Adding Integrity Constraints to OWL

Boris Motik, Ian Horrocks, and Ulrike Sattler

University of Manchester, UK

Abstract. Schema statements in OWL are interpreted quite differently from analogous statements in relational databases. If these statements are meant to be interpreted as integrity constraints (ICs), OWL's interpretation may seem confusing and/or inappropriate. Therefore, we propose an extension of OWL with ICs that captures the intuition behind ICs in relational databases. We show that, if the constraints are satisfied, we can disregard them while answering a broad range of positive queries.

1 Introduction

The Web Ontology Language (OWL) is the language for modeling ontologies in the Semantic Web standardized by the W3C. The logical underpinning for OWL is provided by description logics (DLs) [1]. It is well-known that the OWL DL variant of OWL corresponds to the DL $\mathcal{SHOIN}(\mathbf{D})$. In this paper, we assume the reader to be familiar with the basics of OWL and DLs (please refer to [1] for an introduction), and we refer to OWL and DLs interchangeably.

OWL can be seen as an expressive schema language; however, its axioms have a different meaning from analogous statements in relational databases. When OWL axioms are meant to be interpreted as integrity constraints (ICs), the formal semantics of OWL may seem confusing and/or inappropriate. Consider an application for managing tax returns, in which each person must have a social security number. In a relational database, this would be enforced by adding the following inclusion dependency into the database schema $\mathcal{S}$:

$$\forall x : [Person(x) \rightarrow \exists y : hasSSN(x, y) \wedge SSN(y)] \tag{1}$$

During database updates, such a dependency is interpreted as a check. For example, an insertion of a fact $Person(Peter)$ into a database whose schema $\mathcal{S}$ contains (1) will be rejected, because we did not specify the social security number of $Peter$. Constraint checking in databases is interpreted as *model checking*: a database I satisfies a schema $\mathcal{S}$ iff $I \models \mathcal{S}$ (where $\models$ should be understood as satisfaction of a set of first-order formulae in a relational structure).

Dependency (1) can be expressed in OWL using the following TBox axiom:

$$Person \sqsubseteq \exists hasSSN.SSN \tag{2}$$

It is well known that the formula (1) is equivalent to the DL axiom (2) [3]. The effects of (2) in OWL, however, are quite different from the effects of (1)

in databases. From the fact $Person(Peter)$ and the axiom (2) we can conclude that Peter has some unknown social security number. Hence, the ontology is satisfiable; furthermore, if we query for the social security number of Peter, we do not get an answer, since this number is unknown.

As we discussed in [7, 8], OWL ontologies can be understood as incomplete databases. Many databases encountered in practice are, however, complete. To obtain a flexible schema language, we would like to explicitly control the degree of incompleteness. Integrity constraints (ICs)—formulae that check whether all necessary information has been provided explicitly—seem to be the proper mechanism for this purpose.

In this paper, we propose an extension of OWL with integrity constraints. In particular, in Section 2 we introduce *extended DL knowledge bases*, which allow a modeler to designate a subset of the TBox axioms as integrity constraints. For schema (TBox) reasoning, these axioms are treated as usual. For data (ABox) reasoning, these axioms do not derive new information; instead, they are interpreted as checks. Thus, the relationship between the interpretations of ICs in TBox and ABox reasoning is clear, and is analogous to the one in relational databases. In fact, in Section 3, we argue that, if an ABox satisfies the ICs, we can disregard the ICs while answering positive ABox queries. In [7, 8], we present algorithms for checking IC satisfaction for different kinds of OWL ontologies. Finally, in Section 4 we compare our solution to related approaches based on nonmonotonic modal extensions of DLs.

2 Constraints for OWL

The approach to integrity constraints found in relational databases can be easily applied to OWL: an ABox $\mathcal{A}$ would be interpreted as a single model and the TBox $\mathcal{T}$ as formulae that must be satisfied, and the constraints would be satisfied iff $\mathcal{A} \models \mathcal{T}$. Such an approach, however, is not satisfactory, as it requires an "all-or-nothing" choice: we would have to assume that *all* information in the ABox is complete; furthermore, TBox axioms could only be used to check whether an ABox is of an appropriate form and would not imply new facts. To obtain a more versatile formalism, we propose a combination of inferencing and constraint checking. For example, let $\mathcal{A}_1$ be the following ABox:

$$Student(Peter) \tag{3}$$

$$hasSSN(Peter, nr12345) \tag{4}$$

$$SSN(nr12345) \tag{5}$$

$$Student(Paul) \tag{6}$$

Furthermore, let $\mathcal{T}_1$ consist of (2) and the following axiom:

$$Student \sqsubseteq Person \tag{7}$$

Let us assume that we want to treat (2) as a constraint, but (7) as a normal axiom. Then, we derive $Person(Peter)$ and $Person(Paul)$ by (7). The constraint

(2) is satisfied for *Peter* due to (3), (4), and (5); however, an SSN has not been specified for *Paul*, so we expect (2) to be violated.

Following this intuition, we define extended DL knowledge bases to distinguish the axioms that imply new facts from the ones that check whether all necessary information is derivable. Our definition is applicable to any DL.

Definition 1. *An* extended DL knowledge base *is a triple* $\mathcal{K} = (\mathcal{S}, \mathcal{C}, \mathcal{A})$ *where*

- $\mathcal{S}$ *is a finite set of* standard *TBox axioms,*
- $\mathcal{C}$ *is a finite set of* constraint *TBox axioms, and*
- $\mathcal{A}$ *is a finite set of ABox assertions* $(\neg)A(a)$*,* $R(a, b)$*,* $a \approx b$*, or* $a \not\approx b$*, for* A *an atomic concept,* R *a role, and* a *and* b *individuals.*

In Definition 1, we restrict ourselves to ABoxes with only possibly negated atomic concepts. This does not result in any loss of generality, because we can replace nonatomic concept with new names and add appropriate axioms to $\mathcal{S}$.

Next, we discuss how to define an appropriate semantics for extended DL knowledge bases. The simplest solution is to interpret $\mathcal{A} \cup \mathcal{S}$ in the standard first-order way and to require $\mathcal{C}$ to be satisfied in each model I for which we have $I \models \mathcal{A} \cup \mathcal{S}$. The following example, however, shows that this does not satisfy our intuition. Let $\mathcal{A}_2$ contain only the fact (3), let $\mathcal{S}_2 = \emptyset$, and let $\mathcal{C}_2$ contain only the axiom (2). The interpretation $I = \{\mathit{Student}(\mathit{Peter}), \mathit{Person}(\mathit{Peter})\}$ is a model of $\mathcal{A}_2 \cup \mathcal{S}_2$ that does not satisfy $\mathcal{C}_2$, which would make $\mathcal{C}_2$ not satisfied for $\mathcal{A}_2 \cup \mathcal{S}_2$. Intuitively, though, the fact $\mathit{Person}(\mathit{Peter})$ is not implied by $\mathcal{A}_2 \cup \mathcal{S}_2$, so we should not check whether *Peter* has an SSN at all; $\mathcal{C}_2$ should hold only for the facts that are implied by $\mathcal{A}_2 \cup \mathcal{S}_2$.

These considerations might suggest that $\mathcal{C}$ should hold for all first-order consequences of $\mathcal{A} \cup \mathcal{S}$. In the example from the previous paragraph, this produces the desired behavior: $\mathit{Person}(\mathit{Peter})$ is not a consequence of $\mathcal{A}_2 \cup \mathcal{S}_2$, so the axiom from $\mathcal{C}_2$ should not be checked for *Peter*. Consider, however, the following knowledge base:

$$\mathcal{A}_3 = \{\mathit{Cat}(\mathit{ShereKahn})\} \tag{8}$$

$$\mathcal{S}_3 = \{\mathit{Cat} \sqsubseteq \mathit{Tiger} \sqcup \mathit{Leopard}\} \tag{9}$$

$$\mathcal{C}_3 = \{\mathit{Tiger} \sqsubseteq \mathit{Carnivore}, \quad \mathit{Leopard} \sqsubseteq \mathit{Carnivore}\} \tag{10}$$

Now neither $\mathit{Tiger}(\mathit{ShereKahn})$ nor $\mathit{Leopard}(\mathit{ShereKahn})$ is a first-order consequence of $\mathcal{A}_3 \cup \mathcal{S}_3$, which means that the axioms from $\mathcal{C}_3$ are satisfied; furthermore, we have $\mathcal{A}_3 \cup \mathcal{S}_3 \not\models \mathit{Carnivore}(\mathit{ShereKahn})$. This does not satisfy our intuition: in each model of $\mathcal{A}_3 \cup \mathcal{S}_3$, either $\mathit{Tiger}(\mathit{ShereKahn})$ or $\mathit{Leopard}(\mathit{ShereKahn})$ holds, but $\mathit{Carnivore}(\mathit{ShereKahn})$ does not necessarily hold in either case. Hence, by treating (10) as constraints and not as standard axioms, we neither get a constraint violation nor derive the consequence $\mathit{Carnivore}(\mathit{ShereKahn})$.

Intuitively, the constraints should check whether the facts derivable from $\mathcal{A} \cup \mathcal{S} \cup \mathcal{C}$ are also derivable using $\mathcal{A} \cup \mathcal{S}$ only. This notion seems to be nicely captured by minimal models; hence, we check $\mathcal{C}$ only w.r.t. the *minimal models* of $\mathcal{A} \cup \mathcal{S}$. Roughly speaking, a model I with an interpretation domain $\triangle^I$ of a

formula φ is minimal if each interpretation I' over $\triangle^I$ such that $I' \subset I$ is not a model of φ, where we consider an interpretation to be represented by the set of all positive ground facts that are true in it. Consider again $\mathcal{A}_2$, $\mathcal{S}_2$, and $\mathcal{C}_2$. The fact $\mathit{Person}(\mathit{Peter})$ is not derivable from $\mathcal{A}_2 \cup \mathcal{S}_2$ in any minimal model (in fact, there is only a single minimal model), so the constraint axiom (2) is not violated. In contrast, $\mathcal{A}_3 \cup \mathcal{S}_3$ has exactly two minimal models:

$$\begin{aligned} I_1 &= \{\mathit{Cat}(\mathit{ShereKahn}), \mathit{Tiger}(\mathit{ShereKahn})\} \\ I_2 &= \{\mathit{Cat}(\mathit{ShereKahn}), \mathit{Leopard}(\mathit{ShereKahn})\} \end{aligned}$$

These two models can be viewed as the minimal sets of derivable consequences. The TBox $\mathcal{C}_3$ is not satisfied in all minimal models. In contrast, let $\mathcal{A}_4 = \mathcal{A}_3$ and $\mathcal{C}_4 = \mathcal{C}_3$, and $\mathcal{S}_4 = \{\mathit{Cat} \sqsubseteq (\mathit{Tiger} \sqcap \mathit{Carnivore}) \sqcup (\mathit{Leopard} \sqcap \mathit{Carnivore})\}$. Now the fact $\mathit{Carnivore}(\mathit{ShereKahn})$ is derivable if either $\mathit{Tiger}(\mathit{ShereKahn})$ or $\mathit{Leopard}(\mathit{ShereKahn})$ is derivable, so the constraints should be satisfied. Indeed, $\mathcal{A}_4 \cup \mathcal{S}_4$ has the following two minimal models:

$$\begin{aligned} I_3 &= I_1 \cup \{\mathit{Carnivore}(\mathit{ShereKahn})\} \\ I_4 &= I_2 \cup \{\mathit{Carnivore}(\mathit{ShereKahn})\} \end{aligned}$$

Both I_3 and I_4 satisfy $\mathcal{C}_4$. Also, even though we treat the axioms from (10) as constraints, we have $\mathcal{A}_4 \cup \mathcal{S}_4 \models \mathit{Carnivore}(\mathit{ShereKahn})$.

Minimal models have been used, with minor differences, in an extension of DLs with circumscription [2] and in the semantics of open answer set programs [5]. Consider, however, $\mathcal{A}_5 = \{\mathit{Woman}(\mathit{Alice}), \mathit{Man}(\mathit{Bob})\}$, $\mathcal{S}_5 = \emptyset$, and $\mathcal{C}_5 = \{\mathit{Woman} \sqcap \mathit{Man} \sqsubseteq \bot\}$. No axiom implies that *Alice* and *Bob* should be interpreted as the same individual, so we expect them to be different "by default" and the constraint to be satisfied. The definitions from [2, 5] consider all interpretation domains, so let $\triangle^I = \{\alpha\}$. Because $\triangle^I$ contains only one object, we must interpret both *Alice* and *Bob* as α. Clearly, $I = \{\mathit{Woman}(\alpha), \mathit{Man}(\alpha)\}$ is a minimal model of $\mathcal{A}_5$, and it does not satisfy $\mathcal{C}_5$.

This problem might be remedied by making the unique name assumption (UNA)—that is, by requiring each constant to be interpreted as a distinct individual. This is, however, rather restrictive, and is not compatible with OWL. Another solution is to interpret $\mathcal{A} \cup \mathcal{S}$ in a Herbrand model (i.e., a model in which each constant is interpreted by itself) where $\approx$ is a congruence relation; then, we minimize the interpretation of $\approx$ together with all the other predicates. In such a case, the only minimal model of $\mathcal{A}_5$ is $I' = \{\mathit{Woman}(\mathit{Alice}), \mathit{Man}(\mathit{Bob})\}$ since the extension of $\approx$ is empty due to minimization, so $\mathcal{C}_5$ is satisfied in I'.

Unfortunately, existential quantifiers pose a range of problems for constraints. Let $\mathcal{A}_6 = \{\mathit{HasChild}(\mathit{Peter}), \mathit{HasHappyChild}(\mathit{Peter}), \mathit{TwoChildren}(\mathit{Peter})\}$, and let $\mathcal{S}_6$ contain these axioms:

$$\mathit{HasChild} \sqsubseteq \exists \mathit{hasChild}.\mathit{Child} \tag{11}$$

$$\mathit{HasHappyChild} \sqsubseteq \exists \mathit{hasChild}.(\mathit{Child} \sqcap \mathit{Happy}) \tag{12}$$

Finally, let $\mathcal{C}_6 = \{\mathit{TwoChildren} \sqsubseteq \geq 2\, \mathit{hasChild}.\mathit{Child}\}$. It seems intuitive for $\mathcal{C}_6$ to be satisfied in $\mathcal{A} \cup \mathcal{S}_6$: no axiom in $\mathcal{S}_6$ forces the children of *Peter*—the two

individuals whose existence is implied by (11) and (12)—to be the same, so we might conclude that they are different.

Now let $\mathcal{C}_7 = \mathcal{C}_6$, $\mathcal{A}_7 = \{\mathit{HasChild}(\mathit{Peter}), \mathit{TwoChildren}(\mathit{Peter})\}$, and let $\mathcal{S}_7$ contain the following axiom:

$$\mathit{HasChild} \sqsubseteq \exists \mathit{hasChild}.\mathit{Child} \sqcap \exists \mathit{hasChild}.\mathit{Child} \tag{13}$$

As in the previous example, $\mathcal{C}_7$ is satisfied in $\mathcal{A}_7 \cup \mathcal{S}_7$ since (13) introduces two (possibly identical) individuals in the extension of *Child*. Let $\mathcal{S}'_7$ be a standard TBox containing only the axiom (14):

$$\mathit{HasChild} \sqsubseteq \exists \mathit{hasChild}.\mathit{Child} \tag{14}$$

Now $\mathcal{C}_7$ is not satisfied in $\mathcal{A} \cup \mathcal{S}'_7$ since (14) implies the existence of only one child. Given that $\mathcal{S}'_7$ is semantically equivalent to $\mathcal{S}_7$, this is rather unsatisfactory; furthermore, it suggests that $\mathcal{C}_7$ should not be satisfied in $\mathcal{A}_7 \cup \mathcal{S}_7$, since (13) requires the existence of only one individual. Recall, however, that $\mathcal{S}_6$ and $\mathcal{S}_7$ are quite closely related: the effect of (13) with respect to *Child* is the same as that of (11) and (12). Hence, if (13) should introduce only one individual, then (11) and (12) should do so as well, which is in conflict with our intuition that $\mathcal{C}_6$ should be satisfied in $\mathcal{A}_6 \cup \mathcal{S}_6$.

Thus, our intuition does not give us a clear answer as to the appropriate treatment of existential quantifiers in the standard TBox: the names of the concepts and the structure of the axioms suggest that the existential quantifiers in (11) and (12) should introduce different individuals, whereas the existential quantifiers in (13) should "reuse" the same individual. These two readings seem to pull in opposite directions.

The example involving $\mathcal{S}_7$ and $\mathcal{S}'_7$ reveals an important disadvantage of one possible choice: if we require each existential quantifier to introduce a distinct individual, then it is possible for a constraint TBox $\mathcal{C}$ to be satisfied in $\mathcal{A} \cup \mathcal{S}$, but not in $\mathcal{A} \cup \mathcal{S}'$, even though $\mathcal{S}$ and $\mathcal{S}'$ are semantically equivalent. As we have seen, $\mathcal{C}_7$ is satisfied in $\mathcal{A}_7 \cup \mathcal{S}_7$, but not in $\mathcal{A}_7 \cup \mathcal{S}'_7$, even though $\mathcal{S}_7$ and $\mathcal{S}'_7$ are equivalent. It is clearly undesirable for IC satisfaction to depend on the syntactic structure of the standard TBox.

Introducing distinct individuals for each existential quantifier can be justified by *skolemization* [10], the well-known process of representing existential quantifiers with new function symbols. For example, for $\varphi = \exists y : [R(x, y) \wedge C(y)]$, by skolemization we obtain $\mathsf{sk}(\varphi) = R(x, f(x)) \wedge C(f(x))$: the variable y is replaced by a term $f(x)$, for f a new function symbol. Skolemized formulae are usually interpreted in *Herbrand* models, whose domain consists of all ground terms built from constants and function symbols in the formula.

Definition 2. *Let φ be a first-order formula and $\mathsf{sk}(\varphi)$ the formula obtained by outer skolemization of φ [10]. A Herbrand interpretation w.r.t. φ is a Herbrand interpretation defined over the signature of $\mathsf{sk}(\varphi)$. A Herbrand interpretation I w.r.t. φ is a* model *of φ, written $I \models \varphi$, if it satisfies φ in the usual sense. A Herbrand model I of φ is* minimal *if $I' \not\models \varphi$ for each Herbrand interpretation I'*

such that $I' \subset I$. *We write* $\mathsf{sk}(\varphi) \models_{\mathsf{MM}} \psi$ *if* $I \models \psi$ *for each minimal Herbrand model* I *of* φ.

We now define the notion of IC satisfaction. We use an operator π that translates a set of DL axioms S into an equivalent formula $\pi(S)$ of first-order logic with equality and counting quantifiers [1, 3].

Definition 3. *Let* $\mathcal{K} = (\mathcal{S}, \mathcal{C}, \mathcal{A})$ *be an extended DL knowledge base. The constraint TBox* $\mathcal{C}$ *is* satisfied *in* $\mathcal{K}$ *if* $\mathsf{sk}(\pi(\mathcal{A} \cup \mathcal{S})) \models_{\mathsf{MM}} \pi(\mathcal{C})$. *By an abuse of notation, we often omit* π *and simply write* $\mathsf{sk}(\mathcal{A} \cup \mathcal{S}) \models_{\mathsf{MM}} \mathcal{C}$.

Note that the addition of constraints does not change the semantics of DLs or OWL: Definition 3 is only concerned with the semantics of constraints, and a traditional knowledge base $(\mathcal{T}, \mathcal{A})$ can be seen as an extended knowledge base $(\mathcal{T}, \emptyset, \mathcal{A})$. For subsumption and concept satisfiability tests, we can use $\mathcal{S} \cup \mathcal{C}$ together as the schema, as usual. As discussed above, skolemization introduces a new function symbol for each existential quantifier, which effectively introduces a new individual for each quantifier. We invite the reader to convince himself that Definition 3 closely follows our intuition on the examples presented thus far. Furthermore, in Section 3 we show that, if the constraints are satisfied, we can throw them away without losing any positive consequences; that is, we can answer positive queries by taking into account only $\mathcal{A}$ and $\mathcal{S}$. We take this as confirmation that our semantics of IC satisfaction is intuitive.

Let $\mathcal{A}_8 = \{\, \mathit{Vegetarian}(\mathit{Ian}), \mathit{eats}(\mathit{Ian}, \mathit{soup})\}$, $\mathcal{S}_8 = \emptyset$, and let $\mathcal{C}_8$ contain only the following constraint:

$$\mathit{Vegetarian} \sqsubseteq \forall \mathit{eats}.\neg \mathit{Meat} \tag{15}$$

One might intuitively expect $\mathcal{C}_8$ not to be satisfied for $\mathcal{A}_8$ since the ABox does not state $\neg \mathit{Meat}(\mathit{soup})$. Contrary to our intuition, $\mathcal{C}_8$ is satisfied in $\mathcal{A}_8$: the interpretation I containing only the facts from $\mathcal{A}_8$ is the only minimal Herbrand model of $\mathcal{A}_8$ and $I \models \mathcal{C}_8$. In fact, the axiom (15) is equivalent to the axiom $\mathit{Vegetarian} \sqcap \exists \mathit{eats}.\mathit{Meat} \sqsubseteq \bot$. When written in the latter form, the axiom should be intuitively satisfied, since $\mathit{Meat}(\mathit{soup})$ is not derivable.

As this example illustrates, the intuitive meaning of constraints is easier to grasp if we transform them into the form $C \sqsubseteq D$, where both C and D are negation-free concepts. Namely, our constraints check the positive facts. To check negative facts, we must give them atomic names. Let $\mathcal{A}_9 = \mathcal{A}_8$, $\mathcal{S}_9 = \{\mathit{NotMeat} \equiv \neg \mathit{Meat}\}$, and $\mathcal{C}_9 = \{\, \mathit{Vegetarian} \sqsubseteq \forall \mathit{eats}.\mathit{NotMeat}\}$. The constraint in $\mathcal{C}_9$ is now of the "positive" form $C \sqsubseteq D$, so it is easier to understand the intuition behind it: everything that is eaten by an instance of *Vegetarian* should provably be *NotMeat*. Now, $\mathcal{A}_9 \cup \mathcal{S}_9$ has the following two minimal models, and $I_5 \not\models \mathcal{C}_9$, so $\mathcal{C}_9$ is not satisfied for $\mathcal{A}_9$:

$$\begin{aligned} I_5 &= \{\, \mathit{Vegetarian}(\mathit{Ian}), \mathit{eats}(\mathit{Ian}, \mathit{soup}), \mathit{Meat}(\mathit{soup})\} \\ I_6 &= \{\, \mathit{Vegetarian}(\mathit{Ian}), \mathit{eats}(\mathit{Ian}, \mathit{soup}), \mathit{NotMeat}(\mathit{soup})\} \end{aligned}$$

If we add to $\mathcal{A}_9$ the fact $\mathit{NotMeat}(\mathit{soup})$, then only I_6 is a minimal model, and $\mathcal{C}_9$ becomes satisfied as expected. Hence, it is advisable to restrict constraints to positive formulae in order to avoid such misunderstandings.

3 Constraints and Queries

We now show that, if the ICs are satisfied, we need not consider them while answering unions of positive conjunctive queries. This shows that our semantics of IC satisfaction is reasonable: constraints are checks and, if they are satisfied, we can discard them without losing relevant consequences. Moreover, this result is practically important because it simplifies query answering. Before proceeding, we first remind the reader of the definition of unions of conjunctive queries.

Definition 4. *Let* $\mathbf{x}$ *be a set of* distinguished *and* $\mathbf{y}$ *a set of* nondistinguished *variables. A* conjunctive query $Q(\mathbf{x}, \mathbf{y})$ *is a finite conjunction of positive atoms of the form* $A(t_1, \ldots, t_m)$*, where* t_i *are either constants, distinguished, or nondistinguished variables.*[1] *A* union of n conjunctive queries *is the formula* $\gamma(\mathbf{x}) = \bigvee_{i=1}^{n} \exists \mathbf{y_i} : Q_i(\mathbf{x}, \mathbf{y_i})$*. A tuple of constants* $\mathbf{c}$ *is an* answer *to* $\gamma(\mathbf{x})$ *over a DL knowledge base* $\mathcal{K}$*, written* $\mathcal{K} \models \gamma(\mathbf{c})$*, if* $\pi(\mathcal{K}) \models \gamma(\mathbf{x})[\mathbf{c}/\mathbf{x}]$.[2]

Our result is captured by the following theorem, whose proof is given in [8]:

Theorem 1. *Let* $\mathcal{K}$ *be an extended DL knowledge base that satisfies* $\mathcal{C}$*. Then, for any union of conjunctive queries* $\gamma(\mathbf{x})$ *over* $\mathcal{K}$ *and any tuple of constants* $\mathbf{c}$*,*

$$\mathcal{A} \cup \mathcal{S} \cup \mathcal{C} \models \gamma(\mathbf{c}) \text{ if and only if } \mathcal{A} \cup \mathcal{S} \models \gamma(\mathbf{c}).$$

Consider the example where $\mathcal{S}_{10} = \{Cat \sqsubseteq Pet, \exists hasPet.Pet \sqsubseteq PetOwner\}$, $\mathcal{C}_{10} = \{CatOwner \sqsubseteq \exists hasPet.Cat\}$, and $\mathcal{A}_{10}$ contains the following assertions:

$$CatOwner(John) \tag{16}$$

$$hasPet(John, Garfield) \tag{17}$$

$$Cat(Garfield) \tag{18}$$

Clearly, $\mathcal{S}_{10} \cup \mathcal{C}_{10} \cup \mathcal{A}_{10} \models PetOwner(John)$. Furthermore, $\mathcal{C}_{10}$ is clearly satisfied in $\mathcal{K}$: the only derivable fact about *CatOwner* is $CatOwner(John)$ and the ABox contains the explicit information that *John* owns *Garfield* who is a *Cat*. Therefore, we do not need $\mathcal{C}_{10}$ to imply the existence of the owned cat: whenever we can derive $CatOwner(x)$ for some x, we can derive the information about the cat of x as well. Hence, we can disregard $\mathcal{C}_{10}$ during query answering; we still have $\mathcal{S}_{10} \cup \mathcal{A}_{10} \models PetOwner(John)$.

Note that both entailments in Theorem 1 use the standard semantics of DLs; that is, we do not assume a closed-world semantics for query answering. Furthermore, Theorem 1 does not guarantee preservation of negative consequences; in fact, such consequences may change, as the following example demonstrates. Let $\mathcal{S}_{11} = \emptyset$, $\mathcal{C}_{11} = \{Cat \sqcap Dog \sqsubseteq \bot\}$, and $\mathcal{A}_{11}$ contain the axiom (18). By taking $\mathcal{S}_{11}$ into account, we have $\mathcal{S}_{11} \cup \mathcal{C}_{11} \cup \mathcal{A}_{11} \models \neg Dog(Garfield)$. Furthermore, the constraint is satisfied; however, without $\mathcal{C}_{11}$, we have $\mathcal{S}_{11} \cup \mathcal{A}_{11} \not\models \neg Dog(Garfield)$. A similar example can be given for queries containing universal quantifiers.

[1] The predicate A can be the equality predicate $\approx$, an atomic concept, a role, or an n-ary predicate in case of n-ary DLs.

[2] $\varphi[c/x]$ is the formula obtained from φ by replacing all free occurrences of x with c.

Theorem 1 has an important implication for TBox reasoning. Let $\gamma_1(\mathbf{x})$ and $\gamma_2(\mathbf{x})$ be queries such that $\pi(\mathcal{K}) \models \forall \mathbf{x} : [\gamma_1(\mathbf{x}) \rightarrow \gamma_2(\mathbf{x})]$. Provided that $\mathcal{C}$ is satisfied in $\mathcal{K}$, each answer to $\gamma_1(\mathbf{x})$ w.r.t. $\mathcal{A} \cup \mathcal{S}$ is also an answer to $\gamma_2(\mathbf{x})$ w.r.t. $\mathcal{A} \cup \mathcal{S}$. To summarize, we can check subsumption of unions of conjunctive as usual, by treating $\mathcal{C} \cup \mathcal{S}$ as an ordinary DL TBox. Subsequently, for knowledge bases that satisfy $\mathcal{C}$, we can ignore $\mathcal{C}$ when answering queries, but query answers will still satisfy the established subsumption relationships between queries.

4 Relationship to Autoepistemic DLs

The usefulness of constraint languages has been recognized early on in the knowledge representation community. In [11], Reiter noticed that constraints are epistemic in nature; furthermore, he presented an extension of first-order logics with an autoepistemic knowledge operator **K** that provides for introspection. In [6], Lifschitz presented the logic of Minimal Knowledge and Negation-as-Failure (MKNF) which also provides for a negation-as-failure operator **not**.

MKNF was used in [4] to obtain an expressive, but yet decidable nonmonotonic DL. One of the motivations for this work was to provide a language capable of expressing integrity constraints. For example, the constraint (1) can be expressed as the following axiom (the modal operator **A** corresponds to $\neg\,\mathbf{not}$):

$$\mathbf{K}\,\mathit{Person} \sqsubseteq \exists\,\mathbf{A}\,\mathit{hasSSN}.\,\mathbf{A}\,\mathit{Person} \tag{19}$$

MKNF was also used in [9] to integrate DLs with logic programming. One of the motivations for this work was to allow for constraint modeling. For example, the constraint (1) can expressed using the following rules:

$$\mathbf{K}\,\mathit{OK}(x) \leftarrow \mathbf{K}\,\mathit{hasSSN}(x, y), \mathbf{K}\,\mathit{SSN}(y) \tag{20}$$

$$\bot \leftarrow \mathbf{K}\,\mathit{Person}(x), \mathbf{not}\;\mathit{OK}(x) \tag{21}$$

Although the motivation is the same, these approaches differ from the one presented in this paper in several important aspects. First, the rules (20)–(21) do not have any meaning during TBox reasoning; they can only be used to check whether an ABox is of a required shape. Furthermore, the axiom (19) might be applied during TBox reasoning, but it has a significantly different semantics: it can be applied only to the consequences of other modal axioms, and not to consequences of other first-order axioms. In contrast, the constraint TBox $\mathcal{C}$ has the standard semantics for TBox reasoning and is applicable as usual; it is only for ABox reasoning that $\mathcal{C}$ is applied in a nonstandard way as a check. Thus, the semantics of $\mathcal{C}$ is much closer to the standard semantics of description logics.

Second, the semantics of MKNF makes it almost impossible to express constraints on unnamed individuals. Namely, for a first-order concept C, the concept $\mathbf{K}\,C$ contains the individuals that are in C in all models of C. In most cases, $\mathbf{K}\,C$ contains only the explicitly named individuals, and not the unnamed individuals implied by existential quantifiers; namely, in different models one can choose different individuals to satisfy an existential quantifier. Therefore, MKNF-based

approaches cannot check, for example, whether social security numbers of each person are explicitly known. In contrast, the approach from this paper can express such a constraint: we just need to add an assertion $O(a)$ for each named object a, and then use a constraint $PersonTR \sqsubseteq \exists hasSSN.(O \sqcap SSN)$ (see [7, 8] for more information).

Third, MKNF-based constraints work at the level of consequences and therefore cannot express constraints on disjunctive facts. Consider again the ABox $\mathcal{A}_3$ containing the axiom (8) and the standard TBox $\mathcal{S}_3$ containing the axiom (9). We can express the constraints (10) as follows:

$$\bot \leftarrow \mathbf{K}\, Tiger(x), \mathbf{not}\, Carnivore(x) \tag{22}$$

$$\bot \leftarrow \mathbf{K}\, Leopard(x), \mathbf{not}\, Carnivore(x) \tag{23}$$

Unfortunately, (22) and (23) are satisfied for $\mathcal{A}_3 \cup \mathcal{S}_3$. Namely, $\mathbf{K}\, Tiger(x)$ can, roughly speaking, be understood as "$Tiger(x)$ is a consequence." Due to the disjunction in (9), neither $Tiger(ShereKahn)$ nor $Leopard(ShereKahn)$ is a consequence of $\mathcal{A}_3 \cup \mathcal{S}_3$; hence, the premise of neither rule is satisfied and the constraints are not violated.

Because of these differences, we believe that the semantics of the extended DL knowledge bases captures the intuition behind constraints in a much more intuitive way; furthermore, it seems to fit better with the usual semantics of description logics.

5 Conclusion

Motivated by the problems encountered in the applications of OWL to data-centric problems, we have proposed the notion of *extended* DL knowledge bases, in which a certain subset of TBox axioms can be designated as constraints. For TBox reasoning, constraints behave just like normal TBox axioms; for ABox reasoning, however, they are interpreted in the spirit of relational databases. We define the semantics of IC satisfaction in such a way that they indeed check whether all mandatory assertions are entailed by the given ABox and TBox. We have also shown that, if the constraints are satisfied, we can disregard them while answering positive queries. This indicates that our semantics of constraint satisfaction is indeed reasonable.

In future, we plan to implement our approach in the OWL reasoner KAON2[3] and test its usefulness on practical problems.

References

1. 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.

[3] http://kaon2.semanticweb.org/

2. P. Bonatti, C. Lutz, and F. Wolter. Description Logics with Circumscription. In *Proc. KR 2006*, pages 400–410, Lake District, UK, 2006.
3. A. Borgida. On the Relative Expressiveness of Description Logics and Predicate Logics. *Artificial Intelligence*, 82(1–2):353–367, 1996.
4. F. M. Donini, D. Nardi, and R. Rosati. Description Logics of Minimal Knowledge and Negation as Failure. *ACM Transactions on Computational Logic*, 3(2):177–225, 2002.
5. S. Heymans, D. Van Nieuwenborgh, and D. Vermeir. Conceptual Logic Programs. *Annals of Mathematics and Artificial Intelligence*, 2006. To appear.
6. V. Lifschitz. Minimal Belief and Negation as Failure. *Artificial Intelligence*, 70(1–2):53–72, 1994.
7. B. Motik, I. Horrocks, and U. Sattler. Bridging the Gap Between OWL and Relational Databases. In *Proc. WWW 2007*, Banff, Alberta, Canada. ACM Press. To appear.
8. B. Motik, I. Horrocks, and U. Sattler. Integrating Description Logics and Relational Databases. Technical report, University of Manchester, UK, 2006.
9. B. Motik and R. Rosati. A Faithful Integration of Description Logics with Logic Programming. In *Proc. IJCAI 2007*, Hyderabad, India, 2007.
10. A. Nonnengart and C. Weidenbach. Computing Small Clause Normal Forms. In *Handbook of Automated Reasoning*, volume I, chapter 6, pages 335–367. Elsevier Science, 2001.
11. R. Reiter. What Should a Database Know? *Journal of Logic Programming*, 14(1–2):127–153, 1992.