A Proposal for Combining Formal Concept Analysis and Description Logics for Mining Relational Data

Amine Mohamed Rouane Hacene, Marianne Huchard, Amedeo Napoli, Petko Valtchev

A proposal for combining formal concept analysis and description logics for mining relational data

Marianne Huchard[1], Amedeo Napoli[2], Mohamed Hacene Rouane[3], and Petko Valtchev[3]

[1] LIRMM, 161, rue Ada, F-34392 Montpellier Cedex 5
[2] LORIA, B.P. 239, F-54506 Vandœuvre-lès-Nancy
[3] DIRO, Université de Montréal, C.P. 6128, Montréal, Canada

Abstract. Recent advances in data and knowledge engineering have emphasized the need for formal concept analysis (FCA) tools taking into account structured data. There are a few adaptations of the classical FCA methodology for handling contexts holding on complex data formats, e.g. graph-based or relational data. In this paper, relational concept analysis (RCA) is proposed, as an adaptation of FCA for analyzing objects described both by binary and relational attributes. The RCA process takes as input a collection of contexts and of inter-context relations, and yields a set of lattices, one per context, whose concepts are linked by relations. Moreover, a way of representing the concepts and relations extracted with RCA is proposed in the framework of a description logic. The RCA process has been implemented within the Galicia platform, offering new and efficient tools for knowledge and software engineering.

1 Introduction

Formal concept analysis (FCA) has been successfully applied to a range of knowledge engineering problems [22, 26, 28]. Nevertheless, FCA methods and tools aimed at directly processing data –for producing knowledge units represented within a knowledge representation language based on description logic (DL) [1] such as OWL– are still under study [26]. One key difficulty lies in the presence and management of relational attributes or links in the data, such as spouse, reference, and part-of. For example, a target group for a marketing campaign may be to analyze the class of "spouses of Master Gold credit card holders", that involves both binary and relational attributes.

Current FCA methods and tools have no capabilities for taking into account relational attributes. This is a rather hard problem to solve, since relational attributes introduce dependencies and even cycles between the data items. A standard way for producing DL-like concept descriptions from a formal context including binary and relational attributes remains to be designed. Accordingly, one of the objectives of this paper is to present a methodology for taking into account relational attributes within FCA, leading to what could be called "relational concept analysis" or RCA.

The introduction of relational information, e.g. relational attributes, in the data formats for FCA has been studied for almost a decade now, leading to three main categories of research lines: (i) the relational attributes remain within the formal objects [12–14], (ii) relational attributes are considered as first-class citizens and organized into an independent lattice, separated from the standard concept lattice [17] (just like relation types are represented within the conceptual graph formalism [21]), (iii) relations between concepts are established independently from concept construction, on a manual or semi-automated basis [18]. Although these three approaches successfully deal with relational attributes for solving a specific task, they are still not general enough and do not allow to combine and process binary and relational attributes as object descriptors at the concept formation step. Such a need arises in various practical situations, for example in model engineering for software development or in ontology learning from data.

A first introduction of relational concept analysis (RCA) has been proposed in [11]. The data structure on which is based the relational concept analysis process is called a "relational context family" (RCF): it is composed of a collection of contexts and inter-context relations, the latter being binary relations between pairs of object sets lying in two different contexts. The objective is to build a set of lattices whose concepts are related by relational attributes, similar to DL roles or to UML associations. In addition, there are needs for associating restrictions with relational attributes for describing specific characteristics. RCA has been initially motivated by an application on the engineering of UML static models (see [7]) with an emphasis on expressiveness and algorithmic aspects. Meanwhile, the needs for processing complex data such as relational data has become an important problem, especially in the knowledge discovery in databases field [9], and calls for a formalization of RCA.

In this paper, we propose a global and declarative description of the relational structure within the RCA approach, based on a set of lattices resulting from the processing of the contexts that are successively considered. One feature of the relational structure is that an object lying in the extent of an RCA concept can be connected with another object in the extent of another RCA concept, through a set of relational attributes or links. The inter-concept links can be nested leading to a relational structure of an arbitrary depth. An auxiliary graph structure is defined for covering these inter-object links.

Moreover, as experiences with UML model analysis reveal, the complexity of the final concept descriptions calls for a knowledge representation formalism, for managing and taking into account the semantics of the inter-concept links, e.g. classifying links or checking their consistency. In the second part of the paper, it is shown how concepts and relations from RCA can be mapped into a knowledge base (KB) represented within a DL of the $\mathcal{FL}_0$ family. The connection between the structure of the original data mapped into the ABox of the KB (set of individuals) and the RCA concepts stored in the TBox of the KB (set of concepts) is also studied.

The paper starts with a recall of basic notions from FCA (section 2) and from DL (section 3) that are necessary. Then, the RCA framework is presented in section 4. Section 5 describes the translation of the set of relational concepts into a knowledge base represented within a knowledge representation language. Finally, related work is summarized in section 6.

	(1) Fun95	(2) God93	(3) God95	(4) God98	(5) Huc99	(6) Huc02	(7) Kro94	(8) Kui00	(9) Leb99	(10) Lin95	(11) Lin97	(12) Sah97	(13) Sif97	(14) Sne96	(15) Sne98C	(16) Sne98R	(17) Sne99	(18) Sne00S	(19) Sne00U	(20) Str99	(21) Til03S	(22) Til03T	(23) Ton99	(24) Tone01	(25) Van98
ra																					x				
ad																					x	x			
dd	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x
sm	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x			x	x	x

Table 1. Formal context $\mathcal{K}_{papers}$ of papers. Paper descriptors are: requirement analysis (ra), architectural design (ad), detailed design (dd), software maintenance (sm).

2 FCA basics

FCA is the process of abstracting conceptual descriptions from a set of individuals described by attributes [10]. Formally, a *context* $\mathcal{K}$ associates a set of objects (O) to a set of attributes (A) through an incidence relation $I \subseteq O \times A$. An example of formal context, namely $\mathcal{K}_{papers}$, is depicted in table 1, where O is a set of scientific publications on the applications of FCA in software engineering, and A the set of ISO software engineering activities (this example is adapted from [24]). Two operators, both denoted by $'$, connect the powerset of objects, 2^O and the powerset of attributes 2^A as follows:

$$' : 2^O \to 2^A,\ X' = \{a \in A \mid \forall o \in X, oIa\}$$

The operator $'$ is dually defined on attributes. The pair of $'$ operators induces a Galois connection between 2^O and 2^A [4]. The composition operators $''$ are closure operators and induce two families of closed sets, respectively $\mathcal{C}^o \subseteq 2^O$ and $\mathcal{C}^a \subseteq 2^A$. These two sets, provided with set-inclusion order, form two complete lattices (anti-isomorphic by $'$). A pair (X, Y) where $X \in 2^O$, $Y \in 2^A$, $X = Y'$, and $Y = X'$, is a *(formal) concept*, with X as *extent* and Y as *intent*. The set $\mathcal{C}_\mathcal{K}$ of all concepts extracted from $\mathcal{K}$ ordered by extent inclusion forms a complete lattice, $\mathcal{L}_\mathcal{K} = \langle \mathcal{C}_\mathcal{K}, \leq_\mathcal{K} \rangle$, called the *concept lattice* of the context –or the *Galois lattice*– of the binary relation I). The lattice of $\mathcal{K}_{papers}$ associated with the formal context $\mathcal{K}_{papers}$ is drawn on the left-hand side of Figure 1.

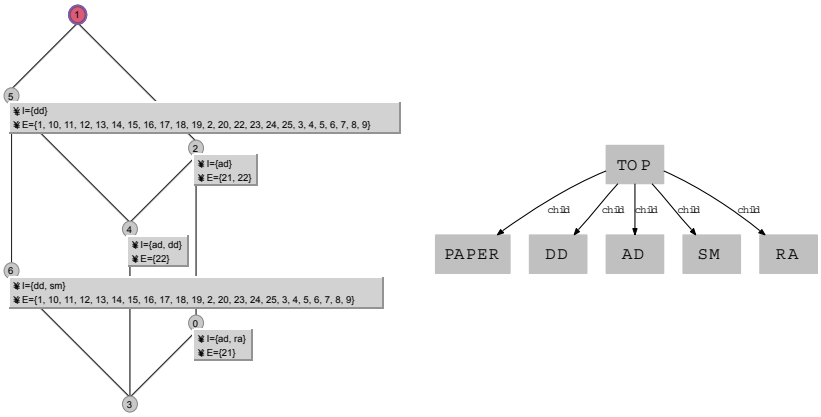

Fig. 1. Left: Initial lattice $\mathcal{L}^0_{papers}$. **Right:** The corresponding `TBox`.

The lattice represents in an exhaustive way the sharing of structures among objects: two objects are in the same concept extent iff they share at least one attribute, meaning, in mathematical terms, that a concept extent may be considered as the intersection of the extent of the attributes associated with the concept. As it will be shown in section 4, RCA relies on a similar principle, extended to relational attributes.

As it will be shown in section 4, RCA relies on a similar closure operator, extended to relational attributes.

Conceptual scaling deals with non-binary data descriptions, organized into a *many-valued context* $\mathcal{K} = (O, A, V, J)$, where J is a ternary relation between objects, attributes and values V [10]. The scaling replaces a many-valued attribute by a set of binary attributes, each one representing a value that the attribute holds. In RCA, as explained farther, scaling is used as well, for structure sharing between referred and referring objects, and managing multi-valued dependencies between objects.

3 Description logics basics

Description logics (DL) are knowledge representation (KR) formalisms based on concepts, roles and individuals [1]. A concept represents a set of individuals while a role determines a binary relationship between concepts. Concepts and roles are designed according to a syntax and a semantics, as in any logic-based formalism. The subsumption relation is a partial ordering relation, used for detecting specialization relations between concepts and roles, and for organizing concepts and roles within a hierarchy. Instance and concept classification are the basic reasoning mechanisms: the former for finding the concepts an individual is an instance of, the latter for searching for the most specific subsumers

(ascendants) and the most general subsumees (descendants) of a concept in the concept hierarchy.

The representation of concepts and roles in DL and in FCA is considered with different points of view: (i) FCA approach is "inductive" and is mainly interested in building concepts from a formal context, (ii) DL approach is "deductive" and is mainly interested in designing concept and inferring subsumption and instantiation relations for reasoning purposes. Accordingly, FCA and DL may play complementary roles in understanding and managing complex data and knowledge units. Attempts for the integration of both approaches may be found in, e.g. [16, 2, 20].

From a practical point of view, a KB in DL consists in a set of concept and role descriptions (respectively comparable to unary and binary predicates in first order predicate logic), that may be primitive or defined (also called a TBox). Primitive concepts are ground description that are used for forming more complex descriptions, the defined concepts, by means of a set of constructors, such as conjunction ($\sqcap$), universal value restriction ($\forall$), existential value restriction ($\exists$), disjunction ($\sqcup$), negation ($\neg$), and others. While a primitive concept can be considered as an atom of the KB, a defined concept is described by a set of conceptual expressions –role introductions– that can be regarded as a set of necessary and sufficient conditions for detecting that an individual is an instance of the concept, allowing for classification-based reasoning.

The choice of a set of constructors has a direct influence on the complexity of the reasoning, i.e. classification and instantiation (see, e.g. [8]). In the following, a simple representation language, called $\mathcal{FL}_0$, is considered, based on the constructors $\sqcap$, $\forall$, the top concept ($\top$, whose extension if the set of all individuals), the bottom concept ($\bot$) (whose extension is the empty set), and concept definition $\equiv$.

For example, based on the data introduced in figure 2, primitive concepts for describing the content of articles on software engineering can be AboutDetailedDesign, AboutMaintenance, AboutArchitecture, and AboutRequirements. A primitive role can be cites, for expressing that a paper cites another paper. In addition, the description of all papers dealing with detailed design and citing only papers on maintenance can be the following concept, denoted by ADD:

$$\mathsf{AboutDetailedDesign} \sqcap \forall \mathsf{cites}.\mathsf{AboutMaintenance}$$

The semantics of the descriptions in a KB is defined by means of an *interpretation*, i.e., a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a set of individuals called the *interpretation domain* and $.^{\mathcal{I}}$ is the *interpretation function*. The later maps a concept description to a subset of $2^{\Delta^{\mathcal{I}}}$ and a role description to a subset of $2^{\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}}$. For example, the interpretation of the concept ADD with respect to an interpretation based on the paper context (in Table 1) and the cite relation (figure 2) is: {1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 23, 24, 25}.

The subsumption relationship between concepts is the main inferential service provided by a DL reasoner: a concept C subsumes D, denoted $D \sqsubseteq C$, iff $D^{\mathcal{I}} \subseteq C^{\mathcal{I}}$. Subsumption is a pre-ordering relation, that can be considered as a partial

ordering up to an equivalence, two concepts being equivalent as soon as the first subsumes the second, and reciprocally, i.e. they have the same set of instances. A concept definition $A \equiv C$ assigns a concept description C to a concept name A. In this way, a KB may be composed of a TBox a$\mathcal{T}$ and an ABox $\mathcal{A}$, where assertions about the actual individuals are stored. Assertions are of two kinds: concept and role names are respectively used as unary and binary predicates, for describe the relations between individuals and concepts, e.g. AboutDetailedDesign(12) and cite(12,13) belong to the ABox representing the current example on scientific papers.

It can be noticed that DL concept descriptions share with formal concept intents from FCA a similar descriptive pattern: both are conjunctions of descriptors which act as predicates on individuals or objects. This observation calls for the introduction of relation between objects in FCA, similar to the relations between individuals in DL, materialzed by roles. In this way, a direct mapping could be obtained by concept description produced by FCA and DL-based description, leading to FCA as a method for building an ontology from data [23, 5, 19]. This is the purpose of the next section.

4 Relational Concept Analysis

Relational Concept Analysis (RCA) is an original approach for extracting formal concepts from sets of data described by attributes and relational attributes. In this section, the formal background of RCA is introduced and detailed.

4.1 Data Model

In RCA, data are organized within a structure composed of a set of contexts $\mathbf{K} = \{\mathcal{K}_i\}$ and of a set of binary relations $\mathbf{R} = \{r_k\}$, where $r_k \subseteq O_i \times O_j$, O_i and O_j being sets of objects (respectively in $\mathcal{K}_i$ and $\mathcal{K}_j$). The structure $(\mathbf{K}, \mathbf{R})$ is called a *relational context family* (RCF) and can be compared to a relational database schema, including both classes of individuals and classes of relations. The following definition gives a formal account of RCF.

Definition 1. *A relational context family $\mathcal{R}$ is a pair $(\mathbf{K}, \mathbf{R})$, where $\mathbf{K}$ is a set of contexts $\mathcal{K}_i = (O_i, A_i, I_i)$, $\mathbf{R}$ is a set of relations $r_k \subseteq O_i \times O_j$ where O_i and O_j are the object sets of the formal contexts $\mathcal{K}_i$ and $\mathcal{K}_j$.*

A relation $r \subseteq O_i \times O_j$ can be seen as a set-valued function $r : O_i \rightarrow 2^{O_j}$. Two functions are defined on relation sets in RCF, *domain* and *range*:

- $dom : \mathbf{R} \rightarrow \mathbf{O}$ with $dom(r : O_i \rightarrow 2^{O_j}) = O_i$
- $ran : \mathbf{R} \rightarrow \mathbf{O}$ with $ran(r : O_i \rightarrow 2^{O_j}) = O_j$,

where $\mathbf{O}$ is the set of all object sets in the RCF, $\mathbf{O} = \{O | \mathcal{K} = (O, A, I) \in \mathbf{K}\}$. Moreover, an auxiliary function maps a context into the set of all relations whose domain corresponds to the object set of the context:

$$rel : \mathbf{K} \rightarrow 2^{\mathbf{R}}; \quad rel(\mathcal{K} == (O, A, I)) = \{r | dom(r) = O\}.$$

The instances of a relation r_k, say $r_k(o_i, o_j)$, where $o_i \in O_i$ and $o_j \in O_j$, are called *links*. For example, the figure 2 shows the binary table of the `cite` relation on the paper example, thus the set of links that are considered in the following. The links can be "scaled" in order to be included as binary attributes in a formal original context, through a mechanism called *relational scaling* and explained in the following sections.

	Fun95	God93	God95	God98	Huc99	Huc02	Kro94	Kui00	Leb99	Lin95	Lin97	Sah97	Sif97	Sne96	Sne98C	Sne98R	Sne99	Sne00S	Sne00U	Str99	Til03S	Til03T	Ton99	Tone01	Van98
(1) Fun95							x																		
(2) God93																									
(3) God95		x																							
(4) God98		x	x				x																		
(5) Huc99		x																							
(6) Huc02		x														x			x						
(7) Kro94																									
(8) Kui00											x		x	x	x	x		x							x
(9) Leb99		x																							
(10) Lin95							x																		
(11) Lin97	x						x							x											
(12) Sah97			x								x		x												
(13) Sif97											x	x		x											
(14) Sne96							x			x															
(15) Sne98C	x						x			x	x		x	x											
(16) Sne98R		x					x				x		x	x	x										
(17) Sne99	x	x		x			x				x		x	x	x	x									
(18) Sne00S							x				x		x	x		x	x								x
(19) Sne00 U		x		x			x				x			x	x	x									
(20) Str99														x	x	x									
(21) Til03S											x					x									
(22) Til03T																									
(23) Ton99							x				x		x	x											
(24) Tone01							x				x		x	x			x								
(25) Van98											x		x		x	x									

Fig. 2. The table of citations between papers of the formal context $\mathcal{K}_{papers}$.

To solve the obvious identification mismatches, all the elements that are manipulated here — contexts, objects, attributes, both initial and relational ones — are assumed to hold *unique names* within a name space holding for the entire RCF. Contexts are uniquely determined by their respective object sets which remain invariant during the relational analysis process. Hence we can speak about the 'same' context on tho different iterations although these may well be two different relations as the attribute sets could diverge. In the same

way, formal concepts are uniquely determined by their extents, which, once the concept is created remain the same along the iterations while intents can grow after each relational scaling step. The addition of new attributes in a formal context leads to an expansion of the underlying lattice: the augmented lattice contains a join-sub-semi-lattice isomorphic to the original one [27, 15].

4.2 Scaling of relations

Provided that the intent of a formal concept is modeled upon concept descriptions in DL which possibly include expressions like $\forall\ \mathsf{r}.\mathsf{C}$ or $\exists\ \mathsf{r}.\mathsf{C}$, where C is the name of a concept, the attribute set of a relational scale should be based on concept names. Thus, given a relation r of the RCF with $dom(r) = O_i$ and $ran(r) = O_j$, new attributes will be added to the context $\mathcal{K}_i$ via r. Although several variants exist, the most reliable source of concept names is the lattice of the context $\mathcal{K}_j$ underlying the object set O_j. Hence, the relation r basically introduces the abstractions from $\mathcal{K}_j$ into $\mathcal{K}_i$. The resulting attributes should clearly bear an indication of the relation that generated them. Now, it only remains to fix the circumstances under which such an attribute is awarded to an object from $\mathcal{K}_i$.

Intuitively, the scaling works as follows. First, observe that the value of r for $o \in O_i$ is a set of objects in O_j ($r(o) \subseteq O_j$). Strictly following the DL model, a scale attribute combining a relation r with a formal concept $c = (X, Y)$ from the lattice $\mathcal{L}_j$ is assigned to an object $o \in O_i$ whenever $r(o)$ is "correlated" with the extent of c. Strong correlation means inclusion whereas an alternative is to search for non-empty intersection between both sets. The two corresponding encoding schemes for relational scaling are called *narrow* (for$\forall\ \mathsf{r}.\mathsf{C}$) and *wide* (for $\exists\ \mathsf{r}.\mathsf{C}$) encodings.

In mathematical terms, given $\mathcal{K}_i = (O_i, A_i, I_i)$, the scaling of $\mathcal{K}_i$ for a relation $r \in rel(\mathcal{K}_i)$ such that $ran(r) = O_j$ with respect to the lattice $\mathcal{L}_j$ yields an extension of A_i and I_i, but keeps O_i unchanged. The attributes added to A_i are of the form $r : C$, and this is made precise in the definition below:

Definition 2. *Given a relation $r \in rel(\mathcal{K}_i)$ and a lattice $\mathcal{L}_j$ on $\mathcal{K}_j = ran(r)$, the narrow scaling operator $sc_{\times}^{(r,\mathcal{L}_j)} : \mathbf{K} \to \mathbf{K}$ is defined as follows:*

$$sc_{\times}^{(r,\mathcal{L}_j)}(\mathcal{K}_i) = (O_i^{(r,\mathcal{L}_j)}, A_i^{(r,\mathcal{L}_j)}, I_i^{(r,\mathcal{L}_j)})$$

where $O_i^{(r,\mathcal{L}_j)} = O_i$, $A_i^{(r,\mathcal{L}_j)} = A_i \cup \{r : c | c \in \mathcal{L}_j\}$, and
$I_i^{(r,\mathcal{L}_j)} = I_i \cup \{(o, r : c) | o \in O_i, c \in \mathcal{L}_j, r(o) \neq \emptyset, r(o) \subseteq extent(c)\})$.

The *wide scaling operator* $sc_{+}^{(r,\mathcal{L}_j)}$ is defined in a way similar to the narrow scaling operator. The only difference lies in the incidence relation for $sc_{+}^{(r,\mathcal{L}_j)}(\mathcal{K}_i)$ which is defined as follows:

$$I_i \cup \{(o, r : C) | o \in O_i, c \in \mathcal{L}_j, r(o) \cap extent(c) \neq \emptyset\}).$$

However, in the following, only narrow scaling is considered. For example, suppose that the context $\mathcal{K}_{papers}$ has to be scaled with respect to the lattice given in Fig. 1. The papers exclusively citing papers on detailed design (dd attribute) correspond to the concept whose intent exclusively includes dd, i.e. the concept denoted by 5. The papers lying in the extent of 5 are all papers except 21. Accordingly, in the scaled context, these objects are associated with the attribute r through the construction $\forall$ r.C, with r being cite and C the concept 5, i.e. $\forall$ r.C.

The complete relational scaling of a context $\mathcal{K}_i$ is the scaling of all the relations in $rel(\mathcal{K}_i)$. Considering a context $\mathcal{K}_i$, the relation set $rel(\mathcal{K}_i) = \{r_l\}_{l=1..p_i}$, and $\mathcal{L}_{j_l}$ the lattice associated to the context of the range object set for r_l for each l in $[1, p_i]$, the result of the scaling of $\mathcal{K}_i$ on all pairs $(r_l, \mathcal{L}_{j_l})$ is denoted by:

$$\mathcal{K}_i^{rel} = sc_{\times}^{(r_1,\mathcal{L}_{j_1})}(sc_{\times}^{(r_2,\mathcal{L}_{j_2})}(\ldots sc_{\times}^{(r_{k_i},\mathcal{L}_{j_{p_i}})}(\mathcal{K}_i)))$$

Thus, when all the contexts of a RCF are scaled for lattice construction, a scaled version of a context $\mathcal{K}_j$ may possibly be no more consistent with a prior scaling $sc_{\times}^{(r,\mathcal{L}_j)}$. When there is no loop within relations in the whole RCF, the inconsistent situation may be avoided by properly ordering contexts and the associated lattice construction tasks. A general method for constructing the lattices of a RCF is presented in the next section.

4.3 Lattice construction

The construction of the set of lattices associated to a RCF is an iterative fixed-point-bound process that alternates pure lattice construction and expansion of the contexts through relational scaling. The process starts with a "bootstrapping" step, i.e., lattice construction on each context which processes all formal objects exclusively with their original binary attributes while ignoring all the relational information. The resulting lattices provide the basis for relational scaling which is universally applied on every relation of the RCF at the next iteration. A new step of lattice construction is then carried on, followed by a new scaling step, and this goes on until the lattices stop evolving between iterations (two consecutive steps produce lattices that are isomorphic). This means that a fixed point for the RCF scaling operator has been met, and the computation ends up.

More formally, consider the evolution of the content of each context, i.e., the gradual increase of its attribute set and hence of its set of incidence pairs along the iterative process. Given a context $\mathcal{K}_i$ from the RCF, its evolution is captured in a series of context contents $\mathcal{K}_i^p$. The starting element is the original configuration of the context $\mathcal{K}_i$, $\mathcal{K}_i^0 = (O_i, A_i^0, I_i^0)$. Each subsequent member of the series is obtained from the previous one by complete relational scaling: $\mathcal{K}_i^{p+1} = (\mathcal{K}_i^p)^{rel_p}$, where the $\_^{rel_p}$ operator denotes the fact that the scaling is computed with respect to the content of the RCF at the step p of the process. The series satisfies an important monotony property. Indeed, although at each step a different set of new attributes is added to the context, it may be proven that the concepts which have appeared at a step k of the process, will not disappear

on further steps. Hence the sizes of the concept set in the subsequent variants of $\mathcal{K}_i$ from the above series form a non-deceasing series themselves. However, the size of the set is bound from above by $2^{|O_i|}$, for obvious reasons. Therefore, the series converges towards $\mathcal{K}_i^{\infty}$, which represents the least fixed-point of the scaling operator. This fact guarantees a termination of the global analysis process.

Furthermore, to express the global evolution of the context set within its RCF, a composite operator is defined for $\mathbf{K}^n$, that is considered as a vector of contexts. The operator, denoted $.^{\mathbf{r}el_p^*}$, denotes the application of $.^{rel_p}$ to all contexts in $\mathbf{K}$. Here again, a series $\mathbf{K}^n$ can be defined by $\mathbf{K}^0 = \mathbf{K}$, $\mathbf{K}^{p+1} = (\mathbf{K}^p)^{\mathbf{r}el_p^*}$ for all $p \geq 0$. The resulting series has an upper bound since all component series are upper bounded. The series is non-decreasing as well, and thus has a limit. This limit, denoted by $\mathbf{K}^{\infty}$, is the element where scaling produces no more concepts in any of the contexts. In the paper example, the final lattice $\mathcal{L}_{papers}^{\infty}$ is given in Fig. 3. s

From a pure computational standpoint, one knows the fixed point is reached whenever at two subsequent steps all the pairs of corresponding lattices remained isomorphic.

5 Mapping the RCA constructs to a DL KB

RCA provides relational descriptions that can be fully exploited by means of a representation formalism supporting reasoning, classification, instance and consistency checking. Hence the choice of the DL formalism is totally appropriate and this is shown and discussed in the section hereafter.

In the following, the DL formalism that is considered is $\mathcal{FL}_0$, where constructors are conjunction, universal quantification, top and bottom concepts, with the introduction of defined concepts. A knowledge base in $\mathcal{FL}_0$ is a pair (TBox,ABox), denoted by $\mathcal{B} = (\mathcal{T}(\mathcal{T}_C, \mathcal{T}_R), \mathcal{A})$, can be designed w.r.t. a RCF $\mathcal{R}$ and the corresponding set of final lattices $\mathbf{L} = \{\mathcal{L}_i\}$ in the following way. First, the set of symbols in $\mathcal{R}$ for context, attribute, object and relation (relational attribute) names are mapped into $\mathcal{B}$ by a bijection α as follows:

- $\forall \mathcal{K}_i \in \mathbf{K}$: $\alpha[\mathcal{K}_i] \in \mathcal{T}_C$,
- $\forall a_i \in A_i$: $\alpha[a_i] \in \mathcal{T}_C$,
- $\forall r_i \in \mathbf{R}$: $\alpha[r_i] \in \mathcal{T}_R$,
- $\forall o_i \in O_i$: $\alpha[o_i] \in individuals(\mathcal{A})$,
- $\forall c_i \in \mathcal{L}_i^{\infty}$: $\alpha[c_i] \in definitions(\mathcal{T})$.

This means that contexts and attributes become primitive concepts, relations (relational attributes) become roles, objects become individuals (constants), and formal concepts in $\mathcal{L}_i^{\infty}$ become defined concepts. Based on this mapping, the actual content of the KB is created.

On the basis of this transformation, the ground facts of the ABox are constructed. Hereafter is presented the translation of all initial incidence facts from A_i^0, all incidences between contexts and objects, the basic relational links, and the concept extents:

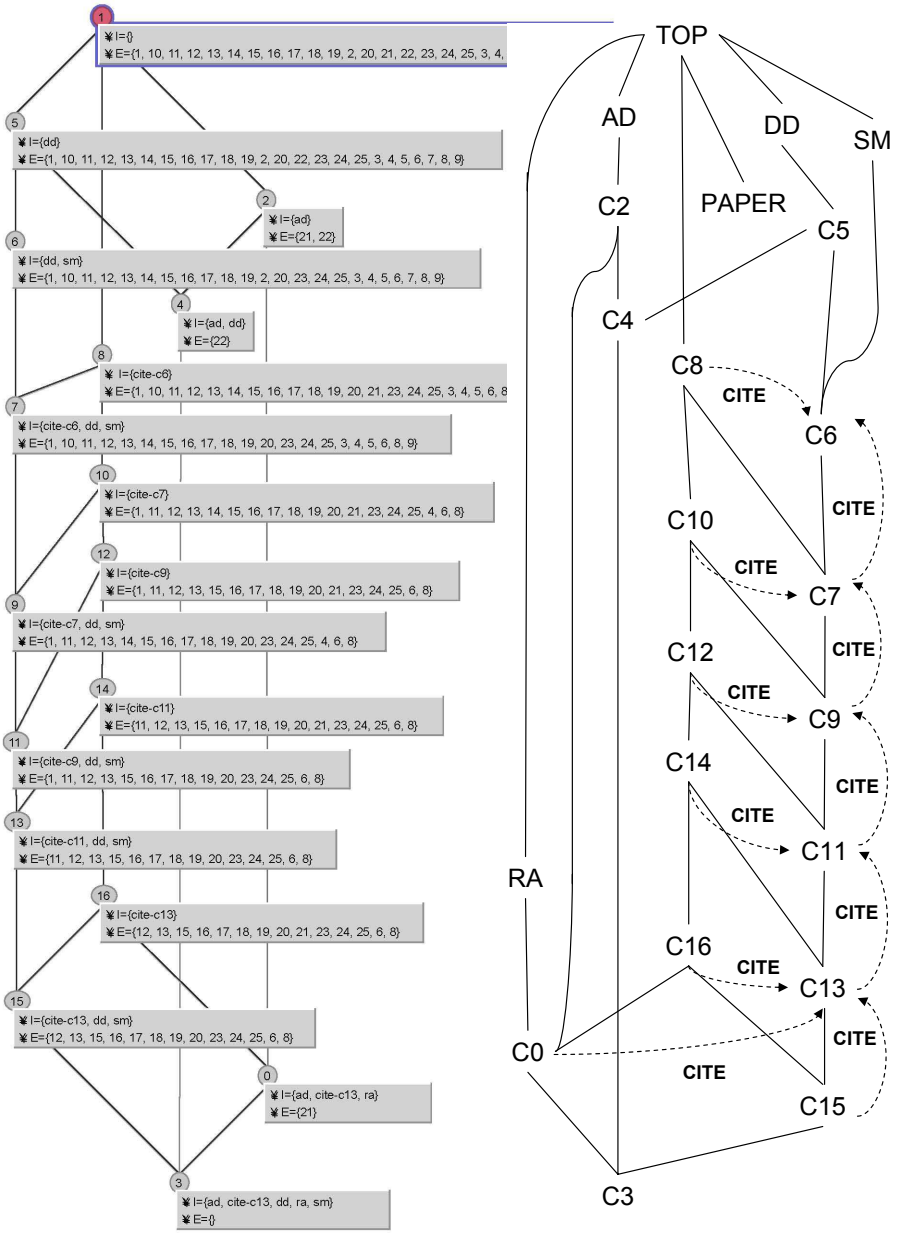

Fig. 3. Left: The final relational lattice $\mathcal{L}^{\infty}_{papers}$. **Right:** The corresponding `TBox`.

– $\forall a_i \in A_i^0, \forall o_i \in a_i'$: $\alpha[a_i](\alpha[o_i]) \in \mathcal{A}$,
– $\forall o_i \in O_i$: $\alpha[\mathcal{K}_i](\alpha[o_i]) \in \mathcal{A}$,
– $\forall r_i \in \mathbf{R}, \forall o_1, o_2 / r(o_1, o_2)$: $\alpha[r_i](\alpha[o_1], \alpha[o_1]) \in \mathcal{A}$,
– $\forall c_i \in \mathcal{L}_i^\infty, \forall o_i \in ext(c_i)$: $\alpha[c_i](\alpha[o_i]) \in \mathcal{A}$.

Finally, the TBox is composed of the translation of the intents of the RCA concepts where attributes rooted in relational scaling are translated into role restrictions:

– $\forall a_{i,j} = r_k \in c_{m,n} \wedge c_{m,n} \in A^\infty \setminus A_i^0$: $\alpha[a_{i,j}] = \forall \alpha[r_k].\alpha[c_{m,n}] \in \mathcal{T}$,
– $\forall c_{i,j} \in \mathcal{L}_i^\infty$: $\alpha[c_{i,j}] \equiv \sqcap_{a_{i,j} \in int(c_{i,j})} \alpha[a_{i,j}] \in \mathcal{T}$.

The connection between the data part in $\mathcal{A}$ and the schema part in $\mathcal{T}$ can be made explicit. In this way, the interpretation domain $\Delta^\mathcal{I}$ is identified to the ABox $\mathcal{A}$, meaning that concept descriptions are interpreted in terms of individuals in the ABox, and role descriptions in terms of pairs of individuals. Moreover, the formal concept extents in $\mathcal{L}_i^\infty$ have been explicitly translated into facts of the ABox. A question remains whether all relations in the data have been expressed in the $\mathcal{FL}_0$ KB $= (\mathcal{T}, \mathcal{A})$. The answer is that under some very reasonable hypotheses, the set of all possible semantics, i.e., Actually, object sets that can be described by a formula of $\mathcal{FL}_0$ with concept and role names in $\mathcal{T}(\mathcal{T}_C, \mathcal{T}_R)$, are the TBox $\mathcal{T}$ constructed as above.

This can be stated in the following property. Given an arbitrary description D in $\mathcal{FL}_0$, with $\mathcal{T}(\mathcal{T}_C, \mathcal{T}_R)$, there exists a concept C in $\mathcal{T}((\mathcal{T}_C, \mathcal{T}_R)$ such that D and C are semantically equivalent for the model provided by the ABox, i.e. $D^\mathcal{A} \equiv C^\mathcal{A}$.

6 Related work

Extensions of the classical, i.e., binary, model of FCA to complex data descriptions, e.g., including relational information, have been studied from various standpoints. For example, in the *power context families* framework [17], inter-object links are regarded as higher-order entities and grouped into formal concepts. Although a uniform way of processing n-ary links comes out of this approach, the resulting layered representation of regularities (separate concept lattices for each n) is hard to map to a classical knowledge representation language where links are mostly binary and they are combined with objects' own properties.

Graph-based data descriptions are tackled in [12–14] whose authors propose efficient extensions of the FCA machinery to complex data formats described as graphs, e.g. chemical compound models, conceptual graphs, RDF triples. Graphs, although complexly structured, are comparable to simple individuals in that they do not refer to other graphs (except for nested graphs which have yet not been studied for knowledge discovery purposes). Hence these works do not face reference porblems and cicularity as we have to. Consequently, they cannot — and are not intended to — provide DL-compatible concept descriptions as we do.

Previous studies of combining FCA and DL [20, 2] have been focusing on the construction of the concept intents out of a static set of DL primitives, i.e., concept and role names. In contrast, our approach is dynamic in the sense that new concept are dicovered all along the analysis process whose names are then used in the descriptions of further concepts, thus potentially creating in the so called *terminological cycles*. Interstingly enough, such cycles have already been tackled with FCA-based techniques in the study reported by F. Baader in [3]. However, our approach is based on a different way of establishing the semantics of the cyclic descriptions, fixed point one as opposed to the descriptive semantics used by Baader.

7 Discussion

In this paper, an extension of FCA to the representation and manipulation of relational data has been proposed leading to a framework we called relational concept analysis. Moreover, the concepts and relations extracted with RCA techniques can be easily translated into DL knowledge base, allowing reasoning and problem-solving.

The RCA approach has been implemented in the GALICIA platform[1] [25], and validated within an application to software re-engineering [6]. The tool is operational for relatively small datasets as scalability is a concern. Indeed, the lattice of a context with some relational components can grow even larger because of the additional combinatorics brought by the links. So far, classical techniques for complexity reduction have been used such as iceberg lattices or Galois sub-hierarchies (which thaditionally work well for class hierarchies datasets in software engineering). Computational cost has been fought by means of incremental lattice construction whereby the principle is straightforward: only new concepts at step k are used for scaling at the followong step.

A challenging research track leads to the coupling of GALICIA with a DL reasoner for knowledge representation purposes, and for ontology and software engineering.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, Cambridge, UK, 2003.
2. F. Baader and B. Sertkaya. Applying formal concept analysis to description logics. In P. Eklund, editor, *Second International Conference on Formal Concept Analysis (ICFCA 2004), Sydney*, Lecture Notes in Artificial Intelligence 2961, pages 261–286. Springer, Berlin, 2004.
3. Franz Baader. Computing the least common subsumer in the description logic $\mathcal{EL}$ w.r.t. terminological cycles with descriptive semantics. In *Proceedings of the 11th International Conference on Conceptual Structures (ICCS 2003)*, volume 2746 of *Lecture Notes in Artificial Intelligence*, pages 117–130. Springer-Verlag, 2003.

[1] `http://sourceforge.net/projects/galicia/`.

4. M. Barbut and B. Monjardet. *Ordre et classification – Algèbre et combinatoire, Tome 2*. Hachette, Paris, 1970.
5. P. Cimiano, A. Hotho, G. Stumme, and J. Tane. Conceptual Knowledge Processing with Formal Concept Analysis and Ontologies. In P.W. Eklund, editor, *Concept Lattices, Second International Conference on Formal Concept Analysis (ICFCA 2004), Sydney, Australia*, Lecture Notes in Computer Science 2961, pages 189–207. Springer, 2004.
6. M. Dao, M. Huchard, M. R. Hacene, C. Roume, and P. Valtchev. Towards practical tools for mining abstractions in UML models. In *Proceedings of the 8th ICEIS'06, Paphos (CY), May*, pages 276–283, 2006.
7. M. Dao, M. Huchard, M. Rouane Hacene, C. Roume, and P. Valtchev. Improving Generalization Level in UML Models: Iterative Cross Generalization in Practice. In K.E. Wolff, H.D. Pfeiffer, and H.S. Delugach, editors, *Conceptual Structures at Work: Proceedings of the 12th International Conference on Conceptual Structures, ICCS 2004, Huntsville, AL*, Lecture Notes in Artificial Intelligence 3127, pages 346–360. Springer, Berlin, 2004.
8. F.M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. *Information and Computation*, 134(1):1–58, 1997.
9. S. Dzeroski and N. Lavrac, editors. *Relational Data Mining*. Springer, Berlin, 2001.
10. B. Ganter and R. Wille. *Formal Concept Analysis*. Springer, Berlin, 1999.
11. M. Huchard, C. Roume, and P. Valtchev. When concepts point at other concepts: the case of UML diagram reconstruction. In V. Duquenne, B. Ganter, M. Liquiere, E. Mephu-Nguifo, and G. Stumme, editors, *Proceedings of the Workshop on Formal Concept Analysis for Knowledge Discovery in Databases (FCAKDD at ECAI-02)*, pages 32–43, 2002.
12. S.O. Kuznetsov. Learning of simple conceptual graphs from positive and negative examples. In *Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery (PKDD '99), Prague, Czech Republic*, Lecture Notes in Computer Science 1704, pages 384–391. Springer, Berlin, 1999.
13. S.O. Kuznetsov. Machine learning and formal concept analysis. In P.W. Eklund, editor, *Concept Lattices, Second International Conference on Formal Concept Analysis (ICFCA 2004), Sydney, Australia*, Lecture Notes in Computer Science 2961, pages 287–312. Springer, 2004.
14. M. Liquiere and J. Sallantin. Structural Machine Learning with Galois Lattice and Graphs. In J.W. Shavlik, editor, *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconson*, pages 305–313. Morgan Kaufmann, 1998.
15. K. Nehmé, P. Valtchev, M.H. Rouane, and R. Godin. On computing the minimal generator family for concept lattices and icebergs. In B. Ganter and R. Godin, editors, *Proceedings of the Third International Conference on Formal Concept Analysis (ICFCA 2005), Lens, France*, Lecture Notes in Computer Science 3403, pages 192–207, 2005.
16. S. Prediger and G. Stumme. Theory-driven logical scaling: conceptual information systems meet description logics. In E. Franconi and M. Kifer, editors, *Proceedings of the 6th International Workshop on Knowledge Representation meets Databases (KRDB'99), Linköping Sweden*, 1999.
17. S. Prediger and R. Wille. The lattice of concept graphs of a relationally scaled context. In W.M. Tepfenhart and W.R. Cyre, editors, *Proceedings of the 7th International Conference on Conceptual Structures (ICCS'99), Blacksburg, Virginia*, Lecture Notes in Computer Science 1640, pages 401–414. Springer, Berlin, 1999.

18. U. Priss. *Relational Concept Analysis: Semantic Structures in Dictionaries and Lexical Databases*. PhD thesis, Aachen University, 1996.
19. T.T. Quan, S.C. Hui, A.C.M. Fong, and T.H. Cao. Automatic generation of ontology for scholarly semantic web. In S.A. McIlraith, D. Plexousakis, and F. Van Harmelen, editors, *International Conference on Semantic Web, ISWC 2004, Hiroshima, Japan*, Lecture Notes in Computer Science 3298, pages 726–740. Springer, 2004.
20. S. Rudolph. Exploring relational structures via fle. In K.E. Wolff, H.D. Pfeiffer, and H.S. Delugach, editors, *Conceptual Structures at Work: 12th International Conference on Conceptual Structures Proceedings (ICCS 2004), Huntsville, AL*, Lecture Notes in Computer Science 3127, pages 261–286. Springer, Berlin, 2004.
21. J.F. Sowa, editor. *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1991.
22. G. Stumme. Formal concept analysis on its way from mathematics to computer science. In U. Priss, D. Corbett, and G. Angelova, editors, *Conceptual Structures: Integration and Interfaces, Proceedingsof the 10th International Conference on Conceptual Structures (ICCS 2002), Borovets, Bulgaria*, Lecture Notes in Artificial Intelligence 2393, pages 2–19, Berlin, 2002. Springer.
23. Gerd Stumme and Alexander Maedche. FCA-MERGE: Bottom-up merging of ontologies. In *Proceedings of IJCAI'01, Seattle (WA)*, pages 225–234, 2001.
24. T. Tilley, R. Cole, P. Becker, and P. Eklund. A survey of formal concept analysis support for software engineering activities. In *Proceedings of the First International Conference on Formal Concept Analysis, Darmstadt, Germany*. Springer Verlag, February 2003.
25. P. Valtchev, D. Grosser, C. Roume, and M.H. Rouane. Galicia: an open platform for lattices. In A. de Moor, W. Lex, and B. Ganter, editors, *Contributions to the 11th International Conference on Conceptual Structures (ICCS'03), Dresden, Germany*, pages 241–254. Shaker Verlag, 2003.
26. P. Valtchev, R. Missaoui, and R. Godin. Formal concept analysis for knowledge discovery and data mining: The new challenges. In P.W. Eklund, editor, *Concept Lattices, Second International Conference on Formal Concept Analysis(ICFCA 2004), Sydney, Australia*, Lecture Notes in Computer Science 2961, pages 352–371. Springer, 2004.
27. P. Valtchev, R. Missaoui, and P. Lebrun. A partition-based approach towards constructing galois (concept) lattices. *Discrete Mathematics*, 256(3):801–829, 2002.
28. R. Wille. Methods of conceptual knowledge processing. In R. Missaoui and J. Schmid, editors, *International Conference on Formal Concept Analysis, ICFCA 2006, Dresden, Germany*, Lecture Notes in Artificial Intelligence 3874, pages 1–29. Springer, 2006.