



Triadic Exploration and Exploration with Multiple Experts

Maximilian Felde^{1,2}(✉)  and Gerd Stumme^{1,2} 

¹ Knowledge and Data Engineering Group, University of Kassel, Kassel, Germany

² Interdisciplinary Research Center for Information System Design,
University of Kassel, Kassel, Germany
{felde, stumme}@cs.uni-kassel.de

Abstract. Formal Concept Analysis (FCA) provides a method called *attribute exploration* which helps a domain expert discover structural dependencies in knowledge domains that can be represented by a formal context (a cross table of objects and attributes). Triadic Concept Analysis is an extension of FCA that incorporates the notion of conditions. Many extensions and variants of attribute exploration have been studied but only few attempts at incorporating multiple experts have been made. In this paper we present *triadic exploration* based on Triadic Concept Analysis to explore *conditional attribute implications* in a triadic domain. We then adapt this approach to formulate attribute exploration with multiple experts that have different views on a domain.

Keywords: Formal concept analysis · Triadic concept analysis · Attribute exploration

1 Introduction

Attribute exploration [3] is a well established knowledge acquisition method from the field of Formal Concept Analysis (FCA) [8]. Attribute exploration works on domains that can be represented as binary tabular data of objects and attributes (also called features or properties). It helps a domain expert to uncover the dependency structure of attributes of the domain. For non-binary tabular data the method of *conceptual scaling*, cf. [7], can be used to transform non-binary attributes into binary ones.

Attribute exploration is based on the idea that we extend domain information through a domain expert. To this end, attribute exploration uses a question-answer scheme to extract dependency information about attributes. The questions are in the form of *implications*, for example, *do attributes A and B imply attribute C?* (also written as $AB \rightarrow C?$). The expert's task is to confirm or refute the validity of such implications in the domain. If the expert refutes the validity of an implication she has to offer a counterexample, for example, in case of the question $AB \rightarrow C?$ an object of the domain that has the attributes *A* and *B* but lacks attribute *C*.

The attribute exploration algorithm asks these questions in an optimized manner such that the expert has to answer as few questions as possible until the validity of every conceivable implication can be inferred from the answers given by the expert. This is the case when every implication either follows from the set of implications accepted as valid or is contradicted by one of the examples given by the expert.

The basic version of attribute exploration requires an all-knowing expert of the domain, i.e. an expert who can answer any question about the domain correctly. It was introduced by Ganter in [3]. Since then, many variants and extensions of attribute exploration have been studied. A good overview can be found in the book *Conceptual Exploration* by Ganter and Obiedkov [6]. These extensions and variants notably include: Attribute exploration with background knowledge and exceptions [4, 18], where the idea is to support the exploration with prior knowledge about some of the relations between attributes, for example if one attribute is the negation of another; attribute exploration with partial information [11–13], where the expert is not required to be all-knowing and is also allowed to answer *I do not know* in addition to confirming or refuting a question. Further, the expert is not required to fully specify a counterexample as long as the specified parts contradict the implication in question; and a sketch of how to explore triadic formal contexts [5, 6], where the idea of attribute exploration is transferred to triadic concept analysis (an extension of FCA with conditions [17]). We elaborate further on this in Sect. 3.

However, most of the extensions and variants of attribute exploration that have been studied are based on the idea of a single expert answering the questions. As far as we know, there exist only a few papers that mention exploration with multiple experts, notably: Paper [16] deals with how to perform exploration in parallel and potentially offers a way to speed up the exploration with multiple experts; [10] addresses collaborative conceptual exploration based on the notions of local experts for subdomains of a given knowledge domain; and [2] studies attribute exploration in a collaborative exploration setting with multiple experts who share the same view on the domain but only have partial knowledge thereof.

When we explore a domain with multiple experts, one of the fundamental problems we face is that different views on a domain, for example different opinions whether an object has an attribute or not, or whether an implication is valid or not in a domain, are impossible to resolve by combining different pieces of information into one. Either, because there is no clear *right* or *wrong*, e.g. in case of opinions, or simply because we can not know which information to trust most. And, even if we used methods such as majority-voting on information, there is a reasonable chance that the result is not always correct. Combined with the inherent non-robustness of implication theories, i.e., small changes in the underlying data can lead to a very different theory, this suggests that merging different views on a domain is a bad idea for attribute exploration. If we take a closer look at the publications mentioned before, we see that all three avoid this issue in their own way. In [16] the experts all have the same complete knowledge about the domain; in [10] the local experts have partial knowledge about the same consistent domain knowledge; and, in [2] the problem was also avoided by defining expert knowledge as partial knowledge of some consistent domain knowledge.

Attribute exploration where multiple experts can have truly different and even opposing views on the domain has to the best of our knowledge not yet been studied. To this end we develop *triadic exploration* based on ideas presented by Ganter and Obiedkov in [5]. We then adapt triadic exploration to the setting of multiple experts with different views on a domain and thus provide a step in the direction of attribute exploration with multiple experts.

The paper is structured as follows: We begin by giving a brief introduction to the problem in Sect. 1. We recollect some fundamentals of Formal and Triadic Concept Analysis in Sect. 2, in particular *formal* and *triadic contexts*, *attribute implications*, the *relative canonical base* and *attribute exploration*. In Sect. 3, we discuss implications in the triadic setting, in particular, we focus on *conditional attribute implications*. Subsequently, we formulate *triadic exploration*. In Sect. 4, we discuss how to adapt *triadic exploration* to model attribute exploration with multiple experts with different views. Finally, Sect. 5 contains conclusion and outlook. Note that for this paper we do not provide a separate section for related work, instead we address related work throughout the paper whenever appropriate.

2 Dyadic and Triadic Formal Contexts

In this section we recollect the fundamentals of (dyadic) Formal Concept Analysis and Triadic Formal Concept Analysis (TCA). We mostly rely on [8, 19] for FCA and on [17, 20] for TCA. We begin with the definition of *formal contexts* and associated notions. We then introduce *triadic contexts* and give an example which will serve as our running example for the remainder of this paper. Afterwards, we briefly cover *attribute implications*, the *relative canonical base* and *attribute exploration*. This serves as a foundation for Sect. 3, where we look at *implications in the triadic setting* and subsequently develop *triadic exploration*.

2.1 Formal Concept Analysis

Formal Concept Analysis was introduced by Wille in [19]. As the theory matured, Ganter and Wille compiled the mathematical foundations of the theory in [8]. A *formal context* $\mathbb{K} = (G, M, I)$ consists of a set G of objects, a set M of attributes and an incidence relation $I \subseteq G \times M$ with $(g, m) \in I$ meaning *object g has attribute m* . We define two derivation operators $(\cdot)′ : \mathcal{P}(M) \rightarrow \mathcal{P}(G)$ and $(\cdot)′ : \mathcal{P}(G) \rightarrow \mathcal{P}(M)$ in the following way: For a set of objects $A \subseteq G$, the set of *attributes common to the objects in A* is provided by $A′ := \{m \in M \mid \forall g \in A : (g, m) \in I\}$. Analogously, for a set of attributes $B \subseteq M$, the set of *objects that have all the attributes from B* is provided by $B′ := \{g \in G \mid \forall m \in B : (g, m) \in I\}$. A *formal concept* of a formal context $\mathbb{K} = (G, M, I)$ is a pair (A, B) with $A \subseteq G$ and $B \subseteq M$ such that $A′ = B$ and $A = B′$. We call A the *extent* and B the *intent* of the formal concept (A, B) . The set of all formal concepts of a context \mathbb{K} is denoted by $\mathfrak{B}(\mathbb{K})$. Note that for any set $A \subseteq G$ the set $A′$ is the intent of a concept and for any set $B \subseteq M$ the set $B′$ is the extent of a concept.

The subconcept-superconcept relation on $\mathcal{B}(\mathbb{K})$ is formalized by: $(A_1, B_1) \leq (A_2, B_2) :\Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_1 \supseteq B_2)$. The set of concepts together with this order relation $(\mathfrak{B}(\mathbb{K}), \leq)$ forms a complete lattice, the *concept lattice*. The vertical combination of two formal contexts $\mathbb{K}_i = (G_i, M, I_i), i \in \{1, 2\}$ with the same set of attributes M is called the *subposition* of \mathbb{K}_1 and \mathbb{K}_2 . Formally, it is defined as $(\hat{G}_1 \cup \hat{G}_2, M, \hat{I}_1 \cup \hat{I}_2)$, where $\hat{G}_i := \{i\} \times G$ and $\hat{I}_i := \{(i, g), m) \mid (g, m) \in I_i\}$ for $i \in \{1, 2\}$. The *subposition* of a set of contexts on the same set of attributes is defined analogously and we denote this by $subpos(\cdot)$.

2.2 Triadic Concept Analysis

Triadic Concept Analysis (TCA) was introduced by Lehmann and Wille in [17] as an extension to Formal Concept Analysis with conditions. In particular they introduced the notion of *triadic concepts* for which Wille proceeded to show the basic theorem of triadic concept analysis in [20] – clarifying the connection between triadic concepts and complete tri-lattices, analogous to the dyadic case.

The basic structure in TCA is a *triadic context* which is similar to the formal context in FCA. A *triadic context* is defined as a quadruple $\mathbb{T} = (G, M, B, Y)$, where G, M and B are sets and $Y \subseteq G \times M \times B$ is a ternary relation on these sets. The elements of G, M and B are called objects, attributes and conditions respectively. For $g \in G, m \in M$ and $b \in B$ with $(g, m, b) \in Y$ we say that *object g has attribute m under condition b* . The conditions are understood in a broad sense, cf. [20]: They comprise, amongst others, relations, interpretations, meanings, purposes and reasons concerning the connections of objects and attributes.

Example 1. The following example¹ will serve as our running example throughout the paper. It shows the situation of public transport at the train station Bf. Wilhelmshöhe with direction to the city center in Kassel. From Bf. Wilhelmshöhe you can travel by one of four bus lines (52, 55, 100 and 500), four tram lines (1, 3, 4 and 7), one night tram (N3) and one regional tram (RT5) to the city center. These are the objects $G_{\text{ex.}}$ of our context. The buses and trams leave the station at different times throughout the day. The attributes $M_{\text{ex.}}$ of our context are the aggregated leave-times, more specifically, we have split each day in five distinct time-slots: early morning (4:00 to 7:00), working hours (7:00 to 19:00), evening (19:00 to 21:00), late evening (21:00 to 24:00) and night (0:00 to 4:00). The conditions $B_{\text{ex.}}$ of our context are the days of the week. A bus or tram line is related to a time-slot on a day if a bus or tram of this line leaves the station at least once during the time-slot on the day. This describes the ternary relation $Y \subseteq G_{\text{ex.}} \times M_{\text{ex.}} \times B_{\text{ex.}}$. We have aggregated Monday to Friday into a single condition, because the schedule is the same for these days. Thus, we obtain the context $\mathbb{T}_{\text{ex.}} = (G_{\text{ex.}}, M_{\text{ex.}}, B_{\text{ex.}}, Y)$. The resulting triadic context can be found in Fig. 1.

Naturally, we can view the triadic context as a family of formal contexts, where each context represents one condition, basically slicing the triadic context

¹ The example is similar to the one given in [6], which inspired it.

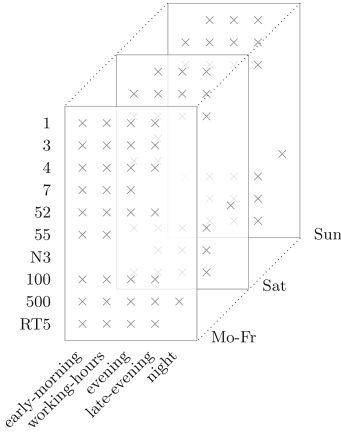


Fig. 1. Triadic context \mathbb{T}_{ex} . of Example 1

	Mo-Fr	early-morning	working-hours	evening	late-evening	night
1		x	x	x	x	
3		x	x	x	x	
4		x	x	x	x	
7		x	x	x		
52		x	x	x	x	
55		x	x			
N3						
100		x	x	x	x	
500		x	x	x	x	
RT5		x	x	x	x	

	Sat	early-morning	working-hours	evening	late-evening	night
1		x	x	x	x	
3		x	x	x	x	
4		x	x	x	x	
7			x			
52		x	x	x	x	
55						
N3						x
100		x	x	x	x	
500		x	x	x	x	
RT5		x	x	x	x	

	Sun	early-morning	working-hours	evening	late-evening	night
1			x	x	x	
3			x	x	x	
4			x	x	x	
7						
52						
55						
N3						x
100			x	x	x	
500			x	x	x	
RT5			x	x	x	

Fig. 2. The triadic context \mathbb{T}_{ex} . from Example 1 represented as context family of the condition contexts \mathbb{K}_{Mo-Fr} , \mathbb{K}_{Sat} and \mathbb{K}_{Sun}

vertically along the conditions. In Fig. 2 we provide the resulting context family of our running example.

Formally such a family of contexts representing a triadic context $\mathbb{T} = (G, M, B, Y)$ is a set of contexts \mathbb{K}_b , $b \in B$ where $\mathbb{K}_b := (G, M, I_b)$ with $(g, m) \in I_b \Leftrightarrow (g, m, b) \in Y$. We will refer to the contexts \mathbb{K}_b as *condition contexts* of the triadic context \mathbb{T} ; for our example these are \mathbb{K}_{Mo-Fr} , \mathbb{K}_{Sat} and \mathbb{K}_{Sun} .

2.3 Attribute Implications

Attribute implications are used to describe dependencies between attributes in a formal context. In the following we give a brief introduction. Let M be a set of attributes. (For a start, we do not require it to be related to a specific context.) An *attribute implication* over M is a pair of subsets $A, B \subseteq M$ of M . We denote this by $A \rightarrow B$. We call A the *premise* and B the *conclusion* of the implication $A \rightarrow B$.

We denote the set of all implications over a set M by $\text{Imp}_M = \{A \rightarrow B \mid A, B \subseteq M\}$.

A subset $T \subseteq M$ *respects* an attribute implication $A \rightarrow B$ over M if $A \not\subseteq T$ or $B \subseteq T$. We then also call T a *model* of the implication. T *respects a set* \mathcal{L} of implications if T respects all implications in \mathcal{L} . An implication $A \rightarrow B$ *holds* in a set of subsets of M if each of these subsets respects the implication.

For a formal context $\mathbb{K} = (G, M, I)$ we say that an implication $A \rightarrow B$ over M *holds in the context* if for every object $g \in G$ the object intent g' respects the implication. We then also call $A \rightarrow B$ a *valid implication* of \mathbb{K} . An implication $A \rightarrow B$ holds in \mathbb{K} if and only if every object $g \in G$ that has all attributes in A also has all attributes in B . Further, an implication $A \rightarrow B$ holds in \mathbb{K} if and only

if $B \subseteq A''$, or equivalently $A' \subseteq B'$. An implication $A \rightarrow B$ follows from a set \mathcal{L} of implications over M if each subset of M respecting \mathcal{L} also respects $A \rightarrow B$. A family of implications is called *closed* if every implication following from \mathcal{L} is already contained in \mathcal{L} . Closed sets of implications are also called *implication theories*.

Relative Canonical Base. The set of all implications that hold in a given context \mathbb{K} have a canonical irredundant representation which is called the *canonical base*, cf. [8,9]. Stumme has generalized this representation to the case where some (background) implications are known [18], i.e. attribute implications that are known to hold based on prior knowledge.

Given a formal context $\mathbb{K} = (G, M, I)$ and a set of (background) implications \mathcal{L}_0 on M that hold in the context \mathbb{K} , a *pseudo-intent* of \mathbb{K} relative to \mathcal{L}_0 is a set $P \subseteq M$ where P respects \mathcal{L}_0 , $P \neq P''$ and if $Q \subseteq P$, $Q \neq P$, is a relative pseudo-intent of \mathbb{K} then $Q'' \subseteq P$. The set $\mathcal{L}_{\mathbb{K}, \mathcal{L}_0} := \{P \rightarrow P'' \mid P \text{ relative pseudo-intent of } \mathbb{K}\}$ is called the *canonical base* of \mathbb{K} relative to \mathcal{L}_0 , or simply the *relative canonical base*. All implications in $\mathcal{L}_{\mathbb{K}, \mathcal{L}_0}$ hold in \mathbb{K} .

Theorem 1. (see [5,18]). *If all implications of \mathcal{L}_0 hold in \mathbb{K} , then*

1. *each implication that holds in \mathbb{K} follows from $\mathcal{L}_{\mathbb{K}, \mathcal{L}_0} \cup \mathcal{L}_0$, and*
2. *$\mathcal{L}_{\mathbb{K}, \mathcal{L}_0}$ is irredundant w.r.t. 1.*

The notion of a relative canonical base combined with Theorem 1 allows us to reduce the amount of questions that need to be posed during a triadic exploration.

2.4 Attribute Exploration

Attribute exploration ([3], cf. also [6,8]) is a knowledge acquisition method based on a question-answer scheme to obtain the implication theory of a domain.

Let us consider a domain (a formal context) (G, M, I) that we do not know completely and that we want to explore and a domain expert for this domain. We start with a (possibly empty) set of known (background) implications \mathcal{L} and a (possibly empty) set $G_E \subseteq G$ of known objects, represented as (possibly empty) formal context $\mathbb{E} = (G_E, M, I_E)$. In every step of the attribute exploration we have a set of already accepted implications \mathcal{L} and a context of already provided counterexamples \mathbb{E} . The attribute exploration algorithm picks the next implication $A \rightarrow B$ that does not follow from \mathcal{L} and that holds in \mathbb{E} . It then asks the expert whether the implication truly holds in the domain. The expert can either confirm that the implication holds or they can refute its validity by providing a counterexample, i.e., an object $g \in G$ whose intent does not respect the implication. If the expert confirms the implication's validity in the domain, it is added to the set \mathcal{L} , otherwise the provided counterexample is added to the context of counterexamples \mathbb{E} . This process is repeated until there is no implication left to be asked.

After performing the attribute exploration we have the (relative) canonical base of implications from which (combined with the background implications) every valid implication in the domain follows. Furthermore, for every implication that is not valid, the set of examples contains a counterexample.

3 Triadic Exploration

In this section we look at *implications* in the triadic setting, in particular, we formally introduce *conditional attribute implications*, and develop a *triadic exploration* for Triadic Concept Analysis as proposed by Ganter and Obiedkov in [5,6].

3.1 Conditional Attribute Implications

In formal contexts (of type (G, M, I)) the matter of *implications* is fairly straightforward: There are *attribute implications* to describe dependencies between attributes (and dually there are *object implications*). In triadic contexts, the notion of implication is not as simple. This manifests in a multitude of types of implications that have been proposed: The earliest suggestion for a *triadic implication* came from Biedermann [1], where he suggested the study of implications of the form $(R \rightarrow S)_C$ which is interpreted as: *If an object has all attributes from R under all conditions from C , then it also has all attributes from S under all conditions from C .*

In [5], Ganter and Obiedkov studied some other types of implications for the triadic setting. They introduced a stronger version of the triadic implication called *conditional attribute implications* to describe dependencies that hold for some conditions. The symmetry arising from the arbitrary choice of objects, attributes and conditions in a triadic context results in five more types of implications. Further, they introduced another generalization of Biedermann’s triadic implication called *attribute \times condition implication* to express dependencies between combinations of attributes and conditions. For the remainder of this paper we will focus on *conditional attribute implications*, because they best serve our goal of developing attribute exploration with multiple experts.

Given a triadic context $\mathbb{T} = (G, M, B, Y)$, a *conditional attribute implication* is an expression of the form $R \xrightarrow{C} S$ where $R, S \subseteq M$, $C \subseteq B$, which reads as: *R implies S under all conditions from C .* A conditional attribute implication $R \xrightarrow{C} S$ holds in a triadic context \mathbb{T} iff for each condition $c \in C$ it holds that if an object $g \in G$ has all the attributes in R it also has all the attributes in S . This is the case if the implication $R \rightarrow S$ holds in every conditional context \mathbb{K}_c for $c \in C$.

Proposition 1. *Let $\mathbb{T} = (G, M, B, Y)$ and \mathbb{K}_c , $c \in B$, be its respective condition contexts. For a conditional implication $R \xrightarrow{C} S$ with $R, S \subseteq M$ and $C \subseteq B$, the following statements are equivalent:*

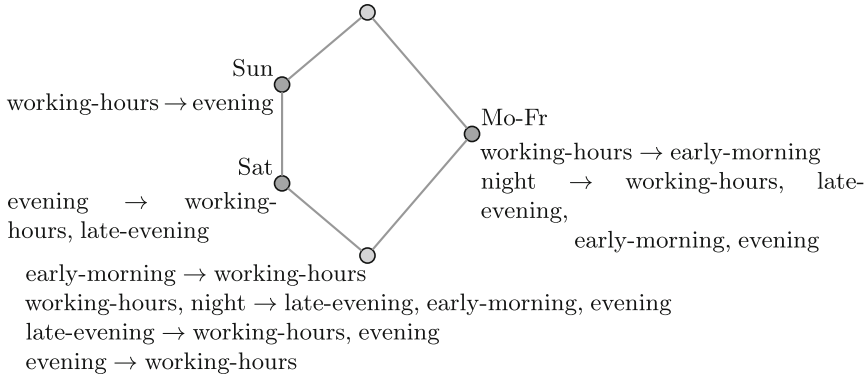


Fig. 3. The lattice of conditional implications of the running example \mathbb{T}_{ex} . with simplified labels, which consist of the relative canonical base with respect to the implications in all nodes below. We omit the top label of implications as the extent of this concept is always Imp_M .

1. $R \xrightarrow{C} S$ holds in \mathbb{T}
2. $R \rightarrow S$ holds in \mathbb{K}_c for every $c \in C$
3. $R \rightarrow S$ holds in $\text{subpos}(\{\mathbb{K}_c | c \in C\})$

Proof. 1. \Leftrightarrow 2. follows directly from the definitions of *holds* in the triadic and dyadic setting. 2. \Leftrightarrow 3. follows from the definition of subposition and that an implication $R \rightarrow S$ holds in a context if and only if for every object g the object intent respects the implication. \square

Example 2. In the context family of Example 1 in Fig. 2 we observe that the implication *early – morning* \rightarrow *working – hours* holds in all three condition contexts $\mathbb{K}_{\text{Mo-Fr}}$, \mathbb{K}_{Sat} and \mathbb{K}_{Sun} , hence, *early – morning* $\xrightarrow{\text{Mo-Fr,Sat,Sun}}$ *working – hours* holds in \mathbb{T}_{ex} . In contrast, the implication *working – hours* \rightarrow *evening* only holds in the condition context \mathbb{K}_{Sun} because tram line 7 is a counterexample in \mathbb{K}_{Sat} and bus line 55 is a counterexample in $\mathbb{K}_{\text{Mo-Fr}}$ and thus *working – hours* $\xrightarrow{\text{Sun}}$ *evening* holds in \mathbb{T}_{ex} , but *working – hours* $\xrightarrow{\text{Mo-Fr,Sat,Sun}}$ *evening* does not.

Clearly, if a conditional implication $R \xrightarrow{C} S$ holds in a triadic context \mathbb{T} then all conditional implications $R \xrightarrow{D} S$ with $D \subseteq C$ hold as well. Further, for every subset $C \subseteq B$ there is a set of conditional implications $R \xrightarrow{C} S$ that hold in \mathbb{T} . This set of conditional implications for a fixed set of conditions C is the implication theory of the subposition of condition contexts $\text{subpos}(\{\mathbb{K}_c | c \in C\})$.

Context of Conditional Implications. A nice way to structure the conditional implications that hold in a triadic context \mathbb{T} is to use the approach

suggested by Ganter and Obiedkov, cf. [5], and to introduce a *context of conditional implications*: Given a triadic context \mathbb{T} , we construct a formal context $\mathcal{C}_{imp}(\mathbb{T}) := (\text{Imp}_M, B, I)$, where the set of all possible implications on M is the object set, the set of conditions B of the triadic context \mathbb{T} is the set of attributes and the incidence relation I is determined by

$$(R \rightarrow S)Ic \iff R \xrightarrow{c} S \text{ holds in } \mathbb{T}.$$

The formal concepts of $\mathcal{C}_{imp}(\mathbb{T})$ are pairs (\mathcal{L}, C) , where \mathcal{L} is a set of implications and C is a set of conditions, such that \mathcal{L} is the set of all implications $R \rightarrow S$ for which $R \xrightarrow{C} S$ holds, and C is the largest set of conditions for which this is the case. These concepts structure the set of conditional implications in a lattice ordered by the conditions for which they hold. Their extents form a system of implication theories.

Example 3. For our running example we present the concept lattice of $\mathcal{C}_{imp}(\mathbb{T}_{ex.})$ with simplified labels in Fig. 3: The extent of the top node always contains the implications that hold under the empty set of conditions, i.e., the whole set Imp_M . We omit this label. For the other nodes we give the relative canonical base with respect to set of implications from all nodes below. Looking at the implications from Example 2, we find the implication *early – morning* \rightarrow *working – hours* at the *bottom node*, because it holds for all three conditions, whereas we find the implication *working – hours* \rightarrow *evening* at the node for *Sunday*, because that is the only condition for which it holds.

3.2 Triadic Exploration

Now, we develop *Triadic Exploration* to explore the conditional implications of a triadic domain.

Previously, we have structured the conditional implications of a triadic domain \mathbb{T} as a system of implication theories by utilizing the context of conditional implications $\mathcal{C}_{imp}(\mathbb{T})$. This was possible because we had complete information about the domains implications in the context \mathbb{T} . However, it is easy to imagine a situation where we can access the information about a domain only indirectly through a domain expert and where an attribute exploration might be useful. For our running example, imagine someone with a bus and train schedule where the information can be looked up but is not fully available at once. Now the question is: How to explore the complete system of conditional implications?

A naive approach is to explore the implication theory for each fixed subset of the conditions, essentially exploring each node of the system of implication theories independently. But, this is clearly not a good idea; it means answering many questions multiple times for each condition.

A better approach might be to only explore the implication theory for every condition, each providing one column in the context of conditional implications \mathcal{C}_{imp} . Then we can compute the concept lattice without any further interactions with the expert.

However, there are some points to consider that suggest a different approach, cf. [6]: First, to stay in the triadic setting, a complete counterexample to a question should describe the new object by the attributes it has *for each of the conditions*, and not only for the one, that is currently under consideration. And second, some implications may hold for several conditions and the domain expert might want to confirm each of them for multiple conditions at once.

Thus, we come back to the context of conditional implications. Ganter and Obiedkov suggested to explore the triadic domain by exploring the nodes in the lattice of conditional implications from the bottom up; using the already known valid implications as background knowledge. Hence, as we explore the system of conditional implications, we successively fill the context of conditional implications.

In the following we describe the nested process of exploring the nodes of the concept lattice of conditional implications with the help of two algorithms: Algorithm 1 for the exploration of the conditional implications for a fixed set of conditions and Algorithm 2 that uses this algorithm as a subroutine to explore all conditional implications of the triadic domain.

Explore Conditional Implications for a Fixed Set of Conditions. For a fixed set of conditions $D \subseteq B$ in a triadic domain $\mathbb{T} = (G, M, B, Y)$, the exploration algorithm is an adapted version of the algorithm for attribute exploration with background implications and exceptions, see [6, 18]. In Algorithm 1 we present an implementation for the exploration in pseudo-code.

The algorithm starts with some background knowledge, in particular: A triadic context $\mathbb{E} = (G_E, M, B, Y_E)$, that contains some examples from the domain \mathbb{T} , and a set of implications \mathcal{L}_0 that are known to hold for all conditions in D ; both of these can be empty. The rest of the domain can only be accessed by the algorithm through interaction with the domain expert. In each step, the algorithm determines the next implication $A \rightarrow A''$ to ask the expert. To determine the next question $A \rightarrow A''$ the algorithm uses both the information from the examples in \mathbb{E} and the known valid implications in \mathcal{L} . It automatically skips questions that follow from the implications in \mathcal{L} or for which \mathbb{E} already contains a counterexample. More precisely, A is the next *relative pseudo-intent* in $\text{subpos}(\{\mathbb{K}_d \mid d \in D\})$, i.e., the lexicographically smallest set A closed under the set of known valid implications and background implications \mathcal{L} that is not already closed in the subposition context of examples for the conditions in D .

Essentially, this algorithm is an attribute exploration with background implications on the subposition of the condition contexts. Additionally, it tracks which implications hold for which conditions in D . This enables us to reduce the amount of interaction required from the expert in subsequent explorations by preventing to ask the same question multiple times for different subposition contexts. The proof of correctness for Algorithm 1 is a straightforward adaption of the proof of [18, Theorem 6] and we therefore omit the details.

Note that we chose to collect all implications that are asked about and the subset of conditions of D for which they hold in Line 13 instead of only adding

Algorithm 1. explore-conditions

Input: a set of conditions $D \subseteq B$, a triadic context $\mathbb{E} = (G_E, M, B, Y_E)$ of examples (possibly empty) and a set \mathcal{L}_0 of background implications known to hold for all conditions in D (also possibly empty)

Interactive Input: (\star) The expert confirms or rejects an implication to hold for the set of conditions D . Upon rejection the expert provides a counterexample g from the domain together with its relation to all conditions and all attributes, i.e., the context $\mathbb{K}_g := (M, B, I)$ where $(m, b) \in I \Leftrightarrow g$ has m under the condition b in the domain.

Output: the relative canonical base $\mathcal{L} \setminus \mathcal{L}_0$ of implications that hold for all conditions in D with respect to \mathcal{L}_0 , a possibly enlarged triadic context of counterexamples \mathbb{E} and the formal context \mathbb{C} of asked implications and the conditions for which they hold.

```

1  $\mathcal{L} := \mathcal{L}_0$ 
2  $A := \emptyset$ 
3  $\mathbb{C} := (\emptyset, B, \emptyset)$ 
4 while  $A \neq M$  do
5   while  $A \neq A''$  in  $S$  where
6      $S := (G_S, M, J) =$  subposition of  $\mathbb{K}_d$  for  $d \in D$  with
7      $K_d := (G_E, M, I_d)$  where  $(g, m) \in I_d \Leftrightarrow (g, m, d) \in Y_E$ 
8   do
9     Ask the expert if  $A \rightarrow A''$  holds for all conditions  $d \in D$   $(\star)$ 
10    if  $A \rightarrow A''$  holds then  $\mathcal{L} := \mathcal{L} \cup \{A \rightarrow A''\}$ 
11    else extend  $\mathbb{E}$  with the counterexample provided by the expert  $(\star)$ 
12
13    extend  $\mathbb{C}$  with the object  $A \rightarrow A''$  and its relation to all conditions  $d \in D$   $(\star)$ 
14  end
15   $A := \text{NextClosure}(A, M, \mathcal{L})$  /* computes the next closure of  $A$  in  $M$  with
16  respect to the implications in  $\mathcal{L}$ ; see for example [6,8] */
17 return  $\mathcal{L} \setminus \mathcal{L}_0, \mathbb{E}$  and  $\mathbb{C}$ 

```

the implications that hold for all conditions in the context \mathbb{C} . Hence, if there is a counterexample, i.e., the implication does not hold for D , we track for which subset of D (if any) the implication does hold. This further reduces the number of questions posed in later explorations. The trade-off is that the background knowledge we have is not just of nodes below the currently explored one in the lattice but may also contain implications that first hold for the conditions of the current node. This has no effect on the implication theory of the node but somewhat complicates the labeling of the node – we cannot simply use the relative canonical base with respect to the knowledge we have. In contrast, if we only added the implications that hold for all conditions in the current exploration then the labels are exactly the implications of the relative canonical base, but, we might have to ask some questions multiple times for some of the conditions. For our running example this approach further reduces the number of questions posed to the expert from fifteen to twelve, cf. Example 5 in Sect. 3.3.

The Order of Explorations. To determine the sequence in which the nodes of the lattice of conditional implications are explored, Ganter and Obiedkov further suggested to follow a linear extension of the lattice of conditional implications, see [5], and later specified this to follow the *NextExtent-Algorithm*, i.e., *NextClosure* on the extents, on the context of conditional attribute implications, see [6].

However, in our setting the *NextExtent-Algorithm* does not fit. The problem is that we may not have the necessary information to correctly determine the next node to explore.² This is because the questions that are asked during the exploration of a node are not guaranteed to discriminate between the conditions that are being explored. Questions that would discriminate between conditions are not asked if there already exists a counterexample for any one of the conditions. This might result in not exploring all nodes of the lattice.

Example 4. Let us illustrate the problem with a small example: Take a look at the domain given by the triadic context \mathbb{T}_1 in Fig. 4. If we explore this context and begin with the bottom node, i.e., the implications that hold for all conditions without any background knowledge, then the first question posed to the expert is $\emptyset \rightarrow ab?$, which the expert refutes with a counterexample – object 1 with all its attributes under all conditions. It substantiates that implication holds for neither of the conditions. The second question that is posed to the expert is $b \rightarrow a?$ which the expert confirms. This concludes the exploration of the bottom node in this example. If we now compute the next extent in the resulting context of conditional implications \mathbb{C} in order to determine which node to explore next, we obtain $\text{NextExtent}(\emptyset) = \{b \rightarrow a\}$ with intent $\{d_1, d_2\}$ which we just explored and then $\text{NextExtent}(\{b \rightarrow a\}) = G_{\mathbb{C}}$ with intent \emptyset which concludes the exploration. However, clearly the implication $\emptyset \rightarrow a$ holds in d_1 but not in d_2 and is missing in \mathbb{C} . The question $\emptyset \rightarrow a?$ was not posed to the expert because there already existed a counterexample for condition d_2 after the first question. Similarly, the implication $a \rightarrow b$ holds in d_2 but not in d_1 and is also missing. In Fig. 4, we present both the lattice of \mathbb{C} and the lattice of $\mathcal{C}_{imp}(\mathbb{T}_1)$. Hence, an exploration that uses the *NextExtent-Algorithm* to determine which nodes of the conditional implications lattice to explore next does not necessarily explore all nodes of the lattice.

To circumvent this problem, we use the suggested strategy of exploring the lattice node by node from the bottom up with the already known valid conditional implications in \mathcal{C}_{imp} as background knowledge. But, instead of using the *NextExtent-Algorithm* to incrementally determine the next combination of conditions to explore, we simply follow a linear extension of $(\mathcal{P}(B) \setminus \emptyset, \supseteq)$. Which means, we walk through all subsets of B sorted by their cardinality from biggest to smallest and stop when we have explored all subsets of cardinality one. At first glance this might look as if we explore more nodes than necessary, because

² For the same reason, the nested application of *NextClosure* for computing all concepts of a triadic context, as described in [14, 15], cannot serve as a base for the triadic exploration.

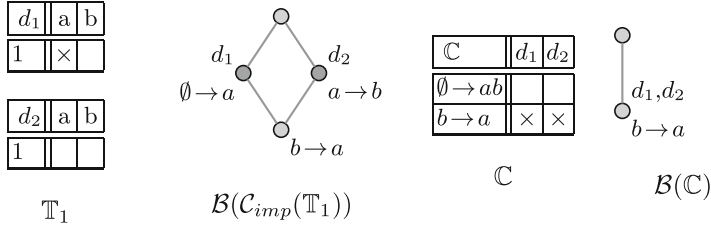


Fig. 4. A triadic context \mathbb{T}_1 , the lattice of conditional implications of \mathbb{T}_1 , the context \mathbb{C} after exploring the conditional implications of \mathbb{T}_1 using the *NextExtent-Algorithm* to determine the next conditions to explore, and the lattice of \mathbb{C}

Algorithm 2. triadic-exploration

Input: a triadic context $\mathbb{E} = (G_E, M, B, Y_E)$ of examples (possibly empty) and a context $\mathbb{C} = (G_C, B, I_C)$ of implications known to hold for some conditions

Output: a triadic context of counterexamples \mathbb{E} and the context of conditional implications \mathbb{C} , from which all valid conditional implications can be inferred

```

1 for  $D$  in linear extension of  $(\mathcal{P}(B) \setminus \emptyset, \supseteq)$  do
2    $\mathcal{L} := D'$  (in  $\mathbb{C}$ )
3    $\mathcal{L}_D, \mathbb{E}_D, \mathbb{C}_D := \text{explore-conditions}(D, \mathbb{E}, \mathcal{L})$ 
4    $\mathbb{E} := \mathbb{E}_D$ 
5    $\mathbb{C} := \mathbb{C} \cup \mathbb{C}_D = (G_C \cup G_{\mathbb{C}_D}, B, I_C \cup I_{\mathbb{C}_D})$ 
6 end
7 return  $\mathbb{E}$  and  $\mathbb{C}$ 

```

the implication theory of a condition might be included in another one and thus is explored at least twice – once in combination and once alone. But, because we only ask questions about implications that are unknown with respect to the knowledge we already have when the condition is explored alone, these questions won't be asked again.

In Algorithm 2 we present the algorithm for *triadic exploration* in pseudocode: We walk through $(\mathcal{P}(B) \setminus \emptyset, \supseteq)$, i.e. the subsets of conditions, in Line 1. For each set of conditions $D \subseteq B$ we determine the implications \mathcal{L} that are known to hold for all conditions in D in Line 2. We compute the canonical base relative to \mathcal{L} in the subposition of condition context of D Line 3. Then, update the known examples E and the known implications in Lines 4 and 5.

3.3 An Example for Triadic Exploration

Example 5. We now give a brief example for a triadic exploration of the domain of our running example (Example 1): Let us assume we only have a triadic expert for this domain and not the whole domain information – imagine someone with access to a search interface for the bus and train schedule of Fig. 2. In Fig. 5 we have listed all interactions with the expert. Each row shows one interaction and the order of interactions is from top to bottom. The resulting lattice of conditional implications is exactly the lattice shown in Fig. 3. The extent of each concept of this lattice is a generating set for the implication theory of implications that hold for all conditions of the intent which follows from Theorem 1, and, because we iteratively computed relative canonical bases. Thus, we know that, for each concept, the implications in its extent are complete, but – as a union of “stacked” relative canonical bases – not necessarily irredundant.

4 Application for Exploration with Multiple Experts

In this section we discuss how to adapt triadic exploration to a setting where we have multiple experts with different views on a domain (i.e., a set of attributes). In Sect. 1, we have briefly discussed the problem of exploration with multiple experts with different views and concluded that combining answers from different experts is not a good strategy for attribute exploration in general. We have further established that all previous methods for multi-expert exploration avoided this problem by assuming that the experts’ knowledge is derived from some consistent domain knowledge.

Conditions	Question – does the implication hold?	Question holds for	Answer
Mo-Fr, Sat, Sun	$\emptyset \rightarrow$ working-hours, late-evening, early-morning, night, evening	\emptyset	RT5
Mo-Fr, Sat, Sun	$\emptyset \rightarrow$ working-hours, late-evening, evening	\emptyset	52
Mo-Fr, Sat, Sun	evening \rightarrow working-hours, late-evening	Sat, Sun	7
Mo-Fr, Sat, Sun	evening \rightarrow working-hours	Mo-Fr, Sat, Sun	true
Mo-Fr, Sat, Sun	night \rightarrow working-hours, late-evening, early-morning, evening	Mo-Fr	N3
Mo-Fr, Sat, Sun	early-morning \rightarrow working-hours	Mo-Fr, Sat, Sun	true
Mo-Fr, Sat, Sun	late-evening \rightarrow working-hours, evening	Mo-Fr, Sat, Sun	true
Mo-Fr, Sat, Sun	working-hours, night \rightarrow late-evening, early-morning, evening	Mo-Fr, Sat, Sun	true
Mo-Fr, Sun	working-hours \rightarrow evening	Sun	55
Mo-Fr, Sat	working-hours, evening \rightarrow early-morning	Mo-Fr	500
Sun	working-hours, late-evening, early-morning, evening \rightarrow night	\emptyset	4
Mo-Fr	working-hours \rightarrow early-morning	Mo-Fr	true

Fig. 5. Triadic Exploration of the running example. Each row represents one interaction with the expert. It comprises the set of conditions that is explored, the question posed in form of an implication, the conditions for which the implication holds, and, the answer given by the expert.

Here, we suggest a different approach that allows for a group of experts with different, opposing views on a domain. The basic idea is to accept all answers equally and look for the subset of knowledge that all experts agree on. To explore

the domain we then explore the agreed-upon knowledge of different subsets of the expert group.

If all experts know about the same objects of the domain we can regard the group of experts as a triadic domain where each experts view is expressed as one condition. In our running example, imagine that there are three experts for the bus and train schedule: One for Monday-Friday, one for Saturday and one for Sunday. The three experts will have different opinions about the implication theory of the time slots.

To explore the dependencies of attributes in this triadic multi-expert domain, we utilize triadic exploration. To ask about a conditional implication then means to ask all experts if the implication holds in their view. However, a simple translation back to the triadic case means that each time an expert gives a counterexample to a question, all experts must be consulted about their view on the counterexample to stay within the triadic setting (because we need the full slice of the triadic context). This is not ideal, but we can adapt the triadic exploration to avoid this issue: Since we do not rely on any specific properties of the triadic context other than being able to form the subposition of the condition contexts \mathbb{K}_d , we can simply leave the triadic setting behind and transfer the idea of conditional attribute implications to a setting where we replace the triadic context with a context family on the same set of attributes (but not necessarily the same set of objects), i.e., a context family $\{\mathbb{K}_e = (G_e, M, I_e) | e \in E\}$ for a group of experts E .

Note that we could also explore the implication theory for each context in such a context family independently and combine the results afterwards, as initially suggested in Sect. 3. It is not obvious how this approach compares to the triadic one. However, the triadic approach also allows to only explore a subset of the system of implication theories.

A real world example for such a context family can be found in the *BSI-IT-Grundschutzkatalog*³, a publication by the German *Federal Office for Information Security*, which contains security recommendations on a wide variety of IT topics. There, a general set of elementary threats is defined and for topics where these threats are present (for example organizational, infrastructure and personnel) a set of measures is defined where each measure combats one or multiple of the threats. Hence, if we regard the elementary threats as attributes, the topics as conditions/experts and the measures as objects, we have a context family on the same set of attributes but with different object sets.

To explore the domain of such a context family, where the set G varies for the different conditions, we have to slightly alter Algorithms 1 and 2. In particular, we need to replace the triadic contexts with context families and the conditions with the experts. The triadic context of counterexamples becomes a context family of counterexamples where each expert has their own context of counterexamples and the objects between them can differ. Hence, in Algorithm 1, A'' is computed on the subposition of the respective contexts of counterexamples and asking about the implication $A \rightarrow A''$ means asking each of the experts.

³ https://www.bsi.bund.de/EN/Topics/ITGrundschutz/itgrundschutz_node.html.

Now, counterexamples from one expert can be accepted without having to ask all other experts about their view on the example.

If we explore a triadic domain in this more abstract setting of context families on the same set of attributes, the trade-off is that we obtain less complete information about the counterexamples. However, we still obtain the same knowledge in terms of conditional attribute implications that hold in the domain.

In addition, we gain the ability to explore context families that do not fit the triadic setting or only do so after some modifications, as for example, the context family of the *BSI-IT-Grundschutzkatalog*. Another example can be derived from the running example: If we look at $\mathbb{K}_{\text{Mo-Fr}}$ in Fig. 2, imagine that instead of one context (and thus one expert) of bus and tram lines we had one for bus lines and one for tram lines. Clearly, this family of two contexts could be transformed into a triadic context, however, to do so we would have to add bus lines to the tram lines context and vice versa – mixing domains that might be perceived as different.

5 Conclusion and Outlook

In this paper, we addressed the problem of multi-expert attribute exploration in Formal Concept Analysis. To this end, we developed *triadic exploration* – an analogue to attribute exploration – for Triadic Concept Analysis, which extends Formal Concept Analysis with the notion of conditions. Triadic exploration helps a triadic domain expert to explore the structure of the conditional attribute implications of the domain.

We adapted triadic exploration to a multi-expert setting by considering the experts' views of a domain as conditions in a triadic setting. We discussed the ramifications of this approach and subsequently suggested to adapt triadic exploration to the more general setting of context families on the same set of attributes.

This paper is a step towards multi-expert exploration where experts can have different views on a domain. In contrast to the few prior works on this subject, here the experts can have opposing views. A next step is the combination of this approach with the notion of partial expert knowledge and a more in depth study of context families as a foundation for multi-expert explorations.

References

1. Biedermann, K.: A foundation of the theory of trilattices. Dissertation, shaker, TU Darmstadt, Aachen (1998)
2. Felde, M., Stumme, G.: Interactive collaborative exploration using incomplete contexts. CoRR abs/1908.08740 (2019). <http://arxiv.org/abs/1908.08740>
3. Ganter, B.: Two basic algorithms in concept analysis. In: Kwuida, L., Sertkaya, B. (eds.) ICFCFA 2010. LNCS (LNAI), vol. 5986, pp. 312–340. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11928-6_22
4. Ganter, B.: Attribute exploration with background knowledge. Theor. Comput. Sci. **217**(2), 215–233 (1999)

5. Ganter, B., Obiedkov, S.: Implications in triadic formal contexts. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS-ConceptStruct 2004. LNCS (LNAI), vol. 3127, pp. 186–195. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27769-9_12
6. Ganter, B., Obiedkov, S.: More expressive variants of exploration. In: Conceptual Exploration, pp. 237–292. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49291-8_6
7. Ganter, B., Wille, R.: Conceptual scaling. In: Roberts, F. (ed.) Applications of Combinatorics and Graph Theory to the Biological and Social Sciences, pp. 139–167. Springer-Verlag (1989)
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag, Berlin/Heidelberg (1999)
9. Guigues, J.L., Duquenne, V.: Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sci. Humaines* **95**, 5–18 (1986)
10. Hanika, T., Zumbrägel, J.: Towards collaborative conceptual exploration. In: Chapman, P., Endres, D., Pernelle, N. (eds.) ICCS 2018. LNCS (LNAI), vol. 10872, pp. 120–134. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91379-7_10
11. Holzer, R.: Methoden der formalen Begriffsanalyse bei der Behandlung unvollständigen Wissens. Dissertation, shaker, TU Darmstadt (2001)
12. Holzer, R.: Knowledge acquisition under incomplete knowledge using methods from formal concept analysis: Part i. *Fundam. Informaticae* **63**(1), 17–39 (2004)
13. Holzer, R.: Knowledge acquisition under incomplete knowledge using methods from formal concept analysis: Part ii. *Fundam. Informaticae* **63**(1), 41–63 (2004)
14. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Discovering shared conceptualizations in folksonomies. *Web Semant.* **6**(1), 38–53 (2008). <https://doi.org/10.1016/j.websem.2007.11.004>
15. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Trias - an algorithm for mining iceberg tri-lattices. In: Proceedings of 6th ICDM conference. Hong Kong, December 2006. <https://doi.org/10.1109/ICDM.2006.162>
16. Kriegel, F.: Parallel attribute exploration. In: Haemmerlé, O., Stapleton, G., Faron Zucker, C. (eds.) ICCS 2016. LNCS (LNAI), vol. 9717, pp. 91–106. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40985-6_8
17. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. In: Ellis, G., Levinson, R., Rich, W., Sowa, J.F. (eds.) ICCS-ConceptStruct 1995. LNCS, vol. 954, pp. 32–43. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60161-9_27
18. Stumme, G.: Attribute exploration with background implications and exceptions. In: Bock, H.H., Polasek, W. (eds.) Data Analysis and Information Systems. Statistical and Conceptual approaches. Proceedings of GfKI 1995. Studies in Classification, Data Analysis, and Knowledge Organization 7, pp. 457–469. Springer, Heidelberg (1996)
19. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel, Dordrecht-Boston (1982)
20. Wille, R.: The basic theorem of triadic concept analysis. *Order* **12**(2), 149–158 (1995). <https://doi.org/10.1007/BF01108624>