Relational Exploration – Reconciling Plato and Aristotle*

Sebastian Rudolph

Universitt Karlsruhe
Institut AIFB
rudolph@aifb.uni-karlsruhe.de

Abstract. We provide an interactive method for knowledge acquisition combining approaches from description logic and formal concept analysis. Based on present data, hypothetical rules are formulated and checked against a description logic theory. We propose an abstract framework (Logical Domain Exploration) for this kind of exploration technique before presenting a concrete instantiation: Relational Exploration. We give a completeness result and provide an overview about some application fields for our approach: machine learning, data mining, and ontology engineering.

1 Introduction

A plethora of research fields is concerned with the question of finding specifications for a given domain. Research areas like machine learning, frequent pattern discovery, and data mining in general aim at extracting these description on the basis of (examplary or complete) data sets – following the Aristotelian paradigm, that every conceptualization has to start from entities actually present. Other approaches intend to deduce these specifications from pre-specified theories – being somehow more Platonic by assuming the primacy of abstract ideas. The latter is the usual *modus operandi* e.g. in description logic or theorem proving.

We reconcile these two antagonistic approaches by combining techniques from two fields of knowledge representation: description logic (DL) and formal concept analysis (FCA).

In our work, we use DL formalisms for defining FCA attributes and FCA exploration techniques to obtain or refine DL knowledge representation specifications. More generally, DL exploits FCA techniques for interactive knowledge acquisition and FCA benefits from DL in terms of expressing relational knowledge.

In most cases, the process of conceptually specifying a domain cannot dispense of human contribution. However, although all information needed in order to describe a domain is in general implicitly present in an expert's knowledge, the process of explicit formal specification may nevertheless be tedious and overstraining. Moreover, it might remain unclear whether a specification is complete, i.e., whether it covers all valid statements about the domain that can be expressed in the chosen specification language.

* This work has been supported by the European Commission under contract IST-2006-027595 NeOn, and by the Deutsche Forschungsgemeinschaft (DFG) under the ReaSem project.

Hence, we provide a method – called Relational Exploration (RE) – that organizes and structures the specification process by successively asking single questions to the domain expert in a way which minimizes the expert's effort (in particular, it does not ask redundant questions) and guarantees that the resulting specification will be complete in the sense stated above. To present our work, which generalize the results from [1] and [2], we will proceed as follows: Section 2 provides a general framework for this kind of procedure, called Logical Domain Exploration. In Section 3, we shortly sketch the FCA basics necessary for our work and give an overview about attribute exploration. Section 4 introduces the notions from description logics needed in this work. In Section 5, we establish the correspondence between DL models and certain formal contexts, which enables us to apply FCA to DL. In Section 6, the RE algorithm is described in detail. Section 7 shows certain completeness properties of the knowledge acquired via RE. Section 9 displays direction for further work. In Section 8, we discuss our results and consider in which fields the presented technique could be applied.

2 The Epistemic Framework: Logical Domain Exploration

Before engaging into the technical details, we sketch the overall setting for our approach, which helps conveying the underlying idea and identifying the contributing components. Doing this on an abstract level, we also give an opportunity to relate alternative approaches. This framework will be called Logical Domain Exploration.

Let Δ be the considered domain of interest the elements of which we will call (DOMAIN) INDIVIDUALS. Let $\mathcal{L}$ be a language the elements of which are called FORMULAE. We write $\Delta \models \varphi$ in order to state for a formula φ that it is valid in the domain. Moreover let the setting be well-behaved in the way that whenever $\Delta \models \varphi$ is *not* true, there is a finite individual set $\Gamma \subseteq \Delta$ witnessing this (we then write $\Gamma \dagger \varphi$ and say Γ SPOILS φ).

- The EXPERT is supposed to be "omniscient" wrt. the described domain and thus able to answer any question about it. In particular, he knows for all $\varphi \in \mathcal{L}$ and $\Gamma \subseteq \Delta$ whether $\Delta \models \varphi$ and whether $\Gamma \dagger \varphi$. Mostly, a human or a group of humans will take the role of the expert.
- The TERMINOLOGY consists of a theory $Th \subseteq \mathcal{L}^{\mathcal{T}}$ about the domain consisting of axioms in some language $\mathcal{L}^{\mathcal{T}} \supseteq \mathcal{L}$ and a reasoning functionality, i.e. for any statement $\varphi \in \mathcal{L}^{\mathcal{T}}$ it can be decided whether Th entails φ.[1]
- The DATA consists of a set of known or recorded individuals $\mathcal{D} \subseteq \Delta$ and is endowed with a special querying capability, i.e., a procedure providing for any $\varphi \in \mathcal{L}$ a set $\Gamma \subseteq \mathcal{D}$ with $\Gamma \dagger \varphi$ if there exists one.
- The SCHEDULER can be conceived as an automated procedure initiating and coordinating the "information flow". It links the other system components by asking questions, processing answers, and assuring that in the end all knowledge is acquired to quickly decide for any $\varphi \in \mathcal{L}$ whether $\Delta \models \varphi$.

[1] Hereby, entailment is as usual defined in a model-theoretic way: Th is said to entail φ if any domain Δ' wherein all formulae of Th are valid also satisfies $\Delta' \models \varphi$.

The system will operate as follows: We start with a (correct but in general incomplete) terminological theory $Th \subset \{\psi \in \mathcal{L}^{\mathcal{T}} \mid \Delta \models \psi\}$ and data $\mathcal{D} \subseteq \Delta$. The scheduler now comes up with hypothetical formulae. Every such hypothetical formula $\varphi \in \mathcal{L}$ is passed both to the terminology and the data. The reasoning service of the terminology component checks whether φ is entailed by Th. The data is queried for a spoiler of φ. Since – due to the starting conditions – the theory is consistent with the data, we get three disjoint possible results:

- φ is entailed by Th. In this case, φ is valid in Δ, which will be responded to the scheduler.
- $\Gamma \in \mathcal{D}$ spoils φ. Then, φ is not valid in Δ and the scheduler will be provided with this negative answer.
- Neither of the previous cases occurs. Then, the current specification leaves room for either possibility and the domain expert will have to be asked this about φ's validity in question. If he confirms the validity of φ in Δ, it will be added to Th. If he denies it, he has to provide a spoiler Γ for φ, which is then added to the data.

Note that querying the data and questioning the terminology can be done in either order or even in parallel. After finishing the procedure every formula $\varphi \in \mathcal{L}$ will either be a consequence of the resulting (updated) terminology or can be excluded via a spoiler present in the data (updated) data. The distinction between $\mathcal{L}$ (the EXPLORATION LANGUAGE) and $\mathcal{L}^{\mathcal{T}}$ (the TERMINOLOGICAL LANGUAGE) is motivated by the assumption that in most cases not all terminologically expressible axioms will be of interest but only those of a certain shape.

In the next chapters, we come down to an instance for the previously described framework for logical domain exploration: Relational Exploration.

3 Formal Concept Analysis

In our instantiation, the scheduler's task will be carried out by an extension of the attribute exploration algorithm well established in FCA. This necessitates to briefly introduce some basic FCA notions. We mainly follow the notation introduced in [3] being *the* reference for FCA theory.

The basic notion FCA is built on is that of a formal context. It is a common claim in FCA that any kind of grounded data can be represented in this way.

Definition 1. *A* FORMAL CONTEXT $\mathbb{K}$ *is a triple* (G, M, I) *with an arbitrary set* G *(called* OBJECTS*), an arbitrary set* M *(called* ATTRIBUTES*), and a relation* $I \subseteq G \times M$ *(called* INCIDENCE RELATION*). We read* gIm *as: "object* g *has attribute* m*." Furthermore, let* $g^I := \{m \mid gIm\}$.

The central means of expressing knowledge in FCA is via implications. Thus, in terms of the general framework from Section 2 the underlying language consists of implications on a fixed attribute set of atomic propositions.

Definition 2. *Let* M *be an arbitrary set. An* IMPLICATION *on* M *is a pair* (A, B) *with* $A, B \subseteq M$. *To support intuition, we write* $A \rightarrow B$ *instead of* (A, B). $A \rightarrow B$ HOLDS *in* *a formal context $\mathbb{K} = (G, M, I)$, if for all $g \in G$ we have that $A \subseteq g^I$ implies $B \subseteq g^I$. We then write $\mathbb{K} \models A \rightarrow B$.*

For $C \subseteq M$ and a set $\mathfrak{I}$ of implications on M, let $C^{\mathfrak{I}}$ denote the smallest set with $C \subseteq C^{\mathfrak{I}}$ that additionally fulfills

$$A \subseteq C^I \quad implies \quad B \subseteq C^I$$

for every implication $A \rightarrow B$ in $\mathfrak{I}$.[2] If $C = C^{\mathfrak{I}}$, we call C $\mathfrak{I}$-CLOSED. We say $\mathfrak{I}$ ENTAILS $A \rightarrow B$ if $B \subseteq A^{\mathfrak{I}}$.[3]

An implication set $\mathfrak{I}$ will be called NON-REDUNDANT, if for any $(A \rightarrow B) \in \mathfrak{I}$ we have that $B \not\subseteq A^{\mathfrak{I} \setminus \{A \rightarrow B\}}$.

An implication set $\mathfrak{I}$ of a context $\mathbb{K}$ will be called COMPLETE, if every implication $A \rightarrow B$ holding in $\mathbb{K}$ is entailed by $\mathfrak{I}$.

$\mathfrak{I}$ will be called an IMPLICATION BASE of a formal context $\mathbb{K}$ if it is non-redundant and complete.

Note that implication entailment is decidable in linear time ([4]). Therefore, knowing a domain's implication base allows fast handling of its whole implicational theory. Moreover, for every formal context, there exists a canonical implication base ([5]).

The attribute exploration algorithm our work is based on was introduced in [6]. Due to space reasons, we omit to display it in detail and refer the reader to the literature.

Essentially, the following happens: the domain to explore is formalized as a formal context $\mathbb{K} = (U, M, I)$. Usually, it is not known completely in advance. However, possibly, some entities of the universe $g \in U$ are already known, as well as their associated attributes g^I.

The algorithm now starts presenting questions of the form

"Does the implication $A \rightarrow B$ hold in the context $\mathbb{K} = (U, M, I)$?"

to the human expert. The expert might confirm this. In this case, $A \rightarrow B$ is archived as part of $\mathbb{K}$'s implicational base $\mathfrak{IB}$. The other case would be that $A \rightarrow B$ does not hold in (U, M, I). But then, there must exist a $g \in U$ with $A \in g^I$ and $B \notin g^I$. The expert is asked to input this g and g^I.[4] The procedure terminates when the implicational knowledge of the universe is completely acquired, i.e., the implications of the formal context built from the entered counterexamples coincide with those entailed by $\mathfrak{IB}$.

In the approach presented here, we will exploit the capability of attribute exploration to efficiently determine an implicational theory. Notwithstanding, we extend the underlying language[5] from purely propositional to certain DL expressions being introduced in the next section.

---

[2] Note, that this is well-defined, since the mentioned properties are closed wrt. intersection.

[3] Actually, this is a syntactic shortcut. Yet, it can be easily seen that this coincides with the usual entailment notion.

[4] Referring to the general framework we mention that in this special case the spoiler (called counterexample) is always a singleton set: $\{g\} \dagger A \rightarrow B$.

[5] There exist already other language extensions, e.g. to Horn-logic with a bounded variable set, see [7].

4 Description Logic

We recall basic notions from DL, following (and recommending for further reading) [8].

Unlike the way DL is normally conceived, we use DL expressions to describe or specify *one particular, fixed* domain.

Thus, we will start our considerations by formally defining the kind of relational structure that we want to "talk about."

Definition 3. *An* INTERPRETATION *for a set $\mathcal{A}$ of* (PRIMITIVE) CLASS NAMES *and a set $\mathcal{R}$ of* ROLE NAMES *is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is some set and $(.)^{\mathcal{I}}$ is a function mapping class names to subsets of $\Delta^{\mathcal{I}}$ and role names to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.*

Verbally, for some primitive class name A, $\mathrm{A}^{\mathcal{I}}$ provides all members of that class and for some role name R, $\mathrm{R}^{\mathcal{I}}$ yields all ordered pairs "connected" by that role.

The DL languages introduced here provide constructors for defining new concept descriptions out of the primitive ones. Table 1 shows those constructors, their interpretation (as usual defined recursively), and their availabilities in the description logics considered here.

	name	interpretation	$\mathcal{FL}_0$	$\mathcal{EL}$	$\mathcal{FLE}$	$\mathcal{ALE}$
A	atomic concept	$\mathrm{A}^{\mathcal{I}}$	×	×	×	×
$\top$	universal concept	$\Delta^{\mathcal{I}}$	×	×	×	×
$\bot$	bottom concept	$\emptyset$	×	×	×	×
$\neg\mathrm{A}$	atomic negation	$\Delta^{\mathcal{I}} \setminus \mathrm{A}^{\mathcal{I}}$				×
$\mathrm{C} \sqcap \mathrm{D}$	conjunction	$\mathrm{C}^{\mathcal{I}} \cap \mathrm{D}^{\mathcal{I}}$	×	×	×	×
$\forall\mathrm{R.C}$	value restriction	$\{\delta \mid \forall\epsilon : (\delta,\epsilon) \in \mathrm{R}^{\mathcal{I}} \to \epsilon \in \mathrm{C}^{\mathcal{I}}\}$	×		×	×
$\exists\mathrm{R.C}$	existential quantification	$\{\delta \mid \exists\epsilon : (\delta,\epsilon) \in \mathrm{R}^{\mathcal{I}} \wedge \epsilon \in \mathrm{C}^{\mathcal{I}}\}$		×	×	×

Table 1. syntax and semantics of the DLs considered in this paper

In the sequel, we will in general speak of a description logic $\mathcal{DL}$ if the presented result or definition refers to any $\mathcal{DL} \in \{\mathcal{FL}_0, \mathcal{EL}, \mathcal{FLE}, \mathcal{ALE}\}$.

Definition 4. *Let $\mathcal{I}$ be an interpretation and C, D be $\mathcal{DL}$ concept descriptions. We say* C IS SUBSUMED BY D *in $\mathcal{I}$ (written: $\mathrm{C} \sqsubseteq_{\mathcal{I}} \mathrm{D}$) if $\mathrm{C}^{\mathcal{I}} \subseteq \mathrm{D}^{\mathcal{I}}$. This kind of subsumption statements is also called* GENERAL CONCEPT INCLUSION AXIOM *(GCI). Moreover, we say C and D are* EQUIVALENT *in $\mathcal{I}$ (written: $\mathrm{C} \equiv_{\mathcal{I}} \mathrm{D}$) if $\mathrm{C}^{\mathcal{I}} = \mathrm{D}^{\mathcal{I}}$.*

5 Subsumptions as Implications

Combinations of FCA and DL have already been described in several publications, e.g. in [9], [10], and [11]. Our approach is motivated by [9] insofar as we use the same way of transferring a DL setting into a formal context by considering the domain individuals as objects and DL concept expressions as attributes.

Definition 5. *Given an interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$ *and a set* M *of* $\mathcal{DL}$ *concept descriptions, we define the corresponding* $\mathcal{DL}$-CONTEXT

$$\mathbb{K}^{\mathcal{I}}(M) := (\Delta^{\mathcal{I}}, M, I)$$

where $\delta I C :\Longleftrightarrow \delta \in \mathrm{C}^{\mathcal{I}}$*, for all* $\delta \in \Delta^{\mathcal{I}}$ *and* $\mathrm{C} \in M$.

The observation in the next theorem – though easy to see – is crucial for applying attribute exploration for the intended purpose.

Theorem 1. *Let* $\mathfrak{I}$ *be an arbitrary interpretation and* $\mathbb{K}^{\mathcal{I}}(M)$ *a corresponding* $\mathcal{DL}$-*context. Then for finite* $\mathcal{C}, \mathcal{D} \subseteq M$*, the implication*

$$\mathcal{C} \rightarrow \mathcal{D}$$

holds in $\mathbb{K}^{\mathcal{I}}(M)$ *if and only if* [6]

$$\bigsqcap \mathcal{C} \sqsubseteq_{\mathcal{I}} \bigsqcap \mathcal{D}.$$

In the sequel, we will exploit this correspondence in the following way: employing the FCA exploration method allows us to collect all information that is valid in a (not explicitly given) interpretation and can be expressed by $\mathcal{DL}$ subsumptions with restricted maximal role depth[7].

6 The Relational Exploration Algorithm

The algorithm we present here is an iterative one. In each step the maximal role depth of the considered $\mathcal{DL}$ concept descriptions will be incremented by one. In each step, the results from previous steps will be exploited in several ways.

In the worst case, the time needed for the attribute exploration algorithm is exponential with respect to the number of attributes. Thus, it is essential to see how the set of attributes can be reduced without losing completeness.

The first exploration step is aimed at clarifying the implicational interdependencies of $\mathcal{DL}$ concept descriptions with quantifier depth 0. Therefore, no roles occur yet and we start with

$$M_0 := \begin{cases} \{\bot\} \cup \{\mathrm{A}, \neg\mathrm{A} \mid \mathrm{A} \in \mathcal{A}\} & \text{if } \mathcal{DL} = \mathcal{ALE} \\ \{\bot\} \cup \mathcal{A} & \text{otherwise.} \end{cases}$$

In the actual exploration step – the interview-like procedure described in Section 3 – takes place with respect to the context $\mathbb{K}_i^{\mathcal{I}} = \mathbb{K}_i^{\mathcal{I}}(M_i)$. Every hypothetical implication $\mathcal{A} \rightarrow \mathcal{B}$ for $\mathcal{A}, \mathcal{B} \subseteq M_i$ presented to the expert has to be interpreted as question about the validity of $\bigsqcap \mathcal{A} \sqsubseteq_{\mathcal{I}} \bigsqcap \mathcal{B}$, and will be passed to the "answering components" as described in Section 2.

[6] We use $\bigsqcap\{\mathrm{C}_1, \ldots, \mathrm{C}_n\}$ to abbreviate $\mathrm{C}_1 \sqcap \ldots \sqcap \mathrm{C}_n$. Moreover, let $\bigsqcap\{\mathrm{C}\} := \mathrm{C}$ and $\bigsqcap\{\emptyset\} := \top$.

[7] As usual, a concept description's role depth indicates how deep quantifiers are nested in it.

The exploration step ends up with an implication base $\mathfrak{IB}_i$, which – as we will prove in Section 7 – represents the complete subsumptional knowledge of the considered domain up to role depth i.

For the next exploration step – incrementing the considered role depth – we have to stipulate the next attribute set M_{i+1}. In case of the concept descriptions preceded by an existential quantification, the previously acquired implication base $\mathfrak{IB}_i$ can be used to reduce the number of attributes to consider, keeping the completeness property.

$$\begin{aligned} M_{i+1} := M_0 & \\ & \cup\{\forall \mathrm{R.C} \mid \mathrm{R} \in \mathcal{R}, \mathrm{C} \in M_i\} \\ & \cup\{\exists \mathrm{R}.\textstyle\bigsqcap \mathcal{C} \mid \mathrm{R} \in \mathcal{R}, \mathcal{C} = \mathcal{C}^{\mathfrak{IB}_i}, \bot \notin \mathcal{C}\} \end{aligned}$$

If considering $\mathcal{EL}$ or $\mathcal{FL}_0$, simply discard the second resp. third line from the definition. In addition to minimizing the cardinality of M_{i+1}, we can accelerate the exploration process by providing implications on M_{i+1} that are already known to be valid. These are the following:

- $\{\bot\} \to M_{i+1}$,
- $\{(\mathrm{A})_{i+1} \mid \mathrm{A} \in \mathcal{A}\} \to \{(\mathrm{B})_{i+1} \mid \mathrm{B} \in \mathcal{B}\}$ for every implication $\mathcal{A} \to \mathcal{B}$ from $\mathfrak{IB}_i$ (i.e., translate[8] all known implications from M_i into M_{i+1}),
- $\{\forall \mathrm{R.A} \mid \mathrm{A} \in \mathcal{A}\} \to \{\forall \mathrm{R.B} \mid \mathrm{B} \in \mathcal{B}\}$ for every implication $\mathcal{A} \to \mathcal{B}$ from $\mathfrak{IB}_i$,
- $\{\exists \mathrm{R}.\bigsqcap \mathcal{A}\} \to \{\exists \mathrm{R}.\bigsqcap \mathcal{B}\}$ for all $\mathfrak{IB}_i$-closed sets $\mathcal{A}, \mathcal{B} \subseteq M_i$ with $\mathcal{A} \subsetneq \mathcal{B}$ where there is no $\mathfrak{IB}_i$-closed set $\mathcal{C}$ with $\mathcal{A} \subsetneq \mathcal{C} \subsetneq \mathcal{B}$, and
- $\{\exists \mathrm{R}.\bigsqcap \mathcal{A}, \forall \mathrm{R.A}\} \to \{\exists \mathrm{R}.\bigsqcap (\mathcal{A} \cup \{\mathrm{A}\})^{\mathfrak{IB}_i}\}$ for every $\mathfrak{IB}_i$-closed set $\mathcal{A} \subseteq M_i \setminus \{\mathrm{A}\}$ and every concept description $\mathrm{A} \in M_i$.

With this attribute set M_{i+1} and the a-priori implications we start the next exploration step.

In theory, this procedure can be continued to arbitrary role depths. In some but not in all cases a complete acquisition of knowledge can be achieved. Yet in practice, with increasing role depth, the questions brought up by the exploration procedure will be increasingly numerous as well as less intuitional and thus difficult to answer for a human expert. So in many cases, one will restrict to small role depths.

7 Verification of the Algorithm

Let $\mathcal{DL}_i$ denote the set of all $\mathcal{DL}$ concept descriptions with maximal role depth i. Now we show a way how the validity of any subsumption on $\mathcal{DL}_i$ can be checked by using just the attribute sets $M_0, \ldots, M_i$ as well as the corresponding implication bases $\mathfrak{IB}_0, \ldots, \mathfrak{IB}_i$ on those sets. First, we will define functions that provide for any concept description $\mathrm{C} \in \mathcal{DL}_i$ a set of attributes $\mathcal{C} \subseteq M_i$ such that $C \equiv_{\mathcal{I}} \bigsqcap \mathcal{C}$. The following definitions and proofs are carried out for $\mathcal{ALE}$ but can be easily adapted to the other DLs by simply removing the irrelevant parts.

[8] We will formally define and justify this translation $(.)_{i+1}$ in Section 7.

Definition 6. *Let $\mathcal{I}$ be an interpretation and the corresponding sequences $(M_i), (\mathbb{K}_i^{\mathcal{I}})$ defined as above. Given the according sequence $\mathfrak{IB}_0, \ldots, \mathfrak{IB}_n$ of implication bases, we define a sequence of functions $\tau_i : \mathcal{DL}_i \to \mathcal{P}(\mathcal{DL}_i)$ in a recursive way:*

$$\begin{aligned}
\tau_i(\mathrm{C}) \quad &= \{\mathrm{C}\} \text{ for } \mathrm{C} \in M_0 \\
\tau_i(\textstyle\bigsqcap \mathcal{C}) \quad &= \textstyle\bigcup\{\tau_i(\mathrm{C}) \mid \mathrm{C} \in \mathcal{C}\} \\
\tau_i(\forall \mathrm{R.C}) &= \{\forall \mathrm{R.D} \mid \mathrm{D} \in \tau_{i-1}(\mathrm{C})\} \\
\tau_i(\exists \mathrm{R.C}) &= \begin{cases} \{\bot\} & \text{if } \bot \in (\tau_{i-1}(\mathrm{C}))^{\mathfrak{IB}_{i-1}}, \\ \{\exists \mathrm{R}.\bigsqcap(\tau_{i-1}(\mathrm{C}))^{\mathfrak{IB}_{i-1}}\} & \text{otherwise.} \end{cases}
\end{aligned}$$

Moreover, let $\bar{\tau}_i(\mathrm{C}) := (\tau_i(\mathrm{C}))^{\mathfrak{IB}_i}$ for all $\mathrm{C} \in \mathcal{DL}_i$.

Note that by this definition, we also have $\bar{\tau}_i(\top) = \bar{\tau}_i(\bigsqcap \emptyset) = \emptyset^{\mathfrak{IB}_i}$. Next, we have to show that the functions just defined behave in the desired way. The following lemma ensures that $\bar{\tau}_i$ and τ_i indeed map to M_i.

Lemma 1. *Suppose $\mathrm{C} \in \mathcal{DL}_i$. Then we have $\tau_i(\mathrm{C}) \subseteq M_i$ and $\bar{\tau}_i(\mathrm{C}) \subseteq M_i$.*

Proof. Obviously, $\bar{\tau}_i(\mathrm{C}) \subseteq M_i$ whenever $\tau_i(\mathrm{C}) \subseteq M_i$. We show the latter by induction on the role depth considering four cases:

- $\mathrm{C} \in \{\mathrm{A}, \neg\mathrm{A} \mid \mathrm{A} \in \mathcal{A}\} \cup \{\bot\}$. Then by definition $\mathrm{C} \in M_i$.
- $\mathrm{C} = \exists \mathrm{R.D}$. If $\bot \in \bar{\tau}_{i-1}(\mathrm{D})$, we get $\tau_i(\mathrm{C}) = \tau_i(\exists \mathrm{R.D}) = \{\bot\} \subseteq M_i$.
  Now suppose $\bot \not\in \bar{\tau}_{i-1}(\mathrm{D})$. As immediate consequence of the induction hypothesis we have $\bar{\tau}_{i-1}(\mathrm{D}) \subseteq M_{i-1}$. Since $\bar{\tau}_{i-1}$ gives an $\mathfrak{IB}_{i-1}$-closed set, we have also $\exists \mathrm{R}.\bigsqcap \bar{\tau}_{i-1}(\mathrm{D}) \in M_i$, as a look to the constructive definition of M_i immediately shows. Therefore, $\tau_i(\mathrm{C}) = \tau_i(\exists \mathrm{R.D}) = \{\exists \mathrm{R}.\bigsqcap \bar{\tau}_{i-1}(\mathrm{D})\} \subseteq M_i$
- $\mathrm{C} = \forall \mathrm{R.D}$. Again, our induction hypothesis yields $\tau_{i-1}(\mathrm{D}) \subseteq M_{i-1}$ which implies $\{\forall \mathrm{R.E} \mid \mathrm{E} \in \tau_{i-1}(\mathrm{D})\} \subseteq M_i$ due to the definition of M_i and therefore also $\tau_i(\mathrm{C}) = \tau_i(\forall \mathrm{R.D}) = \{\forall \mathrm{R.E} \mid \mathrm{E} \in \tau_{i-1}(\mathrm{D})\} \subseteq M_i$.
- $\mathrm{C} = \bigsqcap \mathcal{C}$. W.l.o.g., we presuppose that there is no conjunction outside the quantifier range in any $\mathrm{D} \in \mathcal{C}$. So we have $\tau_i(\mathrm{D}) \subseteq M_i$ due to the three cases above, and subsequently also $\tau_i(\mathrm{C}) = \tau_i(\bigsqcap \mathcal{C}) = \big(\bigcup\{\tau_i(\mathrm{D}) \mid \mathrm{C} \in \mathcal{C}\}\big) \subseteq M_i$. □

The next lemma and theorem show that in our fixed interpretation $\mathcal{I}$, for any concept description $\mathrm{C} \in \mathcal{DL}_i$, the entity sets fulfilling C on the one hand and $\bar{\tau}_i(\mathrm{C})$ as well as $\tau_i(\mathrm{C})$ on the other hand coincide.

Lemma 2. *For any $\mathcal{C} \subseteq M_i$, we have $\bigsqcap \mathcal{C} \equiv_{\mathcal{I}} \bigsqcap \mathcal{C}^{\mathfrak{IB}_i}$.*

Proof. First, observe $(\bigsqcap \mathcal{C})^{\mathcal{I}} = \bigcap\{(\mathrm{C})^{\mathcal{I}} \mid \mathrm{C} \in \mathcal{C}\} = \bigcap\{\mathrm{C}^{I_i} \mid \mathrm{C} \in \mathcal{C}\} = \{\delta \in \Delta^{\mathcal{I}} \mid \delta \in \mathrm{C}^{\mathcal{I}} \text{ for all } \mathrm{C} \in \mathcal{C}\}$. Now, consider $\mathbb{K}_i^{\mathcal{I}}$. Since $\mathfrak{IB}_i$ is an implication base of $\mathbb{K}_i^{\mathcal{I}}$, $\mathcal{C} \to \mathcal{C}^{\mathfrak{IB}_i}$ is an implication valid in $\mathbb{K}_i^{\mathcal{I}}$, ergo all objects of $\mathbb{K}_i^{\mathcal{I}}$ (being the individuals $\delta \in \Delta^{\mathcal{I}}$) fulfill $\mathcal{C} \subseteq \delta^{I_i} \Rightarrow \mathcal{C}^{\mathfrak{IB}_i} \subseteq \delta^{I_i}$. Therefore, one δ has all attributes from $\mathcal{C}$ exactly if it has all attributes from $\mathcal{C}^{\mathfrak{IB}_i}$. Finally, we have then $\{\delta \in \Delta^{\mathcal{I}} \mid \delta \in \mathrm{C}^{\mathcal{I}} \text{ for all } \mathrm{C} \in \mathcal{C}^{\mathfrak{IB}_i}\} = \bigcap\{\mathrm{C}^{\mathcal{I}} \mid \mathrm{C} \in \mathcal{C}^{\mathfrak{IB}_i}\} = (\bigsqcap \mathcal{C}^{\mathfrak{IB}_i})^{\mathcal{I}}$.

□

Theorem 2. *Let* $\mathrm{C} \in \mathcal{DL}_i$*. Then* $\mathrm{C} \equiv_{\mathcal{I}} \bigsqcap \tau_i(\mathrm{C}) \equiv_{\mathcal{I}} \bigsqcap \bar{\tau}_i(\mathrm{C})$.

Proof. The second equivalence is a direct consequence of Lemma 2. We show the first one again via induction on the role depth:

- $\mathrm{C} \in \{\mathrm{A}, \neg\mathrm{A} \mid \mathrm{A} \in \mathcal{A}\} \cup \{\bot\}$. Then, we trivially have $\mathrm{C}^{\mathcal{I}} = (\bigsqcap\{\mathrm{C}\})^{\mathcal{I}}$.
- $\mathrm{C} = \exists\mathrm{R.D}$. By induction hypothesis, we get $\mathrm{D}^{\mathcal{I}} = (\bigsqcap \bar{\tau}_{i-1}(\mathrm{D}))^{\mathcal{I}}$, therefore $(\exists\mathrm{R.D})^{\mathcal{I}} = (\exists\mathrm{R.} \bigsqcap \bar{\tau}_{i-1}(\mathrm{D}))^{\mathcal{I}}$ which by definition equals $(\bigsqcap \tau_i(\exists\mathrm{R.D}))^{\mathcal{I}}$.
- $\mathrm{C} = \forall\mathrm{R.D}$. Again, by induction hypothesis, we get $\mathrm{D}^{\mathcal{I}} = (\bigsqcap \bar{\tau}_{i-1}(\mathrm{D}))^{\mathcal{I}} = \bigcap\{\mathrm{E}^{\mathcal{I}} \mid \mathrm{E} \in \bar{\tau}_{i-1}(\mathrm{D})\}$. Now, observe that the statement $(\delta, \tilde{\delta}) \in \mathrm{R}^{\mathcal{I}} \to \tilde{\delta} \in \mathrm{D}^{\mathcal{I}}$ is equivalent to $\bigwedge_{\mathrm{E} \in \tau_{i-1}(\mathrm{D})} \big((\delta, \tilde{\delta}) \in \mathrm{R}^{\mathcal{I}} \to \tilde{\delta} \in \mathrm{E}^{\mathcal{I}}\big)$ and thus $(\forall\mathrm{R.D})^{\mathcal{I}} = \{\delta \mid (\delta, \tilde{\delta}) \in \mathrm{R}^{\mathcal{I}} \to \tilde{\delta} \in \bigcap\{\mathrm{D}^{\mathcal{I}}\}\} = \{\delta \mid \bigwedge_{\mathrm{E} \in \tau_{i-1}(\mathrm{D})} \delta \in (\forall\mathrm{R.E})^{\mathcal{I}}\} = \bigcap\{(\forall\mathrm{R.E})^{\mathcal{I}} \mid \mathrm{E} \in \tau_{i-1}(\mathrm{D})\} = (\bigsqcap\{\forall\mathrm{R.E} \mid \mathrm{E} \in \tau_{i-1}(\mathrm{D})\})^{\mathcal{I}}$ which by definition is just $(\bigsqcap \tau_i(\forall\mathrm{R.D}))^{\mathcal{I}}$.
- $\mathrm{C} = \bigsqcap\mathcal{C}$. Again, we can presume no conjunction outside the quantifier range in any $\mathrm{D} \in \mathcal{C}$. Then $(\bigsqcap\mathcal{C})^{\mathcal{I}} = \bigcap\{(\mathrm{D})^{\mathcal{I}} \mid \mathrm{D} \in \mathcal{C}\} = \bigcap\{(\bigsqcap \tau_i(\mathrm{D}))^{\mathcal{I}} \mid \mathrm{D} \in \mathcal{C}\}$ because of the cases shown before. Now, this is obviously the same as $\bigcap\{(\mathrm{E})^{\mathcal{I}} \mid \mathrm{E} \in \tau_i(\mathrm{D}), \mathrm{D} \in \mathcal{C}\} = (\bigsqcap(\bigcup\{\tau_i(\mathrm{D}) \mid \mathrm{D} \in \mathcal{C}\}))^{\mathcal{I}} = (\tau_i(\bigsqcap\mathcal{C}))^{\mathcal{I}}$. □

Using these propositions, we can easily provide a method to check – using only the closure operators $\mathfrak{IB}_0, \ldots, \mathfrak{IB}_i$ – the validity of any subsumption on $\mathcal{DL}_i$ with respect to a fixed (but not explicitly known) interpretation $\mathcal{I}$. It suffices to apply $\bar{\tau}_i$ on both sides and then check for inclusion.

Corollary 1. *Let* $\mathrm{C}_1, \mathrm{C}_2 \in \mathcal{DL}_i$*. Then* $\mathrm{C}_1 \sqsubseteq_{\mathcal{I}} \mathrm{C}_2$ *if and only if* $\bar{\tau}_i(\mathrm{C}_2) \subseteq \bar{\tau}_i(\mathrm{C}_1)$.

Proof. Due to Theorem 2, $\mathrm{C}_1 \sqsubseteq_{\mathcal{I}} \mathrm{C}_2$ is equivalent to $\bigsqcap \bar{\tau}_i(\mathrm{C}_1) \sqsubseteq_{\mathcal{I}} \bigsqcap \bar{\tau}_i(\mathrm{C}_2)$. According to Lemma 1, we have $\bar{\tau}_i(\mathrm{C}_1) \subseteq M_i$ and $\bar{\tau}_i(\mathrm{C}_2) \subseteq M_i$. In view of Theorem 1, this means the same as the validity of the implication $\bar{\tau}_i(\mathrm{C}_1) \to \bar{\tau}_i(\mathrm{C}_2)$ in $\mathbb{K}_i$. Now, since the application of $\bar{\tau}$ always gives a closed set with respect to all implications valid in $\mathbb{K}_i$, this is equivalent to $\bar{\tau}_i(\mathrm{C}_2) \subseteq \bar{\tau}_i(\mathrm{C}_1)$. □

Finally, consider the function τ_i from Definition 6. It is easy to see that for any $\mathrm{C} \in M_{i-1}$ by calculating $\tau_i(\mathrm{C})$ we get a singleton set $\{\mathrm{D}\}$ with $\mathrm{D} \in M_i$. We then have even $\mathrm{C} \equiv_{\mathcal{I}} \mathrm{D}$. For the sake of readability we will just write $\mathrm{D} = (\mathrm{C})_i$. Roughly spoken, D is just the "equivalent M_i-version" of C. Note that evaluating τ_i does not need the implication base $\mathfrak{IB}_i$ but only $\mathfrak{IB}_0, \ldots, \mathfrak{IB}_{i-1}$. So we have provided the translation function we promised in Section 6.

8 Conclusion

We have introduced an interactive knowledge acquisition technique for finding DL-style subsumption statements valid in a domain of interest. Its outstanding properties are

- minimal workload for the domain expert (i.e., no redundant questions will be posed) and

– completeness of the resulting specification (any statement from the exploration language is known to hold or not to hold).

Several current fields of AI can benefit from the results presented here.

Ontology engineering would be the first to mention. Since based on DL formalisms, our method can obviously contribute to the development and refinement of ontologies. RE can be used for an organized search for new GCIs[9] of a certain shape (namely those expressible by $\mathcal{DL}$ concept descriptions). Clearly, the description logics nowaday's ontology specifications are based on are much more complex than any of $\mathcal{DL}$. Nonetheless, our algorithm is still applicable since all of them incorporate the DLs considered as exploration language candidates. Hence, any of the existent reasoning algorithms for deciding subsumption (as for instance KAON2 [12] or FaCT [13], both capable of reasoning in $\mathcal{SHIQ}(D)$ – see [14]) can be used for the terminology part. All information beyond $\mathcal{DL}$ would then be treated as background knowledge and "hidden" from the exploration algorithm. As already pointed out, one major advantage of applying this technique is the guarantee that all valid axioms expressible as subsumption statements on $\mathcal{DL}$ with a certain role depth will certainly be found and specified.

Another topic RE can contribute to is machine learning. The supervised case corresponds almost directly to the RE algorithm – mostly one would have large data sets and (almost) empty theories in this setting. Yet also unsupervised machine learning can be carried out – by "short-circuiting" the expert such that every potential statement directed to him would be automatically confirmed. Essentially, the same would be the case for data mining tasks.

Finally, we are confident that an implementation of the RE algorithm will be a very helpful and versatile tool for eliciting information from various knowledge resources.

9 Future Work

So, as the very next step, we plan an implementation of the presented algorithm including interfaces for database querying as well as for DL reasoning. Applying this tool in the ontology engineering area will in turn enable us to investigate central questions concerning practical usability; in particular performance on real-life problems and scalability (being of unprecedented relevance in the Semantic Web Technologies sector), as well as issues concerning user acceptance will be of special interest for evaluation.

References

1. Rudolph, S.: An FCA method for the extensional exploration of relational data. In Ganter, B., de Moor, A., eds.: Using Conceptual Structures, Contributions to ICCS 2003, Dresden, Germany, Shaker, Aachen (2003) 197 – 210
2. Rudolph, S.: Exploring relational structures via FLE. In Wolff, K.E., Pfeiffer, H.D., Delugach, H.S., eds.: Conceptual Structures at Work: 12th International Conference on Conceptual Structures. Volume 3127 of LNCS., Huntsville, AL, USA, Springer (2004) 196 – 212

[9] I.e., GCIs not already logically entailed by the present specification.

3. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1997) Translator-C. Franzke.
4. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. J. Log. Program. **1** (1984) 267–284
5. Guigues, J.L., Duquenne, V.: Familles minimales d'implications informatives resultant d'un tableau de données binaires. Math. Sci Humaines **95** (1986) 5–18
6. Ganter, B.: Two basic algorithms in concept analysis. Technical Report 831, FB4, TH Darmstadt (1984)
7. Zickwolff, M.: Rule Exploration: First Order Logic in Formal Concept Analysis. PhD thesis, FB4, TH Darmstadt (1991)
8. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. In Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: Description Logic Handbook, Cambridge University Press (2003)
9. Prediger, S.: Terminologische Merkmalslogik in der formalen Begriffsanalyse. In Stumme, G., Ganter, B., eds.: Begriffliche Wissensverarbeitung: Methoden und Anwendungen, Springer (2000) 99–124
10. Baader, F.: Computing a minimal representation of the subsumption lattice of all conjunctions of concepts defined in a terminology. In: Proceedings of the International Symposium on Knowledge Retrieval, Use, and Storage for Efficiency, KRUSE 95, Santa Cruz, USA (1995) 168–178
11. Baader, F., Sertkaya, B.: Applying formal concept analysis to description logics. In: ICFCA. (2004) 261–286
12. Motik, B.: Reasoning in Description Logics using Resolution and Deductive Databases. PhD thesis, Universität Karlsruhe (TH), Germany (2006)
13. Horrocks, I.: Using an expressive description logic: FaCT or fiction? In Cohn, A.G., Schubert, L., Shapiro, S.C., eds.: Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR'98), Morgan Kaufmann Publishers, San Francisco, California (1998) 636–647
14. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for expressive description logics. In Ganzinger, H., McAllester, D., Voronkov, A., eds.: Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99). Number 1705, Springer-Verlag (1999) 161–180