An FCA Method for the Extensional Exploration of Relational Data

Sebastian Rudolph*
rudolph@math.tu-dresden.de

Institute of Algebra
Department of Mathematics and Natural Sciences
Dresden University of Technology
Germany

Abstract This paper introduces a technique for the exploration of relational data based on formal concept analysis (FCA). The formal description is followed by a simple example from basic mathematics.
Taking a concrete data set (as per example a relational database) on the one hand and a priori rules (as database constraints) on the other hand the presented algorithm determines all formulae of a certain shape (expressed as description logic statements) valid in this setting. This can even be done interactively: during the process, a domain expert may update the data set in order to witness the invalidity of a rule supposed by the algorithm.
This technique can be used in order to refine or construct ontologies as well as to find hidden regularities in relational data.

1 Introduction

In recent years, the term *semantic web* has grown significantly in interest. One of its central issues is the idea of exchanging "meaningful data" between heterogeneous systems.[1] Enabling such communication is one of the reasons to design *ontologies* for linking terminologies, combining distributed information, and drawing conclusions therefrom. This includes the task to provide background knowledge for the description of web pages contents.
One important formalism developed for describing ontologies is OIL, a language based on description logics (DL). Its major advantages are decidability and the existence of highly optimized reasoning algorithm implementations (e.g. FaCT by Ian Horrocks, see [10]).
Clearly, the design and refinement of ontologies is an expensive task which needs human control. However, there are several difficulties to be faced: often, large ontologies have to be designed and many pieces of information have to be entered into the system. How can one prevent that on the one hand redundant information is given (which means wasting human resources) and on the other

* Supported by DFG/Graduiertenkolleg.
[1] *Heterogeneous* means here endowed with different terminologies.

hand certain information is forgotten to enter? Furthermore, the creator of an ontology has to be an expert in the domain the ontology refers to as well as (more or less) in the underlying theory of ontologies and lots of technical issues.

To deal with these problems, techniques which automatize and structure the ontology creation and refinement process as far as possible and minimize and facilitate the necessary human-computer interaction are highly desirable.

This paper proposes an algorithm which (given atomic roles and concepts as well as possibly some amount of data and background information) determines further implicational dependencies by successively presenting questions in the form of DL entailment statements to a domain expert. Since this is done in an organized way no redundant questions will be asked. So in a certain sense, this approach combines knowledge discovery in databases with ontology creation.

In Section 2, based on DL a class of concept descriptions ($\mathcal{EL}$) is defined together with an extensional semantic using binary power context families. Definitions of entailment and equivalence of that formulae wrt. a fixed semantics are discussed.

In the sequel, Section 3 will shortly recall some notions of formal concept analysis (implication, stem base) as far as they are needed for an understanding of the attribute exploration algorithm developed by Ganter (cf. [4]).

In Section 4, we define a special kind of formal contexts that can be constructed on the basis of a binary power context family from a set of DL-formulae. We observe that implications within such a formal context correspond to valid DL entailment statements.

The algorithm, that consists of a sequence of exploration steps is described in Section 5: initialization, the actual exploration step yielding a stem base $\mathfrak{B}_i$, and how the stem base can be used to determine the attribute set and background knowledge for the next exploration step.

In Section 6, we apply the presented algorithm to an example from basic mathematics.

Section 7 discusses, how the validity of an arbitrary entailment statement between concept descriptions from $\mathcal{EL}_i$ can be decided using just the stem bases $\mathfrak{B}_0, \ldots, \mathfrak{B}_i$ obtained from the exploration process.

Concluding, in Section 8 we discuss how this algorithm can be applied for generating and refining ontologies.

2 The language $\mathcal{EL}$: syntax and semantics

In this section, we will present a way of constructing formulae from two sets of attribute names. These constructed formulae can be evaluated via extensional semantics. As the name $\mathcal{EL}$ already indicates, this is just a reformulation of the very basic ideas of description logic, where a binary power context family takes the role of the interpretation.

Definition 1. *Let $M_{\mathcal{C}}$, $M_{\mathcal{R}}$ be arbitrary finite sets, the elements of which we will call concept*[2] *names and role names resp. By $\mathcal{EL}(M_{\mathcal{C}}, M_{\mathcal{R}})$ (or shortly: $\mathcal{EL}$ if there is no danger of confusion) we denote the set of formulae (also called* CONCEPT DESCRIPTIONS*) inductively defined as follows:*

$$\begin{aligned} M_{\mathcal{C}} \cup \{\top, \bot\} &\subseteq \mathcal{EL} \\ \varphi, \psi \in \mathcal{EL} &\Rightarrow \varphi \sqcap \psi \in \mathcal{EL} \\ \varphi \in \mathcal{EL}, r \in M_{\mathcal{R}} &\Rightarrow \exists r.\varphi \in \mathcal{EL} \end{aligned}$$

Now we describe an extensional semantics for the above defined formulae. *Power context families* have been introduced as a contextual representation of *relational structures*. They have e.g. been used as semantics for conceptual graphs in [14]. Moreover, they suggest to extend the conceptual view[3] from unary predicates to predicates with arbitrary arities. In our case it suffices to restrict to the binary case, defined as follows.

Definition 2. *A* BINARY POWER CONTEXT FAMILY *on a set Δ, called the* UNIVERSE*, with $\Delta \neq \emptyset$ is a pair $(\mathbb{K}_{\mathcal{C}}, \mathbb{K}_{\mathcal{R}})$ consisting of the formal contexts $\mathbb{K}_{\mathcal{C}} := (G_{\mathcal{C}}, M_{\mathcal{C}}, I_{\mathcal{C}})$ and $\mathbb{K}_{\mathcal{R}} := (G_{\mathcal{R}}, M_{\mathcal{R}}, I_{\mathcal{R}})$ with $G_{\mathcal{C}} = \Delta$ and $G_{\mathcal{R}} = \Delta \times \Delta$.*

As we know from the definition of formal context, $M_{\mathcal{C}}$ and $M_{\mathcal{R}}$ are arbitrary sets and $I_{\mathcal{C}} \subseteq G_{\mathcal{C}} \times M_{\mathcal{C}}$ as well as $I_{\mathcal{R}} \subseteq G_{\mathcal{R}} \times M_{\mathcal{R}}$.

Definition 3. *The semantics mapping $[\![.]\!]_{\vec{\mathbb{K}}} : \mathcal{EL}(M_{\mathcal{C}}, M_{\mathcal{R}}) \to \mathcal{P}(\Delta)$ for a binary power context family $\vec{\mathbb{K}}$ on a universe Δ with attribute sets $M_{\mathcal{C}}, M_{\mathcal{R}}$ is defined recursively:*

$$\begin{aligned} [\![\top]\!]_{\vec{\mathbb{K}}} &:= \Delta \\ [\![\bot]\!]_{\vec{\mathbb{K}}} &:= \emptyset \\ [\![m]\!]_{\vec{\mathbb{K}}} &:= m^{I_{\mathcal{C}}} \text{ for all } m \in M_{\mathcal{C}} \\ [\![\varphi \sqcap \psi]\!]_{\vec{\mathbb{K}}} &:= [\![\varphi]\!]_{\vec{\mathbb{K}}} \cap [\![\psi]\!]_{\vec{\mathbb{K}}} \\ [\![\exists r.\phi]\!]_{\vec{\mathbb{K}}} &:= \{x \in \Delta \mid \exists y : (x,y) \in r^{I_{\mathcal{R}}} \wedge y \in [\![\varphi]\!]_{\vec{\mathbb{K}}}\} \text{ for all } r \in M_{\mathcal{R}} \end{aligned}$$

By $\mathcal{EL}_n$ we denote the set of all concept descriptions from $\mathcal{EL}$ with role depth of at most n.

Furthermore, we say a formula φ is VALID IN $\vec{\mathbb{K}}$ (which we denote by $\vec{\mathbb{K}} \models \varphi$), iff $[\![\varphi]\!]_{\vec{\mathbb{K}}} = \Delta$. A formula ψ ENTAILS a formula φ in $\vec{\mathbb{K}}$ (write: $\varphi \models_{\vec{\mathbb{K}}} \psi$), iff $[\![\varphi]\!]_{\vec{\mathbb{K}}} \subseteq [\![\psi]\!]_{\vec{\mathbb{K}}}$.
Two formulae φ and ψ are called $\vec{\mathbb{K}}-$EQUIVALENT, iff $\varphi \models_{\vec{\mathbb{K}}} \psi$ and $\psi \models_{\vec{\mathbb{K}}} \varphi$ (write: $\varphi \equiv_{\vec{\mathbb{K}}} \psi$).
Abbreviation: Let $C = \{c_1, \ldots, c_n\}$ be a finite set of $\mathcal{EL}$ concept descriptions. Then the new concept description $c_1 \sqcap \ldots \sqcap c_n$ will be abbreviated by $\bigsqcap C$. Furthermore, let $\bigsqcap\{c\} = c$ and $\bigsqcap \emptyset = \top$.

[2] Whenever in this article we use the term *concept* we refer to the notion used in Description Logic. If we want to refer to the meaning used in Formal Concept Analysis we use *formal concept*.

[3] More precisely: the way of thinking in conceptual hierarchies.

In our view, $\mathcal{EL}$ comprises the majority of concept descriptions used in human thinking. Most of the concepts employed intuitively have a positive (i.e. negation free) conjunctive structure, while disjunctions and negations are used quite rarely (cf. [13]).
Our algorithm provides for a given number n a canonical set of entailment statements by means of which the validity of every entailment query concerning concept descriptions from $\mathcal{EL}_n$ can be decided.

3 Attribute exploration

We assume the reader to be familiar with the basics of formal concept analysis (for a detailed introduction see [5]).
Let us here only shortly recall the attribute exploration algorithm. Developed by Ganter (see [4]), this algorithm allows for interactively determining the implicational knowledge of a given domain.

Definition 4. *Let M be an arbitrary set. If A and B are two sets with $A, B \subseteq M$ we will call the pair (A, B) an* IMPLICATION *on M. To support intuition we will write it as $A \rightarrow B$ in the sequel. We say an implication* HOLDS *for an attribute set C, iff from $A \subseteq C$ follows $B \subseteq C$. Moreover, an implication* HOLDS *in a formal context $\mathbb{K} = (G, M, I)$ iff it holds for all its object intents.*
Given a set $A \subseteq M$ and a set $\mathfrak{I}$ of implications on M, we write $A^{\mathfrak{I}}$ for the smallest subset of M which

- *contains A and*
- *fulfills all implications from $\mathfrak{I}$.*[4]

Let $Imp(\mathbb{K})$ denote the set of all implications holding in $\mathbb{K}$. A set of implications $\mathfrak{B}$ is called implicational base of $\mathbb{K}$ iff it is

- *complete, i.e., $A^{\mathfrak{B}} = A^{Imp(\mathbb{K})}$ for all $A \subseteq M$ and*
- *irredundant, i.e., for every implication $i \in \mathfrak{B}$ there is an $A \subseteq M$ with $A^{\mathfrak{B}\setminus\{i\}} \neq A^{Imp(\mathbb{K})}$.*

Roughly spoken, an implicational base of a formal context is a small representation of its complete implicational knowledge. Guigues and Duquenne [7] found a characterization of a canonical minimal implicational base - the so called *stem base*.
With attribute exploration we are provided with a tool to determine a stem base for a context which does not need to be known entirely in advance. For a detailed description of the algorithm see [5]. It starts with a fragmentary context, i.e., a context containing only some "example" objects, and systematically determines

[4] Since those two requirements are preserved under intersection, the existence of a smallest such set is assured. Moreover, note that the operation $(.)^{\mathfrak{I}}$ is a closure operator on M. Note also, that given A and $\mathfrak{I}$ the closure can be calculated in linear time (cf. [3]).

possible implications. These hypotheses are presented to an "expert", who either confirms them, or provides a counterexample. This dialogue continues until the complete information is determined.
Note, that the "expert" needs not to be a human being. It might as well be another algorithm (such as a reasoner, a constraint solver, or an automatic theorem prover), which is capable of answering the questions (cf. [1]).

4 $\mathcal{EL}$-Contexts

On the basis of a binary power context family we can define for an arbitrary set of $\mathcal{EL}$ concept descriptions a corresponding context, which states for every object from the underlying universe, whether it fulfills a certain concept description.

Definition 5. *Given a binary power context family* $\overrightarrow{\mathbb{K}} = (\mathbb{K}_{\mathcal{C}}, \mathbb{K}_{\mathcal{R}})$ *on a universe* Δ *and a set* $M \subseteq \mathcal{EL}(M_{\mathcal{C}}, M_{\mathcal{R}})$*, the corresponding* $\mathcal{EL}$-CONTEXT *is defined in the following way:*

$$\mathbb{K}_{\mathcal{EL}}(M) := (\Delta, M, I) \text{ with } \delta I m :\Leftrightarrow \delta \in [\![m]\!]_{\overrightarrow{\mathbb{K}}}$$

Now, assume

$$\{m_1, \ldots, m_k\} \rightarrowtail \{m_{k+1}, \ldots, m_l\}$$

is an implication valid in $\mathbb{K}_{\mathcal{EL}}$. It can be shown easily that this is equivalent to the validity of the entailment statement

$$\bigsqcap\{m_1, \ldots, m_k\} \models_{\overrightarrow{\mathbb{K}}} \bigsqcap\{m_{k+1}, \ldots, m_l\}.$$

Since the wanted result of the exploration process is a means for deciding the validity of any entailment statement between concept descriptions from $\mathcal{EL}_i$ for a given $i \in \mathbb{N}$, we look for a small set $M \subseteq \mathcal{EL}(M_{\mathcal{C}}, M_{\mathcal{R}})$ such that every entailment corresponds to an implication in M.

5 Successive exploration

The exploration technique described here consists of a sequence of single attribute explorations, each step providing necessary information for the next one. The formal context explored in step $i \in \mathbb{N}$ will be denoted with $\mathbb{K}_i = (\Delta, M_i, I_i)$. The set of example objects we start with will be named G_i. The process yields the corresponding stem base $\mathfrak{B}_i$, which is used as background information for the next step as well as to generate the attribute set M_{i+1}. Furthermore, using the stem bases $\mathfrak{B}_0, \ldots, \mathfrak{B}_i$, any entailment statement on $\mathcal{EL}_i$ can be decided (this will be dealt with in Section 7).

We start the exploration sequence with the context $\mathbb{K}_0 = (G_0.M_0, I_0)$ where
$G_0 := G$,
$M_0 := M_\mathcal{C} \cup \{\top, \bot\}$, and
$I_0 := (I_\mathcal{C} \cap G \times M_\mathcal{C}) \cup (G \times \{\top\})$,
with G being a set of objects initially "known" to the algorithm.
After having carried out the preparations, the actual exploration (the dialogue with the expert) takes place. Every implication $\{m_1, \ldots, m_k\} \rightarrowtriangle \{m_{k+1}, \ldots, m_l\}$ being presented to the expert has to be interpreted in the following way: "Do all entities from the universe that fulfill the concept description $m_1 \sqcap \ldots \sqcap m_k$ also fulfill the concept description $m_{k+1} \sqcap \ldots \sqcap m_l$?" The expert either confirms this, or provides an entity that violates this condition. The result of this process is the stem base $\mathfrak{B}_i$.
The attribute set M_{i+1} for the $(i+1)$th exploration step is generated as follows:

$$M_{i+1} := M_0 \cup \{\exists r.m_1 \sqcap \ldots \sqcap m_n \mid r \in M_\mathcal{R}, \{m_1, \ldots, m_n\} = \{m_1, \ldots, m_n\}^{\mathcal{B}_i} \subseteq M_i\}$$

The choice of this attribute set is motivated by the following fact: for any set A of attributes from M_i we have $\bigsqcap A \equiv_{\mathbb{K}} \bigsqcap A^{\mathfrak{B}_i}$ (see the appendix for the proof). Thus (in comparison to creating new attributes for any conjunction) we can reduce the necessary amount of attributes considerably without loosing expressivity. Conjunctions outside any quantifier range do not need to be internalized into new attributes, since the structure of the implications allows to express them anyway (cf. Theorem 1 in the appendix).
We now investigate, which implicational information is automaticly valid in $\mathbb{K}_{i+1}$ because it is valid in any binary power context family, that is compatible with $\mathfrak{B}_0, \ldots, \mathfrak{B}_i$. This can be used as a priori knowledge for the exploration of $\mathbb{K}_{i+1}$ in order to minimize the amount of questions asked to the expert.
First of all, due to the internal structure of the composed attributes there are some trivially valid implications, they are of the form

$$\{\exists r. \bigsqcap A\} \rightarrowtriangle \{\exists r. \bigsqcap B\} \text{ for all } B \subseteq A \text{ and } A, B \in M_i.$$

Next, in every exploration step we want to reuse the knowledge collected in the former steps. Therefore we define a sequence of maps (f_i)

$$f_i : M_i \to M_{i+1}, \begin{cases} m \mapsto m, \text{ if } m \in M_\mathcal{C} \cup \{\top, \bot\} \\ \exists r. \bigsqcap \{m_1, \ldots, m_n\} \mapsto \exists r. \bigsqcap \{f_{i-1}(m_1), \ldots, f_{i-1}(m_n)\}^{\mathfrak{B}_i} \end{cases}$$

where f_i assigns to each attribute of the context $\mathbb{K}_i$ its "updated version" in $\mathbb{K}_{i+1}$. This mapping naturally extends to implications by applying it to the literals. Obviously an implication ι is valid in $\mathbb{K}_i$ iff $f_i(\iota)$ is valid in $\mathbb{K}_{i+1}$. Thus we can use the implicational knowledge collected in exploration step i as background information for the step $i+1$, by adding $\{f(\iota) \mid \iota \in \mathfrak{B}_i\}$ to the a priori information for the next exploration step. When proceeding from $i = 0$ to $i = 1$ this step is trivial since all attributes from M_0 are primitive concepts.
After these preparations, we can carry out the next exploration step. This process can be repeated ($i = 2, 3, \ldots$), yet the number of attributes can increase drastically from step to step; in the worst case we would get $|M_{i+1}| =$

$|M_{\mathcal{C}}| + |M_{\mathcal{R}}| \cdot 2^{|M_i|}$. So practically the algorithm would have to be ended after a few steps. However, the more dependencies there are in the data the less rapid would be that growth.[5]

6 An example from basic mathematics

Let us consider an easy example for this exploration method. As our universe Δ we take the natural numbers $\mathbb{N} := \{0, 1, 2, \ldots\}$. Let $M_{\mathcal{C}}$ and $M_{\mathcal{R}}$ and their corresponding incidence relations be defined as shown in Figure 1.

$c \in M_{\mathcal{C}}$	name	$c^{I_{\mathcal{C}}} := \{n \in \mathbb{N} \mid nI_{\mathcal{C}}c\}$
ev	even	$\{2n \mid n \in \mathbb{N}\}$
od	odd	$\{2n+1 \mid n \in \mathbb{N}\}$
pr	prime	$\{n \geq 2 \mid kl = n \Rightarrow k \in \{1, n\}\}$
e0	equals zero	$\{0\}$
e1	equals one	$\{1\}$
e2	equals two	$\{2\}$
g2	greater than two	$\{n \in \mathbb{N} \mid n \geq 3\}$

$r \in M_{\mathcal{R}}$	name	$r^{I_{\mathcal{R}}} := \{(n, m) \in \mathbb{N}^2 \mid (n, m) I_{\mathcal{R}} r\}$
s	successor	$\{(n, n+1) \mid n \in \mathbb{N}\}$
p	predecessor	$\{(n+1, n) \mid n \in \mathbb{N}\}$
d	divisor	$\{(m, n) \mid \exists k \in \mathbb{N} : m = kn\}$
m	multiple	$\{(n, m) \mid \exists k \in \mathbb{N} : m = kn\}$

Figure 1. Attribute sets $M_{\mathcal{C}}$, $M_{\mathcal{R}}$ and definition of the incidence relations $I_{\mathcal{C}}$, $I_{\mathcal{R}}$ for the example.

So, with the above defined interpretations of the attributes, the crosstables of the corresponding power context family would look like in Figure 2.
Now let us carry out the exploration of this example. The first step consists of an exploration just on the attributes from $M_{\mathcal{C}} \cup \{\top, \bot\} =: M_0$ and yields the stem base and corresponding concept lattice in Figure 3.
Proceeding with our example, we have to generate the attribute set M_1 for the next exploration step. It is shown in Figure 4.
Then we have to generate the a priori knowledge for the second exploration step. Firstly, this would contain all trivially valid implications such as $\exists d.(ev \sqcap pr \sqcap e2) \rightarrowtail \exists d.ev$ or $\exists p.(od \sqcap e1) \rightarrowtail \exists p.\top$.
Moreover, we use the information collected so far. Since we proceed from the first ($i = 0$) to the second ($i = 1$) step and f_0 is the identity function, we can just use $\mathfrak{B}_0$ as additional a priori information without further adaption.

[5] Theoretically, a fixed point - where no more information will be acquired by further exploration steps - will be achieved, if Δ is finite.

$\mathbb{K}_C$	*e0*	*e1*	*e2*	*g2*	*pr*	*ev*	*od*
0	×					×	
1		×					×
2			×		×	×	
3				×	×		×
4				×		×	
5				×	×		×
6				×		×	
7				×	×		×
8				×		×	
9				×			×
...							

$\mathbb{K}_{\mathcal{R}}$	*s*	*p*	*d*	*m*
(0,0)				×
(0,1)	×		×	
(1,0)		×		×
(0,2)			×	
(1,1)			×	×
(2,0)				×
(0,3)			×	
(1,2)	×			×
(2,1)		×	×	
(3,0)				×
...				

Figure 2. Crosstables representing the power context family in the example.

$\{e0\} \to \{ev\}$	$\{ev, od\} \to \{\bot\}$
$\{e1\} \to \{od\}$	$\{g2, e0\} \to \{\bot\}$
$\{e2\} \to \{ev,pr\}$	$\{g2, e1\} \to \{\bot\}$
$\{ev, pr\} \to \{e2\}$	$\{e0, e2\} \to \{\bot\}$
$\{od, pr\} \to \{g2\}$	$\emptyset \to \{\top\}$
$\{pr, g2\} \to \{od\}$	

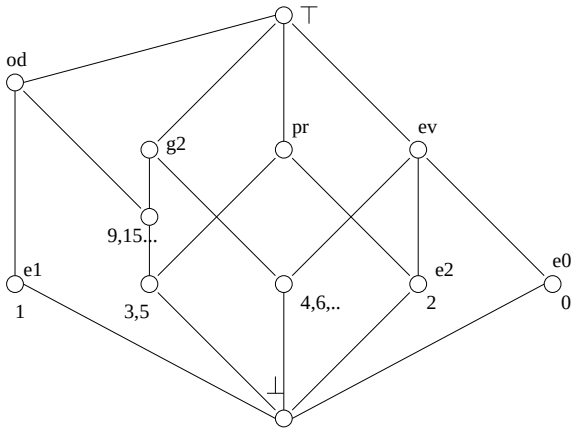

Figure 3. Stem base and concept lattice provided by exploration of $\mathbb{K}_0$.

ev	$\exists s.\top$	$\exists p.\top$	$\exists d.\top$	$\exists m.\top$
od	$\exists s.g2$	$\exists p.g2$	$\exists d.g2$	$\exists m.g2$
pr	$\exists s.pr$	$\exists p.pr$	$\exists d.pr$	$\exists m.pr$
$e0$	$\exists s.od$	$\exists p.od$	$\exists d.od$	$\exists m.od$
$e1$	$\exists s.ev$	$\exists p.ev$	$\exists d.ev$	$\exists m.ev$
$e2$	$\exists s.(od \sqcap e1)$	$\exists p.(od \sqcap e1)$	$\exists d.(od \sqcap e1)$	$\exists m.(od \sqcap e1)$
$g2$	$\exists s.(ev \sqcap g2)$	$\exists p.(ev \sqcap g2)$	$\exists d.(ev \sqcap g2)$	$\exists m.(ev \sqcap g2)$
$\top$	$\exists s.(ev \sqcap e0)$	$\exists p.(ev \sqcap e0)$	$\exists d.(ev \sqcap e0)$	$\exists m.(ev \sqcap e0)$
$\bot$	$\exists s.(od \sqcap g2 \sqcap pr)$	$\exists p.(od \sqcap g2 \sqcap pr)$	$\exists d.(od \sqcap g2 \sqcap pr)$	$\exists m.(od \sqcap g2 \sqcap pr)$
	$\exists s.(ev \sqcap pr \sqcap e2)$	$\exists p.(ev \sqcap pr \sqcap e2)$	$\exists d.(ev \sqcap pr \sqcap e2)$	$\exists m.(ev \sqcap pr \sqcap e2)$

Figure 4. Attributes M_1 for the second exploration step.

After these preparations, we can carry out the next exploration step. The concept lattice of the resulting implicational base is shown in Figure 5.
Much information can be read directly from the lattice: formulae attached to the same diagram node are semantically equivalent wrt. the fixed semantics. So from the bottommost node we can conclude $\exists d.(ev \sqcap e0) \equiv_{\mathbb{R}^{\rightarrow}} \bot$ meaning "no natural number can be divided by zero". Looking at the topmost node, we find e.g. $\exists s.\top \equiv_{\mathbb{R}^{\rightarrow}} \top$: "every natural number has a successor".
The exploration process could be continued for $\mathbb{K}_2, \mathbb{K}_3, \ldots$. In our case, the attribute set M_2 would already have 163 elements.

7 Checking the validity of an entailment statement

Suppose the exploration procedure has been carried out until step i and let $c_1 \models_{\mathbb{R}^{\rightarrow}} c_2$ (with $c_1, c_2 \in \mathcal{EL}_i$) be an entailment statement, the validity of which has to be checked.
In order to do this, we define a recursive function: $\varphi : \mathcal{EL} \times \mathbb{N} \to \mathcal{P}(\mathcal{EL})$ with

$$\begin{aligned} \varphi(m,i) &:= \{m\}^{\mathfrak{B}_i} \text{ for } m \in M_{\mathcal{C}} \cup \{\top, \bot\} \\ \varphi(\exists r.c, i) &:= \{\exists r. \textstyle\prod \varphi(c, i-1)\}^{\mathfrak{B}_i} \\ \varphi(\textstyle\prod C, i) &:= \left(\bigcup \{\varphi(c,i) \mid c \in C\}\right)^{\mathfrak{B}_i} \end{aligned}$$

Note that for all $c \in \mathcal{EL}_i$ we have $\varphi(c,i) \subseteq M_i$ (see the appendix for the proof). Furthermore, φ carries out an equivalency preserving transformation, which means $[\![c]\!]_{\mathbb{R}^{\rightarrow}} = [\![\prod \varphi(c,i)]\!]_{\mathbb{R}^{\rightarrow}}$ for all $c \in \mathcal{EL}_i$.
Due to these facts we can check the validity of any entailment statement between concept descriptions from $\mathcal{EL}_i$ in the following way (see the appendix for the proof):

$$c_1 \models_{\mathbb{R}^{\rightarrow}} c_2 \iff \varphi(c_2, i) \subseteq \varphi(c_1, i)$$

By being able to check the validity of entailment statements we can also check the equivalence of concept descriptions. So, from our implicational base we can

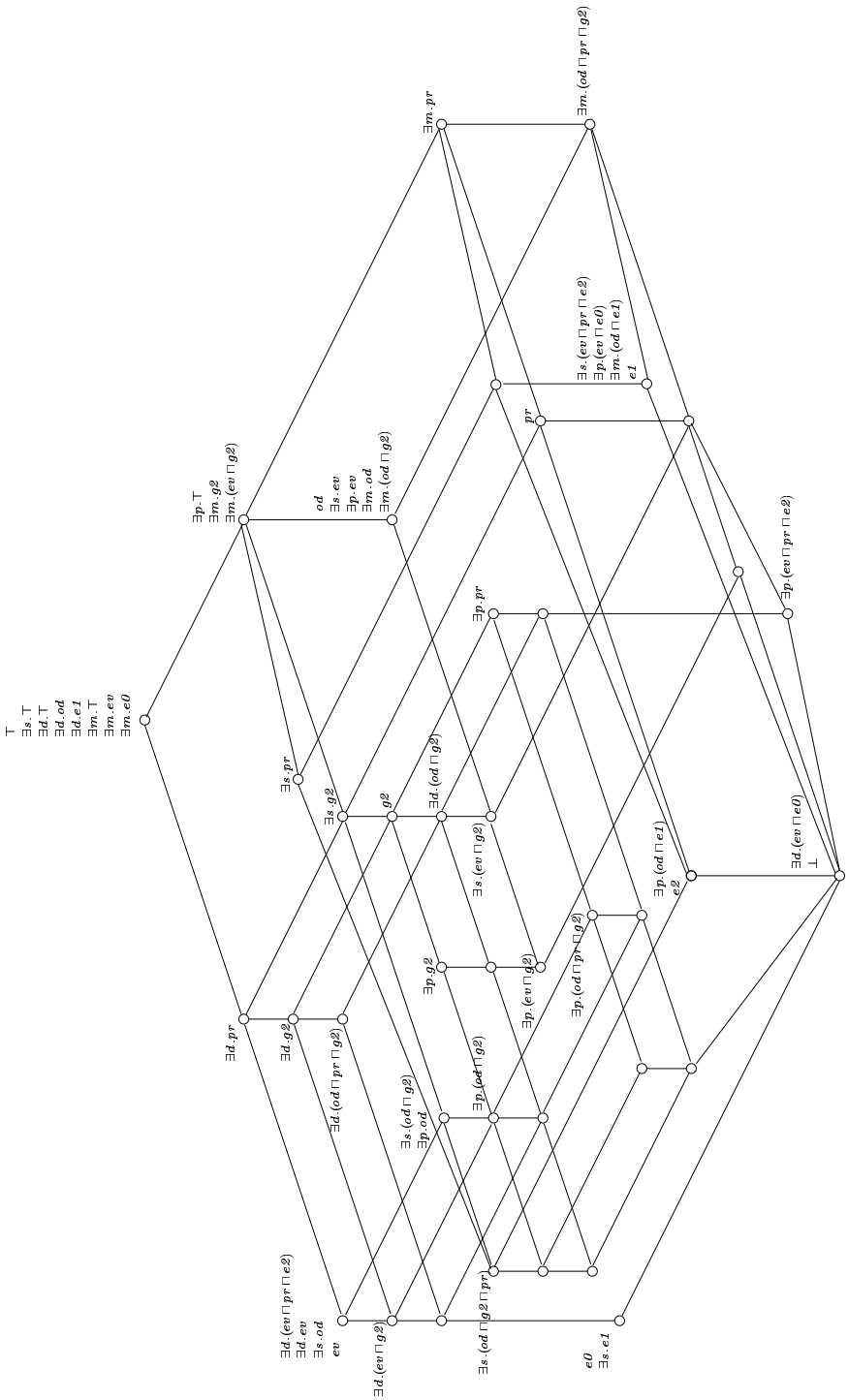

Figure 5. Concept lattice from the second exploration step representing the implicational knowledge in $\mathbb{K}_1$.

derive the fact $ev \models_{\mathcal{R}} \exists d.e2$ as well as $\exists d.e2 \models_{\mathcal{R}} ev$, therefore $ev \equiv_{\mathcal{R}} \exists d.e2$, which is obvious in our case, however, in general this could be a way to minimize the set of primitive concepts by finding definitions (composed formulae that are equivalent) for some of them.

8 A possible application: ontology exploration

After having described the algorithm, we want to discuss how it can be used in order to aid generating or refining ontologies by finding additional domain axioms. We hereby refer to ontology types based on description logics, wherein $\mathcal{EL}$ can be embedded, and for which exist efficient reasoning algorithms. The probably most prominent example is OIL being equivalent to the very expressive description logic $\mathcal{SHIQ}(d)$ (see [9] and [8]).
Deciding which features of the domain entities are of interest to the ontology users (and thus should be incorporated in the ontology) is a task being left to the human domain experts. So suppose, the stipulation of classes ($M_{\mathcal{C}}$) and roles ($M_{\mathcal{R}}$) has been carried out. Possibly also some domain axioms have already been stated.
Additionally there could be some sample data set (e.g. a relational database) containing entities, the classes they belong to, and information about slots.
Now we can apply our algorithm. Thereby the sample data set mentioned above can be incorporated completely into the object set G. If this is impossible (because the data set might not be available as a whole and can only be queried) or not indicated (because of efficiency reasons), we can nevertheless use it by querying it "on the fly" as we shall see in the sequel.
When being started, the algorithm comes up with implications asking for their validity. These are interpreted as hypothetical domain axioms. Such an axiom will be passed to a DL-reasoner, which tries to prove it from the axioms already present in the ontology description. If it succeeds, the validity of the new axiom is confirmed to the exploration algorithm. If it does not, the database will be queried for a counterexample (in case the entire data set has not completely been added to G in advance). If such a counterexample has been found, the validity of the presented hypothetical axiom will be denied and the counterexample be entered. The remaining case is the interesting one: it has been found evidence neither for validity (by a proof) nor for non-validity (by a counterexample) of the hypothetical axiom. In this case, the human expert will be asked for the ultimate decision. If the expert confirms the validity, genuinely new information about the domain has been made explicit and will be added to the ontology description, thereby refining it. In case of denial, the counterexample provided by the expert may not only be added to G but also to our sample data set, which means extending it by an "interesting" entity.
This algorithm assures that all valid domain axioms having the shape of $\mathcal{EL}_n$ entailment statements are found in the exploration step n. Therefore we believe it to be a useful tool for ontology engineers, helping them to structure the process of specifying domain knowledge.

9 Related work

The PhD thesis of Zickwolff [15] can be seen as a first attempt to apply the FCA exploration technique to a logic more expressive than propositional logic. Implicational bases in her sense represent the first order Horn theory of a certain domain.
Formal contexts, where attributes are DL-formulae and the incidence relation is defined by validity, have been described by Prediger in [11]. However, this work presented a way of enlarging a context's attribute set by "interesting" descriptions without dealing with exploration issues.
Baader presented a method for computing the subsumption hierarchy of all concept descriptions, that can be obtained by applying conjunction to concept names in [1]. There he also proposed a "dialogue" between an attribute exploration algorithm and a DL reasoner. Yet there are two main differences to the approach presented here: While Baader deals with subsumptions (i.e., entailments valid wrt. all interpretations), we consider a fixed semantic (the domain, our ontology refers to). Furthermore, the implicational bases from our exploration allow the decision of the validity of any entailment statement between formulae, that is built up from the concept and role names by conjunction *and* existencial restriction while Baader considers just conjunction. This is also the reason why we need an algorithm with several steps: every step allows to explore deeper nestings of existencial restriction.

10 Future work

We are confident to extend the presented approach into several directions in the future:
First, we will enlarge the set of considered formulae, such that it comprises formulae containing universal quantifiers.
Next, we will integrate the techniques published in our paper [6] in order to explore every role separately. Although the algorithms presented there were originally aimed at transition systems, the ideas seem to fit very well into the more general framework of modal and description logics.
Also an exploration of the context $\mathbb{K}_{\mathcal{R}}$ would provide a priori information accelerating the DL exploration process.
All these issues (as well as possibly an implementation) will be part of my PhD thesis, which is going to appear soon.

Appendix: proofs

Theorem 1. *Let $\overrightarrow{\mathbb{K}}$ a binary power context family, $M \subseteq \mathcal{EL}$ and $A, B \subseteq M$. Then the implication $A \rightarrowtriangle B$ is valid in $\mathbb{K}_{\mathcal{EL}}(M)$ if and only if $\bigsqcap A \models_{\overrightarrow{\mathbb{K}}} \bigsqcap B$.*

Proof. $\mathbb{K}_{\mathcal{EL}}(M) \models A \twoheadrightarrow B$ iff for all $\delta \in \Delta$ from $A \subseteq \delta^I$ follows $B \subseteq \delta^I$. This is the case iff $\bigcap\{b^I \mid b \in B\} \subseteq \bigcap\{a^I \mid a \in A\}$ which due to the definition of I is equivalent to $\bigcap\{[\![b]\!]_{\overrightarrow{\mathbb{R}}} \mid b \in B\} \subseteq \bigcap\{[\![a]\!]_{\overrightarrow{\mathbb{R}}} \mid a \in A\}$ and thus also to $[\![\bigsqcap B]\!]_{\overrightarrow{\mathbb{R}}} \subseteq [\![\bigsqcap A]\!]_{\overrightarrow{\mathbb{R}}}$. □

Theorem 2. *Let $c \in \mathcal{EL}_i$. Then $\varphi(c,i) \subseteq M_i$.*

Proof. We first show $\varphi(c,i) \subseteq M_i$, using induction on the role depth considering three cases:

- $c \in M_{\mathcal{C}} \cup \{\top, \bot\}$. Then by definition $c \in M_i$ and thus $\{c\}^{\mathfrak{B}_i} \in M_i$.
- $c = \exists r.\tilde{c}$. As induction hypothesis we have $\varphi(\tilde{c}, i-1) \subseteq M_{i-1}$. Yet, since φ always gives a closed set wrt. $\mathfrak{B}_i$, we have also $\exists r.\bigsqcap \varphi(\tilde{c}, i-1) \in M_i$, as a look to the constructive definition of M_i immediately shows.
- $c = \bigsqcap \tilde{C}$. W.l.o.g. we presuppose there is no conjunction outside the quantifier range in any $\tilde{c} \in \tilde{C}$. So we have $\varphi(\tilde{c}, i) \subseteq M_i$ due to the two cases above, and subsequently also $\left(\bigcup\{\varphi(\tilde{c}, i) \mid c \in C\}\right)^{\mathfrak{B}_i} \subseteq M_i$. □

Lemma 1. *For any $A \subseteq M_i$ we have $\bigsqcap A \equiv_{\overrightarrow{\mathbb{R}}} \bigsqcap A^{\mathfrak{B}_i}$.*

Proof. $[\![\bigsqcap A]\!]_{\overrightarrow{\mathbb{R}}} = \bigcap\{[\![m]\!]_{\overrightarrow{\mathbb{R}}} \mid m \in A\} = \bigcap\{m^{I_i} \mid m \in A\} = A^{I_i} = A^{I_i I_i I_i} = A^{\mathfrak{B}_i I_i} = \bigcap\{m^{I_i} \mid m \in A^{\mathfrak{B}_i}\} = \bigcap\{[\![m]\!]_{\overrightarrow{\mathbb{R}}} \mid m \in A^{\mathfrak{B}_i}\} = [\![\bigsqcap A^{\mathfrak{B}_i}]\!]_{\overrightarrow{\mathbb{R}}}$. □

Theorem 3. *Let $c \in \mathcal{EL}_i$. Then $c \equiv_{\overrightarrow{\mathbb{R}}} \bigsqcap \varphi(c,i)$.*

Proof. We show this again via induction on the role depth:

- $c \in M_{\mathcal{C}} \cup \{\top, \bot\}$. Then we have $[\![c]\!]_{\overrightarrow{\mathbb{R}}} = [\![\bigsqcap\{c\}]\!]_{\overrightarrow{\mathbb{R}}} = [\![\bigsqcap\{c\}^{\mathfrak{B}_i}]\!]_{\overrightarrow{\mathbb{R}}}$ due to Lemma 1.
- $c = \exists r.\tilde{c}$. By induction hypothesis we have $[\![\tilde{c}]\!]_{\overrightarrow{\mathbb{R}}} = [\![\bigsqcap \varphi(\tilde{c}, i-1)]\!]_{\overrightarrow{\mathbb{R}}}$, therefore $[\![\exists r.\tilde{c}]\!]_{\overrightarrow{\mathbb{R}}} = [\![\exists r.\bigsqcap \varphi(\tilde{c}, i-1)]\!]_{\overrightarrow{\mathbb{R}}}$. Moreover, from Lemma 1 follows that $[\![\exists r.\bigsqcap \varphi(\tilde{c}, i-1)]\!]_{\overrightarrow{\mathbb{R}}} = [\![\bigsqcap\{\exists r.\bigsqcap \varphi(\tilde{c}, i-1)\}^{\mathfrak{B}_i}]\!]_{\overrightarrow{\mathbb{R}}}$, which by definition equals $[\![\bigsqcap \varphi(\exists r.\tilde{c}, i)]\!]_{\overrightarrow{\mathbb{R}}}$.
- $c = \bigsqcap \tilde{C}$. Again we can preassume no conjunction outside the quantifier range in any $\tilde{c} \in \tilde{C}$. Then $[\![\bigsqcap \tilde{C}]\!]_{\overrightarrow{\mathbb{R}}} = \bigcap\{[\![\tilde{c}]\!]_{\overrightarrow{\mathbb{R}}} \mid \tilde{c} \in \tilde{C}\} = \bigcap\{[\![\bigsqcap \varphi(\tilde{c}, i)]\!]_{\overrightarrow{\mathbb{R}}} \mid \tilde{c} \in \tilde{C}\}$ because of the cases shown before. Now, this is obviously the same as $\bigcap\{[\![m]\!]_{\overrightarrow{\mathbb{R}}} \mid m \in \varphi(\tilde{c}, i), \tilde{c} \in \tilde{C}\} = [\![\bigsqcap(\bigcup\{\varphi(\tilde{c}, i) \mid \tilde{c} \in \tilde{C}\})]\!]_{\overrightarrow{\mathbb{R}}}$, which in turn is equivalent to $[\![\bigsqcap(\bigcup\{\varphi(\tilde{c}, i) \mid \tilde{c} \in \tilde{C}\})^{\mathfrak{B}_i}]\!]_{\overrightarrow{\mathbb{R}}}$. □

Corollary 1. *Let $c_1, c_2 \in \mathcal{EL}_i$. Then $c_1 \models_{\overrightarrow{\mathbb{R}}} c_2$ if and only if $\varphi(c_2, i) \subseteq \varphi(c_1, i)$.*

Proof. Due to Theorem 3, $c_1 \models_{\overrightarrow{\mathbb{R}}} c_2$ is equivalent to $\bigsqcap \varphi(c_1, i) \models_{\overrightarrow{\mathbb{R}}} \bigsqcap \varphi(c_2, i)$. According to Theorem 2, we have $\varphi(c_1, i) \subseteq M_i$ and $\varphi(c_2, i) \subseteq M_i$. So via Theorem 1, this means the same as the validity of the implication $\varphi(c_1, i) \twoheadrightarrow \varphi(c_2, i)$ in $\mathbb{K}_i$. Now, since the application of φ always gives a closed set wrt. $\mathbb{K}_i$, this is equivalent to $\varphi(c_2, i) \subseteq \varphi(c_1, i)$. □

References

1. F. Baader: 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.
2. Baader, F.: The Description Logic Handbook: Theory, Practice, and Applications. Cambridge University Press, 2003.
3. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. J. Logic Programming 3:267-284, 1984.
4. Ganter, B., Two basic algorithms in concept analysis. FB4-Preprint No 831, TH Darmstadt, 1984.
5. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin-Heidelberg, 1999.
6. Ganter, B., Rudolph, S., Formal Concept Analysis Methods for Dynamic Conceptual Graphs. In: H. S. Delugach, G. Stumme (Eds.): Conceptual Structures: Broadening the Base, Springer-Verlag, 2001.
7. Guigues, J.-L., Duquenne, V.: Familles minimales d'implications informatives resultant d'un tableaux de donnés binaires. Math. Sci. Humaines 95, 1986.
8. Horrocks, I. et al.: The Ontology Inference Layer OIL, URL: http://www.ontoknowledge.org/oil/papers.shtml.
9. Horrocks, I., Sattler, U., Tobies, S.: Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In: D. MacAllester (Ed.), Proceedings of CADE-2000, LNAI 1831, Springer, 2000.
10. Horrocks, I.: Benchmark analysis with fact. In: Proceedings of TABLEAUX 2000, LNAI 1847, Springer, 2000.
11. Prediger, S.: Terminologische Merkmalslogik in der Formalen Begriffsanalyse. In: G. Stumme, R. Wille (Eds.): Begriffliche Wissensverarbeitung: Methoden und Anwendungen. Springer-Verlag, Heidelberg, 2000.
12. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements, In: Artificial Intelligence, 48:1-26, 1991.
13. Sowa, J.: Ontology, Metadata, and Semiotics. In: B. Ganter / G. M. Mineau (Eds.): Conceptual Structures: Logical, Linguistic, and Computational Issues, LNAI 1867, Springer Verlag, 2000.
14. Wille, R.: Conceptual Graphs and Formal Concept Analysis. In: D. Lukose, H. Delugach, M. Keeler, L. Searle, J. Sowa (Eds.): Conceptual Structures: Fulfilling Peirce's Dream, Springer-Verlag, 1997.
15. Zickwolff, M.: Rule Exploration: First Order Logic in Formal Concept Analysis, PhD thesis, TH Darmstadt, 1991.