TECHNISCHE UNIVERSITÄT DRESDEN

Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Model Exploration by Confidence with Completely Specified Counterexamples

Daniel Borchmann

LTCS-Report 13-11

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

http://lat.inf.tu-dresden.de

Visiting Address:
Nöthnitzer Str. 46
Dresden

Model Exploration by Confidence with Completely Specified Counterexamples

Daniel Borchmann*

November 22, 2013

Abstract

We present an extensions of our previous work on axiomatizing confident general concept inclusions in given finite interpretations. Within this extension we allow external experts to interactively provide counterexamples to general concept inclusions with otherwise enough confidence in the given data. This extensions allows us to distinguish between erroneous counterexamples in the data and rare, but valid counterexamples.

1 Introduction

In recent works [1–3, 5, 6], we have investigated the possibility of learning general concept inclusions (GCIs) from erroneous data based on the notion of *confidence*. For this we have argued that, instead of learning valid GCIs from finite interpretations, considering GCIs whose confidence is above a preselected threshold $c \in [0, 1]$ in some given interpretation $\mathcal{I}$ may be more promising, as it allows us to handle errors which may be present in $\mathcal{I}$. However, we have also argued that this approach itself is intrinsically heuristic, as it does not distinguish between errors in the data and rare but valid counterexamples. Distinguishing those two cannot be done by unsupervised algorithms, as an external source of information is necessary.

To this end we envisioned a supervised learning algorithms for GCIs with high confidence based on the algorithm of *attribute exploration*, a supervised learning algorithm from the area of formal concept analysis that enables us to extract implications from domains which are representable as formal contexts. This algorithm has already been extended to *model exploration* to allow for learning valid GCIs from finite interpretations [7], so an extension towards GCIs with high confidence seems reasonable.

However, attribute exploration itself is more concerned with the question of *completeness* of the initial data, i. e. with the question of whether all relevant (valid) counterexamples are already present in the data. The role of the expert in attribute exploration is then to provide missing counterexamples, by posing questions of the form "does $A \to B$ hold in your domain?" and asking for valid counterexamples if this is not the case. In doing so, the implication $A \to B$ is always true in the already given data.

An extension of model exploration to cover also GCIs with high confidence thus requires some more effort. Luckily, an attribute exploration which extends the classical algorithm to allow also for asking non-valid implications $A \to B$ with high confidence in the initial data has been devised in [4]. Since the original model exploration from [7] has been built upon classical attribute

*Supported by DFG Graduiertenkolleg 1763 (QuantLA)

exploration, and algorithm for *model exploration of GCIs with high confidence* could be devised in a similar way, this time based on the extended exploration algorithm from [4].

It is the purpose of this report to give a detailed derivation of such an algorithm for model exploration of GCIs with high confidence. To this end, we mimic the argumentation of [7] for his development of model exploration and carry it over to consider GCIs with high confidence instead of only valid ones. Therefore, we give a detailed overview of Distel's model exploration in Section 4, and the corresponding adaption to GCIs with high confidence in Section 5. As the considerations itself are quite technical, we summarize in Section 3 the main ideas of how an algorithm for model exploration of GCIs with high confidence should work in principle.

2 Preliminaries

We introduce the some notions necessary to keep this report self contained. To this end, we first cover some basic notions from description logics in Section 2.1, such as *interpretations*, the descriptions logics $\mathcal{EL}^\bot$ and $\mathcal{EL}^\bot_{\text{gfp}}$ as well as *general concept inclusions*. This will also include the definition of *confidence* of general concept inclusions in finite interpretations.

Thereafter we shall recall in Section 2.2 some basic definitions from formal concept analysis, the mathematical area from which the classical attribute exploration algorithm has originated.

2.1 The Description Logics $\mathcal{EL}^\bot$ and $\mathcal{EL}^\bot_{\text{gfp}}$

Description logics are a family of logic-based knowledge representation formalisms that allow for effective decision procedure for various kinds of reasoning tasks. The different flavors of description logic differ in both expressiveness and complexity of reasoning, and the particular choice of which description logic to use usually depends on the underlying application.

In this work, we concentrate on the description logic $\mathcal{EL}^\bot$ and, for technical reasons, on $\mathcal{EL}^\bot_{\text{gfp}}$, and extension of $\mathcal{EL}^\bot$ by greatest fixpoint semantics. However, as defining the semantics of $\mathcal{EL}^\bot_{\text{gfp}}$ is rather involved, and not really needed for the purpose of this work, we shall refrain to do this here. Instead, we refer to [1, 7] and the works cited therein.

As any logic, $\mathcal{EL}^\bot$ is constituted of a syntax and a semantics. Both syntax and semantics depend on the set of *concept names* N_C and the set of *role names* N_R. We demand that N_C and N_R are disjoint. For the syntactic part of $\mathcal{EL}^\bot$ we define an $\mathcal{EL}^\bot$ *concept description* to be a term C formed according to the rule

$$C ::= A \mid \bot \mid C \sqcap C \mid \exists r.C$$

where $A \in N_C$ and $r \in N_R$.

The intuitive understanding behind concept descriptions is that they describe individuals which satisfy certain properties, described by those concept descriptions. In doing so, the intuition behind $\sqcap$ is conjunction, while an existential restriction of the form $\exists r.C$ states that an individual satisfies this description if and only if there exists some r-related individual which satisfies C.

Before we are going to make this more precise by introducing the notion of an *interpretation*, let us first consider a simple example. Let us choose $N_C = \{\, \mathsf{Male}, \mathsf{Emacs} \,\}$ and $N_R\{\, \mathsf{uses} \,\}$. Then the concept description

$$\mathsf{Male} \sqcap \exists \mathsf{uses}.\mathsf{Emacs}$$

can be understood as a description of all individuals which are male and use Emacs, i. e. of all male Emacs users.

To make this understanding more precise we introduce the notion of an interpretation. Interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consist of a set $\Delta^\mathcal{I}$ of *individuals* and a function $\cdot^\mathcal{I}$ mapping concept names A to sets of individuals $A^\mathcal{I}$ and role names r to sets $r^\mathcal{I}$ of pairs of individuals, i. e.

$$\begin{aligned} A^\mathcal{I} &\subseteq \Delta^\mathcal{I} \quad (A \in N_C), \\ r^\mathcal{I} &\subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I} \quad (r \in N_R). \end{aligned}$$

The mapping $\cdot^\mathcal{I}$ is then extended in the obvious way to the set of all $\mathcal{EL}^\bot$ concept descriptions by

$$\begin{aligned} \bot^\mathcal{I} &= \varnothing, \\ (C \sqcap D)^\mathcal{I} &= C^\mathcal{I} \cap D^\mathcal{I}, \\ (\exists r.C)^\mathcal{I} &= \{\, x \in \Delta^\mathcal{I} \mid \exists y \in \Delta^\mathcal{I} \colon (x, y) \in r^\mathcal{I} \text{ and } y \in C^\mathcal{I} \,\}, \end{aligned}$$

for C, D being $\mathcal{EL}^\bot$ concept descriptions and $r \in N_R$.

As already mentioned, we also need to consider an extension of $\mathcal{EL}^\bot$ by *greatest fixpoint semantics*. This is mostly due to the fact that *model-based most-specific concept descriptions*, which we shall introduce in Section 4.1 and which are crucial for our considerations, do not necessarily exist in $\mathcal{EL}^\bot$, but do in $\mathcal{EL}^\bot_{\text{gfp}}$.

As we have already noted, the semantics of $\mathcal{EL}^\bot_{\text{gfp}}$ are quite involved, and since they are not strictly necessary for our considerations, we shall spare to define it here. However, we shall give a brief overview over the syntax of $\mathcal{EL}^\bot_{\text{gfp}}$ to give some intuition for this description logic.

Again, let N_C and N_R be two disjoint sets, and let N_D be another, non-empty set being disjoint to both N_C and N_R, the set of *defined concept names*. Then an $\mathcal{EL}^\bot_{\text{gfp}}$ concept description C is either of the form $C = \bot$ or of the form

$$C = (A, \mathcal{T}),$$

where $A \in N_D$ and $\mathcal{T}$ is a *normalized $\mathcal{EL}^\bot$-TBox with defined concept names N_D*. Here an $\mathcal{EL}^\bot$-TBox with defined concept names N_D is a finite set of *primitive concept definitions*, i. e. expressions of the form

$$B \equiv D$$

such that $B \in N_D$ and D being an $\mathcal{EL}^\bot$ concept description with concept names in $N_C \cup N_D$ and role names in N_R. Additionally, for each $B \in N_D$ there is exactly one concept definition of the form $B \equiv D$ in $\mathcal{T}$.

We say that the primitive concept definition $B \equiv D$ is *normalized* if D is of the form

$$D = B_1 \sqcap \cdots \sqcap B_m \sqcap \exists r_1.D_1 \sqcap \ldots \exists r_n.D_n$$

for some $m, n \in \mathbb{N}$, $B_1, \ldots, B_m \in N_C$ and $D_1, \ldots, D_n \in N_D$. We say that $\mathcal{T}$ is normalized if all the concept definitions in $\mathcal{T}$ are normalized.

Let us consider a toy example. For this, let $N_C = \{\, \mathsf{Nerd}, \mathsf{Emacs}, \mathsf{Vim} \,\}$, $N_R = \{\, \mathsf{uses}, \mathsf{dislikes} \,\}$, $N_D = \{\, \mathsf{EmacsUser}, \mathsf{VimUser}, \mathsf{E}, \mathsf{V} \,\}$. Then we can define typical users of Emacs as

$$\begin{aligned} (\mathsf{EmacsUser}, \{\, \mathsf{EmacsUser} &\equiv \mathsf{Nerd} \sqcap \exists \mathsf{uses}.\mathsf{E} \sqcap \exists \mathsf{dislikes}.\mathsf{VimUser}, \\ \mathsf{E} &\equiv \mathsf{Emacs}, \\ \mathsf{VimUser} &\equiv \mathsf{Nerd} \sqcap \exists \mathsf{uses}.\mathsf{V} \sqcap \exists \mathsf{dislikes}.\mathsf{EmacsUser}, \\ \mathsf{V} &\equiv \mathsf{Vim} \,\}). \end{aligned}$$

In this example, an EmacsUser is defined to be a Nerd that uses Emacs and that dislikes someone who is a VimUser.[1] However, the concept of a VimUser is also defined in terms of EmacsUser, expressing a cyclic dependency between these two concepts. Thus, $\mathcal{EL}^{\bot}_{\text{gfp}}$ concept descriptions allow us to express some kind of local recursion. To resolve these cyclic dependencies, $\mathcal{EL}^{\bot}_{\text{gfp}}$ makes use of *greatest fixpoint semantics*, which we are not discussing here. See [7] and the references therein for more details on this.

In the following, we may speak of *concept descriptions* without explicitly noting the description logic used (like in $\mathcal{EL}^{\bot}_{gfp}$ *concept description*). In this case, the actual description logic used is not relevant, and the considerations are then meant to be valid for both $\mathcal{EL}^{\bot}$ and $\mathcal{EL}^{\bot}_{\text{gfp}}$. Furthermore, we shall also not mention the sets N_C, N_R and N_D explicitly if they are clear from the context.

For some interpretation $\mathcal{I}$ and some concept descriptions C, D it may be the case that for each $x \in \Delta^{\mathcal{I}}$ it is always true that $x \in C^{\mathcal{I}}$ implies $x \in D^{\mathcal{I}}$ – in other words, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. In this case we can say that if the concept description C holds, then the concept description D holds as well. This implication-like connection between the two concept descriptions C and D can be captured by the notion of a *general concept inclusion* (GCI). GCIs are written as $C \sqsubseteq D$, and such a GCI is said to be *valid* in $\mathcal{I}$ (or *holds* in $\mathcal{I}$) if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. If $C \sqsubseteq D$ is true in every interpretation $\mathcal{I}$, then we say that C is *subsumed* by D, and write $C \sqsubseteq D$ (meant as a statement, not an expression). The set of all GCIs which are valid in $\mathcal{I}$ is denoted with $\text{Th}(\mathcal{I})$.

The kind of knowledge represented by GCIs can be very useful, and *learning* which GCIs are valid in $\mathcal{I}$ can be of practical importance, for example when creating *knowledge bases*. However, it can be seen quite easily that the number of GCIs valid in $\mathcal{I}$ is infinite in general (i. e. if $N_R \neq \varnothing$), since then if $C \sqsubseteq D$ is valid in $\mathcal{I}$, so is $\exists r.C \sqsubseteq \exists r.D$ for each $r \in N_R$. Therefore, learning all GCIs which are valid in $\mathcal{I}$ is practically unreasonable.

In [7] it has be shown that if $\mathcal{I}$ is finite, then there exists a *finite* set $\mathcal{B}$ of valid GCIs of $\mathcal{I}$ which is *complete* in the sense that every GCIs which is valid in $\mathcal{I}$ already follows from $\mathcal{B}$. Such finite and complete sets of valid GCIs of $\mathcal{I}$ are called *finite bases* of $\mathcal{I}$.

For practical applications it may be relevant to allow the interpretation $\mathcal{I}$ to contain *errors*, in the sense that sometimes $x \in C^{\mathcal{I}}$ may hold although it should not, and vice versa, for $x \in \Delta^{\mathcal{I}}$ and some concept description C. In this case it may happen that some GCIs which actually should be true in $\mathcal{I}$ are not, since some errors inhibit this. In this case, learning only valid GCIs of $\mathcal{I}$ will miss those incidentally invalidated GCIs, although they may be of practical value.

Of course, without further specification of what kinds of errors we actually allow in our data, the problem of learning all those GCIs which *would* actually be valid if all those errors were not present does not have a satisfying theoretical solution. However, we can approach this problem heuristically using the notion of *confidence* as it is used in data mining: assuming that errors are rare enough, a GCI $C \sqsubseteq D$ which actually has "few" counterexamples compared to the number of individuals where it actually holds can be assumed to be only falsified by errors (since otherwise counterexamples in the data would be more frequent). Here the phrase "individuals where it actually holds" is meant to describe all individuals $x \in \Delta^{\mathcal{I}}$ which satisfy $x \in (C \sqcap D)^{\mathcal{I}}$. Those individuals can be thought of as *positive* examples for $C \sqsubseteq D$, where counterexamples are *negative* examples for $C \sqsubseteq D$. Then, intuitively speaking, $C \sqsubseteq D$ has "high" confidence if and only if the number of positive examples for $C \sqsubseteq D$ is "much higher" then the number of negative ones.

[1] Actually, what we would like to express here is that every Emacs user does not like every Vim user and vice versa. However, this cannot be expressed in $\mathcal{EL}^{\bot}_{\text{gfp}}$, and for the sake of keeping the example simple, we just stick with the formulation as given.

To put this formally, we define for $C \sqsubseteq D$ its confidence in $\mathcal{I}$ to be

$$\operatorname{conf}_{\mathcal{I}}(C \sqsubseteq D) := \begin{cases} 1 & \text{if } C = \varnothing, \\ \frac{|(C \sqcap D)^{\mathcal{I}}|}{|C^{\mathcal{I}}|} & \text{otherwise.} \end{cases}$$

Then to say that a GCI has only "few" errors, we choose a threshold $c \in [0, 1]$ and require $\operatorname{conf}_{\mathcal{I}}(C \sqsubseteq D) \geqslant c$. The set of all GCIs which have confidence at least c in $\mathcal{I}$ is denoted with $\operatorname{Th}_c(\mathcal{I})$, and elements of this set are sometimes called *GCIs with high confidence* or just *confident GCIs* (where it is clear from the context which threshold is meant).

Learning GCIs from $\operatorname{Th}_c(\mathcal{I})$ can be promising if $\mathcal{I}$ contains errors which are rare. However, since $\operatorname{Th}(\mathcal{I}) \subseteq \operatorname{Th}_c(\mathcal{I})$, the set $\operatorname{Th}_c(\mathcal{I})$ is infinite in general, and also in general not even closed under entailment of GCIs. However, it has been shown in previous works [1–3, 5, 6] that if $\mathcal{I}$ is finite, it is possible to effectively construct finite sets $\mathcal{B}$ of $\operatorname{Th}_c(\mathcal{I})$ which are *sound* for $\operatorname{Th}_c(\mathcal{I})$, in the sense that all elements of $\mathcal{B}$ are entailed by $\operatorname{Th}_c(\mathcal{I})$, and also complete for $\operatorname{Th}_c(\mathcal{I})$, in the sense that each element of $\operatorname{Th}_c(\mathcal{I})$ is entailed by $\mathcal{B}$. Such sets $\mathcal{B}$ are called *finite bases* of $\operatorname{Th}_c(\mathcal{I})$. If even $\mathcal{B} \subseteq \operatorname{Th}_c(\mathcal{I})$, then $\mathcal{B}$ is called a *confident* base of $\operatorname{Th}_c(\mathcal{I})$.

2.2 Formal Concept Analysis

Formal concept analysis provides the method of attribute exploration on which we want to built our model exploration of confident GCIs. The field of formal concept analysis originally started out as an attempt to restructure mathematical lattice theory, and in particular to provide *meaning* to ordered structures as structures of *concepts* or as *conceptual hierarchies*. However, it has since then evolved into a broad theory with applications beyond its original scope, for example in artificial intelligence and data mining. Indeed, we shall see very little from this original motivation of formal concept analysis in this report.

The most basic notion of formal concept analysis is the one of a *formal context*, a triple $\mathbb{K} = (G, M, I)$ where G and M are sets and $I \subseteq G \times M$. A classical intuition behind this definition is that the set G is thought of as a set of *objects*, the set M as a set of *attributes* and $(g, m) \in I$ is to be understood as that the object g *has* the attribute m. However, this interpretation is only one among many.

Formal contexts are a basic way to represent data, and with the above interpretation we can pose simple questions to this data. For example, if $A \subseteq M$ is a set of attributes, then we can ask for all objects A' in $\mathbb{K}$ which have all attributes in A, i. e.

$$A' := \{\, g \in G \mid \forall m \in A \colon (g, m) \in I \,\}.$$

Dually, if $B \subseteq G$ is a set of objects, then we can look for all attributes B' which are shared by all objects in B, i. e. for the set B' which is defined as

$$B' := \{\, m \in M \mid \forall g \in B \colon (g, m) \in I \,\}.$$

The mappings denoted both by $(\cdot)'$ are called the *derivation operators* of $\mathbb{K}$, and the sets A' and B' are called the *derivations* of A and B in $\mathbb{K}$, respectively.

In the formal context $\mathbb{K}$ it may happen that for two sets $A, B \subseteq M$ of attributes that whenever an object has all attributes from A, it also has all attributes from B. In other words, it may be the case that

$$A' \subseteq B'.$$

This dependency between sets of attributes of formal contexts is captured in the notion of an *implication*. Formally, an implication on the set M is just a pair (A, B) where $A, B \subseteq M$.

To make the intention more clear, implications are mostly written as $A \to B$. The set of all implications on M is denoted by $\mathrm{Imp}(M)$. We say that an implication *holds* in $\mathbb{K}$, written $\mathbb{K} \models (A \to B)$, if and only if $A' \subseteq B'$, i. e. if and only if each object which has all the attributes from A also has all the attributes from B. The set of all valid implications of $\mathbb{K}$ is denoted with $\mathrm{Th}(\mathbb{K})$.

Let $\mathcal{L} \subseteq \mathrm{Imp}(M)$ and let $(A \to B) \in \mathrm{Imp}(M)$. We say that $\mathcal{L}$ *entails* $A \to B$ if and only if for each formal context $\mathbb{L}$ with attribute set M, it is true that if all implications in $\mathcal{L}$ are valid in $\mathbb{L}$, then $A \to B$ is also valid in $\mathbb{L}$. The set of all implications which are entailed by $\mathcal{L}$ is denoted by $\mathrm{Cn}_M(\mathcal{L})$. If the set M is clear from the context, then we may drop the subscript from Cn_M.

Let $X \subseteq M$. Then the set X is said to be *closed* under $\mathcal{L}$ if and only if for all $(A \to B) \in \mathrm{Imp}(M)$, it is true that if $A \subseteq X$, then $B \subseteq X$. It can be shown that there always exists a $\subseteq$-minimal set $\bar{X}$ satisfying $X \subseteq \bar{X}$ such that $\bar{X}$ is closed under $\mathcal{L}$. This set shall be denoted by $\mathcal{L}(X)$.

If the formal context $\mathbb{K}$ is finite, i. e. if both G and M are finite, then the number of valid implications of $\mathbb{K}$ is finite as well. However, this set can be quite large, and it may be desirable to find smaller sets of implications which are sufficient. For this, let $\mathcal{L}, \mathcal{K} \subseteq \mathrm{Imp}(M)$. Then we say that $\mathcal{K}$ is a *base* of $\mathcal{L}$ if and only if $\mathrm{Cn}(\mathcal{K}) = \mathrm{Cn}(\mathcal{L})$. In other words, $\mathcal{K}$ is a base of $\mathcal{L}$ if and only if it entails all implications which are entailed by $\mathcal{L}$ and nothing more. If $\mathcal{L} = \mathrm{Th}(\mathbb{K})$, then bases of $\mathcal{L}$ are also called bases of $\mathbb{K}$.

The idea behind this notion is to find, given a set $\mathcal{L}$, a considerable smaller base $\mathcal{K}$ of $\mathcal{L}$. This base $\mathcal{K}$ then still has the same properties as the set $\mathcal{L}$ in terms of entailment, but might be practically more relevant because of its reduced size. Of course, the best solution here would be to find bases $\mathcal{K}$ which are somehow "smallest." In particular, we say that $\mathcal{K}$ is *non-redundant* if no strict subset of $\mathcal{K}$ is a base of $\mathcal{L}$, and we say that $\mathcal{K}$ is *minimal* if there does not exist another base of $\mathcal{L}$ with fewer elements than $\mathcal{K}$.

Bases can be formed according to some known *background knowledge*. Let $\mathcal{L}, \mathcal{K}' \subseteq \mathrm{Imp}(M)$ be again sets of implications such that $\mathrm{Cn}(\mathcal{K}') \subseteq \mathrm{Cn}(\mathcal{L})$. We may consider the set $\mathcal{K}'$ as implications we "already know," and we may be interested in finding a set $\mathcal{K} \subseteq \mathrm{Imp}(M)$ such that $\mathcal{K}$ together with $\mathcal{K}'$ is a base of $\mathcal{L}$, i. e. such that $\mathrm{Cn}(\mathcal{K} \cup \mathcal{K}') = \mathrm{Cn}(\mathcal{L})$. In this case, we call $\mathcal{K}$ a *base of* $\mathcal{L}$ *with background knowledge* $\mathcal{K}'$. The notions or non-redundancy and minimality of bases $\mathcal{K}$ with background knowledge $\mathcal{K}'$ are as in the case for bases.

A particularly interesting base of a formal context $\mathbb{K}$, which is also a minimal base, is the *canonical base* of $\mathbb{K}$. To define this base we need to introduce the notion of *pseudo-intents* of $\mathbb{K}$ with respect to some background knowledge $\mathcal{K} \subseteq \mathrm{Th}(\mathbb{K})$. Let $P \subseteq M$. Then P is called a pseudo-intent of $\mathbb{K}$ if and only if

i. $P \neq P''$,

ii. $P = \mathcal{K}(P)$, and

iii. for each pseudo-intent $Q \subsetneq P$ it is true that $Q'' \subseteq P$.

Then the canonical base of $\mathbb{K}$ with background knowledge $\mathcal{K}$ is defined as

$$\mathrm{Can}(\mathbb{K}, \mathcal{K}) := \{\, P \to P'' \mid P \text{ pseudo-intent of } \mathbb{K} \,\}.$$

The canonical base is always a minimal base of $\mathbb{K}$.

To compute the canonical base, one commonly uses an algorithm which is base on Next-Closure [8]. Next-Closure is an algorithm which allows for the enumeration of all closed sets of $\mathcal{L}$ in a particular order, the *lectic order*.

Algorithm 1 The Next-Closure Algorithm

```
0 define next-closed-set(M, <, P, L)
1   < := induced lectic order of <
2   for each i ∈ sort(>, M) do ; i.e from largest to smallest
3     P̄ := L({ a ∈ A | a < i } ∪ { i })
4     if P ≺_i P̄ then
5       return P̄
6     end
7   end
8   return null
9 end
```

Let $<$ be a strict order on M, i. e. a transitive and irreflexive binary relation on M. Then for two sets $A, B \subseteq M$ and $i \in M$ we say that A is *lectically smaller then B at position i*, written $A \prec_i B$, if and only if

$$\min_<(A \triangle B) = i \in B.$$

We say that A is *lectically smaller* than B if and only if $A \prec_i B$ for some $i \in M$. We write $A \prec B$ in this case and call $\prec$ on *lectic order induced by $<$ on M*. We write $A \preceq B$ if and only if $A \prec B$ or $A = B$.

It can be shown quite easily that lectic orders are linear orders on the powerset of M, and that these orders extent the usual subset order, i. e.

$$A \subseteq B \implies A \preceq B$$

is true for all $A, B \subseteq M$.

Now given a set $\mathcal{L} \subseteq \mathrm{Imp}(M)$ of implications on M, and a set $P \subseteq M$, the Next-Closure algorithm is able to compute the lectically next set $\bar{P}$ after P which is closed under $\mathcal{L}$, i. e.

$$\bar{P} = \min_\prec \{ Q \subseteq M \mid P \prec Q \text{ and } Q \text{ is closed under } \mathcal{L} \}.$$

The algorithm is shown as Algorithm 1. Note that we shall not discuss the details of this algorithm here, see [8, 9] for a more thorough introduction.

It is now true that

$$\bar{P} = \texttt{next-closed-set}(M, <, P, \mathcal{L}).$$

If $\mathcal{L} = \mathrm{Th}(\mathbb{K})$, we may also write $\bar{P} = \texttt{next-closed-set}(M, <, P, \mathbb{K})$.

The canonical base of $\mathbb{K}$ with respect to some background knowledge $\mathcal{K}$ can be computed with the help of the Next-Closure algorithm. However, instead of giving an explicit algorithm to achieve this, we shall instead the algorithm of *attribute exploration*, which is closely related to computing the canonical base. Indeed, it is rather easy to extract from attribute exploration an algorithm for computing the canonical base.

Attribute exploration is an algorithm which approaches the problem of checking a formal context for *completeness* with respect to representing a certain domain of interest. More precisely, suppose that we are interested in the implicational theory of a certain domain formed of certain instances, from which we assume that they can be modeled as objects of a formal context. However, we also assume that actually constructing this formal context is not feasible, or at least not desired, and that instead a base of all implications which hold in this domain is of interest. We assume furthermore that we have access to an *expert* which is able to answer questions of the form "Does the implication $A \to B$ holds in the domain?" for every implication

Algorithm 2 Attribute Exploration

```
0  define attribute-exploration(𝕂 = (G, M, I), 𝒦)
1    P := ∅
2    forever do
3      P := min≺ { Q ⊆ M | P ⪯ Q, Q = 𝒦(Q), Q ≠ Q'' }
4      if P = null then
5        return 𝕂, 𝒦
6      else if expert accepts P → P'' then
7        𝒦 := 𝒦 ∪ { P → P'' }
8      else
9        𝕂 := 𝕂 augmented with a counterexample for P → P''
10     end
11   end
12 end
```

$(A \to B) \in \mathrm{Imp}(M)$. If the expert answers no, she is required to provide a *counterexample* for the implication $A \to B$, i. e. a set $C \subseteq M$ such that $A \subseteq C$ and $B \nsubseteq C$.

In this setup we can employ attribute exploration to obtain a base of all valid implications of the domain of interest. To this end, the algorithm successively computes implications $A \to B$ and poses them to the expert, collecting provided counterexamples if the expert rejects $A \to B$, or remembering the implication $A \to B$ if the expert accepts it. By collecting those counterexamples, the algorithm *completes* the initial formal context to contain all objects and their corresponding attributes which are relevant to extract a base of all valid implications of the domain of interest.

The algorithm proceeds in detail as follows: starting with an *initial working context* $\mathbb{K}$ which is supposed to contain instances from the domain of interest, and a set $\mathcal{K}$ of *initially known implications*, which are supposed to be true in the domain, the algorithm computes the lectically first closed set P of $\mathcal{K}$ which is not an intent of $\mathbb{K}$. It then poses the question

$$P \to P''$$

to the expert. If the expert accepts this implication, it is added to $\mathcal{K}$. If the expert rejects this implication, the provided counterexample is added to the working context $\mathbb{K}$ as a new object. The formal context $\mathbb{K}$ is then called the *current working context* and the set $\mathcal{K}$ is called the set of *currently known implications*.

In any case, the algorithm continues with computing the $\prec$-smallest set Q satisfying $P \preceq Q$ and not being an intent of $\mathbb{K}$. If Q exists, the expert is asked the implication $Q \to Q''$ and the answer is handled as before. If no such set Q exists, both $\mathbb{K}$ and $\mathcal{K}$ are returned, which are then called the *final working context* and the *final set of known implications*, respectively. In this case, $\mathcal{K}$ is a base of all implications which are valid in the domain of interest, and is equal to the canonical base of $\mathbb{K}$ with the initially known implications as background knowledge [9].

An implementation of attribute exploration is shown as Algorithm 2. The expression

$$\min_{\prec} \{ Q \subseteq M \mid P \preceq Q, Q \neq Q'' \}$$

which appears in this listing is computed by repeated calls to the Next-Closure algorithm. It is supposed to be evaluated to null if the set is empty.

3 An Overview over Model Exploration

The main focus of this work is to introduce an algorithm that allows for the exploration of all general concept inclusions that enjoy a certain minimal confidence in a given finite interpretation. As the following considerations are going to be quite technical, we want to ensure that the reader does not miss this goal. To this end, we want to provide a high-level description of the desired algorithm in this section. Additionally, we shall also introduce some nomenclature that we are going to use throughout our argumentation.

Let us first state the main setting for such an algorithm that implements model exploration by confidence. For this, assume that we are interested to learn knowledge from a particular domain of interest. From this domain we assume that it can be represented as a finite interpretation $\mathcal{I}_{\text{back}}$, which we shall call the *background interpretation*. Furthermore, we assume that we do not explicitly have access to this background interpretation, because it may only exist implicitly. However, we can access the interpretation $\mathcal{I}_{\text{back}}$ through an *external expert*, which has the ability to *answer questions*.

This external expert should, given a GCI $C \sqsubseteq D$, be able to determine if this GCI holds in $\mathcal{I}_{\text{back}}$ or not. If it is the case, then the expert *confirms* this GCI. If $C \sqsubseteq D$ does not hold in $\mathcal{I}_{\text{back}}$, then the expert should provide a *counterexample*, i. e. an element x from the interpretation $\mathcal{I}_{\text{back}}$ such that $x \in C^{\mathcal{I}_{\text{back}}}$ and $x \notin D^{\mathcal{I}_{\text{back}}}$. It is also possible (and, as we shall later, may even be necessary) to not only provide the element x but also a small *subinterpretation* $\mathcal{I}'$ of $\mathcal{I}_{\text{back}}$ such that $C^{\mathcal{I}'} \not\subseteq D^{\mathcal{I}'}$.

The goal of an exploration algorithm in this setting is now to axiomatize valid GCIs of $\mathcal{I}_{\text{back}}$ by posing questions to this expert. Intuitively, this is done by computing GCIs and giving them to the expert. If the expert confirms, those GCIs are stored. If the expert rejects, the provided counterexample is stored as well. Subsequently, GCIs are computed such that they neither follow from already confirmed GCIs nor are invalidated by already provided counterexamples, to avoid unnecessary questions to the expert. The algorithm terminates if no such GCIs can be computed anymore, in which case the set of known GCIs constitutes an axiomatization of the valid GCIs of $\mathcal{I}_{\text{back}}$. Of course, termination of such an algorithm is not clear per se, and heavily depends on the order in the GCIs asked to the expert are computed. That such an algorithm exists has been shown in [7].

In every iteration, the set of GCIs already confirmed by the expert is called the *current background knowledge*, and the interpretation which is constituted as the union of all counterexamples provided so far is called the *current working interpretation*. It is also possible to provide some *initial background knowledge* and an *initial working interpretation* as arguments to the exploration. Clearly, the initial background knowledge should be valid in $\mathcal{I}_{\text{back}}$.

However, in contrast to the exploration algorithm in [7], we do not require the initial working interpretation to be a subinterpretation of $\mathcal{I}_{\text{back}}$. This we allow because we want to consider an initial working interpretation as an "approximation" of (parts of) the background interpretation, to allow also data that is actually not part of the background interpretation to be used as input for an exploration process. In that way, we make exploration applicable to data which itself not correct, but contains only few errors (i. e. , is of "high quality"), a scenario which is quite common for practical applications.

To illustrate why this can happen, let us consider the following fictional scenario. We have access to an expert from the domain of diseases, and we have given an interpretation that contains diseases as elements (among others). To goal is to extract knowledge about diseases from the expert, using the initial interpretation to reduce the number of GCIs to be considered. However, the data itself might very likely contain errors (as all real-world data does), and therefore it cannot be considered as part of the underlying background interpretation. On the other hand,

the number of errors might be small enough, i. e. the interpretation might be of high quality, and simply discarding the data because of some small errors seems wasteful.

To make exploration applicable here, recall that the working interpretation is used to exclude certain GCIs from being asked to the expert because those GCIs are not valid in the working interpretation. If the interpretation contains errors, this approach cannot be followed anymore, as the errors contained in the working interpretation may accidentally invalidate GCIs which actually hold in the background interpretation. However, if we assume that the interpretation contains *not too many* errors, we can still use this data by not asking GCIs that have "enough" counterexamples in the data. To make this more precise, we use the notion of *confidence* of GCIs to quantify when a GCI has "enough" counterexamples in the data.

An algorithm that implements this idea would abstractly work as follows. As initial arguments, the algorithm would receive

i. an initial working interpretation $\mathcal{I}$,

ii. a confidence threshold $c \in [0, 1]$, and

iii. a set $\mathcal{B}$ of known general concept inclusions, satisfying that every element of $\mathcal{B}$ should have confidence at least c in $\mathcal{I}$.

Given these arguments, the algorithm then successively computes general concept inclusions $C \sqsubseteq D$ whose confidence is at least c in $\mathcal{I}$, that are not entailed by GCIs from $\mathcal{B}$ or GCIs previously confirmed by the expert, and that are not invalidated by any counterexample provided by the expert so far. As above, those GCIs having been confirmed by the expert during this process are collected. Upon termination, this set of GCIs constitutes a finite axiomatization of the GCIs which are both valid in the domain represented by the expert and have confidence at least c in $\mathcal{I}$.

Note that we demand the confidence of $C \sqsubseteq D$ to always be computed in the initial working interpretation, and not in the current working interpretation. The reason for this is twofold. Firstly, the counterexamples added by the expert are of a different "quality," since we assume them to come directly from the background interpretation. Therefore, considering them while computing the confidence of GCIs seems unreasonable. Secondly, recall that the initial purpose of allowing an initial working interpretation is to reduce the number of GCIs to be considered during the exploration. In our particular setting now, we do this by considering only those GCIs to be relevant to the expert that enjoy a certain confidence in $\mathcal{I}$. Then, during the exploration, we use the counterexamples provided by the expert to reduce this set of GCIs until it is empty. However, computing the confidence of GCIs in the whole working interpretation would do something completely different.

The remainder of this work is devoted to develop such an exploration algorithm and to prove the claims given here. In particular, it remains to discuss how to compute GCIs to be asked to expert. To this end, we shall first have a look on how model exploration in the case of *valid* GCIs can be achieved (i. e. in the case where the initial working interpretation is actually part of the background interpretation), as it has been described first by Distel [7]. This is the subject of Section 4. The argumentation used there will guide us in Section 5 in our development of a model exploration for confident GCIs, as it has been sketched above.

4 Model Exploration with Valid GCIs

The argumentation we are going to use to develop our algorithm for model exploration with confident GCIs is similar to the argumentation of Distel [7] for his development of a model

exploration algorithm for valid GCIs. Therefore, to make our considerations more comprehensible and to allow comparisons between our exploration algorithm and Distel's, we are going to give a brief description of his model exploration algorithm in the following sections. However, we shall refrain from giving any proofs, as this is not the purpose of this report. Instead, we shall give detailed references to the corresponding passages in [7].

4.1 Axiomatizing GCIs of Finite Models

The algorithm for model exploration developed by Distel relies on a generalization of an algorithm that computes bases of valid GCIs of finite given models, also developed by him [7]. Therefore, to understand the reasoning behind Distel's model exploration algorithm, we shall give a brief introduction into axiomatizing finite models first.

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ a finite interpretation. The main idea behind Distel's approach in axiomatizing $\mathcal{I}$ is to transform this interpretation into a *finite* formal context $\mathbb{K}_{\mathcal{I}}$ in such a way that a base of the valid implications of $\mathbb{K}_{\mathcal{I}}$ yield finite bases of $\mathcal{I}$.

To construct a finite base in such a way, we shall first introduce the notion of *model-based most-specific concept descriptions*. For this, let $X \subseteq \Delta^{\mathcal{I}}$ be a set of individuals. Then a concept description C is a model-based most-specific concept description for X in $\mathcal{I}$ if and only if

i. $X \subseteq C^{\mathcal{I}}$ and

ii. for all concept descriptions D with $X \subseteq D^{\mathcal{I}}$ it is true that $C \sqsubseteq D$.

In other words, C is the most specific concept description that "describes" X in $\mathcal{I}$.

It can be seen easily that if model-based most-specific concept descriptions exist, then they are unique up to equivalence. Therefore, we can denote model-based most-specific concept descriptions by a separate name, and we choose to call it $X^{\mathcal{I}}$, because mapping X to its model-based most-specific concept description can be considered as *reverse* operations to mapping a concept description C to its extension $C^{\mathcal{I}}$, in the very same way as the derivation operators $(\cdot)'$ in formal contexts are reverse operations of each other.[2] To ease notation, we shall write $X^{\mathcal{I}\mathcal{I}}$ instead of $(X^{\mathcal{I}})^{\mathcal{I}}$ and $C^{\mathcal{I}\mathcal{I}}$ instead of $(C^{\mathcal{I}})^{\mathcal{I}}$ for sets X of individuals and concept descriptions C.

However, it can also be seen quite easily that model-based most-specific concept descriptions do not necessarily exist in the description logic $\mathcal{EL}^{\bot}$. For this, we consider the case of $N_C = \varnothing$, $N_R = \{ \mathsf{r} \}$ and $\mathcal{J} = (\{ a \}, \cdot^{\mathcal{J}})$ where $r^{\mathcal{J}} = \{ (a, a) \}$. Then the set $X = \{ a \}$ is the extension of the $\mathcal{EL}^{\bot}$ concept descriptions

$$\top, \exists \mathsf{r}.\top, \exists \mathsf{r}.\exists \mathsf{r}.\top, \ldots$$

and there is no most-specific concept descriptions C such that $X \subseteq C^{\mathcal{I}}$.

Therefore we consider the description logic $\mathcal{EL}^{\bot}_{\text{gfp}}$ instead of $\mathcal{EL}^{\bot}$. For this logic, it can be shown that model-based most-specific concept descriptions always exist, and that they can be computed effectively [7, Theorem 4.7]. Moreover, considering $\mathcal{EL}^{\bot}_{\text{gfp}}$ instead of $\mathcal{EL}^{\bot}$ is not a restriction, as it can be shown that any finite base of $\mathcal{I}$ in $\mathcal{EL}^{\bot}_{\text{gfp}}$ can effectively be transformed into a finite base of $\mathcal{I}$ in $\mathcal{EL}^{\bot}$ [7, Theorem 5.21].

Model-based most-specific concept descriptions provide a way to mimic the derivation operators from formal concept analysis. Recalling our goal of transforming $\mathcal{I}$ into a finite formal context,

[2] Indeed, the mappings $C \mapsto C^{\mathcal{I}}$ and $X \mapsto X^{\mathcal{I}}$ a very similar to the derivations operators in formal concept analysis. In particular, they resemble a Galois connection between the set of concept descriptions ordered by subsumption and the subsets of $\Delta^{\mathcal{I}}$ ordered by set inclusion.

we shall introduce the notion of *induced contexts* next.

Let M be a set of concept descriptions. Then the formal context $\mathbb{K}_{M,\mathcal{I}} = (\Delta^{\mathcal{I}}, M, \nabla)$, where

$$x \nabla C :\Longleftrightarrow x \in C^{\mathcal{I}}$$

for $x \in \Delta^{\mathcal{I}}$ and $C \in M$ is called the *induced formal context* of M and $\mathcal{I}$.

Induced contexts exemplify many commonalities between the operators $(\cdot)^{\mathcal{I}}$ and the derivation operators $(\cdot)'$. One of them is shown in the following result.

4.1 Lemma (partly Lemma 4.10 from [7]) *Let M be a set of concept descriptions and let $\mathcal{I}$ be an interpretation. Then for each $U \subseteq M$ it is true that*

$$(\bigsqcap U)^{\mathcal{I}} = U'.$$

Recall that for a formal context $\mathbb{K} = (G, M, I)$ the set $\{ A \to A'' \mid A \subseteq M \}$ is always a base of $\mathbb{K}$. In particular, this is true for induced formal contexts $\mathbb{K}_{M,\mathcal{I}}$. Now given the similarities between $(\cdot)^{\mathcal{I}}$ and $(\cdot)'$, a desirable result would be to obtain that the set

$$\mathcal{B} := \{ \bigsqcap U \sqcap (\bigsqcap U)^{\mathcal{I}\mathcal{I}} \mid U \subseteq M \} \tag{1}$$

is a base of $\mathcal{I}$. While this is not true in general, we can establish the validity of this result by considering a the particular set

$$M_{\mathcal{I}} = \{ \bot \} \cup N_C \cup \{ \exists r.X^{\mathcal{I}} \mid X \subseteq \Delta^{\mathcal{I}} \},$$

which in addition is also finite, as $\Delta^{\mathcal{I}}$ is a finite set. Then it is shown in [7, Theorem 5.10] that the set $\mathcal{B}$ from (1) is indeed a finite base of $\mathcal{I}$, when choosing $M = M_{\mathcal{I}}$. As the set $M_{\mathcal{I}}$ has this remarkable property, we shall denote the induced formal context $\mathbb{K}_{M_{\mathcal{I}},\mathcal{I}}$ of $M_{\mathcal{I}}$ and $\mathcal{I}$ simply by $\mathbb{K}_{\mathcal{I}}$ and shall call it the *induced formal context* of $\mathcal{I}$.

The set $M_{\mathcal{I}}$ also enjoys another important property: if $X \subseteq \Delta^{\mathcal{I}}$ is a set of individuals, then its model-based most-specific concept description $X^{\mathcal{I}}$ is *expressible in terms of $M_{\mathcal{I}}$*, i. e. it can be expressed (up to equivalence) as a conjunction of concept descriptions from $M_{\mathcal{I}}$ [7, Lemma 5.9]. Formally, for each such set X there exists a set $N \subseteq M_{\mathcal{I}}$ such that

$$X^{\mathcal{I}} \equiv \bigsqcap N.$$

With this property it can be shown that the intents of $\mathbb{K}_{\mathcal{I}}$ are in a one-to-one correspondence with the model-based most-specific concept descriptions of $\mathcal{I}$. A crucial step towards this result is formulated in the following lemma.

4.2 Lemma (Proposition 4.7 from [1]) *Let $\mathcal{I}$ be a finite interpretation and let M be a set of concept descriptions. Let $A \subseteq \Delta^{\mathcal{I}}$. Then*

$$\bigsqcap A' \equiv A^{\mathcal{I}},$$

where the derivation is done in $\mathbb{K}_{\mathcal{I}}$.

If now $U \subseteq M_{\mathcal{I}}$, then we obtain from this and Lemma 4.1

$$(\bigsqcap U'') \equiv (U')^{\mathcal{I}} = (\bigsqcap U)^{\mathcal{I}\mathcal{I}}. \tag{2}$$

It can now be shown that bases of $\mathbb{K}_{\mathcal{I}}$ can be transformed into bases of $\mathcal{I}$. More precisely, if $\mathcal{L}$ is a base of $\mathbb{K}_{\mathcal{I}}$ such that all implications in $\mathcal{L}$ are of the form $U \to U''$, then the set

$$\mathcal{B}_2 := \{ \bigsqcap U \sqsubseteq (\bigsqcap U)^{\mathcal{I}\mathcal{I}} \mid (U \to U'') \in \mathcal{L} \}$$

is a base of $\mathcal{I}$ [7, Theorem 5.12].

However, the base $\mathcal{B}_2$ contains some redundancies, as the base $\mathcal{L}$ of $\mathbb{K}_\mathcal{I}$ has to entail all implications $\{ C \} \to \{ D \}$, where $C \sqsubseteq D$, $C, D \in M_\mathcal{I}$. Although these implications are non-trivial in $\mathbb{K}_\mathcal{I}$, the resulting GCIs $C \sqsubseteq D$ are trivially true in any interpretation, and thus do not need to be contained in $\mathcal{B}_2$. We can circumvent this flaw by introducing the set

$$\mathcal{S}_\mathcal{I} := \{ \{ C \} \to \{ D \} \mid C, D \in M_\mathcal{I}, C \sqsubseteq D \}$$

and computing bases $\mathcal{L}$ of $\mathbb{K}_\mathcal{I}$ with background-knowledge $\mathcal{S}_\mathcal{I}$. Then again, if $\mathcal{L}$ is such a base, the set

$$\mathcal{B}_3 := \{ \bigsqcap U \sqsubseteq (\bigsqcap U)^{\mathcal{I}\mathcal{I}} \mid (U \to U'') \in \mathcal{L} \}$$

is a base of $\mathcal{I}$ [7, Theorem 5.12]. Furthermore, if $\mathcal{L}$ is a base of minimal cardinality (like the canonical base $\mathrm{Can}(\mathbb{K}_\mathcal{I})$ of $\mathbb{K}_\mathcal{I}$), then the resulting base $\mathcal{B}_3$ is also of minimal cardinality among all bases of $\mathcal{I}$ [7, Theorem 5.18].

4.2 Growing Sets of Attributes

Distel based his development of a model-exploration algorithm on the classical attribute exploration algorithm from formal concept analysis. The rough idea is to do attribute exploration in the induced context $\mathbb{K}_\mathcal{I}$ of $\mathcal{I}$. However, there are a number of substantial differences between just exploring a formal context and exploring the induced context $\mathbb{K}_\mathcal{I}$ of $\mathcal{I}$.

One of these differences is the fact that the set of attributes $M_\mathcal{I}$ of $\mathbb{K}_\mathcal{I}$ arises from the interpretation $\mathcal{I}$, which however is not known completely at the beginning of an exploration, as the expert may add counterexamples later on, and thus cannot be computed before the exploration starts. To handle this situation, Distel extends the axiomatization algorithm to allow a growing set of attributes during the axiomatization process. With this extension, he is able to extend the algorithm that computes bases of $\mathcal{I}$ in such a way that the elements of $M_\mathcal{I}$ are not computed in advance, but one by one during the process.

The argumentation for this algorithm starts by extending the classical formal concept analysis algorithm for computing the canonical base of a given formal context. For this we observe that this algorithm makes use of the next-closure algorithm, which in turn computes the necessary intents from starting segments with respect to the lectic order of the set of attributes of the formal context. Thus, the natural idea is now to just put new attributes at the end of the set of attributes. In this way, the algorithm would behave as if all attributes would have been present right from the beginning.

An algorithm that implements this idea is shown in Algorithm 3. Since the algorithm makes use of different formal contexts $\mathbb{K}_k$, we denote the derivations in these formal contexts by $''_k$, respectively. The algorithm starts with a given initial formal context $\mathbb{K}_0$ and background knowledge $\mathcal{S}_0$ and computes the pseudo-intents of the formal context as usual. However, in every iteration k we read in a new formal context $\mathbb{K}_{k+1}$ that extends the previous formal context $\mathbb{K}_k$ with new attributes, i. e. $M_k \subseteq M_{k+1}$ and $I_k = I_{k+1} \cap M_k \times M_k$. Moreover, we also allow the set $\mathcal{S}_k$ of background knowledge to grow during the exploration.

Note, however, that the extension of the formal contexts during the computation cannot be unlimited, i. e. at a certain point ℓ, we need to have $M_k = M_\ell$ for all remaining iterations $k \geqslant \ell$. If this is the case, then we can not only guarantee the termination of Algorithm 3 (since from this point on, the computation behaves similarly to the usual computation of the canonical base), but also that if n is the final iteration of the algorithm, that then $\mathcal{L}_n$ is a base of $\mathbb{K}_n$ with background knowledge $\mathcal{S}_n$.

Algorithm 3 (Algorithm 8 from [7]) Computing a Base of a Formal Context with Growing Sets of Attributes and Background Knowledge

```
0  define base/growing-set-of-attributes(𝕂₀, 𝒮₀)
1    ;; initialization
2    k := 0
3    P₀ := ∅
4    ℒₖ := ∅
5
6    ;; computation
7    while Pₖ ≠ null do
8      ;; add new attributes
9      read 𝕂ₖ₊₁ = (G, Mₖ₊₁, Iₖ₊₁) where Mₖ ⊆ Mₖ₊₁ and Iₖ = Iₖ₊₁ ∩ Mₖ × Mₖ
10     read 𝒮ₖ₊₁ where 𝒮ₖ ⊆ 𝒮ₖ₊₁ ⊆ Th(𝕂ₖ₊₁)
11
12     ;; update ℒₖ₊₁
13     ℒₖ₊₁ := { Pᵣ → Pᵣ^{″ₖ₊₁} | Pᵣ ≠ Pᵣ^{″ₖ₊₁}, r ∈ {0, ..., k} }
14
15     ;; next closed set
16     if Pₖ = Mₖ = Mₖ₊₁ then
17       Pₖ₊₁ := null
18     else
19       Pₖ₊₁ := next-closed-set(Mₖ₊₁, <, Pₖ, ℒₖ₊₁ ∪ 𝒮ₖ₊₁)
20     end
21     k := k + 1
22   end
23
24   ;; return result
25   return ℒₖ
26 end
```

$\mathbb{K}_0 = \mathbb{K}_1 =$

	A
1	×
2	

$\mathbb{K}_2 = \mathbb{K}_3 =$

	A	B
1	×	
2		×

$\mathbb{K}_3 = \mathbb{K}_4 = \mathbb{K}_5 =$

	A	B	C
1	×		×
2		×	

Figure 1: Formal Contexts for Example 4.4

4.3 Theorem (Theorems 6.2 and 6.4 from [7]) *In a run of Algorithm 3, let $\ell \in \mathbb{N}$ be such that $M_k = M_\ell$ is true for all iterations $k \geqslant \ell$. Then Algorithm 3 terminates. If n is the last iteration of this algorithm, then $\mathcal{L}_n$ is a base of $\mathbb{K}_n$ with background knowledge $\mathcal{S}_n$.*

Note that in contrast to the classical computation of bases of formal contexts, we cannot discard intents P_k we encounter during our computation. This is because these sets are only known to be intents of the current context $\mathbb{K}_k$, and it might very well happen that in a later iteration ℓ the set of attributes is extended in a way such that P_k is not an intent of $\mathbb{K}_\ell$ anymore. In this case, the existence of P_k is crucial to guarantee that the algorithm computes a base of the final formal context $\mathbb{K}_n$.

Unfortunately, the fact that we have to keep all the sets P_k has as a consequence that the resulting bases $\mathcal{L}_n$ are not necessarily irredundant. This is shown by the following example.

4.4 Example (Example 6.1 from [7]) We consider the following run of Algorithm 3 with input $\mathbb{K}_0$ and $\mathcal{S}_0 = \varnothing = \mathcal{S}_1 = \ldots = \mathcal{S}_6$:

k	$M_{k+1} \backslash M_k$	$\mathcal{L}_k$	P_k
0	$\varnothing$	$\varnothing$	$\varnothing$
1	$\varnothing$	$\varnothing$	$\{\,\mathrm{A}\,\}$
2	$\{\,\mathrm{B}\,\}$	$\varnothing$	$\{\,\mathrm{B}\,\}$
3	$\varnothing$	$\varnothing$	$\{\,\mathrm{A},\mathrm{B}\,\}$
4	$\{\,\mathrm{C}\,\}$	$\{\{\,\mathrm{A}\,\} \to \{\,\mathrm{A},\mathrm{C}\,\}, \{\,\mathrm{A},\mathrm{B}\,\} \to \{\,\mathrm{A},\mathrm{B},\mathrm{C}\,\}\}$	$\{\,\mathrm{C}\,\}$
5	$\varnothing$	$\{\{\,\mathrm{A}\,\} \to \{\,\mathrm{A},\mathrm{C}\,\}, \{\,\mathrm{A},\mathrm{B}\,\} \to \{\,\mathrm{A},\mathrm{B},\mathrm{C}\,\}, \{\,\mathrm{C}\,\} \to \{\,\mathrm{A},\mathrm{C}\,\}\}$	$\{\,\mathrm{A},\mathrm{B},\mathrm{C}\,\}$
6	$\varnothing$	$\{\{\,\mathrm{A}\,\} \to \{\,\mathrm{A},\mathrm{C}\,\}, \{\,\mathrm{A},\mathrm{B}\,\} \to \{\,\mathrm{A},\mathrm{B},\mathrm{C}\,\}, \{\,\mathrm{C}\,\} \to \{\,\mathrm{A},\mathrm{C}\,\}\}$	null

In iterations 2 and 4, the new attributes B and C are added, as shown in Figure 1. The algorithm terminates in iteration 6 with output $\mathcal{L}_6$, which is clearly non-redundant: the implication $\{\,\mathrm{A},\mathrm{B}\,\} \to \{\,\mathrm{A},\mathrm{B},\mathrm{C}\,\}$ is entailed by $\{\,\mathrm{A}\,\} \to \{\,\mathrm{A},\mathrm{C}\,\}$. ♢

4.3 Computing Bases in a Given A-Priori Model

We have seen how to utilize the fact that we can use the next-closure algorithm for computing bases to allow for attribute sets which grow over time. In this section, we shall discuss how we can apply this utilization to our setting of computing bases of finite interpretations $\mathcal{I}$. Recall that one of our main problems in adapting classical attribute exploration to finite interpretations was the fact that we do not know the complete interpretation $\mathcal{I}$ in advance, as the expert may add counterexamples during the exploration. Therefore, we are not able to compute the set $M_\mathcal{I}$ right away, and instead can only use the parts of the interpretation we already know. To remedy this, we use the algorithm from the previous section, and compute the set $M_\mathcal{I}$ incrementally. The main result is then that the resulting base delivered by the algorithm yields a base of the final interpretation $\mathcal{I}$ in a canonical way.

As a first step into this direction, we assume in this section that the interpretation $\mathcal{I}$ is given completely, but that we add elements of $M_\mathcal{I}$ successively during the computation. In the next

section we shall see how we can relax the condition on knowing $\mathcal{I}$, namely by replacing it with expert interaction, yielding the desired exploration algorithm for finite interpretations.

Adding elements of $M_{\mathcal{I}}$ while the algorithm is running is achieved in the following way: we start with the set

$$M_0 := N_C \cup \{ \bot \}$$

and add elements from the set $\{ \exists r.X^{\mathcal{I}} \mid X \subseteq \Delta^{\mathcal{I}}, X \neq \varnothing \}$ on the fly. For this, whenever a new set P_k is computed, we add the concept expressions

$$\exists r.(\bigsqcap P_k)^{\mathcal{I}\mathcal{I}}, r \in N_R \tag{3}$$

to the set M_k to obtain the set M_{k+1}. More specifically, we only add those concept descriptions of the above form for which there do not already exist equivalent concept descriptions in M_k. In other words, we add only those concept descriptions which are not contained in M_k *up to equivalence*. We shall denote this with the same symbol $\cup$ since there is no danger of confusion.

The above idea to add only concept descriptions of the form $\exists r.(\bigsqcap P_k)^{\mathcal{I}\mathcal{I}}$ is motivated by the following fact. Obviously, all concept descriptions of the form (3) are elements of $M_{\mathcal{I}}$. On the other hand, if we consider a concept description $\exists r.X^{\mathcal{I}} \in M_{\mathcal{I}}$, then $X^{\mathcal{I}} \equiv X^{\mathcal{I}\mathcal{I}\mathcal{I}} = (X^{\mathcal{I}})^{\mathcal{I}\mathcal{I}}$, and $X^{\mathcal{I}}$ is expressible in terms of $M_{\mathcal{I}}$. Therefore, there exists a set $U \subseteq M_{\mathcal{I}}$ such that

$$X^{\mathcal{I}} \equiv \bigsqcap U.$$

It can now be shown that $(\bigsqcap U^{\prime\prime_n})^{\mathcal{I}} \equiv (\bigsqcap U)^{\mathcal{I}}$, where $\cdot^{\prime\prime_n}$ denotes the derivation in the induced context of M_n and $\mathcal{I}$, and where n is the last iteration of the algorithm [7, Lemma 4.10]. This implies that

$$\exists r.X^{\mathcal{I}} \equiv \exists r.X^{\mathcal{I}\mathcal{I}\mathcal{I}} \equiv \exists r.(\bigsqcap U^{\prime\prime_n})^{\mathcal{I}\mathcal{I}}.$$

Therefore, it suffices to consider only intents $U^{\prime\prime_n}$ of the current final context $\mathbb{K}_n$. Another result by Distel shows that all those intents are among the sets P_k.

4.5 Lemma (partly Lemma 6.3 from [7]) *Consider a terminating run of Algorithm 3 with n iterations. Let $Q \subseteq M_n$ such that $Q = Q^{\prime\prime_n}$. Then $Q = P_k$ for some $k \in \{ 0, \ldots, n \}$.*

This motivates the idea that the concept descriptions given in (3) may be sufficient. Therefore, we instantiate Algorithm 3 for the following setting:

- the sets M_{k+1} are defined as already discussed, i. e. $M_{k+1} = M_k \cup \{ \exists r.(\bigsqcap P_k)^{\mathcal{I}} \mid r \in N_R \}$,
- the formal context $\mathbb{K}_{k+1}$ will be the induced context of M_{k+1} and $\mathcal{I}$, and
- the background knowledge will be $\mathcal{S}_{k+1} = \{ \{ A \} \to \{ B \} \mid A, B \in M_{k+1}, A \sqsubseteq B \}$.

This instantiation yields Algorithm 4.

Note that since Algorithm 4 is a special case of Algorithm 3, we can easily argue that it will terminate for any finite interpretation $\mathcal{I}$ as input. This is because all elements which are added as new attributes during the run of the algorithm are, up to equivalence, elements of the set $M_{\mathcal{I}}$, which is finite. Therefore, there exists an iteration $\ell \in \mathbb{N}$ such that for all $k \geqslant \ell$ it is true that $M_k = M_\ell$. Therefore, the algorithm has to terminate.

For the correctness of the algorithm we just note that the rather intuitive argumentation we have provided before can be transformed into a proper proof, showing the following result.

Algorithm 4 (Algorithm 9 from [7]) Computing a Base of an A-Priori Given Model

0 **define** `base/a-priori-model`($\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$)
1 *;; initialization*
2 $k := 0$
3 $P_k := \varnothing$
4 $M_k := N_C \cup \{ \bot \}$
5 $\mathcal{L}_k := \varnothing$
6 $\mathcal{S}_k := \{ \{ \bot \} \to \{ A \} \mid A \in N_C \}$
7
8 *;; computation*
9 **while** $P_k \neq$ null **do**
10 *;; add new attributes (up to equivalence)*
11 $M_{k+1} := M_k \cup \{ \exists r.(\bigsqcap P_k)^{\mathcal{I}\mathcal{I}} \mid r \in N_R \}$
12
13 *;; update* $\mathbb{K}_{k+1}, \mathcal{S}_{k+1}$ *and* $\mathcal{L}_{k+1}$
14 $\mathbb{K}_{k+1} :=$ `induced-context`$(\mathcal{I}, M_{k+1})$
15 $\mathcal{L}_{k+1} := \{ P_r \to P_r^{\prime\prime_{k+1}} \mid P_r \neq P_r^{\prime\prime_{k+1}}, r \in \{0, \ldots, k\} \}$
16 $\mathcal{S}_{k+1} := \{ \{ A \} \to \{ B \} \mid A, B \in M_{k+1}, A \sqsubseteq B \}$
17
18 *;; next closed set*
19 **if** $P_k = M_k = M_{k+1}$ **then**
20 $P_{k+1} :=$ null
21 **else**
22 $P_{k+1} :=$ `next-closed-set`$(M_{k+1}, <, P_k, \mathcal{L}_{k+1} \cup \mathcal{S}_{k+1})$
23 **end**
24 $k := k + 1$
25 **end**
26
27 *;; return result*
28 **return** $\{ \bigsqcap P \sqsubseteq (\bigsqcap P)^{\mathcal{I}\mathcal{I}} \mid (P \to P^{\prime\prime_k}) \in \mathcal{L}_k \}$
29 **end**

4.6 Theorem (Theorem 6.9 from [7]) *Let $\mathcal{I}$ be a finite interpretation and let n be the total number of iterations of Algorithm 4. Then the set*

$$\{ \textstyle\bigsqcap P \sqsubseteq (\bigsqcap P)^{\mathcal{I}\mathcal{I}} \mid (P \to P''_n) \in \mathcal{L}_n \}$$

is a base of $\mathcal{I}$.

The main idea to obtain a proof of this result is to show that the resulting set M_n of attributes is equal, up to equivalence, to the set $M_{\mathcal{I}}$. It can be shown [7, Corollary 5.14] that in this case bases of the induced context of M_n and $\mathcal{I}$ yield bases of $\mathcal{I}$, in a very similar way as described in the above theorem.

4.4 An Algorithm for Exploring Interpretations

We have promised that in this last section we shall develop an exploration algorithm based on Algorithm 4. Indeed, a close inspection of the algorithm reveals that the only dependence to the background interpretation $\mathcal{I}_{\text{back}}$, which is not available during exploration, is for the computation of concept expressions of the form $\exists r.(\bigsqcap P_k)^{\mathcal{I}_{\text{back}}\mathcal{I}_{\text{back}}}$. A crucial observation to turn Algorithm 4 into an exploration algorithm is now that we can compute this model-based most-specific concept description using only *parts* of the background interpretation $\mathcal{I}_{\text{back}}$, provided that an external expert confirms a certain GCI. Therefore, we can lift the requirement of knowing the complete background interpretation $\mathcal{I}_{\text{back}}$ if we allow for expert interaction.

However, allowing expert interaction is not enough. As we have already mentioned, attribute exploration relies on the fact that the expert provides *counterexamples* for invalid GCIs. In the setting of exploring finite interpretations, these counterexamples are provided as parts of the interpretation. The difficulty that arises here is that interpretations employ *closed world semantics* – concept names or role successors which are not present in the interpretation are assumed to also not exist in the background interpretation. Therefore, if we want to provide counterexamples which stem from parts of an interpretation, we have to ensure that, while adding these counterexamples to our known interpretation, *all* its valid concept names as well as all its role successors and its concept names etc. are added as well. The notion to formalize this is the one of *connected subinterpretations*, which we shall introduce shortly.

So, the two main difficulties we have to deal with when extending Algorithm 4 to an exploration algorithm for interpretations are

- computing model-based most-specific concept descriptions $\exists r.(\bigsqcap P_k)^{\mathcal{I}_{\text{back}}\mathcal{I}_{\text{back}}}$ without the knowledge of the interpretation $\mathcal{I}_{\text{back}}$, and
- ensuring that the counterexamples provided by the expert are *complete* in the sense that they contain all the information contained in the background interpretation.

We start addressing these difficulties by discussing the second point first. For this, we introduce the notion of connected subinterpretations.

4.7 Definition (Definition 6.1 from [7]) Let $\mathcal{I}$ be an interpretation with concept names from N_C and role names from N_R. We define the following mappings

$$\begin{aligned} \text{names}_{\mathcal{I}}(x) &= \{ C \in N_C \mid x \in C^{\mathcal{I}} \}, \\ \text{succ}_{\mathcal{I}}(x, r) &= \{ y \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}} \} \end{aligned}$$

for $x \in \Delta^{\mathcal{I}}$ and $r \in N_R$.

An interpretation $\mathcal{J}$ with concept names form N_C and role names from N_R is called a *subinterpretation* of $\mathcal{I}$ if and only if

- $\Delta^{\mathcal{J}} \subseteq \Delta^{\mathcal{I}}$,
- $\text{names}_{\mathcal{I}}(x) = \text{names}_{\mathcal{J}}(x)$ for all $x \in \Delta^{\mathcal{J}}$, and
- $\text{succ}_{\mathcal{J}}(x, r) \subseteq \text{succ}_{\mathcal{I}}(x, r)$ for all $x \in \Delta^{\mathcal{J}}$ and $r \in N_R$.

The interpretation $\mathcal{J}$ is called a *connected subinterpretation* of $\mathcal{I}$ if $\mathcal{J}$ is a subinterpretation of $\mathcal{I}$ and in addition it is true that

$$\text{succ}_{\mathcal{J}}(x, r) = \text{succ}_{\mathcal{I}}(x, r)$$

for all $x \in \Delta^{\mathcal{J}}$ and $r \in N_R$. In this case we shall say that $\mathcal{I}$ *extends* $\mathcal{J}$. ♢

The main difficulty when adding counterexamples lies in the fact that these new counterexamples must not violate any GCI valid in the background interpretation. To avoid this, Distel restricts the way the expert can add counterexamples in the sense that, if $\mathcal{I}_\ell$ is the current working interpretation and the expert wants to add counterexamples for a proposed GCI, that the new interpretation $\mathcal{I}_{\ell+1}$ has to *extend* $\mathcal{I}_\ell$. That this is sufficient is shown in the following results.

4.8 Lemma (Lemma 6.12 from [7]) *Let $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ be a connected subinterpretation of $\mathcal{I}$. For every $\mathcal{EL}^{\perp}_{gfp}$ concept description C it is true that*

$$C^{\mathcal{J}} = C^{\mathcal{I}} \cap \Delta^{\mathcal{J}}.$$

This lemma has the immediate consequence that GCIs valid in interpretations are also valid in connected subinterpretations.

4.9 Theorem (Corollary 6.13 from [7]) *Let $\mathcal{I}$ be an interpretation, $\mathcal{J}$ a connected subinterpretation of $\mathcal{I}$ and let $C \sqsubseteq D$ be an $\mathcal{EL}^{\perp}_{gfp}$ GCI. Then if $C \sqsubseteq D$ is valid in $\mathcal{I}$, it is also valid in $\mathcal{J}$.*

Proof Since $C \sqsubseteq D$ holds in $\mathcal{I}$, it is true that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Using Lemma 4.8, we immediately obtain

$$C^{\mathcal{J}} = C^{\mathcal{I}} \cap \Delta^{\mathcal{J}} \subseteq D^{\mathcal{I}} \cap \Delta^{\mathcal{J}} = D^{\mathcal{J}},$$

therefore, $C \sqsubseteq D$ is valid in $\mathcal{J}$ as well. ▫

To sum up, if we extend the working interpretation by counterexamples such that we always obtain connected subinterpretations of the background interpretation, then we do not introduce unwanted counterexamples to valid GCIs.

We now turn our attention to the problem of computing model-based most-specific concept descriptions of the form $\exists r.(\bigsqcap P_k)^{\mathcal{I}_{\text{back}}\mathcal{I}_{\text{back}}}$ without complete knowledge of the actual background interpretation $\mathcal{I}_{\text{back}}$. The moment we want to compute this model-based most-specific concept description in Algorithm 4 is in iteration k, thus obviously what we can compute is the concept description $\exists r.(\bigsqcap P_k)^{\mathcal{I}_k\mathcal{I}_k}$. However, being able to compute this concept description does not help if we cannot guarantee the equivalence of this concept description to $\exists r.(\bigsqcap P_k)^{\mathcal{I}_{\text{back}}\mathcal{I}_{\text{back}}}$. Fortunately, this guarantee can be achieve with expert interaction.

4.10 Lemma (Lemma 6.14 from [7]) *Let $\mathcal{J}$ be a connect subinterpretation of $\mathcal{I}$, and let C be an $\mathcal{EL}^{\perp}_{gfp}$ concept description. Then if $C \sqsubseteq C^{\mathcal{J}\mathcal{J}}$ is valid in $\mathcal{I}$, then $C^{\mathcal{I}\mathcal{I}} \equiv C^{\mathcal{J}\mathcal{J}}$.*

Algorithm 5 (Algorithm 11 from [7]) An Exploration Algorithm for Interpretations

define `model-exploration`($\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$)
  ;; initialization
  $k := 0$
  $P_k := \varnothing$
  $M_k := N_C \cup \{ \bot \}$
  $\mathcal{L}_k := \varnothing$
  $\mathcal{S}_k := \{ \{ \bot \} \to \{ A \} \mid A \in N_C \}$
  $l := 0$
  $\mathcal{I}_l := \mathcal{I}$

  ;; computation
  while $P_k \neq$ null **do**
    ;; expert interaction
    while expert refutes $\bigcap P_k \sqsubseteq (\bigcap P_k)^{\mathcal{I}_l \mathcal{I}_l}$ **do**
      $\mathcal{I}_{l+1} :=$ new interpretation such that
        – $\mathcal{I}_{l+1}$ extends $\mathcal{I}_l$
        – $\mathcal{I}_{l+1}$ contains counterexamples for $\bigcap P_k \sqsubseteq (\bigcap P_k)^{\mathcal{I}_l \mathcal{I}_l}$.
      $l := l + 1$
    end

    ;; add new attributes (up to equivalence)
    $M_{k+1} := M_k \cup \{ \exists r.(\bigcap P_k)^{\mathcal{I}\mathcal{I}} \mid r \in N_R \}$

    ;; update $\mathbb{K}_{k+1}, \mathcal{S}_{k+1}$ *and* $\mathcal{L}_{k+1}$
    $\mathbb{K}_{k+1} :=$ `induced-context`($\mathcal{I}$, M_{k+1})
    $\mathcal{L}_{k+1} := \{ P_r \to P_r^{\prime\prime k+1} \mid P_r \neq P_r^{\prime\prime k+1}, r \in \{ 0, \ldots, k \} \}$
    $\mathcal{S}_{k+1} := \{ \{ A \} \to \{ B \} \mid A, B \in M_{k+1}, A \sqsubseteq B \}$

    ;; next closed set
    if $P_k = M_k = M_{k+1}$ **then**
      $P_{k+1} :=$ null
    else
      $P_{k+1} :=$ `next-closed-set`(M_{k+1}, $<$, P_k, $\mathcal{L}_{k+1} \cup \mathcal{S}_{k+1}$)
    end
    $k := k + 1$
  end

  ;; return result
  return $\{ \bigcap P \sqsubseteq (\bigcap P)^{\mathcal{I}\mathcal{I}} \mid (P \to P^{\prime\prime k}) \in \mathcal{L}_k \}$
end

Using the expert, we can check whether $C \sqsubseteq C^{\mathcal{I}_k\mathcal{I}_k}$ is valid in the background interpretation or not. Using this fact together with our discussion on adding counterexamples, we are able to extend Algorithm 4 to our desired model exploration algorithm as shown in Algorithm 5.

Note that the set $M_{\mathcal{I}_{\text{back}}}$ is finite, as our background interpretation is supposed to be finite. Therefore, there are only finitely many possible values for sets P_k, thus there eventually must exist a $k \in \mathbb{N}$ such that $P_k = \text{null}$, and the algorithm must terminate. Hence, the algorithm terminates for every valid input.

Furthermore, as expected, upon termination of Algorithm 5, the set of implications collected so far in the set $\mathcal{L}_k$ gives rise to a set of GCIs in the usual way that is an axiomatization of the background interpretation $\mathcal{I}_{\text{back}}$.

4.11 Theorem (Theorem 6.16 of [7]) *Suppose a run of Algorithm 5 with background interpretation $\mathcal{I}_{\text{back}}$, and suppose that the algorithm terminates in the n-th iteration with $\mathcal{I}_\ell$ as final working interpretation. Then the set*

$$\{ \bigcap P \to (\bigcap P)^{\mathcal{I}_\ell\mathcal{I}_\ell} \mid (P \to P''_n) \in \mathcal{L}_n \}$$

is an axiomatization of $\mathcal{I}_{\text{back}}$.

5 Model Exploration with Confident GCIs

In Section 3, we have sketched on a very abstract level how a model exploration with confident GCIs should look like. Furthermore, we have seen in sufficient detail in the previous Section 4 how such a model exploration can be achieved in the case of considering only *valid* GCIs, i. e. for the case of confidence equal to 1.

The purpose of this part of the present work is to extend the argumentation of Distel to the case of confident GCIs. To this end, we shall devise an algorithm that allows an expert to explore the confident theory of a given finite interpretation as described in Section 3. The argumentation we want to use for this is very similar to the one used in the previous section: we shall first investigate in Section 5.2 the case of confident implications and how they can be axiomatized in the presence of a growing set of attributes. Thereafter, we extend in Section 5.3 the obtained results to GCIs in the case where the background interpretation $\mathcal{I}_{\text{back}}$ is completely known, but the set $M_{\mathcal{I}_{\text{back}}}$ is computed not advance, but during the axiomatization. Finally, we show in Section 5.4 how expert interaction can be used to relax the condition that the background interpretation is known beforehand.

There is, however, a difference in exploring confident GCIs that is not (yet) paralleled in the considerations for a model exploration for valid GCIs. The exploration in the latter case is based on an algorithm to axiomatize valid GCIs from a given finite interpretation. This fact is used in the way that a given initial working interpretation is extended by counterexamples in a way such that the original algorithm always "receives" relevant counterexamples when they are needed, i. e. they are added to the working interpretation when they are needed to invalidate a GCI.

Transferring this observation to the case of a model exploration for confident GCIs, we see that our counterexamples are handled in a different way. The main reason for this is that we do not demand our initial working interpretation to be a part of the background interpretation. Recall that what we want to do during the exploration process is that we want to axiomatize the confident GCIs of the initial working interpretation that are valid in the interpretation that is constituted by the counterexamples received so far. We then can regard the initial working interpretation as a set of *untrusted* individuals, and the set of counterexamples provided by the expert as a set of *trusted* individuals. Then, if we would have an algorithm which allows

us to axiomatize all GCIs that are valid for all trusted individuals, and have confidence high enough in the set of trusted individuals, we could think of a model exploration for confident GCIs as an extensions of such an algorithm that "receives" the relevant counterexamples from the background interpretation when they are needed.

Finding an algorithm that allows for computing axiomatizations in the presence of trusted and untrusted individuals is not hard, and we shall present such an algorithm in Section 5.1, before our actual considerations regarding a model exploration for confident GCIs.

5.1 Trusted and Untrusted Individuals

Let $\mathcal{I}$ be an interpretation, and let $\mathcal{J}$ be subinterpretation of $\mathcal{I}$. In our course of devising an exploration algorithm for confident GCIs, we may think of the interpretation $\mathcal{J}$ as the initial working interpretation, and of the interpretation

$$\mathcal{I}\backslash\mathcal{J} := (\Delta^{\mathcal{I}}\backslash\Delta^{\mathcal{J}}, \cdot^{\mathcal{I}\backslash\mathcal{J}})$$

as the interpretation which is constituted by all counterexamples provided by the expert, where

$$\begin{aligned} A^{\mathcal{I}\backslash\mathcal{J}} &:= A^{\mathcal{I}} \cap \Delta^{\mathcal{I}}\backslash\Delta^{\mathcal{J}} = A^{\mathcal{I}}\backslash\Delta^{\mathcal{J}}, \\ r^{\mathcal{I}\backslash\mathcal{J}} &:= r^{\mathcal{I}} \cap (\Delta^{\mathcal{I}}\backslash\Delta^{\mathcal{J}}) \times (\Delta^{\mathcal{I}}\backslash\Delta^{\mathcal{J}}) \end{aligned}$$

for $A \in N_C, r \in N_R$. With respect to this understanding, we shall call $\mathcal{J}$ the interpretation of *untrusted* individuals, and $\mathcal{I}\backslash\mathcal{J}$ the interpretation of *trusted* individuals.

The aim of this section is to devise an algorithm that allows us to axiomatize all GCIs that are valid for all trusted individuals, and whose confidence in the untrusted individuals is above a predefined threshold. More precisely, let us define for $c \in [0, 1]$ the set

$$\mathrm{Th}_c(\mathcal{I}, \mathcal{J}) := \{\, C \sqsubseteq D \mid C^{\mathcal{I}}\backslash\Delta^{\mathcal{J}} \subseteq D^{\mathcal{I}}\backslash\Delta^{\mathcal{J}} \text{ and } |(C \sqcap D)^{\mathcal{I}} \cap \Delta^{\mathcal{J}}| \geqslant c \cdot |C^{\mathcal{I}} \cap \Delta^{\mathcal{J}}| \,\}$$

What we are seeking for now is an algorithm that computes a finite base of the set $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})$. The results of this section have partly been published in [2].

Before we start with our actual considerations, observe that in the definition of $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})$ we have given the constraint on the confidence of a GCI $C \sqsubseteq D$ by stating

$$|(C \sqcap D)^{\mathcal{I}} \cap \Delta^{\mathcal{J}}| \geqslant c \cdot |C^{\mathcal{I}} \cap \Delta^{\mathcal{J}}|. \tag{4}$$

However, as in the case of model exploration with valid GCIs, it will turn out that the counterexamples provided by the expert must be given by connected subinterpretations of the background interpretation. In particular this means that both interpretations $\mathcal{J}$ and $\mathcal{I}\backslash\mathcal{J}$ are connected subinterpretations of $\mathcal{I}$ (i. e. there are no roles between individuals from $\mathcal{J}$ and $\mathcal{I}\backslash\mathcal{J}$).

This can be used to simplify the constraints on the confidence of $C \sqsubseteq D$ a bit. More precisely, recall that from Lemma 4.8 it now follows that

$$\begin{aligned} C^{\mathcal{I}} \cap \Delta^{\mathcal{J}} &= C^{\mathcal{J}} \\ (C \sqcap D)^{\mathcal{I}} \cap \Delta^{\mathcal{J}} &= (C \sqcap D)^{\mathcal{J}}. \end{aligned}$$

Therefore, further observing that for $c \in [0, 1]$, $|(C \sqcap D)^{\mathcal{J}}| \geqslant c \cdot |C^{\mathcal{J}}|$ is equivalent to $\mathrm{conf}_{\mathcal{J}}(C \sqsubseteq D) \geqslant c$, we can simplify Equation 4 to

$$\begin{aligned} \mathrm{Th}_c(\mathcal{I}, \mathcal{J}) &:= \{\, C \sqsubseteq D \mid C^{\mathcal{I}}\backslash\Delta^{\mathcal{J}} \subseteq D^{\mathcal{I}}\backslash\Delta^{\mathcal{J}} \text{ and } \mathrm{conf}_{\mathcal{J}}(C \sqsubseteq D) \geqslant c \,\} \\ &= \mathrm{Th}(\mathcal{I}\backslash\mathcal{J}) \cap \mathrm{Th}_c(\mathcal{J}). \end{aligned}$$

The last equation can be seen as follows: observe if $\mathcal{J}$ is a connected subinterpretation of $\mathcal{I}$, then

$$r^{\mathcal{I}\backslash\mathcal{J}} = r^{\mathcal{I}}\backslash(\Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}})$$

is true for all $r \in N_R$. Then $\mathcal{I}\backslash\mathcal{J}$ is also a connected subinterpretation of $\mathcal{I}$, and thus we obtain for all concept descriptions C

$$C^{\mathcal{I}\backslash\mathcal{J}} = C^{\mathcal{I}} \cap (\Delta^{\mathcal{I}}\backslash\Delta^{\mathcal{J}}) = C^{\mathcal{I}}\backslash\Delta^{\mathcal{J}}.$$

Thus, $C^{\mathcal{I}}\backslash\Delta^{\mathcal{J}} \subseteq D^{\mathcal{I}}\backslash\Delta^{\mathcal{J}}$ is true if and only if $C^{\mathcal{I}\backslash\mathcal{J}} \subseteq D^{\mathcal{I}\backslash\mathcal{J}}$ is, i. e. if $(C \sqsubseteq D) \in \mathrm{Th}(\mathcal{I}\backslash\mathcal{J})$.

To find a base for the set $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})$, we again make use of ideas we have already used to construct bases of all confident GCIs of finite interpretations. To this end, we observe that

$$\mathrm{Th}(\mathcal{I}) \subseteq \mathrm{Th}_c(\mathcal{I}, \mathcal{J}).$$

Therefore, since we can find bases of $\mathrm{Th}(\mathcal{I})$, it suffices to find bases for the "rest" $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})\backslash\,\mathrm{Th}(\mathcal{I})$. More precisely, if we consider the base $\mathcal{B}_2$ from Section 4.1, it suffices to find a complete subset $\mathcal{C} \subseteq \mathrm{Th}_c(\mathcal{I}, \mathcal{J})\backslash\,\mathrm{Th}(\mathcal{I})$ to make the set

$$\mathcal{B}_2 \cup \mathcal{C}$$

a base of $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})$.

Therefore, we shall concentrate in the remainder of this section on finding complete subsets $\mathcal{C}$ of $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})\backslash\,\mathrm{Th}(\mathcal{J})$. For this, we first observe that

$$(C \sqsubseteq D) \in \mathrm{Th}_c(\mathcal{I}, \mathcal{J}) \iff (C^{\mathcal{I}\mathcal{I}} \sqsubseteq D^{\mathcal{I}\mathcal{I}}) \in \mathrm{Th}_c(\mathcal{I}, \mathcal{J}). \tag{5}$$

This is because for all $\mathcal{EL}^{\bot}_{\mathrm{gfp}}$ concept descriptions E, F it is true that

$$E^{\mathcal{I}\mathcal{I}\mathcal{I}} = E^{\mathcal{I}}$$
$$(E^{\mathcal{I}\mathcal{I}} \sqcap F^{\mathcal{I}\mathcal{I}})^{\mathcal{I}} = (E \sqcap F)^{\mathcal{I}}.$$

Consequently, we obtain

$$C^{\mathcal{I}}\backslash\Delta^{\mathcal{J}} \subseteq D^{\mathcal{I}}\backslash\Delta^{\mathcal{J}} \iff C^{\mathcal{I}\mathcal{I}\mathcal{I}}\backslash\Delta^{\mathcal{J}} \subseteq D^{\mathcal{I}\mathcal{I}\mathcal{I}} \subseteq \Delta^{\mathcal{J}},$$
$$\mathrm{conf}_{\mathcal{J}}(C \sqsubseteq D) = \mathrm{conf}_{\mathcal{J}}(C^{\mathcal{I}\mathcal{I}} \sqsubseteq D^{\mathcal{I}\mathcal{I}}),$$

which yields the validity of (5).

Furthermore, if $\mathcal{B}$ is a base of $\mathcal{I}$, then

$$\mathcal{B} \cup \{\, C^{\mathcal{I}\mathcal{I}} \sqsubseteq D^{\mathcal{I}\mathcal{I}} \,\} \models (C \sqsubseteq D)$$

for all $\mathcal{EL}^{\bot}_{\mathrm{gfp}}$ concept descriptions C, D. This is because $\mathcal{B} \models (C \sqsubseteq C^{\mathcal{I}\mathcal{I}})$, as $C \sqsubseteq C^{\mathcal{I}\mathcal{I}}$ is valid in $\mathcal{I}$, and $D^{\mathcal{I}\mathcal{I}} \sqsubseteq D$ is valid in all interpretations. Thus,

$$\mathcal{B} \cup \{\, C^{\mathcal{I}\mathcal{I}} \sqsubseteq D^{\mathcal{I}\mathcal{I}} \,\} \models (C \sqsubseteq C^{\mathcal{I}\mathcal{I}} \sqsubseteq D^{\mathcal{I}\mathcal{I}} \sqsubseteq D).$$

Having these two observations in mind, we define

$$\mathrm{Conf}(\mathcal{I}, c, \mathcal{J}) := \{\, C^{\mathcal{I}\mathcal{I}} \sqsubseteq D^{\mathcal{I}\mathcal{I}} \mid (C^{\mathcal{I}\mathcal{I}} \sqsubseteq D^{\mathcal{I}\mathcal{I}}) \in \mathrm{Th}_c(\mathcal{I}, \mathcal{J}) \,\}.$$

Since $\mathcal{I}$ is finite, the set of model-based most-specific concept descriptions is, up to equivalence, finite as well. Therefore, we immediately obtain the following result.

5.1 Theorem *Let $\mathcal{I}$ be a finite interpretation, $\mathcal{J}$ be a connected subinterpretation of $\mathcal{I}$ and let $c \in [0,1]$. If $\mathcal{B}$ is a finite base of $\mathcal{I}$, then*

$$\mathcal{B} \cup \mathrm{Conf}(\mathcal{I}, c, \mathcal{J})$$

is a finite base of $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})$.

This theorem already solves the initially stated problem of learning GCIs in the presence of trusted and untrusted individuals. However, in the light of our overall goal of devising a model exploration algorithm for confident GCIs, it would be preferable if we could reduce such an axiomatization to a method which is based on formal contexts. An adaption of the classical attribute exploration starting from such an algorithm should be much more promising.

Such a method is provided by the following theorem. In this theorem, let us denote with $\mathbb{K}_\mathcal{I}|_X$ for $X \subseteq \Delta^\mathcal{I}$ the formal context that arises from the induced context $\mathbb{K}_\mathcal{I} = (\Delta^\mathcal{I}, M_\mathcal{I}, \nabla)$ by restricting the set of objects $\Delta^\mathcal{I}$ to X. In other words,

$$\mathbb{K}_\mathcal{I}|_X := (X, M_\mathcal{I}, \nabla \cap X \times M_\mathcal{I}).$$

We start with an auxiliary lemma that provides a connection between entailment of implications and entailment of GCIs. To keep the notation simple, we shall use for a set M of concept descriptions and $\mathcal{L} \subseteq \mathrm{Imp}(M)$ of implications the abbreviation

$$\bigsqcap \mathcal{L} := \{ \bigsqcap X \sqsubseteq \bigsqcap Y \mid (X \to Y) \in \mathcal{L} \}.$$

5.2 Lemma *Let M be a set of concept descriptions, and let $\mathcal{L} \subseteq \mathrm{Imp}(M)$ and $(X \to Y) \in \mathrm{Imp}(M)$. Then $\mathcal{L} \models (X \to Y)$ implies $\bigsqcap \mathcal{L} \models (\bigsqcap X \sqsubseteq \bigsqcap Y)$.*

Proof Let $\mathcal{J} = (\Delta^\mathcal{J}, \cdot^\mathcal{J})$ be an interpretation such that $\mathcal{J} \models \bigsqcap \mathcal{L}$. Let $\mathbb{K}_{M,\mathcal{J}}$ be the induced formal context of M and $\mathcal{J}$.

We shall show now that $\mathbb{K}_{M,\mathcal{J}} \models \mathcal{L}$. Let $(E \to F) \in \mathcal{L}$. Then $(\bigsqcap E)^\mathcal{J} \subseteq (\bigsqcap F)^\mathcal{J}$, since $\mathcal{J} \models \bigsqcap \mathcal{L}$. From Lemma 4.1 we obtain $(\bigsqcap E)^\mathcal{J} = E'$, where the derivation is done in $\mathbb{K}_{M,\mathcal{J}}$. Therefore, it is true that $E' \subseteq F'$, and thus $\mathbb{K}_{M,\mathcal{J}} \models (E \to F)$ holds.

Since $\mathcal{L} \models (X \to Y)$, it is now true that $\mathbb{K}_{M,\mathcal{J}} \models (X \to Y)$, i. e. $X' \subseteq Y'$. As $(\bigsqcap X)^\mathcal{J} = X'$, we can infer $(\bigsqcap X)^\mathcal{J} \subseteq (\bigsqcap Y)^\mathcal{J}$, i. e. $\mathcal{J} \models (\bigsqcap X \sqsubseteq \bigsqcap Y)$. As $\mathcal{J}$ has been chosen arbitrarily, we obtain $\bigsqcap \mathcal{L} \models (\bigsqcap X \sqsubseteq \bigsqcap Y)$. ▫

We now provide a context-based approach to finding bases of confident GCIs in the presence of trusted individuals.

5.3 Theorem *Let $\mathcal{I}$ be a finite interpretation, let $\mathcal{J}$ be a connected subinterpretation of $\mathcal{I}$, and let $c \in [0,1]$. Let*

$$\mathcal{T} := \mathrm{Th}_c(\mathbb{K}_\mathcal{I}|_{\Delta^\mathcal{J}}) \cap \mathrm{Th}(\mathbb{K}_\mathcal{I}|_{\Delta^\mathcal{I} \setminus \Delta^\mathcal{J}})$$

and let $\mathcal{L} \subseteq \mathcal{T}$ be a complete subset of $\mathcal{T}$. Then $\bigsqcap \mathcal{L}$ is a finite base of $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})$.

The proof is an adaptation of the proof given in [2, Appendix A].

Proof We need to show two claims, namely

i. $\bigsqcap \mathcal{L} \subseteq \mathrm{Th}_c(\mathcal{I}, \mathcal{J})$ and

ii. $\bigsqcap \mathcal{L}$ is complete for $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})$.

For the first claim we need to show that for every $(\bigsqcap X \sqsubseteq \bigsqcap Y) \in \bigsqcap \mathcal{L}$ it is true that

i. $\mathrm{conf}_{\mathcal{J}}(\bigsqcap X \subseteq \bigsqcap Y) \geqslant c$

ii. $(\bigsqcap X)^{\mathcal{I}} \backslash \Delta^{\mathcal{J}} \subseteq (\bigsqcap Y)^{\mathcal{I}} \backslash \Delta^{\mathcal{J}}$

For the first subclaim, we observe that $\mathrm{conf}_{\mathbb{K}_{\mathcal{I}}|_{\Delta^{\mathcal{J}}}}(X \to Y) \geqslant c$ for all $(X \to Y) \in \mathcal{L}$, i. e.

$$|(X \cup Y)' \cap \Delta^{\mathcal{J}}| \geqslant c \cdot |X' \cap \Delta^{\mathcal{J}}|$$

Since $(\bigsqcap X)^{\mathcal{I}} = X'$ by Lemma 4.1, we obtain

$$|(\bigsqcap (X \cup Y))^{\mathcal{I}} \cap \Delta^{\mathcal{J}}| \geqslant c \cdot |(\bigsqcap X)^{\mathcal{I}} \cap \Delta^{\mathcal{J}}|,$$

and from $\bigsqcap (X \cup Y) \equiv \bigsqcap X \sqcap \bigsqcap Y$ it follows immediately that

$$|(\bigsqcap X \sqcap \bigsqcap Y)^{\mathcal{I}} \cap \Delta^{\mathcal{J}}| \geqslant c \cdot |(\bigsqcap X)^{\mathcal{I}} \cap \Delta^{\mathcal{J}}|.$$

Since $\mathcal{J}$ is a connected subinterpretation of $\mathcal{I}$, we can simply this fact to

$$|(\bigsqcap X \sqcap \bigsqcap Y)^{\mathcal{J}}| \geqslant c \cdot |(\bigsqcap X)^{\mathcal{J}}|,$$

which means nothing else but $\mathrm{conf}_{\mathcal{J}}(\bigsqcap X \sqsubseteq \bigsqcap Y) \geqslant c$.

For the second subclaim, we observe that $X' \backslash \Delta^{\mathcal{J}} \subseteq Y' \backslash \Delta^{\mathcal{J}}$, since $X \to Y$ is valid in the formal context $\mathbb{K}_{\mathcal{I}}|_{\Delta^{\mathcal{I}} \backslash \Delta^{\mathcal{J}}}$. Since $X' = (\bigsqcap X)^{\mathcal{I}}$ and $Y' = (\bigsqcap Y)^{\mathcal{I}}$ the claim follows.

We have therefore shown that $\bigsqcap \mathcal{L} \subseteq \mathrm{Th}_c(\mathcal{I}, \mathcal{J})$.

We now show that $\bigsqcap \mathcal{L}$ is complete for $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})$. To this end, we shall show that

i. $\bigsqcap \mathcal{L} \models (\bigsqcap U \sqsubseteq (\bigsqcap U)^{\mathcal{I}\mathcal{I}})$ for all $U \subseteq M_{\mathcal{I}}$; in particular $\bigsqcap \mathcal{L} \models \mathcal{B}_2$;

ii. $\bigsqcap \mathcal{L} \models \mathrm{Conf}(\mathcal{I}, c, \mathcal{J})$.

If we can establish these claims, then by Theorem 5.1 we obtain from $\bigsqcap \mathcal{L} \models \mathcal{B}_2 \cup \mathrm{Conf}(\mathcal{I}, c, \mathcal{J})$ the completeness of $\bigsqcap \mathcal{L}$ for $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})$.

Let $U \subseteq M_{\mathcal{I}}$. Since $\mathcal{L}$ entails all valid implications of $\mathbb{K}_{\mathcal{I}}$, we obtain

$$\mathcal{L} \models (U \to U'').$$

By Lemma 5.2, it follows that $\mathcal{L} \models (\bigsqcap U \sqsubseteq (\bigsqcap U''))$. Equation (2) implies $(\bigsqcap U'') \equiv (\bigsqcap U)^{\mathcal{I}\mathcal{I}}$, and we obtain the validity of the subclaim.

For the second subclaim, let $(C^{\mathcal{I}\mathcal{I}} \sqsubseteq D^{\mathcal{I}\mathcal{I}}) \in \mathrm{Conf}(\mathcal{I}, c, \mathcal{J})$. We define $U := C^{\mathcal{I}}, V := D^{\mathcal{I}}$. Then by Lemma 4.2 it is true that $U^{\mathcal{I}} \equiv \bigsqcap U'$ and $V^{\mathcal{I}} \equiv \bigsqcap V'$, so

$$\bigsqcap \mathcal{L} \models (U^{\mathcal{I}} \sqsubseteq V^{\mathcal{I}}) \iff \bigsqcap \mathcal{L} \models (\bigsqcap U' \sqsubseteq \bigsqcap V').$$

It thus suffices to show $\mathcal{L} \models (U' \to V')$. For this we recall that $\mathrm{conf}_{\mathcal{J}}(C^{\mathcal{I}\mathcal{I}} \sqsubseteq D^{\mathcal{I}\mathcal{I}}) \geqslant c$. Since $\mathcal{J}$ is a connected subinterpretation of $\mathcal{I}$, this is equivalent to

$$|(C^{\mathcal{I}\mathcal{I}} \sqcap D^{\mathcal{I}\mathcal{I}})^{\mathcal{I}} \cap \Delta^{\mathcal{J}}| \geqslant c \cdot |(C^{\mathcal{I}\mathcal{I}})^{\mathcal{I}} \cap \Delta^{\mathcal{J}}|$$

Now since $\bigsqcap U' \equiv U^{\mathcal{I}} \equiv C^{\mathcal{I}\mathcal{I}}$ and $\bigsqcap V' \equiv D^{\mathcal{I}\mathcal{I}}$, we obtain

$$|(\bigsqcap U' \sqcap \bigsqcap V')^{\mathcal{I}} \cap \Delta^{\mathcal{J}}| \geqslant c \cdot |(\bigsqcap U')^{\mathcal{I}} \cap \Delta^{\mathcal{J}}|$$

As shown before, this implies that

$$|(U' \cup V')' \cap \Delta^{\mathcal{J}}| \geqslant c \cdot |U'' \cap \Delta^{\mathcal{J}}|,$$

where the derivations are conducted in $\mathbb{K}_{\mathcal{I}}$. In other words, it is true that

$$\mathrm{conf}_{\mathbb{K}_{\mathcal{I}}|_{\Delta^{\mathcal{J}}}}(U' \to V') \geqslant c.$$

Thus, $\mathcal{L} \models (U' \to V')$, and Lemma 5.2 implies $\bigsqcap \mathcal{L} \models (\bigsqcap U' \sqsubseteq \bigsqcap V')$, thus $\bigsqcap \mathcal{L} \models (U^{\mathcal{I}} \sqsubseteq V^{\mathcal{I}}) = (C^{\mathcal{I}\mathcal{I}} \sqsubseteq D^{\mathcal{I}\mathcal{I}})$, as required. ▫

5.2 Axiomatizing Confident Implications with Growing Sets of Attributes

Theorem 5.3 establishes a new bridge between the worlds of formal concept analysis and description logics. More precisely, if we have given an algorithm that computes bases of sets of implications which are valid in one induced context $\mathbb{K}_{\mathcal{I}\setminus\mathcal{J}}$, and enjoy confidence constraints in another induced context $\mathbb{K}_{\mathcal{J}}$, then the resulting base can easily be transferred into a base of $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})$. Therefore, we obtain from such an algorithm working on formal contexts an algorithm that allows us to axiomatize GCIs in the presence of trusted and untrusted individuals.

The purpose of this section is to devise such an algorithm that allows us to axiomatize sets of implications which are given in the form of

$$\mathrm{Th}_c(\mathbb{K}_1) \cap \mathrm{Th}(\mathbb{K}_2) \tag{6}$$

where the attribute sets of $\mathbb{K}_1$ and $\mathbb{K}_2$ are the same. Given such an algorithm, we shall invoke Theorem 5.3 to convert it into an algorithm for axiomatizing GCIs in the presence of trusted and untrusted individuals. The details for this latter step are going to be worked out in the next section.

Note that this section corresponds to the line of argumentation given in Section 4.2. To complete the analogy we shall also consider the possibility here that the sets of attributes may grow during the computation, in the sense that they are not completely available a-priori. Recall that this honors that fact that, during an envisioned model exploration algorithm for confident GCIs, the interpretation of all counterexamples is not known completely, and thus the set $M_{\mathcal{I}}$ cannot be computed in advance, but instead has to be computed on the fly. Since the set $M_{\mathcal{I}}$ is used as the set of attributes in the induced contexts $\mathbb{K}_{\mathcal{I}\setminus\mathcal{J}}$ and $\mathbb{K}_{\mathcal{J}}$, we have to add the possibility of growing sets of attributes to the algorithms of this section.

Before we shall investigate how to obtain bases of (6), we shall start with a simpler problem first: Let $\mathbb{K} = (G, M, I)$ be a finite and non-empty formal context and let $c \in [0, 1]$. We then want to devise an algorithm to axiomatize the confident implications $\mathrm{Th}_c(\mathbb{K})$ of $\mathbb{K}$, where the set M is not given in advance, but is computed during the computation. Note that this is a special case of axiomatizing sets of implications of the form (6) by just choosing $\mathbb{K}_1 = \mathbb{K}$ and $\mathbb{K}_2 = (\varnothing, M, \varnothing)$.

Of course, we can obtain an axiomatization of $\mathrm{Th}_c(\mathbb{K})$ by simply taking the set $\mathrm{Th}_c(\mathbb{K})$ itself, but we aim here at finding a potentially much smaller set of implications which is sufficient to axiomatize $\mathrm{Th}_c(\mathbb{K})$. Moreover, as we seek to use this algorithm as a basis for our forthcoming model exploration algorithm, we would like to not compute implications which can be answered by already computed ones.

An algorithm that honors these criteria can be achieved by suitable adaption of the classical algorithm to compute the canonical base of a formal context, i. e. of Algorithm 2. Recall that in this algorithm, we successively compute sets P of attributes in lectic order that are closed under the already known implications, but are not intents of the context. If such sets are found, the implication $P \to P''$ is added, and the computation stops if no such sets P exist.

It is not difficult to adapt this algorithmic idea to the case of growing sets of attributes, and indeed, we have already seen how this can be done in Algorithm 3. The main idea to transfer this algorithm to the setting of confident implications is to change the closure operator $(\cdot)''$ to an operator that reflects that fact that we are also considering implications whose confidence is not 1. A natural attempt would be to use the closure operator induced by $\mathrm{Th}_c(\mathbb{K})$. However, computing closures under this operator may be practically infeasible (see [4] for a discussion on this.)

To remedy this, we use another approach discussed in [4]. To this end, let $P \subseteq M$. We define

$$P^{\mathbb{K},c} := \{\, m \in M \mid \mathrm{conf}_{\mathbb{K}}(P \to \{\, m \,\}) \geqslant c \,\}.$$

It is imminent from this definition that $P'' \subseteq P^{\mathbb{K},c}$ holds, and that the mapping $(\cdot)^{\mathbb{K},c}$ is extensive. But it is also not hard to see that in general $(\cdot)^{\mathbb{K},c}$ is not monotone and that $P^{\mathbb{K},c} \neq (P^{\mathbb{K},c})^{\mathbb{K},c}$. Because of this, we have to add some more changes to the part of the algorithm that successively computes new implications, to ensure that the resulting set of implications is indeed complete for $\mathrm{Th}_c(\mathbb{K})$. These changes mainly affect the way we compute premises for implications: instead of only considering sets as premises which are closed under the currently known implications (as we do when computing the canonical base), we shall also consider intents of the formal context as premises. The idea behind this is inspired by similar considerations from [4, Section 4.2]: intents of $\mathbb{K}$ may not be closed under $(\cdot)^{\mathbb{K},c}$, and thus implications of the form $P \to P^{\mathbb{K},c}$ may be non-trivial. Furthermore, if we know for a set $Q \subseteq M$ and all intents $P \subsetneq Q$ all implications $P \to P^{\mathbb{K},c}$, then computing the closure $\mathrm{Th}_c(Q)$ is comparably easy: an element $m \in M$ is in $\mathrm{Th}_c(Q)$ if it is in the closure of Q under all implications $P \to P^{\mathbb{K},c}$, $P \subsetneq Q$, or it is an element of $Q^{\mathbb{K},c}$. Thus, if Q is already closed under all those implications $P \to P^{\mathbb{K},c}$, then

$$\mathrm{Th}_c(Q) = Q^{\mathbb{K},c}.$$

We are not going to show these claims here, see [4] for a more thorough discussion on this.

Instead, we consider Algorithm 6, which incorporates all these ideas, and shall show that this algorithm allows us to compute bases of $\mathrm{Th}_c(\mathbb{K})$. Note that, as in the case of argumentation from Section 4.2, we demand that all attributes in the sets M_i are strictly linearly ordered by virtue of a relation $<$, such that for all $i < j$ and that all new attributes in $M_{i+1} \backslash M_i$ are less then all elements in M_i, i. e. , for all $x \in M_i$, $y \in M_{i+1} \backslash M_i$, it is true that $y < x$.

Clearly, Algorithm 6 is not guaranteed to terminate if we do not pose any restrictions on the growth of the sets of attributes. However, our target application of Algorithm 6 is to eventually apply it to growing subsets of $M_{\mathcal{I}_{\mathsf{back}}}$, i. e. to have that the sequence $(M_i \mid i \in \mathbb{N})$ in the algorithm is a monotonically increasing sequence of subsets of $M_{\mathcal{I}_{\mathrm{back}}}$. Since our background interpretation $\mathcal{I}_{\mathrm{back}}$ is finite, so is $M_{\mathcal{I}_{\mathrm{back}}}$, and thus the sets M_i will finally become stationary, i. e. there exists $n \in \mathbb{N}$ such that for all $i \geqslant n$ it is true that $M_i = M_n$. In this case, termination of the algorithm can be verified easily.

To argue that upon termination the resulting set $\mathcal{K}_n$ of implications yields an axiomatization of $\mathrm{Th}_c(\mathbb{K}_n)$, where n is the index of the last iteration, we mimic the argumentation from [7], as it has been presented in Section 4.2.

In the following, we shall encounter contextual derivations in various contexts. To keep the notation unambiguous, where necessary we shall from now on use the notation $(\cdot)''^{\mathbb{K}}$ to denote derivations in a context $\mathbb{K}$, respectively.

Algorithm 6 (Axiomatize Confident Implications with Growing Sets of Attributes)

define `base-confident-implications/growing-attributes`$(c \in [0,1])$
  ;; initialization
  $i := 0$
  $M_i := I_i := \mathcal{S}_i := P_i := \mathcal{K}_i := \varnothing$

  ;; computation
  forever do

    ;; obtain new attributes and new background knowledge
    Retrieve $\mathbb{K}_{i+1} = (G, M_{i+1}, I_{i+1}), \mathcal{S}_{i+1} \subseteq \mathrm{Th}_c(\mathbb{K}_{i+1})$ such that
      $-\ M_i \subseteq M_{i+1}$,
      $-\ \mathcal{S}_i \subseteq \mathcal{S}_{i+1}$,
      $-\ I_i = (G \times M_i) \cap I_{i+1}$.

    ;; update known implications
    $\mathcal{K}_{i+1} := \{\, P_k \to P_k^{\mathbb{K}_{i+1},c} \mid k \in \{\, 0, \ldots, i \,\}, P_k \neq P_k^{\mathbb{K}_{i+1},c} \,\}$

    ;; compute next candidate
    $P_{i+1}^1 :=$ `next-closed-set`$(M_{i+1}, <, P_i, \mathbb{K}_{i+1})$
    $P_{i+1}^2 :=$ `next-closed-set`$(M_{i+1}, <, P_i, \mathcal{S}_{i+1} \cup \mathcal{K}_{i+1})$

    if $P_{i+1}^1 =$ null **and** $P_{i+1}^2 =$ null **do**
      exit
    end

    $P_{i+1} := \min_{\preceq}(P_{i+1}^1, P_{i+1}^2)$.
    $i := i + 1$
  end

  ;; return result
  return $\mathcal{K}_i$
end

5.4 Proposition *In a terminating run of Algorithm 6 let n be the last value of the variable i. Let $Q \subseteq M_n$. Then the following statements are true:*

i. If $Q \neq Q''^{\mathbb{K}_n}$, then Q is not $(\mathcal{S}_n \cup \mathcal{K}_n)$-closed;

ii. If $Q = Q''^{\mathbb{K}_n}$, then $Q = P_k$ for some $k \in \{0, \ldots, n\}$.

Proof The case $Q = \varnothing = P_0$ can be handled quite easily: if $Q \neq Q''^{\mathbb{K}_n}$, then $Q \neq Q^{\mathbb{K}_n, c}$, hence $(Q \to Q^{\mathbb{K}_n, c}) \in \mathcal{K}_n$ and thus Q is not $(\mathcal{S}_n \cup \mathcal{K}_n)$-closed. If on the other hand $Q = Q''^{\mathbb{K}_n}$, then $Q = P_0$ shows the second claim.

We now consider the case $Q \neq \varnothing$. Then there exists $k \in \{0, \ldots, n\}$ such that

$$P_{k-1} \prec Q \preceq P_k. \tag{7}$$

As in the proof of [7, Lemma 6.3] we argue that $Q \subseteq M_k$. Suppose that this is not the case, and let $m \in Q \setminus M_k$. Since $m \notin M_k$, it is smaller than every element in M_k, therefore $M_k \prec \{m\}$. Since $\{m\} \subseteq Q$, it follows that $M_k \prec Q$, in contradiction to $Q \preceq P_k \preceq M_k$. Therefore $Q \subseteq M_k$.

We first consider the case $Q \neq Q''^{\mathbb{K}_n}$. Assume by contradiction that Q is $(\mathcal{S}_n \cup \mathcal{K}_n)$-closed. Then by construction of P_k^2, it is true that $Q \supseteq P_k^2 \supseteq P_k$. Then (7) implies $Q = P_k$. But then the condition $Q \neq Q''^{\mathbb{K}_n}$ means that $P_k \neq (P_k)''^{\mathbb{K}_n}$, thus

$$(P_k \to (P_k)''^{\mathbb{K}_n}) \in \mathcal{K}_n$$

and $Q = P_k$ is not $(\mathcal{S}_n \cup \mathcal{K}_n)$-closed, contradiction.

Now consider the case $Q = Q''^{\mathbb{K}_n}$. We then have to show that $Q = P_\ell$ for some $\ell \in \{0, \ldots, n\}$. Since $Q = Q''^{\mathbb{K}_n}$, it is true that $Q = Q''^{\mathbb{K}_k}$, since the incidence relation I_k of $\mathbb{K}_k$ satisfies $I_k = (G \times M_n) \cap I_n$. This implies that $Q \supseteq P_k^1 \supseteq P_k$, thus together with (7) we obtain $Q = P_k$ as desired. ▫

5.5 Theorem *Let $c \in [0, 1]$. Suppose that Algorithm 6 applied to c terminates and let n be the index of the last iteration. Then $\mathcal{K}_n$ is a base of $\mathrm{Th}_c(\mathbb{K}_n)$ with background knowledge $\mathcal{S}_n$.*

Proof It is clear from the construction that $\mathcal{K}_n \subseteq \mathrm{Cn}(\mathrm{Th}_c(\mathbb{K}_n))$. It thus remains to show that $\mathcal{S}_n \cup \mathcal{K}_n$ is complete for $\mathrm{Th}_c(\mathbb{K}_n)$. For this we shall show that every set $Q \subseteq M_n$ which is $\mathcal{S}_n \cup \mathcal{K}_n$-closed is also $\mathrm{Th}_c(\mathbb{K}_n)$-closed.

We show the claim by contradiction. To this end, let us assume that Q is $(\mathcal{S}_n \cup \mathcal{K}_n)$-closed, but not $\mathrm{Th}_c(\mathbb{K}_n)$-closed. Then there exists an implication $(P \to \{m\}) \in \mathrm{Th}_c(\mathbb{K}_n)$ such that $P \subseteq Q$ and $m \notin Q$. Since Q is $(\mathcal{S}_n \cup \mathcal{K}_n)$-closed, it follows from Proposition 5.4 that $Q = Q''^{\mathbb{K}_n}$. We can therefore assume that $P = P''^{\mathbb{K}_n}$, since

$$\mathrm{conf}_{\mathbb{K}_n}(P \to \{m\}) = \mathrm{conf}_{\mathbb{K}_n}(P''^{\mathbb{K}_n} \to \{m\}).$$

Again by Proposition 5.4 it follows that $P = P_k$ for some k, and thus

$$(P \to P^{\mathbb{K}_n, c}) \in \mathcal{K}_n.$$

(Note that $m \in P^{\mathbb{K}_n, c}$, but $m \notin P$, thus $P \neq P^{\mathbb{K}_n, c}$.) Since Q is $\mathcal{K}_n$-closed and $P \subseteq Q$, it is true that $P^{\mathbb{K}, c} \subseteq Q$. But since $m \in P^{\mathbb{K}, c}$ we obtain $m \in Q$, a contradiction. ▫

Having established the correctness of Algorithm 6 we now can consider some simple modifications thereof. Of course, instead of collection implications of the form $P_k \to (P_k)^{\mathbb{K}_i, c}$, one can also collect

$$P_k \to (P_k)^{\mathbb{K}_i, c} \setminus \mathcal{S}_i(P_k)$$

and adapt the algorithm correspondingly. Furthermore, the algorithm does not ensure that $\mathcal{K}_i \subseteq \mathrm{Th}_c(\mathbb{K}_n)$, i. e. that all implications in $\mathcal{K}_i$ satisfy the constraint that their confidence is at least c. However, if this is needed it can easily be achieved by splitting up the conclusions of the implications in $\mathcal{K}_i$ into singleton sets. Then, by the definition of $(\cdot)^{\mathbb{K},c}$, the resulting implications will all have confidence at least c in $\mathbb{K}_n$.

We have now covered the case of finding axiomatizations of $\mathrm{Th}_c(\mathbb{K})$, where the attribute set of $\mathbb{K}$ is allowed to grow during the run of the computation. We shall now consider our initially stated problem of finding axiomatizations of sets of implications of the form (6), i. e. of sets of implications which are true in some part of a given formal context, and enjoy a certain confidence in the other. For this, we shall adapt Algorithm 6 accordingly.

The main differences to Algorithm 6 is now that we have to deal with two formal contexts $\mathbb{K}_1$ and $\mathbb{K}_2$, instead of only one context $\mathbb{K}_1$. We achieve an extension of Algorithm 6 that can handle this extended setup by partitioning the working context into the two parts $\mathbb{K}_i$ and $\mathbb{L}_i$, and by replacing the expression $P_k^{\mathbb{K}_{i+1},c}$ by $P_k^{\mathbb{K}_{i+1},c} \cap (P_k)''^{\mathbb{L}_{i+1}}$. Here, the formal context $\mathbb{K}_i$ is meant to contain the untrusted objects, and thus plays the role of the context $\mathbb{K}_1$, where the formal context $\mathbb{L}_i$ contains the trusted ones, and thus corresponds to $\mathbb{K}_2$. The working context itself is then simply the subposition of the contexts $\mathbb{K}_i$ and $\mathbb{L}_i$, which is denoted by $\frac{\mathbb{K}_i}{\mathbb{L}_i}$ or (to save vertical space) $\mathbb{K}_i \div \mathbb{L}_i$. The result of these adaptions is shown in Algorithm 7.

Now since

$$P''^{\mathbb{K}_i \div \mathbb{L}_i} \subseteq P^{\mathbb{K}_i,c} \cap P''^{\mathbb{L}_i},$$

the proofs of Proposition 5.4 and Theorem 5.5 can be carried over almost literally to the setting of Algorithm 7, by essentially just replacing all occurrences of $(\cdot)''^{\mathbb{K}_i}$ by $(\cdot)''^{\mathbb{K}_i \div \mathbb{L}_i}$ and $P_k^{\mathbb{K}_{i+1},c}$ by $P_k^{\mathbb{K}_{i+1},c} \cap (P_k)''^{\mathbb{L}_{i+1}}$. We therefore obtain the validity of the following statements.

5.6 Proposition *In a terminating run of Algorithm 7 let n be the last value of the variable i. Let $Q \subseteq M_n$. Then the following statements are true:*

i. If $Q \neq Q''^{\mathbb{K}_n \div \mathbb{L}_n}$, then Q is not $(\mathcal{S}_n \cup \mathcal{K}_n)$-closed;

ii. If $Q = Q''^{\mathbb{K}_n \div \mathbb{L}_n}$, then $Q = P_k$ for some $k \in \{0, \ldots, n\}$.

5.7 Theorem *Let $c \in [0,1]$. Suppose that Algorithm 7 applied to c terminates and let n be the index of the last iteration. Then $\mathcal{K}_n$ is a base of $\mathrm{Th}_c(\mathbb{K}_n) \cap \mathrm{Th}(\mathbb{L}_n)$ with background knowledge $\mathcal{S}_n$.*

5.3 Exploring Confident GCIs with Known Background Interpretation

This section brings together the results of the previous Sections 5.1 and 5.2. This we want to do for the purpose of obtaining an algorithm that allows us to axiomatize confident GCIs in the presence of trusted individuals, satisfying the additional constraint that the set $M_{\mathcal{I}}$ is not computed in advance, but during the run of the algorithm. This not only has the benefit of speeding up the axiomatization process as a whole, but also serves as a basis for our further development of an algorithm implementing confident model exploration.

The argumentation of this section is actually quite simple: given an interpretation $\mathcal{I}$ and a connected subinterpretation $\mathcal{J}$ of $\mathcal{I}$, we want to find an axiomatization of $\mathrm{Th}_c(\mathcal{I}, \mathcal{J})$. To this end, we simply consider the induced formal context $\mathbb{K}_{\mathcal{I}}$ of $\mathcal{I}$, represented as the subposition of the formal contexts $\mathbb{K}_{\mathcal{I}}|_{\Delta^{\mathcal{I}} \setminus \Delta^{\mathcal{J}}}$ of trusted objects and $\mathbb{K}_{\mathcal{I}}|_{\Delta^{\mathcal{J}}}$ of untrusted objects. Then, we apply Algorithm 7 to these contexts, where we extent the set of attributes successively during the

Algorithm 7 (Adding Trusted Objects)

define `base-confident-implications/growing-attributes-trusted-objects`$(c \in [0,1])$
  ;; initialization
  $i := 0$
  $M_i := I_i := \mathcal{S}_i := P_i := \mathcal{K}_i := \varnothing$

  ;; computation
  forever do

    ;; obtain new attributes and background knowledge
    Retrieve $\mathbb{K}_{i+1} = (G_1, M_{i+1}, I_{i+1}), \mathbb{L}_{i+1} = (G_2, M_{i+1}, J_{i+1}), \mathcal{S}_{i+1} \subseteq \mathrm{Th}_c(\mathbb{K}_{i+1}) \cap \mathrm{Th}(\mathbb{L}_{i+1})$
    such that
      $-$ $M_i \subseteq M_{i+1}$,
      $-$ $\mathcal{S}_i \subseteq \mathcal{S}_{i+1}$,
      $-$ $I_i = (G_1 \times M_i) \cap I_{i+1}$,
      $-$ $J_i = (G_2 \times M_i) \cap J_{i+1}$

    ;; update known implications
    $\mathcal{K}_{i+1} := \{\, P_k \to P_k^{\mathbb{K}_{i+1},c} \cap (P_k)''^{\mathbb{L}_{i+1}} \mid k \in \{\, 0, \ldots, i \,\}, P_k \neq P_k^{\mathbb{K}_{i+1},c} \cap (P_k)''^{\mathbb{L}_{i+1}} \,\}$

    ;; compute next candidate
    $P_{i+1}^1 :=$ `next-closed-set`$(M_{i+1}, <, P_i, \frac{\mathbb{K}_{i+1}}{\mathbb{L}_{i+1}})$
    $P_{i+1}^2 :=$ `next-closed-set`$(M_{i+1}, <, P_i, \mathcal{S}_{i+1} \cup \mathcal{K}_{i+1})$

    if $P_{i+1}^1 =$ `null` **and** $P_{i+1}^2 =$ `null` **do**
      exit
    end

    $P_{i+1} := \min_{\preceq}(P_{i+1}^1, P_{i+1}^2)$.
    $i := i + 1$
  end

  ;; return result
  return $\mathcal{K}_i$
end

computation. The final set of implications obtained is then a base of $\mathrm{Th}(\mathbb{K}_\mathcal{I}|_{\Delta^\mathcal{I}\setminus\Delta^\mathcal{J}}) \cap \mathrm{Th}_c(\mathbb{K}_\mathcal{I}|_{\Delta^\mathcal{J}})$ by Theorem 5.7, and thus gives rise to a base of $\mathrm{Th}_c(\mathcal{I},\mathcal{J})$ by Theorem 5.3.

Algorithm 8 shows an instance of Algorithm 7 that follows this line of argumentation (we shall show in a moment that this is indeed the case). In addition, we also make use of the possibility to use background knowledge during our axiomatization process to get rid of implications $\{ C \} \to \{ D \}$, where C, D are $\mathcal{EL}^\bot_{\mathrm{gfp}}$ concept descriptions satisfying $C \sqsubseteq D$. While those implications are non-trivial, the resulting GCIs are, and thus they can be considered as background knowledge.

We claimed that Algorithm 8 is a special case of Algorithm 7. To see this, we need to verify that the way we compute the values for the variables $M_{i+1}, \mathbb{K}_{i+1}, \mathbb{L}_{i+1}, \mathcal{S}_{i+1}$ satisfies the constraints of Algorithm 7, and moreover, that the extra initializations we do in Line 4 is still within the general setting of Algorithm 7.

The former is clear from the definition of the variable values, and the latter can be argued quite easily: the initial values for M_0 could have equally well be added during the first iteration of the main loop, when the value of M_1 is computed. So, instead of initializing M_0 to $\{ \bot \} \cup N_C$, one could have equivalently set M_0 to $\varnothing$, and could have added a special case for $i = 1$, namely

$$M_1 := \{ \bot \} \cup N_C \cup \{ \exists r.\varnothing^{\mathcal{I}\mathcal{I}} \mid r \in N_R \},$$

Therefore, the separate initialization in Line 4 is within the general setting of Algorithm 7.

Note that since the input interpretation $\mathcal{I}$ we are considering here is always finite, the sets M_i in Algorithm 8 have to stabilize from a certain iteration on. Therefore, as we have argued similarly for Algorithm 6, Algorithm 8 always terminates.

Moreover, upon termination, the resulting set of implications gives rise to a base of $\mathrm{Th}_c(\mathcal{I},\mathcal{J})$ in the usual way.

5.8 Theorem *Let $\mathcal{I}$ be a finite interpretation, $\mathcal{J}$ a connected subinterpretation of $\mathcal{I}$ and $c \in [0,1]$. Let $\mathcal{K}_n$ be the result obtained from applying Algorithm 8 to $\mathcal{I}$, $\mathcal{J}$ and c. Then $\bigsqcap \mathcal{K}_n$ is a base of* $\mathrm{Th}_c(\mathcal{I},\mathcal{J})$.

The main idea for showing this result has already been sketched at the beginning of this section. The main technical difficulty we have to overcome is to connect the results from Theorem 5.3 and Theorem 5.7: note that while Theorem 5.3 talks about induced formal contexts with attribute set $M_\mathcal{I}$, Theorem 5.7 talks about arbitrary formal contexts with attribute sets M_n. To connect those theorems, we shall show that $M_n = M_\mathcal{I}$ is true up to equivalence. In this case, upon termination the subposition $\frac{\mathbb{K}_n}{\mathbb{L}_n}$ in Algorithm 8 can be seen as the induced context of $\mathcal{I}$, and we can apply Theorem 5.3.

The argumentation we shall follow in the sequel is almost identical to the one used by [7] to prove a corresponding result for the case of valid GCIs [7, Theorem 6.9]. The only difference is that we need to use Proposition 5.4 instead of its corresponding version [7, Lemma 6.3] for valid GCIs in the proof of the following result.

5.9 Proposition (Adapted from Lemma 6.7 of [7]) *Consider a terminating run of Algorithm 8 with n iterations. Then for every $U \subseteq M_n$ and $r \in N_R$ it is true that*

$$\exists r.(\textstyle\bigsqcap U)^{\mathcal{I}\mathcal{I}} \in M_n$$

up to equivalence.

Proof Note that the formal context $\mathbb{K}_n \div \mathbb{L}_n$ is the induced formal context of M_n and $\mathcal{I}$. Therefore, we obtain from Lemma 4.1 that

$$(\textstyle\bigsqcap U''^{\mathbb{K}_n \div \mathbb{L}_n})^\mathcal{I} = U'''^{\mathbb{K}_n \div \mathbb{L}_n} = U'^{\mathbb{K}_n \div \mathbb{L}_n} = (\bigsqcap U)^\mathcal{I}$$

Algorithm 8

define `base-confident-gcis/trusted-objects`$(c, \mathcal{I}, \mathcal{J})$
  ;; initialization
  $i := 0$
  $P_i := \mathcal{K}_i := \mathcal{S}_i := \emptyset$
  $M_0 := \{ \bot \} \cup N_C$

  ;; computation
  forever do

    ;; obtain new attributes and background knowledge
    $M_{i+1} := M_i \cup \{ \exists r.(\bigsqcap P_i)^{\mathcal{I}\mathcal{I}} \mid r \in N_R \}$ *;; union up to equivalence*
    $\mathbb{K}_{i+1} :=$ `induced-context`$(M_{i+1}, \mathcal{J})$
    $\mathbb{L}_{i+1} :=$ `induced-context`$(M_{i+1}, \mathcal{I} \backslash \mathcal{J})$
    $\mathcal{S}_{i+1} := \{ \{ C \} \to \{ D \} \mid C, D \in M_{i+1}, C \sqsubseteq D \}$

    ;; update known implications
    $\mathcal{K}_{i+1} := \{ P_k \to P_k^{\mathbb{K}_{i+1}, c} \cap (P_k)''_{\mathbb{L}_{i+1}} \mid k \in \{ 0, \ldots, i \}, P_k \neq P_k^{\mathbb{K}_{i+1}, c} \cap (P_k)''_{\mathbb{L}_{i+1}} \}$

    ;; compute next candidate
    $P^1_{i+1} :=$ `next-closed-set`$(M_{i+1}, <, P_i, \frac{\mathbb{K}_{i+1}}{\mathbb{L}_{i+1}})$
    $P^2_{i+1} :=$ `next-closed-set`$(M_{i+1}, <, P_i, \mathcal{S}_{i+1} \cup \mathcal{K}_{i+1})$

    if $P^1_{i+1} =$ `null` **and** $P^2_{i+1} =$ `null` **do**
      exit
    end

    $P_{i+1} := \min_{\preceq}(P^1_{i+1}, P^2_{i+1})$.
    $i := i + 1$
  end

  ;; return result
  return $\mathcal{K}_i$
end

Therefore, it is true that

$$\exists r.(\bigsqcap U''^{\mathbb{K}_n \div \mathbb{L}_n})^{\mathcal{I}\mathcal{I}} \equiv \exists r.(\bigsqcap U)^{\mathcal{I}\mathcal{I}}.$$

Since Algorithm 8 is a special case of Algorithm 7, we can invoke Proposition 5.4 to obtain a $k \in \{0, \ldots, n-1\}$ such that $U''^{\mathbb{K}_n \div \mathbb{L}_n} = P_k$. Since $\exists r.(\bigsqcap P_k)^{\mathcal{I}\mathcal{I}} \in M_{k+1} \subseteq M_n$, we obtain

$$\exists r.(\bigsqcap U)^{\mathcal{I}\mathcal{I}} \equiv \exists r.(\bigsqcap U''^{\mathbb{K}_n \div \mathbb{L}_n})^{\mathcal{I}\mathcal{I}} \equiv \exists r.(\bigsqcap P_k)^{\mathcal{I}\mathcal{I}} \in M_n$$

as required. □

The proof of Theorem 5.8 employs induction of the role depth of acyclic $\mathcal{EL}^{\bot}_{\text{gfp}}$ concept descriptions. However, as our concept descriptions are allowed to be cyclic, we need an extra argument to ensure that restricting our attention to acyclic concept descriptions is enough.

5.10 Lemma (Corollary 5.6 from [7]) *Let C be an $\mathcal{EL}^{\bot}_{gfp}$ concept description and let $\mathcal{I}$ be a finite interpretation. Then there exists an acyclic $\mathcal{EL}^{\bot}_{gfp}$ concept description D such that*

$$C^{\mathcal{I}} = D^{\mathcal{I}} \quad \text{and} \quad C \sqsubseteq D.$$

We also need the following technical result about model-based most-specific concept descriptions.

5.11 Lemma *Let C, D be two $\mathcal{EL}^{\bot}_{\text{gfp}}$ concept descriptions and let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation. Then the following statements are true:*

i. $(C^{\mathcal{I}\mathcal{I}} \sqcap D)^{\mathcal{I}} = (C \sqcap D)^{\mathcal{I}}$,

ii. $(\exists r.C^{\mathcal{I}\mathcal{I}})^{\mathcal{I}} = (\exists r.C)^{\mathcal{I}}$.

Having now these results in place, we are finally able to prove our main claim of this section. Parts of the proof are the same as for [7, Lemma 6.8, Theorem 6.9], and are repeated here for completeness.

Proof (Theorem 5.8) We first show that $M_n = M_{\mathcal{I}}$ is true up to equivalence. For this we note that $M_n \subseteq M_{\mathcal{I}}$ is true up to equivalence, as all elements of M_n are either of the form $\exists r.(\bigsqcap P_k)^{\mathcal{I}\mathcal{I}}$ for some set P_k of $\mathcal{EL}^{\bot}_{\text{gfp}}$ concept descriptions, or are in $\{\bot\} \cup N_C \subseteq M_{\mathcal{I}}$.

To verify that $M_{\mathcal{I}} \subseteq M_n$ is true up to equivalence, we shall show that for each set $X \subseteq \Delta^{\mathcal{I}}$ and $r \in N_R$ there exists $C \in M_n$ such that $C \equiv \exists r.X^{\mathcal{I}}$. To show this, we note that by Lemma 5.10 there exists for the concept description $X^{\mathcal{I}}$ an acyclic $\mathcal{EL}^{\bot}_{\text{gfp}}$ concept description D such that

$$D^{\mathcal{I}} = X^{\mathcal{I}\mathcal{I}}.$$

Now, since $D^{\mathcal{I}\mathcal{I}} = X^{\mathcal{I}\mathcal{I}\mathcal{I}} = X^{\mathcal{I}}$, it is sufficient for our claim to show that for each acyclic concept description D and for each $r \in N_R$, it is true that

$$\exists r.D^{\mathcal{I}\mathcal{I}} \in M_n$$

up to equivalence. We show this claim by induction on the role depth of D.

Base Case $D = \bot$ or D is a conjunction of concept names. The case $D = \bot$ is trivial, as $\exists r.D^{\mathcal{I}\mathcal{I}} \equiv \bot \in M_n$. If $D = \bigsqcap S$ for some $S \subseteq N_C$, then since $S \subseteq M_n$, Proposition 5.9 implies

$$\exists r.D^{\mathcal{I}\mathcal{I}} \equiv \exists r.(\bigsqcap S)^{\mathcal{I}\mathcal{I}} \in M_n$$

up to equivalence.

Step Case Let D be an acyclic concept description of role depth $d > 0$, $r \in N_R$ and assume that $\exists s.E^{\mathcal{II}} \in M_n$ is true up to equivalence for all acyclic $\mathcal{EL}^{\bot}_{\text{gfp}}$ concept descriptions E with role depth smaller than d and role names $s \in N_R$.

Since D is acyclic, there are concept names $U \subseteq N_C$, roles $r_1, \ldots, r_k$ and acyclic $\mathcal{EL}^{\bot}_{\text{gfp}}$ concept descriptions $E_1, \ldots, E_k$ with role depth smaller then d such that

$$D \equiv \bigsqcap U \sqcap \prod_{i=1}^{n} \exists r_i.E_i.$$

Then by Lemma 5.11

$$\begin{aligned} D^{\mathcal{II}} &= (\bigsqcap U \sqcap \prod_{i=1}^{n} \exists r_i.E_i)^{\mathcal{II}} \\ &\equiv (\bigsqcap U \sqcap \prod_{i=1}^{n} \exists r_i.E_i^{\mathcal{II}})^{\mathcal{II}}. \end{aligned}$$

By induction hypothesis, $\exists r_i.E_i^{\mathcal{II}} \in M_n$ up to equivalence. As $U \subseteq M_n$, Proposition 5.9 implies that

$$\exists r.D^{\mathcal{II}} \equiv \exists r.(\bigsqcap (U \cup \{\, \exists r_i.E_i^{\mathcal{II}} \mid i \in \{\, 1, \ldots, n \,\}\,))^{\mathcal{II}} \in M_n$$

is true up to equivalence. This completes the induction step.

We have now shown that $M_n = M_{\mathcal{I}}$ holds up to equivalence. By Theorem 5.7 we know that $\mathcal{K}_n$ is a base of

$$\text{Th}_c(\mathbb{K}_n) \cap \text{Th}(\mathbb{L}_n)$$

with background knowledge $\mathcal{S}_n = \{\, \{\, C \,\} \to \{\, D \,\} \mid C, D \in M_n, C \sqsubseteq D \,\}$. Since $M_n = M_{\mathcal{I}}$ holds up to equivalence, $\mathcal{K}_n$ is, up to equivalence, also a base of

$$\text{Th}_c(\mathbb{K}_{\mathcal{I}}|_{\Delta^{\mathcal{J}}}) \cap \text{Th}(\mathbb{K}_{\mathcal{I}}|_{\Delta^{\mathcal{I}} \setminus \Delta^{\mathcal{J}}}).$$

Therefore, by Theorem 5.3 the set $\bigsqcap(\mathcal{K}_n \cup \mathcal{S}_n)$ is a base of $\text{Th}_c(\mathcal{I}, \mathcal{J})$. Since $\bigsqcap(\mathcal{K}_n \cup \mathcal{S}_n) \equiv \bigsqcap \mathcal{K}_n$, $\bigsqcap \mathcal{K}_n$ is already a base of $\text{Th}_c(\mathcal{I}, \mathcal{J})$, which shows the claim. ▫

5.4 Exploring Confident GCIs with Expert Interaction

We have finally reached the point to present an algorithm for model exploration with confident GCIs. As in the line of argumentation for model exploration in Section 4.4, we shall take our Algorithm 8 that allows for axiomatizing confident GCIs for given background interpretations and replace every reference to the background interpretation by suitable expert interaction. This way we can avoid the need for explicitly knowing the background interpretation a-priori, and hence we obtain an algorithm for model exploration with confident GCIs.

If we consider Algorithm 8, we note that there are two explicit references to the background interpretation $\mathcal{I}$:

i. In the computation of concept descriptions of the form $\exists r.(\bigsqcap P_i)^{\mathcal{II}}$ (line 10), and

ii. In the computation of the formal context $\mathbb{L}_{i+1}$ as the induced context of M_{i+1} and $\mathcal{I} \setminus \mathcal{J}$ (line 12).

For the first point, we use the same idea as Distel used in his model exploration, formulated in Lemma 4.10: if the expert confirms the GCI

$$\bigsqcap P_i \sqsubseteq (\bigsqcap P_i)^{\mathcal{I}_\ell \mathcal{I}_\ell}$$

for some connected subinterpretation $\mathcal{I}_\ell$ of the background interpretation $\mathcal{I}$, then we can infer that $(\bigcap P_i)^{\mathcal{I}_\ell\mathcal{I}_\ell} \equiv (\bigcap P_i)^{\mathcal{I}\mathcal{I}}$.

The second point is more difficult: we have to use expert interaction to ensure that the context $\mathbb{L}_{i+1}$ contains all relevant counterexamples needed for both computing the set $\mathcal{K}_{i+1}$ of currently known implications, as well as for computing the next candidate set P^1_{i+1}. As it turns out, both of this can be achieved by expert interaction, which, regrettably, is much more involved than the one described in the previous paragraph.

To ensure that all implications computed for $\mathcal{K}_{i+1}$ are correct, we just present them to the expert, thus obtaining all the relevant counterexamples.

To ensure that the computation of the set P^1_{i+1} can be done without accessing the background interpretation directly, we propose a rather straight-forward solution. For this we observe that the computation of P^1_{i+1} would *not* be correct if the lectically first intent of the current working context in Algorithm 8 would not be an intent of the working context of the adapted algorithm using expert interaction. If we denote the latter context with $\bar{\mathbb{K}}_{i+1} \div \bar{\mathbb{L}}_{i+1}$, then this would mean that

$$P^1_{i+1} \neq (P^1_{i+1})^{\prime\prime_{\bar{\mathbb{K}}_{i+1} \div \bar{\mathbb{L}}_{i+1}}}.$$

But then, since the all the implications in $\mathcal{S}_{i+1} \cup \mathcal{K}_{i+1}$ are correct in $\bar{\mathbb{L}}_{i+1}$, and the formal context $\bar{\mathbb{K}}_{i+1}$ is already confirmed to be equivalent to $\mathbb{K}_{i+1}$, we obtain that

$$(\mathcal{S}_{i+1} \cup \mathcal{K}_{i+1})(P^1_{i+1}) \neq (\mathcal{S}_{i+1} \cup \mathcal{K}_{i+1})(P^1_{i+1})^{\prime\prime_{\bar{\mathbb{L}}_{i+1}}}.$$

However, P^1_{i+1} is an intent of $\mathbb{L}_{i+1}$, i. e.

$$(\mathcal{S}_{i+1} \cup \mathcal{K}_{i+1})(P^1_{i+1}) = (\mathcal{S}_{i+1} \cup \mathcal{K}_{i+1})(P^1_{i+1})^{\prime\prime_{\mathbb{L}_{i+1}}}.$$

Therefore, the implication

$$(\mathcal{S}_{i+1} \cup \mathcal{K}_{i+1})(P^1_{i+1}) \neq (\mathcal{S}_{i+1} \to \mathcal{K}_{i+1})(P^1_{i+1})^{\prime\prime_{\bar{\mathbb{L}}_{i+1}}}$$

must be rejected by the expert, providing the necessary counterexamples missing in $\bar{\mathbb{L}}_{i+1}$. Thus, if we ask all possible implications of the above form, we can ensure that the context $\bar{\mathbb{L}}_{i+1}$ contains all relevant counterexamples needed for the computation of P^1_{i+1}.

An algorithm that implements all these ideas can now derived easily from Algorithm 8, and the result is shown as Algorithm 9.

To see that this algorithm always terminates if the expert represents a finite background interpretation, we recall that Algorithm 8 terminates if the input background interpretation is finite. Therefore, if we can show that both Algorithm 8 and Algorithm 9 compute the same values up to equivalence in every step of their iterations, we can not only ensure that our confident model exploration terminates eventually, but also that the computed results are as desired.

Let us introduce some notation to make the argumentation of the following proof a bit clearer: if P and Q are sets of concept descriptions, then we write $P \equiv Q$ to denote the fact that each element of P is equivalent to some element in Q and vice versa. Likewise, if M and N are sets of implications where both premise and conclusion are sets of concept descriptions, then we write $M \equiv N$ to denote the fact that for each implication $(X_1 \to Y_1) \in M$, there exists an implication $(X_2 \to Y_2) \in N$ such that $X_1 \equiv X_2$ and $Y_1 \equiv Y_2$ is true, and vice versa.

5.12 Theorem *Assume that an expert represents a finite background interpretation, and that Algorithm 9 is applied to the input $\mathcal{J}$ and $c \in [0,1]$ using this expert.*

Then the run of this algorithm terminates. If n is the number of iterations of the algorithm, and if $\mathcal{I}_\ell$ denotes the final working interpretation of the algorithm, then the set $\bigcap \mathcal{K}_n$ is a base of $\mathrm{Th}_c(\mathcal{I}_\ell, \mathcal{J}) = \mathrm{Th}_c(\mathcal{J}) \cap \mathrm{Th}(\mathcal{I}_\ell \backslash \mathcal{J})$.

Algorithm 9

define `confident-model-exploration`($\mathcal{J}$, c)
  $i := \ell := 0$
  $\mathcal{I}_\ell := \mathcal{J}$
  $\bar{P}_i := \bar{\mathcal{K}}_i := \bar{\mathcal{S}}_i := \varnothing$
  $\bar{M}_0 := \{ \bot \} \cup N_C$

  forever do
    ;; present new GCI to the expert
    while expert rejects $\bigsqcap \bar{P}_i \sqsubseteq (\bigsqcap \bar{P}_i)^{\mathcal{I}_\ell \mathcal{I}_\ell}$ **do**
      $\mathcal{I}_{\ell+1} :=$ expert-defined extension of $\mathcal{I}_\ell$
      $\ell := \ell + 1$
    end

    $M_{i+1} := M_i \cup \{ \exists r.(\bigsqcap P_i)^{\mathcal{I}_\ell \mathcal{I}_\ell} \mid r \in N_R \}$ *;; union up to equivalence*

    ;; ensure relevant counterexamples for already known GCIs
    while expert rejects $\bigsqcap \bar{P}_k \sqsubseteq \bigsqcap (\bar{P}_k^{\mathbb{K}_{\mathcal{J}, \bar{M}_{i+1}}, c} \cap (\bar{P}_k)^{\prime\prime_{\mathbb{K}_{\mathcal{I}_\ell \setminus \mathcal{J}, \bar{M}_{i+1}}}})$ **do**
      $\mathcal{I}_{\ell+1} :=$ expert-defined extension of $\mathcal{I}_\ell$
      $\ell := \ell + 1$
    end

    $\bar{\mathbb{K}}_{i+1} :=$ `induced-context`($\bar{M}_{i+1}$, $\mathcal{J}$)
    $\bar{\mathbb{L}}_{i+1} :=$ `induced-context`($\bar{M}_{i+1}$, $\mathcal{I}_\ell \setminus \mathcal{J}$)
    $\bar{\mathcal{S}}_{i+1} := \{ \{ C \} \to \{ D \} \mid C, D \in \bar{M}_{i+1}, C \sqsubseteq D \}$

    $\bar{\mathcal{K}}_{i+1} := \{ \bar{P}_k \to \bar{P}_k^{\bar{\mathbb{K}}_{i+1}, c} \cap (\bar{P}_k)^{\prime\prime_{\bar{\mathbb{L}}_{i+1}}} \mid k \in \{ 0, \ldots, i \}, \bar{P}_k \neq \bar{P}_k^{\bar{\mathbb{K}}_{i+1}, c} \cap (\bar{P}_k)^{\prime\prime_{\bar{\mathbb{L}}_{i+1}}} \}$

    ;; additional expert interaction
    forall $Q \geq P$ **being** $(\bar{\mathcal{K}}_{i+1} \cup \bar{\mathcal{S}}_{i+1})$-closed **do**
      while expert rejects $\bigsqcap Q \sqsubseteq \bigsqcap Q^{\prime\prime_{\bar{\mathbb{L}}_{i+1}}}$ **do**
        $\mathcal{I}_{\ell+1} :=$ expert-defined extension of $\mathcal{I}_\ell$
        $\ell := \ell + 1$
        $\bar{\mathbb{L}}_{i+1} :=$ `induced-context`($\bar{M}_{i+1}$, $\mathcal{I}_\ell \setminus \mathcal{J}$)
      end
    end

    $\bar{P}^1_{i+1} :=$ `next-closed-set`($\bar{M}_{i+1}$, $<$, $\bar{P}_i$, $\frac{\bar{\mathbb{K}}_{i+1}}{\bar{\mathbb{L}}_{i+1}}$)
    $\bar{P}^2_{i+1} :=$ `next-closed-set`($\bar{M}_{i+1}$, $<$, $\bar{P}_i$, $\bar{\mathcal{S}}_{i+1} \cup \bar{\mathcal{K}}_{i+1}$)

    if $\bar{P}^1_{i+1} =$ `null` **and** $\bar{P}^2_{i+1} =$ `null` **do**
      exit
    end

    $\bar{P}_{i+1} := \min_{\preceq}(\bar{P}^1_{i+1}, \bar{P}^2_{i+1})$.
    $i := i + 1$
  end

  return $\bar{\mathcal{K}}_i$
end

Proof We show that Algorithm 9 with input $\mathcal{J}$ and c has the same output as Algorithm 8 with input $\mathcal{I}_\ell$, up to equivalence. The claims then follow from the results about Algorithm 8, see Theorem 5.8.

To this end, we shall show that for all $k \in \{0, \ldots, n\}$ it is true that $\bar{P}_k \equiv P_k, \bar{M}_k \equiv M_k, \bar{\mathcal{K}}_k \equiv \mathcal{K}_k, \bar{\mathcal{S}}_k \equiv \mathcal{S}_k$. We show this claim by induction over k.

Base Case: For $k = 0$, $P_0 = \varnothing = \bar{P}_0$, $\bar{M}_0 = \{\bot\} \cup N_C = M_0$, $\bar{\mathcal{K}}_0 = \varnothing = \mathcal{K}_0$ and $\bar{\mathcal{S}}_0 = \{\{\bot\} \to \{A\} \mid A \in N_C\} = \mathcal{S}_0$. Thus the claim holds.

Step Case: Let us assume that $i < n$ and that the claim holds for all $m \leqslant i$. Denote with $\mathcal{I}_l$ the current working interpretation when the algorithm has reached line 13. The algorithm can only reach this line if the expert has confirmed $\bigsqcap \bar{P}_i \sqsubseteq (\bigsqcap \bar{P}_i)^{\mathcal{I}_l \mathcal{I}_l}$. Since $\mathcal{I}_l$ is a connected subinterpretation of $\mathcal{I}_\ell$, we obtain by Lemma 4.10 that $(\bigsqcap \bar{P}_i)^{\mathcal{I}_l \mathcal{I}_l} \equiv (\bigsqcap \bar{P}_i)^{\mathcal{I}_\ell \mathcal{I}_\ell}$. Since $\bar{M}_i \equiv M_i$, it is true that $\bar{M}_{i+1} \equiv M_{i+1}$. This also implies $\bar{\mathcal{S}}_{i+1} \equiv \mathcal{S}_{i+1}$, since these sets depend on $\bar{M}_{i+1}$ and M_{i+1} only.

To show that $\bar{\mathcal{K}}_{i+1} \equiv \mathcal{K}_{i+1}$, it is sufficient to verify that

$$\bar{P}_m^{\bar{\mathbb{K}}_{i+1}, c} \cap (\bar{P}_m)''_{\bar{\mathbb{L}}_{i+1}} \equiv P_m^{\mathbb{K}_{i+1}, c} \cap (P_m)''_{\mathbb{L}_{i+1}} \tag{8}$$

is true for all $0 \leqslant m \leqslant i$.

To this end we first observe that $\bar{\mathbb{K}}_{i+1}$ only depends on $\mathcal{J}$ and $\bar{M}_{i+1}$, and likewise, $\mathbb{K}_{i+1}$ only depends on $\mathcal{J}$ and M_{i+1}. In particular, as $\bar{M}_{i+1} \equiv M_{i+1}$ and $\bar{P}_m \equiv P_m$ for all $m \leqslant i$, we obtain

$$\bar{P}_m^{\bar{\mathbb{K}}_{i+1, c}} \equiv P_m^{\mathbb{K}_{i+1, c}}. \tag{9}$$

Now let $\mathcal{I}_m$ be the working interpretation when reaching line 21 of Algorithm 9, and recall that $\mathbb{L}_{i+1}$ is the induced context of $\mathcal{I}_\ell \backslash \mathcal{J}$ and M_{i+1}, and that $\bar{\mathbb{L}}_{i+1}$ is the induced context of $\mathcal{I}_m \backslash \mathcal{J}$ and $\bar{M}_{i+1}$. Therefore, $(\bar{P}_m)''_{\bar{\mathbb{L}}_{i+1}} \supseteq (P_m)''_{\mathbb{L}_{i+1}}$ (up to equivalence), simply because $(\bar{P}_m)'^{\bar{\mathbb{L}}_{i+1}} \subseteq (P_m)'^{\mathbb{L}_{i+1}}$.

Suppose by contradiction that

$$\bar{P}_m^{\bar{\mathbb{K}}_{i+1}, c} \cap (\bar{P}_m)''_{\bar{\mathbb{L}}_{i+1}} \not\equiv P_m^{\mathbb{K}_{i+1}, c} \cap (P_m)''_{\mathbb{L}_{i+1}}$$

is true for some $m \leqslant i$. Then there exists a $\bar{C} \in \bar{P}_m^{\bar{\mathbb{K}}_{i+1}, c} \cap (\bar{P}_m)''_{\bar{\mathbb{L}}_{i+1}}$ that is not equivalent to any element in $P_m^{\mathbb{K}_{i+1}, c} \cap (P_m)''_{\mathbb{L}_{i+1}}$. Then

i. The expert confirmed
$$\bigsqcap \bar{P}_m \sqsubseteq \bigsqcap (\bar{P}_m^{\bar{\mathbb{K}}_{i+1}, c} \cap (\bar{P}_m)''_{\bar{\mathbb{L}}_{i+1}}),$$
in particular, the GCI $\bigsqcap \bar{P}_m \sqsubseteq \bar{C}$ is valid in $\mathcal{I}_\ell \backslash \mathcal{J}$.

ii. The GCI $\bigsqcap P_m \sqsubseteq \bar{C}$ is not confirmed by the expert. To see this, observe that since $\bar{C} \in \bar{M}_{i+1}$, there exists an equivalent $C \in M_{i+1}$. Then if $\bigsqcap P_m \sqsubseteq \bar{C}$ were confirmed by the expert, it would be true that $C \in (P_m)''_{\mathbb{L}_{i+1}}$. Furthermore, $\mathrm{conf}_{\bar{\mathbb{K}}_{i+1}}(\bar{P}_m \to \{\bar{C}\}) = \mathrm{conf}_{\mathbb{K}_{i+1}}(P_m \to \{C\})$, since $\bar{P}_m \equiv P_m$, $\bar{M}_{i+1} \equiv M_{i+1}$ and $\bar{C} \equiv C$. As $\bar{C} \in \bar{P}_m^{\bar{\mathbb{K}}_{i+1}, c}$, it is true that $\mathrm{conf}_{\bar{\mathbb{K}}_{i+1}}(\bar{P}_m \to \{\bar{C}\}) \geqslant c$, thus $\mathrm{conf}_{\mathbb{K}_{i+1}}(P_m \to \{C\}) \geqslant c$ and hence $C \in P_m^{\mathbb{K}_{i+1}, c}$. In other words, using (9), it would be true that
$$\bar{C} \equiv C \text{ and } C \in P_m^{\mathbb{K}_{i+1}, c} \cap (P_m)''_{\mathbb{L}_{i+1}},$$
contradicting our assumption about $\bar{C}$. In particular, $\bigsqcap P_m \sqsubseteq \bar{C}$ is not valid in $\mathcal{I}_\ell \backslash \mathcal{J}$.

However, as $\bar{P}_m \equiv P_m$, the fact that $\bigsqcap \bar{P}_m \sqsubseteq C$ is valid in $\mathcal{I}_\ell \backslash \mathcal{J}$ but $\bigsqcap P_m \sqsubseteq C$ is not yields the desired contradiction. Thus, we have shown the validity of (8) and can infer that $\bar{\mathcal{K}}_{i+1} \equiv \mathcal{K}_{i+1}$.

It remains to be shown that

$$\bar{P}_{i+1} \equiv P_{i+1}. \tag{10}$$

It is clear from the corresponding definitions and the preceding argumentation that $\bar{P}^2_{i+1} \equiv P^2_{i+1}$, since $\bar{P}_i \equiv P_i$ and $\bar{\mathcal{K}}_{i+1} \cup \bar{\mathcal{S}}_{i+1} \equiv \mathcal{K}_{i+1} \cup \mathcal{S}_{i+1}$ are true. Thus, to show (10), it is sufficient to verify

$$\bar{P}^1_{k+1} \equiv P^1_{k+1}. \tag{11}$$

For this, note that since $\bar{M}_{i+1} \equiv M_{i+1}$, the contexts $\bar{\mathbb{K}}_{i+1}$ and $\mathbb{K}_{i+1}$ can be considered the same up to equivalence, and the formal context $\bar{\mathbb{L}}_{i+1}$ can be seen as a subcontext of $\mathbb{L}_{i+1}$, again up to equivalence. From this we can infer that $\bar{P}^1_{i+1} \succeq P^1_{i+1}$ holds up to equivalence.

Suppose by contradiction that $\bar{P}^1_{i+1} \not\succeq P_{i+1}$. Then P^1_{i+1} viewed as a subset of $\bar{M}_{i+1}$ is not an intent of $\frac{\bar{\mathbb{K}}_{i+1}}{\bar{\mathbb{L}}_{i+1}}$, as otherwise $\bar{P}^1_{i+1} \preceq P_{i+1}$. Since $\bar{\mathbb{K}}_{i+1}$ and $\mathbb{K}_{i+1}$ can be considered the same up to equivalence, and since P^1_{i+1} is an intent of $\frac{\mathbb{K}_{i+1}}{\mathbb{L}_{i+1}}$, we can infer that

$$(P^1_{i+1})''_{\mathbb{L}_{i+1}} \backslash (P^1_{i+1})''_{\bar{\mathbb{L}}_{i+1}} \neq \varnothing.$$

Let $D \in (P^1_{i+1})''_{\mathbb{L}_{i+1}} \backslash (P^1_{i+1})''_{\bar{\mathbb{L}}_{i+1}}$. Now since $\bar{\mathcal{K}}_{k+1} \cup \bar{\mathcal{S}}_{k+1}$ is sound for $\mathbb{L}_{i+1}$ up to equivalence, we obtain

$$((\bar{\mathcal{K}}_{i+1} \cup \bar{\mathcal{S}}_{i+1})(P^1_{i+1}))''_{\mathbb{L}_{i+1}} = (P^1_{i+1})''_{\mathbb{L}_{i+1}}$$

and thus

$$D \notin ((\bar{\mathcal{K}}_{i+1} \cup \bar{\mathcal{S}}_{i+1})(P^1_{i+1}))''_{\mathbb{L}_{i+1}}.$$

Therefore, the implication

$$(\bar{\mathcal{K}}_{i+1} \cup \bar{\mathcal{S}}_{i+1})(P^1_{i+1}) \to ((\bar{\mathcal{K}}_{i+1} \cup \bar{\mathcal{S}}_{i+1})(P^1_{i+1}))''_{\bar{\mathbb{L}}_{i+1}}$$

does not hold in $\mathbb{L}_{i+1}$, because

$$((\bar{\mathcal{K}}_{i+1} \cup \bar{\mathcal{S}}_{i+1})(P^1_{i+1}))''_{\mathbb{L}_{i+1}} \not\subseteq ((\bar{\mathcal{K}}_{i+1} \cup \bar{\mathcal{S}}_{i+1})(P^1_{i+1}))''_{\mathbb{L}_{i+1}}.$$

Therefore, the corresponding GCI

$$\bigsqcap (\bar{\mathcal{K}}_{i+1} \cup \bar{\mathcal{S}}_{i+1})(P^1_{i+1}) \sqsubseteq \bigsqcap ((\bar{\mathcal{K}}_{i+1} \cup \bar{\mathcal{S}}_{i+1})(P^1_{i+1}))''_{\mathbb{L}_{i+1}} \tag{12}$$

will be rejected by the expert, since $\mathbb{L}_{i+1}$ is the induced context of $\mathcal{I}_\ell \backslash \mathcal{J}$.

However, when computing P^1_{i+1}, we have passed the lines 28 up to 34, and the expert has confirmed the GCI given in (12), contradiction. Therefore, our initial assumption that $\bar{P}^1_{i+1} \not\succeq P^1_{i+1}$ is not true, and thus we have shown $\bar{P}^1_{i+1} \equiv P^1_{i+1}$, which finishes the proof. ▫

6 Conclusions

In practical applications, automatically generating valid knowledge from data has to always face the problem errors. To distinguish such errors from real facts an additional source of knowledge, an *expert*, has to be used, and in the best case the answers of such an expert are much more reliable. In this case, one can then generate knowledge from erroneous data by conducting supervised learning from the initial data, consulting the expert to check whether the obtained knowledge is valid or not.

In this work we have considered a special case of this general scenario, namely to learn terminological knowledge expressible in the description logic $\mathcal{EL}^\bot$ (or $\mathcal{EL}^\bot_{\text{gfp}}$) from interpretations. To achieve a supervised learning algorithm as sketched above, we have extended the algorithm of *model exploration* as developed by Distel to the case of interpretations which are allowed to contain errors. To this end, we have adapted Distel's argumentation to allow for general concept inclusions which are not necessarily valid in the interpretation, but whose *confidence* therein is large enough, i. e. above a chosen threshold $c \in [0, 1]$. This way, rare errors do not inhibit general concept inclusions from being considered, and the expert can decide whether the counterexamples contained in the interpretation are valid ones, errors, or can provide new, valid counterexamples.

While the motivation of such a method can be argued quite naturally, its practical usefulness remains to be proven. This is especially the case with our Algorithm 9, as it makes extensive use of expert interaction. As we consider expert interaction to be expensive, this high amount of expert interaction diminishes our expectations of the practicability of Algorithm 9. Therefore, a necessary step to improve the results of this work, beside investigating practical use-cases, is to consider simplifications of Algorithm 9 that significantly reduce the number of questions asked to the expert. In the best case, an algorithm which does a minimal number of expert interaction can be devised, as in the case of classical attribute exploration.

Another practicability issue is our assumption that the expert itself does not make errors. This assumption is mainly motivated to keep the theoretical considerations simple, but cannot be upheld for practical applications. A modification of Algorithm 9 that allows experts to revoke previously confirmed implications or provided counterexamples would thus be highly desirable.

Finally, our algorithm of model exploration with confident GCIs can be seen from a completely different point of view: during the exploration, the expert is potentially faced with the question of whether counterexamples in the given data are proper counterexamples or just errors. In case the expert declines that some counterexamples are valid, she implicitly conducts a form of *error correction* in the initial data, thereby improving its overall quality. Therefore, one could think of an adaptation of model exploration with confident GCIs that allows for correcting errors in the data which are relevant for the logical fragment of GCIs expressible in $\mathcal{EL}^\bot$. As $\mathcal{EL}^\bot$ is quite a simple logic, such an error correction may be much more accessible than letting the expert consider the whole data on its own and conduction a detailed error correction. Therefore, model exploration with confident GCIs could provide a practical method for improving the quality of data.

References

[1] Daniel Borchmann. *Axiomatizing Confident $\mathcal{EL}^\bot_{gfp}$-GCIs of Finite Interpretations*. Report MATH-AL-08-2012. Dresden, Germany: Chair of Algebraic Structure Theory, Institute of Algebra, Technische Universität Dresden, Sept. 2012.

[2] Daniel Borchmann. "Axiomatizing Confident $\mathcal{EL}^\bot_{\text{gfp}}$-General Concept Inclusions in the Presence of Untrusted Individuals". In: *Description Logics*. Ed. by Thomas Eiter et al. Vol. 1014. CEUR Workshop Proceedings. CEUR-WS.org, 2013, pp. 65–79.

[3] Daniel Borchmann. "Axiomatizing $\mathcal{EL}^\bot$-Expressible Terminological Knowledge from Erroneous Data". In: *Proceedings of the Seventh International Conference on Knowledge Capture*. (Banff, Canada). 2013.

[4] Daniel Borchmann. *Exploration by Confidence*. LTCS-Report 13-04. See http://lat.inf.tu-dresden.de/research/reports.html. Dresden, Germany: Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, 2013.

[5] Daniel Borchmann. *On Confident GCIs of Finite Interpretations*. LTCS-Report 12-06. See `http://lat.inf.tu-dresden.de/research/reports.html`. Dresden: Institute for Theoretical Computer Science, TU Dresden, 2012.

[6] Daniel Borchmann. "Towards an Error-Tolerant Construction of $\mathcal{EL}^{\perp}$ -Ontologies from Data Using Formal Concept Analysis". In: *Formal Concept Analysis, 11th International Conference, ICFCA 2013, Dresden, Germany, May 21-24, 2013. Proceedings*. Ed. by Peggy Cellier, Felix Distel, and Bernhard Ganter. Vol. 7880. Lecture Notes in Computer Science. Springer, 2013, pp. 60–75.

[7] Felix Distel. "Learning Description Logic Knowledge Bases from Data Using Methods from Formal Concept Analysis". PhD thesis. TU Dresden, 2011.

[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, Mar. 2010, pp. 312–340.

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