Integrity Constraints for the Semantic Web: An OWL 2 DL Extension

Thesis Defense
Jiao Tao

Tetherless World Constellation,
Department of Computer Science,
Rensselaer Polytechnic Institute,
Troy, NY 12180, USA.
The recommended web ontology language (OWL) can be used as an integrity constraint language.

Query answering-based solutions can be made to work for data integrity checking and explanation.

A framework for evaluation of data integrity is possible; furthermore, this thesis provides one.
Increasing amount of data published using RDF. Is the data ready for use?

Data Validation: checking data for correctness and integrity
- Syntax
- Semantics
- Integrity

Integrity Constraint (IC): originated in databases to enforce the legal state of data; also needed on the Semantic Web

IC Validation: a straightforward solution (??)
- Treat the axioms in ontologies such as `rdfs:domain`, `rdfs:range`, `owl:someValuesFrom`, `owl:allValuesFrom`, `owl:cardinality` etc., as ICs
- Validate ICs using OWL reasoners
Lack of Integrity Constraint Support

ICs are **not** supported by the standard semantics of OWL.OWL adopts Open World Assumptions (OWA) and non-Unique Name Assumptions (nUNA).

However, Closed World Assumptions (CWA) and Unique Name Assumptions (UNA) are needed for IC validation.

| OWA vs. CWA | OWA: a statement not known to be true is **unknown**  
  | CWA: a statement not known to be true is **false** |
| nUNA vs. UNA | nUNA: different names may refer to the **same** object  
  | UNA: different names always refer to **different** objects |

---

1 which is based on fundational logics, i.e., description logics (DL).
Lack of Integrity Constraint Support cont.

Example

Suppose we have a wine instance $W$. We might assume the knowledge about wine locations is complete, and express it as a constraint that says “every wine has a location” using the following axiom in wine ontology

$$\text{Wine} \sqsubseteq \exists \text{locatedIn.Region}$$

Due to the OWA, not having a known location for the wine instance $W$ does not mean it is false that $W$ has a location. Therefore, standard OWL reasoning does not yield an IC violation.
Lack of Integrity Constraint Support cont.

Example

Suppose a wine instance $W$ has two makers $m_1$ and $m_2$. One might express the constraint “a wine has exactly one maker” using the following axiom in wine ontology

$$\text{Wine} \sqsubseteq \exists = 1 \text{hasMaker}$$

Due to the nUNA, $m_1$ and $m_2$ may refer to the same object. Once we apply DL reasoning, $m_1$ will be inferred to be the same as $m_2$. Therefore, there are no IC violations. However in many cases such as this, we do NOT want to apply DL reasoning, but to detect a violation since the wine instance $W$ has two makers $m_1$ and $m_2$ which are not known to be the same.
Goal: Integrity Constraints for the Semantic Web

Using OWL as an IC language

- Provide a semantics for OWL that adopts CWA and UNA thus supports ICs
- Design a solution to check ICs over OWL KBs
- Design a solution to compute the justifications for IC violations and the repairs for the violations
Preliminaries

- Description Logics (DL \textit{SROIQ})
- Epistemic Description Logics (EDLs)
- Conjunctive Query Answering (DCQ$^{not}$)
Integrity Constraints in Databases

- Used to represent and enforce the legal states of databases which can be viewed as KBs represented as a set of FOL sentences
- Different views on ICs
  - Conventional view: ICs are FOL sentences [20, 21, 15]
    - IC satisfaction as KB consistencies [13]
    - IC satisfaction as KB entailments (FOL entailment) [21, 15]
  - Epistemic View: ICs are about “what the knowledge base knows”, i.e., ICs are epistemic FOL sentences, Reiter et. al. [22, 23]
    - IC satisfaction as KB entailments (epistemic entailment) [19]
  - Another view: ICs are FOL queries, Lloyd and Topor et. al. [14]
    - IC satisfaction as query answering or KB entailments (non-monotonic entailment)
## Integrity Constraint Extension to the Semantic Web

<table>
<thead>
<tr>
<th>IC Representation</th>
<th>IC Validation</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use DLs as IC Languages</strong> [18], [26]</td>
<td>$\langle T, A \rangle =<em>{MM} IC, \quad Ans(IC</em>{SPARQL}, IC_{DL})$</td>
<td>unintuitive in some scenarios, or no IC semantics</td>
</tr>
<tr>
<td><strong>SPARQL Query-based [29]</strong></td>
<td>$Ans(\mathcal{K}_{DL})$</td>
<td>no IC semantics; limited capability</td>
</tr>
<tr>
<td><strong>Epistemic Query-based [3, 2]</strong></td>
<td>$Ans(\mathcal{K}_{FOL})$</td>
<td>unknown complexity results of answering epistemic queries in more expressive DLs</td>
</tr>
<tr>
<td><strong>ADLs-based [5, 25, 6, 28]</strong></td>
<td>ADL KB, i.e., ${ \mathcal{K}, IC_{ADL} }$, satisfiability</td>
<td>only for less expressive DLs, use strict UNA</td>
</tr>
<tr>
<td><strong>Hybrid KB-based [7, 10, 17]</strong></td>
<td>Rules</td>
<td>Hybrid KB entailments (complexity)</td>
</tr>
</tbody>
</table>
Explanation in DLs

- Explanation of DL reasoning
  - Early work: proof-oriented approach for explanation of the general subsumption reasoning, McGuinness et al. [16]
  - Explanation of axiom entailments
    - Axiom level: justification notion, justification computation algorithms (including a HST-based algorithm), Kalyanpur et. al. [12]
    - Finer level: Kalyanpur et. al. [11], Horridge et. al. [8]

- Explanation of query answers in DLs
  - Explaining query answers over ABoxes in DL lite, Borgida et. al. [1]
Integrity Constraints for the Semantic Web: An OWL 2 DL Extension

Standard Semantics + IC Semantics

OWL KB + IC Axioms

K + C

Ans(Q,K) + Q

Step 2: Execute Q

Step 1: Translation
Q = T (C)
Integrity Constraints for the Semantic Web: An OWL 2 DL Extension cont.

- Proposed an IC semantics for OWL 2 DL that adopts the CWA and the nUNA thus interpreting OWL axioms as ICs.
- Designed a sound and complete solution to IC validation by reduction to conjunctive query answering.
- Designed a solution to explanation and repair of IC violations based on explanations of answers to conjunctive queries.
- Developed a prototype implementation; Used it to evaluate some relatively well published Semantic Web instance data from real applications: Wine, MLSO, CEDAR, BCODMO, Datagov; The results show the effectiveness of this approach in detecting and fixing defects in Semantic Web data.
Integrity Constraints for the Semantic Web: An OWL 2 DL Extension cont.

References:

- J. Tao, E. Sirin, J. Bao, and D. L. McGuinness., Integrity Constraints in OWL, AAAI’2010. (*full paper*) [31]
- J. Tao, Adding Integrity Constraints to the Semantic Web for Instance Data Evaluation, ISWC 2010. (*full paper*) [27]
Standard Semantics vs. Integrity Constraint Semantics

- **Standard Semantics**
  - OWA
  - nUNA
  - OW reasoning

- **IC Semantics**
  - CWA
  - Weak UNA
  - CW IC validation

**OWL 2 DL**: based on DL $\mathcal{SROIQ}$ [9]

**IC semantics**: Motivated by the epistemic nature of ICs

Similar to the semantics of epistemic description logics (EDLs) [4, 24, 3]²

²EDLs extend DLs with K operator which intuitively refers to the KB knows.
IC-interpretation

- A **SROIQ vocabulary** $V = (N_C, N_R, N_I)$
- An **IC-interpretation** $\mathcal{I}, \mathcal{U} = (\Delta^\mathcal{I}, \cdot^\mathcal{I}, \cdot^\mathcal{U})$ w.r.t a vocabulary $V$

$$
A^\mathcal{I}, \mathcal{U} = \{ x^\mathcal{I} | x \in N_I \text{ s.t. } \forall J \in \mathcal{U}, x^J \in A^J \}
$$

$$
R^\mathcal{I}, \mathcal{U} = \{ \langle x^\mathcal{I}, y^\mathcal{I} \rangle | x, y \in N_I \text{ s.t. } \forall J \in \mathcal{U}, \langle x^J, y^J \rangle \in R^J \}
$$

$$
a^\mathcal{I}, \mathcal{U} = a^\mathcal{I} \ldots
$$

where $A \in N_C$, $R \in N_R$, $a \in N_I$.

- Epistemic nature of IC-interpretations:
  - $A^\mathcal{I}, \mathcal{U}$: interpretation of individual names that are **known** to be instances of $A$ in $\mathcal{U}$
  - $R^\mathcal{I}, \mathcal{U}$: interpretation of pairwise individual names that are **known** to be related by $R$ in $\mathcal{U}$
IC-interpretation cont.

IC-interpretation $\mathcal{I}, \mathcal{U}$ extended to inverse roles and complex concepts

$(R^-)^{\mathcal{I}, \mathcal{U}} = \{ \langle x^\mathcal{I}, y^\mathcal{I} \rangle \mid \langle y^\mathcal{I}, x^\mathcal{I} \rangle \in R^{\mathcal{I}, \mathcal{U}} \}$

$(C \cap D)^{\mathcal{I}, \mathcal{U}} = C^{\mathcal{I}, \mathcal{U}} \cap D^{\mathcal{I}, \mathcal{U}}$

$(-C)^{\mathcal{I}, \mathcal{U}} = (N_I)^{\mathcal{I}} \setminus C^{\mathcal{I}, \mathcal{U}}$

$(\geq nR.C)^{\mathcal{I}, \mathcal{U}} = \{ x^\mathcal{I} \mid x \in N_I \text{ s.t. } \# \{ y^\mathcal{I} \mid \langle x^\mathcal{I}, y^\mathcal{I} \rangle \in R^{\mathcal{I}, \mathcal{U}}, y^\mathcal{I} \in C^{\mathcal{I}, \mathcal{U}} \} \geq n \}$

$(\leq nR.C)^{\mathcal{I}, \mathcal{U}} = \{ x^\mathcal{I} \mid x \in N_I \text{ s.t. } \# \{ y^\mathcal{I} \mid \langle x^\mathcal{I}, y^\mathcal{I} \rangle \in R^{\mathcal{I}, \mathcal{U}}, y^\mathcal{I} \in C^{\mathcal{I}, \mathcal{U}} \} \leq n \}$

$(\exists R.\text{Self})^{\mathcal{I}, \mathcal{U}} = \{ x^\mathcal{I} \mid x \in N_I \text{ s.t. } \langle x^\mathcal{I}, x^\mathcal{I} \rangle \in R^{\mathcal{I}, \mathcal{U}} \}$

$\{ a \}^{\mathcal{I}, \mathcal{U}} = \{ a^\mathcal{I} \}$

where $(N_I)^{\mathcal{I}} = \{ x^\mathcal{I} \mid x \in N_I \}$ is the interpretation of $N_I$. 
Weak Unique Name Assumptions (Weak UNA)

- **Weak UNA**: different names denote different objects by default, unless their equality is required for the satisfiability of KB.

- Formalize weak UNA via **Minimal Equality (ME) models**, i.e., the models with minimal equality relations between individual names:

\[
\text{Mod}_{\text{ME}}(\mathcal{K}) = \{\mathcal{I} \in \text{Mod}(\mathcal{K}) \mid \not\exists \mathcal{I}, \mathcal{J} \in \text{Mod}(\mathcal{K}), \mathcal{J} \ll \mathcal{I}\}
\]
IC-satisfaction

- An IC-interpretation $\mathcal{I}, \mathcal{U}$ satisfies $\alpha$, i.e., $\mathcal{I}, \mathcal{U} \models \alpha$, if

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

where $C$, $D$ are concepts, $R_{(i)}$ is a role.

- A SROIQ KB $\mathcal{K}$ IC-satisfies a SROIQ axiom $\alpha$, i.e., $\mathcal{K} \models_{IC} \alpha$, iff $\forall \mathcal{I} \in \mathcal{U}, \mathcal{I}, \mathcal{U} \models \alpha$, where $\mathcal{U} = \text{Mod}_{ME}(\mathcal{K})$

- Closed world flavor of IC-satisfaction:
  - if $\mathcal{K} \not\models_{IC} C(a)$ then $\mathcal{K} \models_{IC} \neg C(a)$, where $C \in N_C$, $a \in N_I$
  - if $\mathcal{K} \not\models_{IC} R(a, b)$ then $\mathcal{K} \models_{IC} \neg R(a, b)$, where $R \in N_R$, $a, b \in N_I$
An extended KB is a pair $\langle K, C \rangle$

- $K$ is a SROIQ KB interpreted with the standard semantics
- $C$ is a set of SROIQ axioms interpreted with the IC semantics
- $\langle K, C \rangle$ is valid if $\forall \alpha \in C, K \models IC \alpha$, otherwise there is an IC violation
Idea: By Reduction to Conjunctive Query Answering

Extended KB \(<K, C>\) is valid?

\[\text{Ans}(Q,K)\]

Step 2: Execute \(Q\)

Step 1: Translation
\(Q = T(C)\)

- Translate IC axioms to conjunctive queries (DCQ\textsuperscript{not})\(^3\): \(C \Rightarrow T(C)\)
- Reduce IC validation to query answering: \(\langle K, C \rangle\) is valid iff \(\forall \alpha \in C, \text{Ans}(T(\alpha), K) = \emptyset\)

\(^3\)DCQ\textsuperscript{not}: distinguished conjunctive queries with NAF
Translation Rules

- $\mathcal{T}_c$: translating concepts

\[
\begin{align*}
\mathcal{T}_c(C_a, x) & := C_a(x) \\
\mathcal{T}_c(\neg C, x) & := \text{not } \mathcal{T}_c(C, x) \\
\mathcal{T}_c(C_1 \sqcap C_2, x) & := \mathcal{T}_c(C_1, x) \land \mathcal{T}_c(C_2, x) \\
\mathcal{T}_c(\geq nR.C, x) & := \bigwedge_{1 \leq i \leq n} (R(x, y_i) \land \mathcal{T}_c(C, y_i)) \land \bigwedge_{1 \leq i < j \leq n} \text{not } (y_i = y_j) \\
\mathcal{T}_c(\leq nR.C, x) & := \text{not } (\mathcal{T}_c(\geq (n + 1)R.C, x)) \\
\mathcal{T}_c(\exists R.C, x) & := \mathcal{T}_c(\geq 1R.C, x) = R(x, y) \land \mathcal{T}_c(C, y) \\
\mathcal{T}_c(\forall R.C, x) & := \text{not } (R(x, y) \land \text{not } \mathcal{T}_c(C, y)) \\
\mathcal{T}_c(\exists R.Self, x) & := R(x, x) \\
\mathcal{T}_c(\{a\}, x) & := (x = a)
\end{align*}
\]
Translation Rules cont.

- $\mathcal{T}$: translating axioms

\[
\mathcal{T}(C_a, x) := C_a(x)
\]
\[
\mathcal{T}(\neg C, x) := \text{not } \mathcal{T}(C, x)
\]
\[
\mathcal{T}(C_1 \cap C_2, x) := \mathcal{T}(C_1, x) \land \mathcal{T}(C_2, x)
\]
\[
\mathcal{T}(\geq nR.C, x) := \bigwedge_{1 \leq i \leq n} (R(x, y_i) \land \mathcal{T}(C, y_i)) \land \bigwedge_{1 \leq i < j \leq n} \text{not } (y_i = y_j)
\]
\[
\mathcal{T}(\leq nR.C, x) := \text{not } (\mathcal{T}(\geq (n + 1)R.C, x))
\]
\[
\mathcal{T}(\exists R.C, x) := \mathcal{T}(\geq 1R.C, x) = R(x, y) \land \mathcal{T}(C, y)
\]
\[
\mathcal{T}(\forall R.C, x) := \text{not } (R(x, y) \land \text{not } \mathcal{T}(C, y))
\]
\[
\mathcal{T}(\exists R.\text{Self}, x) := R(x, x)
\]
\[
\mathcal{T}(\{a\}, x) := (x = a)
\]
Given the IC axiom \( \text{Wine} \sqsubseteq \exists \text{locatedIn.Region} \), applying the above translation rules \( T \), we get:

\[
T(\text{Wine} \sqsubseteq \exists \text{locatedIn.Region}) := T_c(\text{Wine}, x) \land \neg T_c(\exists \text{locatedIn.Region}, x)
\]

\[
:= \text{Wine}(x) \land \neg (\text{locatedIn}(x, y) \land T_c(\text{Region}, y))
\]

\[
:= \text{Wine}(x) \land \neg (\text{locatedIn}(x, y) \land \text{Region}(y))
\]
Theorem

Given an extended KB $\langle K, C \rangle$ with expressivity $\langle SRI, SROIQ \rangle$ or $\langle SROIQ, SROI \rangle$, we say that $K \models IC \alpha$ iff $K \not\models T(\alpha)$, where $\alpha \in C$.

- Note that
  - either $K$ is free of nominals and cardinality restrictions, thus no disjunctive (in)equality axioms
  - or $C$ is free of cardinality restrictions $\geq nR.C$ where $n \geq 2$

- Proof: soundness and completeness
Algorithm: IsValid($\langle K, C \rangle$)

- **Input:** $\langle K, C \rangle$
- **Process:**
  - while ($C \neq \emptyset$)
    - Get $\alpha \in C$, $C = C \setminus \alpha$, translate $\alpha$ to query $T(\alpha)$
    - If $Ans(T(\alpha), K) \neq \emptyset$, then return FALSE, else continue
    - If $C = \emptyset$, then return TRUE, else return FALSE
- **Output:** TRUE: $\langle K, C \rangle$ is valid / FALSE: otherwise
Idea: Conjunctive Query Answering-based Solutions

**IC Violation Explanation**

Why is \(<K,C>\) invalid?

i.e., \(\exists \alpha \in C, \text{Ans}(T(\alpha), K) \neq \emptyset\)

**IC Violation Repair**

How to change \(K\) s.t. \(<K,C>\) is valid?

i.e., \(\forall \alpha \in C, \text{Ans}(T(\alpha), K) = \emptyset\)

**CQ Ans. Exp.**

Why \(\exists \tilde{\alpha} \in \text{Ans}(T(\alpha), K)\)?

**CQ Entail. Just.**

Why is \(K |\models T(\alpha)\) true w.r.t. assignment \(\delta\)?
Problem definition: Given a DL SROIQ KB \( \mathcal{K} \), a DCQ^\text{not} query \( Q(\vec{x}) \), and query answers \( \text{Ans}(Q(\vec{x}), \mathcal{K}) = \{ \vec{t}_1, \vec{t}_2, \ldots \} \)
- Positive answer explanation: why \( \vec{t}_i \) is an answer of \( Q(\vec{x}) \)?
- Negative answer explanation: why \( \vec{t}' \) is not an answer of \( Q(\vec{x}) \)?

Why is it a challenging problem?
- Query \( Q(\vec{x}) \) may include NAF
- When \( Q(\vec{x}) \) contains a negation of conjunction
  - Many possible reasons for a conjunction to be evaluated as false
  - Variables can be mapped to any individual names in \( \mathcal{K} \)
  - Need to answer, given all individual names that query variables might be mapped to, why none of the possible reasons is true
By the semantics of $DCQ^{not}$, a tuple $\bar{t}_i \in Ans(Q(\bar{x}), \mathcal{K})$ because there exist some assignment $\sigma : \bar{x} \rightarrow \bar{t}_i$ s.t. the $\mathcal{K} \models Q(\bar{x})$ w.r.t. $\sigma$ holds.

Explain why $\bar{t}_i$ is an answer of $Q(\bar{x})$: find out the assignment $\sigma$ then explain why the query entailment $\mathcal{K} \models Q(\bar{x})$ holds w.r.t. $\sigma$

$\implies$ query entailment justification
Query Entailment Justification

Given a query entailment $\mathcal{K} \models^\sigma Q$ where $\mathcal{K}$ is a DL $\text{SROIQ}$ KB, $Q$ is a DCQ$^{\text{not}}$ query, $\sigma$ is an assignment:

1. $\mathcal{J}_+$ is a positive justification for $\mathcal{K} \models^\sigma Q$ if $\mathcal{J}_+$ satisfies conditions (1, 2, 3), otherwise $\mathcal{J}_+ = \emptyset$.
   
   1. $\mathcal{J}_+ \subseteq \mathcal{K}$, $\mathcal{J}_+ \models^\sigma Q$;
   2. $\forall S \subseteq \mathcal{K}, \mathcal{J}_+ \cup S \models^\sigma Q$;
   3. $\forall \mathcal{J}' \subset \mathcal{J}_+, \mathcal{J}' \not\models^\sigma Q$.

2. $\mathcal{J}_-$ is a negative justification for $\mathcal{K} \models^\sigma Q$ if $\mathcal{J}_-$ satisfies conditions (1, 2, 3, 4, 5), otherwise $\mathcal{J}_- = \emptyset$.
   
   1. $\mathcal{K} \cap \mathcal{J}_- = \emptyset$, $\mathcal{K} \cup \mathcal{J}_-$ is consistent;
   2. $\mathcal{K} \cup \mathcal{J}_- \not\models^\sigma Q$;
   3. $\forall T \supseteq \mathcal{J}_-, \mathcal{K} \cup T \not\models^\sigma Q$;
   4. $\forall \mathcal{J}' \subset \mathcal{J}_-, \mathcal{K} \cup \mathcal{J}' \models^\sigma Q$;
   5. $\mathcal{J}_-$ is $\sigma(q)$ if $Q \leftarrow \text{not } q$ where $q$ is an atomic query atom.

3. A justification $\mathcal{J}$ for $\mathcal{K} \models^\sigma Q$ is $\mathcal{J} = \langle \mathcal{J}_+, \mathcal{J}_- \rangle$. 
Reduce Explanation of Conjunctive Query Answers to Query Entailment Justification

**Algorithm**: ExplainQAnswer(\(\mathcal{K}, Q(\bar{x}), \bar{a}\))

- **Input**: \(\mathcal{K}, Q(\bar{x}), \bar{a}\)
- **Process**: 
  - Find out assignments \(\sigma : \bar{x} \rightarrow \bar{a}\) s.t. \(\mathcal{K} \models^\sigma Q(\bar{x})\) holds
  - For each assignment \(\sigma\)
    - Apply \(\sigma\) to the query body of \(Q\) to get \(\sigma(Q)\)
    - Recursively break down \(\sigma(Q)\) into \(A_1 \land \ldots \land A_n\)
    - Compute justifications \(J_i\) for each atom \(A_i\)
    - Generate justifications via **combinations** of \(J_i\)
- **Output**: justifications

**Theorem**

Let \(\mathcal{K}\) be a \(\text{SROIQ KB}\), \(Q(\bar{x})\) be a \(\text{DCQ}^{\text{not}}\) query, \(\bar{a}\) be an answer of \(Q(\bar{x})\) over \(\mathcal{K}\), the above algorithm returns all explanations why \(\bar{a}\) is an answer of \(Q(\bar{x})\) over \(\mathcal{K}\).
Recall: given \( \langle \mathcal{K}, \mathcal{C} \rangle \), \( \mathcal{K} \) IC-satisfies \( \alpha \in \mathcal{C} \) iff \( \text{Ans}(\mathcal{T}(\alpha), \mathcal{K}) = \emptyset \).

Therefore, \( \alpha \) is violated because \( \text{Ans}(\mathcal{T}(\alpha), \mathcal{K}) \neq \emptyset \). That is, there exist some tuples, i.e., \( \bar{a} \), in the answer set.

To explain the violation of \( \alpha \) is to explain why \( \bar{a} \) is an answer of \( \mathcal{T}(\alpha) \).
Example

\( \mathcal{K} = \{ \text{Zinfandel} \sqsubseteq \text{Wine}, \text{Wine}(p), \text{Zinfandel}(p), \text{locatedIn}(p, a) \} , \)
\( \alpha : \text{Wine} \sqsubseteq \exists \text{locatedIn.Region}, \)
\( Q(x) \leftarrow \text{Wine}(x) \land \text{not}(\text{locatedIn}(x, y) \land \text{Region}(y)). \)

\( Q(x) \) has answer \( (p) \), so \( \mathcal{K} \) violates \( \alpha \); To explain the violation is to explain why \( (p) \) is an answer of \( Q(x) \), i.e., why \( \mathcal{K} \models Q(x) \) w.r.t. \( \sigma : x \rightarrow p, y \rightarrow a. \)

\( J_{+}^{\text{All}} = \{ J_{1+}, J_{2+} \} = \{ \{ \text{Wine}(p) \}, \{ \text{Zinfandel}(p), \text{Zinfandel} \sqsubseteq \text{Wine} \} \} \)
\( J_{-}^{\text{All}} = \{ J_{-} \} = \{ \{ \text{Region}(a) \} \} \)
\( J_{1} = \langle J_{1+}, J_{-} \rangle = \langle \{ \text{Wine}(p) \}, \{ \text{Region}(a) \} \rangle \)
\( J_{2} = \langle J_{2+}, J_{-} \rangle = \langle \{ \text{Zinfandel}(p), \text{Zinfandel} \sqsubseteq \text{Wine} \}, \{ \text{Region}(a) \} \rangle \)

\( \mathcal{K} \) violates \( \alpha \) because \( p \) is a wine (\( J_{1+}/J_{2+} \)) and \( p \) misses a location (\( J_{-} \)) that is known to be a region.
IC violations can be repaired by invalidating query entailments

A query entailment holds due to existence of $J^+$ and absence of $J^-$

Therefore, by removing $J^+$ from or adding $J^-$ to the KB, violations can be fixed

To have a minimal impact on the KB, just need to remove a Minimal Hitting Set (MHS)\(^4\) of all $J^+$

Conclusion: given a query entailment $\mathcal{K} \models^\sigma Q$, suppose $J^+^{All}$ and $J^-^{All}$ are the collections of all positive and negative justifications resp., $\mathcal{K} \models^\sigma Q$ can be invalidated by updating $\mathcal{K}$ to:

- $\mathcal{K}' = \mathcal{K} \setminus m$ where $m$ is a MHS of $J^+^{All}$;
- Or $\mathcal{K}' = \mathcal{K} \cup J^-$ where $J^- \in J^-^{All}$.

---

\(^4\) Given a collection of sets, a set which contains at least one element from each set in the collection is called a hitting set (HS). A HS is minimal (MHS) if no proper subset of it is a HS.
In previous example, the MHSs of $J^+_{\text{All}}$ are:

- $H_1 = \{\text{Wine}(p), \text{Zinfandel} \sqsubseteq \text{Wine}\}$
- $H_2 = \{\text{Wine}(p), \text{Zinfandel}(p)\}$

To repair the violation, we can update $K$ to:

- $K'_1 = K \setminus H_1 = \{\text{Zinfandel}(p), \text{locatedIn}(p, a)\}$,
- $K'_2 = K \setminus H_2 = \{\text{Zinfandel} \sqsubseteq \text{Wine}, \text{locatedIn}(p, a)\}$,
- $K'_3 = K \cup J^- = \{\text{Zinfandel} \sqsubseteq \text{Wine}, \text{Zinfandel}(p), \text{Wine}(p), \text{locatedIn}(p, a), \text{Region}(a)\}$
Implementation

- Inference materializer
- IC converter
- IC validator
- IC violation explainer
- IC violation repairer
## Evaluation of Instance Data - Datasets

<table>
<thead>
<tr>
<th>Instance Data</th>
<th>Pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wine</td>
<td><a href="http://wineagent.tw.rpi.edu/restaurants/">http://wineagent.tw.rpi.edu/restaurants/</a></td>
</tr>
<tr>
<td>CEDAR</td>
<td><a href="http://escience.rpi.edu/ontology/vsto/2/0/cedar.owl">http://escience.rpi.edu/ontology/vsto/2/0/cedar.owl</a></td>
</tr>
<tr>
<td>BCODMO</td>
<td><a href="http://escience.rpi.edu/ontology/BCO-DMO/1/0/">http://escience.rpi.edu/ontology/BCO-DMO/1/0/</a></td>
</tr>
<tr>
<td>Datagov</td>
<td><a href="http://logd.tw.rpi.edu/vocab/Dataset">http://logd.tw.rpi.edu/vocab/Dataset</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://purl.org/twc/vocab/conversion/MetaDataset">http://purl.org/twc/vocab/conversion/MetaDataset</a></td>
</tr>
</tbody>
</table>

**Table:** Instance data to evaluate
## Evaluation of Instance Data - ICs

<table>
<thead>
<tr>
<th>Instance Data</th>
<th>ICs from Ontologies</th>
<th>Additional ICs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wine</td>
<td>8 [167]</td>
<td>N/A</td>
</tr>
<tr>
<td>MLSO</td>
<td>35 [58]</td>
<td>7</td>
</tr>
<tr>
<td>CEDAR</td>
<td>30 [47]</td>
<td>N/A</td>
</tr>
<tr>
<td>CEDAR Time</td>
<td>3 [47]</td>
<td>12</td>
</tr>
<tr>
<td>BCODMO</td>
<td>7 [47]</td>
<td>2</td>
</tr>
<tr>
<td>Datagov</td>
<td>N/A</td>
<td>4 (^5)</td>
</tr>
</tbody>
</table>

**Table:** ICs for instance data

- Use OWL restrictions in referenced ontologies as ICs \(^6\)
- Generate more if needed

\(^5\) The integrity constraints were generated as a result of S2S listing requirements.

\(^6\) Although any legal RDFS and OWL axioms may be used as ICs, our analysis focused only on OWL restrictions as those were the only axioms our application representatives had identified.
## ICs for Wine

### ICs from onto.

- Wine \( \subseteq 1\text{hasBody.} \top \),
- Wine \( \subseteq 1\text{hasColor.} \top \),
- Wine \( \subseteq 1\text{hasFlavor.} \top \),
- Wine \( \subseteq 1\text{hasSugar.} \top \),
- Wine \( \subseteq \forall\text{hasBody.WineBody} \),
- Wine \( \subseteq \forall\text{hasColor.WineColor} \),
- Wine \( \subseteq \forall\text{hasSugar.WineSugar} \),
- Wine \( \subseteq \forall\text{hasFlavor.WineFlavor} \)

### Additional ICs

N/A

**Table:** ICs for wine data
ICs for MLSO

Ref. document
ICs for CEDAR

Ref. document
## ICs for CEDAR Time

<table>
<thead>
<tr>
<th>ICs from onto.</th>
<th>Additional ICs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset $\subseteq$ 1 isFromInstrument,</td>
<td>Dataset $\subseteq$ 1 isFromInstrumentOperatingMode,</td>
</tr>
<tr>
<td>Dataset $\subseteq$ 1 hasContainedParameter,</td>
<td>Dataset $\subseteq$ $\exists$ hasName.xsd:string,</td>
</tr>
<tr>
<td>Dataset $\subseteq$ $\exists$ hasDateTimeCoverage.DateTimeInterval,</td>
<td>Dataset $\subseteq$ $\exists$ hasDateTimeCoverage.DateTimeInterval,</td>
</tr>
<tr>
<td>DateTimeDescription $\subseteq$ $\exists$ year.xsd:gYear,</td>
<td>DateTimeDescription $\subseteq$ $\exists$ month.xsd:gMonth,</td>
</tr>
<tr>
<td>DateTimeDescription $\subseteq$ $\exists$ day.xsd:gDay,</td>
<td>DateTimeDescription $\subseteq$ $\exists$ minute.xsd:nonNegativeInteger,</td>
</tr>
<tr>
<td>DateTimeDescription $\subseteq$ $\exists$ hour.xsd:nonNegativeInteger,</td>
<td>DateTimeDescription $\subseteq$ $\exists$ second.xsd:decimal,</td>
</tr>
<tr>
<td>DateTimeInterval $\subseteq$ $\exists$ hasBeginning.Instant,</td>
<td>DateTimeInterval $\subseteq$ $\exists$ hasEnd.Instant,</td>
</tr>
<tr>
<td>Instant $\subseteq$ $\exists$ inDateTime.DateTimeDescription,</td>
<td>Instant $\subseteq$ $\exists$ inXSDDateTime.xsd:dateTime</td>
</tr>
</tbody>
</table>

**Table:** ICs for CEDAR Time data
### ICs for BCODMO

#### ICs from onto.

<table>
<thead>
<tr>
<th>Class</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Award</td>
<td>1\text{hasAwardNumber}</td>
</tr>
<tr>
<td>DataSet</td>
<td>1\text{hasDatasetName}</td>
</tr>
<tr>
<td>Deployment</td>
<td>1\text{hasDeploymentId}</td>
</tr>
<tr>
<td>DeploymentDataset</td>
<td>\leq 1\text{fromDeployment}</td>
</tr>
<tr>
<td>DeploymentDatasetCollection</td>
<td>\forall\text{hasInventory}.</td>
</tr>
<tr>
<td>DeploymentDataset</td>
<td></td>
</tr>
<tr>
<td>Person</td>
<td>1\text{hasPersonFirstName}</td>
</tr>
<tr>
<td>Person</td>
<td>1\text{hasPersonLastName}</td>
</tr>
</tbody>
</table>

#### Additional ICs

| DeploymentDataset | \exists\text{hasTimeCoverage}.\top,                                        |
| DeploymentDataset | \exists\text{hasParamter}.\top.                                             |

**Table:** ICs for BCODMO data
## ICs for Datagov

<table>
<thead>
<tr>
<th>ICs from onto.</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Additional ICs</strong></td>
<td></td>
</tr>
<tr>
<td>AbstractDataset ⊑∃title.rdfs:Literal,</td>
<td></td>
</tr>
<tr>
<td>AbstractDataset ⊑∃description.rdfs:Literal,</td>
<td></td>
</tr>
<tr>
<td>AbstractDataset ⊑∃source_agency.⊤,</td>
<td></td>
</tr>
<tr>
<td>AbstractDataset ⊑∃dataset_category.rdfs:Literal</td>
<td></td>
</tr>
</tbody>
</table>

**Table:** ICs for Datagov data
Evaluation of Instance Data - Results

<table>
<thead>
<tr>
<th>Instance Data</th>
<th>ICs Violated</th>
<th>Violations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wine</td>
<td>6</td>
<td>51</td>
</tr>
<tr>
<td>MLSO</td>
<td>1</td>
<td>53</td>
</tr>
<tr>
<td>CEDAR</td>
<td>6</td>
<td>111</td>
</tr>
<tr>
<td>CEDAR Time</td>
<td>3</td>
<td>804</td>
</tr>
<tr>
<td>BCODMO</td>
<td>2</td>
<td>2546</td>
</tr>
<tr>
<td>Datagov</td>
<td>4</td>
<td>892</td>
</tr>
</tbody>
</table>

Table: Results of evaluation on instance data

- A large number of violations
- Violations in the data may cause application failures
- When the data integrity is critical severe problems may happen
Violations in CEDAR and CEDAR Time data

- **CEDAR**
  - Among the 30 ICs, 6 ICs are violated
  - Totally, there are 111 violations
    - Some DataProduct such as HeightVsTimePlot and TimeSeriesPlot miss plotted parameters
    - Some Parameter such as TimeDependentParameter, AuroralBoundaryIndex, AltitudeDependentParameter miss time or altitude coordinates
    - The super physical domains of some SurfaceLine are not Surface

- **CEDAR Time**
  - Among the 15 ICs, 3 ICs are violated
  - Totally, there are 804 violations
    - Some Dataset have more than one value for property isFromInstrument
    - Some Dataset have more than one value for property isFromInstrumentOperatingMode
    - Some Dataset miss values for property hasContainedParameter
Violations in BCODMO data

- Among the 9 ICs, 2 ICs violated
- Totally, there are 2546 violations
  - Some DeploymentDataset miss time coverage
  - Some DeploymentDataset miss parameters
Violations in Datagov data

- All 4 ICs are violated
- The ICs are violated by the same batch of AbstractDataset
  - Missing values for title
  - Missing values for description
  - Missing values for source_agency
  - Missing values for dataset_category
- Totally, there are 892 violations
Evaluation of Instance Data - Discussion

- Restrictions vs. ICs
  - Different experts considered different treatment of restrictions vs. ICs
  - A large number of restrictions in the ontologies (≥ 50% for PML, VSTO) identified as ICs
  - Need to be very careful to distinguish inference axioms and IC axioms

- Ontology evaluation vs. instance data evaluation
  - Some axioms (PML:6, VSTO:6, BCODMO:2) in the ontologies are suggested to be updated/removed
    - VSTO: individual EarthSurface is mistyped as SurfaceRegion, it should be typed as Surface; a Dataset can connect to Instrument, Index, Model, and one connection at most; Atmosphere have at least one of the three states ElectronState, NeutralState, IonState; there is no connection between Wavelength and time coordinates, etc.
    - BCODMO: a Deployment can have more than one platforms; a DeploymentDataset can be from multiple instruments

- Instance data evaluation can help identify potential ontology changes and thus support ontology evaluation
Mapping ICs in databases to OWL IC axioms

- When the data migrated from databases to the Semantic Web, ICs in databases also need to be migrated.
- For example, in BCODMO, more than 100 constraints in BCODMO database; only a small portion modeled as OWL restrictions; same category of ICs are differently modeled; some ICs are not appropriately modeled.
- Need a correct and consistent approach to map ICs in databases to OWL IC axioms.
- A preliminary proposal is given (for MySQL databases)\(^7\)
Performance Analysis - Approach

- Datasets: 10 LUBM datasets representing 1 to 10 universities
- Metrics: validation time, justification time (minimal justification time)
- Reasoning settings: referenced ontologies loading turned on/off

<table>
<thead>
<tr>
<th></th>
<th>1U</th>
<th>2U</th>
<th>3U</th>
<th>4U</th>
<th>5U</th>
<th>6U</th>
<th>7U</th>
<th>8U</th>
<th>9U</th>
<th>10U</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>103K</td>
<td>237K</td>
<td>348K</td>
<td>494K</td>
<td>646K</td>
<td>748K</td>
<td>914K</td>
<td>1036K</td>
<td>1163K</td>
<td>1316K</td>
</tr>
</tbody>
</table>

**Table:** LUBM datasets
ICs: chosen 5 different types of ICs - aiming to cover a variety of restriction types
- Vary in size and complexity
- Behave differently (will point out in the following evaluation slides)

<table>
<thead>
<tr>
<th>IC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC₁</td>
<td>Publication ⊑ Books ⊑ JournalArticle ⊑ ConferencePaper ⊑ TechnicalReport,</td>
</tr>
<tr>
<td>IC₂</td>
<td>Publication ⊑ ∃publicationDate.⊤,</td>
</tr>
<tr>
<td>IC₃</td>
<td>ResearchGroup ⊑ ∃researachProject.⊤,</td>
</tr>
<tr>
<td>IC₄</td>
<td>FullProfessor ⊑ ∀teacher0f.GraduateCourse,</td>
</tr>
<tr>
<td>IC₅</td>
<td>AssistantProfessor ⊑ ≥ 3teacher0f.⊤</td>
</tr>
</tbody>
</table>

Table: ICs for performance analysis
Performance Analysis - Validation Time

- Validation time linearly increases as the dataset size increases.
- Take longer time when the referenced ontologies loading turned on (d.t. larger outputs from inferences).
- Almost same time for different ICs (d.t. similar query execution time).

Validation time affected by the dataset size, reasoning setting.
Performance Analysis - Validation Time cont.
Performance Analysis - Justification Time

- Justification time linearly increases as the dataset size increases
- IC₄ takes the longest time (d.t. more calls of axiom entailment justifications)

Justification time is affected by the dataset size, reasoning setting, and ICs.
Performance Analysis - Justification Time cont.

Justification Time: Off

Datasets

IC1
IC2
IC3
IC4
IC5

Time(s)

1U 2U 3U 4U 5U 6U 7U
Conclusions

- Proposed an IC semantics for OWL 2 DL that adopts the CWA and the nUNA thus interpreting OWL axioms as ICs.
- Designed a sound and complete solution to IC validation by reduction to conjunctive query answering.
- Designed a solution to explanation and repair of IC violations based on explanations of answers to conjunctive queries.
- Developed a prototype implementation; Used it to evaluate some relatively well published Semantic Web instance data from real applications: Wine, MLSO, CEDAR, BCODMO, Datagov; The results show the effectiveness of this approach in detecting and fixing defects in Semantic Web data.
Future work

- The weak UNA in the IC semantics is similar to the minimal model semantics where the equality relations are treated as congruence relations and minimized. Further research about the relationship between them, and, if possible, to generalize the IC semantics.

- Query-answering based IC validation solutions work for most cases of OWL 2 DL extended KBs. For more expressive corner cases, we need alternative solutions.

- Optimize IC violation justification:
  - Optimize query answer explanation algorithms: in the presence of negation of conjunction, need heuristics to decrease the searching space of possible combinations.
  - Optimize justification minimization: DL reasoning invoked to remove redundant axioms, to improve the performance, incremental reasoning may help.
  - Seek alternative DL axiom entailment justification services.
Thanks for your attention!

Questions?
Instance Data

- **Wine**: wine data used by the recommender system Wine Agent which picks the most appropriate wines for meals based on the knowledge of wines and foods described using wine ontology.
- **MLSO (Mauna Loa Solar Observatory)**: solar corona data described using VSTO (Virtual Solar Terrestrial Observatory), PML and others ontologies.
- **CEDAR (Coupling, Energetics and Dynamics of Atmospheric Regions)**: upper atmosphere data from the described using VSTO and other ontologies.
- **BCODMO (Biological and Chemical Oceanography Data Management Office)**: marine biogeochemical, ecological and oceanographic data represented with BCODMO ontology.
- **Datagov**: government data published on or converted from Data.gov and other websites.
Violations in Wine data

- Among the 8 ICs, 6 ICs are violated
- Totally, there are 51 violations
  - Some individuals used as values for property hasColor are not typed as WineColor, such as Golden, Pink, Purple, Yellow
  - Misusage of the WineBody instance Medium and the WineFlavor instance Moderate
  - Weak used as the value for property WineFlavor is not typed as WineFlavor
  - Some wine instances miss values for the four properties, i.e., color, sugar, body, flavor.
Violations in MLSO data

- Only one ontology-independent IC violated
- This IC requires that each Information instance, which is a conclusion of some NodeSet, to have some values for hasFormat property.
- In the given MLSO data, all such Information instances miss hasFormat assertions. Totally, there are 53 violations.
Time increases when the referenced ontology loading switched from off to on (d.t. longer axiom entailment justification time since axioms in the ref. ontologies may contribute to the entailments); $IC_1$ and $IC_2$ increases the most (d.t. classes/properties in ICs that have much more entailments when referenced ontology loading turned on).

Minimal justification time for $IC_5$ is much longer than its justification time (d.t. non-minimality).

Justification time is affected by the dataset size, reasoning setting, and ICs.
Performance Analysis - Justification Time cont.

![Graph showing Justification Time (1U): On vs. Off]

- IC1
- IC2
- IC3
- IC4
- IC5

Time(s)

- Off
- On
Performance Analysis - Justification Time cont.

Justification Time (1U): J. vs. Min. J.

ICs

<table>
<thead>
<tr>
<th>IC1</th>
<th>IC2</th>
<th>IC3</th>
<th>IC4</th>
<th>IC5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Time (s)

0 1 2 3 4 5 6 7 8 9 10

J. Min. J.
Performance Analysis - Justification Time cont.

J. Time and Min. J. Time: Off

Datasets
1U 2U 3U 4U 5U 6U 7U

Time(s)
0 5 10 15 20 25 30 35 40 45 50

IC1
IC1-min
IC2
IC2-min
IC3
IC3-min
IC4
IC4-min
IC5
IC5-min
Performance Analysis - Justification Time cont.

J. Time and Min. J. Time: On

Datasets
- IC1
- IC1-min
- IC2
- IC2-min
- IC3
- IC3-min
- IC4
- IC4-min
- IC5
- IC5-min

Time(s)


Description Logics of Minimal Knowledge and Negation as Failure.  

T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, and H. Tompits.  
Combining Answer Set Programming with Description Logics for the Semantic Web.  

M. Horridge, B. Parsia, and U. Sattler.  
Laconic and Precise Justifications in OWL.  

The Even More Irresistible SROIQ.  

I. Horrocks, B. Motik, R. Rosati, and U. Sattler.  
Can OWL and Logic Programming Live Together Happily Ever After?  
A. Kalyanpur, B. Parsia, and B. C. Grau.

A. Kalyanpur, B. Parsia, M. Horridge, and E. Sirin.
Finding All Justifications of OWL DL Entailments.

R. A. Kowalski.
Logic for Data Description.

Making Prolog more Expressive.

A Basis for Deductive Database Systems.
D. L. McGuinness and A. T. Borgida.
Explaining Subsumption in Description Logics.

B. Motik.
A Faithful Integration of Description Logics with Logic Programming.

B. Motik, I. Horrocks, and U. Sattler.
Bridging the Gap between OWL and Relational Databases.

J.-M. Nicolas and H. Gallaire.
Data Base: Theory vs. Interpretation.

J.-M. Nicolas and K. Yazdanian.
Integrity Checking in Deductive Data Bases.

R. Reiter.
Towards A Logical Reconstruction of Relational Database Theory.

R. Reiter.
On Integrity Constraints.

R. Reiter.
On Asking What A Database Knows.

R. Rosati.
On the Semantics of Epistemic Description Logics.
In *Description Logics*, pages 185–188, 1996.

R. Rosati.
Autoepistemic Description Logics.

E. Sirin and J. Tao.
Towards Integrity Constraints in OWL.

J. Tao.
Adding Integrity Constraints to the Semantic Web for Instance Data Evaluation.

J. Tao, L. Ding, J. Bao, and D. L. McGuinness.
Characterizing and Detecting Integrity Issues in OWL Instance Data.

J. Tao, L. Ding, and D. L. McGuinness.
Instance Data Evaluation for Semantic Web-Based Knowledge Management Systems.

J. Tao and D. L. McGuinness.
Towards Explanation of Conjunctive Query Answers in Description Logics.

J. Tao, E. Sirin, J. Bao, and D. L. McGuinness.
Integrity Constraints in OWL.