Generating GCIs Axioms from Objects Descriptions in $\mathcal{EL}$-Description Logics

Zina Ait-Yakoub[1] and Yassine Djouadi[2]

[1] Department of Computer Science
University of Tizi-Ouzou, BP 17, RP, Tizi-Ouzou, Algeria
ait.yakoub@hotmail.fr
[2] IRIT Toulouse
118 Route de Narbonne, 31062 Toulouse, Cedex 9, France
djouadi@irit.fr

Abstract. Description Logic are well appropriate for knowledge representation. In such a case, intensional knowledge of a given domain is represented in the form of a terminology (TBox) which declares general properties of concepts relevant to the domain. The terminological axioms which are used to describe the objects of the considered domain are usually manually entered. Such an operation being tiresome, Formal Concept Analysis (FCA) has been already used for the automatic learning of terminological axioms from object descriptions (i.e. from concept instances). However, in all existing approaches, induced terminological axioms are exclusively restricted to the conjunctive form, that is, the existential constructor ($\exists r.C$) is not allowed. In this paper, we propose a more general approach that allows to learn existentially quantified general concept inclusion (GCIs) axioms from object descriptions given as assertions in the $\mathcal{EL}$ language.

1 Introduction

Description Logics (DLs) [8] 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 [12] as standard ontology language for the semantic web [14]. 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 concept descriptions using an appropriate concept description language.

Actually, the construction of ontologies is usually performed manually by knowledge engineers. Such a construction is a tedious and tricky task. However, the most important arising problem concerns the computing of a minimal Terminological Base (TBox) of general concept inclusion axioms.

Based on the lattice theory, Formal Concept Analysis (FCA) [9] can be used to analyze data given in the form of a formal context. Particularly, FCA provides efficient algorithms for computing a minimal basis of all implications holding in a given formal context. In this spirit, FCA has been already used for the automatic generation of terminological axioms from object descriptions (i.e. from concept instances). In such a case, implications are assimilated to terminological axioms.

A. Amine et al. (Eds.): *Modeling Approaches and Algorithms*, SCI 488, pp. 75–84.
DOI: 10.1007/978-3-319-00560-7_12

However, it may be remarked that in all existing approaches, generated terminological axioms are restricted to the conjunctive form of the $\mathcal{EL}$-language. That is, the existential constructor ($\exists r.C$) is not allowed for such generated terminological axioms.

In an original way, we propose in this paper a more general approach that allows to induce the minimal set of all general concept inclusion (GCIs) axioms from object descriptions given as assertions. That is, unlike existing approaches, our method allows to generate GCIs existentially quantified.

This work, which constitutes a first attempt in this direction, will make use of a simple DL language, namely the $\mathcal{EL}$ one. Note that, our proposed approach is not restricted to $\mathcal{EL}$. It will certainly be possible to generalize it to other DLs, which will be investigated in further researches.

This paper is organized as follows. Section 2 gives a background on formal concept analysis. Section 3 presents description logics, whereas section 4 relates previous works on the common utilization of FCA and DL. Our proposition is presented in section 5. An illustrative example is also given in the following section. We conclude and present our future researches.

2 Formal Concept Analysis

Formal concept analysis (FCA for short) [9] consists of inducing granules of knowledge called formal concepts from an $Objects \times Attributes$ binary relation. FCA relies on the notion of a formal context which consists of a triple $K = (G, M, I)$ where G is the set of objects, M the set of attributes and I the binary relation s.t. $I \subseteq G \times M$. $(g, m) \in I$ means that the object g satisfies the attribute m. Relation I can be viewed as a table where, for instance, rows correspond to objects, columns to properties, and a table entry contains "×" or nothing, depending on whether the object satisfies or not the corresponding attribute.

Given a formal context $K = (G, M, I)$ and given two sets $A \in 2^G$ and $B \in 2^M$, a powerset operator $(.)'$ (called also Galois derivation operator) is dually defined among the sets 2^G and 2^M as follows:

$$A' = \{m \in M | (g, m) \in I \text{ for all } g \in A\}$$
$$B' = \{g \in G | (g, m) \in I \text{ for all } m \in B\}$$

A formal concept of K is a pair $\langle A, B\rangle$ with $A \subseteq G, B \subseteq M$ such that $A' = B$ and $B' = A$. A is called the extent and B the intent of the formal concept $\langle A, B\rangle$. The set of all formal concepts is denoted by $\mathcal{L}(G, M, I)$. A formal concept corresponds to a maximal rectangle full of crosses in the table representing a formal context. For brevity, we write g' and m' instead of $\{g\}'$ and $\{m\}'$ respectively.

From a formal context, one may also induce the so-called *attribute implications* [9]. Let $K = (G, M, I)$ be a formal context and $P, C \in 2^M$, an attribute implication of the form $P \rightarrow C$ is defined as:

$$P \rightarrow C \Leftrightarrow P' \subseteq C' \Leftrightarrow C \subseteq P'')$$

That is, for every g from G: if every attribute from the premise P applies to the object g, then every attribute from the conclusion C also applies to g. The set P being called the premise and C being called the conclusion of this implication.

Since the set of all attribute implications may contain many redundancies, it is more appropriate to use a condensed (i.e. minimal) representation. Among such representations, the Duquenne-Guigues [7] basis is a minimal set of implication from which we can find every other implications that hold through inference.We give hereafter its definition.

Definition 1. *The set of implications* $\{ X \rightarrow X'' \mid X$ *is a Pseudo-Intent* $\}$ *is called a Duquenne-Guigues Basis.*

Where a pseudo-intent is defined as:

Definition 2. *For a formal context* (G, M, I)*, a set* $P \subseteq M$ *will be called Pseudo-Intent if* $P'' \neq P$ *and* $Q'' \subseteq P$ *holds for every Pseudo-Intent* $Q \subset P$.

3 Description Logics

Description logics [8] (DL for short) are decidable fragments of first-order logic used to represent and reason on knowledge. In order to define concepts in a DL knowledge base, one starts with a set of concept names (unary predicates) N_C, a set of role names (binary predicates) N_R and a set of objects names N_O. The DL paradigm aims to build concept descriptions or, in short, concepts using constructors. The set of constructors determine the expressive power of the DL. In this paper, we restrict ourselves to the DL $\mathcal{EL}$, whose every concept name is a concept description and, for any concept description C and D and any role r, top-concept ($\top$), conjunction ($C \sqcap D$), and existential restriction ($\exists r.C$) are also concept descriptions. The semantics of $\mathcal{EL}$-concept descriptions is defined in terms of an interpretation $\mathcal{I} =< \triangle^{\mathcal{I}}, .^{\mathcal{I}} >$. The domain $\triangle^{\mathcal{I}}$ of $\mathcal{I}$ is a non-empty set of individuals (objects) and the interpretation function $.^{\mathcal{I}}$ maps each concept name $A \in N_C$ to a subset $A^{\mathcal{I}}$ of $\triangle^{\mathcal{I}}$ and each role $r \in N_R$ to a binary relation $r^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$.

A knowledge base consists of an ABox and a TBox. An ABox (Assertions Box) is a finite set of assertions of the form $A(a)$ or $r(a, b)$, where A is a concept name, r is a role name, and a, b are individual names from a set N_O. Interpretations of ABoxes must additionally map each individual name $a \in N_O$ to an element $a^{\mathcal{I}}$ of $\triangle^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is a model of the ABox A iff it satisfies all its assertions, i.e., $a^{\mathcal{I}} \in A^{\mathcal{I}}$ for all concept assertions $A(a)$ in A and $(a^{\mathcal{I}}, b^{\mathcal{I}})$ for all role assertions $r(a, b)$ in A.

A TBox (Terminology Box) represents intensional knowledge of a problem domain which declares general properties of concepts relevant to the domain. One may distinguish two types of Tboxes. In the most basic type, a TBox contains *concept definitions* of the form $A = C$ which define a concept name A by a concept description C. Concept descriptions are terms built from primitive concepts by means of language constructors provided by the DL.

In the more general type, TBoxes contain universally true implications, so-called *general concept inclusion* (GCI) axioms of the form $C \sqsubseteq D$, where both C and D are arbitrary concept descriptions. A GCI $C \sqsubseteq D$ holds iff the extension of C is a subset of the extension of D. Hence, D is implied whenever C holds. From an application point of view, the utility of general TBoxes for DL knowledge bases has long been observed. If $C \sqsubseteq D$ and $D \sqsubseteq C$, we say that D is *equivalent* to C ($C = D$). The TBox $\mathcal{T}$ is called acyclic if it contains only equivalence statements where the left-hand side is not used in the concept description on the right-hand side implicitly or explicitly and the TBox is called cyclic if it contains equivalence statements where the left-hand side is used in the concept description on the right-hand side implicitly or explicitly.

Our proposed approach is dedicated to the second type of Tboxes, namely the ones containing general concept inclusion (GCI) axioms. We give hereafter some useful definitions for the rest of the paper.

Definition 3. *The concept description D subsumes the concept description C w.r.t the TBox $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$. We write $C \sqsubseteq D$ iff C is subsumed by D. Two concept descriptions C, D are called equivalent w.r.t. $\mathcal{T}$ iff they subsume each other, i.e., $C \equiv_{\mathcal{T}} D$ iff $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$*

Definition 4. *Given a collection $C_1, ..., Cn$ of $\mathcal{L}$ concept description, the least common subsume (lcs) of $C_1, ..., C_n$ in $\mathcal{L}$ is the most specific $\mathcal{L}$-concept description that subsumes $C_1, ..., C_n$, i.e, it is an $\mathcal{L}$-concept description D such that 1) $C_i \sqsubseteq D$ for $i = 1, ..., n$ (D is a common subsumer) 2) If E is an $\mathcal{L}$-concept description satisfying $C_i \sqsubseteq E$ for $i = 1, ..., n$, then $D \sqsubseteq E$ (D is least)*

4 Cross-Fertilizing FCA and DL: A Survey

There are two main tendencies for cross-fertilizing FCA and DL between themselves. The first category aims to enrich the language of FCA by borrowing constructors from DL languages [13] whereas, the second category aims to employ FCA methods for solving problems encountered in knowledge representation with DLs [1, 2, 3, 4, 5, 6, 10]. Since this paper, is concerned by the second category, we survey in the following related existing approaches.

In [1], Baader et al. restrict themselves to the case where objects have only a partial description. In the sense that, for some attributes, it is not known whether they are satisfied by the object or not. This extension is necessary in order to deal with the open-world semantics of the description logic knowledge bases, and explore them using attribute exploration.

Baader [2] has used FCA for an efficient computation of an extended subsumption hierarchy of a set of DL concepts. More precisely, he used attribute exploration for computing the subsumption hierarchy of all conjunctions of a set of DL concepts. The main motivation for this work was to determine the interaction between defined concepts, which might not easily be seen by just looking at the subsumption hierarchy of defined concepts.

In [3], Baader and Sertkaya are interested in computing the subsumption hierarchy of all least common subsumers (lcs) of subsets of set of description logic concepts (S), without having to compute the least common subsumer for all subsets of S, using methods from formal concept analysis.

There are very few works which are concerned by learning (inducing) the TBox from object descriptions. In order to obtain complete knowledge about the subsumption relationships in the given model between arbitrary $\mathcal{FLE}$ concepts, Rudolph [6] gives a multi-step exploration algorithm. For each step, he generates implication base from a formal context by applying the attribute exploration method and generates the attribute set for the next exploration step. Rudolph points out that, at an exploration step, there can be some concept descriptions in the attribute set that are equivalent, i.e., attributes that can be reduced. To this aim, he introduces a method that he calls empiric attribute reduction. In principle, it is possible to carry out infinitely many exploration steps, which means that the algorithm will not terminate. In order to guarantee termination, the same author restricts the number of exploration steps.

Another approach which consists of completing the TBox with terminological axioms learned from assertions contained in an ABox is proposed in [10]. This approach translates data from DL formalism, that is instances of the ABox, to a form compliant with FCA (i.e. lattices). More precisely, authors adapt classical FCA algorithms in order to build sets of concept definitions from object descriptions. However, these approaches deal exclusively with the conjunctive form of concept descriptions. For this purpose, the proposition described in the next section aims to learn GCIs containing conjunction of concepts as well as conjunction of roles existentially restricted.

5 Proposition

5.1 Theoretical Aspects

Let us consider the following notations and abbreviations. Given an object $O_i \in N_O$, we consider a mapping τ which associates to each object O_i its corresponding $\mathcal{EL}$-concept description C_i ($C_i = \tau(O_i)$). Let also $concepts(C_i)$ denotes the set of all concept names occurring in C_i, $roles(C_i)$ denotes the set of all role names occurring in an existential restriction of C_i, and $restrict_r(C_i)$ denotes the concept description occurring in an existential restriction on the role r of C_i. For a nonempty subset $\{a_1, \ldots, a_k\}$ of concept names, $\bigsqcap A$ denotes the conjunction $a_1 \sqcap \ldots \sqcap a_k$ of concepts. For a nonempty subset $\{r_1, \ldots, r_s\}$ of role names, we denote the conjunction $r_1 \sqcap \ldots \sqcap r_s$ by $\bigsqcap R$. An object description based on concepts and roles may now be abbreviated as follows:

$$C_i = \tau(O_i) = \bigsqcap_{A \in concepts(C_i)} A \sqcap \bigsqcap_{r \in roles(C_i)} \bigsqcap_{E \in restrict_r(C_i)} \exists r.(E)$$

Example 1. Let C :Man⊓ Father⊓hasChild(Man⊓ Father) be the description of the object Bob (i.e. $C = \tau$(Bob)). Thus, $concepts(C)$=$\{Man, Father\}$, $roles(C)$=$\{hasChild\}$, and $restrict_{hasChild}(C)$ =$\{Man \sqcap Father\}$.

The obvious analogy between an object description C_i related to DL paradigm and a tuple (row) of a formal context related to FCA theory leads us to generate the context formal $K_C = (N_O, N_C \cup N_R, \mathcal{I})$ where the Cartesian product $\mathcal{I}$ (*i.e.* $\mathcal{I} \subseteq N_O \times (N_C \cup N_R)$) is now obtained using the following algorithm. The formal context $K_C = (N_O, N_C \cup N_R, \mathcal{I})$ is defined from in objects descriptions, where each object O_i is described by $\mathcal{EL}$-concept description $C_i = \tau(O_i)$. We propose to determine all entries of $\mathcal{I}$ using the following algorithm:

Algorithm. Gen_Formal_Context

Require: N_O, N_C, N_R
Begin
$\mathcal{I} := \{\emptyset\}$;
For each $o \in N_O$ **Do**
  For each $A \in concept(\tau(o))$ **Do**
    $\mathcal{I} := \{(o, A)\} \cup \mathcal{I}$;
  End For;
  For each $r \in roles(\tau(o))$ **Do**
    $\mathcal{I} := \{(o, r)\} \cup \mathcal{I}$;
  End For;
End For
End

According to the method proposed in [7], we may define the Duquenne-Guigues base or stem base DG_{K_C} of all attribute implications of the formal context K_C as follows. Let us first recall that such a base is not only irredundant, but also has minimal cardinality among all other bases induced from K_C.

Definition 5. *The Duquenne-Guigues base DG_{K_C} of all implications is given as:*
DG_{K_C} $=\{$ $\{A_1, ..., A_m, r_1, ..., r_n$ $\}\rightarrow\{A_{m+1}, ..., A_p, r_{n+1}, ..., r_k\}$ $|$ $\{A_1, ..., A_m, r_1, ..., r_n\}$ *is a Pseudo-Intent* $\wedge$ $\{A_{m+1}, ..., A_p, r_{n+1}, ..., r_k\}$ $=\{A_1, ..., A_m, r_1, ..., r_n\}''\}$

Where $A_1, ..., A_m, A_{m+1}, ..., A_p \in N_C$ and $r_1, ..., r_n, r_{n+1}, ..., r_k \in N_R$. $\{\{A_1, ..., A_m, r_1, ..., r_n$ $\}\rightarrow\{A_{m+1}, ..., A_p, r_{n+1}, ..., r_k\}$ is the general form of each implications $\in DG_{K_C}$ but, two subsets ($\phi(DG_{K_C})$, $\psi(DG_{K_C})$) can be distinguished as formalized in the two following definitions.

Definition 6. *We call* $\phi(DG_{K_C}) =\{ (\{m_1, ..., m_k\} \rightarrow \{m_{k+1}, ..., m_p\}) \in DG_{K_C} \mid m \in \{m_1, ..., m_k, m_{k+1}, ..., m_p\} \Rightarrow m \in N_C\}$ *the set of implications of DG_{K_C} without the implications which have the role names r belonging to the premise or the conclusion, i.e., the implications of $\phi(DG_{K_C})$ are of the form* $\{A_1, ..., A_m\} \rightarrow \{A_1, ..., A_m\}$.

Definition 7. *We call* $\psi(DG_{K_C}) =\{ (\{m_1, ..., m_k\} \rightarrow \{m_{k+1}, ..., m_p\}) \in DG_{K_C} \mid \exists m \in \{m_1, ..., m_k, m_{k+1}, ..., m_p\} \wedge m \in N_R\}$ *the set of implications of DG_{K_C} with the implications which have at least one role names r belonging to the premise or the conclusion.*

Unlike existing approaches which are restricted to attribute implications *imp* without roles (i.e. $imp \in \phi(DG_{K_C})$), our proposed approach allows to induce attribute implications *imp* containing roles (i.e. $imp \in \phi(DG_{K_C}) \cup \psi(DG_{K_C})$). For this purpose, we need to have the type of each role. Given $imp_i \in \psi(DG_{K_C})$, let $\psi(DG^*_{K_C})$ denotes the set of all implications of $\psi(DG_{K_C})$ s.t. a type is assigned for each role of each implication (example: if $\exists r.C$ is a concept description with existential restriction, then r is a role name and C is a type of r). It comes that an implication of $\psi(DG^*_{K_C})$ is of the form:

$Imp_i =\{\{A_1, ..., A_m, \exists r_1.(restrict^*_{r_1}(Imp_i)), ..., \exists r_n.(restrict^*_{r_n}(Imp_i))\} \longrightarrow \{A_{m+1}, ..., A_p,$ $\exists r_{n+1}.(restrict^*_{r_{n+1}}(Imp_i)), ..., \exists r_k.(restrict^*_{r_k}(Imp_i))\}$.

Where $restrict^*_r(imp_i)$ denotes the type of the role r $r \in \{m_1, ..., m_k, m_{k+1}, ..., m_p\}$). The two following propositions establish the theoretical framework which allows to obtain the set $\psi(DG^*_{K_C})$.

Proposition 1. *Assume that* $Imp_i = \{\{m_1, ..., m_k\} \rightarrow \{m_{k+1}, ..., m_p\}\}$ *is an implication of* $\psi(DG_{K_C})$*, that r is the role names of* $\{m_1, ..., m_k, m_{k+1}, ..., m_p\}$*, and that* $\{O_1, ..., O_s\}$ *is the set of objects corresponding to* $\{m_1, ..., m_k, m_{k+1}, ..., m_p\}'$*, then for each r:*
$restrict^*_r(Imp_i) = lcs(\bigsqcap resctict_r(\tau(O_1)), ..., \bigsqcap resctict_r(\tau(O_l)))$

Proof. Due to lack of space, proof is left for a long version of this paper.

In [11] it is shown that the lcs of two or more $\mathcal{EL}$-concept description always exists and it can be computed in polynomial time.

Proposition 2. $B_1 \rightarrow B_2$ *is an implication holds in* $\psi(DG^*_{K_C})$*, then* $\sqcap B_1 \sqsubseteq \sqcap B_2$.

Proof. Assume that the subsumption relationship $\sqcap B_1 \sqsubseteq \sqcap B_2$ does not hold, this is the case iff there exists an objet $g \in (\sqcap B_1)^{\mathcal{I}}$ and $g \notin (\sqcap B_2)^{\mathcal{I}}$, this is equivalent to $g \in m^{\mathcal{I}}$ for all $m \in B_1$,and there exists $p \in B_2$ such that $g \notin p^{\mathcal{I}}$ (using the semantics of the conjunction operator). By definition, $g \in B_1'$, and $g \notin B_2'$ then $B_1' \not\subseteq B_2'$, this shows that the implication $B_1 \rightarrow B_2$ does not hold. Obeviously, all of the conclusions we have made are reversible.

5.2 Learning Algorithm

The proposed algorithm generates *GCIs* (General Concept Inclusion) in order to obtain a TBox with a minimal number of *GCIs* from which we can find every other that hold through inference.It takes as input $\psi(DG_{K_C})$ and $\phi(DG_{K_C})$, It takes also K_C. The algorithm "*Gen_GCIs*" is given as follows:

Algorithm. Gen_GCIs

Require: $\psi(DG_{K_C})$,$\phi(DG_{K_C})$, K_C
Begin
$\psi(DG^*_{K_C}) := \{\emptyset\}$;
For each $Imp = \{\{m_1, ..., m_k\} \rightarrow \{m_{k+1}, ..., m_p\}\} \in \psi(DG_{K_C})$ **Do**
  $Imp^* := Imp$;
  $\{O_1, ..., O_s\}$ is the set of objects, corresponding to $\{m_1, ..., m_k, m_{k+1}, ..., m_p\}'$;
  For each $r \in \{m_1, ..., m_k, m_{k+1}, ..., m_p\}$ **Do**
    $restrict^*_r(Imp) := lcs(\bigsqcap resctict_r(\tau(O_1)), ..., \bigsqcap resctict_r(\tau(O_s)))$;
    replace r by $\exists r.restrict^*_r(Imp)$ in Imp^*;
  End For;
  $\psi(DG^*_{K_C}) := \psi(DG^*_{K_C}) \cup \{Imp^*\}$;
End For
For each $Imp^* = (A \rightarrow B) \in (\psi(DG^*_{K_C}) \cup \phi(DG_{K_C}))$ **Do**
  Add the GCI $(\sqcap A \sqsubseteq \sqcap B)$ to the TBox ;
End For
End

---

6 Illustrative Example

Let us illustrate our proposition through the following example:

N_C ={Man, Woman, Father, Mother, Parent, Aunt, Uncle}, N_r ={hasBrother, hasSister} and N_O ={Hocine, Amine, Idir, Lili, Nora, Ali, Aziz, Sonia, Lilia, Lyes, Samir, Ania, Lina}.

We consider the following set of objects described by concept descriptions:

Hocine: Man ⊓ Father ⊓ Parent
Amine: Man
Idir: Man ⊓ Father ⊓ Parent
Lila: Woman ⊓ Mother ⊓ Parent
Nora: Woman
Ali: Man ⊓ ∃ hasSister.(Aunt ⊓ Woman ⊓ Mother ⊓ Parent) ⊓ Uncle
Aziz: Man ⊓ ∃ hasSister.(Man ⊓ Father ⊓ Parent) ⊓ Uncle
Sonia: Woman ⊓ ∃ hasSister.(Mother ⊓ Parent ⊓ Woman) ⊓ Aunt
Lilia: Woman ⊓ ∃ hasBrother.(Man ⊓ Father ⊓ Parent) ⊓ Aunt
Lyes: Man ⊓ ∃ hasSister.(Woman ⊓ Mother ⊓ Parent) ⊓ Uncle
Samir: Man ⊓ ∃ hasBrother.(Uncle ⊓ Father ⊓ Mother ⊓ Parent) ⊓ Uncle
Ania: Woman ⊓ ∃ hasSiste.(Mother ⊓ Aunt ⊓ Parent ⊓ Woman) ⊓ Aunt
Lina: Woman ⊓ ∃ hasBrother.(Man ⊓ Parent ⊓ Uncle) ⊓ Aunt

Initially, we generate the context formal $K_C = (N_O, N_C \cup N_R, \mathcal{I})$ using the algorithm ”*Gen_Formal_Context*”:

N_O:={ Hocine, Amine, Idir, Lila, Nora, Ali, Aziz, Sonia, Lilia, Lyes, Samir, Ania, Lina }

$N_C \cup N_R$:={Man, Woman, Father, Mother, Parent, Aunt, Uncle, hasSister, hasBrother}

$\mathcal{I}$:={(Hocine, Man), (Hocine, Father), (Hocine, Parent), (Amine, Man), (Idir, Man), (Idir, Father), (Idir, Parent), (Lila, Woman), (Lila, Mother), (Lila, Parent), (Nora, Woman), (Ali, Man), (Ali, hasSister), (Ali,Uncle), (Aziz, Man), (Aziz, hasBrother), (Aziz, Uncle), (Sonia, Woman), (Sonia, hasSister), (Sonia, Aunt), (Lilia, Woman), (Lilia, hasBrother), (Lilia, Aunt), (Lyes, Man), (Lyes, hasSister), (Lyes, Uncle), (Samir, Man), (Samir, hasBrother), (Samir, Uncle), (Ania, Woman),(Ania, hasSister),(Ania, Aunt), (Lina, Woman), (Lina, hasBrother), (Lina, Aunt)}

Implications of Duquenne-Guigues resulting from the context formal $K_C = (N_O, N_C \cup N_R, \mathcal{I})$ are as follow:

$\phi(DG_{K_C})$ ={{Father} → {Man, Parent}, {Mother} → {Woman, Parent}, {Man, Parent} → {Father}, {Woman, Parent} → {Mother}, {Aunt} → {Woman}, {Uncle} → {Man}}

$\psi(DG_{K_C})$ ={{Woman, hasSister} → {Aunt}, {Woman, hasBrother} → {Aunt}, {Man, hasSister} → {Uncle}, {Man, hasBrother} → {Uncle}} Now, we compute $restrict_r^*(Imp_i)$ for each role of each implication ∈ $\psi(DG_{K_C})$ using the algorithm ”*Gen_GCIs*”

- Imp_1: {Woman, hasSister} → {Aunt}
{Sonia, Ania} = {Woman, hasSister, Aunt}′ (to use the formal context K_C)

$$resctict^*_{hasSister}(imp_1) = lcs(\bigsqcap resctict_{hasSister}(\tau(Sonia)), \bigsqcap resctict_{hasSister}(\tau(Ania)))$$

$$resctict^*_{hasSister}(imp_1) = lcs((Mother \sqcap Parent \sqcap Woman), (Mother \sqcap Aunt \sqcap Parent \sqcap Woman))$$

$$resctict^*_{hasSister}(imp_1) = (Parent)$$

the implication {woman, ∃ hasSister.(Parent)} → {Aunt} is added to $\psi(DG^*_{K_C})$

All other implications $\in \psi(DG_{K_C})$ are computed of same manner that we have computed $resctict^*_{hasSister}(imp_1)$ then obtain:

$\psi(DG^*_{K_C})$ ={ {woman, ∃ hasSister.(Parent)} → {Aunt}, {woman, ∃ hasBrother.(Parent)} → {Aunt}, {man, ∃ hasSister.(Parent)} → {Uncle},{man, ∃ hasBrother.(Parent)} → {Uncle}} At the end of the algorithm "*Gen_GCIs*", the following GCI axioms have been found: father ⊑ man ⊓ parent
mother ⊑ woman ⊓ parent
man ⊓ parent ⊑ father
woman ⊓ parent ⊑ mother
Aunt ⊑ woman
Uncle ⊑ man
woman ⊓ ∃ hasSister.(Parent) ⊑ Aunt
woman ⊓ ∃ hasBrother.(Parent) ⊑ Aunt
man ⊓ ∃ hasSister.(Parent) ⊑ Uncle
man ⊓ ∃ hasBrother.(Parent) ⊑ Uncle

7 Conclusion

Cross fertilizing both description logics and formal concept analysis seems an appealing domain of research. There are many tendencies in this direction. Learning concept definitions from object descriptions is mainly addressed in this spirit. However, in all existing approaches such concept definitions are restricted to concepts names. In this paper, we enlarge the learning process and propose an approach which allows to learn General Concept Inclusion (GCI) containing concept names as well as existentially quantified roles. For this purpose, we have established the appropriate propositions. As direct future work, we intend first to generalize our proposition to more expressive DL families. We intend also to address incomplete (missing, uncertain, imprecise) objects descriptions.

Acknowledgments. This work was supported by the National Research Program (PNR) of the Directorate General for Scientific Research and Technological Development (DGRSDT) - Ministry of Higher Education and Scientific Research under the grant 14/IF/2011.

References

[1] Baader, F., Ganter, F., Sattler, U., Sertkaya, B.: Completing description logic knowledge bases using formal concept analysis. LTCS-Report LTCS-06-02, Chair for Automata Theory, Inst. for Theoretical Computer Science, TU Dresden, Germany, (2006)
[2] Baader, F.: Computing a Minimal Representation of the Subsumption Lattice of all conjunctions of Concepts Defined in a Terminology. In: Ellis, G., Levinson, R.A., Fall, A., Dahl, V. (eds.) Proceedings of the 1st International KRUSE Symposium on Knowledge Retrieval, Use and Storage for Efficiency, pp. 168–178 (1995)
[3] Baader, F., Sertkaya, B.: Applying formal concept analysis to description logics. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 261–286. Springer, Heidelberg (2004)
[4] Stumme, G.: The Concept Classification of a Terminology Extended by Conjunction and Disjunction. In: Foo, N.Y., Göbel, R. (eds.) PRICAI 1996. LNCS, vol. 1114, pp. 121–131. Springer, Heidelberg (1996)
[5] Baader, F., Molitor, R.: Building and Structuring Description Logic Knowledge Bases Using Least Common Subsumers Concept Analysis. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS, vol. 1867, pp. 290–303. Springer, Heidelberg (2000)
[6] Rudolph, S.: Relational exploration: Combining Description Logics and Formal Concept Analysis for knowledge specification. Ph.D. dissertation, Fakultat Mathematik und Naturwissenschaften, TU Dresden, Germany (2006)
[7] Guigues, J.L., Duquenne, V.: Familles minimales d'implications informatives resultant d'un tableau de donnees binaires. Math. Sci. Humaines 95, 5–18 (1986)
[8] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Application. Cambridge University Press (2003)
[9] Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999)
[10] Bazin, A., Ganacia, J.: Completing terminological Axioms with Formal Concept Analysis. In: ICFCA 2012, International Conference on Formal Concept Analysis, pp. 30–39 (2012)
[11] Baader, F., Küsters, R., Molitor, R.: Computing least common subsumers in description logics with existential restrictions. In: Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI 1999), pp. 96–101 (1999)
[12] 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)
[13] Prediger, S.: Terminologische Merkmalslogik in der Formalen Begriffsanalyse. In: Stumme, G., Wille, R. (eds.) Begriffliche Wissensverarbeitung – Methoden und Anwendungen, pp. 99–124. Springer, Heidelberg (2000)
[14] Baader, F., Horrocks, I., Sattler, U.: Description logics as ontology languages for the semantic web. In: Hutter, D., Stephan, W. (eds.) Mechanizing Mathematical Reasoning. LNCS (LNAI), vol. 2605, pp. 228–248. Springer, Heidelberg (2005)