A Note on the Translation of Conceptual Data Models into Description Logics: Disjointness and Covering Assumptions

Giovanni Casini
Centre for AI Research
CSIR and UKZN
South Africa
GCasini@csir.co.za

Aurona Gerber
Centre for AI Research
CSIR and UKZN
South Africa
AGerber@csir.co.za

Thomas Meyer
Centre for AI Research
CSIR and UKZN
South Africa
TMeyer@csir.co.za

ABSTRACT

Conceptual modeling is nowadays mostly done using languages such as Entity-Relationship (ER) Models, Unified Modeling Language (UML), and Object-Role Modeling (ORM). These models are used to depict the ontological organization of relevant concepts or entities. Such formalisms share a common modeling approach, based on the notions of class or entity and the relations or associations between classes or entities. Recent developments in knowledge representation using logic-based ontologies have created new possibilities for conceptual data modeling. It also raises the question of how existing conceptual models using ER, UML or ORM could be translated into Description Logics (DLs), a family of logics that have proved to be particularly appropriate for formalizing ontologies and reasoning about them. Given a conceptual data model, two assumptions are usually made that are not explicitly stated but need to be clarified for its DL translation: (1) disjointness assumption: all the classes are to be assumed pairwise disjoint if not specified otherwise; and (2) covering assumption: the content of every class must correspond to the union of its immediate subclasses (this includes the assumption that we do not consider anything apart from what is expressed in the model). In this paper we propose two simple procedures to assist modelers with integrating these assumptions into their models, thereby allowing for a more complete translation into DLs.

Categories and Subject Descriptors

I.2.4 [**Knowledge Representation Formalisms and Methods**]: Representation languages, Semantic networks

General Terms

Algorithms, Design, Languages, Theory

Keywords

Description Logics, UML, ER, ORM, disjointness, covering, non-monotonic reasoning

1. INTRODUCTION

The development of formal languages appropriate for the specification of conceptual models (including ontologies) is an important issue in the Semantic Web research field. In what follows we are going to take under consideration two approaches for the definition of ontologies. On the one hand, the OWL (Web Ontology Language) class of languages represents the main formalism for specifying the structure of complex ontologies [6].

On the other hand, there are more intuitive modeling technologies, the *conceptual* (or *semantic*) *models*, that are used as tools for, for instance, system design or database schema design. At present, the three most used languages for conceptual data modeling are Entity-Relationship (ER) Models [3], Unified Modeling Languages (UML) [10], and Object-Role Modeling (ORM) Languages [4]. Despite their differences, these languages are very similar in that they model entities (objects or classes) and the relationships between these entities. UML, for instance, models the notions of class and of association between classes, thus allowing a user to characterize the organization of classes, specifying classes, attributes of such classes, operations on classes and the associations between such classes.

Both the OWL languages and the conceptual data modeling languages UML, ER, and ORM share the fundamental property of being translatable into *Description Logics* (DLs), a family of logics corresponding to decidable fragments of first order logic. That is, every OWL ontology and every UML/ER/ORM diagram can be translated into a corresponding DL knowledge base, so that we can work on such a knowledge base using logical instruments to analyze its properties (*e.g.* its consistency) or to draw conclusions from it. This has advantages such as reasoning support for DL that could check the consistency of models.

However, when modeling is done using UML/ER/ORM, modelers often make assumptions which are not explicitly stated. When the translation to DL is done, it is necessary to formalize these assumptions. In this paper we discuss the formalization of two assumptions typically present in conceptual data models, which we call (1) disjointness and (2) covering constraints. These two assumptions correspond to a defeasible version of two restrictions that can be imposed on the classes of objects in our diagrams:

- Disjointness: The classes in the set $\{C_1, \ldots, C_n\}$ are pairwise disjoint. Such a property can be expressed in first order logic as

$$\forall x.C_i(x) \supset \bigwedge_{j=i+1}^{n} \neg C_j(x) \quad \text{for } i \in \{1,\dots,n\}$$

- Covering constraint: Assume that the subclasses of a class C are exactly the ones in the set $\{C_1,\dots,C_n\}$ (*i.e.* $\forall i_{1\leq i\leq n}.C_i \supset C$); imposing a covering constraint corresponds to stating that every element of C falls under one of the specified subclasses $C_1,\dots,C_n$. That is,

$$\forall x.C(x) \supset \bigvee_{i=1}^{n} C_i(x).$$

At present such constraints can only be *imposed* in specific cases in conceptual data models, that is, we can indicate in which cases we want them to be necessarily satisfied. But they are also generally considered to be a pair of desirable assumptions, to be held as valid whenever it is possible. Our aim is the formalization of such assumptions in a defeasible version, that is, we are going to provide two procedures that allow us to implement such assumptions in the DL translations of our ontologies whenever they are consistent with the content of the knowledge base.

In what follows, we shall firstly present our procedures for a general case, *i.e.* for knowledge bases defined using the DL $\mathcal{SROIQ}$, that is the logical counterpart of the language OWL2; then we check our proposals w.r.t. conceptual data models, and in particular we shall consider the UML language. The paper is organized as follows. Section 2 is devoted to the expression of the formalisms we are going to use: in section 2.1 we present the DL languages we use, $\mathcal{SROIQ}$ and its sublanguage $\mathcal{ALCQI}$, while in section 2.2 we present the structure of UML diagrams and the rules for their translation into the DL $\mathcal{ALCQI}$; we use UML since it is the main representative of the class of conceptual data models, and what we are going to apply to it will be valid also for ER and ORM. In section 3 we present our proposals, both for the more general case of $\mathcal{SROIQ}$ knowledge bases and in particular for $\mathcal{ALCQI}$ knowledge bases obtained by the translation of UML diagrams.

2. FORMAL LANGUAGES

We briefly present the two kinds of formalisms we are going to use, that is the DLs $\mathcal{SROIQ}$ and $\mathcal{ALCQI}$, and UML diagrams.

2.1 Description logics $\mathcal{SROIQ}$ and $\mathcal{ALCQI}$

DLs correspond to fragments of classical first order logic. They play a particular role since they are the logical counterpart of the web ontology languages of the OWL family [11].

We shall refer here to a particular expressive DL, namely $\mathcal{SROIQ}$ (see [5]), which is well known for being the logical language corresponding to the Web ontology language OWL2.

The signature of the $\mathcal{SROIQ}$ system is based on a set of concept names $\mathcal{A}t = \{A, B, \dots\}$, a set of role names $\mathcal{S} = \{R, S, \dots\}$ containing the universal role U, and a set of individuals $\mathcal{O} = \{a, b, \dots\}$. The set $\mathcal{R}$ of roles is defined as:

- $\mathcal{S} \subseteq \mathcal{R}$;
- If $R \in \mathcal{R}$, then $R^- \in \mathcal{R}$;
- If $\{R_1,\dots,R_n\} \subseteq \mathcal{R}$, $w = R_1 \dots R_n$ is in $\mathcal{R}$

The set $\mathcal{C}$ of $\mathcal{SROIQ}$'s concepts is defined inductively as:

- $\mathcal{A}t \subseteq \mathcal{C}$;
- $\top, \bot \in \mathcal{C}$;
- If $\{a_1,\dots,a_n\} \subseteq \mathcal{O}$, then $\{a_1,\dots,a_n\} \in \mathcal{C}$ (*oneOf* concepts);
- If $C, D \in \mathcal{C}$, then $C \sqcap D, C \sqcup D, \neg C \in \mathcal{C}$;
- If $C \in \mathcal{C}, R \in \mathcal{R}$, then $\exists R.C, \forall R.C, (\geq_n R), (\leq_n R) \in \mathcal{C}$.

Logically, concepts correspond to unary predicates, *i.e.* they describe classes of individuals (*e.g.* 'Persons'), while roles correspond to binary predicates, that is, they specify relations between individuals (*e.g.* 'to be father of').

We use a classical set-theoretic semantics. An interpretation $\mathcal{I}$ is a pair $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is a nonempty set, called *domain*, and the *interpretation function* $\cdot^{\mathcal{I}}$ assigns to every individual a member of the domain $\Delta^{\mathcal{I}}$, to every concept name a subset of $\Delta^{\mathcal{I}}$, and to every role name a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The function $\cdot^{\mathcal{I}}$ is extended to all the concepts and roles in the following way:

- $\{o_1,\dots,o_n\}^{\mathcal{I}} = \{o_1^{\mathcal{I}},\dots,o_n^{\mathcal{I}}\}$
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
- $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
- $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}}/C^{\mathcal{I}}$
- $(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y.(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
- $(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y.(x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
- $(\geq_n R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#(\{y \mid (x,y) \in R^{\mathcal{I}}\}) \geq n\}$
- $(\leq_n R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \#(\{y \mid (x,y) \in R^{\mathcal{I}}\}) \leq n\}$
- $(R^-)^{\mathcal{I}} = \{(a,b) \mid (b,a) \in R\}$
- $(R_1 \dots R_n)^{\mathcal{I}} = R_1^{\mathcal{I}} \circ \dots \circ R_n^{\mathcal{I}}$

where $\circ$ represents the classical composition of binary relations and $\#S$ the cardinality of set $S \subseteq \Delta^{\mathcal{I}}$. A knowledge base L is a pair $\langle \mathcal{A}, \mathcal{T} \rangle$, where $\mathcal{A}$ is an *ABox*, containing information about the individuals, and $\mathcal{T}$ is a *TBox*, containing information about the relations between the concepts (the structure of the ontology we are working on). The kind of axioms contained in the *ABox* and the *TBox* are described in Table 1, with their respective semantic meanings (where the set $Diag^{\mathcal{I}}$ represents the set of pairs $\{\langle x,x \rangle \mid x \in \Delta^{\mathcal{I}}\}$). We use $C \supset D$ and $C = D$ as shorthands of, respectively, the concept $\neg C \sqcup D$ and the axiom $\top \sqsubseteq (\neg C \sqcup D) \sqcap (\neg D \sqcup C)$.

We are going to work also with the DL $\mathcal{ALCQI}$, that corresponds to $\mathcal{SROIQ}$ without the possibility of defining *oneOf* axioms, of composing the roles and without all the axioms regarding roles (*i.e.*, we do not have role inclusions, symmetry, transitivity, reflexivity, irreflexivity, and disjointness axioms).

	Axiom name	Syntax	Semantics
ABox	Concept membership	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
	Role membership	$R(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
TBox	Conc. inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
	Role inclusion	$R_1 \dots R_n \sqsubseteq S$	$R_1^{\mathcal{I}} \circ \dots \circ R_n^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
	Symmetry	$Sym(R)$	$\langle x,y\rangle \in R^{\mathcal{I}} \Rightarrow$ $\Rightarrow \langle y,x\rangle \in R^{\mathcal{I}}$
	Transitivity	$Tra(R)$	$\langle x,y\rangle, \langle y,z\rangle \in R^{\mathcal{I}} \Rightarrow$ $\Rightarrow \langle x,z\rangle \in R^{\mathcal{I}}$
	Reflexivity	$Ref(R)$	$Diag^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
	Irreflexivity	$Irr(R)$	$Diag^{\mathcal{I}} \cap R^{\mathcal{I}} = \emptyset$
	Disjointness	$Dis(R,S)$	$R^{\mathcal{I}} \cap S^{\mathcal{I}} = \emptyset$

Table 1: Axioms of the *ABox* and the *TBox*

With $\models$ we denote the classical, monotonic, consequence relation associated to the specific DL we are working with (that will be clear from the context).

Our aim is the formalization of the following two assumptions:

- *Disjointness assumption*. Specifying an ontology, we assume that the classes we introduce, *i.e.*the elementary concepts in our vocabulary, are pairwise disjoint, if not informed of the contrary.
- *Covering assumption*. The content of a class must correspond to the union of its immediate subclasses. This is a form of closed world assumption: consider an elementary concept, and consider its subconcepts as expressed in the *TBox*; if not informed of the contrary, we assume that the extension of a concept corresponds to the union of the extensions of its subconcepts.

In what follows we are going to assume that our *TBox*es are *classified*, that is, all the inclusion axioms $A \sqsubseteq B$ holding between the atomic concepts of our language are explicitly present in the knowledge base.

2.2 UML class diagrams

A UML class diagram is a static structure that represents the ontological structure of a specific domain: it indicates what kind of objects are part of the domain (its *classes*) and what kind of relations hold between them.

The elements of UML class diagrams are:

- *Classes*: sets of individuals sharing common properties.
- *Associations*: relations holding between classes.

Between associations, two kind of relations have a particular relevance:

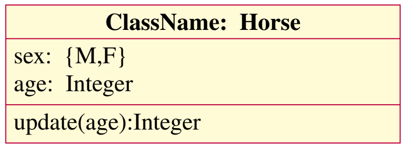

Figure 1: UML class

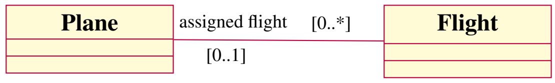

Figure 2: Binary association

- *Aggregations*: ‘part of’ relations between classes (*e.g.* an element of the class ‘piston’ is part of an element of the class ‘engine’).
- *Generalizations*: subsumption relation between classes (*e.g.* the class ‘horse’ is a subclass of the class ‘animal’).

More in detail, *classes* represent sets of objects characterized by common features, and are described by three main components:

- *Name*: the name of the class (*e.g.* ‘Horse’).
- *Attributes*: the list of the properties that characterize all the elements in the class (*e.g.* ‘age’). An attribute a is associated to a *type* T, *i.e.* the domain of the possible value of each attribute (‘Integer’) and a *multiplicity*, *i.e.* an interval $[i \dots j]$ stating that a is specified using from i to j values from T; if the multiplicity corresponds to $[1 \dots 1]$, as in figure 1, we can omit it.
- *Operations*: we can associate to each class a function from an attribute of the class and possible additional parameters; we can indicate an operation as a function $f(P_1, \dots, P_m) : R$, where $P_1, \dots, P_m$ are the types of the m parameters, and R is the type of the result (*e.g.*‘update(age): Integer’).

Excluding the name, all the other data are not mandatory in the definition of a class. Graphically, a class is characterized by a rectangle divided into three parts, each representing a main part of the definition of a class, as depicted in figure 1.

An *association* is a relation between the instances of two or more classes. We can associate to every concept participating in an association a multiplicity $[i \dots j]$ indicating that each instance of a class C can participate in the said association at least i times and at most j times (again, we can omit the multiplicity if it is $[1 \dots 1]$).

Graphically, an association is indicated as in figure 2, where the binary association ‘assigned flight’ relates each flight with a plane.

Among the associations, some play an important role, in particular *aggregations* and *generalizations*.

An aggregation is a binary association that formalizes a ‘part of’ relation between two classes. That is, if there is an aggregation G between two classes C and D, the elements of the class C are

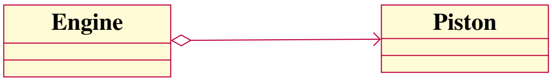

Figure 3: Aggregation

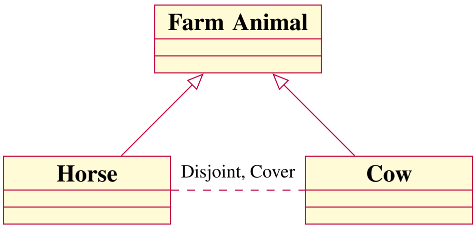

Figure 4: Generalizations

part of the elements of the class D (*e.g.* the elements of the class 'piston' are part of the elements of the class 'engine'). Graphically, it is indicated by a line ending in a diamond, as in figure 3.

A generalization corresponds to a subsumption relation between two classes (*e.g.* the class 'animal' is a generalization of the more specific class 'horse'). All the attributes of a superclass are inherited by the subclass. Graphically, it is indicated by an arrow, as in figure 4.

Disjointness and *covering* are two constraints that are connected to the generalization relation. If we want to *impose* such constraints in particular cases, we can indicate such constraints on our UML diagram, as in figure 4 where we indicate that the classes 'Horse' and 'Cow' *must* be disjoint and their union *must* contain all the individuals falling under the class 'Farm Animal'.

Here we want to formalize a form of defeasible assumptions about disjointness and covering, that is, we want to reason as if all the classes in our model were pairwise disjoint and as if the extension of each class corresponded to the union of the extensions of its immediate subclasses. There is a big difference between the imposition of disjointness and covering and their defeasible assumption: in the former case we want such constraints to be satisfied, and if we find exceptions to such constraints we end up with an inconsistent knowledge base; in the latter case we assume disjointness and covering, and we reason on the basis of our ontology as if such constraints are satisfied everywhere, but, in case we are presented with exceptions, we simply give up the constraints without the need to revise the information in our diagram.

2.2.1 Translation of UML schemas into $\mathcal{ALCQI}$

Each UML schema can be translated into a DL knowledge base. We refer here to the translation of UML schemas into $\mathcal{ALCQI}$ knowledge bases presented in [2].

In order to ease the reading, we have used the same symbols to refer both to the elements of UML diagrams and to the associated elements in $\mathcal{ALCQI}$.

Every UML class is represented simply by an atomic concept C. Each attribute a of type T associated to the class C is translated into a role a that connects C with T, *i.e.*

$$C \sqsubseteq \forall a.T$$

while the associated multiplicity $[i \ldots j]$ is formalized by

$$C \sqsubseteq (\geq_i a.\top) \sqcap (\leq_j a.\top)$$

For multiplicities $[0 \ldots *]$ we can omit the above statement, while for multiplicities $[1 \ldots 1]$ the statement becomes

$$C \sqsubseteq (\exists a.\top) \sqcap (\leq 1 a.\top)$$

An operation $f : S$ without parameters is translated using a role R_f into the statement

$$C \sqsubseteq \forall R_f.S \sqcap (\leq_1 R_f.\top)$$

An operation with m parameters $f(P_1, \ldots, P_m) : S$ for class C cannot be directly expressed in $\mathcal{ALCQI}$, since it would correspond to an $(m+2)$-ary role, while the language of $\mathcal{ALCQI}$ allows only binary roles. So we express it through *reification*: we introduce in the $\mathcal{ALCQI}$ language a concept $C_{f(P_1,\ldots,P_m)}$ and $m+2$ roles $r_1, \ldots, r_{m+2}$, and we add to our knowledge base the following statements:

$$C_{f(P_1,\ldots,P_m)} \sqsubseteq \exists r_1.\top \sqcap (\leq 1 r_1.\top) \sqcap \ldots \sqcap \exists r_{m+1}.\top \sqcap (\leq 1 r_{m+1}.\top)$$

$$C \sqsubseteq \forall r_1^-.(C_{f(P_1,\ldots,P_m)} \supset \forall r_{m+2}.S)$$

For a binary association R_A between two classes C and D we use an atomic role R_A and the following inclusion axiom:

$$\top \sqsubseteq \forall R_A.D \sqcap \forall R_A^-.C$$

that encodes the typing of A, while the multiplicities of A are formalized by

$$\top \sqsubseteq (\geq_{n_l} R_A.\top) \sqcap (\leq_{n_u} R_A.\top)$$

$$\top \sqsubseteq (\geq_{m_l} R_A^-.\top) \sqcap (\leq_{m_u} R_A^-.\top)$$

Since they are binary associations, we use such a translation also for aggregations.

For n-ary ($n > 2$) associations we use reification again. An association A relating the classes $C_1, \ldots, C_n$ is represented by means of an atomic concept A, a set of roles $\{r_1, \ldots, r_n\}$, and the statement

$$A \sqsubseteq \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_n.C_n \sqcap (\leq_1 r_1.\top) \sqcap \ldots \sqcap (\leq_1 r_n.\top)$$

Its multiplicity is represented as

$$\top \sqsubseteq (\geq_{n_l} r_1^- A.\top) \sqcap (\leq_{n_u} r_1^- A.\top)$$

$$\top \sqsubseteq (\geq_{m_l} r_2^- A^-.\top) \sqcap (\leq_{m_u} r_2^- A^-.\top)$$

Eventually, generalizations are represented by means of inclusion axioms, that is, if in the diagram there is an arrow going from class C to class D, we simply add to the knowledge base the inclusion axiom

$$C \sqsubseteq D$$

So, given a UML diagram $\mathcal{D}$, we create an $\mathcal{ALCQI}$ knowledge base $\mathcal{T}_\mathcal{D}$ with the following vocabulary:

- $\mathcal{A}t_C = \{C_1, \ldots, C_l\}$ is the set of the atomic concepts representing the classes;
- $\mathcal{S}_A = \{R_{A_1}, \ldots, R_{A_m}\}$ is the set of the atomic roles representing the binary associations;
- $\mathcal{A}t_A = \{A_1, \ldots, A_n\}$ is the set of the atomic concepts representing n-ary ($n > 2$) associations;
- $\mathcal{A}t_T = \{T_1, \ldots, T_o\}$ is the set of the atomic concepts representing the types of the attributes;
- $\mathcal{A}t_{op} = \{C_{f_1}, \ldots, C_{f_p}\}$ is the set of the atomic concepts representing the parametrized operations;
- $\mathcal{S}_a = \{a_1, \ldots, a_q\}$ is the set of the atomic roles representing the attributes;
- $\mathcal{S}_f = \{R_{f_1}, \ldots, R_{f_s}\}$ is the set of the atomic roles representing the operations without parameters;
- $\mathcal{S}_{ass} = \{r_1, \ldots, r_t\}$ is the set of the atomic roles used for the reification procedure for the n-ary ($n > 2$) associations.

In general, all the atomic concepts we are going to deal with have to be considered pairwise disjoint; the issue of disjointness arises only if we are dealing with the concepts representing the classes, whose intersection could be non-empty. So, in general, for $C \in \mathcal{A}t_C \cup \mathcal{A}t_A \cup \mathcal{A}t_T \cup \mathcal{A}t_P$ and $D \in \mathcal{A}t_A \cup \mathcal{A}t_T \cup \mathcal{A}t_P$ we can impose in our knowledge base

$$\top \sqsubseteq \neg(C \sqcap D)$$

That is, for every pair of atomic concepts C, D we impose their pairwise disjointness if they do not both represent classes.

If our diagram requires a set of classes $C_1, \ldots, C_n$ to be disjoint, in our $\mathcal{ALCQI}$ knowledge base such a constraint corresponds to introducing $n-1$ axioms

$$C_i \sqsubseteq \bigsqcap_{j=i+1}^{n} \neg C_j, \text{ for } i \in \{1, \ldots, n-1\}$$

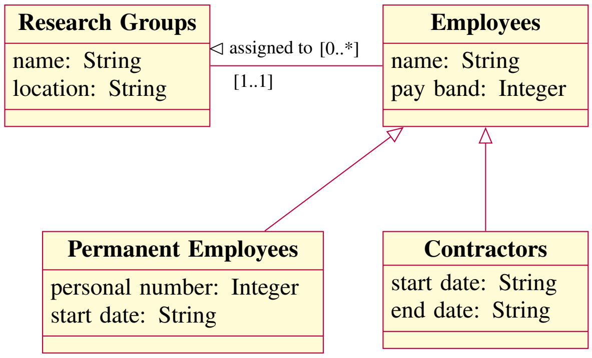

Figure 5: Example 2.1, Diagram $\mathcal{D}$

For the imposition of the covering constraint (an instance of C must be an instance of at least one of $C_1, \ldots, C_n$) we can introduce the following $\mathcal{ALCQI}$ axiom:

$$C \sqsubseteq \bigsqcup_{i=1}^{n} C_i$$

We want to introduce in our knowledge base a kind of defeasible information stating that all the concepts are pairwise disjoint if we are not informed of the contrary, and all the generalizations respect the covering constraint, if we do not face exceptions.

By translating a UML diagram $\mathcal{D}$ into an $\mathcal{ALCQI}$ *TBox* $\mathcal{T}_\mathcal{D}$ we obtain a knowledge base in which all the immediate subsumption relations are explicit, that is, we obtain a complete classification of the *TBox* just making a transitive closure of the expressed inclusion axioms (if $C \sqsubseteq D$ and $D \sqsubseteq E$ are in the *TBox*, then we can add also $C \sqsubseteq E$).

EXAMPLE 2.1. *Consider the diagram $\mathcal{D}$ presented in figure 5. It represents the ontology of the employees of a university, where each employee, besides being associated with a specific research group, can be either a permanent employee or a contractor. Where it is not specified, the multiplicity is to be considered* [1..1].

Let E, RG, PE, C, S, I be the atomic concepts translating, respectively, the classes Employee, Research Groups, Permanent Employees, Contractors, *and the types* String *and* Integer, *while the roles $at, n, pb, l, pn, sd, ed, af$ translate, respectively, the association* assigned to *and the attributes* name, pay band, location, personal number, start date, end date, affiliation. *Hence, the diagram $\mathcal{D}$ is translated into the following TBox $\mathcal{T}_\mathcal{D}$:*

- $E \sqsubseteq \forall n.S$
- $E \sqsubseteq (\exists n.\top) \sqcap (\leq_1 n.\top)$
- $E \sqsubseteq \forall pb.I$
- $E \sqsubseteq (\exists pb.\top) \sqcap (\leq_1 pb.\top)$
- $RG \sqsubseteq \forall n.S$
- $RG \sqsubseteq (\exists n.\top) \sqcap (\leq_1 n.\top)$

- $RG \sqsubseteq \forall l.S$
- $RG \sqsubseteq (\exists l.\top) \sqcap (\leq_1 l.\top)$
- $PE \sqsubseteq \forall pn.I$
- $PE \sqsubseteq (\exists pn.\top) \sqcap (\leq_1 pn.\top)$
- $PE \sqsubseteq \forall sd.S$
- $PE \sqsubseteq (\exists sd.\top) \sqcap (\leq_1 sd.\top)$
- $C \sqsubseteq \forall sd.S$
- $C \sqsubseteq (\exists sd.\top) \sqcap (\leq_1 sd.\top)$
- $C \sqsubseteq \forall ed.S$
- $C \sqsubseteq (\exists ed.\top) \sqcap (\leq_1 ed.\top)$
- $\top \sqsubseteq \forall at.RG \sqcap \forall at^{-}.E$
- $\top \sqsubseteq (\geq_1 at.\top) \sqcap (\leq_1 at.\top)$
- $\top \sqsubseteq (\geq_0 at^{-}.\top) \sqcap (\leq_* at^{-}.\top)$
- $PE \sqsubseteq E$
- $C \sqsubseteq E$

To these we add all the disjunction axioms $C \sqsubseteq \neg D$ for every pair C, D s.t. they do not both represent classes (that is, $\{C, D\} \not\subseteq \{E, RG, PE, C\}$).

Consistency. Dealing with conceptual data models we have to consider two kinds of consistency:

- **Consistency of the whole diagram.** A diagram $\mathcal{D}$ is *consistent* if it admits an *instantiation*, that is, we can create a realization of the diagram in which, respecting all the given restrictions, there is at least an individual in at least one class. From the logical point of view, such a kind of consistency corresponds to the classical logical notion of consistency, that is, given a diagram $\mathcal{D}$ and the corresponding *TBox* $\mathcal{T}_{\mathcal{D}}$, the diagram $\mathcal{D}$ is consistent iff $\mathcal{T}_{\mathcal{D}}$ is a logically consistent *TBox*, that is, $\mathcal{T}_{\mathcal{D}} \not\models \top \sqsubseteq \bot$.
- **Class consistency.** A diagram $\mathcal{D}$ is class consistent if there is a realization respecting all the specified constraints s.t. no class is empty. From the logical point of view, it corresponds to saying that for all the atomic concepts $C_1, \ldots, C_n$ present in $\mathcal{A}t_C$, $\mathcal{T}_{\mathcal{D}} \not\models C_i \sqsubseteq \bot$ ($1 \leq i \leq n$).

In what follows we are going to use the latter kind of consistency as our main notion of consistency, since, working with ontologies, the notion of *class consistency* seems more appropriate: if we are going to add disjointness axioms to an ontology that is class consistent, we want to end up with a new ontology that is again class consistent, since the consistency of each class takes precedence over the application of the disjointness and covering assumptions.

3. DISJOINTNESS ASSUMPTION

We want to implement in our *TBox*es the assumption that, if we are not informed of the contrary, all the atomic concepts are pairwise disjoint. First, we shall introduce our procedure for the more general case of a $\mathcal{SROIQ}$ knowledge base, and later we shall consider how our procedures work w.r.t. a concept data model such as UML diagrams.

3.1 The $\mathcal{SROIQ}$ case

To present our procedure for $\mathcal{SROIQ}$ knowledge bases we assume that we are going to work with classified *TBox*es, that is, all the subsumption relations holding between the atomic concepts are explicitly stated.

Consider two concepts C and D; their disjointness can be expressed with the axiom $\top \sqsubseteq \neg(C \sqcap D)$. We want to implement such an information whenever it is possible, that is, whenever it is consistent with the information contained in the *TBox*. In order to do so, we add to the knowledge base a new kind of axioms, *defeasible axioms*, that have to be considered true just in case they are consistent to the rest of the information contained in the *TBox*. So, we introduce a set $\mathcal{B}$ of *defeasible axioms*, that we indicate as $\top \mathrel{\underset{\sim}{\sqsubset}} \neg(C \sqcap D)$, that can be read as ‘Presumably, the concepts C and D are disjoint’.

Hence, the knowledge base $\mathcal{B}$ will contain defeasible disjointness axioms, one for every pair of concepts named in our knowledge base $\langle \mathcal{A}, \mathcal{T} \rangle$. That is, if we have a vocabulary with $\mathcal{A}t = \{C_1, \ldots, C_n\}$, $\mathcal{B} = \{\top \mathrel{\underset{\sim}{\sqsubset}} \neg(C_i \sqcap C_j) \mid C_i, C_j \in \mathcal{A}t$ and $i < j\}$ (note that $\#\mathcal{B} = \frac{n^2-n}{2}$).

In order to decide which disjointness assumptions we shall hold as true, we refer to the notion of maximally consistent sets, a classical notion used both in nonmonotonic reasoning ([9]), and belief revision ([1]). Given the set $\mathcal{B}$, we consider its *materialization* $\mathcal{B}^{\sqsubseteq}$, that is the set obtained by transforming each defeasible inclusion axiom $\top \mathrel{\underset{\sim}{\sqsubset}} \neg(C \sqcap D)$ in $\mathcal{B}$ into a standard inclusion axiom $\top \sqsubseteq \neg(C \sqcap D)$, that is, $\mathcal{B}^{\sqsubseteq} = \{\top \sqsubseteq \neg(C \sqcap D) \mid \top \mathrel{\underset{\sim}{\sqsubset}} \neg(C \sqcap D) \in \mathcal{B}\}$.

DEFINITION 3.1 (MAXIMALLY CONSISTENT SUBSET). *Assume a TBox $\mathcal{T}$ and a set of inclusion axioms S. $S' \subseteq S$ is a maximal subset of S w.r.t. $\mathcal{T}$ iff S' and $\mathcal{T}$ are consistent ($S' \cup \mathcal{T} \not\models \mathcal{C} \sqsubseteq \bot$ for every atomic concept C, with $\models$ the classical $\mathcal{SROIQ}$ consequence relation), and there is not any other set S'' s.t. $S' \subset S'' \subseteq S$ and S'' is consistent with $\mathcal{T}$.*

So, given the materialization $\mathcal{B}^{\sqsubseteq}$, we call $M_{\mathcal{T}}^{\mathcal{B}^{\sqsubseteq}}$ the set of the maximally $\mathcal{T}$-consistent subsets of $\mathcal{B}^{\sqsubseteq}$. Every maximally $\mathcal{T}$-consistent subset of $\mathcal{B}^{\sqsubseteq}$ represents a maximal amount of ‘disjointness’ we can introduce in our knowledge base, avoiding inconsistencies. Now, we have three main options about how to introduce such disjointness information into our knowledge base.

1. *Choice approach.* We choose just one of the sets in $M_{\mathcal{T}}^{\mathcal{B}^{\sqsubseteq}}$, and add it to $\mathcal{T}$. The resulting consequence relation will be defined as follows.

   DEFINITION 3.2 (CONSEQUENCE RELATION $\models^1_{\mathcal{B}}$). *Given a TBox $\mathcal{T}$ and a set of defeasible inclusion axioms $\mathcal{B}$, and consider just a set $S \in M_{\mathcal{T}}^{\mathcal{B}^{\sqsubseteq}}$. We say that a TBox-axiom α is a consequence of $\mathcal{T}$ and S ($\mathcal{T} \models^1_S \alpha$) iff the* *following holds:*

$$\mathcal{T} \models^1_S \alpha \textit{ iff } \mathcal{T} \cup S \models \alpha$$

2. *Skeptical approach.* Consider each set S in $M_{\mathcal{T}}^{\mathcal{B}^{\sqsubseteq}}$. We can define a consequence relation $\models^2_{\mathcal{B}}$ in the following way.

   DEFINITION 3.3 (CONSEQUENCE RELATION $\models^2_{\mathcal{B}}$). *Given a TBox $\mathcal{T}$ and a set of defeasible inclusion axioms $\mathcal{B}$, we say that a TBox-axiom α is a consequence of $\mathcal{T}$ and $\mathcal{B}$ ($\mathcal{T} \models^2_{\mathcal{B}} \alpha$) iff the following holds:*

$$\mathcal{T} \models^2_{\mathcal{B}} \alpha \textit{ iff } \mathcal{T} \cup S \models \alpha \textit{ for every } S \in M_{\mathcal{T}}^{\mathcal{B}^{\sqsubseteq}}$$

   This option corresponds to the classical default assumption approach presented by Poole [9].

3. We proceed as in the previous case, but we consider only the largest of the subsets of $\mathcal{B}$ that are consistent with $\mathcal{T}$. Define the set $M_{\mathcal{T}}^{max,\mathcal{B}^{\sqsubseteq}} \subseteq M_{\mathcal{T}}^{\mathcal{B}^{\sqsubseteq}}$ composed of the largest sets in $M_{\mathcal{T}}^{\mathcal{B}^{\sqsubseteq}}$, that is $M_{\mathcal{T}}^{max,\mathcal{B}^{\sqsubseteq}} = \{S \mid S \in M_{\mathcal{T}}^{\mathcal{B}^{\sqsubseteq}}$ and there is no $S' \in M_{\mathcal{T}}^{\mathcal{B}^{\sqsubseteq}}$ s.t. $\#(S') > \#(S)\}$. Consider each set S in $M_{\mathcal{T}}^{max,\mathcal{B}^{\sqsubseteq}}$. We can define a consequence relation $\models^3_{\mathcal{B}}$ in the following way:

   DEFINITION 3.4 (CONSEQUENCE RELATION $\models^3_{\mathcal{B}}$). *Given a TBox $\mathcal{T}$ and a set of defeasible inclusion axioms $\mathcal{B}$, we say that a TBox-axiom α is consequence of $\mathcal{T}$ and $\mathcal{B}$ ($\mathcal{T} \models^3_{\mathcal{B}} \alpha$) iff the following holds:*

$$\mathcal{T} \models^3_{\mathcal{B}} \alpha \textit{ iff } \mathcal{T} \cup S \models \alpha \textit{ for every } S \in M_{\mathcal{T}}^{max,\mathcal{B}^{\sqsubseteq}}$$

Between these three options we prefer the third one, $\models^3_{\mathcal{B}}$, since it corresponds to the DL version of a trivial case of a classical nonmonotonic consequence relations, *i.e.* the *lexicographic closure* [8], that is widely acknowledged to be well-behaved. In particular, our consequence relation $\models^3_{\mathcal{B}}$ falls under the case in which all the defeasible inclusion axioms are of level 0 ([8], Definition 1 and Theorem 1). Here we are not going to explain in detail why our construction can be considered an analogue of a trivial case of Lehmann's lexicographic closure, and we shall just limit ourselves in showing that all the presented consequence relations satisfy the structural properties characterizing preferential consequence relations [7], of which the lexicographic closure is a representative. The properties characterizing preferential consequence relations are considered the essential properties a defeasible consequence relation should satisfy; we won't discuss here the importance of the properties characterizing the preferential consequence relations, rather referring the reader to [7] for an in-depth treatment of the theoretical issues.

In order to express the preferential rules in our framework, we have to introduce some additional notation. Let α and β be *TBox* axioms (that is, each of them can be a concept or role inclusion axiom or a transitive axiom); we introduce a disjunction expression for the *TBox* axioms, $\alpha \sqcup \beta$, that is interpreted in the following way:

$$\mathcal{T} \models \alpha \sqcup \beta \quad \text{iff} \quad \mathcal{T} \models \alpha \text{ or } \mathcal{T} \models \beta.$$

This notation is used simply as an aid in the readability of the properties for consequence relations we discuss below. We are not changing the expressivity of the object language. The properties we want our consequence relation to satisfy in order to consider it a preferential consequence relations are the following:

(REF) $\mathcal{T} \models_{\mathcal{B}} \alpha$ for every $\alpha \in \mathcal{T}$

(LLE) $$\frac{\mathcal{T}, \alpha \models_{\mathcal{B}} \gamma \qquad \alpha \models \beta \qquad \beta \models \alpha}{\mathcal{T}, \beta \models_{\mathcal{B}} \gamma}$$

(RW) $$\frac{\mathcal{T} \models_{\mathcal{B}} \alpha \qquad \alpha \models \beta}{\mathcal{T} \models_{\mathcal{B}} \beta}$$

(CT) $$\frac{\mathcal{T}, \alpha \models_{\mathcal{B}} \beta \qquad \mathcal{T} \models_{\mathcal{B}} \alpha}{\mathcal{T} \models_{\mathcal{B}} \beta}$$

(CM) $$\frac{\mathcal{T} \models_{\mathcal{B}} \beta \qquad \mathcal{T} \models_{\mathcal{B}} \alpha}{\mathcal{T}, \alpha \models_{\mathcal{B}} \beta}$$

(OR) $$\frac{\mathcal{T}, \alpha \models_{\mathcal{B}} \gamma \qquad \mathcal{T}, \beta \models_{\mathcal{B}} \gamma}{\mathcal{T}, \alpha \sqcup \beta \models_{\mathcal{B}} \gamma}$$

where $\models$ is the classical monotonic consequence relation associated to $\mathcal{SROIQ}$.

PROPOSITION 3.1. *The consequence relations $\models^2_{\mathcal{B}}$, and $\models^3_{\mathcal{B}}$ are preferential.*

PROOF. The proof for the satisfaction of REF, LLE and RW are immediate. We shall prove the other properties just for $\models^3_{\mathcal{B}}$. The proofs for $\models^2_{\mathcal{B}}$ are analogous.

CT and CM. If $\mathcal{T} \models^3_{\mathcal{B}} \alpha$, it means that the set $M_{\mathcal{T}}^{max,\mathcal{B}^{\sqsubseteq}}$ is consistent also with α, that is, $M_{\mathcal{T}}^{max,\mathcal{B}^{\sqsubseteq}} = M_{\mathcal{T},\alpha}^{max,\mathcal{B}^{\sqsubseteq}}$. For CT, since for every $S \in M_{\mathcal{T}}^{max,\mathcal{B}^{\sqsubseteq}}$ we have that $\mathcal{T} \cup \{\alpha\} \cup S \models \beta$ and $\mathcal{T} \cup S \models \alpha$, we have also $\mathcal{T} \cup S \models \beta$, *i.e.* $\mathcal{T} \models^3_{\mathcal{B}} \beta$. For CM, $\mathcal{T} \models^3_{\mathcal{B}} \beta$ and $M_{\mathcal{T}}^{max,\mathcal{B}^{\sqsubseteq}} = M_{\mathcal{T},\alpha}^{max,\mathcal{B}^{\sqsubseteq}}$, we have $\mathcal{T}, \alpha \models^3_{\mathcal{B}} \beta$.

OR. We have $\mathcal{T}, \alpha \models^3_{\mathcal{B}} \gamma$ and $\mathcal{T}, \alpha \models^3_{\mathcal{B}} \gamma$. Let $M_{\mathcal{T},\alpha}^{max,\mathcal{B}^{\sqsubseteq}}$ and $M_{\mathcal{T},\beta}^{max,\mathcal{B}^{\sqsubseteq}}$ be the respective sets of maxi-consistent sets, with $\#M_{\mathcal{T},\alpha}^{max,\mathcal{B}^{\sqsubseteq}} = n$ and $\#M_{\mathcal{T},\beta}^{max,\mathcal{B}^{\sqsubseteq}} = m$. Both $M_{\mathcal{T},\alpha}^{max,\mathcal{B}^{\sqsubseteq}}$ and $M_{\mathcal{T},\beta}^{max,\mathcal{B}^{\sqsubseteq}}$ are consistent with $\mathcal{T}, \alpha \sqcup \beta$, since the former is consistent with some models satisfying α and the latter is consistent with some models satisfying β.

If $m = n$, $M_{\mathcal{T},\alpha\sqcup\beta}^{max,\mathcal{B}^{\sqsubseteq}} = M_{\mathcal{T},\alpha}^{max,\mathcal{B}^{\sqsubseteq}} \cup M_{\mathcal{T},\beta}^{max,\mathcal{B}^{\sqsubseteq}}$ and the conclusion is immediate.

If $m \neq n$, without lack of generality assume $m > n$. Then we have $M_{\mathcal{T},\alpha\sqcup\beta}^{max,\mathcal{B}^{\sqsubseteq}} = M_{\mathcal{T},\alpha}^{max,\mathcal{B}^{\sqsubseteq}}$. In such a case, we have that for every interpretation $\mathcal{I}$ s.t. $\mathcal{I} \models \mathcal{T}, \alpha, s$, for some $s \in M_{\mathcal{T},\alpha}^{max,\mathcal{B}^{\sqsubseteq}}$, $\mathcal{I} \models \gamma$, while we have no interpretations satisfying $\mathcal{I} \models \mathcal{T}, \beta, s$ for some $s \in M_{\mathcal{T},\alpha}^{max,\mathcal{B}^{\sqsubseteq}}$, otherwise we would have had $n \geq m$. So, since $\mathcal{T}, \alpha \models^3_{\mathcal{B}} \gamma$ we have also $\mathcal{T}, \alpha \sqcup \beta \models^3_{\mathcal{B}} \gamma$. $\square$

In order for $\models^1_{\mathcal{B}}$ to be preferential, we need to define appropriate choice functions over the sets $M^{\mathcal{B}^{\sqsubseteq}}_{\mathcal{T}}$.

EXAMPLE 3.1. *Assume we have a TBox* $\mathcal{T} = \{C \sqsubseteq D \sqcup E, \top \sqsubseteq \forall D.F, F \sqsubseteq G\}$, *and we want to apply the disjointness assumption. The set* $\mathcal{B}$ *will be* $\{\top \underset{\sim}{\sqsubset} \neg(C \sqcap D), \top \underset{\sim}{\sqsubset} \neg(C \sqcap E), \top \sqsubset \sim \neg(C \sqcap F), \top \underset{\sim}{\sqsubset} \neg(C \sqcap G), \top \underset{\sim}{\sqsubset} \neg(D \sqcap E), \top \underset{\sim}{\sqsubset} \neg(D \sqcap F), \top \underset{\sim}{\sqsubset} \neg(D \sqcap G), \top \underset{\sim}{\sqsubset} \neg(E \sqcap F), \top \underset{\sim}{\sqsubset} \neg(E \sqcap G), \top \underset{\sim}{\sqsubset} \neg(F \sqcap G)\}$. *The set* $M^{\mathcal{B}}_{\mathcal{T}}$ *will be composed of two sets* S *and* S', *with* $S = \{\top \underset{\sim}{\sqsubset} \neg(C \sqcap D), \top \underset{\sim}{\sqsubset} \neg(C \sqcap F), \top \underset{\sim}{\sqsubset} \neg(C \sqcap G), \top \underset{\sim}{\sqsubset} \neg(D \sqcap E), \top \underset{\sim}{\sqsubset} \neg(D \sqcap F), \top \underset{\sim}{\sqsubset} \neg(D \sqcap G), \top \underset{\sim}{\sqsubset} \neg(E \sqcap F), \top \underset{\sim}{\sqsubset} \neg(E \sqcap G)\}$ *and* $S' = \{\top \underset{\sim}{\sqsubset} \neg(C \sqcap E), \top \underset{\sim}{\sqsubset} \neg(C \sqcap F), \top \underset{\sim}{\sqsubset} \neg(C \sqcap G), \top \underset{\sim}{\sqsubset} \neg(D \sqcap E), \top \underset{\sim}{\sqsubset} \neg(D \sqcap F), \top \underset{\sim}{\sqsubset} \neg(D \sqcap G), \top \underset{\sim}{\sqsubset} \neg(E \sqcap F), \top \underset{\sim}{\sqsubset} \neg(E \sqcap G)\}$. *Since* S *and* S' *have the same cardinality,* $M^{max,\mathcal{B}^{\sqsubseteq}}_{\mathcal{T}} = M^{\mathcal{B}}_{\mathcal{T}}$, *and consequently working with the TBox* $\mathcal{T}$ *we have* $\models^2_{\mathcal{B}} = \models^3_{\mathcal{B}}$. *We can choose to consider both the sets* S *and* S', *using the consequence relation* $\models^{2/3}_{\mathcal{B}}$, *or to choose only one of them, reasoning with a consequence relation of the kind* $\models^1_{\mathcal{B}}$.

3.2 The UML case

Since $\mathcal{ALCQI}$ is a sublanguage of $\mathcal{SROIQ}$, the application of the disjointness assumption to the *TBox* $\mathcal{T}_{\mathcal{D}}$ obtained by the translation of a UML diagram $\mathcal{D}$ is just a special case of the $\mathcal{SROIQ}$ case. We just have to create the defeasible axiom base $\mathcal{B}$ considering only the concepts in $\mathcal{A}t_C$, the concepts representing classes, since for all the other concepts we already impose the disjointness in the translation.

EXAMPLE 3.2. *Consider the diagram in example 2.1. We want to implement the disjointness assumption. So, we add a set* $\mathcal{B}$ *of defeasible disjointness axioms containing the following expressions:*

- $\top \underset{\sim}{\sqsubset} \neg(RG \sqcap E)$
- $\top \underset{\sim}{\sqsubset} \neg(RG \sqcap PE)$
- $\top \underset{\sim}{\sqsubset} \neg(RG \sqcap C)$
- $\top \underset{\sim}{\sqsubset} \neg(E \sqcap PE)$
- $\top \underset{\sim}{\sqsubset} \neg(E \sqcap C)$
- $\top \underset{\sim}{\sqsubset} \neg(PE \sqcap C)$

We have only one maxi-consistent set S, *composed of:*

- $\top \underset{\sim}{\sqsubset} \neg(RG \sqcap E)$
- $\top \underset{\sim}{\sqsubset} \neg(RG \sqcap PE)$
- $\top \underset{\sim}{\sqsubset} \neg(RG \sqcap C)$
- $\top \underset{\sim}{\sqsubset} \neg(PE \sqcap C)$

We have to eliminate the axiom regarding E *and* PE *since it would have implied that the class* PE *is empty. The same holds for the axiom regarding* E *and* C, *that would have implied that* C *is empty. Since we have just one maxi-consistent set, our three consequence relations are identical, and we can reason holding as valid all the disjointness axioms in* S.

Now, consider that our UML model is augmented as shown in figure 6. In other words, we introduce the new classes 'PostDoc Researcher', 'Professors', and 'Guards' (that we translate in our TBox into, respectively, the elementary concepts PD, P, *and* G*). The new diagram presents us with a case of multiple inheritance: postdoc researchers can be considered as a class falling under both the class of Permanent Employees and the class of Contractors, since they inherit the properties of both (a personnel number, a start date and an end date).*

We have to augment our TBox with the following information:

- $PD \sqsubseteq C$
- $PD \sqsubseteq PE$
- $PD \sqsubseteq \forall af.S$
- $PD \sqsubseteq (\exists af.\top) \sqcap (\leq_1 af.\top)$
- $G \sqsubseteq C$
- $P \sqsubseteq PE$

For this new TBox we extend the set $\mathcal{B}$ *of defeasible axioms with:*

- $\top \underset{\sim}{\sqsubset} \neg(PD \sqcap RG)$
- $\top \underset{\sim}{\sqsubset} \neg(PD \sqcap E)$
- $\top \underset{\sim}{\sqsubset} \neg(PD \sqcap PE)$
- $\top \underset{\sim}{\sqsubset} \neg(PD \sqcap C)$
- $\top \underset{\sim}{\sqsubset} \neg(PD \sqcap P)$
- $\top \underset{\sim}{\sqsubset} \neg(PD \sqcap G)$
- $\top \underset{\sim}{\sqsubset} \neg(P \sqcap RG)$
- $\top \underset{\sim}{\sqsubset} \neg(P \sqcap E)$
- $\top \underset{\sim}{\sqsubset} \neg(P \sqcap PE)$
- $\top \underset{\sim}{\sqsubset} \neg(P \sqcap C)$
- $\top \underset{\sim}{\sqsubset} \neg(P \sqcap G)$
- $\top \underset{\sim}{\sqsubset} \neg(G \sqcap RG)$
- $\top \underset{\sim}{\sqsubset} \neg(G \sqcap E)$
- $\top \underset{\sim}{\sqsubset} \neg(G \sqcap PE)$
- $\top \underset{\sim}{\sqsubset} \neg(G \sqcap C)$

Again, we have only one maxi-consistent set S', *composed of:*

- $\top \underset{\sim}{\sqsubset} \neg(RG \sqcap E)$
- $\top \underset{\sim}{\sqsubset} \neg(RG \sqcap PE)$
- $\top \underset{\sim}{\sqsubset} \neg(RG \sqcap C)$
- $\top \underset{\sim}{\sqsubset} \neg(PD \sqcap RG)$
- $\top \underset{\sim}{\sqsubset} \neg(PD \sqcap P)$

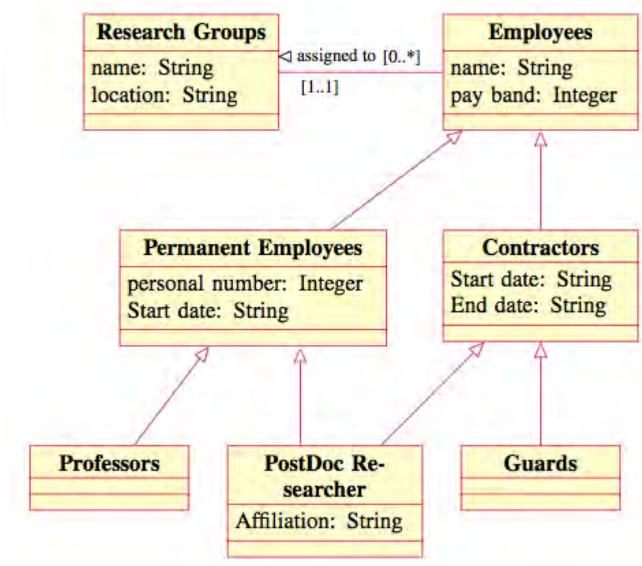

Figure 6: Example 3.2, Diagram $\mathcal{D}$

- $\top \underset{\sim}{\sqsubset} \neg(PD \sqcap G)$
- $\top \underset{\sim}{\sqsubset} \neg(P \sqcap RG)$
- $\top \underset{\sim}{\sqsubset} \neg(P \sqcap C)$
- $\top \underset{\sim}{\sqsubset} \neg(P \sqcap G)$
- $\top \underset{\sim}{\sqsubset} \neg(G \sqcap RG)$
- $\top \underset{\sim}{\sqsubset} \neg(G \sqcap PE)$

We have to eliminate all the axioms imposing the disjointness between PD and, respectively, PE, C, and E, since that would have implied that PD is empty, and analogously for P and G. Moreover, we have to eliminate the previously accepted axiom $\top \underset{\sim}{\sqsubset} \neg(PE \sqcap C)$, since it would have implied, again, that PD is empty.

The constraints in the form of a *TBox* resulting from the translation of a UML diagram have an interesting property w.r.t. our procedure, since we end up with just a single maxi-consistent set.

PROPOSITION 3.2. *Let $\mathcal{T}_{\mathcal{D}}$ be a consistent $\mathcal{ALCQI}$ TBox obtained from the translation of a UML diagram $\mathcal{D}$, and let $\mathcal{B}$ be the set of the defeasible disjointness axioms between the atomic concepts expressing the classes. The set $M^{\mathcal{B}^{\sqsubseteq}}_{\mathcal{T}}$ has a single element.*

In order to prove this proposition we need the following lemmas.

LEMMA 3.3. *Consider a consistent $\mathcal{ALCQI}$ TBox $\mathcal{T}$ composed only of inclusion axioms of the form $C \sqsubseteq D$ or $C \sqsubseteq \neg D$, with C and D atomic concepts. Let $\{\mathcal{T}^+, \mathcal{T}^-\}$ be a partition of $\mathcal{T}$ with $\mathcal{T}^+ = \{C \sqsubseteq D \mid C \sqsubseteq D \in \mathcal{T}\}$ and $\mathcal{T}^- = \{C \sqsubseteq \neg D \mid C \sqsubseteq \neg D \in \mathcal{T}\}$. All the inclusion axiom $E \sqsubseteq F$, with E and F atomic concepts, derivable from $\mathcal{T}$ are derivable from the transitive closure of the axioms in $\mathcal{T}^+$*

PROOF. The constrained forms of the axioms in $\mathcal{T}$, that do not use any role, allow us to reduce the problem to a propositional calculus problem, where we have a knowledge base $\mathcal{T}_p$ composed of formulae $C \supset D$ and $C \supset \neg D$, with C and D atomic sentences, $\mathcal{T}_p^+ = \{C \supset D \mid C \supset D \in \mathcal{T}\}$ and $\mathcal{T}_p^- = \{C \supset \neg D \mid C \supset \neg D \in \mathcal{T}\}$. We have to prove that we can derive a formula $E \supset F$ (that is, we can have $\mathcal{T}_p \models E \supset F$, with E, F atomic sentences and $\models$ the classical propositional consequence relation) iff it is derivable from the transitive closure of the implications in $\mathcal{T}_p^+$.

Obviously, if $E \supset F$ is derivable from the transitive closure of the implications in $\mathcal{T}_p^+$, it is classically derivable from $\mathcal{T}$.

Now, assume that $\mathcal{T} \models C \supset D$. We can use the resolution method to model $\models$, translating every implication $C \supset D$ (respectively, $C \supset \neg D$) into a clause $\{\neg C, D\}$ (respectively, $\{\neg C, \neg D\}$). Let $\overline{\mathcal{T}_p}$ be the translation of $\mathcal{T}_p$ into this clausal form. In order to prove that $E \supset F$ (*i.e.* $\{\neg E, F\}$) is valid, we have to prove that the set of clauses composed of $\overline{\mathcal{T}_p}$, namely $\{E\}, \{\neg F\}$ resolves into an empty set (that is, $\mathcal{T}_p, \neg(E \supset F) \models \bot$).

Since $\mathcal{T}_p$ is consistent, the set of clauses $\overline{\mathcal{T}_p}$ cannot resolve into an empty set. Moreover, since $\overline{\mathcal{T}_p}$ is composed only of binary clauses of the form $\{\neg C, D\}$ or $\{\neg C, \neg D\}$, every reduction step between them gives back again a binary clause; for example, from the clauses $\{\neg C, D\}$ and $\{\neg D, \neg B\}$ we obtain a new binary clause $\{\neg D, \neg B\}$. So, in order to end up with an empty set we need to use both $\{E\}$ and $\{\neg F\}$.

So, the clause $\{\neg F\}$ must be combined with a clause containing F, that must have the form $\{\neg B_i, F\}$ (with B_i an atomic concept), obtaining a clause $\{\neg B_i\}$ that, again, can be combined only with a clause of the form $\{\neg B_j, B_i\}$ (B_j an atomic concept), obtaining $\{\neg B_j\}$, and so on. So, we can use only clauses from $\mathcal{T}_p^+$ in the resolution procedure. Since $\mathcal{T}_p$ is finite, such a procedure has to terminate, and that can happen only if we end up with a clause $\{\neg E\}$ that can be resolved by $\{E\}$, obtaining the empty set. The clauses from $\mathcal{T}_p^+$ that we have used in the resolution procedure correspond to a chain of inclusion axioms in $\mathcal{T}_p^+$ starting from E and ending in F. $\square$

LEMMA 3.4. *Consider a consistent $\mathcal{ALCQI}$ TBox $\mathcal{T}$ composed only of inclusion axioms with the form $C \sqsubseteq D$ or $C \sqsubseteq \neg D$, with C and D atomic concepts. Let $\{\mathcal{T}^+, \mathcal{T}^-\}$ be a partition of $\mathcal{T}$ with $\mathcal{T}^+ = \{C \sqsubseteq D \mid C \sqsubseteq D \in \mathcal{T}\}$ and $\mathcal{T}^- = \{C \sqsubseteq \neg D \mid C \sqsubseteq \neg D \in \mathcal{T}\}$. Consider an axiom $E \sqsubseteq \neg F$. We can derive $\mathcal{T}, E \sqsubseteq \neg F \models C \sqsubseteq \bot$ for some atomic concept C iff $\mathcal{T}^+, E \sqsubseteq \neg F \models E \sqsubseteq \bot$.*

PROOF. *Sketch*. First, we have to prove that $\mathcal{T}, E \sqsubseteq \neg F \models C \sqsubseteq \bot$ for some atomic C only if $\mathcal{T}, E \sqsubseteq \neg F \models E \sqsubseteq \bot$. This can be proven again using the resolution system for propositional logic as in lemma 3.3. $\mathcal{T}_p, E \supset \neg F \models \neg C$ means that adding a clause $\{C\}$ to the set of clauses $\overline{\mathcal{T}_p}, \{\neg E, \neg F\}$ we can end up with the empty set, while we know that $\overline{\mathcal{T}_p}, \{C\}$ does not resolve into an empty set (we have assumed that $\mathcal{T}$ is class-consistent). The combination of $\{C\}$ with the binary clauses in $\overline{\mathcal{T}_p}$ gives back new unary clauses: $\{D\}$ if $\{C\}$ is combined with a clause $\{\neg C, D\}$, or $\{\neg D\}$ if $\{C\}$ is combined with a clause $\{\neg C, \neg D\}$. So, the only way to resolve into an empty set is to have another single clause. Such single clauses can be obtained by only adding $\{\neg E, \neg F\}$ to the set $\overline{\mathcal{T}_p}$, that is, we need to be able to derive from $\overline{\mathcal{T}_p}$ a clause $\{\neg E, F\}$ in order to combine it to $\{\neg E, \neg F\}$ and obtain a unary clause $\{\neg E\}$. It is easy to see that, given the form of the clauses in $\overline{\mathcal{T}}$, we need to derive $\{\neg E\}$ from $\overline{\mathcal{T}_p}, \{\neg E, \neg F\}$ in order to be able to derive from $\overline{\mathcal{T}_p}, \{\neg E, \neg F\}$ unary clauses, that, in turn, is the only way to resolve a clause $\{C\}$ into an empty set.

Now, we can obtain $E \sqsubseteq \bot$ iff $E \sqsubseteq F$ is derivable from $\mathcal{T}$, and, as proved in lemma 3.3, that's the case only if $E \sqsubseteq F$ is derivable from $\mathcal{T}^+$. Hence, $\mathcal{T}, E \sqsubseteq \neg F \models C \sqsubseteq \bot$ for some atomic concept C iff $\mathcal{T}^+, E \sqsubseteq \neg F \models E \sqsubseteq \bot$. □

Now we can prove proposition 3.2.

PROOF. *Sketch.* Consider an $\mathcal{ALCQI}$ knowledge base $\mathcal{T}_\mathcal{D}$ obtained from the translation of a UML diagram $\mathcal{D}$; let $\mathcal{T}_{\mathcal{D},C} \subseteq \mathcal{T}_\mathcal{D}$ be the set of the inclusion axioms in $\mathcal{T}_\mathcal{D}$ modeling the generalizations in the diagram $\mathcal{D}$, *i.e.* $\mathcal{T}_{\mathcal{D},C} = \{C \sqsubseteq D \in \mathcal{T} \mid \{C, D\} \subseteq \mathcal{A}t_C\}$. Since the translation allows sound and complete reasoning with respect to the original UML diagram ([2], section 7.2), we know that every generalization relation between the classes in the diagram (*i.e.* every subsumption relation $E \sqsubseteq F$, with E, F atomic concepts in $\mathcal{A}t_C$) is derivable only if it corresponds to the transitive closure of the generalization relations in the diagram, that is, only if $\mathcal{T}_{\mathcal{D},C} \models E \sqsubseteq F$.

Now, every materialization of a disjointness axiom has the form $\top \sqsubseteq \neg(C \sqcap D)$, that is equivalent to an axiom $C \sqsubseteq \neg D$. That implies that the consistency check of the materializations of our disjointness axioms with respect to $\mathcal{T}_\mathcal{D}$ can be done considering simply $\mathcal{T}_{\mathcal{D},C}$. Now, let $\mathcal{T}_\mathcal{D}$ be the $\mathcal{ALCQI}$ translation of a diagram $\mathcal{D}$ and let $\mathcal{B}$ be a finite set of disjointness axioms. Assume that at least two sets S and S' are in $M_\mathcal{T}^{\mathcal{B}^\sqsubseteq}$. That means that there is at least the materialization of a disjointness axiom $\top \sqsubseteq \neg(C \sqcap D)$ that is in S and not in S', that is, we have $\mathcal{T}_{\mathcal{D},C}, S - \{\top \sqsubseteq \neg(C \sqcap D)\}, \{\top \sqsubseteq \neg(C \sqcap D)\} \not\models \neg E$ and $\mathcal{T}_{\mathcal{D},C}, S', \{\top \sqsubseteq \neg(C \sqcap D)\} \models \neg E$ for some $E \in \mathcal{A}t_C$. By lemma 3.4 we have that corresponds to saying that both $\mathcal{T}_{\mathcal{D},C}, \{\top \sqsubseteq \neg(C \sqcap D)\} \not\models \neg E$ and $\mathcal{T}_{\mathcal{D},C}, \{\top \sqsubseteq \neg(C \sqcap D)\} \models \neg E$ hold at the same time, that is impossible. □

Hence, when we deal with the translation of a UML diagram, applying our procedure we obtain a single maxi-consistent set of disjointness axioms, that implies that the three consequence relations $\models^1_\mathcal{B}$, $\models^2_\mathcal{B}$, and $\models^3_\mathcal{B}$ are identical.

4. COVERING ASSUMPTION

We want to formalize the assumption that every class corresponds to the union of its immediate subclasses; that is, if an object falls under a certain class, it must fall under some of its immediate subclasses that are expressed in the conceptual data model, since we consider that our ontology is completely defined by such a model. For example, in the diagram in figure 6, we want to impose that each instance of the class *Employees* must be an instance of the class *Permanent Employees* or of the class *Contractors*.

In our *TBox* $\mathcal{T}_\mathcal{D}$ we have the axioms $PE \sqsubseteq E$ and $C \sqsubseteq E$, and to satisfy the covering assumption we need to introduce in the *TBox* the necessary pieces of information to derive $E = PE \sqcup C$. To do so, it is sufficient to follow a simple procedure, that transforms the original *TBox* $\mathcal{T}_\mathcal{D}$ into a new *TBox* $\mathcal{T}'_\mathcal{D}$ s.t. $\mathcal{T}_\mathcal{D} \subseteq \mathcal{T}'_\mathcal{D}$. The procedure is the same, whether we are working with a $\mathcal{SROIQ}$ knowlege base or with the translation of a diagram.

- Consider the set of atomic concepts $\mathcal{A}t_C = \{C_1, \ldots C_n\}$ corresponding to the classes in the diagram $\mathcal{D}$. For every atomic concept C_i $(1 \leq i \leq n)$, define a set of atomic concepts Sub_i in the following way:

  - $Sub_i^0 = \emptyset$;

  - for $1 \leq j \leq n$, $Sub_i^j =$

$$= \begin{cases} Sub_i^j = Sub_i^{j-1} & if\ j = i \\ Sub_i^j = Sub_i^{j-1} \cup \{C_j\} & if\ j \neq i\ and\ C_j \sqsubseteq C_i \in \mathcal{T}_\mathcal{D} \\ Sub_i^j = Sub_i^{j-1} & else. \end{cases}$$

- Let $Sub_i = Sub_i^n$;
- For every atomic concept C_i, add to the *TBox* $\mathcal{T}$ the axiom $\models C_i \sqsubseteq \bigsqcup Sub_i$.

EXAMPLE 4.1. *Consider again figure 6. Applying the above procedure, we end up with* $Sub_{RG} = Sub_P = Sub_{PD} = Sub_G = \emptyset$, $Sub_E = \{PE, C\}$, $Sub_{PE} = \{P, PD\}$, $Sub_C = \{PD, G\}$. *That is, now we have a new* TBox $\mathcal{T}' = \mathcal{T} \cup \{E \sqsubseteq (PE \sqcup C), C \sqsubseteq (PD \sqcup G), PE \sqsubseteq (PD \sqcup P)\}$. *Note that, using other pieces of information contained in the* TBox *(*$C \sqsubseteq A$, $D \sqsubseteq A$, $E \sqsubseteq C$, $F \sqsubseteq C$*), the following desired equivalences are derivable from our* TBox*:* $\models E = (PE \sqcup C)$, $\models C = (PD \sqcup G)$, *and* $\models PE = (PD \sqcup P)$.

If the *TBox* were classified, as we are assuming in the case of a $\mathcal{SROIQ}$ knowledge base, we would have had also the axioms $P \sqsubseteq E$, $PD \sqsubseteq E$, and $G \sqsubseteq E$, that would have implied the introduction of a covering axiom $E \sqsubseteq (PE \sqcup C \sqcup P \sqcup PD \sqcup G)$. Since we have in the knowledge base that $P \sqsubseteq PE$, $PD \sqsubseteq PE$, $PD \sqsubseteq C$, and $G \sqsubseteq C$, from the *TBox* we could again derive the covering axiom $\models E = (PE \sqcup C)$, identifying E with the disjointness of its *immediate* subclasses PE and C.

5. CONCLUSIONS

In the present paper we have presented two procedures that are appropriate for the formalization of two assumptions that are typically done about UML class diagrams and analogous formalizations of conceptual models: the disjointness assumption and the covering assumption. Both the procedures proposed are simple and easily implementable, since both of them rely on the underlying monotonic consequence relations characterizing the DL we are working with.

6. REFERENCES

[1] C. E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.*, 50(2):510–530, 1985.

[2] D. Berardi, D. Calvanese, and G. De Giacomo. Reasoning on UML class diagrams. *Artif. Intell.*, 168(1):70–118, Oct. 2005.

[3] P. P.-S. Chen. The entity-relationship model – Toward a unified view of data. *ACM Trans. Database Syst.*, 1(1):9–36, Mar. 1976.

[4] T. Halpin. Object-role modeling (ORM/NIAM). In *Handbook on Architectures of Information Systems*, pages 81–102. Springer-Verlag, 1998.

[5] I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible SROIQ. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, pages 57–67, Menlo Park, CA, USA, 2006. AAAI Press.

[6] I. Horrocks, P. F. Patel-Schneider, and F. V. Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1:2003, 2003.

[7] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intell.*, 44(1-2):167–207, 1990.

[8] D. J. Lehmann. Another perspective on default reasoning. *Ann. Math. Artif. Intell.*, 15(1):61–82, 1995.

[9] D. Poole. A logical framework for default reasoning. *Artif. Intell.*, 36(1):27–47, 1988.

[10] J. Rumbaugh, I. Jacobson, and G. Booch. *Unified Modeling Language Reference Manual, The (2nd Edition)*. Pearson Higher Education, 2004.

[11] W3C. *OWL 2 Web Ontology Language Document Overview, http://www.w3.org/TR/2009/REC-owl2-overview-20091027/, 2009*.