Completing Terminological Axioms with Formal Concept Analysis

Alexandre Bazin and Jean-Gabriel Ganascia

Université Pierre et Marie Curie, Laboratoire d'Informatique de Paris 6
Paris, France
Alexandre.Bazin@lip6.fr
Jean-Gabriel@Ganascia.name

Abstract. Description logics are a family of logic-based formalisms used to represent knowledge and reason on it. That knowledge, under the form of concepts and relationships between them called terminological axioms, is usually manually entered and used to describe objects in a given domain. That operation being tiresome, we would like to automatically learn those relationships from the set of instances using datamining techniques. In this paper, we study association rules mining in the description logic EL. First, we characterize the set of all possible concepts in a given EL language. Second, we use those characteristics to develop an algorithm using formal concept analysis to mine the rules more efficiently.

Keywords: Description Logic, Association Rules Mining, Ontology

1 Introduction

Ontologies are knowledge representation tools used in various domains of application. The semantic web, for example, makes an extensive use of them. They are essentially composed of a list of concepts relevant to a particular domain and relations (mainly inclusion and equivalence, i.e. hierarchical relations) existing between them. Description Logics (DL) are increasingly popular logical frameworks used to represent ontologies and on which is based the OWL[1] language for the semantic Web. They have a great representation power and allow powerful reasoning tools. However, the construction of ontologies, usually performed manually by knowledge engineers, is both a tedious and tricky operation. One of the difficulties is to ensure the consistency and the completeness of the set of relations between concepts. in order to facilitate this step, we propose to automatize, at least partially, the process of relation generation.

Based on the lattice theory, Formal Concept Analysis (FCA) is a mathematical framework that also deals with concepts and their hierarchical relationships. FCA provides solid theoretical foundations for association rule learning tools.

[1] OWL is an acronym for *Ontology Web Language*, which is a W3C standard

It therefore seems to be a good natural candidate for this task, i.e. for the automatic generation of relationships between concepts, from object descriptions, i.e. from concept instances.

Despite differences between the use of the notion of concept in these two formalisms, it would be interesting to combine them both and draw benefits from their mutual advantages. This combination has already been investigated and two main approaches exist. The first integrates operators of FCA to the DL framework in order to be able to apply learning algorithms directly to a knowledge base expressed in DL [4] [8], the second, which corresponds to our present work, translates data from DL to a form comprehensible by FCA, in other words, it interprets DL formalism within the lattice theory [2] [3] [7].

We claim that, by using the specific lattice structure of the set of concepts of description logics, we will be able to modify classical FCA algorithms in order to build complete and consistent sets of terminological axioms from object descriptions given as assertions. This work, which constitutes a first attempt in this direction, will make use of a simple description logic, which is $\mathcal{EL}$. But, the approach is not restricted to $\mathcal{EL}$; it will certainly be possible to generalize it to other DL, which will be investigated in further work.

Apart from the introduction and the conclusion, this paper is divided into four parts. The first briefly recalls the usual definitions in both Description Logics and Formal Concept Analysis, the second characterizes the structure of the set of $\mathcal{EL}$-concepts making use of the function Φ that is the set of subsets of incomparable elements of a language, the third describes a simple association rule learning algorithm that works within the set of $\mathcal{EL}$-concepts previously described. It then studies the properties of the set of terminological axioms that it generates. The last part is dedicated to a brief example, which illustrates the different notions presented in this paper.

2 Definitions and Recalls

2.1 Description Logics

Descriptions logics are decidable fragments of first-order logic used to represent and reason on knowledge. Syntactically, every description logic language makes use of a set of concept names N_C, a set of role names N_R and a set of object names N_O and combines them using constructors to build concept descriptions or, in short, concepts. The set of constructors used defines the language's expression power and the complexity of its reasoning procedures. In this paper, we will consider the logic $\mathcal{EL}$. In it, every concept name is a concept description and, for any concept descriptions A and B and any role r, $A \sqcap B$ and $\exists r.A$ are also concept descriptions. Having only two constructors, this logic is one of the simplest.

Semantics are defined by means of interpretations. An interpretation is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a set of objects called the domain and $.^{\mathcal{I}}$ a function mapping every concept name C to a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and every role name r

to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. As such, concepts are defined by the set of objects which belong to them.

An important notion in description logic systems is the *subsumption* relation between concept descriptions. Given two concept descriptions C and D, we say that D *subsumes* C ($C \sqsubseteq D$) if the set of objects belonging to C is included in the set of objects belonging to D ($C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$) for all interpretations $\mathcal{I}$. For a given TBox $\mathcal{T}$, we say that D *subsumes* C *with respect to* $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model of $\mathcal{T}$. If $C \sqsubseteq D$ and $D \sqsubseteq C$, it gives the definition $C \equiv D$. Constructions such as $C \sqsubseteq D$ and $C \equiv D$ expressing subsumption relations are called *terminological axioms*.

For any given concept C, role r and object names o and o', $o : C$ and $(o, o') : r$ are called *assertional sentences*. The constructions $o : C$ means that the object o belongs to the concept C and $(o, o') : r$ means that the object o' fulfills the role r for the object o.

A *knowledge base* consists of a *TBox* and an *ABox*. The TBox is constituted of terminological axioms, which we try to learn in this paper, and concept definitions. The ABox is a set of assertional sentences and can be viewed as a set of descriptions of objects.

2.2 Formal Concept Analysis

In formal concept analysis (FCA), we call *formal context* a triplet $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ where $\mathcal{O}$ is a set of objects, $\mathcal{A}$ a set of attributes and $\mathcal{R}$ a binary relation between objects and attributes. We say here that $(o, a) \in \mathcal{R}$ means that a describes o.

We have at our disposal two functions $.'$ such as

$$.' : 2^{\mathcal{A}} \mapsto 2^{\mathcal{O}}$$

$$A' = \bigcap_{a \in A} \{o \in \mathcal{O} \mid (o, a) \in \mathcal{R}\} \tag{1}$$

and

$$.' : 2^{\mathcal{O}} \mapsto 2^{\mathcal{A}}$$

$$O' = \bigcap_{o \in O} \{a \in \mathcal{A} \mid (o, a) \in \mathcal{R}\} \tag{2}$$

A' is then the set of objects described by every attribute of A and O' is the set of attributes describing every object of O. If $A \subseteq B$, then $B' \subseteq A'$ and if $O \subseteq P$ then $P' \subseteq O'$. As such, those two functions form a *Galois Connection*.

A *formal concept* is defined as a pair $(E, I) \in A^{\mathcal{O}} \times 2^{\mathcal{A}}$ where $E = I'$ and $I = E'$. We say that E and I are closed. E and I are respectively called the *extent* and the *intent* of the concept. In order to prevent confusion, formal

concept will not be abbreviated and the term concept will be used exclusively for DL-concepts.

We call $FC(\mathcal{O}, \mathcal{A}, \mathcal{R})$ the set of formal concepts we can find in $(\mathcal{O}, \mathcal{A}, \mathcal{R})$. We can define an order $<$ (a relation "is more general than") on this set such as $(E, I) < (F, J) \Leftrightarrow (F \subset E \text{ and } I \subset J)$ and the pair $(FC(\mathcal{O}, \mathcal{A}, \mathcal{R}), <)$ satisfies the properties of a complete lattice. Such a lattice is called a concept (or Galois) lattice. For example, for a formal context in which $\mathcal{O} = \{a, b, c, d, e\}$, $\mathcal{A} = \{1, 2, 3, 4, 5\}$ and $\mathcal{R} = \{(a, 2), (a, 4), (b, 3), (b, 5), (c, 1), (c, 2), (c, 4), (d, 2), (d, 3), (e, 2), (e, 4)\}$ we obtain the following concept lattice.

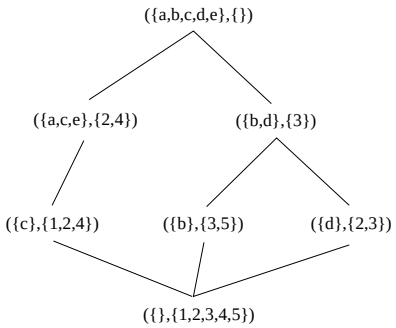


Fig. 1. A Concept Lattice

FCA allows us to find implications in the formal context which are ordered pairs (B, C), often written $B \to C$. An implication $B \to C$ holds in a context if every object described by every attribute in B is also described by every attribute in C.

Definition 1. *We say that a set $X \in A$ respects an implication $B \to C$ if $B \subseteq X$ implies $C \subseteq X$.*

An implication $B \to C$ follows from a set of implications $\mathcal{L}$ if every $X \in A$ that respects every implication in $\mathcal{L}$ respects $B \to C$. A set $\mathcal{L}$ of implications is then called a basis if every implication in $\mathcal{L}$ holds in the context and every implication that holds in the context follows from $\mathcal{L}$.

It is a known fact that $\{X \to X'' \mid X \subseteq A\}$ is an implicational basis which means that, in order to obtain a basis of minimal cardinality, we need only to find implications whose right-hand side are concept intents. Finding suitable left-hand side has thus been the subject of many works.

Definition 2. *A set $X \in A$ is a pseudo-intent of the context (O, A, R) if X is not a concept intent and, for all pseudo-intent $Y \subset X$, $Y'' \subseteq X$.*

Definition 3. *The set of implications* $\{X \rightarrow X'' \mid X$ *is a pseudo-intent*$\}$ *is called a Duquenne-Guigues Basis.*

The Duquenne-Guigues Basis is the minimal set of implication from which we can find every other implications that hold through inference.

3 The Set of Concept Descriptions

Before using an association rules learning algorithm, we will study the structure of the set of concepts one can build with the description logic $\mathcal{EL}$.

We will use Ω to denote the set of terminological axioms $A \sqsubseteq B$ in an acyclic TBox $\mathcal{T}$. $N_C = A_C \cup D_C$ will denote the set of concept names used in $\mathcal{T}$ with A_C the set of atomic concepts and D_C the set of defined concepts, appearing in the left hand side of definitions. The set of pairs $(C1, C2)$ such as $C1 \equiv C2$ will be called $Def(\mathcal{T})$. Ω induces a partial order on the set of equivalence classes of concepts, noted $N_{C_\equiv}$, used in axioms (if $(x \sqcap y \sqsubseteq z) \in \Omega$, we will consider there is some d in D_C such as $d \equiv x \sqcap y$). We will simply use $b \leq a$ for $[a]_\equiv \sqsubseteq [b]_\equiv$. $(N_{C_\equiv}, \leq)$ is then a partially ordered set such as, for all x in N_C, $[\top]_\equiv \leq [x]_\equiv$. For clarity purposes, we will now use CN^0 to denote a set of concept names containing a unique representative of each equivalence class together with the order $\leq$. Obviously, CN^0 is isomorphic to $(N_{C_\equiv}, \leq)$.

We are interested in the set of every possible concept we can construct with N_C, N_R and the constructors $\sqcap$ and $\exists$. Suppose there are two concepts A and B such as $A \sqsubseteq B$. This means that $A^I \subseteq B^I$ so $A \sqcap B \equiv A$. Those two concept descriptions being equivalent we consider they are the same and we do not want to include both of them in the set of possible concepts. As such, we want the set of concepts resulting from the conjunction of incomparable elements.

Definition 4. *We call* $\Phi(CN^0) = \{X \subseteq CN^0 \mid x \in X \wedge y \in X \Rightarrow x||y\}$ *the set of subsets of incomparable elements of* CN^0

We call $\Phi(CN^0)$ the set of subsets of incomparable elements of CN^0 and $\sqcap A$ the concept built from the conjunction of the elements of A. For any two elements $C, D \in \Phi(CN^0)$, we say that $C \leq D$ if and only if $\sqcap D \sqsubseteq \sqcap C$. That is, $C \leq D$ if and only if for every element $c \in C$ there is some $d \in D$ such as $c \leq d$. Evidently, $\Phi(CN^0)$ is isomorphic to the set of ideals of CN^0 ordered by inclusion and its elements are the sets of maximal elements of those ideals. $\Phi(CN^0)$ is then a distributive lattice.

Proposition 1. *For any two elements* $A, B \in \Phi(CN^0)$, $A \wedge B = Max(\{x \in CN^0 \mid \exists a \in A,\ \exists b \in B, x \leq a\ \&\ x \leq b\})$ *and* $A \vee B = Max(A \cup B)$.

$\sqcap(A \wedge B)$ corresponds to the least common subsumer of $\sqcap A$ and $\sqcap B$ and $\sqcap(A \vee B)$ to the most specific concept subsumed by $\sqcap A$ and $\sqcap B$. They can be easily computed from CN^0.

$\Phi(CN^0)$ being finite and distributive, for all A and B in $\Phi(CN^0)$, there is a least element X such as $A \vee X \geq B$ called difference and noted $B \setminus A$. It is equal

to $Max(\downarrow B \setminus \downarrow A)$ where $\downarrow A$ is the set of elements lower or equal to elements of A in CN^0.

Proposition 2. *For a given linear extension σ of CN^0, the relation $A \rightsquigarrow B \Leftrightarrow B \setminus A = Max_\sigma(B)$ defines a spanning tree of the covering graph of $\Phi(CN^0)$.*

The spanning tree gives us, for every element $A \in \Phi(CN^0)$, a unique path from $\{\top\}$ to A in which $A \rightsquigarrow B \Rightarrow A \leq B$.

$\Phi(CN^0)$ is the set of different conjunctions of concept names based on the subsumption relation. However, the TBox can also contain equivalences between elements of $\Phi(CN^0)$. If $(A, B \sqcap C) \in Def(\mathcal{T})$ then $B \leq A$ and $C \leq A$ in CN^0. In $\Phi(CN^0)$, $\{A\}$ is thus strictly greater than $\{B, C\}$. Those two concepts being equivalent, every element greater or equal to $\{B, C\}$ and lower than $\{A\}$ is considered redundant.

Definition 5. *$\Phi(CN^0)_{Def(\mathcal{T})} = \Phi(CN^0) \setminus \{B \mid (A, B) \in Def(\mathcal{T})\}$ is the set of subsets of incomparable elements of CN^0 without the elements corresponding to right-hand sides of definitions of $\Phi(C, \leq_X)_{Def(\mathcal{T})}$ the TBox.*

Proposition 3. *For any two elements $A, B \in \Phi(CN^0)_{Def(\mathcal{T})}$, $A \wedge B$ in $\Phi(CN^0)_{Def(\mathcal{T})}$ is equal to $A \wedge B$ in $\Phi(CN^0)$.*

Proposition 4. *For all A and B in $\Phi(CN^0)_{Def(\mathcal{T})}$, the difference $A \setminus B$ in $\Phi(CN^0)$ is an element of $\Phi(CN^0)_{Def(\mathcal{T})}$.*

These operations on $\Phi(CN^0)_{Def(\mathcal{T})}$ are thus the same than on $\Phi(CN^0)$. The differences appear when we try to compute the upper cover of an element D, i.e. elements immediately greater than D. We call $Cand$ – for candidate – the set of minimal elements not lower than elements of D in CN^0. In $\Phi(CN^0)$, the upper cover of D is then $\{Max(D \cup c) \mid c \in Cand\}$. In $\Phi(CN^0)_{Def(\mathcal{T})}$, if there is some $(L, R) \in Def(\mathcal{T})$ such as $L \geq Max(D \cup c) \geq R$, c must be removed from the list of candidates and L added if it is minimal in $Cand \setminus c$. In order to find the elements following D in the spanning tree of $\Phi(CN^0)_{Def(\mathcal{T})}$ induced by some σ it would then be sufficient to remove the candidates c such as $c \leq_\sigma d$ for some d in D. The algorithm is as follows :

Algorithm 1

Require: CN^n, D
1: $Cand = \{c \in CN^n \mid c \in Min(CN^0 \setminus \downarrow D)$ and $\forall d \in D, c \geq_\sigma d\}$
2: **for** each $c \in Cand$ **do**
3: **if** $\exists (L, R) \in Def(\mathcal{T})$ such as $L \geq Max(D \cup c) \geq R$ **then**
4: $Cand = Min((Cand \setminus c) \cup L)$
5: **end if**
6: **end for**
7: Return $\{Max(D \cup c) \mid c \in Cand\}$

Now, $\Phi(CN^0)_{Def(\mathcal{T})}$ is only the lattice of concepts built from a conjunction of concept name without roles. However, it gives us informations on the structure of the set of role-concepts. We know that, for a given role r, $A \sqsubseteq B \Rightarrow \exists r.A \sqsubseteq \exists r.B$. The partially ordered set of roles of a depth 1 is then isomorphic to $\Phi(CN^0)_{Def(\mathcal{T})}$. We use CN^1_r to denote it. If $CN^1 = CN^0 \bigcup_{i=1}^{|N_R|} CN^1_{r_i}$ is the set of both concept names and roles of depth 1 together with the partial order induced by Ω, then $\Phi(CN^1)_{Def(\mathcal{T})}$ is the lattice of concepts containing roles up to a depth 1. Recursively, $\Phi(CN^n)_{Def(\mathcal{T})}$ where $CN^n = CN^0 \bigcup_{i=1}^{|N_R|} CN^n_{r_i}$ with $CN^n_{r_i}$ isomorphic to $\Phi(CN^{n-1})_{Def(\mathcal{T})}$ is the set of every possible concept descriptions up to an arbitrary role depth n.

4 Learning Axioms with Formal Concept Analysis

As we said previously, we take the approach of creating a formal context corresponding to the DL-objects we want to manipulate. More precisely, we use the formal context $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ where $\mathcal{O}$ is a set of objects, $\mathcal{A} = \Phi(CN^n)_{Def(\mathcal{T})}$ is the set of every possible concept descriptions defined in Section 2.2 and $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{A}$ is the relation associating objects to the most specific concept to which they belong. In that respect, it is very similar to contexts from Logical Concept Analysis [5] or the work of Baader [1] which also deals with finding implications in $\mathcal{EL}$.

We re-define the following operators :

$$.' : \mathcal{A} \mapsto 2^{\mathcal{O}}$$

$$X' \;=\; \{o \in \mathcal{O} \mid oRa \Rightarrow X \leq a\} \tag{3}$$

and

$$.' : 2^{\mathcal{O}} \mapsto \mathcal{A}$$

$$O' \;=\; \bigwedge \{a \in \mathcal{A} \mid o \in O \Rightarrow oRa\} \tag{4}$$

The first operator maps a concept description to the set of objects belonging to it while the second is the generalization of the most specific concepts describing the objects, which corresponds to the infimum in the lattice.

Now, if we want to get implications of the form $X \rightarrow X'' \setminus X$, we cannot use the set-theoretic difference directly. The difference $B \setminus A$ in the distributive lattice defined in the previous section corresponds to the most general concept whose conjunction with A would be more specific than B. It can also be seen intuitively as the part of B not covered by A. Thus, in the remainder of this work, we will use this definition of the difference.

By using the structure of $\Phi(CN^n)_{Def(\mathcal{T})}$, we can enumerate concept descriptions and get a set of implications by using the following algorithm :

Algorithm 2

Require: CN^n, σ
1: $Open = \{\{\top\}\}$
2: **for** every minimal element X of minimal role depth in $Open$ **do**
3: C = X''
4: **if** $C \neq X$ **then**
5: Update CN^n with $X \rightarrow C \setminus X$
6: Add elements following X in the spanning tree of $\Phi(CN^n)_{def(\mathcal{T})}$ to $Open$
7: **end if**
8: **end for**

Beginning with $\{\top\}$, the least element of $\Phi(CN^n)_{def(\mathcal{T})}$, we classically compute its closure. We then compute the closure of every element of $\Phi(CN^n)_{def(\mathcal{T})}$ immediately greater than $\{\top\}$ and so on. Of course, an element of the upper cover of D should not be considered if it contains an element that does not subsume any description of elements of D'. As soon as X'' is different from X a new implication is found and CN^n is updated, adding a new element to D_C if necessary, and X becomes a closed set of the new $\Phi(CN^n)_{def(\mathcal{T})}$.

For any minimal element X in $Open$, the elements of its lower cover are closed sets. As such, for any $Y \subset X$, $Y'' \subseteq X$. If $X \neq X''$, X is a pseudo-intent. Thus, by considering a minimal element of $Open$ at every step of the algorithm, we make sure we obtain the Duquenne-Guigues Basis of the original context. As a new implication $A \rightarrow B$ changes the structure of the lattice for role concepts we must select the minimal elements in ascending role-depth order.

As an element is added to N_C for every $A \sqcap B \sqsubseteq C$ found and $A \sqcap B$ becomes a closed element of the new $\Phi(CN^n)_{def(\mathcal{T})}$, the algorithm terminates with CN^n isomorphic to the concept lattice of the formal context minus the maximal formal concept.

The method we propose in this paper is similar to the one presented by Rudolph in [6]. However, we feel some important differences must be pointed out. First, our algorithm immediately considers all concepts up to the maximum role depth instead of using a different learning phase for each depth. Second, new implications are immediately included in the background knowledge. We believe this is especially important for axioms of the form $A \sqsubseteq B \sqcap \exists r.C$ where $A \sqsubseteq B$ would be found a first time before the step including roles.

5 Example

In our example, N_C = {Man, Woman, Father, Mother, Parent, GrandFather, GrandMother} and N_R = {hasChild}. Moreover, we know that

Mother$\sqsubseteq$ Woman $\sqcap$ Parent

We consider the following set of objects described by concept descriptions

Bob : Man ⊓ Father ⊓ Parent ⊓ ∃hasChild.Man
Bill : Man
Benjamin : Man ⊓ GrandFather ⊓ Father ⊓ Parent ∃hasChild.(Man ⊓ Father ⊓ Parent)
Bertrand : Man ⊓ GrandFather ⊓ Father ⊓ Parent ⊓ ∃hasChild.(Mother ⊓ Parent)
Bernard : Man ⊓ Father ⊓ Parent ⊓ ∃hasChild.Woman
Clara : Mother ⊓ ∃hasChild.Woman
Coralie : Mother ⊓ GrandMother ⊓ ∃hasChild.(Man ⊓ Father ⊓ Parent)
Claire : Mother ⊓ GrandMother ⊓ ∃hasChild.(Mother ⊓ Parent)
Chloe : Woman

Initially, $Open = \{\top\}$ and CN^n is as follows :

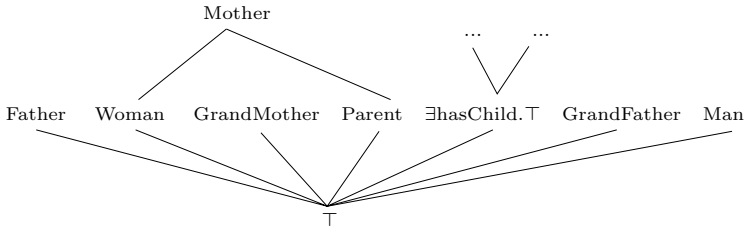


Fig. 2. CN^n at Step 0 (irrelevant role-concepts omitted)

$\top'' = \emptyset$ so there is no new implication.

$Open = \{$Woman, Father, GrandMother, Parent, ∃hasChild.⊤, GrandFather, Man$\}$

Woman$'' = \emptyset$ so there is no new implication.

$Open = \{$Father, GrandMother, Parent, ∃hasChild.⊤, GrandFather, Man, Mother, {Woman, Father}, {Woman, GrandMother}, {Woman, Parent}, {Woman, ∃hasChild.⊤}, {Woman, GrandFather}, {Woman, Man}$\}$

Father$'' = \{$Father, Man, Parent, ∃hasChild.⊤$\}$ so The implication Father $\rightarrow$ {Man, Parent, ∃hasChild.⊤} is added.

CN^n is then updated.

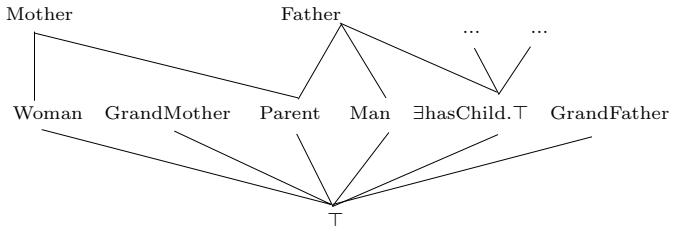


Fig. 3. CN^n at Step 3 (irrelevant role-concepts omitted)

Open = {GrandMother, Parent, ∃hasChild.⊤, GrandFather, Man, Mother, {Woman, Father}, {Woman, GrandMother}, {Woman, Parent}, {Woman, ∃hasChild.⊤}, {Woman, GrandFather}, {Woman, Man}, {Father, GrandMother}, {Father, GrandFather}, {Father, ∃hasChild.Parent}, {Father, ∃hasChild.Woman}, {Father, ∃hasChild.Man}}

Others implications are then found for GrandMother, Parent, ∃hasChild.⊤, {Woman, Parent}, GrandFather, {Father, ∃hasChild.Parent}, {Man, Parent}, {Mother, ∃hasChild.Parent}, {Father, ∃hasChild.Parent}. The algorithm terminates with CN^n in the following state.

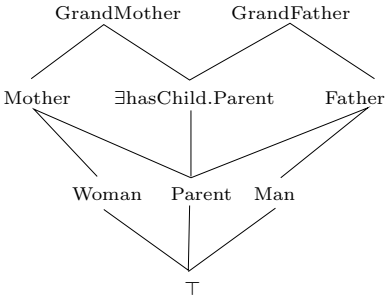


Fig. 4. CN^n at the end of the algorithm (irrelevant role-concepts omitted)

Note that ∃hasChild.⊤ does not appear in CN^n because it has been found equivalent to Parent.

The following terminological axioms have been found :

$$\text{Father} \equiv \text{Parent} \sqcap \text{Man}$$
$$\text{Mother} \equiv \text{Parent} \sqcap \text{Woman}$$
$$\text{GrandMother} \equiv \text{Mother} \sqcap \exists\text{hasChild.Parent}$$
$$\text{GrandFather} \equiv \text{Father} \sqcap \exists\text{hasChild.Parent}$$
$$\exists\text{hasChild.}\top \equiv \text{Parent}$$

6 Conclusion

As mentionned in the introduction, this research aims at completing the TBox with terminological axioms learned from assertions contained in an ABox. Our approach translates data from DL formalism, that is instances of the ABox, to a form homogeneous to FCA, i.e. to lattices. More precisely, by using the lattice structure of the set of concepts of description logics, we modify classical FCA algorithms in order to build complete and consistent sets of terminological axioms from object descriptions given as assertions.

In this paper, we have restricted our approach to $\mathcal{EL}$. We have shown that the set of $\mathcal{EL}$-concept descriptions, ordered by the subsumption relation, is isomorphic to a certain subset – that depends on the definitions of the TBox – of the lattice of ideals of the partially ordered set of equivalence classes built on the union of concept names and role concepts. We then proposed a simple algorithm exploiting this structure to learn terminological axioms from examples. Every implication found in the data is added to the TBox. We can easily make this algorithm interactive. More precisely, it is possible to change it into an attribute exploration-like algorithm in which experts are asked about each axiom and may give counterexamples. In this work, we dealt with description logic $\mathcal{EL}$ but the main idea of considering sets of incomparable concepts names is also valid for DLs with the concept $\bot$ or the constructor $\forall$. However, it no longer works with constructors such as the negation because it adds new constraints between concept names. More complex DL languages will be the subject of future investigations on our part.

References

1. Franz Baader and Felix Distel. A finite basis for the set of el-implications holding in a finite model. In *In ICFCA, vol.4933 of LNAI*, pages 46–61. Springer Verlag, 2008.
2. Franz Baader, Bernhard Ganter, Baris Sertkaya, and Ulrike Sattler. Completing description logic knowledge bases using formal concept analysis. In *In Proc. of IJCAI 2007*, pages 230–235. AAAI Press, 2007.
3. Franz Baader and Baris Sertkaya. Applying formal concept analysis to description logics. In Peter Eklund, editor, *Concept Lattices*, volume 2961 of *Lecture Notes in Computer Science*, pages 593–594. Springer Berlin / Heidelberg, 2004.
4. Felix Distel. *Learning Description Logic Knowledge Bases from Data Using Methods from Formal Concept Analysis.* PhD thesis, Technische Universität Dresden, 2011.
5. Sébastien Ferré and Olivier Ridoux. A logical generalization of formal concept analysis. In *Int. Conf. Conceptual Structures, LNCS 1867*, pages 371–384. Springer, 2000.
6. Sebastian Rudolph. Exploring relational structures via fle. In *Conceptual Structures at Work: 12th International Conference on Conceptual Structures. Volume 3127 of LNCS*. Springer, 2004.
7. Sebastian Rudolph. *Relational Exploration - Combining Description Logics and Formal Concept Analysis for Knowledge Specification.* Universitätsverlag Karlsruhe, December 2006.

8. N. V. Shilov and S.-Y. Han. A proposal of description logic on concept lattices. In *Proceedings of the Fifth International Conference on Concept Lattices and their Applications*, 2007.