Information Sciences Letters
An International Journal

http://dx.doi.org/10.18576/isl/121118

On Construction of Tri-Concept Lattices

S. El-Assar[1,], A. Badawy[1], I. Gad[1] and S. Emad[2]*

[1]Department of Mathematics, Faculty of Science, Tanta University, Tanta, Egypt
[2]Department of Basic Science, Canadian International Collage "CIC", New Cairo, Egypt

Received: 16 June 2023, Revised: 16 Oct. 2023, Accepted: 28 Oct. 2023
Published online: 1 Nov. 2023

Abstract: The main point is to define the structure of a Tri-Concept lattice to deal with data given by different sources and represent it by less complex structures without loosing knowledge. We suggest the algorithm TRI-NEST to form the nested diagrams corresponding to the Tri-Concept lattices. Adding the ICE-T algorithm enables us to generate all frequently closed concepts, which leads to simplifying the Tri-Concept lattices and using the Iceberg Concept lattices as a reduction method to the big data while preserving all information.

Keywords: Trilattices; Multi-valued logic; Concept lattices; Iceberg Concept lattice and Nested Diagrams; Tri-Concept lattices.
2020 MR subject classification 03G10, 06D50, 03B50.

1 Introduction

R. Wille [1] established Formal Concept Analysis (FCA) as a combination of the theory of lattices and conceptual thinking. It has been evolved into a robust approach for data analysis, information retrieval, data mining, software engineering, and knowledge discovery (refer to [2, 3]). Moreover, FCA is a knowledge-processing task model. It deals with data organized into three groups: objects (G), attributes (M), and the relation (I) between them. Such data is represented by a dyadic context $\mathcal{K} \coloneqq (G, M, I)$.

The main structure represented by that context is the concept lattice $\mathcal{B}(G, M, I)$, which arose from the collection of all concepts (N, L); $N \subseteq G, L \subseteq M$ and the partial ordering $(N, L) \leq (N^*, L^*)$ iff $N \subseteq N^*$ $(L \supseteq L^*)$. That concept lattice is complete. Actually, any complete lattice can be represented by a concept lattice of a dyadic context (see [2]). Several algorithms have been suggested for creating the concept lattices (see [4]).

Iceberg concept lattices have been recommended to deal with big data as an appealing approach for data reduction and representation (see [5, 6]). The authors in [5, 6] display the uppermost part of a concept lattice. In addition, they can be utilized as a conceptual clustering approach to figure out the frequent item sets in large datasets.

Choosing an upper part of the lattice does not address that issue since it includes similar nodes. So, nesting is another method for improving the readability of finite diagrams.

Nested diagrams offer no actual reduction and so involve no information loss. They represent the overall structure of concept lattices and may also be used to create a different visualization of the iceberg concept lattices. Nested diagrams split the attribute set into two "or more" subsections, build a concept lattice, and put a diagram of one lattice within each other's node (see [2]). Iceberg concept lattices and Nested diagrams can be merged to form a more reduced diagram without loss of any information.

An approach to the triadic case has been proposed depending on the experience gained from applying concept lattices (see [8, 9]). The foundation of this approach is rooted in the basic notion of a triadic context, which is illustrated by a quadruple (G, M, B, Y). In this context, G, M, and B are sets while Y is a ternary relation among them, with $Y \subseteq G \times M \times B$. The components of G, M, and B are referred to as objects, attributes, and conditions, respectively. When $(g, m, b) \in Y$, it implies that under condition b, the object g possesses the attribute m. A triplet (A_1, A_2, A_3) with $A_1 \times A_2 \times A_3 \subseteq Y$ represents the triadic concept in the context (G, M, B, Y) and is the maximal set based on component-wise inclusion. The three quazi-orders that shape the triadic concepts are based on the inclusion order within each component. As in the case of a dyadic context, R. Wille has analyzed and graphically represented the ordinal structure of the triadic concepts (see [8]).

This work aims to define Tri-Concept lattices that handle data from various sources. The data is represented in the form of iceberg concept lattices and nested diagrams to present it more succinctly. A more readable diagram of the Tri-Concept

*Corresponding author e-mail: sanaa.elassar@science.tanta.edu.eg

lattice can be obtained using the direct product of bi-concept lattices and the representation by nested diagrams, which facilitates knowledge extraction. The TRI-NEST algorithm is suggested to construct the nested diagram of Tri-Concept lattices. Additionally, Iceberg concept lattices are formed corresponding to subcontexts, and a more reduced diagram of the Tri-Concept lattices is obtained using the TRI-NEST algorithm.

To compute all frequent concepts, the ICE-T Algorithm is introduced. It enables us to get a reduced form of the Tri-Concept lattice. Also, to get frequent item sets and generators, the Snow Algorithm [6] is applied. Therefore, the iceberg concept lattice is obtained for each subcontext, and then the corresponding Tri-Concept lattice is built using the TRI-NEST algorithm.

In the following, Section 2 provides the basic definitions and terminologies required in the subsequent sections. We define in Subsection 3.1 the structure of Tri-Concept lattices and give a representation of bi-concept lattices. Adding the algorithm TRI-NEST, the corresponding nested diagram can be described. Applying the results on a real data explains the ideas. The Subsections 3.2, 3.3, and 3.4 show how the splitting of a context leads us to construct iceberg concept lattices using different ways and contain the suggested ICE-T algorithm.

2 Preliminaries:

The basic definitions, terminologies, and notions are reviewed to be used in the following subsections.

Bilattices

A lattice $\mathfrak{F}$ is defined by a pair $\langle \mathfrak{F}\,; \leq \rangle$, where $\mathfrak{F}$ is a non-empty set and $\leq$ denotes a partial order, ensures that both the supremum (Supp$\{o, p\}$) and infimum (Inf$\{o, p\}$) exist for all $o, p \in \mathfrak{F}$.

Using $o \vee p$ and $o \wedge p$ to represent (Supp$\{o, p\}$) and (Inf$\{o, p\}$), respectively, the lattice $\mathfrak{F}$ can be defined as the algebraic structure $\langle \mathfrak{F}\,; \vee, \wedge \rangle$, where the binary operations $\vee$ and $\wedge$ satisfy the associative, commutative, idempotent, and absorption laws.

A lattice is complete if each subset $\mathcal{W}$, (non-empty) of it, has both supremum and infimum, that is, $\bigvee \mathcal{W}$ and $\bigwedge \mathcal{W}$ exist for all $\mathcal{W} \subseteq \mathfrak{F}$. A lattice's upper and lower elements are denoted by 1 and 0, respectively.

A Boolean lattice is defined as a special kind of distributive lattices. The simplest non-trivial Boolean lattice is represented by 2. It consists of $\{0, 1\}$ (a two-element chain). It arises frequently in logic and computer science as the algebra of truth-values $V = \{F, T\}$. The symbols F and T are used instead of 0 and 1 (for more details, see [10]).

Many-valued logics (MVL) extend this condition by permitting V to be the set of degrees of truth, which can be more or less arbitrary.

Generalizing the logic of two valued systems as given in [11–13], bilattices have been found. They fall in the category of multi-valued logics and possess an algebraic structure that makes them useful (for extra details, see [14]). Belnap's four-valued logic [15] extends the classical two-valued logic set $\{F, T\}$ to the power set $V = P(\{F, T\})$ with cardinality $|V| = 2^2 = 4$.

A pre-bilattice $\langle \mathfrak{G};\, \leq_t, \leq_k \rangle$ is essentially a space of extended truth-values, in which $\mathfrak{G}$ is a set (which is not empty) with two lattice orderings, both of which establish the lattice's structure. The first ordering $\leq_t$ represents the degree of truth, whereas the second ordering $\leq_k$ represents the degree of information or knowledge.

Bilattices are more than just a space with two lattice orderings; the proposed connectors make them an intriguing structure. The symbols B and N will denote the case of having true and false concepts simultaneously and the case of being neither true nor false, respectively. Bilattices are pre-bilattices equipped with a negation operation.

Definition 2.1 [11] A bilattice is constituted by a non-empty set $\mathfrak{G}$, two partial orders, $\leq_t$ *and* $\leq_k$ defined on it, and a self-mapping operation, $\sim$, it can be represented as $\langle \mathfrak{G};\, \leq_t, \leq_k, \sim \rangle$. The following conditions hold

1) Both $\leq_t$ and $\leq_k$ result in $\mathfrak{G}$ being a complete lattice,
2) If $o \leq_t p$ then $\sim p \leq_t \sim o$,
3) If $o \leq_k p$ then $\sim o \leq_k \sim p$,
4) $\sim\sim o = o$.

The symbols $\circ$ and $+$ represent meet and join corresponding to the relation $\leq_t$. While $\otimes$ and $\oplus$ denote the operation of meet and join related to the knowledge order $\leq_k$ (see [16]).

If all meet and join operators related to the orders $\leq_t$ and $\leq_k$ exist, then the pre-bilattice $\langle \mathfrak{G};\, \leq_t, \leq_k \rangle$ is complete. The infinitary meet and join concerning $\leq_t$ and $\leq_k$ are denoted by $\wedge$, $\vee$, $\prod$ and $\sum$, respectively.

An algebra $\mathfrak{G} = \langle \mathfrak{G}; \circ, +, \otimes, \oplus \rangle$ is referred to as a pre-bilattice if $\langle \mathfrak{G}; \circ, + \rangle$ and $\langle \mathfrak{G}; \otimes, \oplus \rangle$ are lattices.

Definition 2.2 [17] A pre-bilattice is regarded as interlaced when all the four operations $\{\circ, +, \otimes, \oplus\}$ are monotonic in both orders $\leq_t$ and $\leq_k$. It implies that the following quasi-equations hold:

1) If $o \leq_t o^*$ then $o \otimes n \leq_k o^* \otimes n$,
2) If $o \leq_t o^*$ then $o \oplus n \leq_k o^* \oplus n$,
3) If $o \leq_k o^*$ then $o \circ n \leq_t o^* \circ n$,
4) If $o \leq_k o^*$ then $o + n \leq_t o^* + n$.

Definition 2.3 [14] Let $\mathfrak{F} = \langle \mathfrak{F}; \wedge, \vee \rangle$ and $\mathfrak{F}^* = \langle \mathfrak{F}^*; \wedge^*, \vee^* \rangle$ be the lattices associated with $\leq_t$ and $\leq_k$. The structure $\mathfrak{F} \odot \mathfrak{F}^* = \langle \mathfrak{F} \times \mathfrak{F}^*; \circ, +, \otimes, \oplus \rangle$ is an interlaced pre-bilattice, where;

$\langle o, p \rangle \circ \langle o^*, p^* \rangle = \langle o \wedge o^*, p \vee^* p^* \rangle$,

$\langle o, p \rangle + \langle o^*, p^* \rangle = \langle o \vee o^*, p \wedge^* p^* \rangle$,

$\langle o, p \rangle \otimes \langle o^*, p^* \rangle = \langle o \wedge o^*, p \wedge^* p^* \rangle$,

$\langle o, p \rangle \oplus \langle o^*, p^* \rangle = \langle o \vee o^*, p \vee^* p^* \rangle$,

$\forall \langle o, p \rangle, \langle o^*, p^* \rangle \in \mathfrak{F} \times \mathfrak{F}^*$.

If $\mathfrak{F}$ is isomorphic to $\mathfrak{F}^*$ ($\mathfrak{F} \cong \mathfrak{F}^*$), then it is possible to define the negation operation $\sim$ in $\mathfrak{F} \odot \mathfrak{F}$. Thus, we speak of the product bilattices rather than the product pre-bilattices (see [14]). Negation is defined as $\sim\langle o, p \rangle = \langle p, o \rangle$.

(Pre-)bilattices are represented as a product of lattices (see [14]).

Theorem 2.1 [14]

Let $\mathfrak{G}$ be a bounded pre-bilattice. The following are equivalent

1) $\mathfrak{G}$ is an interlaced pre-bilattice.
2) There are two bounded lattices $\mathfrak{F}$ and $\mathfrak{F}^*$, such that the bilattice $\mathfrak{G}$ is isomorphic to $\mathfrak{F} \odot \mathfrak{F}^*$.

Theorem 2.2 [14]

Let us have a bilattice $\mathfrak{G}$, which is bounded. The following are equivalent

1) $\mathfrak{G}$ is an interlaced bilattice.
2) There is a bounded lattice $\mathfrak{F}$, such that the bilattice $\mathfrak{G}$ is isomorphic to $\mathfrak{F} \odot \mathfrak{F}$.

Trilattices

In [18], Y. Shramko introduced trilattices and defined it on a generalized space of the sixteen-valued logic "Constructive Logic", which represents all possible combinations of truth-values.

As noted in [15], The sixteen-valued truth-degree structure is the base of the multi-valued logic (MVL) systems. The power set $V \coloneqq P\big(P(\{F, T\})\big)$ yields the underlying set of truth-degrees, such that, $|V| = 2^{2^2} = 16$. Recently, the structure of trilattices has been utilized to present a variety of many valued systems that generalize the logic of Belnap.

Trilattices have been proposed as the logic of how a network of several computers should operate effectively while dealing with incomplete and contradictory information (refer to [19]). The trilattice structure can be considered an algebra, incorporating three sets of lattice orders, modeling constructive orders, truth, and information (refer to [18, 20]).

The relational structure $\langle \mathfrak{K}; \leq_t, \leq_k, \leq_c \rangle$ defines a trilattice, where the orders can be understood differently (see [20]). As an algebra, the trilattice can be alternatively represented by the system $\langle \mathfrak{K}; \circ, +, \otimes, \oplus, \dagger, \ddagger \rangle$, which comprises the three reducts $\mathfrak{K}_1 = \langle \mathfrak{K}; \circ, + \rangle$, $\mathfrak{K}_2 = \langle \mathfrak{K}; \otimes, \oplus \rangle$ and $\mathfrak{K}_3 = \langle \mathfrak{K}; \dagger, \ddagger \rangle$ are lattices. These reducts correspond to the orders $\leq_t, \leq_k$, and $\leq_c$ that represent truth, knowledge and constructive data, respectively. All the reducts inherit the property of being interlaced (for further information, see [19]). As bilattices, a trilattice is deemed interlaced if all six-lattice operations maintain monotonicity concerning all orders.

Trilattices can be treated in a more powerful natural form due to the six operations $\circ, +, \otimes, \oplus$,$\dagger$, and $\ddagger$. There are 30 potential distributive laws. A trilattice $\mathfrak{K} = \langle \mathfrak{K}; \circ, +, \otimes, \oplus, \dagger, \ddagger \rangle$ is called distributive if, for all elements in $\mathfrak{K}$, the distributive laws hold (as noted in [19]). Any distributive trilattice is obviously interlaced. The converse is not true since non-distributive interlaced trilattices exist.

Trilattices have been constructed and represented by a pair of pre-bilattices (see [19]). It has been proven that any interlaced pre-bilattice has the form $\mathfrak{F} \odot \mathfrak{F}^*$, where $\mathfrak{F}$ and $\mathfrak{F}^*$ are lattices. Thus, any interlaced trilattice, nevertheless, has the form $\mathfrak{G} \odot \mathfrak{G}^*$, where $\mathfrak{G}$ and $\mathfrak{G}^*$ are interlaced pre-bilattices. Combining these results, every interlaced trilattice $\mathfrak{K} = \langle \mathfrak{K}; \leq_t, \leq_k, \leq_c \rangle$ can be understood as the product $(\mathfrak{F} \odot \mathfrak{F}^*) \odot (\mathfrak{F}^{**} \odot \mathfrak{F}^{***})$, such that $\mathfrak{F} \odot \mathfrak{F}^*$ represents the bilattice $\mathfrak{G}$ and $\mathfrak{F}^{**} \odot \mathfrak{F}^{***}$ represents the bilattice $\mathfrak{G}^*$. Thus, the trilattice $\mathfrak{K}$ can be described by $\mathfrak{K} = \mathfrak{G} \odot \mathfrak{G}^*$ or $\mathfrak{K} = (\mathfrak{F} \odot \mathfrak{F}^*) \odot (\mathfrak{F}^{**} \odot \mathfrak{F}^{***})$.

Theorem 2.3 [21] (Representation) The trilattice $\mathfrak{K}$ is interlaced if and only if $\mathfrak{K}$ is isomorphic to a product $\mathfrak{G} \odot \mathfrak{G}^*$, where $\mathfrak{G}$ and $\mathfrak{G}^*$ are pre-bilattices.

Formal Concept Analysis

Concept Lattices

In 1980, R. Wille introduced the mathematical theory of Formal Concept Analysis (FCA) (see [1]). The formalization and hierarchy of concepts are the primary focus of FCA, and it has found applications in various fields, such as software engineering, knowledge discovery, and information retrieval.

Classifying a given data in the form of objects G, attributes M, and a relation I between them, we get the context $\mathcal{K} \coloneqq (G, M, I)$; where $I \subseteq G \times M$ and $(g, m) \in I$ means gIm, where the object g has the attribute m.

Two derivation operators were defined for arbitrary $N \subseteq G$ and $L \subseteq M$ as

$N^{\#} \coloneqq \{m \in M \mid gIm \; \forall g \in N\}$,
$L^{\#} \coloneqq \{g \in G \mid gIm \; \forall m \in L\}$ (see [3]).

The two derivation operators fulfill the subsequent conditions;

1) $F_1 \subseteq F_2 \Longrightarrow F_1^{\#} \supseteq F_2^{\#}$,
2) $F \subseteq F^{\#\#}$,
3) $F^{\#\#\#} = F^{\#}$.

The pair (N, L) describes a concept with $N \subseteq G$, $L \subseteq M$, $N = L^{\#}$, and $L = N^{\#}$; where N and L denote the formal concept's extent and intent, respectively. The subconcept-superconcept-relation is mathematically represented as $(N, L) \leq (N^*, L^*)$ iff $N \subseteq N^*$ (or $L \supseteq L^*$) (see [3]). The notation $\mathcal{B}(\mathcal{K})$ indicates the set of all formal concepts of the context $\mathcal{K} \coloneqq (G, M, I)$ with the corresponding order relation.

As mentioned by R. Wille in [2], concept lattices product can be illustrated as follows: let $\mathcal{K}_i \coloneqq (G, M_i, I \cap G \times M_i), i = \{1, 2\}$ be two subcontexts, and $\mathcal{B}(\mathcal{K}_i) = \mathcal{B}(G, M_i, I \cap G \times M_i)$ be the corresponding sets of all concepts and $((L \cap M_i)^{\#}, L \cap M_i)$ be a concept in the context $\mathcal{K}_i$. The direct product of the two concept lattices $\mathcal{B}(G, M_1, I \cap G \times M_1)$ and $\mathcal{B}(G, M_2, I \cap G \times M_2)$, is denoted by $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2))$ such that, $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2)) =$

$$\left\{ \begin{matrix} \left(((L \cap M_1)^{\#}, L \cap M_1), ((L \cap M_2)^{\#}, L \cap M_2)\right) : \left(((L \cap M_1)^{\#}, L \cap M_1), ((L \cap M_2)^{\#}, L \cap M_2)\right) \\ \in \mathcal{B}(G, M_1, I \cap G \times M_1) \times \mathcal{B}(G, M_2, I \cap G \times M_2) \end{matrix} \right\},$$

it is isomorphic to $\mathcal{B}(G, M_1 \dot\cup M_2, I_1 \dot\cup I_2 \dot\cup (G_1 \times M_2) \dot\cup (G_2 \times M_1))$, where $\dot\cup$ be the disjoint union operation.

A main theorem of concept lattices is stated in the following

Theorem 2.4 [1] Considering the context $\mathcal{K} \coloneqq (G, M, I)$; the collection $\mathcal{B}(\mathcal{K})$ of all of its concepts forms a complete lattice in which:

$\bigwedge_{t \in T} (N_t, L_t) = (\bigcap_{t \in T} N_t, (\bigcup_{t \in T} L_t)^{\#\#})$,

$\bigvee_{t \in T} (N_t, L_t) = ((\bigcup_{t \in T} N_t)^{\#\#}, \bigcap_{t \in T} L_t)$.

Generally, any complete lattice can be represented as a concept lattice (see [1]).

Iceberg Concept Lattices

Iceberg concept lattice is a conceptual clustering technique with a suitable visualization method for analyzing large databases. It shows the top-most elements in the diagram (see [5]).

First, we recall some definitions.

Definition 2.4 [5] Let $\mathcal{K} \coloneqq (G, M, I)$ be a context, and let $L \subseteq M$. The value $\frac{|L^{\#}|}{|G|}$ is the support count of L (supp(L)). If that value is at least a minsupp, then L is a frequent itemset, where the minsupp is a threshold belonging to $[0, 1]$.

A concept is frequent if it has a frequent intent, where the collection of all frequent concepts is referred to as the iceberg concept lattice (sometimes, it is only a semilattice).

Iceberg concept lattices can also be constructed from frequent closures using generators.

Definition 2.5 [6] If an itemset N has no proper superset (subset) with identical support, it is considered a closed (generator) itemset.

The maximal superset X of an itemset N is called the closure operator of it.

A set $X \subset M$ is classified as a minimal generator (mingen) of a closed set $L \subseteq M$ only if X is the smallest subset of L satisfying the condition $X^{\#\#\#} = L$ (see [22]).

The precedence relation $\prec$ between frequent closed itemsets (FCIs) is defined as follows:

$N \prec X$ iff (i) N is a subset of X ($N \subset X$), and (ii) there exists no O that is a subset of X and a superset of N ($N \subset O \subset X$). Then, N is called the predecessor of X (see [6]).

The FCI family of a dataset in combination with the relation $\prec$ produces the iceberg concept lattice. In the context of a ground set N (where $N \subseteq \wp(N)$), a blocker of N is a set $X \subseteq N$ that has a non-empty intersection with every member of N ($\forall\, O \in N, X \cap O \neq \emptyset$). A minimal blocker cannot be expressed as a subset of any other blocker. The closure lattice employs blockers through related faces, which correspond to the disparities between two adjacent closures present in the lattice. More specifically, given two CIs N *and* N^* such that $N \prec N^*$, its associated face is $F = N^*/N$ (see [6]).

Definition 2.6 [6] A hypergraph is composed of a finite set $V = \{\upsilon_1, \upsilon_2, \dots, \upsilon_n\}$ and a group of subsets of V, denoted as ξ. The vertices constitute the elements of V, while the edges refer to the elements of ξ.

Definition 2.7 [6] For a hypergraph $\mathcal{H} = (V, \xi)$, a set $O \subseteq V$ is called a transversal of $\mathcal{H}$ if it intersects with all edges of $\mathcal{H}$, meaning $\forall\, E \in \xi : O \cap E \neq \emptyset$. If no smaller subset O^* of O can also serve as a transversal, then the transversal O is deemed minimal.

By taking the difference between a closed itemset (CI) N and a face, a predecessor of N can be obtained within the closure lattice.

The Snow Algorithm, initially presented by L. Szathmary et al. [6], is a technique used for determining consequence links of the frequent closed itemsets (FCIs) by generating faces from frequent generators (FGs). Therefore, the algorithm accepts frequent closed itemsets (FCIs) and their corresponding frequent generators (FGs) as input.

Nested Diagrams

Nested diagrams have been proposed as an effective technique for determining and illustrating large concept lattices. This method utilizes the correspondence of direct products to create a diagram (we refer to [2]).

The Basic Theorem of nested diagrams states that:

Theorem 2.5 [2] Considering the context (G, M, I). Let $M = M_1 \cup M_2$. The correspondence

$$(N,\, L) \mapsto \left(((L \cap M_1)^{\#},\, L \cap M_1),\, ((L \cap M_2)^{\#},\, L \cap M_2) \right)$$

gives a $\vee$-preserving order embedding of $\mathcal{B}(G,\, M,\, I)$ into the product of $\mathcal{B}(G,\, M_1, I \cap G \times M_1)$ *and* $\mathcal{B}(G,\, M_2, I \cap G \times M_2)$. The component maps $(N,\, L) \mapsto ((L \cap M_i)^{\#},\, L \cap M_i)$ are surjective on $\mathcal{B}(G,\, M_i,\, I \cap G \times M_i)$.

To construct nested diagrams, the attribute set of the context is partitioned into parts. The concept lattices of the corresponding subcontexts are created. Each subcontext $\mathcal{K}_i = \langle G, M_i, I \cap G \times M_i \rangle, i \in \{1, 2\}$ is represented in the diagram with the standard object and attribute labels. Finally, a nested diagram is sketched to describe the product of the concept lattices $\mathcal{B}(\mathcal{K}_i)$ (see [2]).

3 Main Results

In this part, we introduce the notion of a Tri-Concept lattice and represent it by simpler structures. Application using real data shows the benefits of constructing such a structure.

Construction of Tri-Concept Lattices and TRI-NEST Algorithm

In [23], S. El-Assar et al. introduced the notion of bi-concept lattices as algebras corresponding to the data given by two contexts. According to the work of U. Riviecco [19], trilattice can also be represented as a direct product of two bilattices. It leads us to define the structure, corresponding to the data described by different contexts using that representation.

Suppose $\mathcal{K}_1 = (G, M_1, I_1)$ and $\mathcal{K}_2 = (G, M_2, I_2)$ be two contexts and $\mathcal{B}(\mathcal{K}_1)$ and $\mathcal{B}(\mathcal{K}_2)$ be their concept lattices, denoted as $\mathcal{B}(\mathcal{K}_i) = \langle \mathcal{B}(\mathcal{K}_i), \wedge_i, \vee_i, 0_i, 1_i \rangle$ for $i = \{1,2\}$. Then, the structure $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2)) = \langle \mathcal{B}(\mathcal{K}_1) \times \mathcal{B}(\mathcal{K}_2); \circ, +, \bot, \top, \otimes, \oplus, \bot', \top' \rangle$ forms a bi-concept lattice, where the operations are defined as follows:

$$((N,L),(N^*,L^*)) \circ ((O,P),(O^*,P^*)) = ((N,L) \wedge_1 (O,P), (N^*,L^*) \vee_2 (O^*,P^*)),$$

$$((N,L),(N^*,L^*)) + ((O,P),(O^*,P^*)) = ((N,L) \vee_1 (O,P), (N^*,L^*) \wedge_2 (O^*,P^*)),$$

$$((N,L),(N^*,L^*)) \otimes ((O,P),(O^*,P^*)) = ((N,L) \wedge_1 (O,P), (N^*,L^*) \wedge_2 (O^*,P^*)),$$

$$((N,L),(N^*,L^*)) \oplus ((O,P),(O^*,P^*)) = ((N,L) \vee_1 (O,P), (N^*,L^*) \vee_2 (O^*,P^*)).$$

$$\bot = (0,1^*), \qquad \top = (1,0^*), \qquad \bot' = (0,0^*), \qquad \top' = (1,1^*).$$

$\forall \; ((N,L),(N^*,L^*)) \; and \; ((O,P),(O^*,P^*)) \in \mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2))$ (see [23]).

According to the Representation Theorem of trilattices (Theorem 2.3), as a product of bilattices, we can describe Tri-Concept lattices as follows.

Let $\mathcal{K}_i = (G, M_i, I_i), i = \{1,2,3,4\}$ be four contexts, the bi-concept lattices $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2))$ and $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))$ are the corresponding bi-concept lattices. Then the structure $\mathfrak{Z}\left(\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2)), \mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))\right)$ forms a Tri-Concept lattice concerning the operations $\circ, +, \bot, \top, \otimes, \oplus, \bot', \top', \dagger, \ddagger, \bot''$ and $\top''$, where

$$\left(((N,L),(N^*,L^*)),((N^{**},L^{**}),(N^{***},L^{***}))\right) \circ \left(((O,P),(O^*,P^*)),((O^{**},P^{**}),(O^{***},P^{***}))\right) =$$
$$\left(((N,L) \sqcap_1 (O,P),(N^*,L^*) \sqcup_2 (O^*,P^*)),((N^{**},L^{**}) \sqcap_3 (O^{**},P^{**}),(N^{***},L^{***}) \sqcup_4 (O^{***},P^{***}))\right),$$

$$\left(((N,L),(N^*,L^*)),((N^{**},L^{**}),(N^{***},L^{***}))\right) + \left(((O,P),(O^*,P^*)),((O^{**},P^{**}),(O^{***},P^{***}))\right) =$$
$$\left(((N,L) \sqcup_1 (O,P),(N^*,L^*) \sqcap_2 (O^*,P^*)),((N^{**},L^{**}) \sqcup_3 (O^{**},P^{**}),(N^{***},L^{***}) \sqcap_4 (O^{***},P^{***}))\right),$$

$$\left(((N,L),(N^*,L^*)),((N^{**},L^{**}),(N^{***},L^{***}))\right) \otimes \left(((O,P),(O^*,P^*)),((O^{**},P^{**}),(O^{***},P^{***}))\right) =$$
$$\left(((N,L) \sqcap_1 (O,P),(N^*,L^*) \sqcap_2 (O^*,P^*)),((N^{**},L^{**}) \sqcap_3 (O^{**},P^{**}),(N^{***},L^{***}) \sqcap_4 (O^{***},P^{***}))\right),$$

$$\left(((N,L),(N^*,L^*)),((N^{**},L^{**}),(N^{***},L^{***}))\right) \oplus \left(((O,P),(O^*,P^*)),((O^{**},P^{**}),(O^{***},P^{***}))\right) =$$
$$\left(((N,L) \sqcup_1 (O,P),(N^*,L^*) \sqcup_2 (O^*,P^*)),((N^{**},L^{**}) \sqcup_3 (O^{**},P^{**}),(N^{***},L^{***}) \sqcup_4 (O^{***},P^{***}))\right),$$

$$\left(((N,L),(N^*,L^*)),((N^{**},L^{**}),(N^{***},L^{***}))\right) \dagger \left(((O,P),(O^*,P^*)),((O^{**},P^{**}),(O^{***},P^{***}))\right) =$$
$$\left(((N,L) \sqcap_1 (O,P),(N^*,L^*) \sqcap_2 (O^*,P^*)),((N^{**},L^{**}) \sqcup_3 (O^{**},P^{**}),(N^{***},L^{***}) \sqcup_4 (O^{***},P^{***}))\right),$$

$$\left(((N,L),(N^*,L^*)),((N^{**},L^{**}),(N^{***},L^{***}))\right) \ddagger \left(((O,P),(O^*,P^*)),((O^{**},P^{**}),(O^{***},P^{***}))\right) =$$
$$\left(((N,L) \sqcup_1 (O,P),(N^*,L^*) \sqcup_2 (O^*,P^*)),((N^{**},L^{**}) \sqcap_3 (O^{**},P^{**}),(N^{***},L^{***}) \sqcap_4 (O^{***},P^{***}))\right).$$

$$\bot = ((0,1^*),(0^{**},1^{***})), \qquad \bot' = ((0,0^*),(0^{**},0^{***})), \qquad \bot'' = ((0,0^*),(1^{**},1^{***})),$$

$$\top = ((1,0^*),(1^{**},0^{***})), \qquad \top' = ((1,1^*),(1^{**},1^{***})), \qquad \top'' = ((1,1^*),(0^{**},0^{***})).$$

$\forall \left(((N,L),(N^*,L^*)),((N^{**},L^{**}),(N^{***},L^{***}))\right)$ & $\left(((O,P),(O^*,P^*)),((O^{**},P^{**}),(O^{***},P^{***}))\right)$ are concepts from $\mathfrak{Z}\left(\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2)), \mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))\right)$.

Considering the Fundamental Theorem of Concept Lattices (see [2]), we can formulate

Theorem 3.1 Any Tri-Concept lattice is complete.

Using the Representation by nested diagrams (see [2]), we can formulate the Representation Theorem of Tri-Concept Lattices as a nested diagram. The outer lattice of the diagram represents the bi-concept lattice $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2)) = \mathcal{B}(\mathcal{K}_1) \times \mathcal{B}(\mathcal{K}_2)$ and describes the first component of the Tri-Concept lattice. While the inner lattice represents the bi-concept lattice $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4)) = \mathcal{B}(\mathcal{K}_3) \times \mathcal{B}(\mathcal{K}_4)$ and shows the second component of the Tri-Concept.

Theorem 3.2 Consider the context $\mathcal{K} = (G, M, I)$ and the subcontexts $\mathcal{K}_i's$, $i = \{1,2,3,4\}$. Let $\mathcal{B}(\mathcal{K})$ denote the set of all concepts in $\mathcal{K}$, and define a mapping from $\mathcal{B}(\mathcal{K})$ to the direct product of the bi-concept lattices $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2))$, and $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))$ as follows: $(N, L) \longrightarrow \Big(\big(((L \cap M_1)^{\#}, L \cap M_1), ((L \cap M_2)^{\#}, L \cap M_2)\big), \big(((L \cap M_3)^{\#}, L \cap M_3), ((L \cap M_4)^{\#}, L \cap M_4)\big)\Big)$

The map is a join-preserving order embedding. That correspondence maps $(N, L) \longrightarrow \Big(((L \cap M_i)^{\#}, L \cap M_i), \big((L \cap M_j)^{\#}, L \cap M_j\big)\Big)$ which is surjective, $i \in \{1, 2\}$ and $j \in \{3, 4\}$. Also, $(N, L) \longrightarrow ((L \cap M_k)^{\#}, L \cap M_k)$ is surjective on $\mathcal{B}(G, M_k, I \cap G \times M_k)$.

Proof: Applying the Basic Theorem of nested diagrams (Theorem 2.5), considering the pair (N, L) to be a concept in $\mathcal{K} = (G, M, I)$, then $L \cap M_k$ is an intent in $\mathcal{B}(G, M_k, I \cap G \times M_k)$, and let $L \cap M_i$ and $L \cap M_j$ be intents in $\mathcal{B}(G, M_i, I \cap G \times M_i)$, and $\mathcal{B}(G, M_j, I \cap G \times M_j)$, respectively. Therefore, $\big((L \cap M_j)^{\#}, L \cap M_j\big)$ is the intent of the concept $\Big(((L \cap M_i)^{\#}, L \cap M_i), \big((L \cap M_j)^{\#}, L \cap M_j\big)\Big)$ in the tri-concept lattice $\mathfrak{Z}\Big(\mathfrak{Y}(\mathcal{B}(G, M_i, I \cap G \times M_i)), \mathfrak{Y}\big(\mathcal{B}(G, M_j, I \cap G \times M_j)\big)\Big)$. The union of the objects and attributes in each part of the concept $\Big(((L \cap M_i)^{\#}, L \cap M_i), \big((L \cap M_j)^{\#}, L \cap M_j\big)\Big)$ yields L in the concept (N, L), i.e., the map is injective.

Let o be an intent of $\mathcal{B}(G, M_k, I \cap G \times M_k)$, then $L = o^{\#\#}$ is an intent of (G, M, I) with $L \cap M_k = o$, i.e., the image of the concept $(L^{\#}, L)$ of (G, M, I) under the k^{th} component map is the concept with the intent o; then the map is surjective on $\mathcal{B}(G, M_k, I \cap G \times M_k)$. Also, let $\big((p_i, o_i), (p_j, o_j)\big)$ be a concept in the tri-concept lattice $\mathfrak{Z}\Big(\mathfrak{Y}(\mathcal{B}(G, M_i, I \cap G \times M_i)), \mathfrak{Y}\big(\mathcal{B}(G, M_j, I \cap G \times M_j)\big)\Big)$. Then (p_j, o_j) is an intent of the concept $\big((p_i, o_i), (p_j, o_j)\big)$, then $L = (p_j, o_j)^{\#\#}$ is the intent of (G, M, I) with $\big((L \cap M_j)^{\#}, L \cap M_j\big) = (p_j, o_j)$, i.e., the image of the concept $(L^{\#}, L)$ of (G, M, I) under the i^{th} *and* j^{th} component map is the concept with the intent (p_j, o_j), then the map is surjective on the tri-concept lattice $\mathfrak{Z}\Big(\mathfrak{Y}(\mathcal{B}(G, M_i, I \cap G \times M_i)), \mathfrak{Y}\big(\mathcal{B}(G, M_j, I \cap G \times M_j)\big)\Big)$.

We suggest the algorithm TRI-NEST to construct Tri-Concept lattices using the representations by nested diagrams.

Algorithm 1: TRI-NEST $G' = [G_m, G_n] = \big[[G, G^*], [G^{**}, G^{***}]\big]$

Input:	Graphs G_m, G_n, G, G^*, G^{**} *and* G^{***}
Output:	A nested diagram G'

```
i = 1: #nodes of G
for node n_i in G
    add node n_i to G_m as n_im
    add G* in n_im
end for
j = 1: #nodes of G**
for node n_j in G**
    add node n_j to G_n as n_jn
    add G*** in n_jn
end for
k = 1: #nodes of G_m
for node n_k in G_m
```

add node n_k to G' as n_{kl}

add G_n in n_{kl}

end for

return

Illustrative Example 3.1 Let us have a sample of five patients with COVID from a dataset of 5434 patients, as given in Table 3.1, which includes various types of symptoms, signs (ordinary symptoms), and dangerous symptoms, diseases that make COVID worse and some causes of COVID infection.

Table 3.1: A Formal Context of COVID Dataset

	Some Causes of COVID Infection					A disease that makes COVID worse					Signs of COVID			Dangerous Symptoms				
	GT	**BC**	**JD**	**VI**	**FI**	**HD**	**Di**	**CL**	**As**	**HT**	**Fe**	**DC**	**Fa**	**BP**	**ST**	**RN**	**He**	**Ga**
I		×			×		×			×	×	×	×	×	×	×		×
II		×	×	×			×	×	×		×	×	×	×	×	×	×	×
III	×				×	×				×	×	×			×		×	×
IV		×		×	×		×		×	×	×	×	×	×				
V			×	×	×			×	×	×	×	×		×	×			×

Tables 3.2, 3.3, 3.4, and 3.5 describe four subcontexts of the context in Table 3.1.

	Some Causes of COVID Infection				
	GT	**BC**	**JD**	**VI**	**FI**
I		×			×
II		×	×	×	
III	×				×
IV		×		×	×
V			×	×	×

Table 3.2: $\mathcal{K}_1$

	A disease that makes COVID worse				
	HD	**Di**	**CL**	**As**	**HT**
I		×			×
II		×	×	×	
III	×				×
IV		×		×	×
V			×	×	×

Table 3.3: $\mathcal{K}_2$

	Signs of COVID		
	Fe	**DC**	**Fa**
I	×	×	×
II	×	×	×
III	×	×	
IV	×	×	×
V	×	×	

Table 3.4: $\mathcal{K}_3$

	Dangerous Symptoms				
	BP	**ST**	**RN**	**He**	**Ga**
I	×	×	×		×
II	×	×	×	×	×
III		×		×	×
IV	×				
V	×	×			×

Table 3.5: $\mathcal{K}_4$

For abbreviation, let

GT: Going on a Travel

BC: Being with COVID Patients

JD: Joining Different Gathering

VI: Visiting Infected Places

FI: Families attending Infected Places

HD: Heart Disease

Di: Diabetes

CL: Chronic Lung Disease

As: Asthma

HT: Hyper Tension

Fe: Fever

DC: Dry Cough

Fa: Fatigue

BP: Breathing Problems

ST: Sore Throat

RN: Running Nose

He: Headache

Ga: Gastrointestinal

The following represent objects G and the attributes $M_i, i = \{1,2,3,4\}$, in the four contexts,

$G =$ {Patient I, Patient II, Patient III, Patient IV, Patient V},

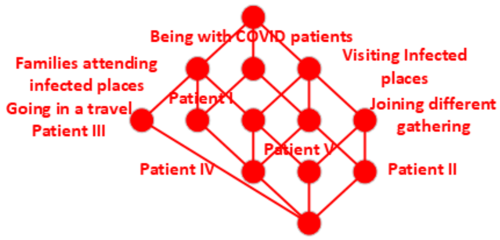
$$M_1 = \left\{\begin{matrix}\text{Going on a Travel, Being with COVID Patients, Joining Different Gathering,} \\ \text{Visit Infected Places, Families attending Infected Places}\end{matrix}\right\},$$

$$M_2 = \{\text{Heart Disease, Diabetes, Chronic Lung Disease, Asthma, Hyper Tension}\},$$

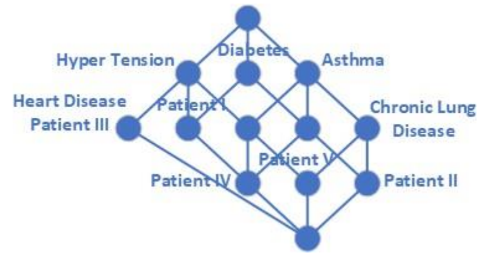
$$M_3 = \{\text{Fever, Dry Cough, Fatigue}\}, and$$

$$M_4 = \{\text{Breathing Problems, Sore Throat, Running Nose, Headache, Gastrointestinal}\}.$$

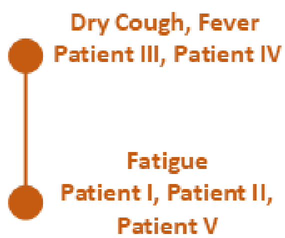
The concept lattices depicted in Fig. 3.1 provide a representation of the contexts $\mathcal{K}_i$, where $i = \{1, 2, 3, 4\}$, as presented in Tables 3.2, 3.3, 3.4, and 3.5.



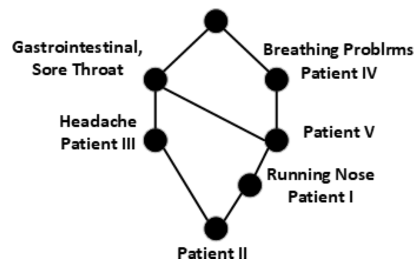
$\mathcal{B}(\mathcal{K}_1)$



$\mathcal{B}(\mathcal{K}_2)$



$\mathcal{B}(\mathcal{K}_3)$



$\mathcal{B}(\mathcal{K}_4)$

Fig. 3.1: The Concept Lattices of the Four Contexts $\mathcal{K}_i$, $i = \{1, 2, 3, 4\}$

Fig 3.2 and 3.3 represent the nested diagrams of the concept lattices $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2))$, and $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))$, respectively.

As we notice, the parallel lines are reduced, so we get a more straightforward diagram.

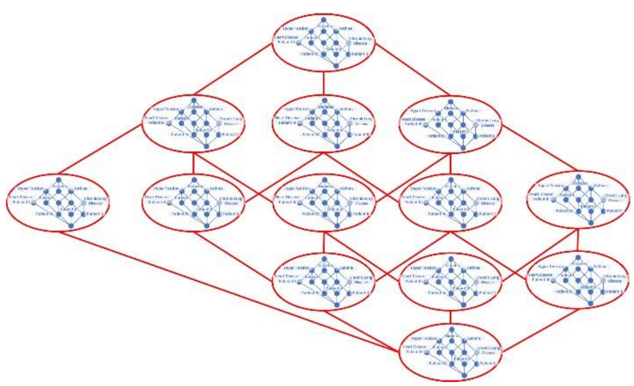


Fig. 3.2: The Bi-Concept Lattice $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2))$

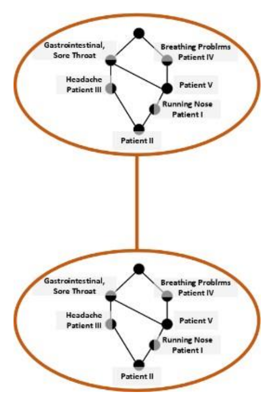


Fig. 3.3: The Bi-Concept Lattice $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))$

Following the Representation Theorem of Tri-Concept lattices (Theorem 3.2), we can construct the Tri-Concept lattice, corresponding to the data given in Tables 3.2, 3.3, 3.4, and 3.5. Also, form the Tri-Concept lattice by the bi-concept lattices $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2))$, and $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))$ as shown in Fig 3.2 and 3.3, respectively. The Tri-Concept lattice represented by $\mathfrak{Z}\left(\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2)), \mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))\right)$ is shown in Fig. 3.4.

Considering the concept; $\left(\begin{matrix}((\{\text{II}\}, \{BC, JD, VI\}), (\{\text{II}, \text{V}\}, \{CL, As\})), \\ ((\{\text{I}, \text{II}, \text{III}, \text{IV}, \text{V}\}, \{Fa, DC\}), (\{\text{II}\}, \{BP, ST, RN, He, Ga\}))\end{matrix}\right)$ in Fig. 3.4, we notice that patient II had a fever and dry cough. In a few days, some symptoms appear, like breathing problems, sore throat, running nose, headache, and gastrointestinal. He tested positive for COVID. The patient became infected by being with COVID patients, joining different gatherings, or visiting infected places. Doctors informed him that the disease might cause some complications in the future because he has chronic lung disease and asthma.

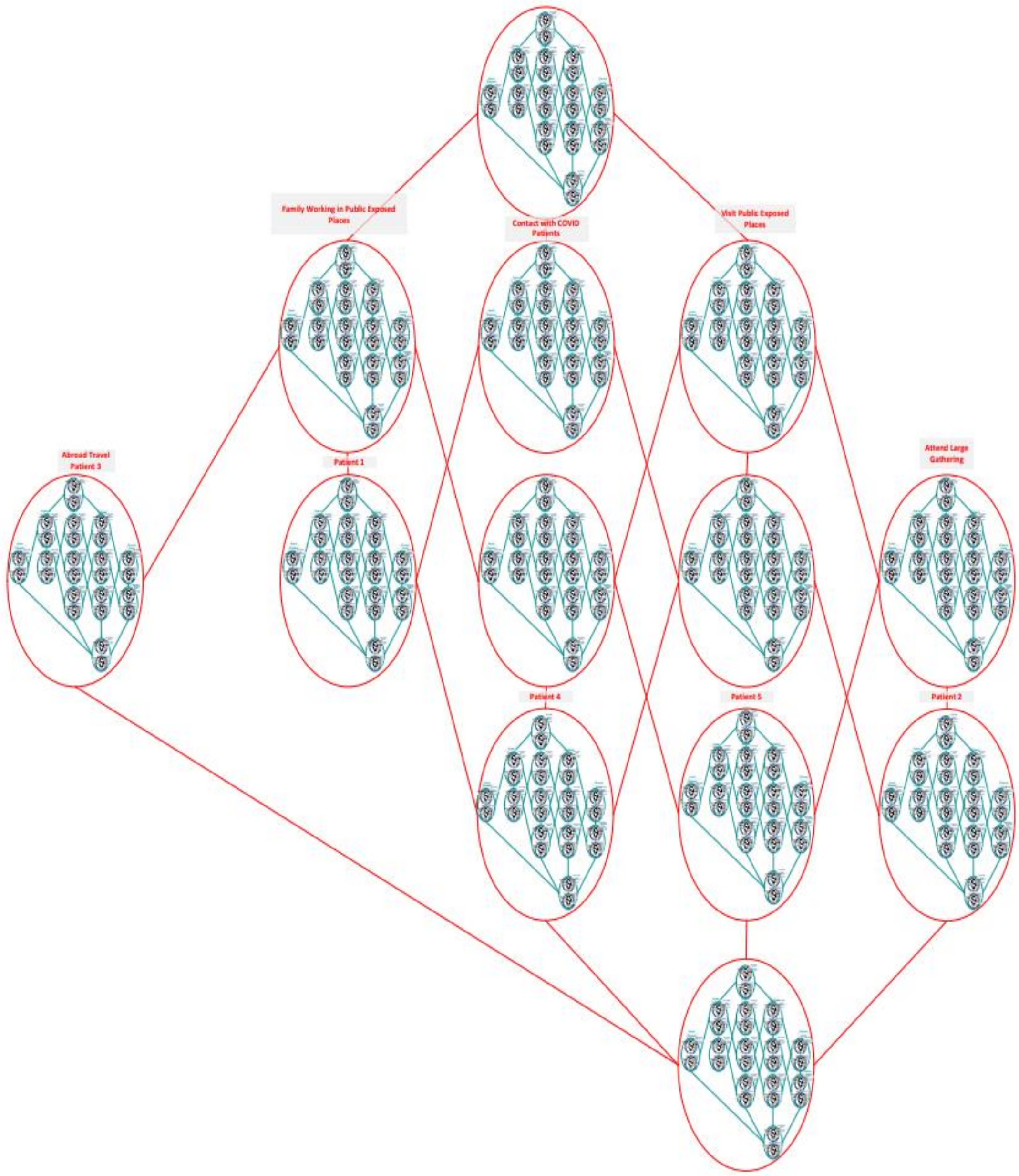


Fig. 3.4: The Tri-Concept Lattice $\mathfrak{Z}\left(\mathfrak{Y}\left(\mathcal{B}(\mathcal{K}_1),\mathcal{B}(\mathcal{K}_2)\right),\mathfrak{Y}\left(\mathcal{B}(\mathcal{K}_3),\mathcal{B}(\mathcal{K}_4)\right)\right)$

Construction using Iceberg Concept Lattice of Subcontexts

In the following, the iceberg concept lattice is created for each subcontext of the main context $\mathcal{K}=(G,M,I)$. Two bi-concept lattices, $\mathfrak{Y}\left(\mathcal{B}(\mathcal{K}_1),\mathcal{B}(\mathcal{K}_2)\right)$ and $\mathfrak{Y}\left(\mathcal{B}(\mathcal{K}_3),\mathcal{B}(\mathcal{K}_4)\right)$, can be constructed, corresponding to $\mathcal{K}_i's$ from the concept lattices $\mathcal{B}(\mathcal{K}_i)$, $i=\{1,2,3,4\}$, respectively. Utilizing the Representation Theorem of Tri-Concept lattices (Theorem 3.2), a more concise diagram is obtained by representing the Tri-Concept lattice as a nested diagram incorporating the two reduced bi-concept lattices.

Example 3.2 Using a Python Code to extract the concepts corresponding to the Tri-Concept lattices arose from the data given in Tables 3.2, 3.3, 3.4, and 3.5, we get 128 Tri-Concept s out of 2366 concepts after reduction.

Now, we construct the iceberg concept lattice for each concept lattice of the four contexts. Using a $\text{minsupp}=0.45$, as explained in Fig. 3.5, we get four iceberg concept lattices.

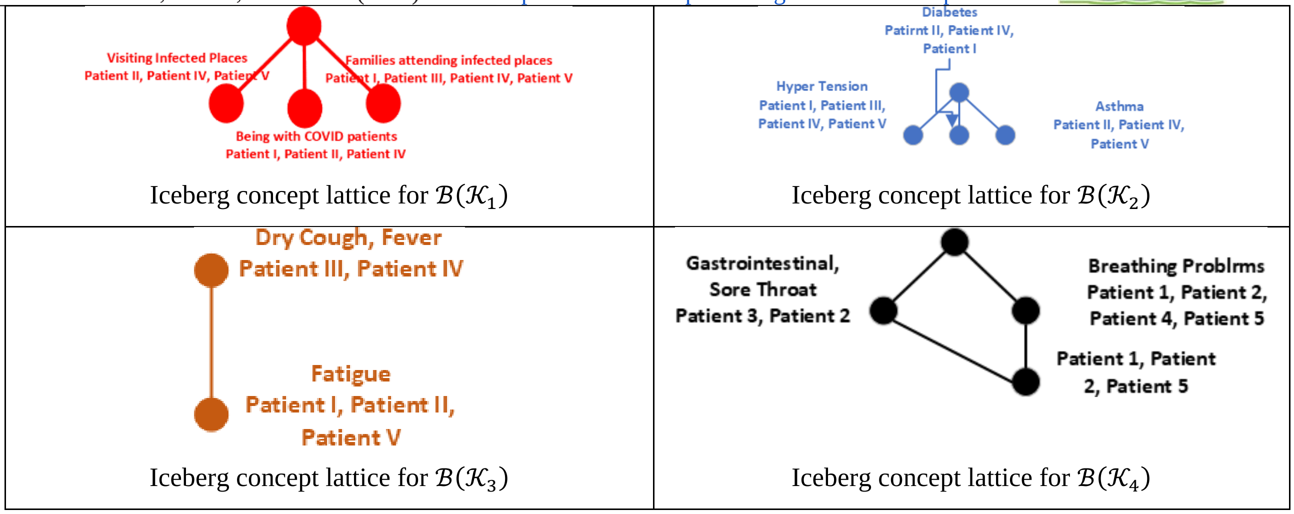


Fig. 3.5: Iceberg Concept Lattices of the Four Contexts $\mathcal{K}_i$, $i = \{1, 2, 3, 4\}$

Using the TRI-NEST algorithm and the Representation Theorem of Tri-Concept lattices (Theorem 3.2), we get the nested diagram of the Tri-Concept lattice as shown in Fig. 3.7. It consists of the bi-concept lattices $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2))$ and $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))$, which are displayed in Fig. 3.6.

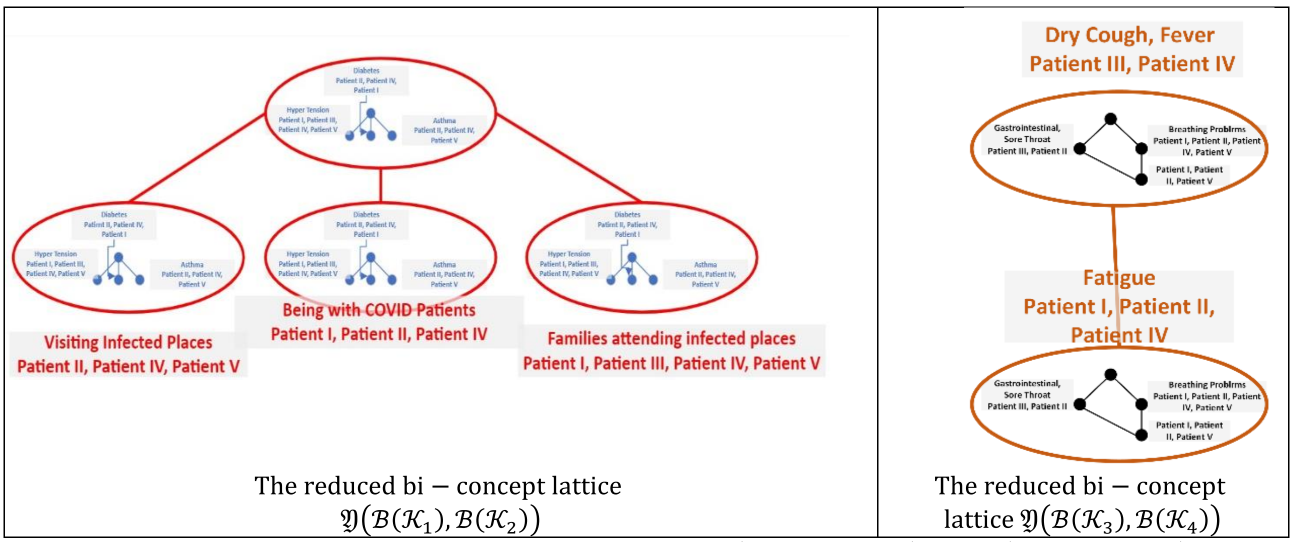


Fig. 3.6: Iceberg Diagrams of the Bi-Concept Lattices $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2))$ and $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))$

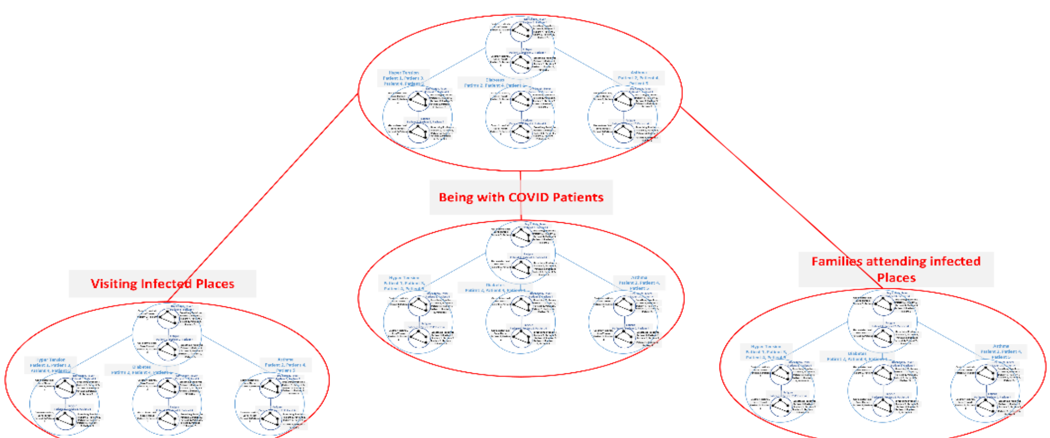


Fig. 3.7: The Reduced Tri-Concept Lattice $\mathfrak{Z}\left(\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2)), \mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))\right)$

The ICE-T Algorithm for Mining All Frequent Concepts

Introducing the ICE-T Algorithm enables us to compute all frequent concepts from any Tri-Concept lattice, which helps construct the iceberg concept lattice.

The tri-support, "tri-supp" of the concept $\left(\left((N,L),(N^*,L^*)\right),\left((N^{**},L^{**}),(N^{***},L^{***})\right)\right)$, is the average of the supports of its intents.

Definition 3.1 Given a context $\mathcal{K}=(G,M,I)$ and four subcontexts from $\mathcal{K}$, $\mathcal{K}_1,\mathcal{K}_2,\mathcal{K}_3$, and $\mathcal{K}_4$. Let $\mathfrak{Z}\left(\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1),\mathcal{B}(\mathcal{K}_2)),\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3),\mathcal{B}(\mathcal{K}_4))\right)$ be a Tri-Concept lattice corresponding to the context $\mathcal{K}$. Define the tri-support "tri_{supp}" of the concept $\left(\left((N,L),(N^*,L^*)\right),\left((N^{**},L^{**}),(N^{***},L^{***})\right)\right)$ as:

$$tri_{supp}\left(\left((N,L),(N^*,L^*)\right),\left((N^{**},L^{**}),(N^{***},L^{***})\right)\right)=\frac{\left[\frac{[supp(\mathrm{L})+supp(\mathrm{L}^*)]}{2}+\frac{[supp(L^{**})+supp(L^{***})]}{2}\right]}{2}$$

$$=\frac{[supp(\mathrm{L})+supp(\mathrm{L}^*)+supp(L^{**})+supp(L^{***})]}{4}$$

where $supp(L)=\frac{|L^{\#}|}{|G|}$, $supp(L^*)=\frac{|(L^*)^{\#}|}{|G|}$, $supp(L^{**})=\frac{|(L^{**})^{\#}|}{|G|}$ *and* $supp(L^{***})=\frac{|(L^{***})^{\#}|}{|G|}$ are the supports of $(N,L),(N^*,L^*),(N^{**},L^{**})$, and (N^{***},L^{***}) in the concept lattices $\mathcal{B}(\mathcal{K}_1)$, $\mathcal{B}(\mathcal{K}_2)$, $\mathcal{B}(\mathcal{K}_3)$, and $\mathcal{B}(\mathcal{K}_4)$, respectively.

Proposition 3.1 Let $\left(\left((N,L),(N^*,L^*)\right),\left((N^{**},L^{**}),(N^{***},L^{***})\right)\right)$ be a concept in the Tri-Concept lattice $\mathfrak{Z}\left(\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1),\mathcal{B}(\mathcal{K}_2)),\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3),\mathcal{B}(\mathcal{K}_4))\right)$. It is a frequent concept if and only if, for a fixed threshold minsupp, we get

$$tri_{supp}\left(\left((N,L),(N^*,L^*)\right),\left((N^{**},L^{**}),(N^{***},L^{***})\right)\right)\geq minsupp$$

Consequently, the set of all frequent concepts for the Tri-Concept lattice "tri_{ice}" can be explained as

$$tri_{ice}=\left\{\begin{matrix}\left(\left((N,L),(N^*,L^*)\right),\left((N^{**},L^{**}),(N^{***},L^{***})\right)\right)\in\mathfrak{Z}\left(\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1),\mathcal{B}(\mathcal{K}_2)),\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3),\mathcal{B}(\mathcal{K}_4))\right)\\ :tri_{supp}\left(\left((N,L),(N^*,L^*)\right),\left((N^{**},L^{**}),(N^{***},L^{***})\right)\right)\geq minsupp\end{matrix}\right\}$$

It forms a join semilattice of $\mathfrak{Z}\left(\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1),\mathcal{B}(\mathcal{K}_2)),\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3),\mathcal{B}(\mathcal{K}_4))\right)$ that we deal with, as a result, after using a relevant minsupp.

ICE-T Algorithm:

The First FreqCon and *Next* FreqCon algorithms, previously introduced in [9], have been extended to include the First FreqTri-Con and Next FreqTri-Con algorithms, optimized for utilizing the Tri-Concept lattices. Furthermore, the ICE-T algorithm has been introduced to identify all frequent concepts with support greater than or equal to a specific minimum threshold.

First FreqTri-Con Algorithm

Input:	$\mathcal{K}_1=(G,M_1,I_1),\mathcal{K}_2=(G,M_2,I_2),\mathcal{K}_3=(G,M_3,I_3)$ and $\mathcal{K}_4=(G,M_4,I_4)$ four contexts τ_{M_1} – minimal support of $\mathcal{K}_1$ & (N,L) is a formal concept of $\mathcal{K}_1$ τ_{M_2} – minimal support of $\mathcal{K}_2$ & (N^*,L^*) is a formal concept of $\mathcal{K}_2$ τ_{M_3} – minimal support of $\mathcal{K}_3$ & (N^{**},L^{**}) is a formal concept of $\mathcal{K}_3$ τ_{M_4} – minimal support of $\mathcal{K}_4$ & (N^{***},L^{***}) is a formal concept of $\mathcal{K}_4$
Output:	$\left(\left((N,L),(N^*,L^*)\right),\left((N^{**},L^{**}),(N^{***},L^{***})\right)\right)$
	$(N,L)=First\ FreqConc\left((G,M_1,I_1),\tau_{M_1}\right)$ $(N^*,L^*)=First\ FreqConc\left((G,M_2,I_2),\tau_{M_2}\right)$ $(N^{**},L^{**})=First\ FreqConc\left((G,M_3,I_3),\tau_{M_3}\right)$ $(N^{***},L^{***})=First\ FreqConc\left((G,M_4,I_4),\tau_{M_4}\right)$ Return $\left(\left((N,L),(N^*,L^*)\right),\left((N^{**},L^{**}),(N^{***},L^{***})\right)\right)$

Next FreqTri-Con Algorithm

Input:	$\mathcal{K}_1 = (G, M_1, I_1), \mathcal{K}_2 = (G, M_2, I_2), \mathcal{K}_3 = (G, M_3, I_3)$ and $\mathcal{K}_4 = (G, M_4, I_4)$ four contexts τ_{M_1} – minimal support of $\mathcal{K}_1$ & (N, L) is a formal concept of $\mathcal{K}_1$ τ_{M_2} – minimal support of $\mathcal{K}_2$ & (N^*, L^*) is a formal concept of $\mathcal{K}_2$ τ_{M_3} – minimal support of $\mathcal{K}_3$ & (N^{**}, L^{**}) is a formal concept of $\mathcal{K}_3$ τ_{M_4} – minimal support of $\mathcal{K}_4$ & (N^{***}, L^{***}) is a formal concept of $\mathcal{K}_4$
Output:	$\left(\left((N_i, L_i), (N_i^*, L_i^*)\right), \left((N_i^{**}, L_i^{**}), (N_i^{***}, L_i^{***})\right)\right)$
	$(N_i, L_i) = Next\ FreqConc\left((N, L), (G, M_1, I_1), \tau_{M_1}\right)$ $(N_i^*, L_i^*) = Next\ FreqConc\left((N^*, L^*), (G, M_2, I_2), \tau_{M_2}\right)$ $(N_i^{**}, L_i^{**}) = Next\ FreqConc\left((N^{**}, L^{**}), (G, M_3, I_3), \tau_{M_3}\right)$ $(N_i^{***}, L_i^{***}) = Next\ FreqConc\left((N^{***}, L^{***}), (G, M_4, I_4), \tau_{M_4}\right)$ Return $\left(\left((N_i, L_i), (N_i^*, L_i^*)\right), \left((N_i^{**}, L_i^{**}), (N_i^{***}, L_i^{***})\right)\right)$

ICE-T Algorithm:

Input:	$\mathcal{K}_1 = (G, M_1, I_1), \mathcal{K}_2 = (G, M_2, I_2), \mathcal{K}_3 = (G, M_3, I_3)$ and $\mathcal{K}_4 = (G, M_4, I_4)$ four contexts τ_M — minimal support threshold
Output:	$\mathfrak{Z} = \left\{\left(\left((N, L), (N^*, L^*)\right), \left((N^{**}, L^{**}), (N^{***}, L^{***})\right)\right)\right\}$

$M = M_1 \cup M_2 \cup M_3 \cup M_4$
$I_1 = I \cap (G \times M_1)$
$I_2 = I \cap (G \times M_2)$
$I_3 = I \cap (G \times M_3)$
$I_4 = I \cap (G \times M_4)$
$I \in (I_1 \times I_2) \times (I_3 \times I_4) = (G \times M_1) \times (G \times M_2) \times (G \times M_3) \times (G \times M_4)$
$= \left(\left((G \times M_1) \times (G \times M_2)\right) \times \left((G \times M_3) \times (G \times M_4)\right)\right)$
$\mathfrak{Z} = \emptyset$
repeat
$\left(\left((N, L), (N^*, L^*)\right), \left((N^{**}, L^{**}), (N^{***}, L^{***})\right)\right) :=$First FreqTri-Con
$\left(\left(\left((G, M_1, I_1), (G, M_2, I_2)\right), \left((G, M_3, I_3), (G, M_4, I_4)\right)\right), \tau_M\right)$
Repeat
$Tri - supp = \frac{1}{4}\left(\frac{|L^{\#}|}{|G|} + \frac{|(L^*)^{\#}|}{|G|} + \frac{|(L^{**})^{\#}|}{|G|} + \frac{|(L^{***})^{\#}|}{|G|}\right)$
    if $N = (L)^{\#},\ N^* = (L^*)^{\#}, N^{**} = (L^{**})^{\#}\ and\ N^{***} = (L^{***})^{\#}$ then
        if $Tri - supp = \tau_M$ then
            add $\left(\left((N, L), (N^*, L^*)\right), \left((N^{**}, L^{**}), (N^{***}, L^{***})\right)\right)$ to $\mathfrak{Z}$
        end if
        if $Tri - supp > \tau_M$ then
            add $\left(\left((N, L), (N^*, L^*)\right), \left((N^{**}, L^{**}), (N^{***}, L^{***})\right)\right)$ to $\mathfrak{Z}$
        end if
    end if
until not NextFreqTriCon
$\left(\left(\left((N, L), (N^*, L^*)\right), \left((N^{**}, L^{**}), (N^{***}, L^{***})\right)\right), \left(\left((G, M_1, I_1), (G, M_2, I_2)\right), \left((G, M_3, I_3), (G, M_4, I_4)\right)\right), \tau_M\right)$
until max(M)

Example 3.3 Applying the Representation Theorem of Tri-Concept lattices (Theorem 3.2) and the ICE-T Algorithm will help us obtaining f 140 reduced tri-concepts out of 2366 concepts as presented in Table 3.6. The results are obtained using Python. They are based on the data provided in Tables 3.2, 3.3, 3.4, and 3.5.

Table 3.6: A sample of concepts of the Tri-Concept lattice

Tri-concept	Supp (L)	Supp (L^{*})	Supp (L^{**})	Supp (L^{***})	Tri-Supp
$\left(\begin{array}{c}((\{\mathrm{I,II,III,IV,V}\},\{\,\}),(\{\mathrm{II,IV}\},\{Di,As\})),\\((\{\mathrm{I,II,III,IV,V}\},\{Fe,DC\}),(\{\mathrm{I,II,III,IV,V}\},\{\,\}))\end{array}\right)$	1	0.4	1	1	0.85
$\left(\begin{array}{c}((\{\mathrm{I,II,III,IV,V}\},\{\,\}),(\{\mathrm{II,IV}\},\{Di,As\})),\\((\{\mathrm{I,II,IV}\},\{Fe,DC,Fa\}),(\{\mathrm{I,II,III,IV,V}\},\{\,\}))\end{array}\right)$	1	0.4	0.6	1	0.75
$\left(\begin{array}{c}((\{\mathrm{I,II,III,IV,V}\},\{\,\}),(\{\mathrm{II,IV}\},\{Di,As\})),\\((\{\mathrm{I,II,III,IV,V}\},\{Fe,DC\}),(\{\mathrm{I,II,III,IV,V}\},\{\,\}))\end{array}\right)$	1	0.4	1	1	0.85
$\left(\begin{array}{c}((\{\mathrm{I,II,III,IV,V}\},\{\,\}),(\{\mathrm{II,V}\},\{CL,As\})),\\((\{\mathrm{I,II,IV}\},\{Fe,DC,Fa\}),(\{\mathrm{I,II,III,IV,V}\},\{\,\}))\end{array}\right)$	1	0.4	0.6	1	0.75
$\left(\begin{array}{c}((\{\mathrm{I,II,III,IV,V}\},\{\,\}),(\{\mathrm{II,V}\},\{CL,As\})),\\((\{\mathrm{I,II,III,IV,V}\},\{Fe,DC\}),(\{\mathrm{I,II,III,IV,V}\},\{\,\}))\end{array}\right)$	1	0.4	1	1	0.85
$\left(\begin{array}{c}((\{\mathrm{I,II,III,IV,V}\},\{\,\}),(\{\mathrm{IV,V}\},\{As,HT\})),\\((\{\mathrm{I,II,III,IV,V}\},\{Fe,DC\}),(\{\mathrm{I,II,III,IV,V}\},\{\,\}))\end{array}\right)$	1	0.4	1	1	0.85

If the minsupp is 0.70, the reduced Tri-Concept lattice can be represented, as shown in Fig. 3.8.

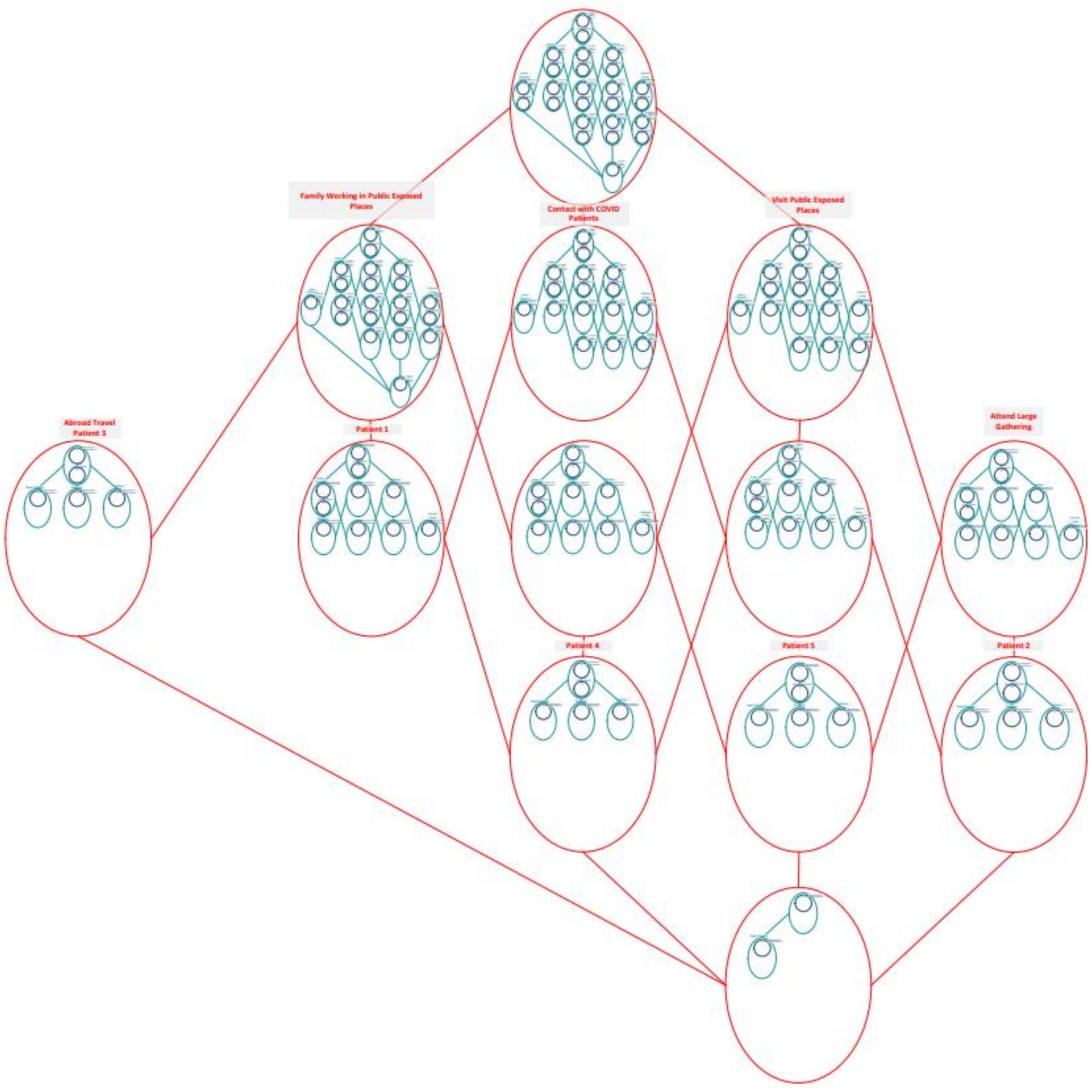


Fig. 3.8: Iceberg concept lattice of Tri-Concept lattice $\mathfrak{Z}\left(\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1),\mathcal{B}(\mathcal{K}_2)),\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3),\mathcal{B}(\mathcal{K}_4))\right)$

Construction of Tri-Concept Lattices using the Iceberg Concept lattices corresponding to Frequent Generators

In this part, Tri-Concept lattices are constructed using Iceberg concept lattices obtained from frequent closures and the associated generators. The Snow Algorithm is utilized to process each subcontext, following the steps outlined below:

1) Form the concept lattice of each subcontext.

2) Generate the set of all closed itemsets and their generator for each concept lattice.

3) Apply the Snow Algorithm for each set of closed itemsets (CI) and their associated generators to get all frequent itemsets and frequent generators.

4) Build the iceberg concept lattice for each dyadic context.

5) Build two bi-concept lattices; one of them corresponds to the concept lattices $\mathcal{B}(\mathcal{K}_1)$ and $\mathcal{B}(\mathcal{K}_2)$, while the other corresponds to $\mathcal{B}(\mathcal{K}_3)$ and $\mathcal{B}(\mathcal{K}_4)$. Then form their nested diagram that represents the Tri-Concept lattice in a more reduced form using **Error! Reference source not found.** and the TRI-NEST Algorithm.

By setting a minsupp to 45%, the concept lattices are described in Fig 3.9 and 3.10. The diagram shown in Fig. 3.11 is obtained by constructing the nested diagram representing the Tri-Concept lattice.

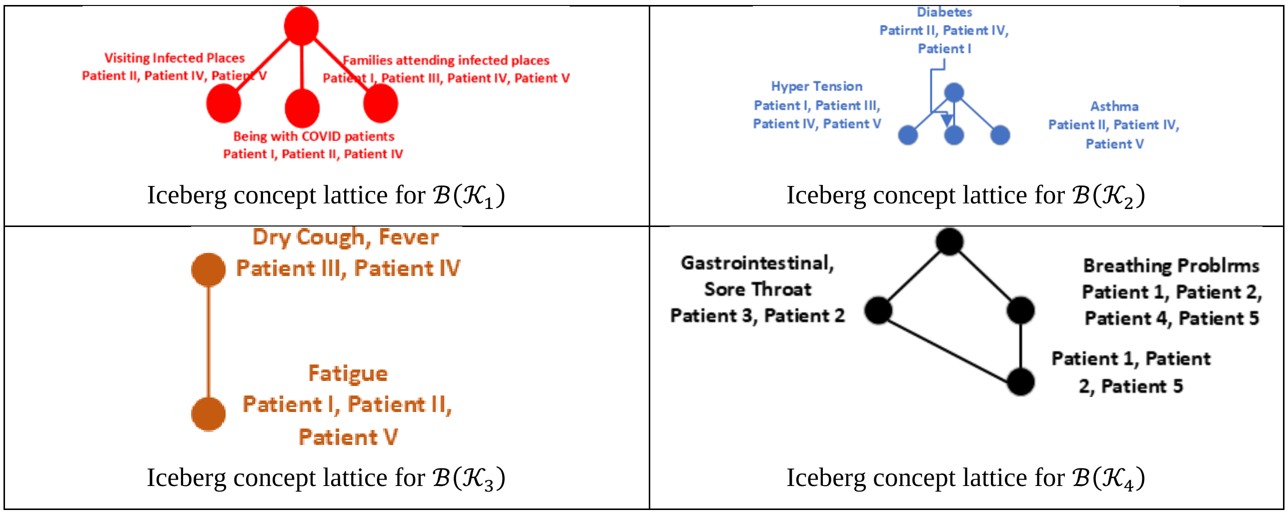


Fig. 3.9: Iceberg Concept Lattices of the Four Contexts $\mathcal{K}_i$, $i = \{1,2,3,4\}$

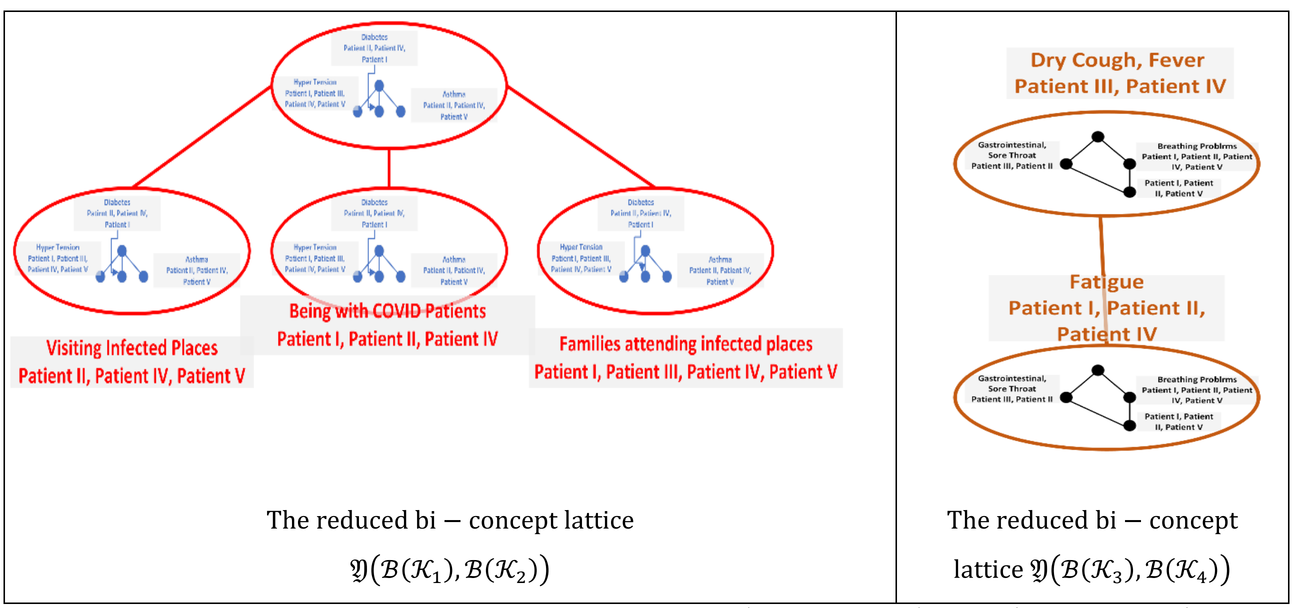


Fig. 3.10: Iceberg Diagrams of the Bi-Concept Lattices $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1), \mathcal{B}(\mathcal{K}_2))$ and $\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3), \mathcal{B}(\mathcal{K}_4))$

Forming the reduced form of the Tri-Concept lattice shows that iceberg concept lattices effectively reduce the noise in concept lattices by reducing the nodes in the diagram, preserving all data without loss.

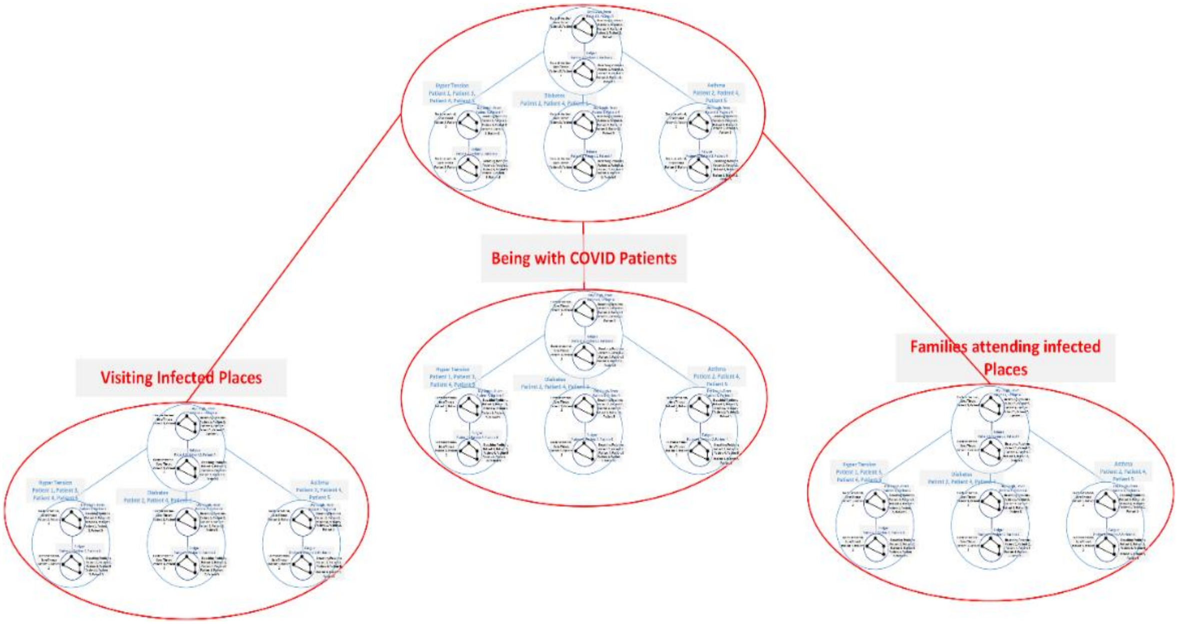


Fig. 3.11: The Reduced Tri-Concept Lattice $\mathfrak{Z}\left(\mathfrak{Y}(\mathcal{B}(\mathcal{K}_1),\mathcal{B}(\mathcal{K}_2)),\mathfrak{Y}(\mathcal{B}(\mathcal{K}_3),\mathcal{B}(\mathcal{K}_4))\right)$

4 Conclusion and Discussion

Introducing the notion of Tri-Concept lattices enables us to deal with data arising from different information sources. The representation by the product of bi-concept lattices leads to the generalization of nested diagrams using the suggested algorithm TRI-NEST.

Nested Diagrams, a well-established and widely recognized tool in Formal Concept Analysis (FCA), that makes it possible to distribute representation details across several levels. Adding the ICE-T algorithm and applying the snow algorithm facilitate the computation of all frequent concepts of Tri-Concept lattices. This computation is needed to construct the iceberg concept lattices, a perfect tool for analyzing large databases. It represents the most essential part of the Tri-Concept lattice without compromising any vital information.

Applying an example of real-world data provides us with a valuable opportunity to understand the role played by the added structures in data analysis. This application involves using different methods to reduce the complexity of extracting information from the concept lattices.

5 Future Work

Our main interest is to continue analyzing big data using the structure of the bi-concept lattice and its representations, such as association rule mining and concept stability.

6 Conflict of interest

The authors declare that there is no conflict regarding the publication of this paper.

7 References

[1] R. Wille. *Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts*, in Formal Concept Analysis, S. Ferré and S. Rudolph, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 314–339 (2009).

[2] B. Ganter and R. Wille. *Formal Concept Analysis,* Berlin, Heidelberg: Springer (1999).

[3] R. Wille. *Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies*, in Formal Concept Analysis: Foundations and Applications, B. Ganter, G. Stumme, and R. Wille, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 1–33 (2005).

[4] S. Kuznetsov and S. Obiedkov. *Algorithms for the Construction of Concept Lattices and Their Diagram Graphs*, presented at the Principles of Data Mining and Knowledge Discovery - Lecture Notes in Computer Science, pp. 289–300, (2001).

[5] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal. Computing Iceberg Concept Lattices with Titanic, *Data & Knowledge Engineering*, vol. 42, no. 2, Art. no. 2, (2002).

[6] L. Szathmary, P. Valtchev, A. Napoli, and R. Godin. *Constructing Iceberg Lattices from Frequent Closures Using Generators*, in Discovery Science, J. François, M. R. Berthold, and T. Horváth, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 136–147 (2008).

[7] R. Wille. Line Diagrams of Hierarchical Concept Systems, Ko *Knowledge Organization*, vol. 11, no. 2, pp. 77–86 (1984).

[8] F. Lehmann and R. Wille. *A Triadic Approach to Formal Concept Analysis*, in Conceptual Structures: Applications, Implementation and Theory, G. Ellis, R. Levinson, W. Rich, and J. F. Sowa, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 32–43 (1995).

[9] R. Jäschke, A. Hotho, C. Schmitz, B. Ganter, and G. Stumme. Trias--an Algorithm for Mining Iceberg Tri-Lattices, *Sixth International Conference on Data Mining (ICDM'06)*, (2006).

[10] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, (2002).

[11] M. Fitting. *Bilattices in Logic Programming*, presented at the Proceedings of The International Symposium on Multiple-Valued Logic, pp. 238–246 (1990).

[12] M. Ginsberg. multi-valued Logics, presented at *the AAAI Conference on Artificial Intelligence*, (1986).

[13] M. Ginsberg. multi-valued Logics: A Uniform Approach to Reasoning in Artificial Intelligence, *Computational Intelligence*, vol. 4, no. 3, pp. 265–316 (1988).

[14] U. Rivieccio. *An Algebraic Study of Bilattice-Based Logics*. arXiv, (2010).

[15] A.Steen and C. Benzmüller. Sweet Sixteen: Automation Via Embedding into Classical Higher-Order Logic, in *Logic and Logical Philosophy* (2016).

[16] M. Fitting. *Notes on Bilattice*. Http:// melvinfitting.org/forclasses /phil76500spring2018/LectureNotes/BilatticesNotes /BilatticesPhiLog.pdf (2018).

[17] F. Bou, R. Jansana, and U. Rivieccio. Varieties of Interlaced Bilattices, *Algebra Univers.*, vol. 66, no. 1, pp. 115 (2011).

[18] Y. Shramko, J. Dunn, and T. Takenaka. The Trilattice of Constructive Truth Values, *Journal of Language and Computation*, vol. 11, pp. 761–788 (2001).

[19] U. Rivieccio. Representation of Interlaced Trilattices, *Journal of Applied Logic*, vol. 11, no. 2, pp. 174–189 (2013).

[20] Y. Shramko and H. Wansing. Some Useful Sixteen-Valued Logics: How a Computer Network Should Think, *J Philos Logic*, vol. 34, no. 2, pp. 121–153 (2005).

[21] T. M. Ferguson. Rivals to Belnap–Dunn Logic on Interlaced Trilattices, *Stud Logica*, vol. 105, no. 6, pp. 1123–1148 (2017).

[22] K. Nehmé, P. Valtchev, M. H. Rouane, and R. Godin. *On Computing the Minimal Generator Family for Concept Lattices and Icebergs*, in Formal Concept Analysis, B. Ganter, Ed., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 192–207 (2005).

[23] S. El-Assar, M. Atallah, and E. Ghareeb. Bi-Concepts and Mining Association Rules, *Information-An International Interdisciplinary Journal*, vol. 18, pp. 431 (2015).