Exploring Faulty Data

Daniel Borchmann

Institute of Theoretical Computer Science, TU Dresden
Center for Advancing Electronics Dresden

Abstract. Within formal concept analysis, attribute exploration is a powerful tool to semi-automatically check data for completeness with respect to a given domain. However, the classical formulation of attribute exploration does not take into account possible errors which are present in the initial data. To remedy this, we present in this work a generalization of attribute exploration based on the notion of *confidence*, that will allow for the exploration of implications which are not necessarily valid in the initial data, but instead enjoy a minimal confidence therein.

1 Introduction

Attribute exploration is one of the most important algorithms in the area of formal concept analysis [9], a branch of mathematical order theory with applications in artificial intelligence, machine learning and data mining. The main purpose of this algorithm is to check a given set of *initial data* for completeness, in the sense that this algorithm assists a domain expert in checking whether this initial data completely represents the particular domain the expert is interested in. In doing so, the algorithm presents *implications* to the expert, who has to either validate them or has to provide a counterexample from the domain of discourse. When the algorithm has finished, the initial data has been extended to a *complete* set of examples whose valid implications are exactly all implications valid in the domain.

However, this approach requires the initial data to be free of *errors* in the sense that all the data really stems from the domain. In practical applications, this may not be reasonable to assume, as it may likewise not be reasonable to check the data for correctness. However, the data itself may still be of "high quality" and could thus still be useful, yet only directly applying attribute exploration is not possible anymore.

One way to consider a data set to be of "high quality" is to say that errors occur only "rarely." To handle a scenario like this, an approach is proposed that is based on the notion of *confidence* from data mining [1]. The idea of this approach is not only to explore the implications which are valid in the initial data set, but also to explore those that satisfy a certain lower bound on their confidence. Of course, this will only provide us with a heuristic algorithm, but in a setting like this, where errors can occur randomly, this is the best we can expect to get. Moreover, an *exploration by confidence* has to be thought of as a first step in a completion process, where the resulting set of implications and set of data should be used further on. As an example, the implications obtained from the exploration by confidence could be used as a background knowledge for a classical attribute exploration which starts out with an empty data set.

Of course, this work is not the first to consider implications together with their confidence. The most notable previous work here is from Luxenburger [10], who considered implications together with their confidence and support in formal contexts. However, while he also considered bases of implications with confidence and support, he did not consider an attribute exploration of them.

On the other hand, there has also been previous research on making attribute exploration more suitable for practical applications. Notable works here are *exploration with incomplete knowledge* by Burmeister and Holzer [5], and *attribute exploration with background implications and exceptions* by Stumme [11]. The former extends attribute exploration to the setting of *incomplete data*, i.e., where the data-set in question may have unspecified entries. However, those entries specified must still be correct. The latter work allows *exceptions* in the exploration, by simply removing unwanted corner cases from the domain of discourse. But again, the data that is used for exploration must be free of errors. In this sense, the problem we want to consider in this paper, an exploration of data that may contain errors, is fundamentally different from previously considered extensions of attribute exploration.

The main contribution of this work is an algorithm for exploration by confidence, which shall be discussed in Section 5. This algorithm arises as an instance of a generalized formulation of attribute exploration, which shall be introduced in Section 4. A naive and direct application of this generalized algorithm will yield a first version of exploration by confidence, which however is only "approximative," in a sense that will be discussed in Section 5.1. A slight modification of this approximative version presented in Section 5.2 will then yield the desired algorithm for exploration by confidence.

The results of this work are taken from [4], which not only contains the proofs of the claims in this paper (which we omit due to space restrictions), but also an adaptation of exploration by confidence which also works with *general concept inclusions*, logical objects akin to implications used in the field of *description logics*. We shall give a very brief outlook about this adaptation results in Section 6.

2 Implications and Confidence

We assume the reader has some familiarity with the basic notions of formal concept analysis, as we will not repeat them here. However, we shall repeat some notions and fix some notations about implications that are crucial for the understanding of this paper.

Denote with $\mathrm{Imp}(M)$ the set of all implications on a set M. Recall that an implication $(A \to B) \in \mathrm{Imp}(M)$ is *valid* in a formal context $\mathbb{K} = (G,M,I)$ if and only if

$$A' \subseteq B'.$$

We shall denote with $\mathrm{Th}(\mathbb{K})$ the set of all valid implications on M that are valid in $\mathbb{K}$.

Let $\mathcal{L} \subseteq \mathrm{Imp}(M)$ be a set of implications, and let $(A \to B) \in \mathrm{Imp}(M)$. Recall that the set $\mathcal{L}$ *entails* $(A \to B)$ if and only if in all formal contexts $\mathbb{L}$ with attribute set M, it is true that if all implications from $\mathcal{L}$ are valid in $\mathbb{L}$, then $(A \to B)$ is valid in $\mathbb{L}$ as well. In other words,

$$\mathbb{L} \models \mathcal{L} \Longrightarrow \mathbb{L} \models (A \to B),$$

where we write $\mathbb{L} \models \mathcal{L}$ to mean that all implications in $\mathcal{L}$ are valid in $\mathbb{L}$. If $\mathcal{L}$ entails $(A \to B)$ we shall also write $\mathcal{L} \models (A \to B)$. The subset of $\operatorname{Imp}(M)$ of all implications on M which is entailed by $\mathcal{L}$ is denoted by $\operatorname{Cn}_M(\mathcal{L})$. We shall drop the subscript if the set M is clear from the context.

Entailment between implications can be characterized in a different manner. For this we introduce the notion of *closure operators* induced by sets of implications. More precisely, we define for $A \subseteq M$ the operators

$$\begin{aligned} \mathcal{L}^1(A) &:= A \cup \bigcup \{ Y \mid (X \to Y) \in \mathcal{L}, X \subseteq A \}, \\ \mathcal{L}^{i+1}(A) &:= \mathcal{L}^1(\mathcal{L}^i(A)) \quad (i \in \mathbb{N}_{>0}), \\ \mathcal{L}(A) &:= \bigcup_{i \in \mathbb{N}_{>0}} \mathcal{L}^i(A). \end{aligned}$$

We shall call the mapping $A \mapsto \mathcal{L}(A)$ the *closure operator* induced by $\mathcal{L}$, and we shall call the set A to be *closed under* $\mathcal{L}$ if and only if $A = \mathcal{L}(A)$. The closure operator induced by $\mathcal{L}$ can now be used to characterize entailment of implications as follows:

$$\mathcal{L} \models (A \to B) \Longleftrightarrow B \subseteq \mathcal{L}(A).$$

Let $\mathcal{K} \subseteq \operatorname{Imp}(M)$ be another set of implications. We shall call $\mathcal{L}$ a *base* of $\mathcal{K}$ if and only if $\operatorname{Cn}(\mathcal{L}) = \operatorname{Cn}(\mathcal{K})$. In other words, all implications in $\mathcal{K}$ are entailed by $\mathcal{L}$ and vice versa. If $\mathcal{K} = \operatorname{Th}(\mathbb{K})$, then we shall call $\mathcal{L}$ a *base of* $\mathbb{K}$. Note that a base of $\mathcal{K}$ is always a base of $\operatorname{Cn}(\mathcal{K})$, and vice versa.

Bases allow us to represent sets $\mathcal{K}$ of implications in different ways, without changing their behavior with respect to entailment. This fact is mostly exploited by searching for bases of $\mathcal{K}$ which are of considerably smaller size than $\mathcal{K}$ itself. Those bases are preferably *non-redundant* or even *minimal*. More precisely, if $\mathcal{L}$ is a base of $\mathcal{K}$, then $\mathcal{L}$ is called *non-redundant* if no proper subset of $\mathcal{L}$ is a base of $\mathcal{K}$ as well. Furthermore, $\mathcal{L}$ is called *minimal* if and only if there does not exist another base $\mathcal{L}'$ of $\mathcal{K}$ satisfying $|\mathcal{L}'| < |\mathcal{L}|$.

If we search for bases of $\mathcal{K}$, it might be the case that we do not want to include a certain set $\mathcal{L}_{\text{back}}$ of implications which we already "know." We can think of these implications as given *a-priori*, or as *background knowledge*. If we are given such background knowledge, to find a base of $\mathcal{K}$ it only remains to find a base of all those implications in $\mathcal{K} \setminus \operatorname{Cn}(\mathcal{L}_{\text{back}})$. We thus shall call a set $\mathcal{L} \subseteq \operatorname{Imp}(M)$ a *base of* $\mathcal{K}$ *relative to* $\mathcal{L}_{back}$ (or a *base of* $\mathcal{K}$ *with background knowledge* $\mathcal{L}_{back}$) if and only if $\mathcal{L} \cup \mathcal{L}_{\text{back}}$ is a base of $\mathcal{K}$. The notions of non-redundancy and minimality for relative bases are the same as in the case of bases. Note that if the background knowledge is empty, then relative bases are just bases.

A particular relative base that is known to have minimal cardinality is the *canonical base* $\operatorname{Can}(\mathcal{K}, \mathcal{L}_{\text{back}})$. To define this base, we need to introduce the notion of $\mathcal{L}_{back}$*-pseudo-closed sets of* $\mathcal{K}$ [11]. Let $P \subseteq M$. Then P is called an $\mathcal{L}_{\text{back}}$-pseudo-closed set of $\mathcal{K}$ if and only if the following conditions hold.

i. $P = \mathcal{L}_{\text{back}}(P)$;
ii. $P \neq \mathcal{K}(P)$;
iii. for all $Q \subsetneq P$ which are $\mathcal{L}_{\text{back}}$-pseudo-closed sets of $\mathcal{K}$ it is true that $\mathcal{K}(Q) \subseteq P$.

Then

$$\operatorname{Can}(\mathcal{K},\mathcal{L}_{\text{back}}) := \{ P \to \mathcal{K}(P) \mid P \subseteq M \text{ an } \mathcal{L}_{\text{back}}\text{-pseudo-closed set of } \mathcal{K} \}.$$

It is well-known that $\operatorname{Can}(\mathcal{K},\mathcal{L}_{\text{back}})$ is a base of $\mathcal{K}$ with background-knowledge $\mathcal{L}_{\text{back}}$ of minimal cardinality; see [6, 9] for a proof on this.[1]

Let $\mathbb{K} = (G,M,I)$ be a formal context, and let $(A \to B) \in \operatorname{Imp}(M)$. A *counterexample* (*negative example*) for $(A \to B)$ in $\mathbb{K}$ is an object $g \in A' \setminus B'$. It is obvious that $A \to B$ is valid in $\mathbb{K}$ if and only if $\mathbb{K}$ does not contain counterexample for $A \to B$. Conversely, we call g a *model* (*positive example*) of $A \to B$ if and only if $g \notin A'$ or $g \in B'$.

Related to the notion of counterexamples we define the *confidence* of $A \to B$ in $\mathbb{K}$ as

$$\operatorname{conf}_{\mathbb{K}}(A \to B) := \begin{cases} 1 & \text{if } A' = \varnothing, \\ \frac{|(A \cup B)'|}{|A'|} & \text{otherwise .} \end{cases}$$

In other words, $\operatorname{conf}_{\mathbb{K}}(A \to B)$ is the conditional probability that a randomly chosen object $g \in G$ (in a uniform way), that has all the attributes from A also has all the attributes from B. It is clear that $A \to B$ holds in $\mathbb{K}$ if and only if its confidence in $\mathbb{K}$ is 1.

Let $c \in [0,1]$. We shall denote with $\operatorname{Th}_c(\mathbb{K})$ the set of all implications in $\operatorname{Imp}(M)$ whose confidence is at least c. If c is chosen properly, we may think of $\operatorname{Th}_c(\mathbb{K})$ as the set of implications which are "almost valid" in $\mathbb{K}$; finding a base $\mathcal{L}$ for this set might therefore be desirable. However, the set $\operatorname{Th}_c(\mathbb{K})$ is not closed under entailment, and thus $\mathcal{L} \subseteq \operatorname{Th}_c(\mathbb{K})$ may not necessarily be true. However, a base of $\operatorname{Th}_c(\mathbb{K})$ might be of more use if the element of the base are also "almost valid," i. e., have a confidence in $\mathbb{K}$ which is at least c. We shall therefore call $\mathcal{L}$ a *confident base* of $\operatorname{Th}_c(\mathbb{K})$ (or just $\mathbb{K}$, if c is clear from the context) if and only if $\mathcal{L}$ is a base of $\operatorname{Th}_c(\mathbb{K})$ and $\mathcal{L} \subseteq \operatorname{Th}_c(\mathbb{K})$.

3 Classical Attribute Exploration

It is the purpose of this section to introduce attribute exploration as it is needed in the exposition of this paper. This includes a description of the classical attribute exploration algorithm, which we shall give now. Thereafter, we shall discuss a generalized form of attribute exploration in Section 4, which uses similar ideas as but is different from the one given in [3].

We have already mentioned that attribute exploration is an algorithm which assists experts in completing implicational knowledge about a certain domain of interest. More specifically, let us suppose that we have fixed a finite set M of attributes which are relevant for our considerations. We then can understand the *domain of interest* as a collection $\mathcal{D}$ of objects where each object possesses some attributes from M. In other words, a domain $\mathcal{D}$ on a set M can be viewed as a formal context. Let us

[1] This proof is only for the special case $\mathcal{K} = \operatorname{Th}(\mathbb{K})$, which however is easily generalized to our general case.

furthermore suppose that we are given a set $\mathcal{K}$ of implications from which we definitively know that they are valid in our domain $\mathcal{D}$. Finally, we assume that we have an initial collection of some *examples* from our domain, given again as a formal context.

We are now interested in finding all implications that hold in our domain $\mathcal{D}$, i. e., to find all implications which are not invalidated by objects from the domain $\mathcal{D}$. The difficulty of this problems stems from the fact that enumerating all these objects may be algorithmically infeasible. What we can assume, however, is that we are given an *expert* which is able to provide us with the information whether there *exists*, for a given implication $(A \to B) \in \operatorname{Imp}(M)$, an object in our domain $\mathcal{D}$ which is a counterexample for (i. e., not a model of) $A \to B$, and in that case, also provides such a counterexample.

Abstractly, attribute exploration now proceeds as follows. From all implications in $\operatorname{Cn}(\mathcal{K})$ we already known that they are valid in our domain $\mathcal{D}$. Furthermore, for all implications which are invalidated by objects from $\mathbb{K}$, we known that they are not valid in $\mathcal{D}$. For all other implications we do not know whether they hold in $\mathcal{D}$ or not, i. e., all implications in

$$U(\mathbb{K},\mathcal{K}) := \operatorname{Th}(\mathbb{K}) \setminus \operatorname{Cn}(\mathcal{K})$$

are *undecided* in the sense that they could be valid in $\mathcal{D}$ or not. Then, for the implications in $U(\mathbb{K},\mathcal{K})$ we have to consult the expert. Attribute exploration now does this in a systematic and somehow efficient way, provided that M is finite.

To make this more precise, we shall proceed by describing attribute exploration in a formal way. This description shall be much more formal than usual, to provide the necessary notions we need for our generalized attribute exploration. To this end, we shall first provide some necessary definitions. After that, we give a formal description of the algorithm. Finally, we shall note some well-known properties of attribute exploration.

We shall start by formalizing our initial, subjective notion of a *domain expert*. Intuitively, a domain expert for a domain $\mathcal{D}$ is just a “function” p that, given an implication $A \to B$, returns “true” if $A \to B$ is not invalidated in $\mathcal{D}$, or returns an object from $\mathcal{D}$ which is a counterexample for $A \to B$. We shall take this understanding as the motivation for the following definition. See also [3].

Definition 1. *Let M be a set. A* domain expert *on M is a function*

$$p \colon \operatorname{Imp}(M) \to \{\top\} \cup \mathfrak{P}(M),$$

where $\top \notin \mathfrak{P}(M)$, such that the following conditions hold:

i. If $(X \to Y) \in \operatorname{Imp}(M)$ such that $p(X \to Y) = C \neq \top$, then $C \not\models (X \to Y)$, i. e., $X \subseteq C, Y \nsubseteq C$. ($p$ gives counterexamples for false implications)

ii. If $(A \to B), (X \to Y) \in \operatorname{Imp}(M)$ such that $p(A \to B) = \top, p(X \to Y) = C \neq \top$, then $C \models (A \to B)$. (counterexamples do not invalidate correct implications)

We say that p confirms *an implication $A \to B$ if and only if $p(A \to B) = \top$. Otherwise, we say that p* rejects *$A \to B$ with counterexample $p(A \to B)$. The* theory $\operatorname{Th}(p)$ *of p is the set of all implications on M confirmed by p.*

It is easy to see that every domain gives rise to a domain expert.

Lemma 1. *Let $\mathcal{D}$ be a domain (formal context) on a set M. For each $(A \to B) \in \operatorname{Imp}(M)$ for which there exists a counterexample in $\mathcal{D}$, let $C_{A\to B}$ such a counterexample. Then the mapping*

$$p_{\mathcal{D}}(X \to Y) := \begin{cases} C_{X\to Y} & \textit{if } C_{X\to Y} \textit{ exists} \\ \top & \textit{otherwise} \end{cases}$$

is a domain expert on M.

Note that the definition of $p_{\mathcal{D}}$ depends on the particular choice of the counterexamples, therefore $\mathcal{D}$ may give rise to more than one domain expert.

Let p be a domain expert on a set M, and define

$$\mathcal{D}_p := (\{p(A \to B) \mid (A \to B) \in \operatorname{Imp}(M)\} \setminus \{\top\}, M, \ni).$$

Then clearly $\mathcal{D}_p$ is a domain, and it is easy to see that each domain expert p on M can be obtained as a domain expert of the form $p_{\mathcal{D}_p}$, and that for each domain $\mathcal{D}$ on M it is true that $\mathcal{D} = \mathcal{D}_{p_{\mathcal{D}}}$.

The crucial observation is now that domain experts can answer the question of *validity* in the domains they represent.

Lemma 2. *Let M be a set and let p be a domain expert on M. Then for each $(A \to B) \in \operatorname{Imp}(M)$ it is true that*

$$(A \to B) \textit{ is valid in } \mathcal{D}_p \iff p(A \to B) = \top.$$

We have now formally captured the notion of an expert, and we are ready to describe the algorithm of attribute exploration in a formal way, as presented in Algorithm 1. In this exposition, we assume that the set M is equipped with a strict linear order, which then gives rise to a lectic order as it is needed for applying the Next-Closure algorithm [8]. Furthermore, for better readability, we denote a formal context that arises from another formal context $\mathbb{K}$ by adding a new object with attributes from C by $\mathbb{K} + C$.

Note that the computation of the set P_{i+1} from P_i, $\mathcal{K}_i$ and $\mathbb{K}_i$ and $\mathcal{L}_i$ can be done using the Next-Closure algorithm. As the details are not relevant for our further discussion, we refer the interested reader to the literature [4].

The following results are well known properties of Algorithm 1, and corresponding proofs can be found in [6, 7, 9, 11].

Theorem 1. *Let p, $\mathcal{K}$ and $\mathbb{K}$ be valid input for Algorithm 1. Then Algorithm 1 terminates with input p, $\mathcal{K}$ and $\mathbb{K}$. If $\mathcal{K}'$ and $\mathbb{K}'$ are the corresponding values returned by the algorithm, then the following statements are true:*

i. $\mathcal{K} \subseteq \mathcal{K}' \subseteq \operatorname{Th}(\mathbb{K}') \subseteq \operatorname{Th}(\mathbb{K})$.
ii. $\operatorname{Th}(p) = \operatorname{Th}(\mathbb{K}') = \operatorname{Cn}(\mathcal{K}')$.
iii. *The cardinality of $\mathcal{K}' \setminus \mathcal{K}$ is the smallest possible with respect to $\operatorname{Th}(p) = \operatorname{Cn}(\mathcal{K}')$. More specifically, $\mathcal{K}' \setminus \mathcal{K} = \operatorname{Can}(\mathbb{K}', \mathcal{K})$.*

Algorithm 1.

```
Input  A domain expert $p$ on a finite set $M$, a set $\mathcal{K} \subseteq \mathrm{Imp}(M)$ and a formal context
    $\mathbb{K}$ with attribute set $M$ such that $\mathcal{K} \subseteq \mathrm{Th}(p) \subseteq \mathrm{Th}(\mathbb{K})$.
Procedure
    i.   Initialize $i := 0, P_i := \mathcal{K}(\emptyset), \mathcal{K}_i := \mathcal{K}, \mathbb{K}_i := \mathbb{K}$.
    ii.  Let $P_{i+1}$ be the smallest $\mathcal{K}_i$-closed set lectically larger or equal to $P_i$, which is
         not an intent of $\mathbb{K}_i$. If no such set exists, terminate.
    iii. If $p$ confirms $P \to P''$, then
           – $\mathcal{K}_{i+1} := \mathcal{K}_i \cup \{P \to P''\}$,
           – $\mathbb{K}_{i+1} := \mathbb{K}_i$.
    iv.  If $p$ provides a counterexample $C$ for $P \to P''$, then
           – $\mathcal{K}_{i+1} := \mathcal{K}_i$,
           – $\mathbb{K}_{i+1} := \mathbb{K}_i + C$.
    v.   Set $i := i+1$ and go to ii.
Output  Return $\mathcal{K}_i$ and $\mathbb{K}_i$.
```

4 Exploring Sets of Implications

We are given a precise formulation of attribute exploration in the previous section. However, this formulation is not applicable to our setting of exploring implications with a certain minimal confidence. To address this issue, we shall develop in this section a more general formulation of attribute exploration which goes beyond the classical one.

In the classical case, we are given a formal context $\mathbb{K}$ and a set of implications $\mathcal{K} \subseteq \mathrm{Th}(\mathbb{K})$, as well as a domain expert p, who confirms all implications in $\mathcal{K}$ and where all implications confirmed by p are contained in $\mathrm{Th}(\mathbb{K})$. The task attribute exploration then solves is to provide a method to guide the expert p through all implications in $\mathrm{Th}(\mathbb{K}) \setminus \mathrm{Cn}(\mathcal{K})$ for deciding whether these implications are valid in the domain or not. At the end, attribute exploration both provides a a relative base of all valid implications of the domain p represents, and a set of objects from the domain such that an implication is valid in the domain if and only if all these objects are models of this implication. This set of objects forms itself a domain, and it can be thought of as a sufficient excerpt of the domain represented by p.

We want to try to lift this description of attribute exploration to the case of exploration by confidence. There, our setting is a bit more involved. As in the case of classical attribute exploration, we are given a domain expert p, a formal context $\mathbb{K}$ and a set of implications $\mathcal{K}$. Additionally, we are given a $c \in [0,1]$, the *confidence threshold* for our exploration. Then, in contrast to the classical setting, exploration by confidence considers not only the implications $\mathrm{Th}(\mathbb{K}) \setminus \mathrm{Cn}(\mathcal{K})$, but also those in $\mathrm{Th}_c(\mathbb{K}) \setminus \mathrm{Cn}(\mathcal{K})$. We assume that $\mathcal{K}$ is a set of implications with confidence at least c and that all implications in $\mathcal{K}$ are confirmed by p; in other words, $\mathcal{K} \subseteq \mathrm{Th}(p)$ and $\mathcal{K} \subseteq \mathrm{Th}_c(\mathbb{K})$. While the first condition is rather clear, the second is not strictly necessary, but adopted for simplicity.

An attribute exploration algorithm which then works in this setting should guide the expert through the implications in $\mathrm{Th}_c(\mathbb{K}) \setminus \mathrm{Cn}(\mathcal{K})$, asking whether some implications are correct or not. The counterexamples provided by the expert are then used to falsify certain implications in $\mathrm{Th}_c(\mathbb{K})$. They are not used, however, for computing the confidence; this is solely done in the initial context $\mathbb{K}$, because we want to find a base of $\mathrm{Th}_c(\mathbb{K})$. At the end, the attribute exploration algorithm should both compute a set $\mathcal{L}$ of implications and a formal context $\mathbb{L}$ such that each implication in $\mathrm{Th}_c(\mathbb{K})$ is either not valid in $\mathbb{L}$ or follows from $\mathcal{L}\cup\mathcal{K}$.

What we now want to describe is a more general formulation of attribute exploration that is applicable to our setting of exploration by confidence. For this, we shall develop in the remainder of this section a general formulation of attribute exploration that works with a set of *certain implications* and a set of *interesting implications* and provides a method to guide an expert through the set of *undecided implications*, until no more are left. The properties this algorithm should have should be the same as in the classical case, as far as this is possible. Then later on, we shall apply this algorithm to our setting of exploration of confidence.

To this end, let us recapitulate our setting for the exploration algorithm, this time a bit more general: we are given a finite set M, a domain expert p on M, and two sets $\mathcal{K}$,$\mathcal{L}$ of implications. In our classical case, $\mathcal{L}=\mathrm{Th}(\mathbb{K})$ for some formal context $\mathbb{K}$; in our setting of exploration by confidence, we would have $\mathcal{L}=\mathrm{Th}_c(\mathbb{K})$, again for some formal context $\mathbb{K}$ and some $c\in[0,1]$. We assume that $\mathcal{K}\subseteq\mathrm{Th}(p)$ and $\mathcal{K}\subseteq\mathcal{L}$. We then consider the set $\mathcal{K}$ as the (initial) set of *certain implications.* During our exploration we only consider implications in $\mathcal{L}$, wherefore we shall call this set the set of *interesting implications.* Finally, for each implication in $\mathcal{L}\setminus\mathrm{Cn}(\mathcal{K})$ it is not clear yet whether p confirms it or not. Therefore, we call this set the (current) set of *undecided implications.*

An exploration for this abstract setting now should compute a relative base of $\mathcal{L}\cap\mathrm{Th}(p)$ with background knowledge $\mathcal{K}$ by interacting with the expert p. At best, this interaction is kept at a minimum (i. e., the number of times the expert is invoked is as small as possible), as expert interaction is assumed to be expensive.

Considering the classical attribute exploration algorithm, it is not very difficult to come up with a reformulation which is reasonably applicable to this general setting. To this end, let us fix a finite set M and a lectic order $\leq$ on $\mathfrak{P}(M)$. Then such a reformulation is given in Algorithm 2.

The problem this algorithm has is that it does not ensure that the implications asked to the expert are elements of $\mathcal{L}_i$, the current set of all interesting implications. Because of this, we cannot expect this algorithm to actually compute a relative base of $\mathcal{L}\cap\mathrm{Th}(p)$. However, what this algorithm achieves is to compute an "approximation" of a relative base of $\mathcal{L}\cap\mathrm{Th}(p)$ in the sense that if n is the index of the last iteration of the algorithm, then $\mathcal{K}_n$ is such that

$$\mathrm{Th}(p)\cap\mathrm{Cn}(\mathcal{L})\supseteq\mathrm{Cn}(\mathcal{K}_n)\supseteq\mathrm{Cn}(\mathrm{Th}(p)\cap\mathcal{L}).$$

So what this algorithm does is not computing a relative base $\mathcal{K}_n$ of $\mathrm{Cn}(\mathrm{Th}(p)\cap\mathcal{L})$, but a complete superset of it. However, this set $\mathcal{K}_n$ is not too far away from being sound for $\mathrm{Cn}(\mathrm{Th}(p)\cap\mathcal{L})$, as $\mathrm{Cn}(\mathcal{K}_n)\subseteq\mathrm{Th}(p)\cap\mathrm{Cn}(\mathcal{L})$. On the other hand, the set $\mathcal{K}_n$ is as small as possible for being sound and complete for itself.

We shall not prove the following result due to space restrictions, but instead refer the interested reader to [4].

Algorithm 2 (General Attribute Exploration).

```
Input  A domain expert $p$ on a finite set $M$ and sets $\mathcal{K},\mathcal{L}\subseteq\mathrm{Imp}(M)$ such that $\mathcal{K}\subseteq\mathrm{Th}(p)$
    and $\mathcal{K}\subseteq\mathcal{L}$.
Procedure
    i.   Initialize $i:=0, P_i:=\mathcal{K}(\varnothing), \mathcal{K}_i:=\mathcal{K}, \mathcal{L}_i:=\mathcal{L}, \mathbb{L}_i:=(\varnothing,M,\varnothing)$.
    ii.  Let $P_{i+1}$ be the smallest $\mathcal{K}_i$-closed set lectically larger or equal to $P_i$, which is
         not $\mathcal{L}_i$-closed. If no such set exists, terminate.
    iii. If $p$ confirms $P_{i+1}\to\mathcal{L}_i(P_{i+1})$, then
           – $\mathcal{K}_{i+1}:=\mathcal{K}_i\cup\{P_{i+1}\to\mathcal{L}_i(P_{i+1})\}$,
           – $\mathcal{L}_{i+1}:=\mathcal{L}_i$,
           – $\mathbb{L}_{i+1}:=\mathbb{L}_i$.
    iv.  If $p$ provides a counterexample $C$ for $P_{i+1}\to\mathcal{L}_i(P_{i+1})$, then
           – $\mathcal{K}_{i+1}:=\mathcal{K}_i$,
           – $\mathcal{L}_{i+1}:=\{(A\to B)\in\mathcal{L}_i \mid C\models(A\to B)\}$,
           – $\mathbb{L}_{i+1}:=\mathbb{L}_i+C$.
    v.   Set $i:=i+1$ and go to ii.
Output  Return $\mathcal{K}_i$ and $\mathbb{L}_i$.
```

Theorem 2. *Let $p,\mathcal{K},\mathcal{L}$ be valid input for Algorithm 2. Then the algorithm with this input terminates after finitely many steps. Let n be the last iteration of the algorithm. Then*

1. $\mathrm{Th}(p)\cap\mathcal{L}\supseteq\mathrm{Cn}(\mathcal{K}_n)\supseteq\mathrm{Cn}(\mathrm{Th}(p)\cap\mathcal{L})$,
2. *for each* $(A\to B)\in\mathcal{L}$, *either* $(A\to B)\in\mathrm{Cn}(\mathcal{K}_n)$ *or* $(A\to B)\notin\mathrm{Th}(\mathbb{L}_n)$,
3. $\mathcal{K}_n\setminus\mathcal{K}=\mathrm{Can}(\mathcal{K}_n,\mathcal{K})$.

Since we do not have any control about whether the implications asked by Algorithm 2 are in the set $\mathcal{L}$ of interesting implications, we cannot expect that instantiating this algorithm with $\mathcal{L}=\mathrm{Th}_c(\mathbb{K})$ will indeed yield an algorithm for exploration by confidence. We shall therefore discuss another, even further generalized version of attribute exploration, which will allow for more freedom in which implications are asked to the expert. This generalization arises from Algorithm 2 by observing that instead of asking implications of the form $P_{i+1}\to\mathcal{L}_i(P_{i+1})$, it would be sufficient for the correctness of the algorithm to just ask implications of the form $P_{i+1}\to Q_{i+1}$, where Q_{i+1} is such that $P_{i+1}\subsetneq Q_{i+1}\subseteq\mathcal{L}_i(P_{i+1})$, $Q_{i+1}\nsubseteq\mathcal{K}_i(P_{i+1})$.

Applying this idea to Algorithm 2 yields Algorithm 3. The latter algorithm retains all properties of the former, except for the fact that it does not necessarily compute a minimal base anymore.

Theorem 3. *Let $p,\mathcal{K},\mathcal{L}$ be valid input for Algorithm 3. Then the algorithm applied to this input terminates after finitely many steps. Let n be the last iteration of the algorithm. Then*

1. $\mathrm{Th}(p)\cap\mathcal{L}\supseteq\mathrm{Cn}(\mathcal{K}_n)\supseteq\mathrm{Cn}(\mathrm{Th}(p)\cap\mathcal{L})$,
2. *for each* $(A\to B)\in\mathcal{L}$, *either* $(A\to B)\in\mathrm{Cn}(\mathcal{K}_n)$ *or* $(A\to B)\notin\mathrm{Th}(\mathbb{L}_n)$.

However, as we shall see in Section 5.2, we can use Algorithm 3 to devise an algorithm for exploration by confidence, by choosing the sets Q_{i+1} appropriately.

Algorithm 3 (General Attribute Exploration, Weaker Version).

```
Input  A domain expert $p$ on a finite set $M$ and sets $\mathcal{K},\mathcal{L}\subseteq\mathrm{Imp}(M)$ such that $\mathcal{K}\subseteq\mathrm{Th}(p)$
    and $\mathcal{K}\subseteq\mathcal{L}$.
Procedure
    i.   Initialize $i:=0,P_i:=\mathcal{K}(\varnothing),\mathcal{K}_i:=\mathcal{K},\mathcal{L}_i:=\mathcal{L},\mathbb{L}_i:=(\varnothing,M,\varnothing)$.
    ii.  Let $P_{i+1}$ be the smallest $\mathcal{K}_i$-closed set lectically larger or equal to $P_i$, which is
         not $\mathcal{L}_i$-closed. If no such set exists, terminate.
    iii. Choose $Q_{i+1}\subseteq M$ such that $P_{i+1}\subsetneq Q_{i+1}\subseteq\mathcal{L}(P_{i+1})$, $Q_{i+1}\nsubseteq\mathcal{K}_i(P_{i+1})$.
    iv.  If $p$ confirms $P_{i+1}\to Q_{i+1}$, then
           – $\mathcal{K}_{i+1}:=\mathcal{K}_i\cup\{P_{i+1}\to Q_{i+1}\}$,
           – $\mathcal{L}_{i+1}:=\mathcal{L}_i$,
           – $\mathbb{L}_{i+1}:=\mathbb{L}_i$.
    v.   If $p$ provides a counterexample $C$ for $P_{i+1}\to Q_{i+1}$, then
           – $\mathcal{K}_{i+1}:=\mathcal{K}_i$,
           – $\mathcal{L}_{i+1}:=\{(A\to B)\in\mathcal{L}_i\mid C\models(A\to B)\}$,
           – $\mathbb{L}_{i+1}:=\mathbb{L}_i+C$.
    vi.  Set $i:=i+1$ and go to ii.
Output  Return $\mathcal{K}_i$ and $\mathbb{L}_i$.
```

5 Exploration by Confidence

Based on the generalizations we have discussed in the previous section, we shall now turn our attention to our original question, namely to devise an algorithm for exploration by confidence. Recall that for this we are given a finite set M, a formal context $\mathbb{K}$ with attribute set M, an expert p on M, some background knowledge $\mathcal{K}\subseteq\mathrm{Th}(p)$, and some number $c\in[0,1]$. What an algorithm for exploration by confidence now should achieve is to compute a base of $\mathrm{Th}(p)\cap\mathrm{Th}_c(\mathbb{K})$ with background knowledge $\mathcal{K}$. Ideally, for this it should invoke the expert p as few times as possible.

We shall start this section by presenting a first algorithm that is not precisely an algorithm for exploration by confidence, but instead is an *approximative* algorithm in the sense as discussed in the previous section. This first algorithm will be obtained by instantiating the generalized attribute exploration algorithm from Section 4. We shall do this in Section 5.1. A proper algorithm for exploration by confidence will then be discussed in Section 5.2, where we shall instantiate the weaker generalization of attribute exploration from Section 4.

5.1 An Approximative Exploration by Confidence

Our first idea is as simple as straightforward: we use Algorithm 2 and instantiate it with our setting of exploration by confidence, i. e., we set $\mathcal{L}=\mathrm{Th}_c(\mathbb{K})$. The resulting algorithm is shown as Algorithm 4. The properties of Algorithm 2, as given in Theorem 2, immediately yield the following result.

Corollary 1. *Let $\mathbb{K}=(G,M,I)$ be a finite and non-empty formal context, $c\in[0,1]$, p be a domain expert on M and $\mathcal{K}\subseteq\mathrm{Th}_c(\mathbb{K})\cap\mathrm{Th}(p)$. Then Algorithm 4 terminates with input p, c and $\mathcal{K}$. Let n be the last iteration of this run of the algorithm. Then*

Algorithm 4 (Approximative Exploration by Confidence).

```
Input  A domain expert $p$ on a finite set $M$, a formal context $\mathbb{K}$, $c \in [0,1]$ and a set
    $\mathcal{K} \subseteq \mathrm{Th}_c(\mathbb{K})$ such that $\mathcal{K} \subseteq \mathrm{Th}(p)$.
Procedure
    i.   Initialize $i := 0, P_i := \mathcal{K}(\varnothing), \mathcal{K}_i := \mathcal{K}, \mathcal{L}_i := \mathrm{Th}_c(\mathbb{K}), \mathbb{L}_i := (\varnothing, M, \varnothing)$.
    ii.  Let $P_{i+1}$ be the smallest $\mathcal{K}_i$-closed set lectically larger or equal to $P_i$, which is
         not $\mathcal{L}_i$-closed. If no such set exists, terminate.
    iii. If $p$ confirms $P_{i+1} \to \mathcal{L}_i(P_{i+1})$, then
           – $\mathcal{K}_{i+1} := \mathcal{K}_i \cup \{P_{i+1} \to \mathcal{L}_i(P_{i+1})\}$,
           – $\mathcal{L}_{i+1} := \mathcal{L}_i$,
           – $\mathbb{L}_{i+1} := \mathbb{L}_i$.
    iv.  If $p$ provides a counterexample $C$ for $P_{i+1} \to \mathcal{L}_i(P_{i+1})$, then
           – $\mathcal{K}_{i+1} := \mathcal{K}_i$,
           – $\mathcal{L}_{i+1} := \{(A \to B) \in \mathcal{L}_i \mid C \models (A \to B)\}$,
           – $\mathbb{L}_{i+1} := \mathbb{L}_i + C$.
    v.   Set $i := i+1$ and go to ii.
Output  Return $\mathcal{K}_i$ and $\mathbb{L}_i$.
```

i. $\mathrm{Th}(p) \cap \mathrm{Cn}(\mathrm{Th}_c(\mathbb{K})) \supseteq \mathrm{Cn}(\mathcal{K}_n) \supseteq \mathrm{Cn}(\mathrm{Th}(p) \cap \mathrm{Th}_c(\mathbb{K}))$,
ii. $\mathrm{Can}(\mathcal{K}_n, \mathcal{K}) = \mathcal{K}_n \setminus \mathcal{K}$.

Evidently, Algorithm 4 does not guarantee that the implications asked are actually elements of $\mathcal{L} = \mathrm{Th}_c(\mathbb{K})$, i. e., those implications do not need to have a confidence of at least c in $\mathbb{K}$. This may or may not be an issue, depending on the application one is currently dealing with.

What is also important for Algorithm 4 to be practical is to be able to compute closures under $\mathcal{L}_i = \mathrm{Th}_c(\mathbb{K}) \cap \mathrm{Th}(\mathbb{L}_i)$. However, it is by far obvious how to compute closures under these sets of implications. Of course, one does not want to compute these sets explicitly, and indeed it is true that

$$\mathcal{L}_i(A) = A''_{\mathbb{L}_i} \cap \mathrm{Th}_c(\mathbb{K})(A),$$

for each $A \subseteq M$, where $A''_{\mathbb{L}_i}$ denotes double derivation in $\mathbb{L}_i$. Thus, to make Algorithm 4 practicably applicable, one only needs a way to compute closures of sets A under $\mathrm{Th}_c(\mathbb{K})$.

While it is possible to compute these closures effectively without computing the set $\mathrm{Th}_c(\mathbb{K})$ explicitly [4], the computational overhead might be unwelcomed. One may be tempted to think that we can eliminate the problem of computing closures under $\mathrm{Th}_c(\mathbb{K})$ by using the following approach: instead of asking implications of the form

$$P_{i+1} \to \mathcal{L}_i(P_{i+1}), \tag{1}$$

where $\mathcal{L}_i(P_{i+1}) = \mathrm{Th}_c(\mathbb{K})(P_{i+1}) \cap (P_{i+1})''_{\mathbb{L}_i}$, in Algorithm 4 we could just as well ask implications of the form

$$P_{i+1} \to \{m \in M \mid \mathrm{conf}_{\mathbb{K}}(P_{i+1} \to \{m\}) \geqslant c\}. \tag{2}$$

$\mathbb{K}$	a	b	c
1	×	×	
2	×	×	
3	×	×	
4	×	×	×
5	×	×	×
6	×		×
7	×		×
8			
9			
10			

Fig. 1. Context which shows that a simple approach to exploration by confidence does not work

This would have the evident advantage that the right-hand side of the implication is easy to compute. However, it turns out that with this modification the algorithm is not correct anymore, in the sense that the set of implications accepted by the expert is not complete for $\mathrm{Th}_c(\mathbb{K})$.

Example 1 (Example 6.2.2 from [4]).

Consider the formal context $\mathbb{K}$ as given in Figure 1, let $\mathcal{K} = \{\{\mathsf{a}\} \to \{\mathsf{b}\}\}$, and choose $c = \frac{1}{2}$. Suppose that we apply exploration by confidence in the simplified version as described before, i. e., we ask implications of the form of (2) instead of those in (1). Then since all sets P_i are closed under $\mathcal{K}$, the implication $\{\mathsf{a}\} \to \{\mathsf{c}\}$ is never asked to the expert, because $\{\mathsf{a}\}$ is not closed under $\mathcal{K}$. On the other hand,

$$\mathrm{conf}_{\mathbb{K}}(\{\mathsf{a}\} \to \{\mathsf{c}\}) = \frac{4}{7} > \frac{1}{2},$$

i. e. $(\{\mathsf{a}\} \to \{\mathsf{c}\}) \in \mathrm{Th}_c(\mathbb{K})$, and is thus an interesting implication. Furthermore, the implication $\{\mathsf{a}\} \to \{\mathsf{c}\}$ also does not follow from other implications asked to the expert, as the implications $\{\mathsf{b}\} \to \{\mathsf{c}\}$, $\{\mathsf{a},\mathsf{b}\} \to \{\mathsf{c}\}$, and $\varnothing \to \{\mathsf{c}\}$ will also not be asked to the expert, because

$$\begin{aligned}\mathrm{conf}_{\mathbb{K}}(\{\mathsf{b}\} \to \{\mathsf{c}\}) &= \frac{2}{5} < \frac{1}{2}\\ \mathrm{conf}_{\mathbb{K}}(\{\mathsf{a},\mathsf{b}\} \to \{\mathsf{c}\}) &= \frac{2}{5} < \frac{1}{2}\\ \mathrm{conf}_{\mathbb{K}}(\varnothing \to \{\mathsf{c}\}) &= \frac{4}{10} < \frac{1}{2}\end{aligned}$$

Thus, if we assume that the expert p confirms all proposed implications, and if we denote the set of confirmed implications by $\mathcal{K}_n$, then

$$\mathcal{K}_n(\{\mathsf{a}\}) = \{\mathsf{a},\mathsf{b}\}.$$

But $\mathrm{Th}_c(\mathbb{K}) \cap \mathrm{Th}(p) = \mathrm{Th}_c(\mathbb{K})$, and

$$\mathrm{Th}_c(\mathbb{K})(\{\mathsf{a}\}) = \{\mathsf{a},\mathsf{b},\mathsf{c}\}.$$

Thus, the set $\mathcal{K}_n$ is not complete for $\mathrm{Th}_c(\mathbb{K}) \cap \mathrm{Th}(p)$.

5.2 An Exact Exploration by Confidence

The previous example shows that our simple idea of avoiding the computational overhead of computing closures under $\mathrm{Th}_c(\mathbb{K})$ did not work. In this section we shall show how we can make this idea work nonetheless, by further suitably modifying the algorithm. For this we shall use the weaker generalization of Algorithm 3. As a pleasant side-effect, by this we will obtain a proper algorithm for exploration by confidence, i. e., the new algorithm will indeed compute a base of $\mathrm{Th}(p) \cap \mathrm{Th}_c(\mathbb{K})$. On the downside, since this algorithm is based on the weaker generalization of attribute exploration, we cannot expect it to compute a base of minimal cardinality.

The main idea for this adaption is as follows: the weaker generalization of Algorithm 3 instantiated for our setting of exploration by confidence does not require us to compute closures under $\mathrm{Th}_c(\mathbb{K})$. Instead, all we need to check is whether a given set of attributes is closed under $\mathrm{Th}_c(\mathbb{K})$. The main problem with the latter is that in general we need to consider all subsets of $B \subseteq A$ and all elements $m \in M \setminus A$ checking whether they satisfy

$$\mathrm{conf}_{\mathbb{K}}(B \to \{m\}) \geqslant c.$$

This is because

$$A = \mathrm{Th}_c(\mathbb{K})(A) \iff (\forall B \subseteq A \forall m \in M\colon \mathrm{conf}_{\mathbb{K}}(B \to \{m\}) \geqslant c \implies m \in A).$$

On the other hand, if A would have the property that for each $m \in M$ and every $B \subsetneq A$ with $\mathrm{conf}_{\mathbb{K}}(B \to \{m\}) \geqslant c$ it is true that $m \in A$, then checking whether A is closed under $\mathrm{Th}_c(\mathbb{K})$ would be easy, as in this case

$$A = \mathrm{Th}_c(\mathbb{K})(A) \iff (\forall m \in M\colon \mathrm{conf}_{\mathbb{K}}(A \to \{m\}) \geqslant c \implies m \in A).$$

Let's make this more precise. In what follows, we shall write the context subposition of two contexts $\mathbb{K}_1 = (G_1, M, I_1), \mathbb{K}_2 = (G_2, M, I_2)$ as $\mathbb{K}_1 \div \mathbb{K}_2$, i. e.,

$$\mathbb{K}_1 \div \mathbb{K}_2 := (G_1 \cup G_2, M, I_1 \cup I_2).$$

Here we assume that G_1 and G_2 are disjoint. In the following proposition, we have $\mathbb{K}_1 = \mathbb{K}$ and $\mathbb{K}_2 = \mathbb{L}_i$, and we can think of the former as the initial formal context of our exploration process, while the latter contains all counterexamples collected up to iteration i. Then $\mathbb{K} \div \mathbb{L}_i$ is the currently known context of iteration i.

Proposition 1 (Proposition 6.2.5 from [4]). *Let $\mathbb{K} = (G, M, I)$ be a finite formal context, and let $c \in [0,1]$. Let $\mathbb{L}_i = (G_i, M, I)$ be another finite formal context such that G_i and G are disjoint, and define $\mathcal{L}_i = \mathrm{Th}_c(\mathbb{K}) \cap \mathrm{Th}(\mathbb{L}_i)$. Let $A \subseteq M$ be such that for every intent $X \subsetneq A$ of $\mathbb{K} \div \mathbb{L}_i$ it is true that*

$$\forall m \in X''_{\mathbb{L}_i}\colon \mathrm{conf}_{\mathbb{K}}(X \to \{m\}) \geqslant c \implies m \in \mathcal{K}_i(X). \tag{3}$$

In addition, let A be $\mathcal{K}_i$-closed. Then it is true that A is $\mathcal{L}_i$-closed if and only if

$$A = A''_{\mathbb{K} \div \mathbb{L}_i} \text{ and } \forall m \in A''_{\mathbb{L}_i} \setminus A\colon \mathrm{conf}_{\mathbb{K}}(A \to \{m\}) < c. \tag{4}$$

Algorithm 5 (Exploration by Confidence).

```
Input  A domain expert $p$ on a finite set $M$, a formal context $\mathbb{K}$, $c \in [0,1]$ and a set
    $\mathcal{K} \subseteq \mathrm{Th}_c(\mathbb{K})$ such that $\mathcal{K} \subseteq \mathrm{Th}(p)$.
Procedure
    i.   Initialize $i := 0, P_i := \mathcal{K}(\varnothing), \mathcal{K}_i := \mathcal{K}, \mathcal{L}_i := \mathrm{Th}_c(\mathbb{K}), \mathbb{L}_i := (\varnothing, M, \varnothing)$.
    ii.  Let $P_{i+1} := \min_{\leq}(P^1_{i+1}, P^2_{i+1})$, where
           – $P^1_{i+1}$ is the lectically smallest intent $P$ of $\mathbb{K} \div \mathbb{L}_i$ such that $P_i \leq P$, and there
             exists some $m \in P''_{\mathbb{L}_i} \setminus \mathcal{K}_i(P)$ with $\mathrm{conf}_{\mathbb{K}}(P \to \{m\}) \geqslant c$.
           – $P^2_{i+1}$ is the lectically smallest set $P$ such that $P_i \leq P$, that is closed under
             $\mathcal{K}_i$, but is not an intent of $\mathbb{K} \div \mathbb{L}_i$.
    iii. If $P_{i+1} = P^1_{i+1}$, then set $Q_{i+1} := P_{i+1} \cup \{m\}$, otherwise set $Q_{i+1} := (P_{i+1})''_{\mathbb{K} \div \mathbb{L}_i}$.
    iv.  If $p$ confirms $P_{i+1} \to Q_{i+1}$, then
           – $\mathcal{K}_{i+1} := \mathcal{K}_i \cup \{P_{i+1} \to Q_{i+1}\}$,
           – $\mathcal{L}_{i+1} := \mathcal{L}_i$,
           – $\mathbb{L}_{i+1} := \mathbb{L}_i$.
    v.   If $p$ provides a counterexample $C$ for $P_{i+1} \to Q_{i+1}$, then
           – $\mathcal{K}_{i+1} := \mathcal{K}_i$,
           – $\mathcal{L}_{i+1} := \{(A \to B) \in \mathcal{L}_i \mid C \models (A \to B)\}$,
           – $\mathbb{L}_{i+1} := \mathbb{L}_i + C$.
    vi.  Set $i := i+1$ and go to ii.
Output  Return $\mathcal{K}_i$ and $\mathbb{L}_i$.
```

Based on this result, we shall now adapt our exploration algorithm to ensure that all sets of which we need to check closedness under $\mathrm{Th}_c(\mathbb{K})$ satisfy (3). We can do this as follows: as usual, we consider subsets X of M in lectic order, and for each such set X that is closed under the set $\mathcal{K}_i$ of currently known implications but is not an intent of $\mathbb{K} \div \mathbb{L}_i$, we ask the expert the implication

$$X \to X''_{\mathbb{K} \div \mathbb{L}_i}.$$

Additionally, in accordance with Proposition 1, for each X that is an intent of $\mathbb{K} \div \mathbb{L}_i$ we ask the implication

$$X \to \{m \in M \mid \mathrm{conf}_{\mathbb{K}}(X \to \{m\}) \geqslant c\}.$$

The resulting algorithm is shown in Algorithm 5. It is not hard to see that this algorithm is indeed an instance of Algorithm 3. Therefore, from the general results of Theorem 3 about Algorithm 3, we immediately obtain the following result. Moreover, the algorithm only asks implications with confidence at least c, wherefore Algorithm 5 is a proper algorithm for exploration by confidence.

Corollary 2. *Let $\mathbb{K} = (G, M, I)$ be a finite formal context, p a domain expert on M, $c \in [0,1]$, and $\mathcal{K} \subseteq \mathrm{Th}(p) \cap \mathrm{Th}_c(\mathbb{K})$. Then Algorithm 5 applied to this input terminates after finitely many steps. Let n be the last iteration of the algorithm. Then $\mathbb{K}_n$ is a confident base of* $\mathrm{Th}(p) \cap \mathrm{Th}_c(\mathbb{K})$*, i. e., $\mathcal{K}_n \subseteq \mathrm{Th}_c(\mathbb{K})$ and*

$$\mathrm{Cn}(\mathcal{K}_n) = \mathrm{Cn}(\mathrm{Th}(p) \cap \mathrm{Th}_c(\mathbb{K})).$$

Moreover, for each $(A \to B) \in \mathcal{L}$, either $(A \to B) \in \mathrm{Cn}(\mathcal{K}_n)$ or $(A \to B) \notin \mathrm{Th}(\mathbb{L}_n)$.

6 Outlook and Further Results

In this paper we have addressed the issue of applying attribute exploration to faulty data. We did this by extending the classical attribute exploration algorithm to not only ask implications that are valid in the data, but to ask also those implications that enjoy a *high confidence* therein. The motivation behind this approach was to assume that data which is only slightly faulty will invalidate important implications only with few counterexamples, compared to the number of examples where this implication does apply. Of course, this approach is purely heuristic, and should be treated as such.

In our discussion about how to design an exploration algorithm that takes the confidence of implications into account, we first formalized the notion of an expert. After that, we discussed how classical attribute exploration can be seen as an exploration of *sets of interesting implications*. For this more abstract view, we discussed a straight-forward generalization of the classical algorithm, as well as a weaker generalization which allowed for more freedom in the choice of the implications asked to the experts. Based on these generalization, we developed an approximative as well as an exact algorithm for exploration by confidence.

This paper deliberately avoids giving proofs for the statements it presents. Those proofs can be found in [4]. There we also discuss a generalization of the present results to *general concept inclusions* (GCIs). GCIs are logical formulas which provide a generalization of implications to the realm of *description logics* [2]. It is not hard to generalize the notion of confidence to GCIs, and one can then build upon the results presented in this paper and devise an algorithm for exploring general concept inclusions with high confidence. The immediate advantage of this would be the increase in expressivity provided by the use of description logics.

To generalize exploration by confidence to general concept inclusions, one has to extend the algorithm to also be able to work with *growing sets of attributes*. More precisely, during the exploration, the attribute set M, which is supposed to be fixed in this paper, may grow in a consistent way. Exploration by confidence can be adapted to this setting as well, much like [6] adapts classical attribute exploration to this setting.

A main motivation for considering the case of faulty data is that in real applications data is never free of errors. With respect to this, one could argue that the results of this paper contribute to making attribute exploration more usable in practice. However, this argumentation would be much more convincing if we could provide real-world use cases of exploration by confidence. Finding and evaluating such use cases is a main task for future research.

Another interesting application not discussed so far is the following. As soon as the expert accepts an implication with confidence not equal to 1, all counterexamples of this implication are false. Our algorithm could be adapted to propose these faulty objects to the expert for correction, thereby increasing the quality of the data-set during the course of the exploration. This form of error correction could be more efficient than walking through the whole data-set and correcting all errors. This is because an error-correcting exploration by confidence would only propose errors for correction that are relevant for the exploration process.

Acknowledgments This work has been partially supported by the DFG Research Training Group 1763 “QuantLA”, and by the Cluster of Excellence “Center for Advancing Electronics Dresden” (cfAED). Additionally, the author is grateful to the anonymous reviewers for the detailed and helpful comments.

References

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. “Mining Association Rules between Sets of Items in Large Databases”. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 1993, pp. 207–216.

[2] Franz Baader et al., eds. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[3] Daniel Borchmann. *A General Form of Attribute Exploration*. LTCS-Report 13-02. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, 2013.

[4] Daniel Borchmann. “Learning Terminological Knowledge with High Confidence from Erroneous Data”. PhD thesis. Technische Universität Dresden, 2014.

[5] Peter Burmeister and Richard Holzer. “Treating Incomplete Knowledge in Formal Concept Analysis”. In: *Formal Concept Analysis, Foundations and Applications*. Ed. by Bernhard Ganter, Gerd Stumme, and Rudolf Wille. Vol. 3626. Lecture Notes in Computer Science. Springer, 2005, pp. 114–126.

[6] Felix Distel. “Learning Description Logic Knowledge Bases from Data Using Methods from Formal Concept Analysis”. PhD thesis. Technische Universität Dresden, 2011.

[7] Bernhard Ganter. “Attribute Exploration with Background Knowledge”. In: *Theoretical Computer Science* 217.2 (1999), pp. 215–233.

[8] Bernhard Ganter. “Two Basic Algorithms in Concept Analysis”. In: *Proceedings of the 8th Interational Conference of Formal Concept Analysis*. (Agadir, Morocco). Ed. by Léonard Kwuida and Barış Sertkaya. Vol. 5986. Lecture Notes in Computer Science. Springer, 2010, pp. 312–340.

[9] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999.

[10] Michael Luxenburger. “Implikationen, Abhängigkeiten und Galois-Abbildungen”. PhD thesis. TH Darmstadt, 1993.

[11] Gerd Stumme. “Attribute Exploration with Background Implications and Exceptions”. In: *Data Analysis and Information Systems*. Ed. by Hans-Hermann Bock and Wolfgang Polasek. Studies in Classification, Data Analysis, and Knowledge Organization. Heidelberg: Springer, 1996, pp. 457–469.