ABSTRACT
In many data-centric applications, it is desirable to use OWL as an expressive schema language with which one expresses constraints that must be satisfied by instance data. However, specific aspects of OWL's standard semantics—i.e., the Open World Assumption (OWA) and the absence of Unique Name Assumption (UNA)—make it difficult to use OWL in this way. In this paper, we present an Integrity Constraint (IC) semantics for OWL axioms, show that IC validation can be reduced to query answering, and present our preliminary results with a prototype implementation using Pellet.

1. MOTIVATION
Standard OWL semantics adopting OWA and non-UNA make it difficult to use OWL for data validation. What triggers a constraint violation in closed world systems leads to new inferences in standard OWL systems. To use OWL both as a knowledge representation language and an integrity constraint language for data validation, we must combine open world reasoning and closed world constraint checking. Tao et al. [5] characterize the typical integrity issues in Semantic Web data using autoepistemic operators, and propose an integrity issue checking solution based on SPARQL queries. Although the semantics of typical integrity constraints are provided in this work, a comprehensive approach describing the semantics of all possible constraints would be preferable. Motik et al. [3] propose a semantics for ICs based on the satisfaction of IC axioms in minimal Herbrand models of the KB. Even though this semantics is reasonable, it has several unsatisfying features: First, constraints can be satisfied by unnamed individuals even if almost all ICs are meant to be satisfied only by named individuals. Second, to define a constraint to be satisfied only by named individuals, a predicate $O$ has to be added into the original constraint axiom, which makes the constraint definition unnecessarily complex. Third, the disjunction axioms and ICs interact in an unintuitive way. Due to these limitations, we need an alternative semantics for OWL axioms used for ICs.

2. IC SEMANTICS
Our approach is inspired by Reiter [4] which argues that ICs are epistemic in nature and are about “what the knowledge base knows”. Further investigation is found in [1] where an epistemic extension of DL, $\mathcal{ELC}$, is proposed: ICs are interpreted with epistemic interpretation $(I, W)$, $(\mathcal{KC})^{(I, W)}$ (resp. $\mathcal{KR}^{(I, W)}$) represents the individuals (resp. pair of individuals) that are known to be instances of $C$ (resp. known to be associated by $R$) in $W$, and a constraint axiom $\alpha$ is satisfied by $K$ if for every $I \in W$, interpretation $(I, W)$ satisfies $\alpha$ where $W = \text{Mod}(K)$ standing for all models of $K$.

However we find several restrictions of this work making it not suitable to represent the semantics of ICs in expressive DL: (1) It does not address how to interpret $\mathcal{KC}$ (resp. $\mathcal{KR}$) when $C$ (resp. $R$) is not an atomic concept (resp. a simple role). (2) It adopts UNA which is not compatible with OWL. A more reasonable solution is to treat two different identifiers as distinct if doing this does not cause logical inconsistencies, otherwise they should be interpreted as same. (3) We also find out that allowing $W$ including all models of $K$ is not satisfying. It is more intuitive to let $W$ be the set of models that minimally handle the equality relationship between individuals, i.e., two different individuals are interpreted to be same only when this is necessary for the consistency of KB.

Taking above considerations into account, now we describe an IC semantics for OWL. First we define the notion of IC interpretation $(I, W)$ where $I$ is a $\mathcal{SROIQ}$ [2] (logic underpinning of OWL2) interpretation and $W$ is a set of $\mathcal{SROIQ}$ interpretations. Note, for compatibility reasons we use the same representation of epistemic interpretation. Also note, different from epistemic interpretation, IC-interpretation does not adopt UNA. With IC-interpretation, atomic concepts and simple roles are interpreted as follows:

\[
C^{(I, W)} = \{ a^I | a \in N_I \text{ s.t. } \forall J \in W, a^J \in C^J \}
\]

\[
R^{(I, W)} = \{ (a^I, b^J) | a, b \in N_I \text{ s.t. } \forall J \in W, (a^J, b^J) \in R^J \}
\]

where $N_I$ denotes the set of named individuals in KB. It is easy to see that $C^{(I, W)}$ and $R^{(I, W)}$ represent the individuals that are known to be instances of $C$ in $W$ and the pair of individuals that are known to be associated by $R$ in $W$ respectively. The interpretation of complex roles and concept descriptions is described in Figure 1. Note that $(C \sqcup D)^{(I, W)}$ represents the individuals that are known to be instances of $C$ in $W$ or known to be instances of $D$ in $W$. We define its semantics this way because it is more intuitive for constraint modeling. Consider $K = \{ C(a), (C_1 \sqcup C_2)(a) \}$ and constraint $C \subseteq C_1 \sqcup C_2$. The more reasonable meaning
of this constraint is “every known instance of C should be a
known instance of C1 or a known instance of C2”. As a re-
sult, this constraint is violated by individual a since we don’t
know whether a is an instance of C1 or C2. It is if really
the intention of the ontology modeler to express a constraint
with the meaning of "every known instance of C should be a
known instance of C1 or C2", we can designate a new name
C’ = C1 ⊔ C2 and represent the constraint as C ⊑ C’. Also
note that (¬C’)I(W) is interpreted as the individuals which
are not known to be instances of C. As a result, ¬C has a
closed world meaning with the IC interpretation.

The satisfaction of SROIQ axioms in IC-interpretations is
defined in above table where W = ModMMe(K) is defined as
follows:

{I | I ∈ Mod(K) s.t. either ∃J ⊂ I or ∀J ⊂ I, J ∉ Mod(K)}

where J ⊂ I is true if the following three conditions hold at
the same time:

For all concepts C, a ∈ C implies a ∈ C;

For all roles R, (a, b) ∈ R implies (a, b) ∈ R;

∀a, b ∈ Ni s.t. a ̸= b and a ̸= b.

According to above definition, it is easy to see every model in
ModMMe(K) interprets different identifiers to be same only when
it is necessary for the logical consistency of KB. We say
that an axiom α is IC-satisfied by a SROIQ knowledge base
K, written as K ⊨ IC α, if for every I ∈ W the IC-
interpretation (I, W) satisfies α where W = ModMMe(K).
A SROIQ DL KB extended with integrity constraint axioms
C denoted as (K, C), satisfies C if K ⊨ IC C for every
α ∈ C.

3. IMPLEMENTATION OF IC VALIDATION

We find out when the given ontology is a SHI KB or the
constraints are not in the form of cardinality constraints with
n > 1, IC validation can be reduced to query (with NAF) an-
swering over the KB. In the future, we will provide a formal
proof of reducing IC validation to corresponding query an-
swering, and research how to validate ICs in more expressive
KBs. We have built a prototype IC validator1 by extending
Pellet2. The prototype include a parser, a translator, and
a validator for ICs that can read, process and validate
ICs written as OWL, OWL2, or SWRL axioms. The IC
validator can be accessed via a command-line program or the
validation API that validate ICs and output validation
results. The prototype is implemented by translating ICs
to SPARQL queries and then execute the queries over Pel-
let reasoner. The performance study shows that our pro-
totype IC validator can be used to efficiently validate rela-
tively large datasets. For instance it only takes 2 seconds
to check an IC over a KB containing 100K instances and
800K assertions. To validate arbitrary-sized data, we plan
to implement incremental IC validation in the future.

4. REFERENCES

[1] Francesco M. Donini, Maurizio Lenzerini, Daniele
Nardi, Werner Nutt, and Andrea Schaerf. An epistemic
operator for description logics. Artificial Intelligence,
[2] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The even
the gap between owl and relational databases. In
[5] Jiao Tao, Li Ding, Jie Bao, and Deborah L.
McGuinness. Characterizing and detecting integrity

1http://clarkparsia.com/pellet/oicv-0.1.2.zip
2http://clarkparsia.com/pellet