$ returns all the transactions in which the input itemset X occurs in the database, and $\delta(T)$ returns the items common to all transactions T.

So, CS and CI have similar meaning of closure but in a different context. However, we consider here that the set CS does not represent all the particularities of the structured patterns that are hidden in the sequential data. For example, if we consider the database as in figure 1, then the sequence $\langle(A)\rangle$ is a closed sequential pattern and so, it belongs to CS, because there are no super-sequences having the same support; the sequence $\langle(A)(C)\rangle$ is not closed because

Seq id	Sequence
t_1	$\langle(A)(C)(B)\rangle$
t_2	$\langle(A)\rangle$
t_3	$\langle(C)(B)(A)\rangle$
t_4	$\langle(B)(C)(A)\rangle$

Fig. 1. Example of a sequential database

of $\langle(A)(C)(B)\rangle$, that has the same support and it is its supersequence. However, the simplicity of the representation of set CS does not allow to express the relation hold by sequences like $\langle(C)(A)\rangle$ and $\langle(B)(A)\rangle$: both are closed sequential patterns, and both occur in exactly the same transactions and have the same support; but, one is not the inclusion of the other.

Formally, it can exist two closed sequential patterns s and s' such that they occur in the same transactions, so $support(s) = support(s')$, but $s \nsubseteq s'$ and $s' \nsubseteq s$. This fact cannot be captured by the unidimensional representation of CS; however, it can be interesting to keep track of this relation between those different patterns in order to represent still more information. For this reason, we propose the study of the concept lattice theory on the ordered context, and derive from the new representation the set of closed patterns.

4 Concept Lattice Theory for Ordered Data

We pretend here to use formal concept analysis to create a new closure system taking into account the feature of order. We consider that the resulting concept lattice model will be a more general representation to express all the characteristics of our data. For that, we will formalize a definition of ordered context; then we define two derivation operators forming a Galois connection and, finally, we will present the formal closed concepts in a new particular lattice that will be used as a model.

4.1 Ordered Context

We start by specifying the ordered context that will serve as a basis of our lattice.

Definition 3. *An* **ordered context** $\mathbb{K} = (\mathcal{O}, \mathcal{I}, \mathcal{T}, R)$*, consists of sets* $\mathcal{O}, \mathcal{I}$ *and* $\mathcal{T}$*, and a ternary relation* R *such that* $R \subseteq \mathcal{O} \times \mathcal{I} \times \mathcal{T}$*.*

The elements of $\mathcal{O}$ are called the **objects** of the context (i.e, the set of original transactions), those of $\mathcal{I}$ the **attributes**, and those of $\mathcal{T}$ the **time of occurence** of the attribute. For an entry $(o, i, t) \in R$ we read as "the attribute i occurs at the time t in the object o", so, t represents the *order* of occurrence of attribute

i with respect to the other atributes in the same object. Since an attribute can occur more than once in a object, it could also have more than one order in the same object.

Like the one-valued contexts treated so far (for binary data), ordered contexts can be also represented by a cross table, the rows of which are labelled by the objects and the columns labelled by the attributes. The entry in row o and column i represents the order/s of the attribute in the object. If the attribute i does not have an order of occurrence for the object o, there will be no entry.

	A	**B**	**C**
o_1	1	3	2
o_2	1		
o_3	3	2	1
o_4	3	1	2

Fig. 2. Ordered context $\mathbb{K}$ for database in figure 1

Since each object can be interpreted as a sequence itself, the intersection of a set of objects in the context can give rise to a set of sequences, each one of them intersects with each input object. In the following subsection, we formalize the closure operator working over this proposed context.

4.2 A New Closure Operator

For our goals we define the following two new derivation operators:

Definition 4. *For a set $O \subseteq \mathcal{O}$ of objects we define,*

$$\phi(O) = \{s \in \mathcal{S} | \ s \ \textbf{maximal} \ \textit{contained in} \ o, \ \forall o \in O\}$$

This $\phi(O)$ is the set of sequences common to all the objects in O, i.e, we can say that $\phi(O)$ represents the set of maximal sequences coming from the intersection of all the input objects O. Correspondingly, for a set $S \subseteq \mathcal{S}$ of sequences we can define,

$$\psi(S) = \{o \in \mathcal{O} | \ s \ \textit{contained in} \ o, \ \forall s \in S\}$$

This $\psi(S)$ is the set of objects containing all the sequences in S.

Given that we are in an ordered context and so, the intersection of a set of objects can result in more than one sequence, we have defined for our purposes that $\phi : 2^{\mathcal{O}} \rightarrow 2^{\mathcal{S}}$. It is worth noting that the condition of maximality stated in the definition of the derivation operator ϕ is an important necessary condition for our later need of representing the notion of closure for individual sequences as we will explain. The other derivation operator is defined as $\psi : 2^{\mathcal{S}} \rightarrow 2^{\mathcal{O}}$, since it might be that a given set of sequences S occurred always together, in the same set of objects (for this reason, the intersection of a collection of objects can give rise to a set of sequences). To be able to prove that our two derivation operators form a Galois connection, we need to specify the convenient order on the sets of sequences and on the set of objects.

Definition 5. *We say that a set of sequences S is* **more specific** *than another set of sequences S', i.e $S \preceq S'$, if and only if $\forall s \in S$, $\exists s' \in S'$ s.t. $s \subseteq s'$. Then S' is also said to be* **more general** *than S.*

According to the definition, the set of sequences $\{\langle(B)\rangle, \langle(C)\rangle, \langle(A)\rangle\}$ is more specific than the set $\{\langle(B)(A)\rangle, \langle(C)(A)\rangle\}$. Finally the two orders we consider on the sets of objects and sequences are the following: 1/ We can partially order sets of sequences by the relation $\preceq$; 2/ We can partially order sets of objects by the relation $\subseteq$ (standard inclusion). The following proposition 1 shows that the two derivation operators using the proposed orders above, form a Galois connection between the power-set lattices on $\mathcal{O}$ and on $\mathcal{S}$.

Proposition 1. *The maps $(\phi, \subseteq)$ and $(\psi, \preceq)$ form a* Galois connection*.*

Proof. See proof as appendix A.

Once the Galois connection is proved, we can get a new closure operator, from the composition of both derivation operators. Proposition 2 exactly states this result.

Proposition 2. *Composition of ϕ and ψ, $\Delta = \psi \cdot \phi$, or $\Delta = \phi \cdot \psi$, is a closure operator.*

According to [4] or [5], the following properties on the composition of ψ and ϕ have to be proved to show that we have a closure operator: 1/ Monotonicity: $S \preceq S' \Rightarrow \Delta(S) \preceq \Delta(S')$; 2/ Extensivity: $S \preceq \Delta(S)$; 3/ Idempotency: $\Delta(\Delta(S)) = \Delta(S)$. These conditions can be easily obtained by combining well-known properties stemming from any Galois connection. Thus, for simplicity purposes, we refer the reader to [4, 5] to prove proposition 2. Now, with the new closure operator Δ we can obtain two clousure systems: one on $\mathcal{O}$ (from $\psi \cdot \phi$) and another on $\mathcal{S}$ (from $\phi \cdot \psi$), that are dually isomorphic to each other. These two closure systems will lead to the characterization of the concept lattice as we will see in the next section.

4.3 Formal Concepts

How do we assign concepts of the lattice from an ordered context? We could first think of forgetting any new definition of the derivation operators, and try to transform the ordered context into a single normalized one-valued table, as [5] proposed to do for the multi-valued contexts. This transformation process in [5] consists of applying a "conceptual scaling" that is not at all uniquely determined. In particular, the scaling consists on expanding each original atribute with the labels of any of its possible values, in such a way that the context is translated into a binary one. So, each attribute is transformed in as many attributes as original values it has, making the intersection of objects occurring naturally when applying the closure operator. In the ordered context this scaling would not work: the new labels of each attribute would be the different positions that the attribute can take in the different sequences; however, attributes do not always occur in the same position in all the objects, which this can make difficult their intersection. For instance, a subsequence $\langle(A)(B)\rangle$ can occur in positions $\langle(A,1),(B,2)\ldots\rangle$ in one object, but in positions $\langle\ldots(A,3),(B,5)\ldots\rangle$ in another object. The intersection of these two objects should clearly give the sequence $\langle(A)(B)\rangle$; however, if we expand the labels of each attribute with their values as in the scalling, then they will never intersect since labels $A3$, $A1$ and $B5$, $B2$ will be different. We also note that the different ordinal contexts presented in [5] do not serve either to represent and express the order we want on the attributes of an object. In fact, an ordinal contex is just an specific case of multi-valued contexts where the order is specified among all the possible values of a single attribute (intra-attribute), and we would better need an inter-attribute order. For this reason, we discard the former methods in our analysis, and we present here our own definition and proposals for the formal concepts.

Definition 6. *A* **formal concept** *of the ordered context* $\mathbb{K}$ *is a pair* (O,S) *where* $O \subseteq \mathcal{O}$*,* $S \subseteq \mathcal{S}$*, and* $\phi(O) = S$ *and* $\psi(S) = O$*. Then* O *is the extent and* S *is the intent of the concept. These concepts are called* **closed** *since* $S = \Delta(S)$ *and* $O = \Delta(O)$*.*

The important difference of this formal concept with the one-valued concept treated so far is mainly in the the intent part, that consists on a *set* of *maximal* sequences contained in the objects of the extent. In other words, the intent *is not* a single sequence of ordered itemsets, as it would to be expected as a direct comparison with the concept of the one-valued context, but a set of sequences.

Extent	**Intent**
1234	$\{\ \langle(A)\rangle\ \}$
134	$\{\ \langle(A)\rangle,\ \langle(B)\rangle,\ \langle(C)\rangle\ \}$
34	$\{\ \langle(B)(A)\rangle,\ \langle(C)(A)\rangle\ \}$
13	$\{\ \langle(C)(B)\rangle,\ \langle(A)\rangle\ \}$
1	$\{\ \langle(A)(C)(B)\rangle\ \}$
3	$\{\ \langle(C)(B)(A)\rangle\ \}$
4	$\{\ \langle(B)(C)(A)\rangle\ \}$

Fig. 3. Closed concepts

So, for every set $O \subseteq \mathcal{O}$, $\phi(O)$ is the intent of some concept, since $(\psi(\phi(O)), \phi(O))$ will be always a concept. Consequently, a set $O \subseteq \mathcal{O}$ is an extent if an only if $O = \psi(\phi(O)) = \Delta(O)$. The same applies to intents. So, for the intent part, we are trying to find closed set of sequences, that is $S = \phi(\psi(S)) = \Delta(S)$ as a result of all the intersections of objects where each sequence in S occurs as a subsequence. All the sequences of events belonging to the same intent are called a **closed set of sequences**, and we say that these sequences form an **equivalence class**.

Given the context of figure 2, the set of all possible closed concepts are shown in figure 3. The intent of each concept is composed by the set of maximal sequences that result from the intersection of the set of objects in the intent (these objects in the intent are also the maximum number whose intersection generates those closed set of sequences). The extent is the set of indexes of the objects where the intent appears.

The basic properties of lattices still hold in our model: the union of extents does not result in another extent. On the other hand, the intersection of any number of extents/intents is always an extent/intent.

Proposition 3. *The intersection of intents/extents is another intent/extent.*

Proof. See proof as appendix B.

From the set of all our formal concepts we can construct now the concept lattice for the ordered context. Actually, the concept lattice representation is just a way to graphically express the partial orders on the concepts defined by:

Definition 7. *If* (O_1, S_1) *and* (O_2, S_2) *are concepts of a context,* (O_1, S_1) *is a* subconcept *of* (O_2, S_2) *if* $O_2 \subseteq O_1$ *(which is equivalent to* $S_1 \preceq S_2$*, i.e,* S_2 *is more general). Then* (O_2, S_2) *is a superconcept of* (O_1, S_1) *and we can write* $(O_1, S_1) \leq (O_2, S_2)$*.*

The set of all concepts of the context $\mathbb{K} = (\mathcal{O}, \mathcal{I}, \mathcal{T}, R)$ ordered by $\leq$ is denoted as $\underline{\mathfrak{B}}(\mathcal{O}, \mathcal{I}, \mathcal{T}, R)$ and is called the concept lattice of the context. This lattice contains just the summary of all the closed set of sequences in the context and the objects where each equivalence class is contained. In figure 4 we show the representation of the final lattice for the context of figure 2: each node is a concept, and concepts are ordered by $\leq$.

As we see, this final concept lattice can capture all the special characteristics of a sequential database in the intents of each concept: those sequences in the same equivalence class appear in the same transactions (objects) and have the same support, but they are not comparable by $\preceq$. Now we are interested in this lattice as a model to derive interesting patterns: what do the sequences in each concept mean? and, which general patterns can be derived from the set of sequences belonging to the same intent?

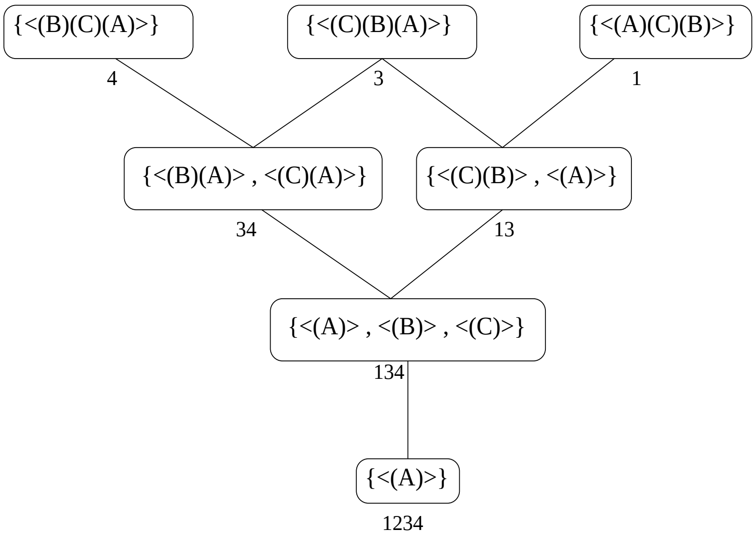

Fig. 4. Example of a concept lattice $\underline{\mathfrak{B}}(\mathcal{O}, \mathcal{I}, \mathcal{T}, R)$

5 Closed Sequential Patterns in the Lattice

Once the concept lattice is characterized, we have in each of the concepts a notion of *closed set of sequences* (the set of sequences in the same equivalence class that fullfils $S = \Delta(S)$). In other words, the closure operator Δ of the model can be only used for a set of sequences, not single individual sequences. However, we also did want in our model a notion of closure for a single sequence, like in CloSpan, in order to also characterize from our lattice the individual closed patterns that belong to set CS. It may be tempting to say that a single sequence s is closed when the set $\{s\}$ (containing only the individual s) is closed, but this definition does not work: for instance, in the sequential database of figure 1, the individual sequence $\langle(C)\rangle$ is closed according to algorithm CloSpan (since $\langle(C)\rangle \in CS$); however, if we apply the closure operator to the set $\{\langle(C)\rangle\}$ we get: $\Delta(\{\langle(C)\rangle\}) = \{\langle(A)\rangle, \langle(B)\rangle, \langle(C)\rangle\}$, so, we would conclude that is not closed because $\Delta(\{\langle(C)\rangle\}) \neq \{\langle(C)\rangle\}$. We can see then that the direct use of our closure operator that builds the lattice does not work to distinguish individual closed sequences; however, we would like to derive those sequences in CS also from our model using the closure operator. In order to achieve this goal, we first start by defining the following relation on the individual sequences s and the closure operator Δ. This relation states an important property that always holds in our lattice model and that will help in the subsequent analysis.

Proposition 4. *For any individual sequence s, it always true that $\{s\} \preceq \Delta(\{s\})$.*

Proof. See proof as appendix C.

From this last proposition, we conclude that if we try to close a single sequence s with the closure operator $\Delta(\{s\})$, we get a set of sequences where at least one of them is a supersequence of s. In other words, any s always has a "representative" in some concept. Another important and necessary property always true in our lattice model is stated by the following proposition.

Proposition 5. *For S a closed set of sequences, we have that $\nexists s, s'$ s.t $s \in S$, $s' \in S$ and $s \subseteq s'$.*

Proof. See proof as appendix C.

So, the individual sequences in the concepts of the lattice are maximal. Intuitively, this last proposition 5 can be only proved because of the maximality condition imposed on the derivation operator $\phi(O)$. In other words, the definition of $\phi(O)$ is made with the aim of avoiding that two different sequences, being one the inclusion of the other, can belong to the same closed set S. This important fact will validate our subsequent analysis. For a more detailed explanation about this refer to proof in appendix C.

The importance of these two last properties is that they will allow to formulate the subsequent characterization of closure for individual sequences. Hence, we are now interested in deriving from our lattice those sequences that are stable in support, in the same way that CloSpan produces the set CS (individual sequences covering the support of others). For our formalization purposes, we define the following notion of stability.

Definition 8. *A single sequence s is* **stable** *under in tersection if $s \in \Delta(\{s\})$*

This notion of stability gathers all those *maximal* sequences occuring in a *maximal* set of objects of the context $\mathbb{K}$. In other words, this maximal s is stable in a set of objects O when we cannot add any other object to the set O without losing this s in the intersection of all objects in O. Not all the sequences are stable (later we will prove that just the sequences of CloSpan follow this notion of stability and so, CloSpan is related to Δ). For example, sequence $\langle(B)(C)\rangle$ from the sequential database of figure 1 is not stable since $\Delta(\{\langle(B)(C)\rangle\}) = \{\langle(B)(C)(A)\rangle\}$. Now we can already propose the following lemma that exactly stablishes the location of stable sequences in the lattice.

Lemma 1. *Let S be a closed set of sequences, then $\forall s \in S$, s is an stable sequence.*

Proof. See proof as appendix C.

Lemma 1 shows us that the intents of the concepts of our lattice (that are by defition closed set of sequences), also contain the individual stable sequences, i.e, for all $s \in S$ where $\Delta(S) = S$, then s is stable.

Now that we have characterized all the stable sequences in our lattice, we are interested in defining exactly how the notion of stability can be equivalent to the sequences in CS found by CloSpan. This following theorem proves that the closed set of sequences in the lattice are exactly composed by the closed sequential patterns discovered in the set CS by CloSpan.

Theorem 1. *The set of all stable sequences is exactly the set of closed sequential patterns extracted by the algorithm CloSpan [9].*

Proof. See proof as appendix C.

From theorem 1 we reach the conclusion that the notion of closed sequences of CloSpan can be captured by each one of those sequences belonging to the equivalence classes in the concepts of our lattice, and so, the individual notion of closure is also contained in the concepts. The reason to consider an individual notion of closure interesting for our problem, is that these individual closed patterns represent a **cover in terms of support**, i.e, each stable sequence s is the maximal belonging also to the set CS (that is, all non-stable sequences s' such that $s' \subset s$ are covered by s, that has the same support). And actually, CloSpan is currently the only algorithm that mines stable sequences. It is worth noting that defining a concept lattice where the intents are single individual stable sequences (closed sequences of CS) is not possible. This is due to the derivation operators ϕ and ψ that would not form a Galois connection if we restrict their definition. The only way to create a closure system is through a set of closed sequences, where each element in the set is stable. This representation will be more general and will conceive the possibility to derive other kind of structured patterns from the concepts as we will motivate in the next section.

6 Conclusions and Future Work

In this paper we present the initial notions of a framework for mining general patterns from structured data. We develop on the hypothesis that the most representative of these general patterns are the closed ones; for that, we take here the basic ordered case and we characterize a new concept lattice in terms of Galois connection that exactly represents all the characteristics of our sequential context. The new lattice model turns to have quite useful properties and this will allow us to derive interesting closed patterns from the concepts in the nodes. In particular, we show in this work how the individual plain stable sequences in the concepts of the model exactly characterize the closed sequences found by CloSpan. This fact makes this algorithm useful to construct our lattice model as well.

There are many interesting research problems that can stem from this first work. For example, the formalization of how to derive other kind of complex patterns from this proposed lattice; in particular, patterns such as closed episodes (serial, parallel and other tree-like forms such as hybrid episodes in [7]). Intuitively, if each object in the context represents an sliding window in a long sequence of events, then, stable sequences could be transformed into closed serial episodes (that in turn would be related to CloSpan as well). Closed parallel and hybrid episodes would be derived from each one of the equivalence classes in the nodes of the lattice. Furthermore, it can be also interesting in this line of research to prove how similar are these closed patterns to the first closed episodes proposed by [6].

Finally, we expect that this model can be also extended to the processing of data organized in more complex partially ordered structures (such as hierarchical elements). We also expect that the framework could be completed by deriving knowledge from other kinds of multi-relational data always using the hypothesis of closure systems.

References

1. R.Agrawal, T. Imielinski and A.Swami. Mining Association Rules Between Sets of Items in Large Databases. *Proc. of the ACM SIGMOD Int'l Conference on Management Dada*, pages 207-216. 1993.
2. R.Agrawal and R.Srikant. Mining Sequential Patterns. *Proc. Int'l Conference on Data Engineering*, pages 3-14. 1995.
3. Y.Bastide, N.Pasquier, R.Taouil, G.Stumme and L.Lakhal. Mining Minimal Non-Redundant Association Rules using Frequent Closed Itemsets. *First Int'l Conference on Computational Logic*. 2000.
4. B.A.Davey and H.A.Priestley. *Introduction to Lattices and Order*. Cambridge, 2002.
5. B.Ganter and R.Wille. *Formal Concept Analysis. Mathematical Foundations*. Springer, 1998.
6. S.Harms, J.Deogun, J.Saquer and T.Tadesse. Discovering Representative Episodal Association Rules from Event Sequences Using Frequent Closed Episode Sets and Event Constraints. *Int'l Conference on Data Miminig*. 2001.
7. H.Mannila, H.Toivonen and I.Verkamo. Discovery of frequent episodes in event sequences. *Proc. of the Int'l Conference on Knowledge Discovery and Data Mining*. 1995.
8. N.Pasquier, Y.Bastide, R.Taouil, and L.Lakhal. Closed set based discovery of small covers for association rules. In *Proc. of the 15th. Conference on Advanced Databases*, pages 361-381. 1999.
9. X.Yan, J.Han and R.Afshar. CloSpan: Mining Closed Sequential Patterns in Large Databases. *Int'l Conference SIAM Data Mining*. 2003.
10. M.Zaki and M.Ogihara. Theoretical Foundations of Association Rules. *Third SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. 1998.
11. M.Zaki. Generating non-redundant Association Rules.In *Proc. of the Sixth Int'l Conference on Knowledge Discovery and Data Mining*, pages 34-43. 2000.

Appendix A

Proposition 1 *The maps* $(\phi, \subseteq)$ *and* $(\psi, \preceq)$ *form a* Galois connection.

Proof. For sets of objects $O, O_1, O_2 \subseteq \mathcal{O}$, and sets of sequences $S, S_1, S_2 \subseteq \mathcal{S}$ we should prove:

1) $O_1 \subseteq O_2 \Rightarrow \phi(O_2) \preceq \phi(O_1)$	1') $S_1 \preceq S_2 \Rightarrow \psi(S_2) \subseteq \psi(S_1)$
2) $O \subseteq \psi(\phi(O))$	2') $S \preceq \phi(\psi(S))$

1) *For all* $s' \in \phi(O_2)$ *we have that, for all* $o' \in O_2$, s' *is subsequence of* o', *that is, in particular* s' *is subsequence of* o *for all* $o \in O_1$, *if* $O_1 \subseteq O_2$, *and thus* $\exists s \in \phi(O_1)$ *s.t* $s' \subseteq s$, *which means* $\phi(O_2) \preceq \phi(O_1)$.
2) *For all* $o \in O$ *we have that,* s *is subsequence of* o *for all* $s \in \phi(O)$, *and thus* $o \in \psi(\phi(O))$.
1') *For all* $o' \in \psi(S_2)$ *we have that, for all* $s' \in S_2$, s' *is subsequence of* o', *that is, in particular* s *is subsequence of* o' *for all* $s \in S_1$, *if* $S_1 \preceq S_2$, *and thus* $o' \in \psi(S_1)$, *which means* $\psi(S_2) \subseteq \psi(S_1)$.
2') *For all* $s \in S$ *we have that,* s *is subsequence of* o *for all* $o \in \psi(S)$, *and thus* $\exists s' \in \phi(\psi(S))$ *s.t* $s \subseteq s'$ *which implies* $S \preceq \phi(\psi(S))$.

□

Appendix B

In this appendix we show that the property on the intersection of intents/extents of the classical one-valued model, also holds for our new lattice on sets of sequences. We consider that the intersection of two set of sequences $S_1 \bigcap S_2$ is the cross intersection of single sequences $s_1 \cap s_2$ for all $s_1 \in S_1$ and all $s_2 \in S_2$.

Proposition 3 *The intersection of intents/extents is another intent/extent.*

Proof. We want to reject the following false hypothesis: from S_1 and S_2 two closed set of sequences, we suppose that $S_1 \bigcap S_2$ is not closed, i.e, it is not an intent, which can we written as:

$$S_1 \bigcap S_2 \preceq \Delta(S_1 \bigcap S_2) \quad and \quad S_1 \bigcap S_2 \neq \Delta(S_1 \bigcap S_2)$$

In particular, this implies that we can have one of the two following situations:

1/ for some $s \in S_1 \bigcap S_2$, $\exists s' \in \Delta(S_1 \bigcap S_2)$ s.t $s \subset s'$, which implies that $s' \notin S_1 \bigcap S_2$.

2/ it exists $s' \in \Delta(S_1 \bigcap S_2)$ s.t $s' \notin S_1 \bigcap S_2$.

In any case, we get that a sequence $s' \notin S_1 \bigcap S_2$ and at the same time $s' \in \Delta(S_1 \bigcap S_2)$. But we have supposed that S_1 and S_2 are two closed set of sequences, and we have that by property of monotony of the closure operator Δ:

1/ $S_1 \bigcap S_2 \preceq S_1 \Rightarrow \Delta(S_1 \bigcap S_2) \preceq \Delta(S_1) = S_1$, and

2/ $S_1 \bigcap S_2 \preceq S_2 \Rightarrow \Delta(S_1 \bigcap S_2) \preceq \Delta(S_2) = S_2$

Then we get that $s' \in S_1$ and $s' \in S_2$, which contradicts the fact that $s' \notin S_1 \bigcap S_2$. Intersection of extents can be proved similarly. □

Appendix C

Following propositions 4 and 5 describe two important properties that hold in our lattice model.

Proposition 4 *For any given sequence s, it always true that $\{s\} \preceq \Delta(\{s\})$.*

Proof. Let $O = \psi(\{s\})$, then by definition of the derivation operator ψ we have that $\forall\ o \in O$, s is contained in o. If we now calculate the intersection of all these objects O, we get $\phi(O) = S$, where $\forall\ s' \in S$, s' is maximal contained in O.

So, $\phi(\psi(\{s\}))$ is a set of sequences s' such that each s' is maximal contained in all $o \in O$. Since we know by definition that s is also a sequence contained in all $o \in O$, then, we can conclude that at least one of these maximal $s' \in \phi(\psi(\{s\})) = \Delta(\{s\})$ is a superset of s, that is, at least exists $s' \in \Delta(\{s\})$ such that $s \subseteq s'$.

□

Proposition 5 *For S a closed set of sequences, we have that $\nexists s, s'$ s.t $s \in S$, $s' \in S$ and $s \subseteq s'$.*

Proof. If S is a closed set of sequences, then $\Delta(S) = S$ which can be rewritten as $\phi(\psi(S)) = S$, that is, $O = \psi(S)$ and $S = \phi(O)$. So, $\forall s \in S$, s is maximal contained in o, $\forall o \in O$. It means that all the sequences $s \in S$ are *maximal* by definition of our derivation operators, and it cannot exist $s' \subseteq s$ and at the same time $s' \in S$ and $s \in S$, otherwise s' would not be maximal.

□

We see with this proof that the fact that the derivation operator $\phi(O)$ is defined to return a set of *maximal* sequences common to all O is important otherwise we could have that two different sequences (but being one the inclusion of the other), can belong to the same set of closed sequences S. Now we have the tools to prove the following lemma, that characterizes the relation between stable sequences and our lattice model.

Lemma 1 *Let S be a closed set of sequences, then $\forall s \in S$, s is an stable sequence.*

Proof. Let S be the closed set of sequences of the proposition, then we have by definition that $S = \Delta(S)$, that is, $\psi(S) = O$ and $\phi(O) = S$. Now, for each one of the single sequence $s \in S$ we can find its closure as a single element of a set, i.e, we want to find the result of $\Delta(\{s\}) = \phi(\psi(\{s\}))$ to see if s is stable.

If we examine the first the part $\psi(\{s\})$, and suppose that $\psi(\{s\}) = O'$.

– If $O' = O$, then $\phi(\psi(\{s\})) = S$, and since $s \in S$ then we have that s is stable.
– If $O \subset O'$ (since it could well be that $\{s\}$ alone would be contained in more objects than S, since $\{s\} \preceq S$), then by definition of Galois connection we have that $O \subseteq O' \Rightarrow \phi(O') \preceq \phi(O)$. Thus, $\phi(O') \preceq S$, which can be rewritten as $\phi(\psi\{s\}) \preceq S$. Since $s \in S$, then we get along with proposition 4 the following relation on the sets $\{s\} \preceq \phi(\psi(\{s\})) \preceq S$; finally, by proposition 5 we know that s is *maximal* in the set S and $\nexists s' \in S$ s.t $s \subseteq s'$. So, $s \in \phi(\psi(\{s\})) = \Delta(\{s\})$, and we conclude that s is stable.
– It cannot occur that $O' \subset O$, since $\{s\} \preceq S$ and so $\{s\}$ has to be contained in at least the same objects where S is contained.

We can apply the same reasoning to all $s \in S$.

□

Next theorem serves to relate CloSpan to our lattice model.

Theorem 1 *The set of all stable sequences is exactly the set of closed sequential patterns extracted by the algorithm CloSpan [9].*

Proof.

$\Rightarrow$) If a sequence s is stable then $s \in \phi(\psi(\{s\}))$. This implies that: $1/s$ is a subsequence of o for all $o \in O = \psi(\{s\})$, then, $|O| = support(s)$ by definition of ψ; and, $2/$ s will be *maximal* in objects O because $s \in \phi(O)$ by definiton of ϕ.

Then we can say that, for this stable s, $\nexists s'$ $s.t$ $s \subseteq s'$ and s' is also contained in o $\forall o \in O$, which implies that $\nexists s'$ $s.t$ $s \subseteq s'$ and $support(s') = |O|$. Since we have stated that $|O| = support(s)$, then, $\nexists s'$ $s.t$ $s \subseteq s'$ and $support(s') = support(s)$. Thus, $s \in CS$.

$\Leftarrow$) Let s be a sequence belonging to set CS and let T be the set of all transactions in the database where this s appears ($|T| = support(s)$). Then, we can say that $\nexists s'$ $s.t$ $s \subseteq s'$ and $support(s') = |T|$ by definition of CS, which implies that, $\nexists s'$ $s.t$ $s \subseteq s'$ and s' appears in the set of transactions T; so, s is the *maximal* contained in the set of transactions T.

But we have defined a context $\mathbb{K}$ where from each transaction in T we can create an object o in O: let O be the set of objects stemming from transactions T in the original database, then, we can rewrite "s maximal contained in the transactions T" as $s \in \phi(O)$ and $O = \psi(\{s\})$, which implies that $s \in \phi(\psi(\{s\}))$; so, s is stable.

□