Adding Integrity Constraints to the Semantic Web for Instance Data Evaluation

Jiao Tao

Department of Computer Science, Rensselaer Polytechnic Institute,
Troy, NY 12180, USA

Abstract. This paper presents our work on supporting evaluation of integrity constraint issues in semantic web instance data. We propose an alternative semantics for the ontology language, i.e., OWL, a decision procedure for constraint evaluation by query answering, and an approach of explaining and repairing integrity constraint violations by utilizing the justifications of conjunctive query answers.
keywords: Instance Data, Evaluation, Integrity Constraint, OWL

1 Introduction: Motivation and Challenges

With the rapid growth of semantic web [1] technologies, a lot of semantic web applications such as Twine, Freebase, TrueKnowledge, Hakia, etc, are emerging on the web. To consume the data generated by these applications, it is critical to evaluate the data and ensure it meets the needs of users first. The data on the semantic web includes the ontologies that describe the schema of the domain, and the instance data that refers to the ground level data particular to the applications. There has been a lot of research aimed at ontology evaluation [2] [3] [4], however there is little, if any, research focusing on instance data evaluation, even though instance data usually accounts for orders of magnitude more than ontologies on the web.

We have identified three categories of issues [5] that may occur in instance data which are syntax errors, logical inconsistencies, and integrity constraint issues: syntax errors are the issues indicating that the syntax representation of instance data does not conform to the corresponding syntax specifications such as RDF/XML, N3, N-Triple, Turtle, etc; logical inconsistencies are the issues showing that the underlying logical theory of the instance data includes contradictory axioms; integrity constraint issues are the issues caused by the failure of the instance data to follow the restrictions that are imposed by the integrity constraints. While the evaluation of first two categories of issues are well studied [6] [7] [8] [9] and are being supported by existing tools such as W3C RDF validation service [1], the DL reasoner Pellet [2], Chimaera [10], the BBN validator [3], and ODEval [11], evaluation support for integrity constraint issues in semantic web instance data still remains an open research problem.

---

[1] http://www.w3.org/RDF/Validator/
[2] http://clarkparsia.com/pellet/
[3] http://projects.semwebcentral.org/projects/vowlidator/

The concept of integrity constraint (IC) was invented in the field of relational databases where ICs are used to ensure data consistency [12]. As an important functionality, almost all database systems support IC checking. The main approach for checking ICs in databases is to represent ICs as part of the database schema and translate the constraints to corresponding queries. Whenever there is an update to the data, the queries are executed first to see if the ICs are violated, therefore preventing potential constraint violations. One might wonder if similar approach can be used for the task of evaluating IC issues on the semantic web: modeling ICs as part of the domain knowledge using knowledge representation languages on the semantic web such as OWL [13] then translating IC axioms to queries and validating ICs by corresponding query answering. The standard semantics of OWL is based on Description Logics (DL) which has the following two characteristics:

- Open World Assumption (OWA): i.e., a statement cannot be inferred to be false on the basis of failures to prove it.
- Absence of the Unique Name Assumption (UNA): i.e., two different names may refer to the same object.

Due to the above characteristics, what triggers constraint violations in closed world systems, such as databases, leads to new inferences in standard OWL systems. Therefore, it is difficult to use OWL for IC evaluation. In this paper, we present our work on supporting evaluatin of integrity constraint issues in semantic web instance data by enabling OWL as an IC language.

2   Related Work

Several existing proposals on enabling OWL as an IC language combine OWL with different formalisms such as rules, epistemic queries, or epistemic logics. The rule-based approach [14] [15] expresses ICs as rules with a special predicate and check ICs by examining if the special predicate is entailed by the hybrid DL-rule knowledge base (KB). With this approach the developers need to be familiar with rules. The epistemic query-based approach [16] expresses ICs as epistemic queries and evaluates ICs by checking the epistemic query answers. However the complexity results of this approach in expressive DLs are still unknown. The epistemic DL-based approach extends DLs with epistemic logics and expresses ICs with epistemic DL axioms. With this approach, IC evaluation is to determine if the IC axioms are epistemic-entailed by the DL KB. However, this approach has two limitations: first, it focuses on less expressive DLs; second, it adopts the strict UNA which is not compatible with OWL since two different OWL individual names might refer to the same object.

In this paper, we focus on approaches that reuse OWL as an IC language. Our closest related work is a minimal Herbrand model-based approach [17]. With this approach, an OWL IC axiom is satisfied if all minimal Herbrand models of the KB satisfy it. This approach may result in counterintuitive results or modeling burden: first, existentially quantified individuals can satisfy ICs, which is not

desirable for IC evaluation; second, with this approach, if an IC needs to be satisfied only by individual names, then a special concept O has to be added into the original IC axiom, and every individual name should be asserted as an instance of O. This adds a significant maintenance burden on ontology developers while still not capturing the intuition behind ICs; third, the disjunctions and ICs may interact in unexpected ways.

3   Research Objectives and Plan

This work is aimed to support evaluating IC issues in semantic web instance data. For this purpose, we identify the following research objectives and plan:

- Enabling OWL as an IC language. Aside from being an ontology language, OWL is also an IC language that can be used to represent constraints that the instance data has to satisfy.
- Providing decision procedures for IC evaluation. Given a set of instance data and IC axioms, the decision procedures decide if the ICs are violated by the instance data.
- Providing explanation services for IC violations which explain why certain ICs are violated and recommend how to repair the violations.
- Implementation and Evaluation. Implementing a prototype and evaluating IC issues in instance data.

4   Research Progress

Until now we have completed the review of the state of the art, and compared the different approaches of enabling OWL as an IC language. Due to the various issues of these approaches that we have discussed in Section 2, we decided that designing an alternative semantics for OWL which supports ICs and correctly captures the intuitions behind ICs would be a promising solution. The work that we have done so far includes an IC semantics for OWL, a decision procedure for IC evaluation, an approach to explain and repair IC violations, and a partial implementation.

4.1   IC Semantics for OWL

In this section, we will describe an IC semantics [18] that we have proposed for OWL [19] which is based on DL $\mathcal{SROIQ}$[20].

In the IC semantics, we adopt a weak form of UNA [4]: two individual names with different identifiers are assumed to be different by default unless their equality is required to satisfy the axioms in the KB. We formalize this notion of weak UNA by Minimal Equality (ME) models. Given a $\mathcal{SROIQ}$ KB $\mathcal{K}$, $\mathcal{I}$ and $\mathcal{J}$ are two $\mathcal{SROIQ}$ interpretations of $\mathcal{K}$, we say $\mathcal{J} \prec_= \mathcal{I}$ if: (1)$\forall C \in N_C$, $\mathcal{J} \models C(a)$

---

[4] With UNA, two different names always refer to different entities.

iff $\mathcal{I} \models C(a)$; (2) $\forall R \in N_R$, $\mathcal{J} \models R(a,b)$ iff $\mathcal{I} \models R(a,b)$; (3) $E_{\mathcal{J}} \subset E_{\mathcal{I}}$ where $E_{\mathcal{F}} = \{\langle a,b\rangle \mid a,b \in N_I \text{ s.t. } \mathcal{F} \models a = b\}$. The Minimal Equality (ME) models, i.e., $Mod_{ME}(\mathcal{K})$, are the models of $\mathcal{K}$ with minimal equality between individual names. Formally, we define

$$Mod_{ME}(\mathcal{K}) = \{\mathcal{I} \in Mod(\mathcal{K}) \mid \nexists \mathcal{J}, \mathcal{J} \in Mod(\mathcal{K}), \mathcal{J} \prec_= \mathcal{I}\}$$

First, we define an *IC-interpretation* $\mathcal{I},\mathcal{U} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I},\mathcal{U}})$ where $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a $\mathcal{SROIQ}$ interpretation and $\mathcal{U}$ is a set of $\mathcal{SROIQ}$ interpretations. The IC-interpretation function $\cdot^{\mathcal{I},\mathcal{U}}$ maps concepts, roles, and individuals as follows:

$$C_a{}^{\mathcal{I},\mathcal{U}} = \{x^{\mathcal{I}} \mid x \in N_I \text{ s.t. } \forall \mathcal{J} \in \mathcal{U}, x^{\mathcal{J}} \in C_a{}^{\mathcal{J}}\},$$
$$R^{\mathcal{I},\mathcal{U}} = \{\langle x^{\mathcal{I}}, y^{\mathcal{I}}\rangle \mid x,y \in N_I \text{ s.t. } \forall \mathcal{J} \in \mathcal{U}, \langle x^{\mathcal{J}}, y^{\mathcal{J}}\rangle \in R^{\mathcal{J}}\},$$
$$a^{\mathcal{I},\mathcal{U}} = a^{\mathcal{I}}, \quad (R^-)^{\mathcal{I},\mathcal{U}} = \{\langle x^{\mathcal{I}}, y^{\mathcal{I}}\rangle \mid \langle y^{\mathcal{I}}, x^{\mathcal{I}}\rangle \in R^{\mathcal{I},\mathcal{U}}\},$$
$$(C \sqcap D)^{\mathcal{I},\mathcal{U}} = C^{\mathcal{I},\mathcal{U}} \cap D^{\mathcal{I},\mathcal{U}}, \quad (\neg C)^{\mathcal{I},\mathcal{U}} = (N_I)^{\mathcal{I}} \setminus C^{\mathcal{I},\mathcal{U}},$$
$$(\geq nR.C)^{\mathcal{I},\mathcal{U}} = \{x^{\mathcal{I}} \mid x \in N_I \text{ s.t. } \#\{y^{\mathcal{I}} \mid \langle x^{\mathcal{I}}, y^{\mathcal{I}}\rangle \in R^{\mathcal{I},\mathcal{U}} \text{ and } y^{\mathcal{I}} \in C^{\mathcal{I},\mathcal{U}}\} \geq n\},$$
$$(\exists R.\text{Self})^{\mathcal{I},\mathcal{U}} = \{x^{\mathcal{I}} \mid x \in N_I \text{ s.t. } \langle x^{\mathcal{I}}, x^{\mathcal{I}}\rangle \in R^{\mathcal{I},\mathcal{U}}\}, \quad \{a\}^{\mathcal{I},\mathcal{U}} = \{a^{\mathcal{I}}\}.$$

where $(N_I)^{\mathcal{I}} = \{x^{\mathcal{I}} \mid x \in N_I\}$, $C_a \in N_C$ (atomic concepts), $R \in N_R$ (atomic roles), $a \in N_I$ (individual names), C, D are concepts.

Then, the satisfaction of axiom α in an IC-interpretation $\mathcal{I},\mathcal{U}$, denoted as $\mathcal{I},\mathcal{U} \models \alpha$, is defined in Table 1. Note that, there are also four kinds of ABox axioms ($C(a)$, $R(a,b)$, $a = b$, $a \neq b$). Their semantics is given by encoding them as TBox axioms ($\{a\} \sqsubseteq C$, $\{a\} \sqsubseteq \exists R.\{b\}$, $\{a\} \sqsubseteq \{b\}$, $\{a\} \sqsubseteq \neg\{b\}$, resp.).

Type	Axiom	Condition on $\mathcal{I},\mathcal{U}$
TBox	$C \sqsubseteq D$	$C^{\mathcal{I},\mathcal{U}} \subseteq D^{\mathcal{I},\mathcal{U}}$
RBox	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I},\mathcal{U}} \subseteq R_2^{\mathcal{I},\mathcal{U}}$
	$R_1 \ldots R_n \sqsubseteq R$	$R_1^{\mathcal{I},\mathcal{U}} \circ \ldots \circ R_n^{\mathcal{I},\mathcal{U}} \subseteq R^{\mathcal{I},\mathcal{U}}$
	$\text{Ref}(R)$	$\forall x \in N_I : \langle x^{\mathcal{I},\mathcal{U}}, x^{\mathcal{I},\mathcal{U}}\rangle \in R^{\mathcal{I},\mathcal{U}}$
	$\text{Irr}(R)$	$\forall x \in N_I : \langle x^{\mathcal{I},\mathcal{U}}, x^{\mathcal{I},\mathcal{U}}\rangle \notin R^{\mathcal{I},\mathcal{U}}$
	$\text{Dis}(R_1, R_2)$	$R_1^{\mathcal{I},\mathcal{U}} \cap R_2^{\mathcal{I},\mathcal{U}} = \emptyset$

Table 1. Axiom satisfactions in IC-interpretation $\mathcal{I},\mathcal{U}$

Given a $\mathcal{SROIQ}$ KB $\mathcal{K}$ and a $\mathcal{SROIQ}$ axiom α, we say $\mathcal{K}$ IC-satisfies α, i.e., $\mathcal{K} \models_{IC} \alpha$, iff $\forall \mathcal{I} \in \mathcal{U}$, $\mathcal{I},\mathcal{U} \models \alpha$, where $\mathcal{U} = Mod_{ME}(\mathcal{K})$. We define an *extended KB* as a pair $\langle \mathcal{K}, \mathcal{C}\rangle$ where $\mathcal{K}$ is a $\mathcal{SROIQ}$ KB interpreted with the standard semantics and $\mathcal{C}$ is a set of $\mathcal{SROIQ}$ axioms interpreted with the IC semantics. We say that $\langle \mathcal{K}, \mathcal{C}\rangle$ is valid if $\forall \alpha \in \mathcal{C}, \mathcal{K} \models_{IC} \alpha$, otherwise there is an IC violation. Note that, the IC-satisfaction has a closed world flavor: given an atomic concept C (R resp.), if $\mathcal{K} \not\models_{IC} C(a)(R(a,b)$ resp.) then we conclude

$\mathcal{K} \models_{IC} \neg C(a)$ ($\neg R(a, b)$ resp.). We have verified [18] that this CWA [5] and the weak UNA addresses the issues caused by the OWA and absence of UNA of OWL standard semantics, therefore enabling OWL as an IC language.

4.2 IC Evaluation

In this section, we describe a decision procedure for IC evaluation. That is, deciding if a KB IC-satisfies an IC axiom. First, we present the translation rules from IC axioms to DCQ$^{\mathbf{not}}$ queries. Then we show that IC evaluation can be reduced to corresponding DCQ$^{\mathbf{not}}$ query answering. Due to space limitation we do not introduce DCQ$^{\mathbf{not}}$ here. Please refer to [18] for more details.

The translation rules are similar in the spirit to the Lloyd-Topor transformation [21] but instead of rules we generate DCQ$^{\mathbf{not}}$ queries. The idea behind the translation is to translate an IC axiom into a query such that when the IC is violated the query is true. In other words, whenever the answer set of the query is not empty, we can conclude that the IC is violated. The translation contains two operators: $\mathcal{T}_c$ for translating concepts and $\mathcal{T}$ for translating axioms:

$$\mathcal{T}_c(C_a, x) := C_a(x)$$

$$\mathcal{T}_c(\neg C, x) := \mathbf{not}\, \mathcal{T}_c(C, x)$$

$$\mathcal{T}_c(C_1 \sqcap C_2, x) := \mathcal{T}_c(C_1, x) \wedge \mathcal{T}_c(C_2, x)$$

$$\mathcal{T}_c(\geq nR.C, x) := \bigwedge_{1 \leq i \leq n} (R(x, y_i) \wedge \mathcal{T}_c(C, y_i)) \bigwedge_{1 \leq i < j \leq n} \mathbf{not}\, (y_i = y_j)$$

$$\mathcal{T}_c(\exists R.\mathrm{Self}, x) := R(x, x)$$

$$\mathcal{T}_c(\{a\}, x) := (x = a)$$

$$\mathcal{T}(C_1 \sqsubseteq C_2) := \mathcal{T}_c(C_1, x) \wedge \mathbf{not}\, \mathcal{T}_c(C_2, x)$$

$$\mathcal{T}(R_1 \sqsubseteq R_2) := R_1(x, y) \wedge \mathbf{not}\, R_2(x, y)$$

$$\mathcal{T}(R_1 \ldots R_n \sqsubseteq R) := R_1(x, y_1) \wedge \ldots R_n(y_{n-1}, y_n) \wedge \mathbf{not}\, R(x, y_n)$$

$$\mathcal{T}(\mathtt{Ref}(R)) := \mathbf{not}\, R(x, x)$$

$$\mathcal{T}(\mathtt{Irr}(R)) := R(x, x)$$

$$\mathcal{T}(\mathtt{Dis}(R_1, R_2)) := R_1(x, y) \wedge R_2(x, y)$$

where C_a is an atomic concept, $C_{(i)}$ is a concept, $R_{(i)}$ is a role, a is an individual name, x and $y_{(i)}$ are variables.

Example 1 *Suppose* $\alpha : \mathtt{Product} \sqsubseteq \exists \mathtt{hasProducer}.\mathtt{Producer}$, *then we have:*

$\mathcal{T}(\mathtt{Product} \sqsubseteq \exists \mathtt{hasProducer}.\mathtt{Producer})$

$:= \mathcal{T}_c(\mathtt{Product}, x) \wedge \mathbf{not}\, \mathcal{T}_c(\exists \mathtt{hasProducer}.\mathtt{Producer}, x)$

$:= \mathtt{Product}(x) \wedge \mathbf{not}\, (\mathtt{hasProducer}(x, y) \wedge \mathcal{T}_c(\mathtt{Producer}, y))$

$:= \mathtt{Product}(x) \wedge \mathbf{not}\, (\mathtt{hasProducer}(x, y) \wedge \mathtt{Producer}(y))$

We now obtain the main decision procedure for IC evaluation:

---

[5] With CWA, a statement is inferred to be false if it is not known to be true, which is the opposite of OWA.

Theorem 1 *Given an extended KB $\langle \mathcal{K}, \mathcal{C} \rangle$ with expressivity $\langle \mathcal{SRI}, \mathcal{SROIQ} \rangle$ ($\langle \mathcal{SROIQ}, \mathcal{SROI} \rangle$ resp.), $\mathcal{K} \models_{IC} \alpha$ iff the query answers $\mathbf{A}(\mathcal{T}(\alpha), \mathcal{K})$ are empty, i.e., $\mathcal{K} \not\models \mathcal{T}(\alpha)$, where $\alpha \in \mathcal{C}$.*

We require $\langle \mathcal{K}, \mathcal{C} \rangle$ to be less expressive than $\langle \mathcal{SRI}, \mathcal{SROIQ} \rangle$ or $\langle \mathcal{SROIQ}, \mathcal{SROI} \rangle$ because, otherwise, the disjunctive individual (in)equivalence axioms in $\mathcal{K}$ and the cardinality restrictions in $\mathcal{C}$ will cause some problematic interactions such that the IC axioms are satisfied in different ways at different interpretations of $\mathcal{K}$ and IC evaluation can not be reduced to query answering. Please refer to [22] for more details.

4.3   IC Violation Explanation and Repair

By Theorem 1, $\mathcal{K}$ violates an IC axiom α if $\mathcal{K} \models \mathcal{T}(\alpha)$. To explain the violations of α, we just need to justify why the query entailment $\mathcal{K} \models \mathcal{T}(\alpha)$ holds. In this section, we present our recent work on justification of conjunctive query answers [23] and show how to use the justifications to explain and repair IC violations.

Given a DCQ$^{\mathbf{not}}$ entailment $\mathcal{K} \models^\sigma Q$ where $\mathcal{K}$ is a $\mathcal{SROIQ}$ DL KB, Q is a DCQ$^{\mathbf{not}}$, σ is an assignment mapping query variables to individuals, a justification for $\mathcal{K} \models^\sigma Q$ is $\mathcal{J} = \langle \mathcal{J}_+, \mathcal{J}_- \rangle$ where $\mathcal{J}_+$ and $\mathcal{J}_-$ are the positive and negative justifications respectively such that: (1) $\mathcal{J}_+ \subseteq \mathcal{K}$, $\mathcal{J}_+ \models^\sigma Q$; $\forall S \subseteq \mathcal{K}$, $\mathcal{J}_+ \cup S \models^\sigma Q$; $\forall \mathcal{J}' \subset \mathcal{J}_+$, $\mathcal{J}' \not\models^\sigma Q$. (2) $\mathcal{K} \cup \mathcal{J}_-$ is consistent; $\mathcal{K} \cup \mathcal{J}_- \not\models^\sigma Q$; $\forall T \supseteq \mathcal{J}_-$, $\mathcal{K} \cup T \not\models^\sigma Q$; $\forall \mathcal{J}' \subset \mathcal{J}_-$, $\mathcal{K} \cup \mathcal{J}' \models^\sigma Q$. That is, the existence of $\mathcal{J}_+$ in $\mathcal{K}$ and the absence of $\mathcal{J}_-$ from $\mathcal{K}$ are sufficient for $\mathcal{K} \models^\sigma Q$ to hold.

Example 2 *Suppose we have the following KB $\mathcal{K}$ and IC axiom α*

$\mathcal{K} = \{\texttt{Toy} \sqsubseteq \texttt{Product}, \texttt{ToyProducer} \sqsubseteq \texttt{Producer}, \texttt{Toy}(p_1), \texttt{Product}(p_1),$
$\quad\quad \texttt{hasProducer}(p_1, s_1), \texttt{hasProducer}(p_1, s_2)\}$

$\alpha : \texttt{Product} \sqsubseteq \exists\texttt{hasProducer}.\texttt{Producer}.$

Then $\mathcal{T}(\alpha)$ is $Q \leftarrow \texttt{Product}(x) \wedge \mathbf{not}\,(\texttt{hasProducer}(x, y) \wedge \texttt{Producer}(y))$ and query answer $\mathbf{A}(Q, \mathcal{K}) = \{\sigma : x \rightarrow p_1, y \rightarrow s_1, \sigma' : x \rightarrow p_1, y \rightarrow s_2, \sigma'' : x \rightarrow p_1, y \rightarrow p_1\}$ is not empty indicating α is violated.

To explain the violation of α, we first compute the justifications for $\mathcal{K} \models Q$. We have proposed algorithms for justification computation [23]. According to the algorithms, the justifications for $\mathcal{K} \models Q$ w.r.t. σ are:

$\mathcal{J}_1 = \langle \mathcal{J}_{1+}, \mathcal{J}_{1-} \rangle = \langle \{\texttt{Product}(p_1)\}, \{\texttt{Producer}(s_1)\} \rangle,$
$\mathcal{J}_2 = \langle \mathcal{J}_{2+}, \mathcal{J}_{2-} \rangle = \langle \{\texttt{Toy}(p_1), \texttt{Toy} \sqsubseteq \texttt{Product}\}, \{\texttt{Producer}(s_1)\} \rangle.$

Since the entailment $\mathcal{K} \models^\sigma Q$ contributes to the non-emptiness of query answers $\mathbf{A}(Q, \mathcal{K})$, the violation of α can be explained by $\mathcal{J}_1$ and $\mathcal{J}_2$. That is, α is violated because p_1 is a product (by $\mathcal{J}_{1+}/\mathcal{J}_{2+}$) and p_1 does not have a known producer (by $\mathcal{J}_{1-}/\mathcal{J}_{2-}$).

To repair the above violations, we need to invalidate $\mathcal{K} \models Q$. According to the definition of query entailment justification, $\mathcal{K} \models Q$ holds because the existence of positive justifications and the absence of negative justifications. Therefore, we

can either remove a minimal hitting set (mhs) [6] of positive justifications from $\mathcal{K}$ such that no positive justification exists in $\mathcal{K}$, or add a negative justification to $\mathcal{K}$. For instance, the positive justifications for $\mathcal{K} \models^\sigma Q$ and its mhs are:

$$\mathcal{J}_+{}^{All} = \{\mathcal{J}_{1+}, \mathcal{J}_{2+}\} = \{\{\texttt{Product}(p_1)\}, \{\texttt{Toy}(p_1), \texttt{Toy} \sqsubseteq \texttt{Product}\}\},$$

$$mhs(\mathcal{J}_+{}^{All}) = \{H_1, H_2\} = \{\{\texttt{Product}(p_1), \texttt{Toy} \sqsubseteq \texttt{Product}\}, \{\texttt{Product}(p_1), \texttt{Toy}(p_1)\}\}.$$

So we can update $\mathcal{K}$ to one of the three KBs: (1) $\mathcal{K}_1' = \mathcal{K} \setminus H_1$; (2) $\mathcal{K}_2' = \mathcal{K} \setminus H_2$; (3) $\mathcal{K}_3' = \mathcal{K} \cup \mathcal{J}_{1-}$. Similarly we can invalidate $\mathcal{K} \models Q$ w.r.t. σ' and σ''. It is easy to verify that after the updates to $\mathcal{K}$ constraint α is satisfied.

5 Further Research Plan

The research that remains to be done includes:

- Proposing a decision procedure for IC evaluation for the fully expressive $\mathcal{SROIQ}$ KBs. The existing query answering-based decision procedure only works when KBs are less expressive than $\mathcal{SROIQ}$ KBs. To address this issue, we need to explore other approaches such as Tableau-based approaches.
- Optimizing the algorithms for query answer justifications. The computation complexity of the algorithms is a linear function in most cases except that the complexity follows the power law when the queries include negation of conjunction. We need to utilize some optimizations to solve this problem.
- Finishing the implementation and evaluating the integrity issues in semantic web instance data by employing the IC modeling, evaluation, and explanation approaches that we have proposed. The evaluation should cover functionality, performance, and scalability aspects.

6 Conclusions

Our work addresses the issues of supporting IC evaluation for instance data on the semantic web. We propose an alternative semantics for OWL that adopts CWA and a weakened UNA thus enabling OWL to serve as an IC language. Further, we describe a decision procedure for IC evaluation and an approach for IC violation explanation and repair. With the contributions of this work, users can easily discover and fix the issues in the data. This enables them to obtain data that is checked and appropriate for their uses, thus improving the process of interactions between various parties on the web.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American **284**(5) (2001) 34–43

---

[6] Given a collection of sets, a set which intersects all sets in the collection in at least one element is called a hitting set. The minimal hitting set is the hitting set of smallest size.

2. Gmez-Prez, A.: Some Ideas and Examples to Evaluate Ontologies. In: AIA'1995. 299
3. Gmez-Prez, A.: Evaluation of Ontologies. Int. J. Intell. Syst. **16**(3) (2001) 391–409
4. Baclawski, K., Matheus, C.J., Kokar, M.M., Letkowski, J., Kogut, P.A.: Towards a Symptom Ontology for Semantic Web Applications. In: ISWC'2004. 650–667
5. Tao, J., Ding, L., Bao, J., McGuiness, D.: Instance Data Evaluation for Semantic Web-Based Knowledge Management Systems. In: HICSS'42. 1–10
6. Bechhofer, S., Volz, R.: Patching Syntax in OWL Ontologies. In: ISWC'2004. 668–682
7. Parsia, B., Sirin, E., Kalyanpur, A.: Debugging OWL ontologies. In: WWW'2005. 633–640
8. Wang, H., Horridge, M., Rector, A., Drummond, N., Seidenberg, J.: Debugging OWL-DL Ontologies: A Heuristic Approach. In: ISWC'2005. 745–757
9. Plessers, P., Troyer, O.D.: Resolving Inconsistencies in Evolving Ontologies. In: ESWC'2006. 200–214
10. McGuinness, D.L., Fikes, R., Rice, J., Wilder, S.: An Environment for Merging and Testing Large Ontologies. In: KR'2000. 483–493
11. scar Corcho, Gmez-Prez, A., Gonzlez-cabero, R., Surez-figueroa, C.: ODEval: A Tool for Evaluating RDF(S), DAML+OIL, and OWL Concept Taxonomies. In: AIAI'2004. 369–382
12. Codd, E.F.: A Relational Model of Data for Large Shared Data Banks. Commun. ACM **13**(6) (1970) 377–387
13. Smith, M.K., Welty, C., McGuiness, D.: OWL Web Ontology Language Guide (2004)
14. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining Answer Set Programming with Description Logics for the Semantic Web. AI **172**(12-13) 1495–1539
15. Motik, B.: A Faithful Integration of Description Logics with Logic Programming. In: IJCAI'2007. 477–482
16. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: EQL-Lite: Effective First-Order Query Processing in Description Logics. In: IJCAI'2007. 274–279
17. Motik, B., Horrocks, I., Sattler, U.: Bridging the Gap between OWL and Relational Databases. In: WWW'2007. 807–816
18. Tao, J., Sirin, E., Bao, J., McGuiness, D.: Integrity Constraints in OWL. In: AAAI'2010
19. Motik, B., Patel-Schneider, P.F., Grau, B.C.: OWL 2 Web Ontology Language Direct Semantics (2009)
20. Horrocks, I., Kutz, O., Sattler, U.: The Even More Irresistible SROIQ. In: KR'2006. 57–67
21. Lloyd, J.W.: Foundations of Logic Programming. (1987)
22. Tao, J., Sirin, E., Bao, J., McGuiness, D.: Integrity Consrtiants in OWL. Technical report, Rensselaer Polytechnic Institute
23. Tao, J., Sirin, E., McGuiness, D.: Towards Justification of Conjunctive Query Answers in Description Logics. Technical report, Rensselaer Polytechnic Institute