Exploring Finite Models in the Description Logic $\mathcal{EL}_{\text{gfp}}$

Franz Baader* and Felix Distel**

Theoretical Computer Science, TU Dresden, Germany
{baader,felix}@tcs.inf.tu-dresden.de

Abstract. In a previous ICFCA paper we have shown that, in the Description Logics $\mathcal{EL}$ and $\mathcal{EL}_{\text{gfp}}$, the set of general concept inclusions holding in a finite model always has a finite basis. In this paper, we address the problem of how to compute this basis efficiently, by adapting methods from formal concept analysis.

1 Introduction

Description Logics (DLs) [3] are a well-investigated family of logic-based knowledge representation formalisms, which are employed in various application domains, such as natural language processing, configuration, databases, and bio-medical ontologies, but their most notable success so far is the adoption of the DL-based language OWL [11] as standard ontology language for the semantic web. From the Description Logic point of view, an ontology is a finite set of general concept inclusion axioms (GCIs) of the form $C \sqsubseteq D$, where C, D are concepts defined using an appropriate concept description language. Such a concept description language allows one to construct complex concepts out of concept names (unary predicates, interpreted as sets) and roles (binary predicates, interpreted as binary relations) using certain concept constructors. Complex concepts are again interpreted as sets. To be more precise, given an interpretation of the concept and role names, the semantics of the concept constructors determines, for every complex concept, a unique set as the extension of this concept. The GCI $C \sqsubseteq D$ states that, in a model of the ontology, the extension of the concept C must be a subset of the extension of the concept D.

When defining a DL-based ontology, one must first decide on which vocabulary (i.e., concept and role names) to use, and then define appropriate constraints on the interpretation of this vocabulary using GCIs. The work described in this paper is motivated by the fact that coming up with the right GCIs by hand is usually not an easy task. Instead, we propose an approach where the knowledge engineer is required to provide us with a finite model, which should be seen as an abstraction or approximation of the application domain to be modeled. We then automatically generate a finite basis of the GCIs holding in the model, i.e.,

* Supported by DFG under grant BA 1122/12-1.
** Supported by the Cusanuswerk.

S. Ferré and S. Rudolph (Eds.): ICFCA 2009, LNAI 5548, pp. 146–161, 2009.

a finite set of GCIs that hold in this model and from which all GCIs holding in the model and expressible in the employed concept description language follow. The knowledge engineer can use the computed basis as a starting point for the definition of the ontology. She may want to weaken or even remove some of the GCIs if the chosen model was too restricted, and thus satisfies GCIs that actually do not hold in all intended models. As an example, assume that we want to define a family ontology, using the concept names Male, Father, Female, Mother, and the role name child. Consider a finite model with two families. The first family consists of John, Michelle, and Mackenzie, where John is male and a father (i.e., John belongs to the interpretation of the concept names Male and Father), Michelle is female and a mother, and Mackenzie is female and a child of both John and Michelle. The second family consists of Paul, Linda, and James, where Paul is male and a father, Linda is female and a mother, and James is male and a child of both Paul and Linda. In this model, the GCIs

$$\mathsf{Father} \sqsubseteq \mathsf{Male} \sqcap \exists \mathsf{child}.\top \quad \text{and} \quad \mathsf{Mother} \sqsubseteq \mathsf{Female} \sqcap \exists \mathsf{child}.\top$$

hold. The first one says that every father is male and has a child, and the second one says that every mother is female and has a child. If we had used a model consisting of only the first family, then we would have obtained the too specific GCIs $\mathsf{Father} \sqsubseteq \mathsf{Male} \sqcap \exists \mathsf{child}.\mathsf{Female}$ and $\mathsf{Mother} \sqsubseteq \mathsf{Female} \sqcap \exists \mathsf{child}.\mathsf{Female}$, where mothers and fathers always have female children.

For the approach sketched above to work, the set of GCIs holding in a finite model and expressible in the employed concept description language must have a *finite* basis. Using methods from formal concept analysis (FCA), we have shown in [5] that this is the case for the language $\mathcal{EL}$, which allows for the concept constructors $\top$ (top concept), $C \sqcap D$ (conjunction), and $\exists r.C$ (existential restriction). Though being quite inexpressive, $\mathcal{EL}$ has turned out to be very useful for representing biomedical ontologies such as SNOMED [14] and the Gene Ontology [16]. A major advantage of using an inexpressive DL like $\mathcal{EL}$ is that it allows for efficient reasoning procedures [2,7]. Because of the nice algorithmic properties of $\mathcal{EL}$, the new OWL standard will contain a profile, called OWL 2 EL, that is based on $\mathcal{EL}$.

In [5], the existence of a finite basis is actually first shown for $\mathcal{EL}_{\text{gfp}}$, which extends $\mathcal{EL}$ with cyclic concept definitions interpreted with greatest fixpoint semantics. The advantage of using $\mathcal{EL}_{\text{gfp}}$ rather than $\mathcal{EL}$ is that, in $\mathcal{EL}_{\text{gfp}}$, every set of objects (i.e., elements of the domain of a given finite model) always has a most specific concept describing these objects. Going from a set of objects to its most specific concept corresponds to the $\cdot'$ operator in FCA, which goes from a set of objects in a formal context to the set of all attributes that these objects have in common. The existence of most specific concepts in $\mathcal{EL}_{\text{gfp}}$ thus allowed us to employ methods from FCA. In a second step, we have shown in [5] that the $\mathcal{EL}_{\text{gfp}}$-basis can be turned into an $\mathcal{EL}$-basis by unraveling cyclic concept definitions up to a level determined by the cardinality of the given finite model.

In [5], we concentrated on showing the *existence* of a finite basis for $\mathcal{EL}_{\text{gfp}}$ and $\mathcal{EL}$. Of course, if the approach for automatically generating GCIs sketched

above is to be used in practice, we also need to find efficient algorithms for computing such bases. This is the topic of the present paper. First, we show that the algorithm for computing an implication basis of a given formal context known from classical FCA can be adapted to our purposes. In contrast to the classical case, we cannot assume that all attributes of the context are known from the beginning. Instead, the set of attribute can be extended during the run of the algorithm. This is vital for obtaining an efficient algorithm. In a second step, we then extend this algorithm to an exploration algorithm. The advantage of this second algorithm is that it no longer requires the finite model to be completely represented in the computer from the beginning. As in the case of classical attribute exploration [9], the model is assumed to be "known" by an expert, who during the exploration process extends the represented part of the model in order to provide counterexamples to implication questions.

We concentrate on computing a finite $\mathcal{EL}_{\text{gfp}}$-basis since this basis can be turned into an $\mathcal{EL}$-basis as described in [5]. Due to the space limitation, we cannot give complete proofs of our results. They can be found in [4]. We also assume that the reader is familiar with the basic notion and results of formal concept analysis (FCA).

2 A Finite Implication Basis for $\mathcal{EL}_{\text{gfp}}$

We start by defining $\mathcal{EL}$, and show how it can be extended to $\mathcal{EL}_{\text{gfp}}$. Then we define most specific concepts in $\mathcal{EL}_{\text{gfp}}$, and show how they can be used to obtain a finite basis of the $\mathcal{EL}_{\text{gfp}}$-GCIs holding in a finite model.

The DLs $\mathcal{EL}$ and $\mathcal{EL}_{\text{gfp}}$

Because of the space restriction, we can only give a very compact introduction into these DLs (see [1] for more details). Concept descriptions of $\mathcal{EL}$ are built from a set $\mathcal{N}_c$ of concept names and a set $\mathcal{N}_r$ of role names, using the constructors top concept, conjunction, and existential restriction:

- concept names and the top concept $\top$ are $\mathcal{EL}$-concept descriptions;
- if C, D are $\mathcal{EL}$-concept descriptions and r is a role name, then $C \sqcap D$ and $\exists r.C$ are $\mathcal{EL}$-concept descriptions.

In the following, we assume that the sets $\mathcal{N}_c$ and $\mathcal{N}_r$ are finite. This assumption is reasonable since a finite ontology can contain only finitely many concept and role names.

Models of $\mathcal{EL}$ are pairs $(\Delta_i, \cdot^i)$, where Δ_i is a non-empty set, and $\cdot^i$ maps role names r to binary relations $r^i \subseteq \Delta_i \times \Delta_i$ and $\mathcal{EL}$-concept descriptions C to their *extensions* $C^i \subseteq \Delta_i$ such that

$$\top^i = \Delta_i, \qquad (C_1 \sqcap C_2)^i = C_1^i \cap C_2^i, \text{ and}$$
$$(\exists r.D)^i = \{d \in \Delta_i \mid \exists e \in D^i \text{ such that } (d, e) \in r^i\}.$$

Subsumption and equivalence between $\mathcal{EL}$-concept descriptions is defined in the usual way, i.e., C is subsumed by D (written $C \sqsubseteq D$) iff $C^i \subseteq D^i$ for all models i, and C is equivalent to D (written $C \equiv D$) iff $C \sqsubseteq D$ and $D \sqsubseteq C$.

$\mathcal{EL}_{\text{gfp}}$ is the extension of $\mathcal{EL}$ by cyclic concept definitions interpreted with greatest fixpoint (gfp) semantics. In $\mathcal{EL}_{\text{gfp}}$, we assume that the set of concept names is partitioned into the set $\mathcal{N}_{\text{prim}}$ of primitive concepts and the set $\mathcal{N}_{\text{def}}$ of defined concepts. A *concept definition* is of the form

$$B_0 \equiv P_1 \sqcap \ldots \sqcap P_m \sqcap \exists r_1.B_1 \sqcap \ldots \sqcap \exists r_n.B_n$$

where $B_0, B_1, \ldots, B_n \in \mathcal{N}_{\text{def}}$, $P_1, \ldots, P_m \in \mathcal{N}_{\text{prim}}$, and $r_1, \ldots, r_n \in \mathcal{N}_r$. The empty conjunction (i.e., $m = 0 = n$) stands for $\top$. A *TBox* is a finite set of concept definitions such that every defined concept occurs at most once as a left-hand side of a concept definition.

Definition 1 ($\mathcal{EL}_{\text{gfp}}$-concept description). *An $\mathcal{EL}_{\text{gfp}}$-concept description is a tuple $(A, \mathcal{T})$ where $\mathcal{T}$ is a TBox and A is a defined concept occurring on the left-hand side of a definition in $\mathcal{T}$.*

Models of $\mathcal{EL}_{\text{gfp}}$ are of the form $i = (\Delta_i, \cdot^i)$ where Δ_i is a non-empty set, and $\cdot^i$ maps role names r to binary relations $r^i \subseteq \Delta_i \times \Delta_i$ and primitive concepts to subsets of Δ_i. The mapping $\cdot^i$ is extended to $\mathcal{EL}_{\text{gfp}}$-concept descriptions $(A, \mathcal{T})$ by interpreting the TBox $\mathcal{T}$ with gfp-semantics: consider all extensions of i to the defined concepts that satisfy the concept definitions in $\mathcal{T}$, i.e., assign the same extension to the left-hand side and the right-hand side of each definition. Among these extensions of i, the *gfp-model of $\mathcal{T}$ based on i* is the one that assigns the largest sets to the defined concepts (see [1] for a more detailed definition of gfp-semantics). The *extension $(A, \mathcal{T})^i$ of $(A, \mathcal{T})$* in i is the set assigned to A by the gfp-model of $\mathcal{T}$ based on i.

Subsumption and equivalence between $\mathcal{EL}_{\text{gfp}}$-concept descriptions is defined as in the case of $\mathcal{EL}$-concept descriptions. It is easy to see that acyclic $\mathcal{EL}_{\text{gfp}}$-concept descriptions (i.e., ones where the TBox component is acyclic) correspond exactly to $\mathcal{EL}$-concept descriptions. This shows that $\mathcal{EL}$ can indeed be seen as a sublanguage of $\mathcal{EL}_{\text{gfp}}$. In the following, we will not distinguish an acyclic $\mathcal{EL}_{\text{gfp}}$-concept description from its equivalent $\mathcal{EL}$-concept description.

Most Specific Concepts in $\mathcal{EL}_{\text{gfp}}$

In FCA, the prime operators $\cdot'$ play an important rôle. Given a set of attributes B, the set B' consists of the objects of the given context satisfying all these attributes. In DL, the operator $\cdot^i$ plays a similar rôle: given a concept description C, the set C^i consists of all objects in the model i (i.e., elements of Δ_i) satisfying C, i.e., belonging to the extension of C. In FCA, the prime operator can also be applied in the other direction: given a set of objects A, it yields the set A' of attributes common to the objects in A. This is equivalent to defining $A' = B_{\max}$, where $B_{\max}$ is the greatest subset of M such that $A \subseteq B'_{\max}$. In DL, the most specific concept plays the rôle of this $\cdot'$ operator.

Definition 2 (Most specific concept). *Let i be a finite $\mathcal{EL}_{\text{gfp}}$-model and $X \subseteq \Delta_i$. The $\mathcal{EL}_{\text{gfp}}$-concept description C is the* most specific $\mathcal{EL}_{\text{gfp}}$-concept of X in i *if it is the least $\mathcal{EL}_{\text{gfp}}$-concept description such that $X \subseteq C^i$. By* least $\mathcal{EL}_{\text{gfp}}$-concept description *we mean that every other $\mathcal{EL}_{\text{gfp}}$-concept description $\bar{C}$ satisfying $X \subseteq \bar{C}^i$ also satisfies $C \sqsubseteq \bar{C}$.*

Calling an $\mathcal{EL}_{\text{gfp}}$-concept description satisfying the above definition *the* most specific $\mathcal{EL}_{\text{gfp}}$-concept of X in i is justified by the fact that most specific concepts are obviously unique up to equivalence. In [5] it is shown that, for $\mathcal{EL}_{\text{gfp}}$, the most specific concept always exists.[1]

Theorem 1. *For any finite $\mathcal{EL}_{\text{gfp}}$-model i and any set $X \subseteq \Delta_i$, the most specific $\mathcal{EL}_{\text{gfp}}$-concept of X in i exists and can be computed effectively.*

In the following, we denote the most specific $\mathcal{EL}_{\text{gfp}}$-concept of X in i by X^i. This overloading of the notation $\cdot^i$ corresponds to the one employed in FCA for $\cdot'$. The following lemma (taken from [5]) shows that the operators $\cdot^i$ indeed behave similarly to the $\cdot'$ operators.

Lemma 1. *Let $\mathcal{L}$ be a language for which X^i exists for every $X \subseteq \Delta_i$ and every $i \in \mathcal{I}$. Let $i \in \mathcal{I}$ be an interpretation, $X, Y \in \Delta_i$ sets of objects and C, D be concept descriptions. Then the following statements hold*

1. $X \subseteq Y \Rightarrow X^i \sqsubseteq Y^i$
2. $C \sqsubseteq D \Rightarrow C^i \subseteq D^i$
3. $X \subseteq X^{ii}$
4. $C^{ii} \sqsubseteq C$
5. $X^i \equiv X^{iii}$
6. $C^i = C^{iii}$
7. $X \subseteq C^i \Leftrightarrow X^i \sqsubseteq C$.

The Set of GCIs Holding in a Finite Model and a Basis for this Set

An expression of the form $C \rightarrow D$, where C, D are $\mathcal{EL}_{\text{gfp}}$-concept descriptions, is called an $\mathcal{EL}_{\text{gfp}}$-GCI (or simply GCI).[2] We say that an GCI $C \rightarrow D$ *holds* in the model i iff $C^i \subseteq D^i$. Given a set of GCIs $\mathcal{B}$, we say that the GCI $C \rightarrow D$ *follows from* $\mathcal{B}$ iff $C \rightarrow D$ holds in all models in which all implications from $\mathcal{B}$ hold.

Definition 3 (Basis). *For a given finite model i we say that a set of $\mathcal{EL}_{\text{gfp}}$-GCIs $\mathcal{B}$ is a* basis for the $\mathcal{EL}_{\text{gfp}}$-GCIs holding in i *if $\mathcal{B}$ is*

- sound *for i, i.e., it contains only $\mathcal{EL}_{\text{gfp}}$-GCIs holding in i, and*
- complete *for i, i.e., any $\mathcal{EL}_{\text{gfp}}$-GCI that holds in i follows from $\mathcal{B}$.*

The following lemma, taken from [5], shows that GCIs of the form $C \rightarrow C^{ii}$ play a special rôle.

[1] Note that this is not true if we use $\mathcal{EL}$ instead of $\mathcal{EL}_{\text{gfp}}$ (see [5] for an example).

[2] GCI is an abbreviation for "general concept inclusion." In DL, GCIs are usually written as $C \sqsubseteq D$. Here, we prefer to use the arrow notation to emphasize the connection to implications in FCA and to avoid confusion with subsumption statements.

Lemma 2. *Let C, D be $\mathcal{EL}_{\text{gfp}}$-concept descriptions and i a finite $\mathcal{EL}_{\text{gfp}}$-model. Then*

- *$C \to C^{ii}$ holds in i, and*
- *if $C \to D$ holds in i, then $C \to D$ follows from $\{C \to C^{ii}\}$.*

This lemma reinforces the similarity between the $\cdot'$ operators from FCA and our $\cdot^i$ operators. In fact, in FCA a basis of all implications holding in a finite context can be obtained by taking all implications $P \to P''$ where P is a so-called pseudo-intent of the context (see Section 3 below). Following the lead of FCA, we thus need to determine which $\mathcal{EL}_{\text{gfp}}$-concept descriptions can play the rôle of pseudo-intents, i.e., we want to find a *finite* set Λ_i of left-hand sides for GCI such that the set of GCIs $C \to C^{ii}$ for $C \in \Lambda_i$ is a basis for the $\mathcal{EL}_{\text{gfp}}$-GCIs holding in i.

Before we can define such a set, we need to introduce one more notation. Given a finite set U of $\mathcal{EL}_{\text{gfp}}$-concept descriptions, $\bigsqcap U := \bigsqcap_{C \in U} C$ denotes their conjunction. The set Λ_i will be obtained as the set of all such conjunctions for subsets of a basic set M_i.

Definition 4. *Let i be a finite $\mathcal{EL}_{\text{gfp}}$-model. The sets M_i, Λ_i are defined as*

$$M_i := \mathcal{N}_{\text{prim}} \cup \{\exists r.X^i \mid r \in \mathcal{N}_r \text{ and } X \subseteq \Delta_i\} \quad \text{and} \quad \Lambda_i := \{\bigsqcap U \mid U \subseteq M_i\}.$$

Since $\mathcal{N}_{\text{prim}}$, $\mathcal{N}_r$, and Δ_i are finite, M_i and Λ_i are finite as well. Thus, the basis introduced in the next theorem is finite as well.

Theorem 2. *The set of GCIs $\mathcal{B}_i := \{C \to C^{ii} \mid C \in \Lambda_i\}$ is a finite basis for the $\mathcal{EL}_{\text{gfp}}$-GCIs holding in i.*

This basis actually differs from the one defined in [5]. However, the proof that this is indeed a basis for the $\mathcal{EL}_{\text{gfp}}$-GCIs holding in i is very similar to the one given in [5] for the basis introduced there.

The definition of $\mathcal{B}_i$ also provides us with a brute-force method for computing this basis. To compute M_i, all we have to do is consider the (finitely many) subsets X of Δ_i, and compute their most specific concepts. The set Λ_i is then obtained by considering all subsets of M_i, and $\mathcal{B}_i$ is obtained from the elements C of Λ_i by first computing their extensions in i, and then building the most specific concepts of these extensions.

This brute-force approach has two disadvantages. First, up to equivalence of $\mathcal{EL}_{\text{gfp}}$-concept descriptions, the set $\{X^i \mid X \subseteq \Delta_i\}$ may be considerably smaller than the powerset of Δ_i. In fact, not every subset of Δ_i needs to be an extension of an $\mathcal{EL}_{\text{gfp}}$-concept description, and thus different subsets of Δ_i may have the same most specific concept. Second, we also want to be able to deal with a situation where the model i is not explicitly given, but rather "known" to an expert. Similar to the case of attribute exploration in FCA, we then want to elicit enough information about i from the expert to be able to compute a basis, but without having to ask too many questions. In this situation, neither all subsets

of Δ_i nor their most specific concepts can be assumed to be known/computable at the beginning of the exploration process.

In order to obtain a more practical algorithm for computing a basis, we will view the set M_i as the set of attributes in a classical formal context induced by the model i. In the next section, we define this induced context and state some interesting connections between the $\cdot'$ operations in this context and the $\cdot^i$ operations defined in the present section. Basically, we want to apply to the induced context the classical FCA algorithm for computing an implication basis. However, there are two differences compared to the classical case. First, we cannot assume that all the attributes (i.e., all the elements of M_i) are known from the beginning. Second, since our attributes are $\mathcal{EL}_{\text{gfp}}$-concept descriptions, we can use the known subsumption algorithm for this DL [1] to obtain background knowledge about relationships between these attributes. Thus, we use an algorithm for computing an implication basis that can handle background knowledge [15], and extend it such that it can deal with a growing set of attributes.

3 Formal Concept Analysis

Because of space constraints, we cannot give an introduction into FCA here. We thus assume that the reader is familiar with basic notions such as formal contexts; attributes and objects; the $\cdot'$ operators; intents, extents, and pseudo-intents; and implications and implication bases (see, e.g., [10]). At several points in this paper we use the so-called *Next-Closure Algorithm*, which can also be found in [10]. Recall that a total order on a finite set of attributes M induces the so-called *lectic order*, which is a total order on the powerset of M. Given a set of attributes U and a set of implications $\mathcal{B}$, the Next-Closure Algorithm computes the lectically smallest set of attributes V that is closed with respect to $\mathcal{B}$ (i.e., respects all implications in $\mathcal{B}$) and lectically greater than U.

Background Knowledge and Growing Sets of Attributes

We adopt Stumme's approach for handling background knowledge [15], where the background knowledge is given by a set of implications holding in the context under consideration. We say that a set of implications $\mathcal{B}$ is an *implication basis for the context* $\mathbb{K}$ *w.r.t. the set of background implications* $\mathcal{S}$ if $\mathcal{B} \cup \mathcal{S}$ is a sound and complete set of implications for $\mathbb{K}$. As in the case without background knowledge, pseudo-intents provide us with the left-hand sides of such a basis. Given a set $\mathcal{S}$ of background implications, the notion of a pseudo-intent is extended as follows.

Definition 5. *Let (G, M, I) be a formal context and $\mathcal{S}$ a set of implications holding in (G, M, I). The set $P \subseteq M$ is called* $\mathcal{S}$-pseudo-intent *if P respects all implications in $\mathcal{S}$ and $Q'' \subseteq P$ holds for every $\mathcal{S}$-pseudo-intent $Q \subsetneq P$.*

Stumme shows that this notion of pseudo-intents yields a minimal implication basis w.r.t. the background knowledge. To be more precise, he proves that the following holds for the set of implications

$$\mathcal{B}_{\mathcal{S}} := \{P \to P'' \mid P \text{ is } \mathcal{S}\text{-pseudo-intent in } \mathbb{K}\}$$

Algorithm 1. Construction of an implication basis w.r.t. background knowledge for the case of a growing set of attributes

1: Input: $\mathbb{K}_0 = (G, M_0, I_0)$, $\mathcal{S}_0$
2: $\mathit{\Pi}_0 := \emptyset$, $P_0 := \emptyset$, $k := 0$
3: **while** $P_k \neq \mathsf{null}$ **do**
4: $\quad \mathit{\Pi}_{k+1} := \mathit{\Pi}_k \cup \{P_k\}$
5: $\quad k := k + 1$
6: $\quad$ Input: $\mathbb{K}_k = (G, M_k, I_k)$, $\mathcal{S}_k$
7: $\quad$ **if** $M_k = M_{k-1} = P_k$ **then**
8: $\quad\quad P_k := \mathsf{null}$
9: $\quad$ **else**
10: $\quad\quad P_k :=$ lectically smallest set of attributes that is
– closed with respect to $\{P_j \to P_j^{\prime\prime_k} \mid P_j \in \mathit{\Pi}_k\}$ and $\mathcal{S}_k$, and
– lectically larger than P_{k-1}.
11: $\quad$ **end if**
12: **end while**

- $\mathcal{B}_{\mathcal{S}}$ is an implication basis for $\mathbb{K}$ w.r.t. $\mathcal{S}$, and
- $\mathcal{B}_{\mathcal{S}}$ has minimal cardinality among all implication bases for $\mathbb{K}$ w.r.t. $\mathcal{S}$.

Algorithm 1 looks at a setting where the set of objects is fixed, while the set of attributes as well as the background knowledge can grow. It starts with a context $\mathbb{K}_0 = (G, M_0, I_0)$ and a set of background implications $\mathcal{S}_0$ that hold in $\mathbb{K}_0$. In each step, new attributes and new background implications may be added by the user, thus yielding a new context $\mathbb{K}_k = (G, M_k, I_k)$ and an new implication set $\mathcal{S}_k$. We require for all $k \geq 1$ that (i) $M_{k-1} \subseteq M_k$; (ii) I_k agrees with I_{k-1} on M_{k-1}, i.e., for all $g \in G$ and for all $m \in M_{k-1}$ we have $(g, m) \in I_k$ iff $(g, m) \in I_{k-1}$; (iii) $\mathcal{S}_{k-1} \subseteq \mathcal{S}_k$; (iv) the implications of $\mathcal{S}_k$ hold in $\mathbb{K}_k$. The Next-Closure Algorithm used in line 10 of the algorithm requires a total order on the set of attributes. We assume that the total order on M_k extends the one on M_{k-1} such that $a < b$ for all $a \in M_{k-1}$ and $b \in M_k \setminus M_{k-1}$. To make clear which context we are referring to when using the prime operators, we add the index of the context; e.g., $A^{\prime\prime_k}$ is used to denote the set obtained from A by applying the prime operator of the context $\mathbb{K}_k$ twice.

It is easy to see that Algorithm 1 terminates if, and only if, from some point on the set of attributes is no longer extended. Now, assume that the algorithm has terminated after the n-th step. We want to show that the set of implications

$$\mathcal{B}_{\mathcal{S}_n}^{(n)} := \{P_j \to P_j^{\prime\prime_n} \mid P_j \in \mathit{\Pi}_n\}$$

is an implication basis for the final context $\mathbb{K}_n$ w.r.t. the final set of background implications $\mathcal{S}_n$. To prove this, we first need to show that the set of left-hand sides $\mathit{\Pi}_n$ "covers" all the quasi-closed sets of attributes for $\mathbb{K}_n$. A set of attributes U is called *quasi-closed* for a context $\mathbb{K}$ iff, for all subsets $V \subseteq U$, it holds that either $V'' \subseteq U$ or $V'' = U''$.

Lemma 3. *If Q is a set of attributes that is quasi-closed for $\mathbb{K}_n$ and respects all the background implications in $\mathcal{S}_n$, then there is some $P \in \Pi_n$ such that $P \subseteq Q$ and $P''^n = Q''^n$.*

It is a well-known fact that all pseudo-intents are quasi-closed [8]. Likewise, we can show that all $\mathcal{S}_n$-pseudo-intents are quasi-closed for $\mathbb{K}_n$ [4]. In addition, $\mathcal{S}_n$-pseudo-intents by definition respect all implications of $\mathcal{S}_n$. Thus, Stumme's result implies completeness of $\{Q \to Q''^n \mid Q \text{ is quasi-closed in } \mathbb{K}_n \text{ and respects all implications of } \mathcal{S}\} \cup \mathcal{S}$. Obviously, if $P \subseteq Q$ and $P''^n = Q''^n$, then the implication $P \to P''^n$ has the implication $Q \to Q''^n$ as a consequence. Thus, Lemma 3 yields completeness of $\{P \to P''^n \mid P \in \Pi_n\} \cup \mathcal{S}$.[3]

Theorem 3. *Assume that Algorithm 1 has terminated after the n-th step. Then $\mathcal{B}^{(n)}_{\mathcal{S}_n}$ is an implication basis for $\mathbb{K}_n$ w.r.t. $\mathcal{S}_n$.*

Note that, in contrast to the case of fixed set of attributes, in step k we must add P_k to the set of left-hand sides even if P_k is an intent of $\mathbb{K}_k$, i.e., $P_k = P_k''^k$. This is so because it might happen that $P_k = P_k''^k$, but $P_k \neq P_k''^n$ because the attributes in $P_k''^n \setminus P_k''^k$ have only been added at a later point.

The Induced Context

What we call induced contexts in this work are formal contexts whose attributes are concept descriptions and whose set of objects is the domain of a finite model i. In such a context, an object x has an attribute C if x is in the extension of the concept C in the model i. Similar contexts have been introduced in [12,13]. In the following, we examine the connection between the $\cdot'$-operators in the induced context and the $\cdot^i$-operators in the model i. Induced contexts establish the connection between the DL world and the FCA world which we need for the algorithms introduced in the next section. But let us first give a more formal definition of the induced context for the cases of $\mathcal{EL}_{\text{gfp}}$.[4]

Definition 6 (induced context). *Let i be a finite $\mathcal{EL}_{\text{gfp}}$-model and M a finite set of $\mathcal{EL}_{\text{gfp}}$-concept descriptions. The* context induced by M and i *is the formal context $\mathbb{K} = (G, M, I)$, where $G = \Delta_i$ and $I = \{(x, C) \mid C \in M \text{ and } x \in C^i\}$.*

In FCA, an object is in the extension of a set of attributes U iff it has all the attributes from U. In DL terms, this means that x is in the extension of the conjunction over all elements of U. Thus, the set of attributes $U \subseteq M$ corresponds to the concept $\bigsqcap_{C \in U} C$. In the other direction, we can approximate an arbitrary concept description C by the set of all attributes $D \in M$ that subsume C. Since M in general contains only a small number of concept descriptions, this is really

[3] Note that soundness is trivial since it is well-known that implications $P \to P''$ hold in the context that defines the prime operators used.

[4] Note, however, that the definitions and results given here do not really depend on $\mathcal{EL}_{\text{gfp}}$. They hold for any concept description language in which the most specific concept exists.

just an approximation, i.e., the conjunction of these concepts D may strictly subsume C.

Definition 7. *Let $\mathbb{K}$ be the context induced by M and i, C an $\mathcal{EL}_{\text{gfp}}$-concept description and $U \subseteq M$. We define $\text{pr}_{\mathbb{K}}(C) := \{D \in M \mid C \sqsubseteq D\}$, and call this the* projection of C to $\mathbb{K}$. *Conversely, we define $\bigsqcap U := \bigsqcap_{D \in U} D$, and call this the* concept defined by U. *We say that C* can be expressed in terms of M *iff there is some $V \subseteq M$ such that $C \equiv \bigsqcap V$.*

As an immediate consequence of this definition we obtain that the mappings $C \mapsto \text{pr}_{\mathbb{K}}(C)$ and $U \mapsto \bigsqcap U$ are antitonic:

- $C \sqsubseteq D$ implies $\text{pr}_{\mathbb{K}}(D) \subseteq \text{pr}_{\mathbb{K}}(C)$,
- $U \subseteq V$ implies $\bigsqcap V \sqsubseteq \bigsqcap U$.

In general, not all $\mathcal{EL}_{\text{gfp}}$-concept descriptions can be expressed in terms of M. Therefore, it is quite obvious that information is lost when we make the transformation from a concept description to the corresponding attribute set and back. This is the reason why, in the following lemma, we only have subsumption and subset relationships rather than equivalence and equality relationships.

Lemma 4. *Let $\mathbb{K}$ be the context induced by M and i, C an $\mathcal{EL}_{\text{gfp}}$-concept description, and $U \subseteq M$. Then the following statements hold:*

1. $C \sqsubseteq \bigsqcap \text{pr}_{\mathbb{K}}(C)$
2. $\text{pr}_{\mathbb{K}}(C)'' \subseteq \text{pr}_{\mathbb{K}}(C^{ii})$
3. $U \subseteq \text{pr}_{\mathbb{K}}(\bigsqcap U)$
4. $(\bigsqcap U)^{ii} \sqsubseteq \bigsqcap U''$

If a concept description is expressible in terms of M, then no information is lost by the conversion to the corresponding attribute set. This is the reason why, under additional expressibility conditions, the subsumption and subset relationships of the above lemma can be turned into equivalence and equality relationships.

Lemma 5. *Let C be an $\mathcal{EL}_{\text{gfp}}$-concept description and $U \subseteq M$ a set of attributes such that both C and $(\bigsqcap U)^{ii}$ can be expressed in terms of M. Then the following statements hold:*

1. $C \equiv \bigsqcap \text{pr}_{\mathbb{K}}(C)$
2. $\text{pr}_{\mathbb{K}}(C^{ii}) = \text{pr}_{\mathbb{K}}(C)''$
3. $\bigsqcap U'' \equiv (\bigsqcap U)^{ii}$

4 Computing a Basis for the $\mathcal{EL}_{\text{gfp}}$-GCIs Holding in a Finite $\mathcal{EL}_{\text{gfp}}$-Model

First, we consider the case where the finite model i is given right from the beginning. In this case, we basically apply Algorithm 1 to the context induced by M_i (see Definition 4) and i. In a second step, we extend the algorithm obtained this way to a model exploration algorithm, which can deal with the case where the model i is not explicitly given, but rather "known" to an expert.

Algorithm 2. Computing a basis for an a priori given model i

Input: finite model $i = (\Delta_i, \cdot^i)$
$M_0 := \mathcal{N}_{\text{prim}}$, $\mathbb{K}_0 :=$ the context induced by M_0 and i, $\mathcal{S}_0 := \emptyset$
$\Pi_0 := \emptyset$, $P_0 := \emptyset$, $k := 0$
while $P_k \neq \mathsf{null}$ **do**
  $\Pi_{k+1} := \Pi_k \cup \{P_k\}$
  $M_{k+1} := M_k \cup \{\exists r.(\bigsqcap P_k)^{ii} \mid r \in \mathcal{N}_r\}$
  $\mathcal{S}_{k+1} := \{\{C\} \to \{D\} \mid C, D \in M_k, C \sqsubseteq D\}$
  $k := k + 1$
  if $M_k = M_{k-1} = P_k$ **then**
    $P_k := \mathsf{null}$
  else
    $P_k :=$ lectically next set of attributes that respects all implications in $\{P_j \to P_j''^{k} \mid 1 \leq j < k\}$ and $\mathcal{S}_k$
  end if
end while

The Case of an A Priori Given Model

Let i be a finite $\mathcal{EL}_{\text{gfp}}$-model. Recall that the basis $\mathcal{B}_i$ introduced in Section 2 is the set of all implications $C \to C^{ii}$ where the left-hand sides C are of the form $C = \bigsqcap U$ for some subset U of

$$M_i = \mathcal{N}_{\text{prim}} \cup \{\exists r.X^i \mid r \in \mathcal{N}_r \text{ and } X \subseteq \Delta_i\}.$$

Therefore, it is natural to look at the induced context for the attribute set M_i. The elements of M_i are $\mathcal{EL}_{\text{gfp}}$-concept descriptions, and thus there may be subsumption relationships between them, which can be computed using the known polynomial-time subsumption algorithm for $\mathcal{EL}_{\text{gfp}}$ [1]. We will use these subsumption relationships as background knowledge. Obviously, if $C \sqsubseteq D$ for $\mathcal{EL}_{\text{gfp}}$-concept descriptions $C, D \in M_i$, then the GCI $C \to D$ holds in i, and thus the implication $\{C\} \to \{D\}$ holds in the context induced by M_i and i.

Since Algorithm 1 allows for a growing set of attributes, we do not start with the whole set M_i. Instead, we start with the set $\mathcal{N}_{\text{prim}}$ of primitive concepts, and then extend the current set of attributes by adding $\mathcal{EL}_{\text{gfp}}$-concept descriptions of the form $\exists r.X^i$ whenever a new set of objects X is obtained as the extension of a concept $\bigsqcap P$ for an already computed left-hand side P. Algorithm 2 shows the instance of Algorithm 1 obtained this way.

Algorithm 2 always terminates since there are only finitely many attributes that can be added. In fact, every attribute that is added is an element of M_i, and we have already shown in Section 2 that M_i is finite. Now, assume that Algorithm 2 has terminated after the nth step. Then the algorithm has generated a set Π_n of subsets of $M_n \subseteq M_i$. This set Π_n gives rise to the following set of GCIs:

$$\mathcal{B}_n := \{\bigsqcap P_k \to (\bigsqcap P_k)^{ii} \mid P_k \in \Pi_n\}.$$

Theorem 4. *Assume that Algorithm 2 terminates after the n-th step. Then $\mathcal{B}_n$ is a finite basis for the $\mathcal{EL}_{\text{gfp}}$-GCIs holding in i.*

Outline of the proof: Obviously, $\mathcal{B}_n$ is finite. In addition, since $\mathcal{B}_n$ is a subset of $\mathcal{B}_i$, we know that it is sound. Thus, to show that $\mathcal{B}_n$ is a finite basis for the $\mathcal{EL}_{\text{gfp}}$-GCIs holding in i, it is enough to show completeness, i.e., any $\mathcal{EL}_{\text{gfp}}$-GCI that holds in i follows from $\mathcal{B}_n$. Completeness can be proved in two steps. The first step is to show that, up to equivalence, M_n contains all attributes of the form $\exists r.X^i$ for $X \subseteq \Delta_i$. The second step then uses this fact to actually prove completeness of $\mathcal{B}_n$. *Step 1* is again divided into two parts.

(a) For a set of attributes $U \subseteq M_n$, we consider its closure U''^n under the double-prime operator $\cdot''^n$ of the context $\mathbb{K}_n$. As an intent of $\mathbb{K}_n$, U''^n is closed under $\cdot''^n$, and it respects any implication that holds in $\mathbb{K}_n$. Hence it is quasi-closed and respects all the implications of $\mathcal{S}_n$. Therefore, Lemma 3 ensures that there is some $P_k \in \Pi_n$ such that $P_k \subseteq U''^n$ and $P_k''^n = U''^n$. After the k-th step of the algorithm, all attributes of the form $\exists r.(\bigsqcap P_l)^{ii}$, where $0 \leq l \leq k$, have been added to the set of attributes. Using Lemma 4 and 5, it is possible to prove that $(\bigsqcap P_k)^{ii} \equiv (\bigsqcap U)^{ii}$ (see [4] for details). This shows that, up to equivalence, for every set $U \subseteq M_n$ the descriptions $\exists r.(\bigsqcap U)^{ii}$ must be in M_n.

(b) The fact that M_n contains all attributes of the form $\exists r.X^i$ for $X \subseteq \Delta_i$ can now be proved by induction on the depth of X^i, where we say that X^i has depth d iff d is the least role depth of $\mathcal{EL}$-concept descriptions D such that $X^i = D^{ii}$. In [5] it is shown that this notion of a depth is indeed well-defined. The base case is easy. In fact, if X^i has depth 0, then it can be written as conjunction of primitive concepts, i.e., $X^i = (\bigsqcap U)^{ii}$ for $U \subseteq M_0 \subseteq M_n$. But then it follows from (a) that M_n contains an attribute that is equivalent to $\exists r.(\bigsqcap U)^{ii} = \exists r.X^i$. The step case is very similar, except that one has to show that every X^i of role depth d can be written as the conjunction of primitive concept names and concept descriptions of the form $\exists r.Y^i$ where Y^i has depth less than d (details can be found in [4]).

Step 2. By Theorem 3, we know that the set $\mathcal{S} \cup \{P \to P''^n \mid P \in \Pi_n\}$ is a basis for the implications in $\mathbb{K}_n$. Let $L \in \Lambda_i$ be a premise of some implication from the basis $\mathcal{B}_i$ that is not an intent w.r.t. i, i.e., $L \neq L^{ii}$. We can show that not only L, but also L^{ii} belongs to Λ_i, and thus both can be expressed in terms of M_n, as shown in Step 1. Lemma 4 can be used to derive $\mathrm{pr}_{\mathbb{K}_n}(L) \neq \mathrm{pr}_{\mathbb{K}_n}\left(L^{ii}\right) = \mathrm{pr}_{\mathbb{K}_n}(L)''^n$. Consequently, $\mathrm{pr}_{\mathbb{K}_n}(L)$ is not an intent of $\mathbb{K}_n$, and hence there must be an implication $P_k \to P_k''^n$ for $P_k \in \Pi_n$ that $\mathrm{pr}_{\mathbb{K}_n}(L)$ does not respect, i.e., $P_k \subseteq \mathrm{pr}_{\mathbb{K}_n}(L)$, but $P_k''^n \not\subseteq \mathrm{pr}_{\mathbb{K}_n}(L)$. But then Lemma 4 implies that $L \sqsubseteq \bigsqcap P_k$, but $L \not\sqsubseteq (\bigsqcap P_k)^{ii}$.

Thus, for every concept description $L \in \Lambda_i$ that is not an intent w.r.t. i, there is some $P_k \in \Pi_n$ such that $L \sqsubseteq \bigsqcap P_k$, but $L \not\sqsubseteq (\bigsqcap P_k)^{ii}$. Since $\bigsqcap P_k \to (\bigsqcap P_k)^{ii}$ belongs to $\mathcal{B}_n$, the GCI $L \to L \sqcap (\bigsqcap P_k)^{ii}$ follows from $\mathcal{B}_n$. Since $L \not\sqsubseteq (\bigsqcap P_k)^{ii}$, the concept description $L \sqcap (\bigsqcap P_k)^{ii}$ is strictly subsumed by L, and it can be shown that $L \sqcap (\bigsqcap P_k)^{ii} \in \Lambda_i$. If $L \sqcap (\bigsqcap P_k)^{ii}$ is not an intent, then we can use the same argument, and find $P_l \in \Pi_n$ such that $L \sqcap (\bigsqcap P_k)^{ii} \to L \sqcap (\bigsqcap P_k)^{ii} \sqcap (\bigsqcap P_l)^{ii}$ follows from $\mathcal{B}_n$ and $L \sqcap (\bigsqcap P_k)^{ii} \sqcap (\bigsqcap P_l)^{ii}$ belongs to Λ_i and is strictly subsumed by $L \sqcap (\bigsqcap P_k)^{ii}$, etc. Since Λ_i is finite, this cannot go on forever, and thus we must reach an intent, which can actually be shown to be equal to L^{ii} (see [4] for

more details). The whole chain of implications thus implies the single implication $L \to L^{ii}$. This proves that all implications from $\mathcal{B}_i$ follow from $\mathcal{B}_n$. Because $\mathcal{B}_i$ is complete, $\mathcal{B}_n$ is also complete. □

The Exploration Algorithm

Now, we extend Algorithm 2 to a model exploration algorithm, which can deal with the case where the finite model i (called *background model* in the following) is not explicitly given, but rather "known" to an expert. We assume that, at the beginning of the exploration process, only some "parts" of the model i are given to the exploration algorithm as *working model* i_0. In the following, we assume that the model i_0 as well as its extensions i_j generated during the exploration process are *connected submodels* of i, i.e., we have $\Delta_{i_0} \subseteq \Delta_i$, $x \in A^{i_0}$ iff $x \in A^i$ for all $A \in \mathcal{N}_{\text{prim}}$ and all $x \in \Delta_{i_0}$, and Δ_{i_0} is closed under i-role successors: if $x \in \Delta_{i_0}$ and $(x, y) \in r^i$ for a role r, then $y \in \Delta_{i_0}$ and $(x, y) \in r^{i_0}$. It is easy to see that this implies $x \in C^{i_0}$ iff $x \in C^i$ for all $\mathcal{EL}_{\text{gfp}}$-concept descriptions C and all $x \in \Delta_{i_0}$.

Algorithm 3 describes our model exploration algorithm. The modification with respect to Algorithm 2 merely consists of adding a second while-loop to the algorithm. Intuitively, this loop is used to determine the proper conclusion $(\bigsqcap P_k)^{ii}$ for a given premise $\bigsqcap P_k$. Since i is not explicitly given, $(\bigsqcap P_k)^{ii}$ cannot be computed directly, but only by interacting with the expert. This is done in the following way. The implication $\bigsqcap P_k \to (\bigsqcap P_k)^{i_j i_j}$ is presented to the expert. If the expert refutes the implication (i.e., says that it does not hold) then she is required to provide a counter-example, i.e., a connected submodel i_{j+1} of i that extends i_j (i.e., satisfies $\Delta_{i_j} \subseteq \Delta_{i_{j+1}}$). This is repeated until the expert states that $\bigsqcap P_k \to (\bigsqcap P_k)^{i_j i_j}$ holds in i.

Since the set M_i is finite, only finitely many attributes can be added by Algorithm 3. Therefore, the outer while-loop can only be entered a finite number of times. With every pass of the inner while-loop, the working model is extended. Since the working models are submodels of the finite background model, this can only happen a finite number of times. This shows that Algorithm 3 terminates after a finite number of steps. Soundness and completeness of Algorithm 3 are easy consequences of soundness and completeness of Algorithm 2.

Theorem 5. *Assume that Algorithm 3 terminates after the n-th iteration of the outer while loop and that i_ℓ is the actual working model. Then $\{\bigsqcap P_k \to (\bigsqcap P_k)^{i_\ell i_\ell} \mid P_k \in \Pi_n\}$ is a finite basis for the $\mathcal{EL}_{\text{gfp}}$-GCIs holding in i.*

An Example

We illustrate Algorithm 2 using the example from the introduction. The domain of the background model thus consists of six persons: John, Michelle and their daughter Mackenzie, as well as Paul, Linda and their son James.[5] As primitive

[5] Since this is a very simple model, it satisfies GCIs not holding in the "real world."

Algorithm 3. The model exploration algorithm

Input: working model i_0 (connected submodel of the finite background model i)
$M_0 := \mathcal{N}_{\text{prim}}$, $\mathbb{K}_0 :=$ the context induced by M_0 and i_0, $\mathcal{S}_0 := \emptyset$
$\Pi_0 := \emptyset$, $P_0 := \emptyset$, $k := 0$, $j := 0$
while $P_k \neq \mathsf{null}$ **do**
  while expert refutes $\bigsqcap P_k \to (\bigsqcap P_k)^{i_j i_j}$ **do**
    $j := j + 1$
    Ask the expert for a new working model i_j that extends i_{j-1}, is a connected submodel of i, and contains a counterexample for $\bigsqcap P_k \to (\bigsqcap P_k)^{i_{j-1} i_{j-1}}$
  end while
  $\Pi_{k+1} := \Pi_k \cup \{P_k\}$
  $M_{k+1} := M_k \cup \{\exists r.(\bigsqcap P_k)^{i_j i_j} \mid r \in \mathcal{N}_r\}$
  $\mathcal{S}_{k+1} := \{\{C\} \to \{D\} \mid C, D \in M_k, C \sqsubseteq D\}$
  $k := k + 1$
  if $M_k = M_{k-1} = P_k$ **then**
    $P_k := \mathsf{null}$
  else
    $P_k :=$ lectically next set of attributes that respects all implications in $\{P_l \to P_l''^{k} \mid 1 \leq l < k\}$ and $\mathcal{S}_k$
  end if
end while

concepts we use *Male* (M), *Female* (F), *Father* (Ft) and *Mother* (Mt), and as role *child* (c). Let us assume that the initial working model i_0 contains only the first family, i.e., Δ_{i_0} consists of John, Michelle, and Mackenzie, and we have

$M^{i_0} = Ft^{i_0} = \{\text{John}\}$, $Mt^{i_0} = \{\text{Michelle}\}$,
$F^{i_0} = \{\text{Michelle}, \text{Mackenzie}\}$, $c^{i_0} = \{(\text{Michelle}, \text{Mackenzie}), (\text{John}, \text{Mackenzie})\}$.

1st Iteration: The algorithm starts with $P_0 = \emptyset$. We have $\bigsqcap P_0 = \top$ and $\top^{i_0 i_0} = \top$, and thus the expert is asked whether the GCI $\top \to \top$ holds in i. Obviously, the answer must be "yes," and we continue by computing the new set of attributes M_1 by adding $\exists c.\top$ to $M_0 = \mathcal{N}_{\text{prim}}$. The induced context $\mathbb{K}_1$ obtained this way is

	Ft	M	Mt	F	$\exists c.\top$
John	X	X			X
Michelle			X	X	X
Mackenzie			X		

where we assume that the elements of M_1 are ordered as listed in the table.

2nd Iteration: The lectically next set that is closed with respect to $\{\emptyset \to \emptyset''^{1}\} = \{\emptyset \to \emptyset\}$ is $\{Ft\}$. We have $Ft^{i_0 i_0} = \{\text{John}\}^{i_0} = Ft \sqcap M \sqcap \exists c.F$, which gives rise to the GCI $Ft \to Ft \sqcap M \sqcap \exists c.F$. Thus, the expert is presented with the question: "Is it true that every father is male and has a child that is female?". This is not true in the background model i since Paul is a father without daughter. The expert refutes the GCI by adding Paul as a counterexample. Note that she must also add James, because the new working model i_1 must be a connected submodel of i. Based on this model, the algorithm computes a new right-hand-side for the GCI: $Ft^{i_1 i_1} =$

$Ft \sqcap M \sqcap \exists c.\top$. The new GCI $Ft \to Ft \sqcap M \sqcap \exists c.\top$ is presented to the expert, who accepts it. Consequently, the new attribute $\exists c.(Ft \sqcap M \sqcap \exists c.\top)$ is added.

We do not look at the *next iterations* in as much detail as for the first two. The following GCIs are found:

1. $Mt \to Mt \sqcap F \sqcap \exists c.F$ (Refuted, Linda added as counterexample)
2. $Mt \to Mt \sqcap F \sqcap \exists c.\top$ (Accepted)
3. $F \sqcap M \to Aa$ (Accepted)
4. $\exists c.\top \sqcap M \to Ft \sqcap M \sqcap \exists c.\top$ (Accepted)
5. $\exists c.\top \sqcap F \to Mt \sqcap F \sqcap \exists c.\top$ (Accepted)
6. $\exists c.M \sqcap \exists c.F \to Aa$ (Accepted)
7. $\exists c.\exists c.\top \to Aa$ (Accepted)

Here Aa ("all attributes") stands for the cyclic $\mathcal{EL}_{\text{gfp}}$-concept description $(\mathcal{T}, A)$ where $\mathcal{T} = \{A \equiv M \sqcap F \sqcap Mt \sqcap Ft \sqcap \exists c.A\}$. Note that Aa is subsumed by any $\mathcal{EL}_{\text{gfp}}$-concept description that can be formulated using the primitive concepts M, F, Ft, Mt and the role c. As such, it is the best approximation of the bottom concept that $\mathcal{EL}_{\text{gfp}}$ can come up with.

Interestingly, all the GCIs accepted during the exploration process, except for the last two (6. and 7.), hold in the "real world." The GCIs 6. and 7. are artefacts of the simple model i used for the exploration. They are due to the fact that, in i, there are no grandparents, and no one has both a son and a daughter.

5 Related and Future Work

The context induced by a finite model and a finite set of concept descriptions as attributes has been considered before (e.g., in [12,13]). However, since this previous work did not make use of the most specific concept, the authors could not show and utilize the connections between the $\cdot^i$ operators in the model and the $\cdot'$ operators in the induced context. The work whose objectives is closest to ours is [13],[6] where Rudolph considers attributes defined in the DL $\mathcal{FLE}$, which is more expressive than $\mathcal{EL}$. Given a finite $\mathcal{FLE}$-model, he considers an infinite family of induced contexts $\mathbb{K}_n$, where the finite attribute sets are obtained by considering all $\mathcal{FLE}$-concept descriptions (modulo equivalence) up to role depth n. He then applies classical attribute exploration to these induced contexts, in each step increasing the role depths until a certain termination condition applies. Rudolph shows that the implication bases of the contexts considered up to the last step contain enough information to decide, for any GCI between $\mathcal{FLE}$-concept descriptions, whether this GCI holds in the given model or not. However, these implication bases do not appear to yield a basis for all the GCIs holding in the given finite model, though it might be possible to modify Rudolph's approach such that it produces a basis in our sense. The main problem with this approach is, however, that the number of attributes grows very fast when the role depth grows (this number increases at least by one exponential in each step). In contrast

[6] see `http://relexo.ontoware.org/` for a tool that realizes this approach.

to considering all concept descriptions up to a certain role depth, our approach only adds an attribute of the form $\exists r.(\bigsqcap P)^{ii}$ if P has been generated as the left-hand side of a GCI in our basis.

The main topic for future research is to show that the approach for using attribute exploration to complete DL knowledge bases introduced in [6] can be extended to the model exploration algorithm introduced in this paper.

References

1. Baader, F.: Terminological cycles in a description logic with existential restrictions. In: Proc. of IJCAI 2003, pp. 325–330. Morgan Kaufmann, San Francisco (2003)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proc. of ICJAI 2005, pp. 364–369. Morgan Kaufmann, San Francisco (2005)
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
4. Baader, F., Distel, F.: Exploring finite models in the description logic $\mathcal{EL}_{\text{gfp}}$. LTCS-Report 08-05, Chair for Automata Theory, TU Dresden (2008)
5. Baader, F., Distel, F.: A finite basis for the set of $\mathcal{EL}$-implications holding in a finite model. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 46–61. Springer, Heidelberg (2008)
6. Baader, F., Ganter, B., Sattler, U., Sertkaya, B.: Completing description logic knowledge bases using formal concept analysis. In: Proc. of IJCAI 2007. AAAI Press/The MIT Press (2007)
7. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL—a polynomial-time reasoner for life science ontologies. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 287–291. Springer, Heidelberg (2006)
8. Ganter, B.: Two basic algorithms in concept analysis. Preprint 831, Fachbereich Mathematik, TU Darmstadt, Darmstadt, Germany (1984)
9. Ganter, B.: Attribute exploration with background knowledge. Theoretical Computer Science 217(2), 215–233 (1999)
10. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, New York (1997)
11. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The making of a web ontology language. Journal of Web Semantics 1(1), 7–26 (2003)
12. Prediger, S.: Logical scaling in formal concept analysis. In: Delugach, H.S., Keeler, M.A., Searle, L., Lukose, D., Sowa, J.F. (eds.) ICCS 1997. LNCS, vol. 1257, pp. 332–341. Springer, Heidelberg (1997)
13. Rudolph, S.: Relational Exploration: Combining Description Logics and Formal Concept Analysis for Knowledge Specification. PhD thesis, Technische Universität Dresden (2006)
14. Spackman, K.A., Campbell, K.E., Cote, R.A.: SNOMED RT: A reference terminology for health care. J. of the American Medical Informatics Association, 640–644 (1997); Fall Symposium Supplement
15. Stumme, G.: Attribute exploration with background implications and exceptions. In: Bock, H.-H., Polasek, W. (eds.) Data Analysis and Information Systems, pp. 457–469. Springer, Berlin (1996)
16. The Gene Ontology Consortium. Gene Ontology: Tool for the unification of biology. Nature Genetics 25, 25–29 (2000)