Reliability Engineering and System Safety 202 (2020) 107002

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Introduction to formal concept analysis and its applications in reliability engineering

Claudio M. Rocco[a,*], Elvis Hernandez-Perdomo[b], Johnathan Mun[c]

[a] Universidad Central de Venezuela, Venezuela
[b] International Institute of Professional Education and Research IIPER, United States
[c] Naval Postgraduate School, United States

ARTICLE INFO

Keywords:
Defense strategies
Failure analysis
Formal concept analysis
Knowledge space theory
Partial order
Reliability engineering

ABSTRACT

Formal Analysis of Concepts (FCA) is a method of data analysis that helps to study the relationship between a set of objects and a set of attributes (the formal context). FCA not only allows detecting data groups (concepts) and their graphical visualization, but also extracting rules that could reveal the underlying structure of the analyzed context. The main idea of this paper is to present the fundamentals of FCA and how it can be used in reliability engineering problems. To this aim, examples in reliability engineering, from both the literature and authors' experience, have been selected for analysis. Comments on the new insights provided by FCA are also highlighted. Finally, the results from the examples selected show that other reliability areas could benefit from using an FCA-based approach.

1. Introduction

The formal analysis of concepts (FCA) is a method of data analysis that allows studying the relationship between a set of objects and a particular set of attributes, which can reveal their structure [1]. FCA is a branch of lattice theory that was proposed by Rudolf Wille in 1981 [2]. FCA starts with the relation between a set of objects and a set of attributes, represented by a cross table. As a result of processing the input data, FCA produces: 1) groups that represent "natural" concepts in terms of the attributes of the data; and 2) a collection of implications that describes a specific dependency that exists in the data.

FCA has been used in different areas like software mining [4], web mining [5], web-documents browsing [6–7], text mining and linguistics [8], medicine [5,9–10], biology [11], chemistry [12–15], ontology engineering [16–18], psychological data [19], functional magnetic resonance imaging (fMRI) scans [20], sentiments analysis [21,22], decision-making [23], complex systems [24–25], e-learning [26], information sciences [27], criminal trajectories [28], anomaly detection [29], terrorist threat [30], breast cancer [31], image processing [32], preference analysis [33], material research [34], vessel accidents [35], among other applications (See [36–37] for a complete list of references).

The main idea of this paper is to show the fundamentals of FCA and how it can also be used in several reliability engineering problems. As it will be presented, FCA allows the visualization, navigation, information retrieval and interpretation of concepts and rules, in a natural way, with no statistical assumption on data.

To motivate the discussion, consider two problems faced in the reliability engineering context (these examples will be discussed in detail in the manuscript). The first is related to the analysis of the cut-sets in a system composed of two-state components, a well-known approach to analyze how the system could fail. Given an input table that describes the relationship between the status of the components (operating or failed) and the state of the system (failed), FCA allows directly answering questions that a reader could pose such as: What components belong to a specific state or a specific cut-set? What components define a minimum cut-set? What is the relationship among cut-sets? What components simultaneously belong to several cut-sets?

The second example is related to the definition of a set of protection strategies that could be used on a set of malevolent attacking scenarios on a system. In this case, starting again with an input table that describes which protection strategy could be used to defend the system against a specific attack, FCA allows answering, for example: What strategy could protect several attacks? What attacks do not have any defined defense strategy? What attack has a single protection strategy? What protection strategies safeguard the same set of attacks? What protection strategies, employed for a specific attack, could also be used for other attacks?

After processing the input data, FCA produces two results to manage such questions:

* Corresponding author.
E-mail address: croccoucv@gmail.com (C.M. Rocco).

https://doi.org/10.1016/j.ress.2020.107002
Received 29 January 2019; Received in revised form 9 March 2020; Accepted 30 April 2020
Available online 21 May 2020

1) The first is a lattice in which the data are grouped into a collection of "formal concepts", that is, groups representing "natural" concepts in terms of the attributes of the data. Such groups are further divided using other attributes, and finally "arranged hierarchically by a subconcept-superconcept relation" [3] (e.g., "cut-sets in a network"). A detailed visualization of the relationship is also possible.
2) The second output of FCA is a collection of implications that describes a dependency that exists in the data, such as, "protection strategies A or B, could be used in the case of an attack S1".

As previously mentioned, FCA is an approach well-known in the literature. As a reviewer mentioned, there is enough literature explaining FCA, and this paper does not present new insights concerning the "mathematics" of FCA. However, the set of selected examples, both found in the literature or based on the authors' experience, will represent, in our opinion, the beginning of applications of FCA in the reliability area. These examples range from simple problems to more complicated situations and show the main features of the FCA-based approach.

So, for the FCA practitioner, the presented cases represent new examples that enrich the FCA world. At the same time, for the reliability practitioner's view, FCA could be the start of another effective technique to enhance a better understanding of different problems. For these reasons, the paper is organized as follows: Section 2 presents a summary of the most important concepts of FCA, with comments regarding the advantages and disadvantages. Section 3 is devoted to developing several applications and examples. Finally, Section 4 concludes with the most important considerations of this novel data analysis technique.

2. Introduction to formal concept analysis

FCA is a data analysis technique, introduced by Wille [2] that studies the hierarchical structures induced by a binary relation between a pair of sets, namely subsets of objects and attributes. The main and original idea is to mimic the conceptual thinking of the human being. In this section, we describe some of its main characteristics (for further details revise [38–39]).

As is customary in the FCA literature, the example to be developed will be directly related to reliability engineering. For that purpose, the selected "toy example" to illustrate the main concepts of FCA is related to various modeling and analysis techniques used for assessing the reliability of communication networks, an important milestone presented by Ahmad et al. [58].

2.1. Formal context, formal concepts and lattice concepts

FCA requires as input a "formal context" that is a data matrix where rows correspond to objects and columns to attributes. The entries of the matrix describe the relationship between objects and attributes. In general, such relationship is represented by one (1) or zero (0) to reflect the presence or the absence of the attribute for that object. In many cases, it is normal to replace 1s by "X" and leave a blank space for 0s. It is important to remark that such replacement has the advantage that the elements of the data matrix cannot be confused with numbers. Indeed, an "X" has the meaning of "an object has the attribute" or "the attribute is realized by object…".

Mathematically, a formal context (G, M, I) consists of a set G of objects, a set M of attributes and a binary relation I ⊂ G x M. "g I m" means: "object g has attribute m". Table 1 shows the formal context table selected to illustrate FCA: "a generic overview of the major reliability modeling and analysis techniques in the domain of communication networks" [58].

Rows in Table 1 represent the modeling techniques, while columns are the attributes selected by Ahmad et al. [58] for characterizing each technique. The modeling techniques or the set of objects is $G=$

Table 1
Formal context for modeling techniques for assessing reliability [58].

Technique	Success domain (SD)	Failure domain (FD)	Top-down approach (TDA)	Identification and prevention of failure (IPF)	Large and complex systems (LCS)	Combinatorial problems (CP)	Non-combinatorial problems (NCP)
Reliability Block Diagram (RBD)	X		X	X	X	X	
Fault Tree (FT)		X	X	X	X	X	
Markov Chain (MCh)	X	X	X	X		X	X
Bayesian Networks (BN)	X	X	X	X		X	X

{Reliability Block Diagrams (RBDs), Fault Tree (FT), Markov Chain (MCh), Bayesian Networks (BN)}. The set of attributes is M = {Success domain (SD), Failure domain (FD), Top-down approach (TDA), Identification and prevention of failure (IPF), Large and complex systems (LCS), Combinatorial problems (CP), Non-combinatorial problems (NCP)}.

The seven attributes evaluate several aspects related to the advantages and limitations of the modeling techniques. For example, some techniques, such as RDB, MCh and BN could be "utilized to evaluate overall system reliability" (attribute "success domain") while FT "could mainly be used to model the failure relations among" components in a system (attribute "failure domain"). However, MCh and BN are "unable to handle large and complex systems due to its state-based nature" (attribute "Large and complex systems"). But not all systems can be modeled with combinations of series or parallel branches (RDB) or logical gates (FT) (attribute "Non-combinatorial problems"). (For example, "RDB I SD" means RDB is utilized "to evaluate overall system reliability").

Note that, in terms of graph theory, a formal context is a bigraph, consisting of a set of object vertices G, a set of attribute vertices M, and a set of edges set $I \subseteq G \times M$ which specifies a binary relation between these two sets [41] (Each edge is adjacent to one object and one attribute vertex). This means that results from FCA could be also applied in analyzing networks. In this case, the formal context is the incidence matrix of the graph [42].

In general, it is possible to define, in every formal context, a pair of derivation operators or concept-forming operators [43]. The Type I Derivation Operators $\uparrow: 2^G \rightarrow 2^M$ defines for every set $A \subset G$ (i.e., a selection of objects) based on the question: Which attributes from M are common to all these objects? That is $A' := A\uparrow := \{m \in M \mid g\ I\ m \text{ for all } g \in A\}$. In other words, A↑ is just the set of all attributes shared by all objects from A. For example, A = {RDB, MCh}, $A \subset G$ and A' = {SD, CP} is an example of the Type I operator as well as B = {FT,BN} and B' = {FD,TDA}.

The Type II derivation operator $\downarrow: 2^M \rightarrow 2^G$ defines for every set of attributes $B \subset M$ based on the question: Which objects have all the attributes from B? That is $B' := B\downarrow := \{g \in G \mid g\ I\ m \text{ for all } m \in B\}$. In other words, B↓ is the set of all objects sharing all attributes from B. For example, consider B = {CP,NCP}, $B \subset M$ and B' = {MCh,BN}, is an example of the Type II operator.

It is possible to show some facts for the derivation operators [38-39]. Let (G, M, I) be a formal context. Let A, A_1, $A_2 \subset G$ sets of objects and B, B_1, $B_2 \subset G$ sets of attributes. Then:

1) $A_1 \subset A_2 \Rightarrow A'_2 \subset A'_1$

1′) $B_1 \subset B_2 \Rightarrow B'_2 \subset B'_1$

2) $A \subset A''$

2′) $B \subset B''$

3) $A' = A'''$

3′) $B' = B'''$

4) $A \subset B' \Leftrightarrow B \subset A' \Leftrightarrow A \times B \subset I$

Note that a formal context expresses a relation between objects and attributes. However, as indicated in [44], the goal is "to characterize a subset of objects uniquely by a subset of properties and a subset of properties uniquely characterized by a subset of objects".

Let (G, M, I) be a formal context, where $A \subset G$ and $B \subset M$. (A, B) is a formal concept of (G, M, I), iff $A' = B$ and $B' = A$. In other words: (A,B) is a formal concept if and only if A has just objects sharing all attributes from B and B has just attributes shared by all objects from A. The set A is called the extent, and the set B is called the intent of the formal concept (A, B).

The term "concept" is a central, technical term and has its origin in philosophy. On the other side, "formal" means that a mathematical description is used. Note that formal concepts correspond to maximal bicliques of a bigraph [41].

For small formal contexts, it is easy to find formal concepts. Indeed, a formal concept (A, B) corresponds to a filled rectangular sub-table with a row set A and column set B. For example, for the formal context shown in Table 1, ({MCh, BN}, {SD, FD, TDA, IPF, CP, NCP}) is a formal concept. Note that the description of a formal concept allows a "symmetric view on objects and attributes" [44] because each one of the two parts determines the other. In reliability terms, it provides a "redundant" description.

At this point, it is possible to define a relation between formal concepts. Indeed, formal concepts are naturally ordered using the relation subconcept-superconcept, which is based on inclusion relation on objects and attributes [38–39].

Formally, the subconcept-superconcept relation is defined as follows. Let (A_1,B_1) and (A_2,B_2) be formal concepts of (G,M,I). (A_1,B_1) is a sub-concept of (A_2,B_2) (denoted as $(A_1, B_1) \leq (A_2, B_2)$) iff $\Leftrightarrow A_1 \subset A_2$ (iff $\Leftrightarrow B_2 \subset B_1$). In this case (A_2, B_2) is a super-concept of (A_1, B_1). In practical terms, $(A_1, B_1) \leq (A_2, B_2)$ means that the set A_1 is more specific than the set A_2 in (A_2,B_2), because B_1 requires more "has the attribute" relations than B_2. Note that $\leq$ is interpreted as the sub-concept-superconcept ordering. The notion of ordering allows defining two additional elements for every two formal concepts (A_1,B_1) and (A_2,B_2) of a given formal context: their greatest common subconcept or infimum and the least common superconcept or supremum:

Infimun (or meet): $(A_1,B_1) \wedge (A_2,B_2) = (A_1 \cap A_2, (B_1 \cup B_2)'')$ and Supremum (or join): $(A_1,B_1) \bigvee (A_2,B_2) = (A_1 \cup A_2, (B_1 \cap B_2)'')$

The set of all formal concepts of (G, M, I) is called the concept lattice (Galois lattice [38–40]) of the formal context (G, M, I) and is denoted by B (G,M,I).

An important theorem [39] states that the concept lattice of a formal context is a partially ordered set (*poset*), defined as: Let P be a set and $\leq$ is a binary relation on P. A partially ordered set is a pair (P, $\leq$), iff

$x \leq x$ (reflexive)

$x \leq y$ and $x \neq y \Rightarrow \neg\, y \leq x$ (antisymmetric)

$x \leq y$ and $y \leq z \Rightarrow x \leq z$ (reflexive)

for all x, y, z ∈ P.

That means that the concept lattice can be graphically represented by means of a special Hasse diagram (in the formalism of FCA it is called line diagram). Indeed, a Hasse diagram is a representation method of ordered sets that shows a hierarchy in the diagram [44]. Fig. 1 shows the graphical representation for the formal context in Table 1. The lattice is generated using the ConExp tool [45] (Freely available at URL: http://conexp.sourceforge.net/).

In this single-source, single-sink, labeled, directed acyclic graph (DAG), every node (or vertex) represents one formal concept while edges express the ordering of concepts. Two concepts are comparable if there is a directed path between their corresponding vertices in the DAG; otherwise, they are incomparable. As mentioned in [44], the filled upper semicircle indicates that there is at least one attribute attached to the corresponding concept, whereas the filled lower semicircle indicates the set object attached to the corresponding concept. In this figure, the so-called reduced labeling is used. That means that attributes are written only at the first node (concept) they appear in. This convention is also applied for objects [46] Note that the reduced labeling does not lead to a loss of information [1].

In Fig. 1, we notice the presence of two main elements: 1) A top concept (supremum of the lattice) whose extent includes all the objects and whose intent includes all attributes predicable of all objects, that is, the more general concept; 2) a bottom concept (infimum of the lattice) whose intent includes all attributes and whose extent include all objects

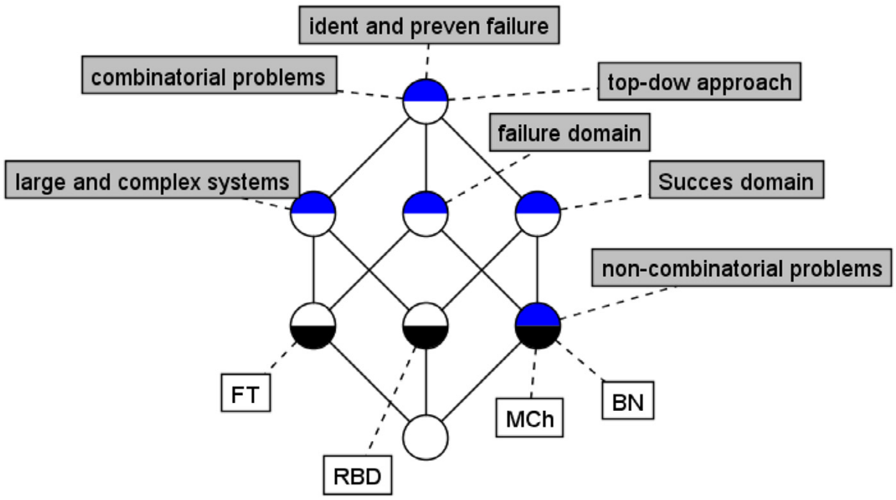

Fig. 1. Concept lattice representation of the formal context in Table 1.

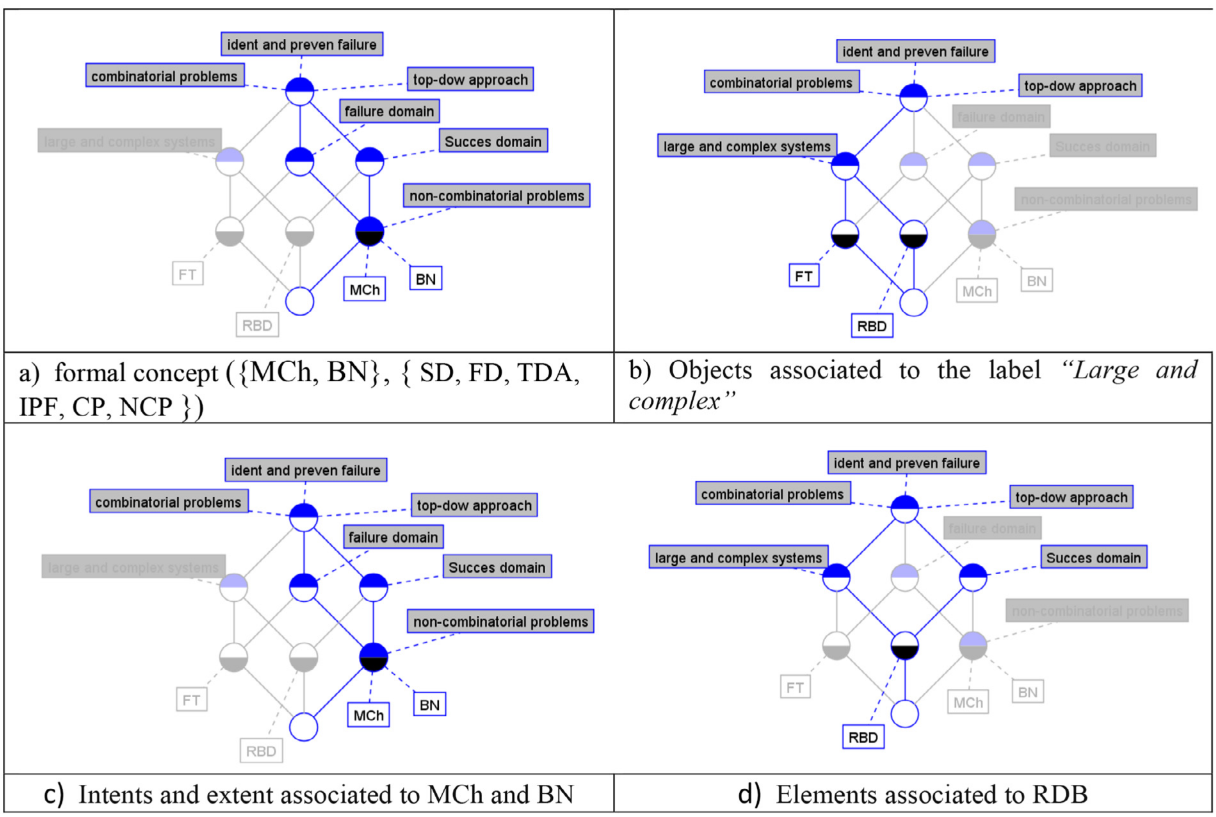

Fig. 2. Extents and intents of a selected vertex based on Table 1. a) formal concept ({MCh, BN}, {SD, FD, TDA, IPF, CP, NCP}). b) Objects associated to the label *"Large and complex"*. c) Intents and extent associated to MCh and BN. d) Elements associated to RDB.

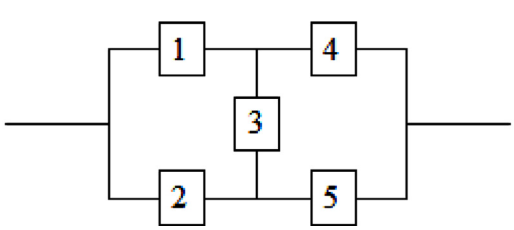

Fig. 3. A five-component network.

that share all attributes, i.e., the least general concept. In our example, the upper node reflects that all the techniques considered simultaneously have three out of seven attributes defined. Markov Chain and Bayesian Networks are the only techniques that can represent both failure and success domains, while only RDB and FT could be used for large and complex systems. Finally, the lower node represents (Ø, G), where Ø is the null set: no technique simultaneously shares all the attributes. Moving from top to bottom means considering more specific concepts, while moving from bottom to top means considering more general concepts.

Note the "navigation" in the lattice allows extracting the full information with respect to the extent and the intent of every concept. Indeed, starting at a vertex and moving upward, we find a set of attributes of the covering concept. If we move downward, we find the

Table 2
Failed states associated with Fig. 3.

Component	State e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}	e_{12}	e_{13}	e_{14}	e_{15}	e_{16}
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0
3	1	1	1	1	0	0	0	0	1	1	0	1	1	0	1	0
4	0	1	0	1	0	1	0	1	1	0	1	1	1	1	1	1
5	0	1	1	0	0	1	1	0	1	1	1	1	0	1	1	1

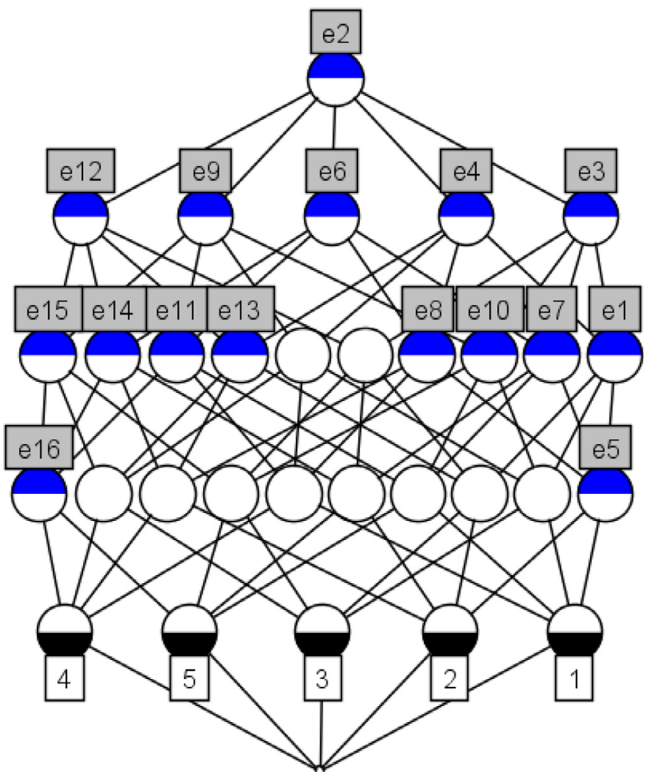

Fig. 4. Concept lattice diagram corresponding to the formal context of Table 2.

objects of the covered vertex.

For example, Fig. 2a highlights the elements that define one of the formal concepts previously detected ({MCh, BN}, {SD, FD, TDA, IPF, CP, NCP})

Fig. 2b shows the elements associated with the label "Large and complex". It means that "Large and Complex" is a property of the extents FT, RDB. The concept lattice also allows detecting all the intents of a selected object. For example, Fig. 2c shows that MCh or BN has six intents. Fig. 2d allows analyzing the lattice when RDB is selected.

2.2. Obtaining association rules [47]

From the previous section, it is clear that FCA allows two basic types of knowledge: a cross table or the formal context (i.e., the relation between objects and attributes), and the concept lattice (i.e., the set of all concepts of the context (G, M, I), ordered by the subconcept-superconcept relation $\leq$). However, in many cases, a third basic of knowledge can be also derived: implications and partial implications sets between the set of attributes. Dependencies or implications have the "if-then" structure, that is, A → C, where $A \subseteq M$, $C \subseteq M$ and $A \cap C = \emptyset$. A is the premise or antecedent and B is the consequent. The set of implications between attributes can be directly derived from the concept lattice. In fact, given $A \to C$, if an object has the attributes of the antecedent then it also contains the attributes of the consequent [1].

On the other side, partial implications [48] (also known as "association rules" [49]) hold almost for all objects. In other words, association rules do not apply to all their involved objects, and the probability of relationship of one set of attributes with the other set of attributes is less than 100%. [47].

Two indexes are used to characterize an association rule: *support* and *confidence*. The support of A → C is defined as support (A → C) = $|(A \cup C)|$ and measures "the number of objects that contain both the attributes of premise and conclusion of the association rule" [1]. The confidence of the association rule A → C is defined as confidence (A → C) $= |(A \cup C)| / |A|$ and represents "which percentage of objects that contain the premise of the association rule also contain the conclusion of the association rule" [1]. Note that while the support is a measure of the frequency of a rule, the confidence is a measure of the strength of the relationship between sets of items [49].

For example, if the *Sup* of a rule $A \Longrightarrow C$ is 20% and its *Conf* is 95%, then the rule states that when A happens, there is a 95% probability of having C and that A and C are expressed together in 20% of the cases. In general, the user defines threshold values minimum_*Sup* and minimum_*Conf* values, for selecting meaningful rules [50–52].

The ConExp tool [45] is used in this work to extract the implications between properties, and the association rules. The ConExp tool shows association rules as: "$< x > Q = [\%] = > \langle y \rangle R$", which can be read as follows: "There are y objects having attribute R out of x objects having attribute Q. Or, "% of the objects having Q have R" [15].

2.3. Discussion

Formal Concept Analysis is an important approach for knowledge representation with applications in different areas. Starting from a given a formal context it is possible to perform visualization, navigation and information retrieval, and interpretation of concepts and rules. FCA can simultaneously consider a different type of data, such as real numbers, ordinal numbers, or nominal numbers. In general, many-valued attributes could be easily converted using a scaling transformation [39,63]. Note that no statistical assumption on data is required for performing an FCA. This fact represents a clear advantage of other statistical approaches (e.g., correlation or clustering analysis).

One of the main disadvantages of the presented approach is the fact that FCA could derive a very large number of formal concepts, even if a small formal context is used [54]. However, as mentioned in [53], the worst-case complexity ($O(2^{\min(|G|,|M|)})$ "is rarely found in practice" and special techniques of concept lattice reductions have been developed to "produce a concept lattice isomorphic to the original" [54] and cope with this characteristic (See [54,59] for a detailed analysis of the main approaches for reduction). As previously mentioned, in the case of multivalued tables, a transformation is necessary to obtain a suitable context table, but with an increase of the cardinality of the attribute set (examples of such transformations will be presented on the next section). One way to cope with this fact is the use of fuzzy FCA approaches. In these cases, fuzzy concept lattices are generated providing additional information [65] (e.g., membership values of objects in each fuzzy formal concept). However, in this paper, only crisp values will be considered. The interested reader could check the works in [65–68].

In the next section, several examples related to the reliability domain illustrate how FCA is performed.

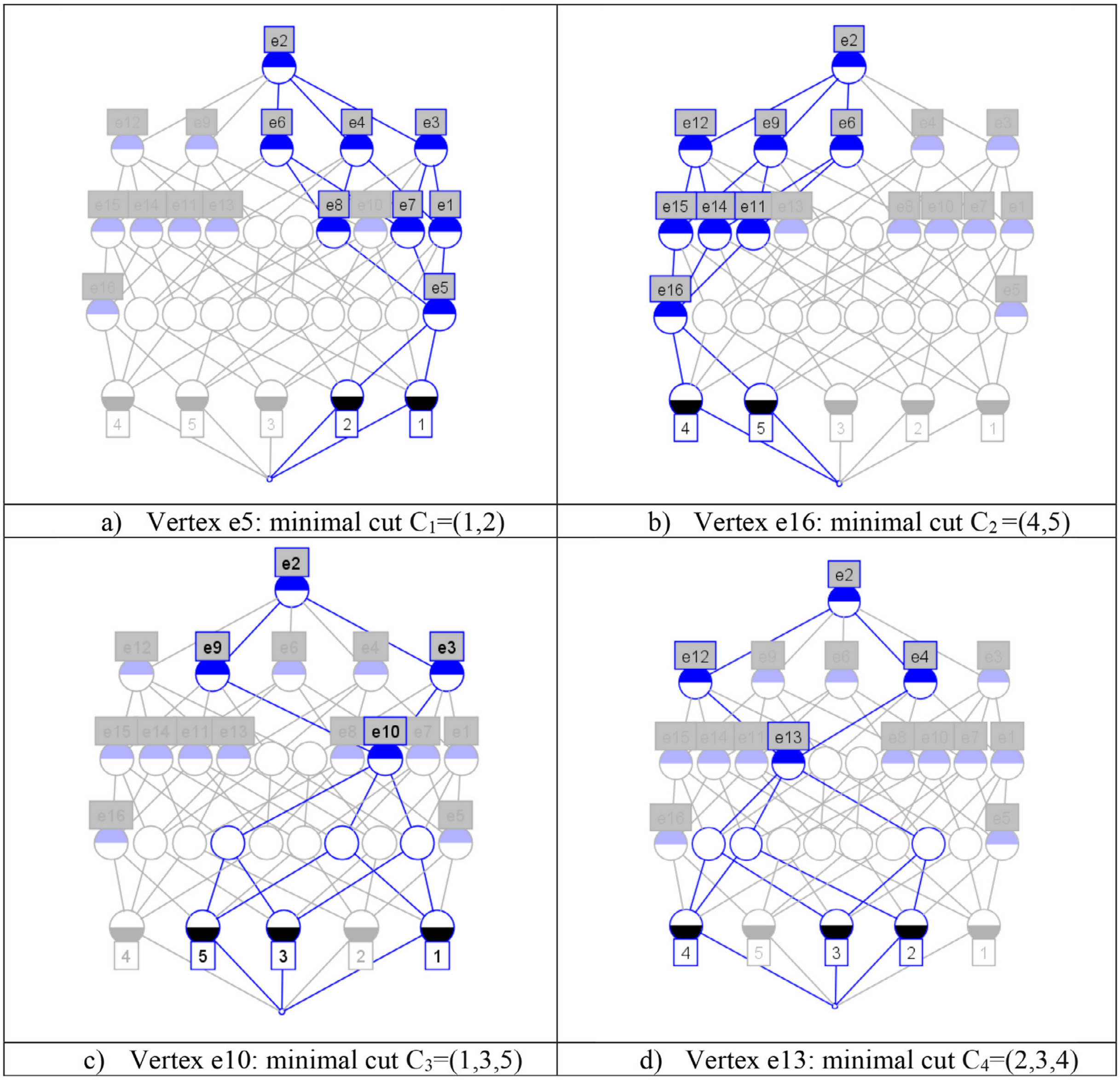

Fig. 5. Extents and intents of a selected vertex that defines the four minimal cut-sets. a) Vertex e5: minimal cut $C_1=(1,2)$. b) Vertex e16: minimal cut $C_2=(4,5)$. c) Vertex e10: minimal cut $C_3=(1,3,5)$. d) Vertex e13: minimal cut $C_4=(2,3,4)$.

3. Applications

In this section, four examples are analyzed using the FCA approach. The examples, related to well-known reliability experiences, are used to better understand the basic concepts of FCA and how the reliability area could benefit. Some examples are based on real data, while others have been adapted from the literature.

The first example describes the FCA approach based on the cut-sets in systems modeled as networks. The second example uses notions derived from the knowledge space theory [55] to analyze which protection strategy could be used to prevent different types of attack scenarios in a given network. The last two examples illustrate the extension of binary formal contexts to assess: a) failure events under different causes (levels of granularity); and b) the importance of nodes in an electric power system based on several measures of importance [56] (attributes with multiple values).

3.1. Cut-sets in a network

A cut-set is a minimum set of branches of a connected graph, so that when all of elements of the sets are removed from the graph, the graph is separated into two distinct parts [57]. In reliability analysis, a cut-set is defined as a set of components of a system which, when failed, causes system failure. Among all of the cut-sets, reliability analysis seeks the minimum cut set, that is any cut set which does not contain any other cut set as a subset, and all components of a minimal cut set must fail to cause system failure.

Fig. 3 shows a reliability block diagram of a network with five components. It is easy to show that there are 4 minimum cut-sets: $C_1=(1,2)$, $C_2=(4,5)$, $C_3=(1,3,5)$ and $C_4=(2,3,4)$, where (x,y) is referred to failed components x and y. The cardinality of each set defines its order. For example, $C_3=(1,3,5)$ is a third-order minimum cut-set. Note that high order minimum cut-sets do not include low-order minimum cut-sets (e.g., C_3 does include neither C_1 nor C_2).

To illustrate the FCA approach, let consider Table 2. This table shows the 16 failed state e_i of the system (The rest of the total $2^5=32$ states correspond to operational states and are not shown). This table will define the formal context of our example. Note that, as mentioned in the previous section, the table describes the relationship between components and the states of the system: if 1, then the failed component

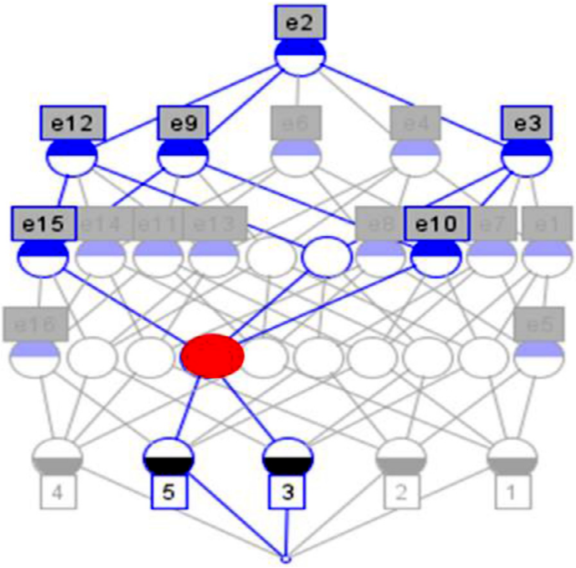

Fig. 6. Extents and intents associated with the filled (red) vertex. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

belongs to a state; 0 otherwise. For example, in state e_1, components 1, 2 and 3 are failed.

Fig. 4 shows the concept lattice corresponding to the formal context of Table 2. The lattice shows all the relations among states and components. Note that the upper node is not represented as an empty node, as in the previous example. Indeed, it has associated the label e2 meaning that this node includes all the states (concepts) downwards (remember that in state e2, all components are failed). On the other side, the lower node is empty, meaning that there is no component that simultaneously belongs to all states. The rest of the nodes with a label mimic the concept of cut-sets, but only a few correspond to minimal cut-sets.

Fig. 5 shows the extent and intents of selected nodes that define the four minimal cut sets. For example, Fig. 4a shows both intents and extents for the concept e5 that defines the minimal cut set $C_1 = (1,2)$. Indeed, concept e5 considers only components "failed" 1 and 2. The rest of intents clearly contain e5 and thus, by definition, are not minimal cut sets. A similar comment explains Fig. 4b: in this case, concept e16 defines the minimum cut-set $C_2 = (4,5)$. The analysis of Fig. 5 immediately suggests to the reader that the nodes labelled and with extents formed only by components are effectively minimal cut-sets. Note that in this example, an implication is interpreted as "a given minimum cut set $\Longrightarrow$ a cut set" or "a cut set is included in another cut set". For example: $e5 = (1,1,0,0,0) \Longrightarrow e1 = (1,1,1,0,0) \Longrightarrow e3 = (1,1,1,01)$.

The nodes with no label correspond to the union of components that are needed to build the lattice but do not define a failed state. For example, let us consider the intents and extents for the filled red vertex in Fig. 6. Components 3 and 5 are the two extents. Note that there is no state in Table 2 with only components 3 and 5 failed. A simple inspection of Fig. 3 shows that the system is operating when only components 3 and 5 are failed. It is interesting to note, that one of the intents associated with this red node is e15. This means that components 3 and 5 are included in state e15. However, e15 is not a minimum cut set since it includes the states e16 (see Fig. 5d).

The set of implications and rules derived by ConExp just refer to evident implications that could also be derived directly from the previous figures (for example $e_{10} \Longrightarrow e_3$, see Fig. 5c). So FCA ends at this point.

As mentioned in the introduction, FCA allows answering a set of questions related to components, states and cut-sets. Of course, for the example at hand, the questions posed are very easily answered even with no FCA. But for complex systems, where many min-cut-sets are derived, for example, using a Monte Carlo approach and some machine learning approach, such questions could be very difficult to be answered [69–71]. Finally, it is important to mention that the concept lattice can highlight the relations of generalization and specialization between attributes and objects.

3.2. Robust protection strategy on network under diverse attacks

In [60], the authors present a three-step approach for identifying a set of a robust defense strategy against a set of possible attacks to a network. Given a network and its performance function, and a set of possible attacks, the approach: 1) solves a bi-objective formulation, which minimizes system vulnerability (or, more specifically, maximizes survivability) while minimizing cost; 2) produces Pareto-optimal frontiers for each specific attack; and 3) makes use of a multi-criteria decision-making technique to aggregate the attacks together to find a set of common protection strategies, that is, a set of protection strategies that could be used on a set of specific attacks.

To illustrate how FCA could be used in this context, let consider the information presented in Table 3 (adapted from 61). In this case, the formal context consists of 10 protection strategies - objects, which could be used in 20 different attack scenarios - attributes.

Fig. 7a shows the corresponding concept lattice. From the concept lattice graphical layout, it is evident that several protection strategies could be used in many attack scenarios. However, there is no single strategy to protect all the possible scenarios. It is important to realize that strategy S2, could be only used in attack scenarios 5 and 6 (Fig. 7b). For this reason, the resulting concept lattice is asymmetric (a dense left branch with protection strategies for many attack scenarios). Also, note that attack scenario 5 could be only protected by strategy S2. Thus, attack scenario 5 is the most difficult scenario to protect.

Note that protection strategies S8 and S9 (objects of formal concept 12) could be indifferently selected for the same scenarios, that is, they have the same "values" for the attributes considered. In the FCA literature, objects that have the same set of attribute values are viewed as

Table 3
Formal context for protection strategies and attack scenarios (modified from [61]).

Strategy	Attack scenarios																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
S1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S2	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S3	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0
S4	0	1	1	0	0	1	1	0	0	1	1	1	0	0	0	1	1	1	1	0
S5	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	1	1	0
S6	0	1	0	0	0	1	1	0	0	1	1	0	1	0	0	1	1	1	1	0
S7	0	1	0	0	0	1	1	0	0	0	1	0	1	0	0	1	1	1	1	0
S8	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0
S9	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	1	1	0
S10	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1

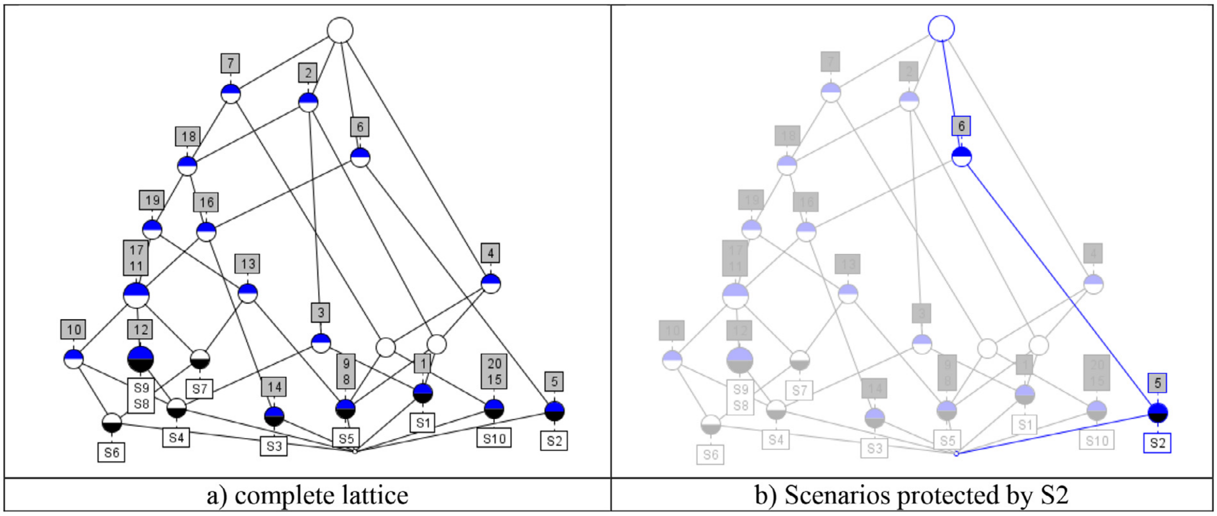

Fig. 7. Concept lattice for Table 3. a) complete lattice. b) Scenarios protected by S2.

Table 4
Some implications derived from Table 3.

	Implications	Support%
1	18 ⟹ 2,7	70
2	2,7 ⟹ 18	70
3	16 ⟹ 2,6,7,18	60
4	6,18 ⟹ 2,7,16	60
5	2,6 ⟹ 7,16,18	60
6	6,7 ⟹ 2,16,18	60
7	19 ⟹ 2,7,18	60
8	11 ⟹ 2,6,7,16,17,18,19	50
9	17 ⟹ 2,6,7,11,16,17,18,19	50
10	16,19 ⟹ 2,6,7,11,17,18	50
11	6,19 ⟹ 2,7,11,16,17,18	50

equivalent. Additionally, attributes possessed by the same set of objects are considered as equivalent (e.g., scenarios 11 and 17). This fact means that the original formal context could be reduced. ConExp allows performing redundant information removal such as clarifying and reducing. Of course, such actions on the input context are performed prior to the beginning of analysis. For additional details on these procedures, see [54].

Up to this point, the insightful reader may have noticed that the first FCA result, mentioned in the introduction (that is, groups that represent "natural" concepts) greatly facilitates the understanding of the information presented in Table 3, visualizing strategies and scenarios in a clearer way as well as their relationships.

As previously mentioned, it is possible to derive a list of association rules, i.e., a list of relational dependencies between attributes. By defining a minimal support of 10% (one object) and minimal confidence of 100% (only the clear implications between the attributes in the formal context) we found 36 association rules.

Table 4 shows a list of 11 association rules with a minimal support of at least 50% and confidence of 100%. For example, rule 1 has a support of 70%, that is, it is true for seven different protection strategies. It means that protection strategies (S3 to S9) used for attack scenario 18 could be always used for attack scenarios 2 and 7.

Let assume that minimum_*Sup* = 6 and minimum_*Conf* = 80%. ConExp generates the following rules with 80% < *Conf* < 100%:

< 8 > 7 = [88%] = > < 7 > 2 18;

< 8 > 2 = [88%] = > < 7 > 7 18;

< 7 > 2 7 18 = [86%] = > < 6 > 19;

< 7 > 2 7 18 = [86%] = > < 6 > 6 16;

< 7 > 6 = [86%] = > < 6 > 2 7 16 18;

For example, rule < 7 > 2 7 18 = [86%] = > < 6 > 19 states that 86% of the protection strategies that could be used for scenarios 2, 7 and 18 could be also used for scenario 19. Indeed, seven protection strategies (S3 to S9) could be used for scenarios 2, 7 and 18. But only six strategies (S4 to S9) could be used for scenario 19.

At this point, the second main role of the FCA, particularly the finding of implications, should be clear. Indeed, if the user knows, a priori, some implications, these could also be verified through a simple analysis of the data table (the formal context). If there are implications not known, the FCA is able to derive them.

Since the concept lattice is an acyclic directed graph, it is possible to order the protection strategies with respect to the number of extents. Fig. 8 shows the concept lattice generated using the transposed of the formal context of Table 3. Fig. 8a shows which attack scenarios are related to protection strategies while Fig. 8b shows the relations in term of number and percentage of extents. For example, S2 has 2 extents (attacks scenarios 5 and 6). Fig. 9b allows ordering the protection strategies with respect to the number of extents. One of such ordering is S2 – (S1 – S10) – S3 – S5 – (S7 – S8 – S9) – S6 – S4: strategies near the top of the list could be used in fewer attack scenarios while strategies at the end of the list cover many attacks scenarios (strategies in parenthesis have the same number of extents).

3.3. Formal concept analysis over attributes with levels of granularity

In many cases, the interpretation of a context table using FCA is not satisfactory because the number of formal concepts derived is too small for providing a clear understood. This problem is caused when the attributes selected are too "simple" and concentrate a broad meaning, that is, the set of attributes "does not provide a sufficiently fine granulation of the input objects". That means that in order to have a more precise reasoning, additional subdivisions related to the attribute under study are included. For example, let suppose that C = {technical cause, external cause}, represents the set of attributes related to the cause of failures in an electric distribution system. For example, a technical cause could be the erroneous opening of a circuit breaker while an external cause could be a car accident affecting a pole and power lines.

In order to have a better understanding, the level of granularity is increased (i.e., a finer granularity) by defining a granularity-level tree (gl-tree): a rooted tree with additional attributes. For example, we can divide the attribute external cause with attributes, weather-related and

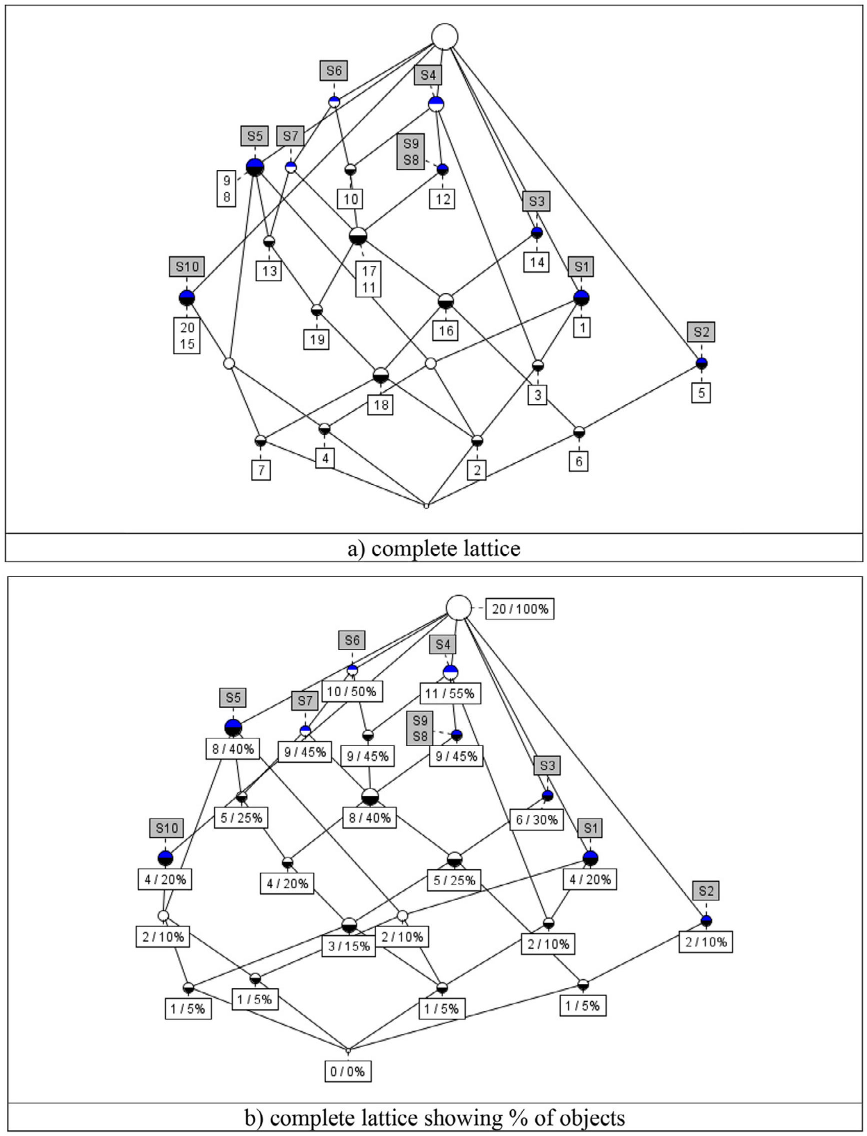

Fig. 8. concept lattice derived from Table 4 transposed. a) complete lattice. b) complete lattice showing% of objects.

car-accidents causes.

To illustrate the usage of levels of granularity, the Electric Emergency Incident and Disturbance Report (Form OE-417) is selected [62]. This report "collects information on electric incidents and emergencies in the USA. The Department of Energy uses the information to fulfill overall national security and other energy emergency management responsibilities, as well as for analytical purposes" [62]. From the available data, a subset of 18 events of the disturbance report, related to January 2018, for three regional electric reliability councils is selected (a): The South Reliability Corporation (SERC); b) Texas Reliability Entity (TRE); and c) Western Electricity Coordinating Council (WECC)).

Table 5 shows the formal context used. Here the set of objects represents events while the set of attributes represents the time (T) of the events, the regional council affected (R), the causes (C), and if the event causes load-shed (L). In the original report, there are two attributes to characterize a load-shed: the amount of load interrupted, and the amount of customer affected. Some events have an "unknown" label. For these events, labels "unknown" and "0 customers affected" are considered as no load-shed events.

Tree gl-trees are considered in this example: $T=$\{daytime, afternoon\}; R $=$\{SERC, TRE, WECC\}; $C=$ \{weather-related, technical, and attack\}.

For example, the first event happened in the afternoon, in the SERC region, was caused by weather conditions, with no load disrupted. Fig. 9 shows the lattice generated. Note that the option "Show object count" is selected in ConExp: a count and the percentage of the total objects are presented. It is interesting to realize that only 7 out of 18 events caused load-shed. In addition to examples of evident implications, like Attack = > Loadshed, it is possible to detect more elaborated examples. Indeed, Fig. 9 shows the implication WECC = > Loadshed, Afternoon. The list of implications detected by ConExp is:

1 < 4 > daytime Technical = > SERC;

2 < 3 > afternoon loadshed = > WECC;

3 < 3 > SERC loadshed = > daytime;

4 < 3 > WECC = > afternoon loadshed;

5 < 3 > Attack = > loadshed;

6 < 2 > afternoon TRE = > Weather;

7 < 1 > TRE loadshed = > daytime Attack;

8 < 1 > Weather loadshed = > daytime SERC;

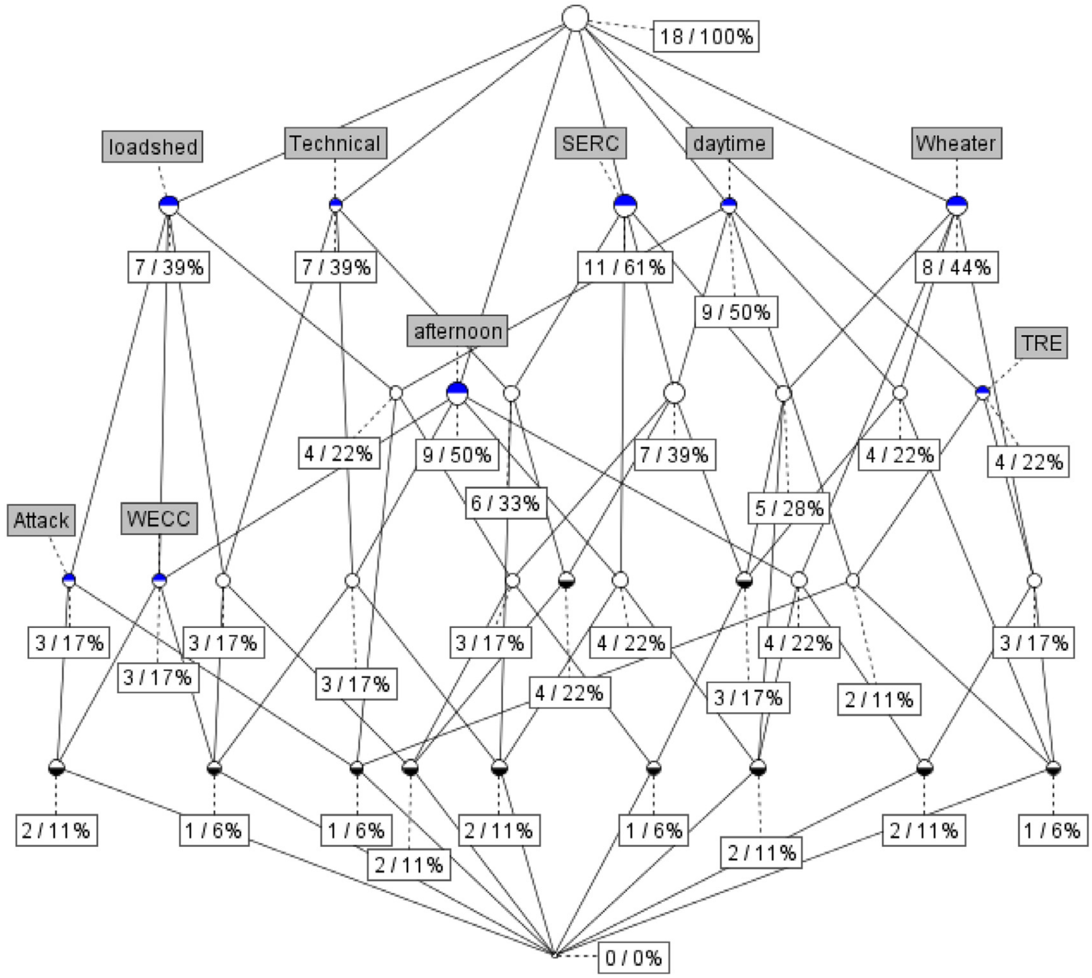

Fig. 9. concept lattice derived from the table.

Table 5
Formal context for failures events [62].

	Time		Region			Causes			
event	daytime	afternoon	SERC	TRE	WECC	Weather	Technical	Attack	Load-shed
1	0	1	1	0	0	1	0	0	0
2	0	1	0	1	0	1	0	0	0
3	0	1	1	0	0	0	1	0	0
4	1	0	1	0	0	1	0	0	1
5	1	0	1	0	0	1	0	0	0
6	1	0	0	1	0	0	0	1	1
7	1	0	0	1	0	1	0	0	0
8	0	1	1	0	0	1	0	0	0
9	0	1	1	0	0	0	1	0	0
10	0	1	0	1	0	1	0	0	0
11	1	0	1	0	0	0	1	0	1
12	1	0	1	0	0	0	1	0	0
13	1	0	1	0	0	1	0	0	0
14	1	0	1	0	0	0	1	0	1
15	1	0	1	0	0	0	1	0	0
16	0	1	0	0	1	0	0	1	1
17	0	1	0	0	1	0	1	0	1
18	0	1	0	0	1	0	0	1	1

9 < 1 > daytime Attack loadshed = > TRE;

By using minimum_*Sup* = 6 and minimum_*Conf* = 60%, ConExp derived the following rules:

1 < 18 > {} = [61%] = > < 11 > SERC;

2 < 9 > daytime = [78%] = > < 7 > SERC;

3 < 11 > SERC = [64%] = > < 7 > daytime;

4 < 7 > Technical = [86%] = > < 6 > SERC;

5 < 8 > Weather = [62%] = > < 5 > SERC;

6 < 4 > daytime Technical = [100%] = > < 4 > SERC;

Table 6
Multi-valued Pollutions indices (partially extracted from [63]).

ID	FC	CP
1	2	0
2	1	0
5	3	3
27	5	0
31	4	5

Table 7
Formal context for Table 6.

ID	FC2	FC3	FC4	FC5	CP3	CP5
1	1	0	0	0	0	0
2	0	0	0	0	0	0
5	1	1	0	0	1	0
27	1	1	1	1	0	0
31	1	1	1	0	1	1

7 < 6 > SERC Technical = [67%] = > < 4 > daytime;

This example could be easily extended to include additional attributes with different types of data. For example, the duration of the event (DE), an attribute reflecting conceptually a between relation, is modeled as an inter-ordinal scale (e.g., DE $\leq$ 1; 1〈DE $\leq$ 3; DE〉3).

3.4. Formal concept analysis over attributes with multivalue [63]

Previous examples have considered only binary relations, that is, an object has or has not an attribute. In many cases, the context table has attributes in the form of multi-value ordinal type and can assume an ordered range of scores. For example, Table 6 shows the partial information extracted from [63]. A set of five objects are evaluated using two pollution indices (attributes) FC and CP: the higher the attribute value the higher the pollution level. The observed values for FC and CP are {1,2,3,4,5} and {0,3,5} respectively. It is clear that, for example, object 27 shows the higher FC index, while object 31 shows the higher values for CP.

Table 6 must be transformed in order to be used in FCA. The transformation is called "scaling" [63]. As described in [39], the transformation consists on "replacing every many-valued attribute by the corresponding attribute-category pairs, with each object being described by one attribute-category pair per many-valued attribute". That means that the context table is built-up as a Boolean matrix where each attribute value occupies a column. Note that the lowest value of each attribute is considered as the default one. For example, attribute FC is converted to 4 attributes FC2, FC3, FC4 and FC5. Using this transformation, a value of "1", say in column FC2, means that the corresponding objects scored at least 2 for the FC attribute, i.e., the ordinal property of attribute FC is conserved. Table 7 shows the final formal context.

To illustrate this type of transformation, let us consider Table 8. In this table, objects represent the nodes of the IEEE 30 test power system, described as a network, and attributes represent the importance of each node, measured through different indexes (High values of indicators mean less importance of a node) [56]. For example, node 6 is considered the most important node using attributes AC, Pagerank, NWR, Degree and Betweenness.

Note that each attribute ranges from 1 to a maximum value of 30 (the number of nodes in the system). Since a complete transformation would require a very high number of additional columns (i.e., a high granularity) a simpler version that properly illustrates the approach is considered. As previously mentioned, additional approaches, based on fuzzy techniques, have been used to tackle this problem. However, here, only the classical transformation is presented.

Each attribute will be transformed using two alternatives based on the values from Table 8. For example, the two categories, High and Medium represent the rank of each node for each attribute considered. In this way, category High (H) is defined for nodes ranked between the

Table 8
Rank of node importance for the IEEE 30 power system [56].

Node_ID	AC_rank	DC_rank	Reliability_rank	Pagerank_rank	NWRank_rank	Degree	Betweenness_rank
1	28	26	16	22	24	23	21
2	4	2	8	6	7	5	10
3	22	23	24	23	22	21	20
4	14	13	14	7	3	4	3
5	26	27	21	24	25	22	22
6	**1**	14	3	**1**	**1**	**1**	**1**
7	7	3	19	25	16	15	19
8	2	**1**	26	27	18	13	23
9	29	29	5	10	9	8	13
10	9	10	2	2	2	2	2
11	30	30	28	29	29	29	23
12	6	5	7	3	4	3	4
13	12	28	22	30	28	28	23
14	19	15	25	21	20	16	23
15	15	9	6	5	8	6	8
16	21	16	20	18	17	19	18
17	10	11	17	20	15	17	15
18	24	17	10	14	19	26	16
19	13	8	15	13	23	27	16
20	20	19	11	17	13	18	14
21	8	4	23	26	21	14	23
22	17	20	9	11	11	10	11
23	18	18	12	19	14	20	12
24	11	12	13	9	10	11	7
25	25	22	4	8	12	12	9
26	23	21	30	28	30	30	23
27	5	6	**1**	4	5	7	5
28	3	25	18	12	6	9	6
29	27	24	27	15	26	25	23
30	16	7	29	16	27	24	23

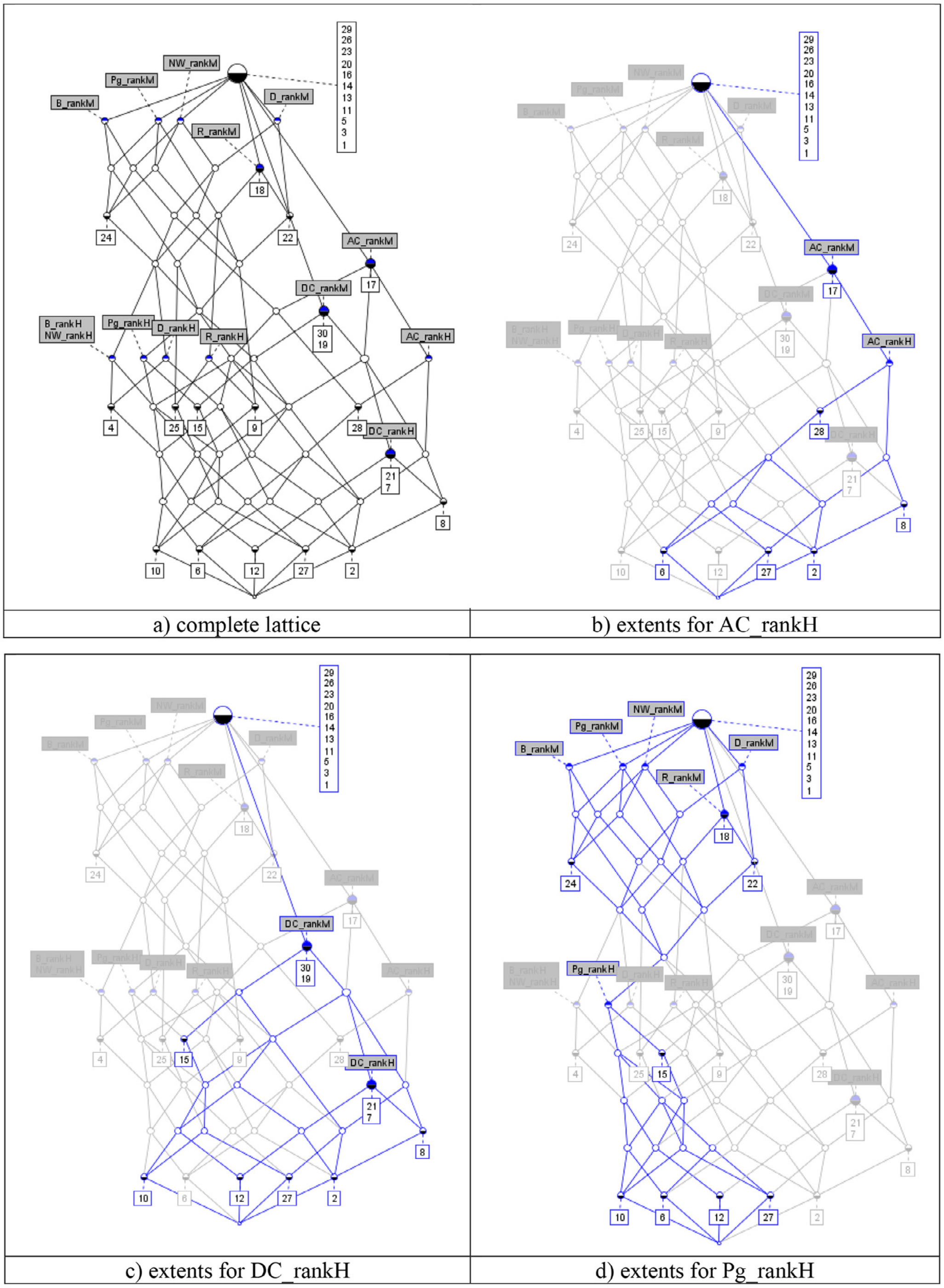

Fig. 10. Concept lattice for the IEEE 30 ranking example. a) complete lattice. b) extents for AC_rankH. c) extents for DC_rankH. d) extents for Pg_rankH.

first and the fifth position while category Medium (M) is for nodes ranked between sixth and 10th. Note that each node that scores a high rank in a specific attribute also scores the corresponding medium rank. The same reasoning applies to nodes that rank medium: they also rank better than the default score (i.e., rank better than 11th). In this way, the ordinal property of attributes is conserved.

Fig. 10a shows the concept lattice (59 concepts derived) for the corresponding multi-valued data. The bottom node is empty, indicating that there is no dominant node (i.e., no node rank better in all attributes). The set of objects at the lowest level {10, 6, 12, 27, 2} could be considered as the most important nodes in the network, according to the importance measures selected. On the other side, the set of objects at the highest level includes the least important nodes: none of these nodes scores on the defined scales (H or M).

At first glance, the analysis of the lattice shows trivial implications. For example, Fig. 10b shows the relationship between AC_rankH and AC_rankM. This trivial relationship is a direct consequence of setting the multi-valued context table (Table 9). The same behavior is observed in Fig. 11c, among DC_rankH and DC_rankM.

A more interesting relationship is presented in Fig. 10d. The selected concept Pg_rankH is related to six attributes, including the trivial implication with Pg_rankM. Also note that among the extents, there are 4 of the 5 most important nodes of the network: the set {6,10,12,27}. This set is also derived when the concept {B_rankH,NW_rankH} is selected. From Fig. 10a some equivalent attributes or objects are also detected, such as B_rankH and NW_rankH as well as nodes 7 and 21. It is

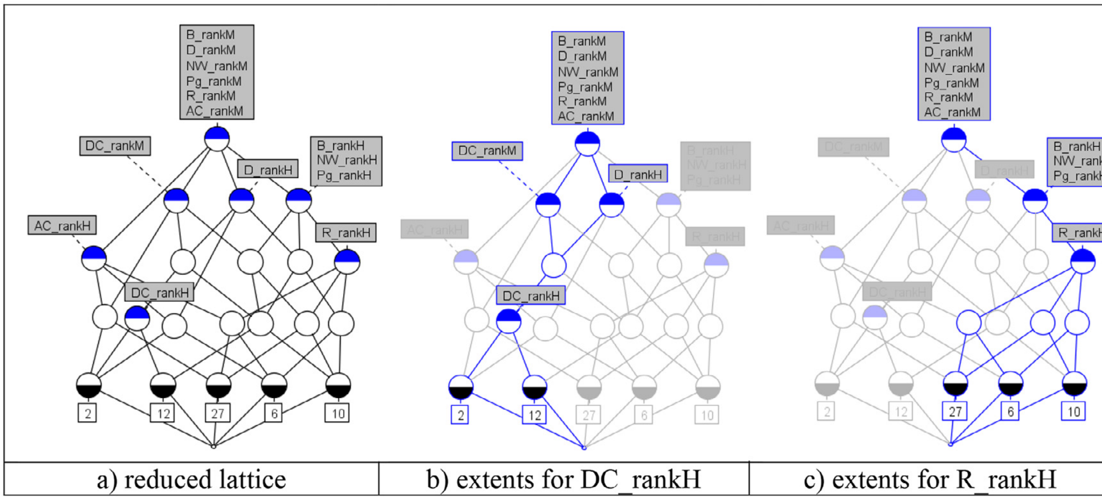

Fig. 11. Simplified concept lattice for the IEEE 30 ranking example showing the most important nodes. a) reduced lattice. b) extents for DC_rankH. c) extents for R_rankH. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

important to realize that this analysis is valid for the network considered. Other networks could produce different conclusions [64].

An additional fact that could be derived from the concept lattice is the presence of partial ordering among nodes. For example, in Fig. 10a, node 2 is more important than node 8. Additional chains could be derived.

By using minimum_*Sup*= 10 and minimum_*Conf* =80%, ConExp derived the following rules:

1 < 10 > B_rankM =[90%]= > < 9 > NW_rankM;

2 < 10 > B_rankM =[90%]= > < 9 > Pg_rankM;

3 < 10 > D_rankM =[90%]= > < 9 > NW_rankM;

4 < 10 > NW_rankM =[90%]= > < 9 > B_rankM;

5 < 10 > NW_rankM =[90%]= > < 9 > D_rankM;

6 < 10 > NW_rankM =[90%]= > < 9 > Pg_rankM;

7 < 10 > Pg_rankM =[90%]= > < 9 > B_rankM;

8 < 8 > Pg_rankM D_rankM =[100%]= > < 8 > NW_rankM;

9 < 8 > D_rankM B_rankM =[100%]= > < 8 > NW_rankM;

10 < 9 > NW_rankM B_rankM =[89%]= > < 8 > D_rankM;

11 < 9 > NW_rankM B_rankM =[89%]= > < 8 > Pg_rankM;

12 < 9 > NW_rankM D_rankM =[89%]= > < 8 > Pg_rankM;

13 < 10 > R_rankM =[80%]= > < 8 > D_rankM;

Fig. 11a shows a reduced concept lattice with the relationship among the attributes for the set of the most important nodes. Figs. 11b and 11c show the concepts required to define the set of the most important nodes (at least for the IEEE 30 case).

4. Conclusions

The aim of this paper is to introduce Formal Concept Analysis to the reliability practitioner and to show how the approach is easily applied in some situations that could exist in the reliability field. However, both areas of knowledge benefit from the applications presented.

FCA is a mathematical method for data analysis with emphasis on structure, relations and visualizations. FCA only requires an input context table (objects, attributes and binary relation details). As a result, FCA produces a clear graphical output of the data structure (formal concepts and order relations) and derives a set of implications and association rules (possibly hidden dependencies in the data). Both outputs would allow a better understanding of the data.

The results of the four examples analyzed by FCA confirm once again that the approach provides an effective way to analyze a set of data. For the reader familiar with FCA, it is certainly not a new fact. However, for the reliability practitioner, there are several aspects to highlight. First, the representation of data through a lattice turns out to be a tool of high utility and understanding. Indeed, objects and attributes can be viewed and related directly. For example, knowing immediately which components (objects) belong simultaneously to more than one cut-set is useful for maintenance purposes (example 3.1). Or, being able to quickly detect that there is only one valid protection strategy for an attack scenario, allows the decision-maker to define alternative protection schemes for a proper redundant system (example 3.2). Or, knowing the set of the most and the least important nodes in a network (example 3.4) could suggest a convenient restructuration in the network. Second, FCA allows detecting implications not known by the user. For example, in example 3.3, the occurrence of an event in the afternoon causes loadshed in a region. This fact could require additional insights from the operator of the system.

FCA analyzes situations in which it is possible to define a binary relation between a set of objects and a set of attributes. This is the case, for example, of fault diagnosis assessment, in which, for a certain event, the components involved are detailed. Of course, other applications could benefit from using an FCA-based approach, such as threat assessment, community detection in networks, software reliability, among others: only a formal context table is required.

CRediT authorship contribution statement

Claudio M. Rocco: Conceptualization, Methodology, Software, Investigation, Writing - review & editing, Supervision. **Elvis Hernandez-Perdomo:** Investigation, Writing - review & editing. **Johnathan Mun:** Writing - review & editing.

Declaration of Competing Interest

None

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ress.2020.107002.

References

[1] Yevtushenko S. Computing and visualizing concept lattices. Dissertationsschrift der Technischen Universitât Darmstadt 2004http://tuprints.ulb.tu-darmstadt.de/488/1/diss-yevtushenko.pdf.
[2] Wille R. Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival I, editor. Ordered sets. Nato advanced study institutes series (Series C — mathematical and physical sciences), 83. Dordrecht: Springer; 1982.
[3] Belohla Radim. Introduction to formal concept analysis, Department of computer science, faculty of Science, Palacky University. OLOMOUC 2008http://belohlavek.inf.upol.cz/vyuka/IntroFCA.pdf accessed on May, 30th,2019.
[4] Tilley T, Eklund P. Citation analysis using formal concept analysis: a case study in software engineering. 18th International workshop on database and expert systems applications (DEXA 2007), 3-7 Sept. 2007. Regensburg, Germany; 2007.
[5] Cole R, Eklund P. Application of formal concept analysis to information retrieval using a hierarchically structured thesaurus. International conference on conceptual graphs, iccs '96. Sydney: University of New South Wales; 1996. p. 1–12.
[6] Carpineto C, Romano G. Exploiting the potential of concept lattices for information retrieval with CREDO. J Univ Comput 2004;10(8):985–1013.
[7] Carpineto C, Romano G, et al. Using concept lattices for text retrieval and mining. In: Ganter B, editor. Formal concept analysis. LNAI, 3626. Springer; 2005. p. 161–71.
[8] Priss U. PhD Thesis. Relational concept analysis: semantic structures in dictionaries and lexical databases. Aachen: Verlag Shaker; 1998.
[9] Jiang G, Ogasawara K, Endoh A, Sakurai T. Context-based ontology building support in clinical domains using formal concept analysis. Int J Med Inf J 2003;71:71–81.
[10] Schnabel M. Representing and processing medical knowledge using formal concept analysis. Methods Inf Med 2002;41:33–48.
[11] Ganter B, Wille R. Conceptual scaling. In: Roberts F, editor. Applications of combinatorics and graph theory to the biological and social sciences. Berlin: Springer; 1989. p. 139–67.
[12] Bruggemann R, Bartel HG. A theoretical concept to rank environmentally significant chemicals. J Chem Inf Comput Sci 1999;39:211–7.
[13] Bruggemann R, Voigt K, Steinberg CEW. Application of formal concept analysis to evaluate environmental databases. Chemosphere 1997;35(3):479–86.
[14] Bartel H-G, Bruggemann R. Application of formal concept analysis to structure-activity relationships. Fresenius J Anal Chem 1998;361:23–8.
[15] Restrepo G, Basak SC, Mills D. Comparison of QSARs and characterization of structural basis of bioactivity using partial order theory and formal concept analysis: a case study with mutagenicity. Curr Comput Aided Drug Des 2011;7:109–21.
[16] Richards D, Compton P. Combining formal concept analysis and ripple down rules to support reuse. Software engineering knowledge engineering SEKE'97. Springer Verlag; 1997.
[17] Stumme G, Maedche A. FCA-MERGE: bottom-up merging of ontologies. IJCAI 2001. 2001. p. 225–34.
[18] Cimiano P, Hotho A, Staabw S. Learning concept hierarchies from text corpora using formal concept analysis. J Artif Intell Res (JAIR) 2005:305–39.
[19] Busch P, Richards D. Modeling tacit knowledge via questionnaire data. ICFCA lnai 2961. Springer; 2004. p. 321–8.
[20] Endres D, Adam R, Giese MA, Noppeney U. Understanding the semantic structure of human fmri brain recordings with formal concept analysis eds In: Domenach F, Ignatov DI, Poelmans J, editors. Formal concept analysis. ICFCA 2012. Lecture notes in computer science, 7278. Berlin, Heidelberg: Springer; 2012.
[21] Formica Anna. Concept similarity in Formal Concept Analysis: An information content approach. Knowl Based Syst 2008;21(1):80–7.
[22] Li ST, Tsai FC. A fuzzy conceptualization model for text mining with application in opinion polarity classification. Knowl Based Syst 2013;39:23–33.
[23] Yang YP, Shieh HM, Tzeng GZ, Yen L, Shao MW. Combined rough sets with flow graph and formal concept analysis for business aviation decision-making. J Intell Inf Syst 2011;36(3):347–66.
[24] Zhao J, Liu L. Construction of concept granule based on rough set and representation of knowledge-based complex system. Knowl Based Syst 2011;24(6):809–15.
[25] Sun ZeJun, Wang Bin, Sheng JinFang, Hu YiXiang, Wang YiHan, Shao JunMing. Identifying influential nodes in complex networks based on weighted formal concept analysis. IEEE access 2017;5:3777–89. Date of Publication: 07 March 2017.
[26] De Maio C, Fenza G, Gaeta M, Loia V, Orciuoli F, Senatore S. RSS-based e-learning recommendations exploiting fuzzy FCA for knowledge modeling. Appl Soft Comput 2012;12(1):113–24.
[27] Priss U. Formal concept analysis in information science. Ann Rev Inf Sci Technol 2006;40(1):521–43.
[28] Poelmans J, Elzinga P, Dedene G, et al. Retrieval of criminal trajectories with an FCA-based approach. In: Kuznetsov O, editor. Proceedings of fcair, 977. Moscow; 2013. p. 83–94.
[29] Romanov V, Poluektova A, Sergienko O. Adaptive EIS with business rules discovered by formal concept analysis. In: Moller C, Chaudhry S, editors. *Reconceptualizing enterprise information systems*, lecture notes in business information processing, 105. Berlin/ Heidelberg: Springer; 2012. p. 105–17.
[30] Elzinga P, Viaene S, Poelmans J, Dedene G, Morsing S. Terrorist threat assessment with formal concept analysis. Proceedings of the 2010 IEEE International Conference on Intelligence and Security Informatics. 2010. p. 77–82.
[31] Amin II, Kassim SK, Hassanien AE, Hefny HA. Formal concept analysis for mining hypermethylated genes in breast cancer tumor subtypes. Proceedings of 12th ISDA, 2012. 2012. p. 764–9.
[32] Atif J, Hudelot C, Bloch I. Explanatory reasoning for image understanding using formal concept analysis and description logics. IEEE Trans Syst Man Cybernet A 2014;44(4):552–70.
[33] Antoni L, Krajci S, Kridlo O, Macek B, Piskova L. On heterogeneous formal contexts. Fuzzy Sets Syst 2014;234:22–33.
[34] Klimeš J. Formal concept analysis applied to the prediction of additives for galvanizing process. Metalurgija Croatian Metall Soc 2010;49(2):95–8.
[35] Hashemi RR, Le Blanc LA, Kobayashi T. Formal concept analysis for investigation of normal accidents. Int J Gen Syst 2004;33(5):469–84. https://doi.org/10.1080/03081070412331283789.
[36] Poelmans J, Kuznetsov SO, Ignatov DI, Dedene G. Formal concept analysis in knowledge processing: a survey on applications. Expert Syst Appl 2013;40(16):6538–60.
[37] Kumar SP, KCh Aswani, Gani A. A comprehensive survey on formal concept analysis, its research trends and applications. Int J Appl Math Comput Sci 2016;26(2):495–516.
[38] Ganter B, Wille R. Formal concept analysis: mathematical foundations. Berlin: Springer-Verlag; 1998.
[39] Carpineto C, Romano G. Concept data analysis: theory and applications. Hoboken: John Wiley & Sons; 2004.
[40] Kropat E. Formal Concept Analysis, AACIMP 2011. 6th Summer School "Achievements and Applications of Contemporary Informatics, Mathematics and Physics ", Aug 8-20. Ukraine; 2011. Kyiv https://www.slideshare.net/ssakpi/formal-concept-analysis.
[41] Pattison, T. (2014). "Interactive visualization of formal concept lattices." ED/GViP@Diagrams.
[42] Freeman LC. Cliques, Galois lattices, and the structure of human social groups. Soc Networks 1996;18(3):173–87. August.
[43] Ignatov DI. Introduction to formal concept analysis and its applications in information retrieval and related fields eds In: Braslavski P, Karpov N, Worring M, Volkovich Y, Ignatov D, editors. Information retrieval. russir 2014. communications in computer and information science, 505. Cham: Springer; 2015.
[44] Brüggemann R, Patil G. Ranking and Prioritization for Multi-Indicator Systems. Introduction to partial orders applications. Environmental and ecological statistics 2011.
[45] Yevtushenko SA. System of data analysis "Concept Explorer" In Russian Proceedings of the 7th national conference on Artificial Intelligence KII-20002000. p. 127–34. ConExp-2000.
[46] Toprak M. A Lattice-Based Approach For News Chain Construction. A master of science thesis submitted to the Graduate School of Engineering and Sciences of İzmir Institute of Technology July 2015. 2015.
[47] Quintero Reyes NY. A thesis of Doctor in Biotechnology, Universidad de Antioquia, Medellin. Colombia; 2018.
[48] Bruggemann R, Restrepo G. Estimating octanol/water partition coefficients by order preserving mappings. Croat Chem Acta 2013;86:509–17.
[49] Dasseni E, Verkios VS, Elmagarmid AK;, Bertino E. Hiding Association Rules by Using Confidence and Support. Comput Sci Tech Reports 2000. Paper 1488 http://docs.lib.purdue.edu/cstech/1488.
[50] Lenca P, Meyer P, Vaillant B, Lallich S. On selecting interestingness measures for association rules: user oriented description and multiple criteria decision aid. Eur J Oper Res 2008;184:610–28.
[51] Lattice Miner. https://sourceforge.net/projects/lattice-miner/.
[52] Quintero NY, Bruggemann R, Restrepo G. Mapping Posets into low dimensional spaces: the case of U trappers. MATCH communications in mathematical and in computer chemistry. 2018 available at http://match.pmf.kg.ac.rs/electronic_versions/Match80/n3/match80n3_793-820.pdf.
[53] Godin R, Saunders E, Gecsei J. Lattice model of browsable data spaces. Inf Sci (Ny) 1986;40:89–116http://dx.doi.org/10.1016/0020-0255(86)90001-0.
[54] Dias SM, Vieira NJ. Concept lattices reduction: definition, analysis and classification. Expert Syst Appl 2015;42:7084–97. 2015.
[55] Korossy K. Modeling Knowledge as Competence and Performance Eds. In: Albert D, Lukas J, editors. Knowledge spaces: theories, empirical research, and applicationsMahwah, New Jersey: Lawrence Erlbaum Associates; 1999. p. 103–32.
[56] Li J, Dueñas-Osorio L, Chen C, Shi C. AC Power Flow Importance Measures Considering Multi-Element Failures. Reliab Eng Syst Saf 2017;160(April):89–97. https://doi.org/10.1016/j.ress.2016.11.010.
[57] Brînzei N, Aubry JF. Graphs models and algorithms for reliability assessment of coherent and non-coherent systems. 232. 2018. p. 201–15. 2018.
[58] Ahmad Waqar, Hasan Osman, Pervez Usman, Qadir Junaid. Reliability modeling and analysis of communication networks. J Netw Comput Appl 2017;78:191–215.
[59] Yan Hehua, Zou Caifeng, Liu Jianqi, Wang Zhonghai. Formal concept analysis and concept lattice: perspectives and challenges. Int. J. Autonom Adaptive Commun Syst 2015;8(1):81.
[60] Ramirez-Marquez JE, Rocco CM, Barker K. Bi-objective vulnerability reduction formulation for a network under diverse attacks. J Risk Uncertainty Eng Syst 2017;3(4):04017025. 2017.
[61] Škopljanac-Mačina F, Blašković B. Formal concept analysis. Overview Appl Proc Eng 2014;69:1258–67.
[62] Electric Disturbance Events (OE-417). http://www.oe.netl.doe.gov/oe417.aspx http://www.oe.netl.doe.gov/download.aspx?type&OE417XLS&ID=78.
[63] Annoni P, Bruggemann R. The dualistic approach of FCA: a further insight into Ontario Lake sediments. Chemosphere 2008;70:2025–31.
[64] Analysis of importance measures for node ranking in electric power systems: some results based in order theory. To be submitted.
[65] Cao Tho TQuanSiu Cheung HuiTru Hoang. A fuzzy FCA-based approach to conceptual clustering for automatic generation of concept hierarchy on uncertainty

data. Conference: Proceedings of the CLA 2004 International Workshop on Concept Lattices and their Applications. 2004. p. 1–12. September 23-24.

[66] Butka P, Pócs J, Pócsová J. Representation of fuzzy concept lattices in the framework of classical FCA. J Appl Math Hindawi Publishing Corporation 2013:7. Article ID 236725.

[67] Kerber A, Bruggemann R. Problem Driven Evaluation of Chemical Compounds and Its Exploration. MATCH Commun. Math. Comput. Chem 2015;73:577–618. 2015.

[68] Brito A, Barros L, Laureano E, Bertato F, Coniglio M. Fuzzy formal concept analysis. In: Barreto G, Coelho R, editors. Fuzzy information processing. NAFIPS 2018. Communications in computer and information science, 831. Cham: Springer; 2018. p. 192–205.

[69] Rocco C. A rule induction approach to improve monte carlo system reliability assessment. Reliab Eng Syst Safety 2003:87–94. 82/1.

[70] Ramirez-Marquez JE, Rocco CM. “Evolutionary optimization technique for multi-state two-terminal reliability allocation in multi-objective problems. IIE Trans, IIE Trans 2010;42:539–52.

[71] Rocco CM, Miller AJ, Moreno JA, Carrasquero N. Sensitivity and uncertainty analysis in optimisation programs using an evolutionary approach: a maintenance application. Reliab Eng Syst Safety 2000;63:249–56.