Defeasible reasoning in ORM2

Giovanni Casini[1] and Alessandro Mosca[2]

[1] Centre for Artificial Intelligence Research, CSIR Meraka Institute and UKZN, South Africa
Email: GCasini@csir.co.za
[2] Free University of Bozen-Bolzano, Faculty of Computer Science, Italy
Email: mosca@inf.unibz.it

Abstract. The *Object Role Modeling* language (ORM2) is one of the main conceptual modeling languages. Recently, it has been proposed a translation of a main fragment of ORM2 (ORM2$^{\mathsf{zero}}$) into the description logic $\mathcal{ALCQI}$, allowing the use of logical instruments in the analysis of ORM schemas. On the other hand, in many ontological domains there is a need for the formalization of *defeasible information* and of *nonmonotonic* forms of reasoning. Here we introduce two new constraints in the ORM2 language, in order to formalize defeasible information into the schemas, and we explain how to translate such defeasible information in $\mathcal{ALCQI}$.

1 Introduction

ORM2 ('Object Role Modelling 2') is a graphical fact-oriented approach for modelling, transforming, and querying business domain information, which allows for a verbalisation in language readily understandable by non-technical users [1]. ORM2 is at the core of the OGM standard SBVR language ('Semantics of Business Vocabulary and Business Rules'), and of the conceptual modelling language for database design in Microsoft Visual Studio (VS. In particular, the Neumont ORM Architect (NORMA) tool is an open source plug-in to VS providing the most complete support for the ORM2 notation.

On the other hand, in the more general field of formal ontologies in the last years a lot of attention has been dedicated to the implementations of forms of *defeasible reasoning*, and various proposals, such as [2,3,4,5,6,7,8], have been made in order to integrate nonmonotonic reasoning mechanisms into DLs.

In what follows we propose an extension of ORM2 with two new formal constraints, with the main aim of integrating a form of defeasible reasoning in the ORM2 schemas; we explain how to translate such enriched ORM2 schemas into $\mathcal{ALCQI}$ knowledge bases, and how to use them to check the schema consistency and draw conclusions. In particular, the paper presents a procedure to implement a particular construction in nonmonotonic reasoning, *i.e.* Lehmann and Magidor's *Rational Closure* (LMRC)[9], that is known for being characterized by good logical properties and for giving back intuitive results.

2 Fact-oriented modelling in ORM2

'Fact-oriented modelling' began in the early Seventies as a conceptual modelling approach that views the world in terms of simple facts about individuals and the roles they play [1]. *Facts* are assertions that are taken to be true in the domain of interest about

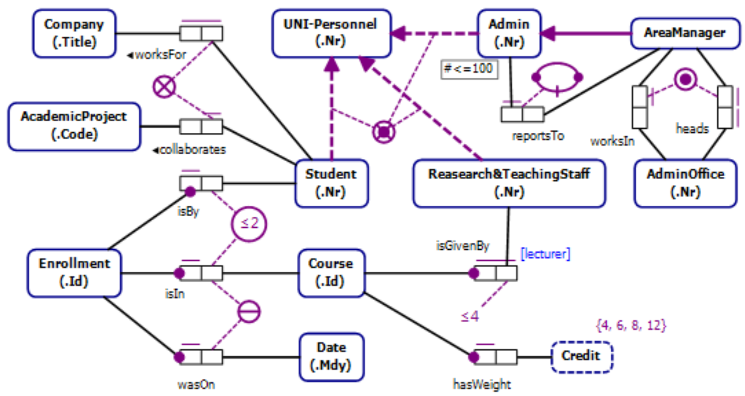

Fig. 1. A conceptual schema including an instantiation of most of the ORM2 constraints.

objects playing certain *roles* (e.g. 'Alice is enrolled in the Computer Science program'). In ORM2 one has **entities** (e.g. a person or a car) and **values** (e.g. a *character string* or a *number*). Moreover, entities and values are described in terms of the **types** they belong to, where a type (e.g. House, Car) is a set of instances. Each entity in the domain of interest is, therefore, an instance of a particular type. The roles played by the entities in a given domain are introduced by means of logical **predicates**, and each predicate has a given set of **roles** according to its arity. Each role is connected to exactly one object type, indicating that the role is played only by the (possible) instances of that type (notice that, unlike ER, ORM2 makes no use of 'attributes'). ORM2 also admits the possibility of making an object type out of a relationship. Once a relation has been transformed into an object type, this last is called the **objectification** of the relation.

According to the ORM2 design procedure, after the specification of the relevant **object types** (i.e. entity and value types) and predicates, the static *constraints* must be considered. The rest of this section is devoted to an informal introduction of the constraint graphical representation, together with their intended semantics. Fig. 1 shows an example of ORM2 conceptual schema modelling the 'academic domain' (where the soft rectangles are entity types, the dashed soft rectangles are value types, and the sequences of one or more role-boxes are predicates). The example is not complete w.r.t. the set of all the ORM2 constraints but it aims at giving the feeling of the expressive power of the language. The following are among the constraints included in the schema:

1. **Subtyping** (depicted as thick solid and dashed arrows) representing 'is-a' relationships among types. A **partition**, made of a combination of an **exclusive** constraint (a circled 'X' saying that 'Research&TeachingStaff, Admin, Student' are *mutually disjoint*), and a **total** constraint (a circled dot for 'Research&TeachingStaff, Admin, Student completely *cover* their common super-type').
2. An **internal frequency occurrence** saying that *if* an instance of Research&TeachingStaff plays the role of being lecturer in the relation isGivenBy, that instance can play the role at most 4 times. A frequency occurrence may span over more than one role, and suitable frequency *ranges* can be specified. *At most one* cardinalities (depicted as continuos bars) are special cases of frequency occurrence called **internal uniqueness** constraints.
3. An **external frequency occurrence** applied to the roles played by Student and Course, meaning that 'Students are allowed to enrol in the same course *at most twice*'.

4. An **external uniqueness** constraint between the role played by Course in isIn and the role played by Date in wasOn, saying that 'For each combination of Course and Date, *at most one* Enrollment isIn that Course and wasOn that Date'.
5. A **mandatory participation** constraints (graphically represented by a dot), among several other, saying that 'Each Course is given by *at least one* instance of the Research&TeachingStaff type' (combinations of mandatory and uniqueness translate into *exaclty one* cardinality constraints).
6. A disjunctive mandatory participation, called **inclusive-or** constraint (depicted as a circled dot), linking the two roles played by the instances of AreaManager meaning that 'Each area manager *either* works in *or* heads (or *both*)'.
7. An **object cardinality** constraint forcing the number of the Admin instances to be less or equal to 100 (**role cardinality** constraints, applied to role instances, are also part of ORM2).
8. An **object type value** constraint indicating which values are allowed in Credit (**role value** constraints can be also expressed to indicate which values are allowed to play a given role).
9. An **exclusion** constraint (depicted as circled 'X') between the two roles played by the instances of Student, expressing the fact that no student can play *both* these roles. Exclusion constraint can also span over arbitrary sequences of roles. The combination of exclusion and inclusive-or constraints gives rise to **exclusive-or** constraints meaning that each instance in the attached entity type plays *exactly one* of the attached roles. Exclusion constraints, together with **subset** and **equality**, are called *set-comparison* constraints.
10. A **ring** constraint expressing that the relation reportsTo is *asymmetric*.

A comprehensive list of all the ORM2 constraints, together with their graphical representation, can be found in [1].

3 The $\mathcal{ALCQI}$ encoding of ORM2$^{\mathsf{zero}}$

With the main aim of relying on available tools to reason in an effective way on ORM2 schemas, an encoding in the description logic $\mathcal{ALCQI}$ for which tableaux-based reasoning algorithms with a tractable computational complexity have been developed [10]. $\mathcal{ALCQI}$ corresponds to the basic DL $\mathcal{ALC}$ equipped with *qualified cardinality restrictions* and *inverse roles*, and it is a fragment of the OWL2 web ontology language (a complete introduction of the syntax and semantics of $\mathcal{ALCQI}$ can be found in [11]). We also introduce in the $\mathcal{ALCQI}$ language the expression '$C \supset D$' as an abbreviation for the expression '$\neg C \sqcup D$'.

Now, the discrepancy between ORM2 and $\mathcal{ALCQI}$ poses two main obstacles that need to be faced in order to provide the encoding. The first one, caused by the absence of n-ary relations in $\mathcal{ALCQI}$, is overcome by means of *reification*: for each relation R of arity $n \geq 2$, a new atomic concept A_R and n functional roles $\tau(R.a_1), \ldots, \tau(R.a_n)$ are introduced. The tree-model property of $\mathcal{ALCQI}$ guarantees the *correctness* encoding w.r.t. the reasoning services over ORM2. Unfortunately, the second obstacle fixes, once for all, the limits of the encoding: $\mathcal{ALCQI}$ does not admit neither arbitrary set-comparison assertions on relations, nor external uniqueness or uniqueness involving more than one role, or arbitrary frequency occurrence constraints. In other terms, it can be proved that $\mathcal{ALCQI}$ is strictly contained in ORM2. The analysis of this inclusion thus led to identification of the fragment called ORM2$^{\mathsf{zero}}$ which is maximal with respect to the expressiveness of $\mathcal{ALCQI}$, and still expressive enough to capture the most frequent usage patterns of the conceptual modelling community. Let ORM2$^{\mathsf{zero}}$ $=$ $\{\mathsf{TYPE}, \mathsf{FREQ}^-, \mathsf{MAND}, \mathsf{R\text{-}SET}^-, \mathsf{O\text{-}SET}_{\mathsf{Isa}}, \mathsf{O\text{-}SET}_{\mathsf{Tot}}, \mathsf{O\text{-}SET}_{\mathsf{Ex}}, \mathsf{OBJ}\}$ be the

fragment of ORM2 where: (i) FREQ^- can only be applied to single roles, and (ii) $\mathsf{R\text{-}SET}^-$ applies either to entire relations of the same arity or to two single roles. The encoding of the semantics of ORM2$^{\mathsf{zero}}$ shown in table 4 is based on the $\mathcal{S}^{\mathcal{ALCQI}}$ signature made of: (i) A set $E_1, E_2, \ldots, E_n$ of concepts for *entity types*; (ii) a set $V_1, V_2, \ldots, V_m$ of concepts for *value types*; (iii) a set $A_{R_1}, A_{R_2}, \ldots, A_{R_k}$ of concepts for objectified *n*-ary *relations*; (iv) a set $D_1, D_2, \ldots, D_l$ of concepts for *domain symbols*; (v) $1, 2, \ldots, n_{max}+1$ roles. Additional *background axioms* are needed here in order to: (i) force the interpretation of the $\mathcal{ALCQI}$ knowledge base to be correct w.r.t. the corresponding ORM2 schema, and (ii) guarantee that that any model of the resulting $\mathcal{ALCQI}$ can be 'un-reified' into a model of original ORM2$^{\mathsf{zero}}$ schema.
The correctness of the introduced encoding is guaranteed by the following theorem (whose complete proof is available at [12]):

Theorem 1. *Let Σ^{zero} be an* ORM2$^{\mathsf{zero}}$ *conceptual schema and $\Sigma^{\mathcal{ALCQI}}$ the $\mathcal{ALCQI}$ KB constructed as described above. Then an object type O is consistent in Σ^{zero} if and only if the corresponding concept O is satisfiable w.r.t. $\Sigma^{\mathcal{ALCQI}}$.*

Let us conclude this section with some observation about the complexity of reasoning on ORM2 conceptual schemas, and and taking into account that all the reasoning tasks for a conceptual schema can be reduced to object type consistency. Undecidability of the ORM2 object type consistency problem can be proved by showing that arbitrary combinations of subset constraints between n-ary relations and uniqueness constraints over single roles are allowed [13]. As for ORM2$^{\mathsf{zero}}$, one can conclude that object type consistency is EXPTIME-complete: the upper bound is established by reducing the ORM2$^{\mathsf{zero}}$ problem to concept satisfiability w.r.t. $\mathcal{ALCQI}$ KBs (which is known to be EXPTIME-hard) [14], the lower bound by reducing concept satisfiability w.r.t. $\mathcal{ALC}$ KBs (which is known to be EXPTIME-complete) to object consistency w.r.t. ORM2$^{\mathsf{zero}}$ schemas [15]. Therefore, we obtain the following result:

Theorem 2. *Reasoning over* ORM2$^{\mathsf{zero}}$ *schemas is* EXPTIME*-complete.*

4 Rational Closure in $\mathcal{ALCQI}$

Now we briefly present the procedure to define the analogous of LMRC for the DL language $\mathcal{ALCQI}$. A more extensive presentation of such a procedure can be found in [4]: it is defined for $\mathcal{ALC}$, but it can be applied to $\mathcal{ALCQI}$ without any modifications. LMRC is one of the main construction in the field of nonmonotonic logics, since it has a solid logical characterization, it maintains the same complexity level of the underlying monotonic logic, and it does not give back counter-intuitive conclusions; its main drawback is in its inferential weakness, since there could be desirable conclusions that we won't be able to draw (see example 2 below).

As seen above, each ORM2$^{\mathsf{zero}}$schema can be translated into an $\mathcal{ALCQI}$ TBox. A TBox $\mathcal{T}$ for $\mathcal{ALCQI}$ consists of a finite set of general inclusion axioms (GCIs) of form $C \sqsubseteq D$, with C and D concepts. Now we introduce also a new form of information, the *defeaible inclusion axioms* $C \sqsubseteq\!\!\!\sim D$, that are read as 'Typically, an individual falling under the concept C falls also under the concept D'. We indicate with $\mathcal{B}$ the finite set of such inclusion axioms.

Table 1. $\mathcal{ALCQI}$ encoding.

Background domain axioms:	$E_i \sqsubseteq \neg(D_1 \sqcup \cdots \sqcup D_l)$ for $i \in \{1, \ldots, n\}$ $V_i \sqsubseteq D_j$ for $i \in \{1, \ldots, m\}$, and some j with $1 \leq j \leq l$ $D_i \sqsubseteq \sqcap_{j=i+1}^{l} \neg D_j$ for $i \in \{1, \ldots, l\}$ $\top \sqsubseteq A_{\top_1} \sqcup \cdots \sqcup A_{\top_{n_{max}}}$ $\top \sqsubseteq (\leq 1 i.\top)$ for $i \in \{1, \ldots, n_{max}\}$ $\forall i.\bot \sqsubseteq \forall i+1.\bot$ for $i \in \{1, \ldots, n_{max}\}$ $A_{\top_n} \equiv \exists 1.A_{\top_1} \sqcap \cdots \sqcap \exists n.A_{\top_1} \sqcap \forall n+1.\bot$ for $n \in \{2, \ldots, n_{max}\}$ $A_R \sqsubseteq A_{\top_n}$ for each atomic relation *R* of arity *n* $A \sqsubseteq A_{\top_1}$ for each atomic concept *A*
TYPE$(R.a, O)$	$\exists \tau(R.a)^{-}.A_R \sqsubseteq O$
FREQ$^{-}(R.a, \langle \text{min}, \text{max} \rangle)$	$\exists \tau(R.a)^{-}.A_R \sqsubseteq \geq \text{min}\ \tau(R.a)^{-}.A_R \sqcap \leq \text{max}\ \tau(R.a)^{-}.A_R$
MAND$(\{R^1.a_1, \ldots, R^1.a_n,$ $\ldots, R^k.a_1, \ldots, R^k.a_m\}, O)$	$O \sqsubseteq \exists \tau(R^1.a_1)^{-}.A_{R^1} \sqcup \cdots \sqcup \exists \tau(R^1.a_n)^{-}.A_{R^1} \sqcup \cdots \sqcup$ $\exists \tau(R^k.a_1)^{-}.A_{R^k} \sqcup \cdots \sqcup \exists \tau(R^k.a_m)^{-}.A_{R^k}$
$^{(A)}$ R-SET$^{-}_{\text{Sub}}(A, B)$ $^{(A)}$ R-SET$^{-}_{\text{Exc}}(A, B)$	$A_R \sqsubseteq A_S$ $\quad {}^{(A)}\ A = \{R.a_1, \ldots, R.a_n\}, B = \{S.b_1, \ldots, S.b_n\}$ $A_R \sqsubseteq A_{\top_n} \sqcap \neg A_S$
$^{(B)}$ R-SET$^{-}_{\text{Sub}}(A, B)$ $^{(B)}$ R-SET$^{-}_{\text{Exc}}(A, B)$	$\exists \tau(R.a_i)^{-}.A_R \sqsubseteq \exists \tau(S.b_j)^{-}.A_S$ $\quad {}^{(B)}\ A = \{R.a_i\}, B = \{S.b_j\}$ $\exists \tau(R.a_i)^{-}.A_R \sqsubseteq A_{\top_n} \sqcap \neg \exists \tau(S.b_j).A_S$
O-SET$_{\text{Isa}}(\{O_1, \ldots, O_n\}, O)$	$O_1 \sqcup \cdots \sqcup O_n \sqsubseteq O$
O-SET$_{\text{Tot}}(\{O_1, \ldots, O_n\}, O)$	$O \sqsubseteq O_1 \sqcup \cdots \sqcup O_n$
O-SET$_{\text{Ex}}(\{O_1, \ldots, O_n\}, O)$	$O_1 \sqcup \cdots \sqcup O_n \sqsubseteq O$ and $O_i \sqsubseteq \sqcap_{j=i+1}^{n} \neg O_j$ for each $i = 1, \ldots, n$
OBJ(R, O)	$O \equiv A_R$

Example 1. Consider a modification of the classical 'penguin example', with the concepts P, B, F, I, Fi, W respectively read as 'penguin', 'bird', 'flying', 'insect', 'fish', and 'have wings', and a role $Prey$, where a role instantiation $(a, b){:}Prey$ read as 'a preys for b'. We can define a defeasible KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$ with $\mathcal{T} = \{P \sqsubseteq B, I \sqsubseteq \neg Fi\}$ and $\mathcal{B} = \{P \mathrel{\tilde{\sqsubseteq}} \neg F,\ B \mathrel{\tilde{\sqsubseteq}} F,\ P \mathrel{\tilde{\sqsubseteq}} \forall Prey.Fi, B \mathrel{\tilde{\sqsubseteq}} \forall Prey.I, B \mathrel{\tilde{\sqsubseteq}} W\}$.

In order to define the rational closure of a knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$, we have first of all to transform the knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$ into a new knowledge base $\langle \Phi, \Delta \rangle$, s.t. while $\mathcal{T}$ and $\mathcal{B}$ are sets of inclusion axioms, Φ and Δ are simply sets of concepts. Then, we shall use the sets $\langle \Phi, \Delta \rangle$ to define a nonmonotonic consequence relation that models the rational closure. Here we just present the procedure, referring to [4] for a more in-depth explanation of the various steps.

Transformation of $\langle \mathcal{T}, \mathcal{B} \rangle$ into $\langle \Phi, \Delta \rangle$. Starting with $\langle \mathcal{T}, \mathcal{B} \rangle$, we apply the following steps.

Step 1. Define the set representing the *strict form* of the set $\mathcal{B}$, *i.e.* the set $\mathcal{B}^{\sqsubseteq} = \{C \sqsubseteq D \mid C \mathrel{\tilde{\sqsubseteq}} D \in \mathcal{B}\}$, and define a set $\mathfrak{A}_{\mathcal{B}}$ as the set of the antecedents of the conditionals in $\mathcal{B}$, *i.e.* $\mathfrak{A}_{\mathcal{B}} = \{C \mid C \mathrel{\tilde{\sqsubseteq}} D \in \mathcal{B}\}$.

Step 2. We determine an *exceptionality ranking* of the sequents in $\mathcal{B}$ using the set of the antecedents $\mathfrak{A}_{\mathcal{B}}$ and the set $\mathcal{B}^{\sqsubseteq}$.

Step 2.1. A concept is considered *exceptional* in a knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$ only if it is classically negated (*i.e.* we are forced to consider it empty), that is, C is exceptional in $\langle \mathcal{T}, \mathcal{B} \rangle$ only if

$$\mathcal{T} \cup \mathcal{B}^{\sqsubseteq} \models \top \sqsubseteq \neg C$$

where $\models$ is the classical consequence relation associated to $\mathcal{ALCQI}$. If a concept is considered exceptional in $\langle \mathcal{T}, \mathcal{B} \rangle$, also all the defeasible inclusion axioms in $\mathcal{B}$ that have such a concept as antecedent are considered exceptional. So, given a knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$ we can check which of the concepts in $\mathfrak{A}_{\mathcal{B}}$ are exceptional (we indicate the set containing them as $E(\mathfrak{A}_{\mathcal{B}})$), and consequently which of the axioms in $\mathcal{B}$ are exceptional (the set $E(\mathcal{B}) = \{C \mathrel{\tilde{\sqsubseteq}} D \mid C \in E(\mathfrak{A}_{\mathcal{B}})\}$).

So, given a knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$ we can construct iteratively a sequence $\mathcal{E}_0, \mathcal{E}_1, \ldots$ of subsets of $\mathcal{B}$ in the following way:

- $\mathcal{E}_0 = \mathcal{B}$
- $\mathcal{E}_{i+1} = E(\mathcal{E}_i)$

Since $\mathcal{B}$ is a finite set, the construction will terminate with an empty set ($\mathcal{E}_n = \emptyset$ for some n) or a fixed point of E.

Step 2.2 Using such a sequence, we can define a ranking function r that associates to every axiom in $\mathcal{B}$ a number, representing its level of exceptionality:

$$r(C \mathrel{\tilde{\sqsubseteq}} D) = \begin{cases} i & \text{if } C \mathrel{\tilde{\sqsubseteq}} D \in \mathcal{E}_i \text{ and } C \mathrel{\tilde{\sqsubseteq}} D \notin \mathcal{E}_{i+1} \\ \infty & \text{if } C \mathrel{\tilde{\sqsubseteq}} D \in \mathcal{E}_i \text{ for every } i \,. \end{cases}$$

Here we shall assume that the procedure terminates into an empty set, *i.e.* every concept has a finite ranking value, and we shall deal with the occurrence of of some concept with ∞ as ranking value in the following section.

Step 3. Now we build a new formalization of the information contained in the knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$, translating each of the two sets of axioms into two sets of concepts, Φ and Δ respectively. The set Φ will simply correspond to the *materialization* of the inclusion axioms, *i.e.* the concepts translating the axioms.

$$\Phi = \{C \supset D \mid C \sqsubseteq D \in \mathcal{T}\}$$

In order to define the set Δ, given the rank value of the sequents in $\mathcal{B}$, we construct a set of *default concepts* $\Delta = \{\delta_0, \ldots, \delta_n\}$ (with n the highest rank-value in $\mathcal{B}$), with

$$\delta_i = \bigsqcap \{C \supset D \mid C \mathrel{\tilde{\sqsubseteq}} D \in \mathcal{B} \text{ and } r(C \mathrel{\tilde{\sqsubseteq}} D) \geq i\} \,.$$

Hence we substitute the conceptual system $\langle \mathcal{T}, \mathcal{B} \rangle$ with the pair $\langle \Phi, \Delta \rangle$, where Φ and Δ are sets of concepts, the former containing concepts to be considered valid for every individual of the domain, the latter containing concepts to be considered *defeasibly* valid, *i.e.* we apply such default concepts to an individual only if they are consistent with the information in our knowledge base. It is not difficult to see that the concepts in Δ are linearly ordered by $\models$, that is, for every δ_i, $0 \leq i < n$, $\models \delta_i \sqsubseteq \delta_{i+1}$.

Rational Closure. Consider now $\Phi = \{C_1 \supset D_1, \ldots, C_m \supset D_m\}$ and $\Delta = \{\delta_0, \ldots, \delta_n\}$. We define a nonmonotonic consequence relation between the concepts $\mid\!\sim_{\langle \Phi, \Delta \rangle}$ that determines what presumably follows from a finite set of concepts Γ. Simply, a concept D is a defeasible consequence of Γ if it classically follows from Γ, the strict information contained in the knowledge base (*i.e.* Φ), and the first default concept δ_i that in the sequence $\langle \delta_0, \ldots, \delta_n \rangle$ results classically consistent with the rest of the premises.

Definition 1. *$\Gamma \mid\!\sim_{\langle \Phi, \Delta \rangle} D$ iff $\models \bigsqcap \Gamma \sqcap \bigsqcap \Phi \sqcap \delta_i \sqsubseteq D$, where δ_i is the first $(\Gamma \cup \Phi)$-consistent formula* [3] *of the sequence $\langle \delta_0, \ldots, \delta_n \rangle$.*

[3] That is, $\not\models \bigsqcap \Phi \sqcap \bigsqcap \Gamma \sqsubseteq \neg \delta_i$.

You can find in [4] an explanation of why the above procedure for DL corresponds to the rational closure defined by Lehmann and Magidor for propositional languages, and satisfies the DL translation of the basic properties characterizing rational consequence relations.

Proposition 1 ([4], Proposition 4). *$\mid\!\sim_{\langle \Phi, \Delta \rangle}$ is a consequence relation containing $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$ and satisfying the properties of the rational consequence relations.*

Moreover, as deciding entailment in $\mathcal{ALCQI}$ is EXPTIME-complete (see Theorem 2), and since the decidability problem for the rational closure is reducible to a finite number of decision w.r.t. the classical $\mathcal{ALCQI}$ consequence relation, we obtain immediately that

Proposition 2. *Deciding $C \mid\!\sim_{\langle \widetilde{\mathcal{T}}, \widetilde{\Delta} \rangle} D$ in $\mathcal{ALC}$ is an EXPTIME-complete problem.*

Example 2. Consider the KB of Example 1. Hence, we start with $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$. The strict form of $\mathcal{B}$ is $\mathcal{B}^{\sqsubseteq} = \{P \sqsubseteq \neg F, B \sqsubseteq F, P \sqsubseteq \forall Prey.Fi, B \sqsubseteq \forall Prey.I, B \sqsubseteq W\}$, with $\mathfrak{A}_{\mathcal{B}} = \{P, B\}$. Following the procedure at **Step 2**, we obtain the exceptionality ranking of the sequents: $\mathcal{E}_0 = \{P \sqsubseteq \neg F, B \sqsubseteq F, P \sqsubseteq \forall Prey.Fi, B \sqsubseteq \forall Prey.I, B \sqsubseteq W\}$; $\mathcal{E}_1 = \{P \sqsubseteq \neg F, P \sqsubseteq \forall Prey.Fi\}$; $\mathcal{E}_2 = \emptyset$. Automatically, we have the ranking values of every sequent in $\mathcal{B}$: namely, $r(B \sqsubseteq F) = r(B \sqsubseteq \forall Prey.I) = r(B \sqsubseteq W) = 0$; $r(P \mathrel{\tilde{\sqsubseteq}} \neg F) = r(P \mathrel{\tilde{\sqsubseteq}} \forall Prey.Fi) = 1$. From such a ranking, we obtain a set of default concepts $\Delta = \{\delta_0, \delta_1\}$, with

$$\delta_0 = (B \supset F) \sqcap (B \supset \forall Prey.I) \sqcap (P \supset \neg F) \sqcap (P \supset \forall Prey.Fi) \sqcap (B \supset W)$$
$$\delta_1 = (P \supset \neg F) \sqcap (P \supset \forall Prey.Fi)\ .$$

Now, referring to definition 1, we can derive a series of desirable conclusions, as $\neg F \mid\!\sim \neg B$, , $B \wedge green \mid\!\sim F$, $P \wedge black \mid\!\sim \neg F$, $P \mid\!\sim \forall Prey.\neg I$. Instead, other counterintuitive connections are not valid, such as $B \wedge \neg F \mid\!\sim P$, $B \wedge \neg F \mid\!\sim \neg P$, or $P \mid\!\sim F$. Here we can notice the main weakness of the Rational Closure: even if it would be intuitive to conclude that penguins have wings ($P \mid\!\sim W$), we cannot conclude that a class that results atypical (as penguins) cannot inherit *any* of the typical properties of its superclasses (as having wings), even if such properties are not logically connected to the ones that determine the exceptionality (not flying and eating fishes).

5 Defeasible constraints for ORM2

As seen above, in order to introduce defeasible reasoning in DL we introduce the *defeasible inclusion axiom* $C \mathrel{\tilde{\sqsubseteq}} D$, indicating that the elements of the concept C *typically*, but not necessarily, are elements of the concept D. We want to introduce in the ORM2$^{\text{zero}}$schemas constraints playing an analogous role, *i.e.* representing defeasible constraints in the ontological organization of a particular domain. With this goal in mind, two constraints aimed at representing forms of defeasible constraints between classes, and classes and their properties, are introduced.

– A *defeasible subclass relation*: we introduce an arrow '$\rightsquigarrow$', where '$C \rightsquigarrow D$' indicates that each element of the class C is also an element of the class D, if not informed of the contrary.

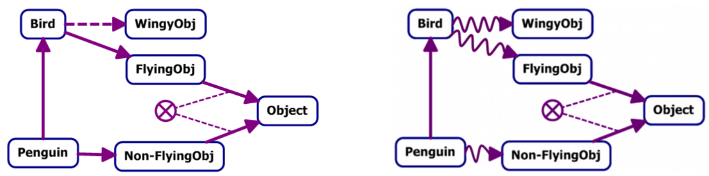

Fig. 2. Example 3, strict (left) and defeasible (right) version.

Table 2. $\mathcal{ALCQI}$ encoding of the ORM2$^{\text{zero}}$ 'Non-Flying Birds' example.

Signature:	$\mathsf{Bird}, \mathsf{Penguin}, \mathsf{WingyObject},$ $\mathsf{FlyingObject}, {}^{\mathsf{Non}}\mathsf{FlyingObject}, \mathsf{Object}$
Subtyping Constraints:	$\mathsf{FlyingObject} \sqcup {}^{\mathsf{Non}}\mathsf{FlyingObject} \sqsubseteq \mathsf{Object}$ $\mathsf{FlyingObject} \sqsubseteq \neg {}^{\mathsf{Non}}\mathsf{FlyingObject}$ $\mathsf{Penguin} \sqsubseteq \mathsf{Bird}, \mathsf{Penguin} \sqsubseteq {}^{\mathsf{Non}}\mathsf{FlyingObject}$ $\mathsf{Bird} \sqsubseteq \mathsf{WingyObject}, \mathsf{Bird} \sqsubseteq \mathsf{FlyingObject}$

Example 3 (Defeasible subclass relation). Consider figure 2. The schema on the left represents in ORM2 the classic penguin example: penguins are birds and do not fly (the class Penguin is a subclass, respectively, of the classes Birds and Non-FlyingObj), while birds fly and have wings (the class Birds is a subclass, respectively, of the classes FlyingObj and WingyObj). The translation procedure into $\mathcal{ALCQI}$gives back the TBox $\mathcal{T}$ in table 2. From $\mathcal{T}$ we can derive that the schema is inconsistent, since we have $\mathcal{T} \models \neg\mathsf{Penguin}$, *i.e.* the concept Penguin must be empty. We can modify the knowledge base introducing defeasible information, in particular stating that birds *typically* fly and *typically* have wings, and penguins *typically* do not fly. In this way we obtain the schema on the right, and in $\mathcal{ALCQI}$ we obtain a set $\mathcal{B} = \{\mathsf{Bird} \mathrel{\tilde{\sqsubseteq}} \mathsf{WingyObject}, \mathsf{Bird} \mathrel{\tilde{\sqsubseteq}} \mathsf{FlyingObject}, \mathsf{Penguin} \mathrel{\tilde{\sqsubseteq}} \mathsf{NonFlyingObject}\}$, substituting the corresponding classical inclusion axioms in the TBox.

– A *defeasible mandatory participation*: we introduce a new mandatory participation constraint '$\square$', to use instead of the classically mandatory constraint '$\bullet$'. If the connection between a class C and a relation R is constrained by an constraint $\square$, we read it as 'each element of the class C participates to the relation R, if we are not informed of the contrary'.

Example 4 (Defeasible mandatory participation). Consider figure 3. The schema represents the organization of a firm: the class Manager is a subclass of the class Employee, and every employee *must* work for a project. while every project must have at least an employee working on it. The class Manager is partitioned into AreaManager and TopManager. Each top managers mandatorily manages a project. The translation procedure into $\mathcal{ALCQI}$ of the left version of the schema gives back the TBox $\mathcal{T}$ in table 3. Since managing and working for a project are not compatible roles, $\mathcal{T}$ implies that the class TopManager is empty, since a top manager would manage and would work for a project at the same time. Instead, if we declare that *typically* an employee works for a project, we can consider the top managers as exceptional kind of employees; hence we substitute the mandatory constraint between Employee and WorkFor with a defeasible constraint (*i.e.* the schema on the right in figure 3); from such

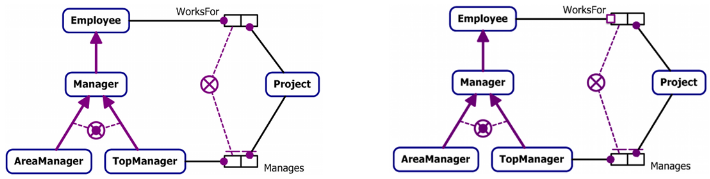

Fig. 3. Example 4, strict (left) and defeasible (right) version.

Table 3. $\mathcal{ALCQI}$ encoding of the ORM2$^{\text{zero}}$ 'Non-Managing Employees' example. Notice that: (i) according to the introduced encoding, the relations $\mathsf{WorksFor}$, $\mathsf{Manages}$ have been reified into atomic concepts, and (ii) for the sake of clarity, we write $\exists \mathsf{R}^-.\mathsf{C} \sqsubseteq \mathsf{D}$ instead of $\mathsf{C} \sqsubseteq \exists \mathsf{R}^-.\mathsf{D}$).

Signature:	$\mathsf{Employee}, \mathsf{Manager}, \mathsf{AreaManager}, \mathsf{TopManager}, \mathsf{Project},$ $\mathsf{WorksFor}, \mathsf{Manages}, \mathsf{A}_{\top 1}, \mathsf{A}_{\top 2}$ $\mathsf{f1}, \mathsf{f2}, \mathsf{f3}$
Background axioms:	$\top \sqsubseteq \mathsf{A}_{\top 1} \sqcup \mathsf{A}_{\top 2}$ $\top \sqsubseteq (\leq 1 \mathsf{f1}.\top), \quad \top \sqsubseteq (\leq 1 \mathsf{f2}.\top)$ $\forall \mathsf{f1}.\bot \sqsubseteq \forall \mathsf{f2}.\bot, \quad \forall \mathsf{f2}.\bot \sqsubseteq \forall \mathsf{f3}.\bot$ $\mathsf{A}_{\top 2} \equiv \exists \mathsf{f1}.\mathsf{A}_{\top 1} \sqcap \exists \mathsf{f2}.\mathsf{A}_{\top 1} \sqcap \forall \mathsf{f3}.\bot$ $\mathsf{WorksFor} \sqsubseteq \mathsf{A}_{\top 2}, \quad \mathsf{Manages} \sqsubseteq \mathsf{A}_{\top 2}$ $\mathsf{Employee} \sqsubseteq \mathsf{A}_{\top 1}, \quad \mathsf{Manager} \sqsubseteq \mathsf{A}_{\top 1},$ $\mathsf{AreaManager} \sqsubseteq \mathsf{A}_{\top 1}, \quad \mathsf{TopManager} \sqsubseteq \mathsf{A}_{\top 1}$
Typing Constraints:	$\mathsf{WorksFor} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{Employee}, \quad \mathsf{WorksFor} \sqsubseteq \exists \mathsf{f2}^-.\mathsf{Project}$ $\mathsf{Manages} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{TopManager}, \quad \mathsf{Manages} \sqsubseteq \exists \mathsf{f2}^-.\mathsf{Project}$
Frequency Constraints:	$\exists \mathsf{f1}^-.\mathsf{Manages} \sqsubseteq = 1\, \mathsf{f1}^-.\mathsf{Manages}$
Mandatory Constraints:	$\mathsf{Employee} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{WorksFor}$ $\mathsf{TopManager} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{Manages}$ $\mathsf{Project} \sqsubseteq \exists \mathsf{f2}^-.\mathsf{WorksFor}$ $\mathsf{Project} \sqsubseteq \exists \mathsf{f2}^-.\mathsf{Manages}$
Exclusion Constraints:	$\exists \mathsf{f1}^-.\mathsf{WorksFor} \sqsubseteq \mathsf{A}_{\top 2} \sqcap \neg \exists \mathsf{f1}.\mathsf{Manages}$
Subtyping Constraints:	$\mathsf{Manager} \sqsubseteq \mathsf{Employee} \sqcap (\mathsf{AreaManager} \sqcup \mathsf{TopeManager})$ $\mathsf{AreaManager} \sqsubseteq \neg \mathsf{TopeManager}$

a change we obtain a knowledge base as the one above, but with the defeasible inclusion axiom $\mathsf{Employee} \tilde{\sqsubseteq} \exists \mathsf{f1}^-.\mathsf{WorksFor}$ instead of the axiom $\mathsf{Employee} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{WorksFor}$.

Introducing such constraints, we introduce the forms of defeasible subsumptions appropriate for modeling nonmonotonic reasoning. In particular:

- A subclass relation, as the ones in example 3, is translated into an inclusion axiom $C \sqsubseteq D$, and correspondingly we translate the defeasible connection $C \leadsto D$ into a defeasible inclusion axiom $C \tilde{\sqsubseteq} D$.
- Analogously, consider the strict form of the example 4. The mandatory participation of the class B to the role A_N is translated into the axiom $\mathsf{B} \sqsubseteq \exists \mathsf{f1}^-.\mathsf{A_N}$. If we use the defeasible mandatory participation constraint, we simply translate the structure using the defeasible inclusion $\tilde{\sqsubseteq}$, obtaining the axiom $\mathsf{B} \tilde{\sqsubseteq} \exists \mathsf{f1}^-.\mathsf{A_N}$.

Hence, from a ORM graph with defeasible constraints we obtain an $\mathcal{ALCQI}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$, where $\mathcal{T}$ is a standard $\mathcal{ALCQI}$ Tbox containing concept inclusion axioms $C \sqsubseteq D$, while the set $\mathcal{B}$ contains defeasible axioms of the form $C \mathrel{\tilde{\sqsubseteq}} D$. Once we have our knowledge base $\mathcal{K}$, we apply to it the procedure presented in the previous section, in order to obtain the rational closure of the knowledge base.

Consistency. In ORM2, and in conceptual modeling languages in general, the notion of consistency is slightly different from the classical form of logical consistency. That is, generally from a logical point of view a knowledge base $\mathcal{K}$ is considered inconsistent only if we can classically derive a contradiction from it; in DLs that corresponds to saying that $\mathcal{K} \models \top \sqsubseteq \bot$, *i.e.* every concept in the knowledge base results empty. Instead, dealing with conceptual modeling schemas we generally desire that our model satisfies a stronger form of consistency constraint, that is, we want that none of the classes present in the schema are forced to be empty.

Definition 2 (Strong consistency). *A TBox $\mathcal{T}$ is* strongly consistent *if none of the atomic concepts present in its axioms are forced to be empty, that is, if $\mathcal{T} \not\models \neg A$ for every atomic concept A appearing in the inclusion axioms in $\mathcal{T}$.*

As seen above, the introduction of defeasible constraints into ORM2$^{\text{zero}}$ allows to build schemas that in the standard notation would be considered inconsistent, but that, once introduced the defeasible constraints, allow for an instantiation such that all the classes result non-empty. Hence it is necessary to redefine the notion of consistency check in order to deal with such situations.

Such a consistency check is not problematic, since we can rely on the ranking procedure presented above. Consider a TBox $\mathcal{T}$ obtained by an ORM2$^{\text{zero}}$ schema, and indicate with $\mathcal{C}$ the set of all the atomic concepts used in $\mathcal{T}$. It is sufficient to check the exceptionality ranking of all the concepts in $\mathcal{C}$ with respect to $\mathcal{T}$: if a concept C has an exceptionality ranking $r(C) = n$, with $0 < n < \infty$, then it represents an atypical situation, an exception, but that is compatible with the information conveyed by the defeasible inclusion axioms. For example, in the above examples the penguins and the top managers would be empty classes in the classical formalization, but using the defeasible approach they result exceptional classes in our schemas, and we can consider them as non-empty classes while still considering the schema as consistent. The only case in which a class has to be considered necessarily empty, is when it has ∞ as ranking value: that means that, despite we eliminate all the defeasible connections we can, such a concept still results empty. Then, the notion of strong consistency for ORM2$^{\text{zero}}$ with defeasible constraints is the following.

Definition 3 (Strong consistency with defeasible constraints). *A knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$ is* strongly consistent *if no one of the atomic concepts present in its axioms are forced to be empty, that is, if $r(A) \neq \infty$ for every atomic concept A appearing in the inclusion axioms in $\mathcal{K}$.*

Given a defeasible ORM2$^{\text{zero}}$schema Σ, eliminate from it all the defeasible constraints (call Σ_{strict} the resulting schema). From the procedures defined above, it is immediate to see that if Σ_{strict} is a strongly inconsistent ORM2$^{\text{zero}}$schema, in the 'classical' sense, then Σ is a strongly inconsistent defeasible schema: simply, if the negation of a concept is forced by the strict part of a schema, it will be necessarily forced at each ranking level, resulting in a ranking value of ∞.

On the other hand, there can be also strongly inconsistent defeasible schemas, which inconsistency depends not only on the strict part of the schema, but also on the defeasible part. For example, the schema in figure 4 is inconsistent, since the class A results to have a ranking value of ∞ (the schema declares that the class A is *directly* connected to two incompatible concepts). Now, we can check the results of the defined procedure in the examples presented.

Example 5. Consider example 3. From the translation of the defeasible form of the schema we conclude that the axiom $\mathsf{Penguin} \sqsubseteq \mathsf{NonFlyingObject}$has rank 1, while the $\mathsf{Bird} \sqsubseteq \mathsf{WingyObject}, \mathsf{Bird} \sqsubseteq \mathsf{FlyingObject}$ have rank 0, that means that we end up with two default concepts:

- $\delta_0 := \bigsqcap\{\mathsf{Penguin} \supset \mathsf{NonFlyingObject}, \mathsf{Bird} \supset \mathsf{WingyObject}, \mathsf{Bird} \supset \mathsf{FlyingObject}\}$;
- $\delta_1 := \mathsf{Penguin} \supset \mathsf{NonFlyingObject}$

We can derive the same kind of conclusions as in example 2, and again we can see the limits of the rational closure, since we cannot derive the desirable conclusion that $Penguin \mid\!\sim WingyObject$.

Example 6. Consider the knowledge base obtained in the example 4. We have only a defeasible inclusion axiom $\mathsf{Employee} \mathrel{\tilde{\sqsubseteq}} \exists \mathsf{f1}^-.\mathsf{WorksFor}$, and, since $\mathsf{Employee}$ does not turn out to be an exceptional concept, we end up with a single default concept in $\mathcal{B}$:

- $\delta_0 := \{\mathsf{Employee} \supset \exists \mathsf{f1}^-.\mathsf{WorksFor}\}$;

Since TopManager is not consistent with all the strict information contained in the schema plus δ_0, we cannot associate δ_0 to TopManager and, despite we have the information that for non-exceptional cases an employee works for a project, we are not forced to conclude that for the exceptional class of the top managers.

6 Conclusions and further work

In this paper we have presented a way to implement a form of defeasible reasoning into the ORM2 formalism. Exploiting the possibility of encoding ORM2$^{\mathsf{zero}}$, that represents a big portion of the ORM2 language, into the description logic $\mathcal{ALCQI}$ on one hand, and a procedure appropriate for modeling one of the main forms of nonmonotonic reasoning, *i.e.* rational closure, into DLs on the other hand, we have defined two new constraints, a *defeasible subclass relation* and a *defeasible mandatory participation*, that are appropriate for modeling defeasible information into ORM2, and that, once translated into $\mathcal{ALCQI}$, allow for the use of the procedures characterizing rational closure to reason about the information contained into an ORM2$^{\mathsf{zero}}$ schema.

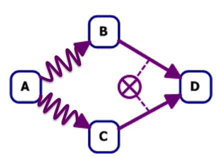

Fig. 4. Inconsistent schema.

The present proposal deals only with reasoning on the information contained in the TBox obtained from an ORM2 schema, but, once we have done the rational closure of the TBox, we can think also of introducing an ABox, that is, the information about a particular domain of individuals. A first proposal in such direction is in [4]. Actually we still lack a complete semantic characterization of rational closure in DLs, but hopefully we shall obtain soon such a result (a first step in such a direction is in [3]). Another future step will be the implementation of nonmonotonic forms of reasoning that extend rational closure, overcoming its inferential limits (see example 2), such as the *lexicographic closure* [16] or the *defeasible inheritance based* approach [5].

References

1. Halpin, T., Morgan, T.: Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design. 2nd edn. Morgan Kaufmann (2001)
2. Bonatti, P.A., Faella, M., Sauro, L.: Defeasible inclusions in low-complexity dls. J. Artif. Intell. Res. (JAIR) **42** (2011) 719–764
3. Britz, A., Meyer, T., Varzinczak, I.J.: Concept model semantics for dl preferential reasoning. In Rosati, R., Rudolph, S., Zakharyaschev, M., eds.: Description Logics. Volume 745 of CEUR Workshop Proceedings., CEUR-WS.org (2011)
4. Casini, G., Straccia, U.: Rational closure for defeasible description logics. In Janhunen, T., Niemelä, I., eds.: JELIA. Volume 6341 of Lecture Notes in Computer Science., Springer (2010) 77–90
5. Casini, G., Straccia, U.: Defeasible inheritance-based description logics. In: IJCAI-11. (2011) 813–818
6. Giordano, L., Olivetti, N., Gliozzi, V., Pozzato, G.L.: Alc + t: a preferential extension of description logics. Fundam. Inform. **96**(3) (2009) 341–372
7. Grimm, S., Hitzler, P.: A preferential tableaux calculus for circumscriptive $\mathcal{ALCO}$. In: RR-09. Number 5837 in LNCS, Berlin, Heidelberg, Springer-Verlag (2009) 40–54
8. Straccia, U.: Default inheritance reasoning in hybrid kl-one-style logics. IJCAI-93 (1993) 676–681
9. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artif. Intell. **55**(1) (1992) 1–60
10. Franconi, E., Mosca, A., Solomakhin, D.: Orm2: formalisation and encoding in owl2. In: OTM 2012 Workshops, Springer (2012) 368–378
11. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The description logic handbook: theory, implementation, and applications. Cambridge University Press, New York, NY, USA (2003)
12. Franconi, E., Mosca, A.: The formalisation of ORM2 and its encoding in OWL2. Technical Report KRDB12-2, KRDB Research Centre, Free University of Bozen-Bolzano (2012) Available at `http://www.inf.unibz.it/krdb/pub/TR/KRDB12-2.pdf`.
13. Calvanese, D., De Giacomo, G., Lenzerini, M.: Identification constraints and functional dependencies in description logics. In: Proceedings of the 17th international joint conference on Artificial intelligence. (2001) 155–160
14. Berardi, D., Cali, A., Calvanese, D., Giacomo, G.D.: Reasoning on UML class diagrams. ARTIFICIAL INTELLIGENCE **168** (2003)
15. Artale, A., Calvanese, D., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M.: Reasoning over extended ER models. In: Proc. of ER 2007, 26th International Conference on Conceptual Modeling, Springer (2007) 277–292
16. Lehmann, D.J.: Another perspective on default reasoning. Ann. Math. Artif. Intell. **15**(1) (1995) 61–82