Full Satisfiability of UML Class Diagrams (Extended Abstract)

Alessandro Artale, Diego Calvanese, and Angelica Ibáñez-García

KRDB Research Centre, Free University of Bozen-Bolzano, Italy
{artale,calvanese}@inf.unibz.it yibanez@gmail.com

Abstract. Class diagrams are one of the most important components of UML, the de-facto standard formalism for the analysis and design of software. The semantics of UML class diagrams is by now well established, and one can exploit automated reasoning tools that are based on the languages underlying their formalization to detect relevant properties, such as class satisfiability and subsumption. Among the reasoning tasks of interest, the basic one is detecting full satisfiability of a diagram, i.e., whether *all* classes and associations of the diagram can be simultaneously populated without violating any constraints of the diagram. While the complexity of class satisfiability has been studied extensively, full satisfiability received less attention. In this paper we address this problem, and establish tight upper and lower bounds for full satisfiability of UML class diagrams. Our results confirm the intuition that full satisfiability has the same computational complexity as class satisfiability.

1 Introduction

UML (Unified Modeling Language)[1] is the de-facto standard formalism for the analysis and design of software. One of the most important components of UML are *class diagrams* (UCDs), which model the information on the domain of interest in terms of objects organized in classes and associations between them (representing relations between class instances). The semantics of UCDs is by now well established, and several works propose to represent it using various kinds of formal systems, e.g., [12,11,13,4]. Hence, one can in principle reason on a UCD and formally prove properties about it. The properties that one is interested in are, e.g., subsumption between two classes, i.e., the fact that each instance of one class is necessarily also an instance of another class, satisfiability of a specific class or association in the diagram, i.e., the fact that the information encoding it in the diagram is not contradictory, and *full satisfiability* of the diagram [15], i.e., the fact that all classes and associations in the diagram are simultaneously satisfiable. The latter property is of importance since the presence of some unsatisfiable class or association actually means either that the diagram contains unnecessary information that should be removed, or that there is some modeling error that lead to the loss of satisfiability. In fact, it can be considered as the most fundamental property that should be satisfied by a UCD.

[1] http://www.omg.org/spec/UML/

The proposed formalizations of UCDs indicate that one can resort to the powerful automated inference mechanisms provided by the adopted models to verify the above mentioned properties. Such a reasoning support [9] is of importance for various tasks related to the design, maintenance, evolution, and integration of UCDs. For example, the formalization in terms of expressive Description Logics (DLs) [3] provided in [4] is well suited for this purpose. DLs are decidable logics that are specifically designed for the conceptual representation of an application domain in terms of classes and relationships between them. Representing conceptual data models by means of DLs has gathered consensus over the years, cf. [5,6,2,9,10,7,4], and allows one to exploit state-of-the-art DL reasoners [17] for inference in such models.

However, to avoid possible performance bottlenecks that could result from using a too powerful inference mechanism, and to be able to select the appropriate one to use for the above mentioned tasks, a fundamental question that needed to be addressed was that of the intrinsic complexity of reasoning on UCDs (independently of the formal tool adopted for describing them). This problem was addressed first in [4], where, somewhat surprisingly, it was shown that the simultaneous presence of multiplicity constraints and of completeness constraints on class and association hierarchies leads to ExpTime-hardness of *class satisfiability*. This result was then strengthened in [1] to UCDs[2] with simple ISA between associations (and completeness constraints on class hierarchies only).

However, no work had addressed explicitly the complexity of full satisfiability of UCDs[3]. In this paper, we fill this gap, by showing that the complexity of full satisfiability coincides with that of classical satisfiability. Our results build on the formalization of UML CDs in terms of DLs given in [4]. In fact, the upper bound is an almost direct consequence of the corresponding upper bound for UCDs derived from the DL formalization. Instead, our lower bound is more involved, as it requires a careful analysis of the corresponding proof for class satisfiability.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the DL $\mathcal{ALC}$, on which we base our results, and show that full satisfiability in $\mathcal{ALC}$ is ExpTime-complete. In Section 3, we recall the formalization of UCDs. In Section 4, we provide our main results on full satisfiability of UCDs. Finally, in Section 5, we draw some conclusions.

2 Full Satisfiability in the Description Logic $\mathcal{ALC}$

We start by studying *full satisfiability* for the DL $\mathcal{ALC}$, one of the basic variants of DLs [3]. The basic elements of $\mathcal{ALC}$ are atomic concepts and roles, denoted by A and P, respectively. Complex concepts C, D are defined by the following

[2] The results in [1] are formulated in terms of the Entity-Relationship model, but they carry directly over also to UML class diagrams.

[3] An exception is [15], which provides a PSpace upper bound for full satisfiability. However, our results here show that the algorithm of [15] must be incomplete.

rules:

$$C, D \ ::= \ A \ \mid \ \neg C \ \mid \ C \sqcap D \ \mid \ \exists P.\, C$$

The semantics of $\mathcal{ALC}$, as usual in DLs, is specified in terms of an *interpretation*. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, with a non empty *domain* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$, assigns to each concept C a subset of $\Delta^{\mathcal{I}}$, and to each role name P a binary relation in $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ such that the following conditions are satisfied:

$$\begin{aligned} A^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}}, \\ (\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}, \\ (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}}, \\ (\exists P.\, C)^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists b.\, (a, b) \in P^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}. \end{aligned}$$

We use the standard abbreviations $C_1 \sqcup C_2 := \neg(\neg C_1 \sqcap \neg C_2)$, and $\forall P.\, C := \neg\exists P.\, C$, with the corresponding semantics.

An $\mathcal{ALC}$ *terminological box* (TBox) $\mathcal{T}$ is a finite set of concept inclusion axioms of the form $C \sqsubseteq D$. An interpretation $\mathcal{I}$ *satisfies* an axiom of the form $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. A TBox $\mathcal{T}$ is *satisfiable* if there is an interpretation $\mathcal{I}$ that satisfies every axiom in $\mathcal{T}$ (such an interpretation is called a *model* of $\mathcal{T}$). A concept C is *satisfiable w.r.t. a TBox* $\mathcal{T}$ if there is a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$. It can be shown that TBox satisfiability and concept satisfiability w.r.t. a TBox are reducible to each other (in polynomial time). Moreover, reasoning w.r.t $\mathcal{ALC}$ TBoxes is ExpTime-complete (see e.g., [3]).

We now define the notion of *full satisfiability* of a TBox and show that for $\mathcal{ALC}$ it has the same complexity as classical satisfiability.

Definition 1 (TBox Full Satisfiability). *Let $\mathcal{T}$ be an $\mathcal{ALC}$ TBox. $\mathcal{T}$ is said to be* fully satisfiable *if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $A^{\mathcal{I}} \neq \emptyset$, for every atomic concept A in $\mathcal{T}$.*

Lemma 2. *Concept satisfiability w.r.t. $\mathcal{ALC}$ TBoxes can be linearly reduced to full satisfiability of $\mathcal{ALC}$ TBoxes.*

Proof. Let $\mathcal{T}$ be an $\mathcal{ALC}$ TBox and C an $\mathcal{ALC}$ concept. As pointed out in [8], C is satisfiable w.r.t. $\mathcal{T}$ if and only if $C \sqcap A_{\mathcal{T}}$ is satisfiable w.r.t. the TBox $\mathcal{T}_1$ consisting of the single assertion

$$A_{\mathcal{T}} \sqsubseteq \bigsqcap_{C_1 \sqsubseteq C_2 \in \mathcal{T}} (\neg C_1 \sqcup C_2) \sqcap \bigsqcap_{1 \leq i \leq n} \forall P_i.\, A_{\mathcal{T}}$$

where $A_{\mathcal{T}}$ is a fresh new atomic concept and $P_1, \ldots, P_n$ are all the atomic roles in $\mathcal{T}$ and C. In order to reduce the problem to full satisfiability, we extend $\mathcal{T}_1$ to $\mathcal{T}_2 = \mathcal{T}_1 \cup \{A_C \sqsubseteq C \sqcap A_{\mathcal{T}}\}$, with A_C a fresh new atomic concept, and prove that

$$C \sqcap A_{\mathcal{T}} \text{ is satisfiable w.r.t. } \mathcal{T}_1 \text{ iff } \mathcal{T}_2 \text{ is fully satisfiable}$$

($\Rightarrow$) Let $\mathcal{I}$ be a model of $\mathcal{T}_1$ such that $(C \sqcap A_{\mathcal{T}})^{\mathcal{I}} \neq \emptyset$. Construct a model of $\mathcal{T}_2$, $\mathcal{J} = (\Delta^{\mathcal{I}} \cup \{d^{top}\}, \cdot^{\mathcal{J}})$, with $d^{top} \notin \Delta^{\mathcal{I}}$, such that:

$$
\begin{aligned}
&A_{\mathcal{T}}^{\mathcal{J}} = A_{\mathcal{T}}^{\mathcal{I}}, \quad A_C^{\mathcal{J}} = (C \sqcap A_{\mathcal{T}})^{\mathcal{I}}, \\
&A^{\mathcal{J}} = A^{\mathcal{I}} \cup \{d^{top}\} \quad \text{for all atomic concepts } A \text{ in } \mathcal{T} \text{ and } C, \\
&P^{\mathcal{J}} = P^{\mathcal{I}} \quad \text{for all atomic roles,}
\end{aligned}
$$

Obviously, the extension of every atomic concept is non empty in $\mathcal{J}$. Next, we show that $\mathcal{J}$ is indeed a model of $\mathcal{T}_2$, relying on the fact (easily proved by structural induction) that $D^{\mathcal{I}} \subseteq D^{\mathcal{J}}$, for each subconcept D of concepts in $\mathcal{T}_1$. Then, it is easy to show that $\mathcal{J}$ satisfies every assertion in $\mathcal{T}_2$:

$$
\begin{aligned}
A_{\mathcal{T}}^{\mathcal{J}} = A_{\mathcal{T}}^{\mathcal{I}} &\subseteq \Big(\prod_{C_1 \sqsubseteq C_2 \in \mathcal{T}} (\neg C_1 \sqcup C_2) \sqcap \prod_{1 \leq i \leq n} \forall P_i . A_{\mathcal{T}}\Big)^{\mathcal{I}} \subseteq \\
&\subseteq \Big(\prod_{C_1 \sqsubseteq C_2 \in \mathcal{T}} (\neg C_1 \sqcup C_2) \sqcap \prod_{1 \leq i \leq n} \forall P_i . A_{\mathcal{T}}\Big)^{\mathcal{J}} \\
A_C^{\mathcal{J}} = (C \sqcap A_{\mathcal{T}})^{\mathcal{I}} &\subseteq (C \sqcap A_{\mathcal{T}})^{\mathcal{J}}
\end{aligned}
$$

($\Leftarrow$) Conversely, every *full model* $\mathcal{J}$ of $\mathcal{T}_2$ is also a model of $\mathcal{T}_1$ with $(C \sqcap A_{\mathcal{T}})^{\mathcal{J}} \neq \emptyset$, as $A_C^{\mathcal{J}} \subseteq (C \sqcap A_{\mathcal{T}})^{\mathcal{J}}$. □

Theorem 3. *Full satisfiability of $\mathcal{ALC}$ TBoxes is* ExpTime*-complete.*

Proof. The ExpTime membership is straightforward, as deciding full satisfiability of an $\mathcal{ALC}$ TBox $\mathcal{T}$ can be reduced to deciding satisfiability of the TBox

$$\mathcal{T} \cup \bigcup_{1 \leq i \leq n} \{\top \sqsubseteq \exists P' . A_i\},$$

where $A_1, \ldots, A_n$ are all the atomic concepts in $\mathcal{T}$, and P' is a fresh new atomic role. The ExpTime-hardness follows from Lemma 2. □

We now modify the reduction of Lemma 2 so that it applies also to *primitive* $\mathcal{ALC}^-$ TBoxes, i.e., TBoxes that contain only axioms of the form:

$$A \sqsubseteq B, \quad A \sqsubseteq \neg B, \quad A \sqsubseteq B \sqcup B', \quad A \sqsubseteq \forall P . B, \quad A \sqsubseteq \exists P . B,$$

where A, B, B' are atomic concepts, and P is an atomic role.

Theorem 4. *Full satisfiability of primitive $\mathcal{ALC}^-$ TBoxes is* ExpTime*-complete.*

Proof. The ExpTime membership follows from Theorem 3. For proving the ExpTime-hardness, we use a result in [4] showing that concept satisfiability in $\mathcal{ALC}$ can be reduced to atomic concept satisfiability w.r.t. primitive $\mathcal{ALC}^-$ TBoxes. Let $\mathcal{T}^- = \{A_j \sqsubseteq D_j \mid 1 \leq j \leq m\}$ be a primitive $\mathcal{ALC}^-$ TBox, and A_0

an atomic concept. By Lemma 2, we have that A_0 is satisfiable w.r.t. $\mathcal{T}^-$ if and only if the TBox $\mathcal{T}_2'$ containing the axioms

$$A_{\mathcal{T}^-} \sqsubseteq \bigsqcap_{A_j \sqsubseteq D_j \in \mathcal{T}^-} (\neg A_j \sqcup D_j) \sqcap \bigsqcap_{1 \leq i \leq n} \forall P_i.\, A_{\mathcal{T}^-}, \qquad A_0' \sqsubseteq A_0 \sqcap A_{\mathcal{T}^-}\,.$$

is fully satisfiable, with $A_{\mathcal{T}^-}, A_0'$ fresh new atomic concepts. $\mathcal{T}_2'$ is not a primitive $\mathcal{ALC}^-$ TBox, but it is equivalent to the TBox containing the assertions:

$$\begin{array}{lll} A_0' \sqsubseteq A_{\mathcal{T}^-} & A_{\mathcal{T}^-} \sqsubseteq \neg A_1 \sqcup D_1 & A_{\mathcal{T}^-} \sqsubseteq \forall P_1.\, A_{\mathcal{T}^-} \\ & \vdots & \vdots \\ A_0' \sqsubseteq A_0 & A_{\mathcal{T}^-} \sqsubseteq \neg A_m \sqcup D_m & A_{\mathcal{T}^-} \sqsubseteq \forall P_n.\, A_{\mathcal{T}^-}, \end{array}$$

Finally, to get a primitive $\mathcal{ALC}^-$ TBox, $\mathcal{T}_2^-$, we replace each axiom of the form $A_{\mathcal{T}^-} \sqsubseteq \neg A_j \sqcup D_j$ by $A_{\mathcal{T}^-} \sqsubseteq B_j^1 \sqcup B_j^2$, $B_j^1 \sqsubseteq \neg A_j$, and $B_j^2 \sqsubseteq D_j$, with B_j^1, B_j^2 fresh new atomic concepts, for $j = 1, \ldots, m$.

We show now that $\mathcal{T}_2'$ is fully satisfiable iff $\mathcal{T}_2^-$ is fully satisfiable:

($\Rightarrow$) Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a full model of $\mathcal{T}_2'$. We extend $\mathcal{I}$ into a full model $\mathcal{J}$ of $\mathcal{T}_2^-$. Let $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}} \cup \{d^+, d^-\}$, with $\{d^+, d^-\} \cap \Delta^{\mathcal{I}} = \emptyset$, and define $\cdot^{\mathcal{J}}$ as follows:

$$\begin{array}{rl} A_{\mathcal{T}^-}^{\mathcal{J}} = A_{\mathcal{T}^-}^{\mathcal{I}}, \quad (A_0')^{\mathcal{J}} = (A_0')^{\mathcal{I}}, & \\ A^{\mathcal{J}} = A^{\mathcal{I}} \cup \{d^+\}, & \text{for every other atomic concept in } \mathcal{T}_2', \\ (B_j^1)^{\mathcal{J}} = (\neg A)^{\mathcal{J}}, \text{ and } (B_j^2)^{\mathcal{J}} = (D_j)^{\mathcal{J}}, & \text{for each } A_{\mathcal{T}^-} \sqsubseteq B_j^1 \sqcup B_j^2 \in \mathcal{T}_2^-, \\ P^{\mathcal{J}} = P^{\mathcal{I}} \cup \{(d^+, d^+)\}, & \text{for every atomic role } P \text{ in } \mathcal{T}_2^-. \end{array}$$

It is easy to see now, that $\mathcal{J}$ fully satisfies $\mathcal{T}_2^-$.

($\Leftarrow$) Trivial. Every model of $\mathcal{T}_2^-$ is a model of $\mathcal{T}_2'$. □

3 Formalizing UML Class Diagrams

In this section, we briefly describe UCDs and provide their semantics in terms of First Order Logic (the formalization adopted here is based on previous presentations in [4,10]).

A *class* in a UCD denotes a set of objects with common features. Formally, a class C corresponds to a unary predicate C. An *association* represents a relation between instances of two or more classes. Names of associations (as names of classes) are unique in a UCD. A binary association between two classes C_1 and C_2 is graphically rendered as in Fig. 1. The *multiplicity* constraint $n_l..n_u$ written on one end of the binary association specifies that each instance of the class C_1 participates at least n_l times and at most n_u times in the association R, and the multiplicity constraint $m_l..m_u$ specifies an analogous constraint for each instance of the class C_2. When a multiplicity constraint is omitted, it is intended to be $0..*$. Formally, an association R between the classes

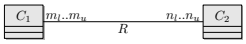

Fig. 1. Binary association

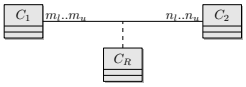

Fig. 2. Binary assoc. with related class

C_1, C_2 is captured by a binary predicate R that satisfies the FOL assertion $\forall x_1, x_2.\,(R(x_1, x_2) \rightarrow C_1(x_1) \wedge C_2(x_2))$, while multiplicities are formalized by the following FOL assertions:

$$\forall x.\,(C_1(x) \rightarrow \exists_{\geq n_l} y.\,R(x, y) \wedge \exists_{\leq n_u} y.\,R(x, y))$$
$$\forall y.\,(C_2(y) \rightarrow \exists_{\geq m_l} x.\,R(x, y) \wedge \exists_{\leq m_u} x.\,R(x, y)),$$

where we use counting quantifiers to abbreviate the FOL formula encoding the multiplicity constraints.

An association class describes properties of the association, such as attributes, operations, etc. (see Fig. 2). A binary association with a related *association class* C_R is formalized in FOL by reifying the association into a unary predicate C_R with two binary predicates P_1, P_2, one for each component of the association. We enforce the following semantics for $i = 1, 2$:

$$\begin{aligned}
\forall x.(C_R(x) &\rightarrow \exists y.\,P_i(x, y)),\\
\forall x, y.(C_R(x) \wedge P_i(x, y) &\rightarrow C_i(y)),\\
\forall x, y, y'.(C_R(x) \wedge P_i(x, y) \wedge P_i(x, y') &\rightarrow y = y'),\\
\forall y_1, y_2, x, x'.(C_R(x) \wedge C_R(x') \wedge (\textstyle\bigwedge_{i\in\{1,2\}} P_i(x, y_i) \wedge P_i(x', y_i)) &\rightarrow x = x').
\end{aligned}$$

For associations with a related class, the multiplicity constraints are formalized by the following FOL assertions:

$$\begin{aligned}
&\forall y_1.(C_1(y_1) \rightarrow \exists_{\geq n_l} x.\,(C_R(x) \wedge P_1(x, y_1)) \wedge \exists_{\leq n_u} x.\,(C_R(x) \wedge P_1(x, y_1)))\,,\\
&\forall y_2.(C_2(y_2) \rightarrow \exists_{\geq m_l} x.\,(C_R(x) \wedge P_2(x, y_2)) \wedge \exists_{\leq m_u} x.\,(C_R(x) \wedge P_2(x, y_2)))\,.
\end{aligned}$$

Generalizations (called also ISA constraints) between two classes C_1 and C specify that each instance of C_1 is also an instance of C. Several generalizations can be grouped together to form a class *hierarchy*, as shown in Fig. 3. *Disjointness* and *completeness* constraints can also be enforced on a class hierarchy, by adding suitable labels to the diagram. The class hierarchy shown in Fig. 3 is formally captured by means of the assertion $\forall x.\,C_i(x) \rightarrow C(x)$ for $i = 1, \ldots, n$. *Disjointness* among the classes $C_1, \ldots, C_n$ is expressed by $\forall x.\,C_i(x) \rightarrow \bigwedge_{j=i+1}^{n} \neg C_j(x)$ for $i = 1, \ldots, n-1$. Finally, the *completeness* constraint expressing that each instance of C is an instance of at least one of $C_1, \ldots, C_n$ is given by $\forall x.\,C(x) \rightarrow \bigvee_{i=1}^{n} C_i(x)$.

We can also have generalization between associations and between association classes with the obvious subset semantics as for generalization between classes. Finally, we do also allow for attributes associated to classes. Since the addition of attributes does not change the complexity of the satisfiability problem we do not present here attributes and their semantics.

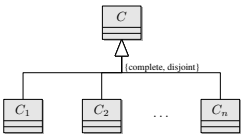

Fig. 3. A class hierarchy in UML

4 Full Satisfiability of UML Class Diagrams

Formally, three notions of UCD satisfiability have been proposed in the literature [16,4,15,14] First, *diagram satisfiability* of an UML diagram refers to the existence of a model of the diagram. Such model does not need to satisfy (populate) any class or association per se. The only condition is that all constraints are satisfied by the given model. Second, *class satisfiability* refers to the existence of a model of the diagram that satisfies (populates) a given class. Third, we can check whether there is a model of an UML diagram that satisfies *all* classes and *all* associations in a diagram. This last notion of satisfiability, referred here as *full satisfiability* and introduced in [15] is thus stronger than diagram satisfiability as a model of a diagram that satisfies all classes is, by definition, also a model of that diagram.

Definition 5 (UML Full Satisfiability). *A UCD $\mathcal{D}$ is* fully satisfiable *if there is an FOL interpretation $\mathcal{I}$ that satisfies all the constraints expressed in $\mathcal{D}$ and such that $C^{\mathcal{I}} \neq \emptyset$ for every class C in $\mathcal{D}$, and $R^{\mathcal{I}} \neq \emptyset$ for every association R in $\mathcal{D}$. We say that $\mathcal{I}$ is a* full model *of $\mathcal{D}$.*

We now address the complexity of full satisfiability for UCDs. We use the results presented in Section 2 and reduce full satisfiability of primitive $\mathcal{ALC}^-$ TBoxes to full satisfiability of UCDs. This reduction is based on the ones used in [4,1] for determining the lower complexity bound of schema satisfiability in the extended Entity-Relationship model.

Given a primitive $\mathcal{ALC}^-$ TBox $\mathcal{T}$, construct an UCD $\Sigma(\mathcal{T})$ as follows: for each atomic concept A in $\mathcal{T}$, introduce a class A in $\Sigma(\mathcal{T})$. Additionally, introduce a class O that generalizes (possibly indirectly) all the classes in $\Sigma(\mathcal{T})$ that encode an atomic concept in $\mathcal{T}$. For each atomic role P, introduce a class C_P, which reifies the binary relation P. Further, introduce two functional associations P_1, and P_2 that represent the first and second components of P. The assertions in $\mathcal{T}$ are encoded as follows:

1. For each assertion of the form $A \sqsubseteq B$, introduce a generalization between the classes A and B.
2. For each assertion of the form $A \sqsubseteq \neg B$, construct the hierarchy shown in Fig. 4.
3. For each assertion of the form $A \sqsubseteq B_1 \sqcup B_2$, introduce an *auxiliary* class B, and construct the diagram in Fig. 5.

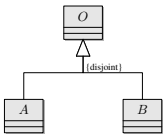

Fig. 4. Encoding of $A \sqsubseteq \neg B$

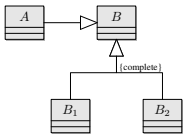

Fig. 5. Encoding of $A \sqsubseteq B_1 \sqcup B_2$

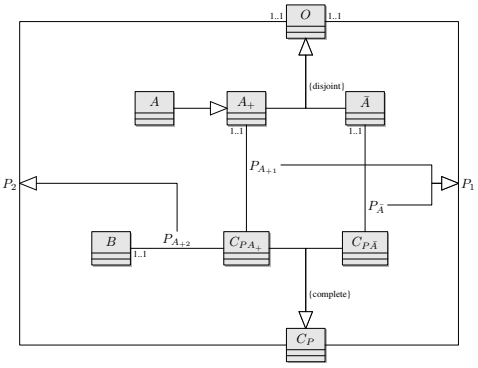

Fig. 6. Encoding of $A \sqsubseteq \forall P.\, B$

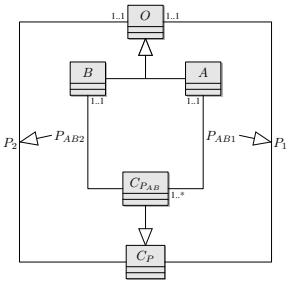

Fig. 7. Encoding of $A \sqsubseteq \exists P.\, B$

4. For each assertion of the form $A \sqsubseteq \forall P.\, B$, add the auxiliary classes C_{PA_+} and $C_{P\bar{A}}$, and the associations $P_{\bar{A}}$, $P_{A_{+1}}$ and $P_{A_{+2}}$, and construct the diagram shown in Fig. 6.
5. For each assertion of the form $A \sqsubseteq \exists P.\, B$, add the auxiliary class $C_{P_{AB}}$, and construct the diagram shown in Fig. 7.

Lemma 6. *A primitive $\mathcal{ALC}^-$ TBox $\mathcal{T}$ is fully satisfiable iff the UCD $\Sigma(\mathcal{T})$, constructed as above, is fully satisfiable.*

Proof. ($\Leftarrow$) Let $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ be a full model of $\Sigma(\mathcal{T})$. We construct a full model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{T}$ by taking $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$. Further, we define $A^{\mathcal{I}} = A^{\mathcal{J}}$ and $P^{\mathcal{I}} = (P_1^- \circ P_2)^{\mathcal{J}}$ for every concept name A and for every atomic role P in $\mathcal{T}$, respectively. Let us show that $\mathcal{I}$ satisfies every assertion in $\mathcal{T}$.

1. For assertions of the form $A \sqsubseteq B$, $A \sqsubseteq \neg B$, and $A \sqsubseteq B_1 \sqcup B_2$, the statement easily follows from the construction of $\mathcal{I}$.
2. For each assertion of the form $A \sqsubseteq \forall P.B$, let $o \in A^{\mathcal{I}} = A^{\mathcal{J}}$ and $o' \in \Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$, such that $(o, o') \in P^{\mathcal{I}}$. Since $P^{\mathcal{I}} = (P_1^- \circ P_2)^{\mathcal{J}}$, there is $o'' \in \Delta^{\mathcal{J}}$ such that $(o, o'') \in (P_1^-)^{\mathcal{J}}$, and $(o'', o') \in {P_2}^{\mathcal{J}}$. Then, $o'' \in C_P^{\mathcal{J}}$, and by the completeness constraint, $o'' \in C_{PA_+}^{\mathcal{J}} \cup C_{P\bar{A}}^{\mathcal{J}}$. We claim that $o'' \in C_{PA_+}^{\mathcal{J}}$. Suppose otherwise, then there is a unique $a \in \Delta^{\mathcal{J}}$, such that $(o'', a) \in P_{\bar{A}}^{\mathcal{J}}$ and $a \in \bar{A}^{\mathcal{J}}$. It follows from $P_{\bar{A}}^{\mathcal{J}} \subseteq P_1^{\mathcal{J}}$ and by the multiplicity constraint over C_P, that $a = o$. This rises a contradiction, because $o \in A^{\mathcal{J}} \subseteq A_+^{\mathcal{J}}$ and, $A_+^{\mathcal{J}}$ and $\bar{A}^{\mathcal{J}}$ are disjoint. Then $o'' \in C_{PA_+}^{\mathcal{J}}$. Further, there is a unique $b \in \Delta^{\mathcal{J}}$

with $(o'', b) \in P^{\mathcal{J}}_{A_{+2}}$ and $b \in B^{\mathcal{J}}$. From $P^{\mathcal{J}}_{A_{+2}} \subseteq P^{\mathcal{J}}_2$ and the multiplicity constraint on C_P, it follows that $b = o'$. Thus, we have that $o' \in B^{\mathcal{J}} = B^{\mathcal{I}}$, and therefore, $o \in (\forall P.B)^{\mathcal{I}}$.

3. For each assertion of the form $A \sqsubseteq \exists P.B$ in $\mathcal{T}$, let $o \in A^{\mathcal{I}} = A^{\mathcal{J}}$. Then, there is $o' \in \Delta^{\mathcal{J}}$ such that $(o', o) \in P^{\mathcal{J}}_{AB1}$ and $o' \in C^{\mathcal{J}}_{P_{AB}}$. Since $o' \in C^{\mathcal{J}}_{P_{AB}}$, there is $o'' \in \Delta^{\mathcal{J}}$ with $(o', o'') \in P^{\mathcal{J}}_{AB2}$ and $o'' \in B^{\mathcal{J}} = B^{\mathcal{I}}$. Then, as $P^{\mathcal{J}}_{AB2} \subseteq P^{\mathcal{J}}_2$, $P^{\mathcal{J}}_{AB1} \subseteq P^{\mathcal{J}}_1$ and $P^{\mathcal{I}} = (P^-_1 \circ P_2)^{\mathcal{J}}$, we can conclude that $(o, o'') \in P^{\mathcal{I}}$ and therefore, that $o \in (\exists P.B)^{\mathcal{I}}$.

($\Rightarrow$) Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a full model of $\mathcal{T}$, and $role(\mathcal{T})$ be the set of role names in $\mathcal{T}$. Extend $\mathcal{I}$ to a legal instantiation $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ of $\Sigma(\mathcal{T})$, by assigning suitable extensions for the auxiliary classes and associations in $\Sigma(\mathcal{T})$. Let $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}} \cup \Gamma \cup \Lambda$, where: $\Lambda = \biguplus_{A \sqsubseteq \forall P.B \in \mathcal{T}} \{a_{A_+}, a_{\bar{A}}\}$, such that $\Delta^{\mathcal{I}} \cap \Lambda = \emptyset$, and $\Gamma = \biguplus_{P \in role(\mathcal{T})} \Delta_P$, with:

$$\Delta_P = \{(o, o') \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (o, o') \in P^{\mathcal{I}}\} \cup \bigcup_{A \sqsubseteq \forall P.B \in \mathcal{T}} \{(a_{A_+}, b), (a_{\bar{A}}, \bar{o})\}$$

with b an arbitrary instance of B, and $\bar{o}$ an arbitrary element of $\Delta^{\mathcal{I}}$. We set $O^{\mathcal{J}} = \Delta^{\mathcal{I}} \cup \Lambda$, $A^{\mathcal{J}} = A^{\mathcal{I}}$ for each class A corresponding to an atomic concept in $\mathcal{T}$, and $C^{\mathcal{J}}_P = \Delta_P$ for each $P \in role(\mathcal{T})$. Additionally, the extensions of the associations P_1 and P_2 are defined as follows:

$$P^{\mathcal{J}}_1 = \{((o, o'), o) \mid (o, o') \in C^{\mathcal{J}}_P\}, \qquad P^{\mathcal{J}}_2 = \{((o, o'), o') \mid (o, o') \in C^{\mathcal{J}}_P\}.$$

We now show that $\mathcal{J}$ is a full model for $\Sigma(\mathcal{T})$.

1. For the portions of $\Sigma(\mathcal{T})$ due to TBox axioms of the form $A \sqsubseteq B$, $A \sqsubseteq \neg B$, and $A \sqsubseteq B_1 \sqcup B_2$, the statement follows from the construction of $\mathcal{J}$.
2. For each TBox axiom in $\mathcal{T}$ of the form $A \sqsubseteq \forall P.\, B$, let us define

$$\begin{array}{ll}
A^{\mathcal{J}}_+ = A^{\mathcal{I}} \cup \{a_{A_+}\}, & \bar{A}^{\mathcal{J}} = O^{\mathcal{J}} \setminus A^{\mathcal{J}}_+, \\
C^{\mathcal{J}}_{PA_+} = \{(o, o') \in C^{\mathcal{J}}_P \mid o \in A^{\mathcal{J}}_+\}, & C^{\mathcal{J}}_{P\bar{A}} = \{(o, o') \in C^{\mathcal{J}}_P \mid o \in \bar{A}^{\mathcal{J}}\}, \\
P^{\mathcal{J}}_{A_{+1}} = \{((o, o'), o) \in P^{\mathcal{J}}_1 \mid o \in A^{\mathcal{J}}_+\}, & P^{\mathcal{J}}_{\bar{A}} = \{((o, o'), o) \in P^{\mathcal{J}}_1 \mid o \in \bar{A}^{\mathcal{J}}\}, \\
P^{\mathcal{J}}_{A_{+2}} = \{((o, o'), o') \in P^{\mathcal{J}}_2 \mid o \in A^{\mathcal{J}}_+\}. &
\end{array}$$

It is not difficult to see that $\mathcal{J}$ satisfies the fragment of $\Sigma(\mathcal{T})$ as the one in Fig. 6. It remains to show that each class and each association have a non empty extension. This is clearly the case for classes that encode atomic concepts in $\mathcal{T}$. For the classes A_+, $\bar{A}$, C_{PA_+}, and $C_{P\bar{A}}$ we have that

$$a_{A_+} \in A^{\mathcal{J}}_+, \qquad a_{\bar{A}} \in \bar{A}^{\mathcal{J}}, \qquad (a_{A_+}, b) \in C^{\mathcal{J}}_{PA_+}, \qquad (a_{\bar{A}}, \bar{o}) \in C^{\mathcal{J}}_{P\bar{A}}.$$

For the associations P_1, P_2, $P_{A_{+1}}$, $P_{A_{+2}}$ and $P_{\bar{A}}$ we have that

$$\begin{array}{ll}
((a_{A_+}, b), a_{A_+}) \in P^{\mathcal{J}}_{A_{+1}} \subseteq P^{\mathcal{J}}_1, & ((a_{\bar{A}}, \bar{o}), a_{\bar{A}}) \in P^{\mathcal{J}}_{\bar{A}}, \\
((a_{A_+}, b), b) \in P^{\mathcal{J}}_{A_{+2}} \subseteq P^{\mathcal{J}}_2. &
\end{array}$$

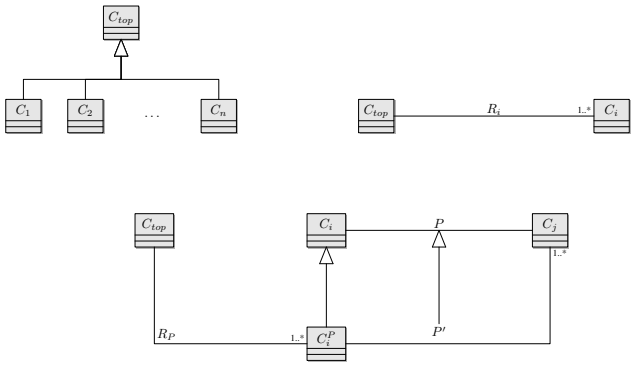

Fig. 8. Reducing UML full satisfiability to class satisfiability

3. For each TBox axiom in $\mathcal{T}$ of the form $A \sqsubseteq \exists P.B$, let us define the extensions for the *auxiliary* classes and associations as follows:

$$C^{\mathcal{J}}_{P_{AB}} = \{(o, o') \in \Delta_P \mid o \in A^{\mathcal{I}} \text{ and } o' \in B^{\mathcal{I}}\},$$
$$P^{\mathcal{J}}_{AB1} = \{((o, o'), o) \in P^{\mathcal{J}}_1 \mid (o, o') \in C^{\mathcal{J}}_{P_{AB}}\},$$
$$P^{\mathcal{J}}_{AB2} = \{((o, o'), o') \in P^{\mathcal{J}}_2 \mid (o, o') \in C^{\mathcal{J}}_{P_{AB}}\}.$$

We have that $C^{\mathcal{J}}_{P_{AB}} \neq \emptyset$ as there exists a pair $(a, b) \in \Delta_P$ with $a \in A^{\mathcal{I}}$, and $b \in B^{\mathcal{I}}$. Since $C^{\mathcal{J}}_{P_{AB}} \neq \emptyset$, we have that $P^{\mathcal{J}}_{AB1} \neq \emptyset$ and $P^{\mathcal{J}}_{AB2} \neq \emptyset$. □

Theorem 7. *Full satisfiability of UCDs is* ExpTime*-complete.*

Proof. The upper complexity bound can be established by reducing full consistency of UCDs to class consistency on UCDs, which is known to be ExpTime-complete [4]. Given a UCD $\mathcal{D}$, with classes $C_1, \ldots, C_n$, we construct the UCD $\mathcal{D}'$ by adding to $\mathcal{D}$ a new class C_{top} and a new association R_i for $i \in \{1, \ldots, n\}$. Besides, in order to ensure that every association is also populated, we consider each association P between the classes C_i and C_j such that neither C_i nor C_j is constrained to participate at least once in P. We add two new associations, R_P and P' and a new class C_i^P. Finally, we add the constraints shown in Fig. 8. Clearly, we have that $\mathcal{D}$ is fully satisfiable if and only if the class C_{top} is satisfiable.

The ExpTime-hardness follows from Lemma 6 and Theorem 4. □

5 Conclusions

This paper investigates the problem of *full satisfiability* in the context of UML class diagrams, i.e., whether *all* classes and associations of the diagram can be simultaneously populated without violating any constraints of the diagram. We show that the complexity of checking full satisfiability is ExpTime-complete,

thus matching the complexity of the classical schema satisfiability check. We show a similar result also for the problem of checking the full satisfiability of a TBox expressed in the description logic $\mathcal{ALC}$.

As a future work, we plan to extend the above results to UML class diagrams containing ad hoc subsets of the full sets of constructors. Furthermore, we intend to investigate the problem under the finite model assumption.

References

1. A. Artale, D. Calvanese, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Reasoning over extended ER models. In *Proc. of ER 2007*, volume 4801 of *LNCS*, pages 277–292. Springer, 2007.
2. A. Artale, F. Cesarini, and G. Soda. Describing database objects in a concept language environment. *IEEE Trans. on Knowledge and Data Engineering*, 8(2):345–351, 1996.
3. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
4. D. Berardi, D. Calvanese, and G. De Giacomo. Reasoning on UML class diagrams. *Artificial Intelligence*, 168(1–2):70–118, 2005.
5. S. Bergamaschi and C. Sartori. On taxonomic reasoning in conceptual design. *ACM Trans. on Database Systems*, 17(3):385–422, 1992.
6. A. Borgida. Description logics in data management. *IEEE Trans. on Knowledge and Data Engineering*, 7(5):671–682, 1995.
7. A. Borgida and R. J. Brachman. Conceptual modeling with description logics. In Baader et al. [3], chapter 10, pages 349–372.
8. M. Buchheit, F. M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge representation systems. *J. of Artificial Intelligence Research*, 1:109–138, 1993.
9. D. Calvanese, M. Lenzerini, and D. Nardi. Description logics for conceptual data modeling. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 229–264. Kluwer Academic Publishers, 1998.
10. D. Calvanese, M. Lenzerini, and D. Nardi. Unifying class-based representation formalisms. *J. of Artificial Intelligence Research*, 11:199–240, 1999.
11. T. Clark and A. S. Evans. Foundations of the Unified Modeling Language. In D. Duke and A. Evans, editors, *Proc. of the 2nd Northern Formal Methods Workshop*. Springer, 1997.
12. A. Evans, R. France, K. Lano, and B. Rumpe. Meta-modelling semantics of UML. In H. Kilov, editor, *Behavioural Specifications for Businesses and Systems*, chapter 2. Kluwer Academic Publishers, 1999.
13. D. Harel and B. Rumpe. Modeling languages: Syntax, semantics and all that stuff. Technical Report MCS00-16, The Weizmann Institute of Science, Rehovot, Israel, 2000.
14. M. Jarrar and S. Heymans. Towards pattern-based reasoning for friendly ontology debugging. *Int. J. on Artificial Intelligence Tools*, 17(4):607–634, 2008.
15. K. Kaneiwa and K. Satoh. Consistency checking algorithms for restricted UML class diagrams. In *Proc. of FoIKS 2006*, pages 219–239, 2006.
16. M. Lenzerini and P. Nobili. On the satisfiability of dependency constraints in entity-relationship schemata. *Information Systems*, 15(4):453–461, 1990.

17. R. Möller and V. Haarslev. Description logic systems. In Baader et al. [3], chapter 8, pages 282–305.