Integrity Constraints in OWL

Jiao Tao[1], **Evren Sirin**[2], **Jie Bao**[1], **Deborah L. McGuinness**[1]
[1] Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
[2] Clark&Parsia, Washington, DC 20001, USA

Abstract

In many data-centric applications, we would like to use OWL to express constraints that must be satisfied by instance data. However, the Open World Assumption (OWA) in OWL's standard semantics, combined with the absence of the Unique Name Assumption (UNA), makes it difficult to use OWL in this way. What triggers constraint violations in closed world systems leads to new inferences in standard OWL systems. In this paper, we present an Integrity Constraint (IC) semantics for OWL axioms that is based on Closed World Assumption (CWA) and weak UNA to address this issue. Ontology modelers can choose which axioms will be interpreted with IC semantics, thus combine open world reasoning with closed world constraint validation in a flexible way. We also show that IC validation can be reduced to query answering under certain conditions. Finally, we briefly describe our prototype implementation based on the OWL reasoner Pellet.

1. Introduction

Web Ontology Language (OWL) is an expressive ontology language based on Description Logics (DL)[1] with sound and complete reasoning algorithms. The semantics of OWL addresses distributed knowledge representation scenarios where complete knowledge about the domain cannot be assumed. And it has the following features:

- 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.

However, these characteristics make it difficult to use OWL for data validation purposes in real-world applications where complete knowledge can be assumed for some or all parts of the domain.

Example 1. *Suppose we have a KB $\mathcal{K}$ containing information about products in a company's inventory as follows:*

$\mathcal{K} = \{\texttt{Product}(p)\}$

One might add the following axiom to express the constraint "every product is produced by a producer":

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

In this example, due to OWA, *not* having a known producer for p does not cause a logical inconsistency. Therefore, we cannot use α to detect (or prevent) that a product is added to the KB without the producer information.

Example 2. *Suppose the inventory KB $\mathcal{K}$ looks like this:*

$\mathcal{K} = \{\texttt{Product}(\texttt{p}), \texttt{hasProducer}(p, m_1), \texttt{hasProducer}(p, m_2)\}$

One might and the following axiom to express the constraint "a product has at most one producer":

$\alpha : \texttt{Product} \sqsubseteq \leq 1 \texttt{hasProducer}.\top$

Since m_1 and m_2 are not explicitly defined to be different from each other, they will be inferred to be same due to cardinality restriction. However, in many cases, the reason to use functional properties is not to draw this inference, but to detect an inconsistency. When the information about instances are coming from multiple sources we cannot always assume explicit inequalities will be present.

In these scenarios, there is a strong need to use OWL as an Integrity Constraint (IC) language with closed world semantics. That is, we would like to adopt OWA without UNA for parts of the domain where we have incomplete knowledge, and to use the Closed World Assumption (CWA)[2] with UNA otherwise. This calls for the ability to combine the open world reasoning of OWL with closed world constraint validation. One approach to achieve this combination is to couple OWL with rule-based formalisms and express ICs as rules without heads as in (Eiter et al. 2008) and (Motik 2007). However, there are several issues with this approach: not only would ontology developers have to deal with two different formalisms to model the domain, but the encoding of ICs as rules can be complex and unintuitive.

In this paper, we describe an alternative IC semantics for OWL, which enables developers to augment OWL ontologies with IC axioms. Standard OWL axioms in the ontologies are used to compute inferences with open world semantics and ICs are used to validate instance data using closed world semantics. Our goal is to enable efficient data validation with OWL, especially in settings where OWL KBs are integrated with relational databases and ICs are needed to enforce the individual names in KBs to have some *known* values.

Copyright © 2010, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]Throughout the paper we use the terms OWL and DL interchangeably.

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

We show that IC validation can be reduced to query answering when the KB expressivity is $\mathcal{SRI}$ or the constraint expressivity is $\mathcal{SROI}$. The queries generated from ICs can be expressed in the SPARQL query language allowing existing OWL reasoners to be used for IC validation easily.

2. Related Work

Several existing proposals for integrating ICs with OWL express ICs with formalisms such as rules (Eiter et al. 2008) (Motik 2007), or epistemic queries (Calvanese et al. 2007). In this paper, we focus on approaches that reuse OWL as an IC language. Our closest related work is a proposal by Motik et al. (Motik, Horrocks, and Sattler 2007) based on a minimal Herbrand model semantics of OWL: here, a constraint axiom is satisfied if all minimal Herbrand models satisfy it. This approach may result in counterintuitive results or significant modeling burden in the following cases.

First, existential individuals can satisfy constraints, which is not desirable for closed world data validation.

Example 3. *Consider the KB $\mathcal{K}$ that contains a product instance and its unknown producer which is an existential individual, and the constraint α that every product has a known producer:*

$\mathcal{K} = \{\texttt{Product}(p), \exists\texttt{hasProducer}.\texttt{Producer}(p)\}$

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

Since p has a producer in every minimal Herbrand model of $\mathcal{K}$, α is satisfied, even though the producer is unknown.

Second, if a constraint 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, but still does not capture the intuition behind the constraint;

Example 4. *Suppose we have a KB $\mathcal{K}$ where there are two possible producers for a product and a constraint α:*

$K = \{\texttt{Product}(p), (\exists\texttt{hasProducer}.\{m_1, m_2\})(p), O(p),$
$\texttt{Producer}(m_1), \texttt{Producer}(m_2), O(m_1), O(m_2)\}$

$\alpha : \texttt{Product} \sqsubseteq \exists\texttt{hasProducer}.(\texttt{Producer} \sqcap O)$

The intuition behind constraint α is that the producer of every product should be known. Even though we do not know the producer of p is m_1 or m_1 for sure, α is still satisfied by the semantics of (Motik, Horrocks, and Sattler 2007) because in every minimal Herbrand model there is a producer for p that is also an instance of `Producer` and O.

Third, the disjunctions and ICs may also interact in unexpected ways.

Example 5. *Consider the following KB $\mathcal{K}$ where there are two categories for products and a constraint α defined on one of the categories:*

$\mathcal{K} = \{\texttt{Product} \sqsubseteq \texttt{Category1} \sqcup \texttt{Category2},$
$\texttt{Product}(p)\}$

$\alpha : \texttt{Category1} \sqsubseteq \exists\texttt{categoryType}.\top$

Since we do not know for sure that p belongs to `Category1`, it is reasonable to assume that the constraint α will not apply to p and α will not be violated. However, with (Motik, Horrocks, and Sattler 2007) semantics, α is violated because there is a minimal model where p belongs to `Category1` but it does not have a `categoryType` value.

In this paper, we present a new IC semantics for OWL that overcomes the above issues and enables efficient IC validation for OWL.

3. Preliminaries

3.1 Description Logics $\mathcal{SROIQ}$

In this section, we give a brief description of the syntax and semantics of the Description Logic $\mathcal{SROIQ}$ (Horrocks, Kutz, and Sattler 2006), which is the logical underpinning of OWL 2 (Motik, Patel-Schneider, and Grau 2009). More details can be found in (Horrocks, Kutz, and Sattler 2006). In this paper, to keep the presentation simple, we will focus only on the object domain and not datatypes, a simplified version of the concrete domain (Baader and Hanschke 1991), though the approach described in this paper can easily be extended to handle datatypes.

A $\mathcal{SROIQ}$ vocabulary $V = (N_C, N_R, N_I)$ is a triple where N_C, N_R, and N_I are non-empty and pair-wise disjoint sets of *atomic concepts*, *atomic roles* and *individual names* respectively. The set of $\mathcal{SROIQ}$ roles (roles, for short) is the set $N_R \cup \{R^- \mid R \in N_R\}$, where R^- denotes the inverse of the atomic role R. Concepts are defined inductively using the following grammar:

$C \leftarrow A \mid \neg C \mid C_1 \sqcap C_2 \mid \geq nR.C \mid \exists R.\text{Self} \mid \{a\}$

where $A \in N_C$, $a \in N_I$, $C_{(i)}$ a concept, R a role. For expressions in the form $\geq nR.C$ where $n > 1$ we require R to be a simple role as defined in (Horrocks, Kutz, and Sattler 2006). We use the following standard abbreviations for concept descriptions:

$$\bot = C \sqcap \neg C, \quad \top = \neg\bot, \quad C \sqcup D = \neg(\neg C \sqcap \neg D)$$
$$\exists R.C = (\geq 1\, R.C), \quad \forall R.C = \neg(\exists R.\neg C)$$
$$\geq n\, R = (\geq n\, R.\top), \quad \leq nR.C = \neg(\geq n+1\, R.C)$$
$$\{a_1, \ldots, a_n\} = \{a_1\} \sqcup \cdots \sqcup \{a_n\}.$$

A *$\mathcal{SROIQ}$-interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ w.r.t. a vocabulary $V = (N_C, N_R, N_I)$ consists of a nonempty set $\Delta^\mathcal{I}$ which is the *interpretation domain*, and $.^\mathcal{I}$ which is the *interpretation function* that maps $A \in N_C$ to a subset of $\Delta^\mathcal{I}$, $R \in N_R$ to a subset of $\Delta^\mathcal{I} \times \Delta^\mathcal{I}$, $a \in N_I$ to an element of $\Delta^\mathcal{I}$. The interpretation is extended to inverse roles and complex concepts as follows:

$$(R^-)^\mathcal{I} = \{\langle y, x\rangle \mid \langle x, y\rangle \in R^\mathcal{I}\}$$
$$(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}$$
$$(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$$
$$(\geq nR.C)^\mathcal{I} = \{x \mid \#\{y.\langle x, y\rangle \in R^\mathcal{I} \text{ and } y \in C^\mathcal{I}\} \geq n\}$$
$$(\leq nR.C)^\mathcal{I} = \{x \mid \#\{y.\langle x, y\rangle \in R^\mathcal{I} \text{ and } y \in C^\mathcal{I}\} \leq n\}$$
$$(\exists R.\text{Self})^\mathcal{I} = \{x \mid \langle x, x\rangle \in R^\mathcal{I}\}$$
$$\{a\}^\mathcal{I} = \{a^\mathcal{I}\}$$

where $\#$ denotes the cardinality of a set.

A $\mathcal{SROIQ}$ knowledge base $\mathcal{K}$ is a collection of $\mathcal{SROIQ}$ axioms which are listed in Table 1, where C, D are concepts,

$R_{(i)}$ is a role, a, b are individual names. We say that an interpretation $\mathcal{I}$ w.r.t vocabulary V *satisfies* a $\mathcal{SROIQ}$ axiom α, denoted $\mathcal{I} \models \alpha$, if α can be constructed using the elements in V and the corresponding condition on interpretation $\mathcal{I}$ in Table 1 is satisfied. The interpretation $\mathcal{I}$ is a *model* of the KB $\mathcal{K}$ if it satisfies all the axioms in $\mathcal{K}$. We define $Mod(\mathcal{K})$ to be the set of all $\mathcal{SROIQ}$-interpretations that are models of $\mathcal{K}$. We say $\mathcal{K}$ *entails* α, written as $\mathcal{K} \models \alpha$, if $\mathcal{I} \models \alpha$ for all models $\mathcal{I} \in Mod(\mathcal{K})$.

Type	Axiom	Condition on $\mathcal{I}$
TBox	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RBox	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
	$R_1 \dots R_n \sqsubseteq R$	$R_1^{\mathcal{I}} \circ \dots \circ R_n^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
	$\texttt{Ref}(R)$	$\forall x \in \Delta : \langle x, x\rangle \in R^{\mathcal{I}}$
	$\texttt{Irr}(R)$	$\forall x \in \Delta : \langle x, x\rangle \notin R^{\mathcal{I}}$
	$\texttt{Dis}(R_1, R_2)$	$R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} = \emptyset$
ABox	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
	$R(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$
	$\neg R(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \notin R^{\mathcal{I}}$
	$a = b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$
	$a \neq b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$

Table 1: Axiom satisfactions in $\mathcal{SROIQ}$-interpretation $\mathcal{I}$

Throughout the text, we will use the standard abbreviations for axioms by using $\equiv$ symbol for concept (resp. role) equivalence axioms instead of a pair of concept (resp. role) inclusion axioms, $\texttt{Trans}(R)$ instead of $RR \sqsubseteq R$, and $\texttt{Fun}(R)$ instead of $\top \sqsubseteq \leq 1R$.

3.2 Distinguished Conjunctive Queries (DCQs)

We now describe the syntax and semantics of *distinguished conjunctive queries* (DCQs). Let N_V be a non-empty set of variable names disjoint from N_I, N_C, and N_R. A *query atom* is an ABox axiom where variables can be used in place of individuals. Formally, it is defined as follows:

$$q \leftarrow C(x) \mid R(x, y) \mid \neg R(x, y) \mid x = y \mid x \neq y$$

where $x, y \in N_I \cup N_V$, C is a concept, and R is a role. A *conjunctive query (CQ)* is the conjunction of query atoms:

$$Q \leftarrow q \mid Q_1 \wedge Q_2$$

A DCQ is the CQ containing only *distinguished variables*, i.e., the variables can be mapped to only individual names, i.e., elements from N_I.

The semantics of DCQs are given in terms of interpretations defined in Section 3.1. We define an *assignment* $\sigma : N_V \rightarrow N_I$ to be a mapping from the variables used in the query to individual names in the KB. We define $\sigma(Q)$ to denote the application of an assignment σ to a query Q such that the variables in the query are replaced with individuals according to the mapping. We say a KB $\mathcal{K}$ entails a DCQ query Q with an assignment σ, written as $\mathcal{K} \models^{\sigma} Q$, if:

$$\mathcal{K} \models^{\sigma} q \quad \text{iff} \quad \mathcal{K} \models \sigma(q)$$

$$\mathcal{K} \models^{\sigma} Q_1 \wedge Q_2 \quad \text{iff} \quad \mathcal{K} \models^{\sigma} Q_1 \text{ and } \mathcal{K} \models^{\sigma} Q_2$$

We define the *answers to a query*, $\mathbf{A}(Q, \mathcal{K})$, to be the set of all assignments for which the KB entails the query. That is, $\mathbf{A}(Q, \mathcal{K}) = \{\sigma \mid \mathcal{K} \models^{\sigma} Q\}$. We say that a query is true w.r.t. a KB, denoted $\mathcal{K} \models Q$, if there is at least one answer for the query, and false otherwise.

Note that, the restriction to individual names in query answers of DCQs allows us to use simpler and more efficient query answering algorithms similar to the ones described in (Sirin and Parsia 2006). Using only distinguished variables in queries also changes the query answers as illustrated by the following example.

Example 6. *Suppose we have the following KB and query*

$$\mathcal{K} = \{A \equiv \exists R.\top, A(a), A(b), R(b, c)\}$$

$$Q \leftarrow A(x) \wedge R(x, y).$$

Then, the $\mathbf{A}(Q, \mathcal{K})$ *contains a single assignment* $\{x \rightarrow b, y \rightarrow c\}$. *Note that, there are no answers containing* $\{x \rightarrow a\}$ *query even tough* $R(a, y)$ *is satisfied in all the interpretations of the KB. This is due the fact that* all *the variables in the query need to be mapped to an individual name.*

4. IC Semantics

There has been a significant amount of research to define the semantics of ICs for relational databases, deductive databases, and knowledge representation systems in general. There are several proposals based on KB consistency or KB entailment. Against both of these approaches, Reiter argued that ICs are epistemic in nature and are about "what the knowledge base knows" (Reiter 1988). He proposed that ICs should be epistemic first-order queries that will be asked to a standard KB that does not contain epistemic axioms.

We agree with Reiter in his assessment about the epistemic nature of ICs and believe this is the most appropriate semantics for ICs. In the following section, we describe an alternative IC semantics for OWL axioms, which is similar to how the semantics of epistemic DL $\mathcal{ALCK}$ (Donini et al. 1998) is defined. Then, in Section 4.2 we discuss how the IC semantics addresses the issues explained in Section 1 and Section 2 and enables OWL to be an IC language.

4.1 Formalization

We define an *IC-interpretation* $\mathcal{I}$,$\mathcal{U}$ $= (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I},\mathcal{U}})$ w.r.t a vocabulary V where $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a $\mathcal{SROIQ}$ interpretation w.r.t V and $\mathcal{U}$ = $\mathcal{I}_1$, ..., $\mathcal{I}_n$ is a set where each $\mathcal{I}_k$ = $(\Delta^{\mathcal{I}_k}, \cdot^{\mathcal{I}_k})$ is a $\mathcal{SROIQ}$ interpretation w.r.t. V. The IC-interpretation function $\cdot^{\mathcal{I},\mathcal{U}}$ maps $A \in N_C$ to a subset of $\Delta^{\mathcal{I}}$, $R \in N_R$ to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, $a \in N_I$ to an element of $\Delta^{\mathcal{I}}$ as follows:

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

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

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

According to this definition, $A^{\mathcal{I},\mathcal{U}}$ is the set of objects in $\Delta^{\mathcal{I}}$ that are the mappings of individual names which are instances of A in every $\mathcal{SROIQ}$ interpretation from $\mathcal{U}$. That is, $A^{\mathcal{I},\mathcal{U}}$ represents the interpretation of individual names that are known to be instances of A in $\mathcal{U}$. $R^{\mathcal{I},\mathcal{U}}$ can be understood similarly.

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

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

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

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

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

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

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

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

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

We can see that, different from the $\mathcal{SROIQ}$ interpretation $\mathcal{I}$ of $\neg C$ which is the set of objects in $\Delta^\mathcal{I}$ that are not instances of C in $\mathcal{I}$, the IC-interpretation $\mathcal{I},\mathcal{U}$ of $\neg C$ is the set of objects in $\Delta^\mathcal{I}$ that are mappings of N_I but not instances of C in every $\mathcal{SROIQ}$ interpretation from $\mathcal{U}$.

Note that, although the IC interpretations have some similarities to the interpretations of $\mathcal{ALCK}$ (Donini et al. 1998), there is an important difference. In $\mathcal{ALCK}$, strict UNA is used by the interpretations. However, this is not desirable in our case since it is possible that standard OWL axioms infer that two different names identify the same individual. Therefore the interpretations in $\mathcal{U}$ do *not* adopt strict UNA.

In our IC semantics, we want to adopt a weak form of UNA; that is, 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. This idea is similar to minimal model semantics where equality relation is treated as a congruence relation and minimized.

We formalize this notion of weak UNA by defining Minimal Equality (ME) models. We start by defining the *less than relation w.r.t. equality*, denoted as $\prec_=$, between interpretations. Formally, 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 all of the following conditions hold:

- For every atomic concept C, $\mathcal{J} \models C(a)$ iff $\mathcal{I} \models C(a)$;
- For every atomic role R, $\mathcal{J} \models R(a,b)$ iff $\mathcal{I} \models R(a,b)$;
- $E_\mathcal{J} \subset E_\mathcal{I}$

where $E_\mathcal{I}$ is the set of equality relations between individual names (equality relations, for short) satisfied by interpretation $\mathcal{I}$:

$E_\mathcal{I} = \{\langle a,b\rangle \mid a,b \in N_I \text{ s.t. } \mathcal{I} \models a = b\}$

Here, a and b could be any individual names in $\mathcal{K}$.

$Mod_{ME}(\mathcal{K})$ is the models of $\mathcal{K}$ with minimal equality between individual names. Formally, we define

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

We can see that two different individual names are interpreted as equivalent in $\mathcal{I} \in Mod_{ME}(\mathcal{K})$ only if this equality is necessary to make $\mathcal{I}$ being a model of $\mathcal{K}$. For example, suppose we have the axiom $a = \{b\} \sqcup \{c\}$ in $\mathcal{K}$.

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

Table 2: Axiom satisfactions in IC-interpretation $\mathcal{I},\mathcal{U}$

Then, $\forall \mathcal{I} \in Mod(\mathcal{K})$, one of the following three conditions hold: (1) $a^\mathcal{I} = b^\mathcal{I}, a^\mathcal{I} \neq c^\mathcal{I}$; (2) $a^\mathcal{I} = c^\mathcal{I}, a^\mathcal{I} \neq b^\mathcal{I}$; (3) $a^\mathcal{I} = b^\mathcal{I} = c^\mathcal{I}$. If (1) or (2) holds, then $\mathcal{I} \in Mod_{ME}(\mathcal{K})$ because a has to be interpreted to be equivalent to at least one of b and c to make $\mathcal{I}$ being a model of $\mathcal{K}$. Whereas for case (3), $\mathcal{I} \notin Mod_{ME}(\mathcal{K})$ since the equality relations in $\mathcal{I}$ are not minimal.

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 2, where C, D are concepts, $R_{(i)}$ is a role, a, b are individual names. Note that, when N_I is an empty set, every role is both reflexive and irreflexive, which is different from the conventional interpretations.

Given a $\mathcal{SROIQ}$ KB $\mathcal{K}$ and a $\mathcal{SROIQ}$ axiom α, the IC-satisfaction of α by $\mathcal{K}$, i.e., $\mathcal{K} \models_{IC} \alpha$, is defined as:

$\mathcal{K} \models_{IC} \alpha$ iff $\forall \mathcal{I} \in \mathcal{U}, \mathcal{I},\mathcal{U} \models \alpha$, where $\mathcal{U} = Mod_{ME}(\mathcal{K})$

The above IC-satisfaction has a closed world flavor: (1) when C is an atomic concept, if $\mathcal{K} \not\models_{IC} C(a)$ then we conclude $\mathcal{K} \models_{IC} \neg C(a)$ where a is individual name. (2) when R is a role, if $\mathcal{K} \not\models_{IC} R(a,b)$ then we conclude $\mathcal{K} \models_{IC} \neg R(a,b)$ where a, b are individual names. The first case holds because, if $\mathcal{K} \not\models_{IC} C(a)$ then $\exists \mathcal{I} \in \mathcal{U}$ such that $a^{\mathcal{I},\mathcal{U}} = a^\mathcal{I} \notin C^{\mathcal{I},\mathcal{U}}$. That is, $\exists \mathcal{I} \in \mathcal{U}$ such that $a^\mathcal{I} \notin C^\mathcal{I}$. Therefore, $\forall \mathcal{J} \in \mathcal{U}, a^\mathcal{J} \notin C^{\mathcal{J},\mathcal{U}}$ thus $\forall \mathcal{J} \in \mathcal{U}, a^{\mathcal{J},\mathcal{U}} = a^\mathcal{J} \in (N_I)^\mathcal{J} \setminus C^{\mathcal{J},\mathcal{U}} = (\neg C)^{\mathcal{J},\mathcal{U}}$. So we conclude that $\mathcal{K} \models_{IC} \neg C(a)$. The second case can be similarly proved. These examples show that the IC semantics use CWA for atomic concepts and atomic roles. Failure to satisfy an ABox assertion results in the satisfaction of its negation which is not the case with standard OWL semantics.

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 (Horrocks, Kutz, and Sattler 2006) 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.

4.2 Discussion

It is easy to verify that the IC semantics provides expected results for examples presented in Section 1and Section 2.and enables OWL to be an IC language. In Example 1, we get an IC violation since the IC interpretation of `Product` contains p but the IC interpretation of $(\exists \texttt{hasProducer.Producer})$ is empty. We also get an IC violation for Example 2 because due to the weak UNA m_1 and m_2 are interpreted as different individuals causing the IC interpretation of $(\leq 1\texttt{hasProducer}.\top)$ to be empty. In Example 3 and Example 4 there are also IC violations because the constraint requires the same named producer to exist in every ME model of the KB which is not the case. Since our IC semantics is targeted at individual names, one does not need to use the concept O as in Example 4. In Example 5, there is a model of the KB where p is not an instance of `Category1`, the IC interpretation of `Category1` is empty, therefore the constraint does not apply to p and there is no violation.

The following example shows how weak UNA allows the individuals that are not asserted to be equal to be treated different for constraint validation purposes.

Example 7. *Consider the KB $\mathcal{K}$ and the constraint α:*

$\mathcal{K} = \{C(c), R(c, d_1), R(c, d_2), D(d_1), D(d_2)\}$

$\alpha : C \sqsubseteq \geq 2R.D$

With the weak UNA, d_1 and d_2 are interpreted to be different in every ME model. Therefore, the IC-interpretation of $(\geq 2R.D)$ includes c, and the constraint α is satisfied by $\mathcal{K}$.

Now we illustrate another point regarding disjunctions in constraints.

Example 8. *Suppose we have the KB $\mathcal{K}$ and constraint α:*

$\mathcal{K} = \{C(a), (C_1 \sqcup C_2)(a)\}$

$\alpha : \quad C \sqsubseteq C_1 \sqcup C_2$

Constraint α should be read as "every instance of C should be either a known instance of C_1 or a known instance of C_2". Since we do not know *for sure* whether a belongs to C_1 or C_2, α is expected to be violated by $\mathcal{K}$. Indeed, according to our semantics we get $C^{\mathcal{I},\mathcal{U}} = \{a^{\mathcal{I}}\}$ and $(C_1 \sqcup C_2)^{\mathcal{I},\mathcal{U}} = \emptyset$. Therefore $C^{\mathcal{I},\mathcal{U}} \not\subseteq (C_1 \sqcup C_2)^{\mathcal{I},\mathcal{U}}$ and we conclude there is an IC violation.

If we want to represent the alternative constraint: "every instance of C should be an instance of C_1 or C_2", we can define a new name C' in the KB to substitute $C_1 \sqcup C_2$, thus having the new KB $\mathcal{K}'$ and constraint α' as follows:

$\mathcal{K}' = \{C(a), (C_1 \sqcup C_2)(a), C' \equiv C_1 \sqcup C_2\}$

$\alpha' : C \sqsubseteq C'$

There is no IC violation in this version because now the disjunction is interpreted as standard OWL axioms. As these examples show, we can model the constraints to express different disjunctions in a flexible way.

5. IC Validation

We have defined in Section 4.1that, the extended KB $\langle \mathcal{K}, \mathcal{C} \rangle$ is valid if for every IC axiom in $\mathcal{C}$ is IC-satisfied by $\mathcal{K}$. In this section, we describe how to do IC validation, i.e., check IC-satisfaction by translating constraint axioms to queries with *Negation As Failure (NAF)* operator "**not**"[3]. We start by giving the formal semantics for **not** in DCQs, then describe the translation rules from IC axioms to DCQ$^{\textbf{not}}$ and finally provide a theorem showing that IC validation can be reduced to answering DCQ$^{\textbf{not}}$ under certain conditions.

5.1 DCQ$^{\textbf{not}}$

In Section 3.2 we introduced standard DCQs. However, the expressivity of standard DCQs is not enough to capture the closed world nature of IC semantics. For this reason, we add **not** operator to DCQs to get DCQ$^{\textbf{not}}$ queries. The syntax of DCQ$^{\textbf{not}}$ is defined as follows:

$Q \leftarrow q \mid Q_1 \wedge Q_2 \mid \textbf{not}\, Q$

The semantics of **not** is defined as:

$\mathcal{K} \models^{\sigma} \textbf{not}\, Q \qquad \text{iff} \qquad \nexists \sigma' \text{ s.t. } \mathcal{K} \models^{\sigma'} \sigma(Q)$

And we use the abbreviation $Q_1 \vee Q_2$ for $\textbf{not}\,(\textbf{not}\, Q_1 \wedge \textbf{not}\, Q_2)$. It is easy to see

$\mathcal{K} \models^{\sigma} Q_1 \vee Q_2 \qquad \text{iff} \qquad \mathcal{K} \models^{\sigma} Q_1 \text{ or } \mathcal{K} \models^{\sigma} Q_2$

Recall that, assignment functions σ map variables only to individual names so a brute force way to check if a **not** atom is entailed can be done by enumerating all possible assignments and checking for entailment.

We would also like to emphasize that the above abbreviation $\vee$ in DCQ$^{\textbf{not}}$ is not same as conventional disjunctive queries and interacts with the disjunctions in the KB in a restricted way as the next example shows.

Example 9. *Suppose we have the following KB and query*

$\mathcal{K} = \{A \sqsubseteq B \sqcup C, A(a)\}$

$Q \leftarrow B(x) \vee C(x).$

Then, the $\mathbf{A}(Q, \mathcal{K})$ is empty. This is because the answers to queries $Q_1 \leftarrow B(x)$ and $Q_2 \leftarrow C(x)$ are both empty and their union is empty even tough a is known to be an instance of $B \sqcup C$.

5.2 Translation Rules: from ICs to DCQ$^{\textbf{not}}$

We now present the translation rules from IC axioms to DCQ$^{\textbf{not}}$ queries. The translation rules are similar in the spirit to the Lloyd-Topor transformation (Lloyd 1987) but instead of rules we generate DCQ$^{\textbf{not}}$ queries. The idea behind the translation is to translate a constraint axiom into a query such that when the constraint is violated the KB entails the query. In other words, whenever the answer set of the query is not empty, we can conclude that the constraint is violated.

The translation contains two operators: $\mathcal{T}_c$ for translating concepts and $\mathcal{T}$ for translating axioms. $\mathcal{T}_c$ is a function that takes a concept expression and a variable as input and re-

[3]NAF is widely used in logic programming systems. With NAF, axioms that cannot be proven to be true are assumed to be false

turns a DCQ$^{\textbf{not}}$ query as the result:

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

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

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

$\mathcal{T}_c(\geq nR.C, x) :=$

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

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

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

where C_a is an atomic concept, $C_{(i)}$ is a concept, R is a role, a is an individual, x is an input variable, and $y_{(i)}$ is a fresh variable.

$\mathcal{T}$ is a function that maps a $\mathcal{SROIQ}$ axiom to a DCQ$^{\textbf{not}}$ query as follows:

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

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

$\mathcal{T}(R_1 \ldots R_n \sqsubseteq R) :=$

$R_1(x, y_1) \wedge \ldots R_n(y_{n-1}, y_n) \wedge \textbf{not}\, R(x, y_n)$

$\mathcal{T}(\texttt{Ref}(R)) := \textbf{not}\, R(x, x)$

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

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

$\mathcal{T}(C(a)) := \textbf{not}\, \mathcal{T}_c(C, a)$

$\mathcal{T}(R(a, b)) := \textbf{not}\, R(a, b)$

$\mathcal{T}(\neg R(a, b)) := R(a, b)$

$\mathcal{T}(a = b) := \textbf{not}\, (a = b)$

$\mathcal{T}(a \neq b) := (a = b)$

where $C_{(i)}$ is a concept, $R_{(i)}$ is a role, x, $y_{(i)}$ is a variable, a and b are individuals.

Example 10. *Given the constraint α in Example 1, applying the above translation rules, we get:*

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

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

$:= \texttt{Product}(x) \wedge$

$\textbf{\textit{not}}\, (\texttt{hasProducer}(x, y) \wedge \mathcal{T}_c(\texttt{Producer}, y))$

$:= \texttt{Product}(x) \wedge \textbf{\textit{not}}\, (\texttt{hasProducer}(x, y) \wedge \texttt{Producer}(y))$

5.3 Reducing IC Validation to Answering DCQ$^{\textbf{not}}$

We have discussed in Section 5.2we want to reduce the problem of IC validation to query answering. However, a careful examination of the problem reveals that when both the KB and the constraints use full expressivity of $\mathcal{SROIQ}$, we cannot use the query translation approach in a straightforward way as we show with the next example.

Example 11. *Suppose we have a KB $\mathcal{K}$ and a constraint α:*

$\mathcal{K} = \{D(d), R(d, a), R(d, b), R(d, c), \{a\} \sqsubseteq \{b, c\}\}$

$\alpha : D \sqsubseteq \leq 2R.\top$

In all models of $\mathcal{K}$, a is either interpreted to be equivalent to b or c. Therefore, in all interpretations d has less than 2 R values satisfying the constraint. However, the query translation will yield atoms in the form $\textbf{not}\, (y_1 = y_2)$ which will be true for any individual pair in this KB. As a result, the answer set for this query will include d which incorrectly indicates an IC violation.

The problematic axiom ($\{a\} \sqsubseteq \{b, c\}$) is asserting disjunctive equality between individuals which result in a situation where the IC is satisfied in different ways at different interpretations. This is a very challenging task to tackle so we focus on cases where there will not be an interaction between cardinality constraints and disjunctive (in)equivalence axioms.

We can limit this interaction by either prohibiting cardinality restrictions in ICs or by prohibiting disjunctive (in)equality to appear in KBs. In $\mathcal{SROIQ}$, there are only three ways to infer (in)equality between individuals: using (1) explicit (in)equivalence axioms; (2) nominals (as seen above); and (3) cardinality restrictions. Obviously, explicit ABox assertions cannot be disjunctive so they are not problematic. By excluding nominals and cardinality restrictions from $\mathcal{SROIQ}$, we get the DL $\mathcal{SRI}$.

In what follows, we first describe several lemmas, then present the main theorem which shows that IC validation via query answering is sound and complete for the cases where $\langle \mathcal{K}, \mathcal{C} \rangle$ expressivity is either $\langle \mathcal{SRI}, \mathcal{SROIQ} \rangle$ or $\langle \mathcal{SROIQ}, \mathcal{SROI} \rangle$. Since answers to the queries corresponding to constraint axioms $\mathcal{C}$ are affected by the inferences computed by standard OWL axioms in $\mathcal{K}$, open world reasoning and closed world constraint validation are combined on the extended KB $\langle \mathcal{K}, \mathcal{C} \rangle$.

Lemma 1. *Let $\mathcal{K}$ be a $\mathcal{SROIQ}$ KB, suppose $\mathcal{I} \in Mod(\mathcal{K})$ and $\mathcal{I} \notin Mod_{ME}(\mathcal{K})$, then there is a model $\mathcal{J}$ such that $\mathcal{J} \in Mod_{ME}(\mathcal{K})$ and $\mathcal{J} \prec_= \mathcal{I}$*

Proof: By the definition of $Mod_{ME}(\mathcal{K})$, we know $\exists \mathcal{W}_1 \in Mod(\mathcal{K})$ such that $\mathcal{W}_1 \prec_= \mathcal{I}$. If $\mathcal{W}_1 \in Mod_{ME}(\mathcal{K})$ then we find the model $\mathcal{J} = \mathcal{W}_1$, otherwise $\exists \mathcal{W}_2 \in Mod(\mathcal{K})$ such that $\mathcal{W}_2 \prec_= \mathcal{W}_1$. We continue this process. After finite steps the process terminates and we find n models $\mathcal{W}_i \in Mod(\mathcal{K})(1 \leq i \leq n), \mathcal{W}_n \prec_= \mathcal{W}_{n-1} \ldots \prec_= \ldots \mathcal{W}_2 \prec_= \mathcal{W}_1 \prec_= \mathcal{I}$ and we could not find another model $\mathcal{W}_{n+1}$ such that $\mathcal{W}_{n+1} \in Mod(\mathcal{K}), \mathcal{W}_{n+1} \prec_= \mathcal{W}_n$. This process terminates after at most n steps because $\mathcal{K}$ contains finite individual names and there are at most n models $\mathcal{W}_i$ to satisfy the $\prec_=$ relation. If one of the n models belongs to $Mod_{ME}(\mathcal{K})$ then we find the model $\mathcal{J}$, otherwise none of the n models belongs to $Mod_{ME}(\mathcal{K})$. Then by the definition of $Mod_{ME}(\mathcal{K})$, $\exists \mathcal{W}_{n+1} \in Mod(\mathcal{K}), \mathcal{W}_{n+1} \prec_= \mathcal{W}_n$, which is a contradiction. $\square$

Lemma 2. *Let $\mathcal{K}$ be a $\mathcal{SROIQ}$ KB, $C \in N_C$, $R \in N_R$, and $a, b \in N_I$. Then,*
(1)$\forall \mathcal{I} \in Mod_{ME}(\mathcal{K})$, $a^{\mathcal{I}} \in C^{\mathcal{I}}$ iff $\forall \mathcal{J} \in Mod(\mathcal{K})$, $a^{\mathcal{J}} \in C^{\mathcal{J}}$.
(2)$\forall \mathcal{I} \in Mod_{ME}(\mathcal{K})$, $\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$ iff $\forall \mathcal{J} \in Mod(\mathcal{K})$, $\langle a^{\mathcal{J}}, b^{\mathcal{J}} \rangle \in R^{\mathcal{J}}$.
(3)$\forall \mathcal{I} \in Mod_{ME}(\mathcal{K})$, $a^{\mathcal{I}} = b^{\mathcal{I}}$ iff $\forall \mathcal{J} \in Mod(\mathcal{K})$, $a^{\mathcal{J}} = b^{\mathcal{J}}$.

Proof: We only show the proof for the first case. The proofs for case (2) and (3) are similar.
$\Rightarrow$ Assume to the contrary that $\forall \mathcal{I} \in Mod_{ME}(\mathcal{K})$, $a^{\mathcal{I}} \in C^{\mathcal{I}}$

and $\exists \mathcal{J} \in Mod(\mathcal{K})$, $a^{\mathcal{J}} \notin C^{\mathcal{J}}$ (or $\mathcal{J} \not\models C(a)$). This means $\mathcal{J} \notin Mod_{ME}(\mathcal{K})$. By Lemma 1 we know $\exists \mathcal{W} \in Mod_{ME}(\mathcal{K})$ such that $\mathcal{W} \prec_= \mathcal{J}$. Since $\mathcal{W} \in Mod_{ME}(\mathcal{K})$, we can say $a^{\mathcal{W}} \in C^{\mathcal{W}}$ which means $\mathcal{W} \models C(a)$. This is a contradiction because, by the definition of $\prec_=$, $\mathcal{W} \prec_= \mathcal{J}$ implies that $\mathcal{W} \models C(a)$ if and only if $\mathcal{J} \models C(a)$ which is not true.
$\Leftarrow$ This is trivially true since $Mod_{ME}(\mathcal{K}) \subseteq Mod(\mathcal{K})$. □

Lemma 3. *Let $\mathcal{K}$ be a $\mathcal{SRI}$ KB, $\forall a, b \in N_I$. Then.*
(1) either $\forall \mathcal{I} \in Mod_{ME}(\mathcal{K})$, $a^{\mathcal{I}} = b^{\mathcal{I}}$.
(2) or $\forall \mathcal{I} \in Mod_{ME}(\mathcal{K})$, $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.

Proof: Assume to the contrary of the lemma that $\exists \mathcal{J} \in Mod_{ME}(\mathcal{K})$ for which $a^{\mathcal{J}} = b^{\mathcal{J}}$ and $\exists \mathcal{W} \in Mod_{ME}(\mathcal{K})$ for which $a^{\mathcal{W}} \neq b^{\mathcal{W}}$. It is easy to see that $\mathcal{K} \not\models a = b$ and $\mathcal{K} \not\models a \neq b$. We construct a new interpretation $\mathcal{J}$' from $\mathcal{J}$ as follows:

- $c^{\mathcal{J}'} = c^{\mathcal{J}}$ for all $c \in N_I$ where $c \neq a$.
- $a^{\mathcal{J}'} = s$ where s is a fresh new value that is different from $c^{\mathcal{J}'}$ where $c \in N_I \setminus \{a\}$.
- For every concept C, if $a^{\mathcal{J}} \in C^{\mathcal{J}}$ then $C^{\mathcal{J}'} = C^{\mathcal{J}} \cup a^{\mathcal{J}'}$, otherwise $C^{\mathcal{J}'} = C^{\mathcal{J}}$.
- For every role R, if $\langle a^{\mathcal{J}}, m^{\mathcal{J}} \rangle \in R^{\mathcal{J}}$ then $R^{\mathcal{J}'} = R^{\mathcal{J}} \cup \langle a^{\mathcal{J}'}, m^{\mathcal{J}'} \rangle$; if $\langle m^{\mathcal{J}}, a^{\mathcal{J}} \rangle \in R^{\mathcal{J}}$ then $R^{\mathcal{J}'} = R^{\mathcal{J}} \cup \langle m^{\mathcal{J}'}, a^{\mathcal{J}'} \rangle$; otherwise $R^{\mathcal{J}'} = R^{\mathcal{J}}$.

We can see $\mathcal{J}' \in Mod(\mathcal{K})$. This is because changing the interpretation of individual a to s in $\mathcal{J}$' will only affect interpretation of nominals, cardinality restrictions ($\geq nR.C$ with $n > 1$), explicit equality (=) and inequality ($\neq$) axioms. First, we know the $\mathcal{SRI}$ KB is free of nominals and cardinality restrictions $\geq nR.C$ with $n > 1$(it still allows $\geq 1R.C$ which is equivalent to $\exists R.C$); Second, there are neither explicit equivalence axiom $a = b$ nor explicit inequivalence axiom $a \neq b$ because $\mathcal{K} \not\models a = b$ and $\mathcal{K} \not\models a \neq b$. So we conclude that $\mathcal{J}' \in Mod(\mathcal{K})$. According to the definition of $\prec_=$ we have $\mathcal{J}' \prec_= \mathcal{J}$, and by the definition of $Mod_{ME}(\mathcal{K})$ we know $\mathcal{J} \notin Mod_{ME}(\mathcal{K})$ which is a contradiction of our assumption. Therefore the correctness of Lemma 3 is proved. □

Lemma 4. *Given an extended KB $\langle \mathcal{K}, \mathcal{C} \rangle$ with expressivity $\langle \mathcal{SRI}, \mathcal{SROIQ} \rangle$ ($\langle \mathcal{SROIQ}, \mathcal{SROI} \rangle$ rep.), $\mathcal{K} \models_{IC} C(a)$ if and only if $\mathcal{K} \models \mathcal{T}_c(C, a)$, where C is a concept in constraint axioms $\mathcal{C}$, and $a \in N_I$.*

Proof: For the $\langle \mathcal{SRI}, \mathcal{SROIQ} \rangle$ KB, the proof is constructed inductively on the structure of concept C as follows.

Base Case: $\mathcal{K} \models_{IC} C_a(a)$ iff $\mathcal{K} \models \mathcal{T}_c(C_a, a)$, $C_a \in N_C$.
$\mathcal{K} \models_{IC} C_a(a)$
$\iff \forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I},\mathcal{U}} \in (C_a)^{\mathcal{I},\mathcal{U}}, \mathcal{U} = Mod_{ME}(\mathcal{K})$
$\iff \forall \mathcal{I} \in Mod_{ME}(\mathcal{K}), a^{\mathcal{I}} \in (C_a)^{\mathcal{I}}$
$[a^{\mathcal{I},\mathcal{U}} = a^{\mathcal{I}}, (C_a)^{\mathcal{I},\mathcal{U}} \subseteq (C_a)^{\mathcal{I}}]$
$\iff \forall \mathcal{I} \in Mod(\mathcal{K}), a^{\mathcal{I}} \in (C_a)^{\mathcal{I}}$
[by Lemma 2]
$\iff \mathcal{K} \models C_a(a)$
$\iff \mathcal{K} \models \mathcal{T}_c(C_a, a)$
[by def. of $\mathcal{T}_c(C_a, x)$]

Induction Steps: We now prove by induction that for any complex SROIQ concept C, $\mathcal{K} \models_{IC} C(a)$ iff $\mathcal{K} \models \mathcal{T}_c(C, a)$. As the basis of the proof, assume that $\mathcal{K} \models_{IC} C_{(i)}(a)$ iff $K \models \mathcal{T}_c(C_{(i)}, a)$, where subscript i=1,2 is optional. We enumerate all possible cases of SROIQ concepts as follows.

Case 1: $\mathcal{K} \models_{IC} (\neg C)(a)$ iff $\mathcal{K} \models \mathcal{T}_c(\neg C, a)$
$\mathcal{K} \models_{IC} (\neg C)(a)$
$\iff \forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I},\mathcal{U}} \in (\neg C)^{\mathcal{I},\mathcal{U}} = (N_I)^{\mathcal{I}} \setminus C^{\mathcal{I},\mathcal{U}}$
$\iff \exists \mathcal{J} \in \mathcal{U}, a^{\mathcal{J}} \notin C^{\mathcal{J}}$
$\iff$ It is false that
$\forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I},\mathcal{U}} \in (C)^{\mathcal{I},\mathcal{U}}$
$\iff \mathcal{K} \not\models_{IC} C(a)$
$\iff \mathcal{K} \not\models \mathcal{T}_c(C, a)$
[by assumption]
$\iff \mathcal{K} \models$ **not** $\mathcal{T}_c(C, a)$
$\iff \mathcal{K} \models \mathcal{T}_c(\neg C, a)$
[by def. of $\mathcal{T}_c(\neg C, x)$]

Case 2: $\mathcal{K} \models_{IC} (C_1 \sqcap C_2)(a)$ iff $\mathcal{K} \models \mathcal{T}_c(C_1 \sqcap C_2, a)$
$\mathcal{K} \models_{IC} (C_1 \sqcap C_2)(a)$
$\iff \forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I},\mathcal{U}} \in (C_1 \sqcap C_2)^{\mathcal{I},\mathcal{U}} = {C_1}^{\mathcal{I},\mathcal{U}} \cap {C_2}^{\mathcal{I},\mathcal{U}}$
$\iff \mathcal{K} \models_{IC} C_1(a), \mathcal{K} \models_{IC} C_2(a)$
$\iff \mathcal{K} \models \mathcal{T}_c(C_1, a), \mathcal{K} \models \mathcal{T}_c(C_2, a)$
[by assumption]
$\iff \mathcal{K} \models \mathcal{T}_c(C_1 \sqcap C_2, a)$
[by def. of $\mathcal{T}_c(C_1 \sqcap C_2, x)$]

Case 3: $\mathcal{K} \models_{IC} (\geq nR.C)(a)$ iff $\mathcal{K} \models \mathcal{T}_c(\geq nR.C, a)$

$\mathcal{K} \models_{IC} (\geq nR.C)(a)$

$\Longleftrightarrow \forall \mathcal{I} \in \mathcal{U}, \#\{y^{\mathcal{I}} \mid \langle a^{\mathcal{I},\mathcal{U}}, y^{\mathcal{I},\mathcal{U}}\rangle \in R^{\mathcal{I},\mathcal{U}}, y^{\mathcal{I},\mathcal{U}} \in C^{\mathcal{I},\mathcal{U}}\} \geq n$

$\Longleftrightarrow \exists y_i \in N_I (1 \leq i \leq n)$ s.t.

$\forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I}}, (y_i)^{\mathcal{I}}\rangle \in R^{\mathcal{I}},$

$[a^{\mathcal{I},\mathcal{U}} = a^{\mathcal{I}}, (y_i)^{\mathcal{I},\mathcal{U}} = (y_i)^{\mathcal{I}}, (R)^{\mathcal{I},\mathcal{U}} \subseteq (R)^{\mathcal{I}}]$

$\forall \mathcal{I} \in \mathcal{U}, (y_i)^{\mathcal{I},\mathcal{U}} \in C^{\mathcal{I},\mathcal{U}},$

$\forall \mathcal{I} \in \mathcal{U}, (y_i)^{\mathcal{I}} \neq (y_j)^{\mathcal{I}} (1 \leq i < j \leq n)$

[by Lemma 3]

$\Longleftrightarrow \exists y_i \in N_I (1 \leq i \leq n)$ s.t.

$\forall \mathcal{I} \in Mod(\mathcal{K}), \langle a^{\mathcal{I}}, (y_i)^{\mathcal{I}}\rangle \in R^{\mathcal{I}},$

[by Lemma 2]

$\mathcal{K} \models_{IC} C(y_i),$

$\mathcal{K} \models$ **not** $(y_i = y_j)(1 \leq i < j \leq n)$

$\Longleftrightarrow \exists y_i \in N_I (1 \leq i \leq n)$ s.t.

$\mathcal{K} \models R(a, y_i),$

$\mathcal{K} \models \mathcal{T}_c(C, y_i),$

[by assumption]

$\mathcal{K} \models$ **not** $(y_i = y_j)(1 \leq i < j \leq n)$

$$\Longleftrightarrow \mathcal{K} \models \bigwedge_{1 \leq i \leq n} (R(a, y_i) \land \mathcal{T}_c(C, y_i)) \bigwedge_{1 \leq i < j \leq n} \textbf{not}\,(y_i = y_j)$$

$\Longleftrightarrow \mathcal{K} \models \mathcal{T}_c(\geq nR.C, a)$

Case 4: $\mathcal{K} \models_{IC} (\exists R.\text{Self})(a)$ iff $\mathcal{K} \models \mathcal{T}_c(\exists R.\text{Self}, a)$

$\mathcal{K} \models_{IC} (\exists R.\text{Self})(a)$

$\Longleftrightarrow \forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I},\mathcal{U}} \in (\exists R.\text{Self})^{\mathcal{I},\mathcal{U}}$

$\Longleftrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I},\mathcal{U}}, a^{\mathcal{I},\mathcal{U}}\rangle \in R^{\mathcal{I},\mathcal{U}}$

$\Longleftrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I}}, a^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$

$[a^{\mathcal{I},\mathcal{U}} = a^{\mathcal{I}}, (R)^{\mathcal{I},\mathcal{U}} \subseteq (R)^{\mathcal{I}}]$

$\Longleftrightarrow \forall \mathcal{I} \in Mod(\mathcal{K}), \langle a^{\mathcal{I}}, a^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$

[by Lemma 2]

$\Longleftrightarrow \mathcal{K} \models R(a, a)$

$\Longleftrightarrow \mathcal{K} \models \mathcal{T}_c(\exists R.\text{Self}, a)$

[by def. of $\mathcal{T}_c(\exists R.\text{Self}, x)$]

Note that, the proof for the $\langle \mathcal{SROIQ}, \mathcal{SROI}\rangle$ KB is same as above except that case 3 will not happen. $\square$

We now obtain the main theorem of this section:

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.), we say that $\mathcal{K} \models_{IC} \alpha$ iff $\mathcal{K} \not\models \mathcal{T}(\alpha)$ where $\alpha \in \mathcal{C}$.*

Proof: We enumerate all possible cases of α as follows.

Case 1: $\mathcal{K} \models_{IC} C_1 \sqsubseteq C_2$ iff $\mathcal{K} \not\models \mathcal{T}(C_1 \sqsubseteq C_2)$

($\Longrightarrow$): if $\mathcal{K} \models_{IC} C_1 \sqsubseteq C_2$ then $\mathcal{K} \not\models \mathcal{T}(C_1 \sqsubseteq C_2)$.

Assume to the contrary

$\mathcal{K} \models \mathcal{T}(C_1 \sqsubseteq C_2)$

$\Longrightarrow \mathcal{K} \models \mathcal{T}_c(C_1, x) \land$ **not** $\mathcal{T}_c(C_2, x)$

[by def. of $\mathcal{T}(C_1 \sqsubseteq C_2)$]

$\Longrightarrow \exists \sigma : x \to a, \mathcal{K} \models \mathcal{T}_c(C_1, a), \mathcal{K} \not\models \mathcal{T}_c(C_2, a)$

$\Longrightarrow \mathcal{K} \models_{IC} C_1(a), \mathcal{K} \not\models_{IC} C_2(a)$

[by Lemma 4]

$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I},\mathcal{U}} \in (C_1)^{\mathcal{I},\mathcal{U}}, \exists \mathcal{J} \in \mathcal{U}, a^{\mathcal{J},\mathcal{U}} \notin (C_2)^{\mathcal{J},\mathcal{U}}$

$\Longrightarrow$ It is false that

$\forall \mathcal{I} \in \mathcal{U}, (C_1)^{\mathcal{I},\mathcal{U}} \subseteq (C_2)^{\mathcal{I},\mathcal{U}}$

$\Longrightarrow \mathcal{K} \not\models_{IC} C_1 \sqsubseteq C_2$

Which is a contradiction.

($\Longleftarrow$): if $\mathcal{K} \not\models \mathcal{T}(C_1 \sqsubseteq C_2)$ then $\mathcal{K} \models_{IC} C_1 \sqsubseteq C_2$.

$\mathcal{K} \not\models \mathcal{T}(C_1 \sqsubseteq C_2)$

$\Longrightarrow \mathcal{K} \not\models \mathcal{T}_c(C_1, x) \land$ **not** $\mathcal{T}_c(C_2, x)$

[by def. of $\mathcal{T}(C_1 \sqsubseteq C_2)$]

$\Longrightarrow \nexists \sigma : x \to a, K \models \mathcal{T}_c(C_1, a), \mathcal{K} \not\models \mathcal{T}_c(C_2, a)$

$\Longrightarrow \forall \sigma : x \to a,$ if $\mathcal{K} \models \mathcal{T}_c(C_1, a),$ then $\mathcal{K} \models \mathcal{T}_c(C_2, a)$

$\Longrightarrow \forall x \in N_I,$ if $\mathcal{K} \models_{IC} C_1(x),$ then $\mathcal{K} \models_{IC} C_2(x)$

[by Lemma 4]

$\Longrightarrow \forall x \in N_I,$ if $\forall \mathcal{I} \in \mathcal{U}, x^{\mathcal{I},\mathcal{U}} \in (C_1)^{\mathcal{I},\mathcal{U}},$

then $\forall \mathcal{I} \in \mathcal{U}, x^{\mathcal{I},\mathcal{U}} \in (C_2)^{\mathcal{I},\mathcal{U}}$

$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, (C_1)^{\mathcal{I},\mathcal{U}} \subseteq (C_2)^{\mathcal{I},\mathcal{U}}$

$\Longrightarrow \mathcal{K} \models_{IC} C_1 \sqsubseteq C_2$

Case 2: $\mathcal{K} \models_{IC} R_1 \sqsubseteq R_2$ iff $\mathcal{K} \not\models \mathcal{T}(R_1 \sqsubseteq R_2)$

($\Longrightarrow$): if $\mathcal{K} \models_{IC} R_1 \sqsubseteq R_2$ then $\mathcal{K} \not\models \mathcal{T}(R_1 \sqsubseteq R_2)$.

Assume to the contrary

$\mathcal{K} \models \mathcal{T}(R_1 \sqsubseteq R_2)$

$\Longrightarrow \mathcal{K} \models R_1(x, y) \land$ **not** $R_2(x, y)$

[by def. of $\mathcal{T}(R_1 \sqsubseteq R_2)$]

$\Longrightarrow \exists \sigma : x \to a, y \to b, \mathcal{K} \models R_1(a, b), \mathcal{K} \not\models R_2(a, b)$

$\Longrightarrow \forall \mathcal{I} \in Mod(\mathcal{K}), \langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R_1^{\mathcal{I}},$

$\exists \mathcal{J} \in Mod(\mathcal{K}), \langle a^{\mathcal{J}}, b^{\mathcal{J}}\rangle \notin R_2^{\mathcal{I}}$

$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R_1^{\mathcal{I}}, \exists \mathcal{J}' \in \mathcal{U}, \langle a^{\mathcal{J}'}, b^{\mathcal{J}'}\rangle \notin R_2^{\mathcal{J}'}$

[by Lemma 2]

$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I},\mathcal{U}}, b^{\mathcal{I},\mathcal{U}}\rangle \in R_1^{\mathcal{I},\mathcal{U}}, \exists \mathcal{J}' \in \mathcal{U}, \langle a^{\mathcal{J}',\mathcal{U}}, b^{\mathcal{J}',\mathcal{U}}\rangle \notin R_2^{\mathcal{J}',\mathcal{U}}$

$\Longrightarrow$ It is false that

$\forall \mathcal{I} \in \mathcal{U}, (R_1)^{\mathcal{I},\mathcal{U}} \subseteq (R_2)^{\mathcal{I},\mathcal{U}}$

$\Longrightarrow \mathcal{K} \not\models_{IC} R_1 \sqsubseteq R_2$

Which is a contradiction.

($\Longleftarrow$): if $\mathcal{K} \not\models \mathcal{T}(R_1 \sqsubseteq R_2)$ then $\mathcal{K} \models_{IC} R_1 \sqsubseteq R_2$.

$\mathcal{K} \not\models \mathcal{T}(R_1 \sqsubseteq R_2)$
$\Longrightarrow \mathcal{K} \not\models R_1(x,y) \wedge \textbf{not}\, R_2(x,y)$
$[\texttt{by def. of } \mathcal{T}(R_1 \sqsubseteq R_2)]$
$\Longrightarrow \nexists \sigma : x \rightarrow a, y \rightarrow b, K \models R_1(a,b), \mathcal{K} \not\models R_2(a,b)$
$\Longrightarrow \forall \sigma : x \rightarrow a, y \rightarrow b, \texttt{if } \mathcal{K} \models R_1(a,b), \texttt{then } \mathcal{K} \models R_2(a,b)$
$\Longrightarrow \forall x,y \in N_I, \texttt{if } \forall \mathcal{I} \in Mod(\mathcal{K}), \langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in R_1^{\mathcal{I}}$
$\texttt{then } \forall \mathcal{I} \in Mod(\mathcal{K}), \langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in R_2^{\mathcal{I}}$
$\Longrightarrow \forall x,y \in N_I, \texttt{if } \forall \mathcal{I} \in \mathcal{U}, \langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in R_1^{\mathcal{I}}$
$\texttt{then } \forall \mathcal{I} \in \mathcal{U}, \langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in R_2^{\mathcal{I}}$
$[\texttt{by Lemma 2}]$
$\Longrightarrow \forall x,y \in N_I, \texttt{if } \forall \mathcal{I} \in \mathcal{U}, \langle x^{\mathcal{I},\mathcal{U}}, y^{\mathcal{I},\mathcal{U}} \rangle \in (R_1)^{\mathcal{I},\mathcal{U}},$
$\texttt{then } \forall \mathcal{I} \in \mathcal{U}, \langle x^{\mathcal{I},\mathcal{U}}, y^{\mathcal{I},\mathcal{U}} \rangle \in (R_2)^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, (R_1)^{\mathcal{I},\mathcal{U}} \subseteq (R_2)^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \models_{IC} R_1 \sqsubseteq R_2$

Case 3: $\mathcal{K} \models_{IC} R_1 \ldots R_n \sqsubseteq R$ iff $\mathcal{K} \not\models \mathcal{T}(R_1 \ldots R_n \sqsubseteq R)$

($\Longrightarrow$): if $\mathcal{K} \models_{IC} R_1 \ldots R_n \sqsubseteq R$ then $\mathcal{K} \not\models \mathcal{T}(R_1 \ldots R_n \sqsubseteq R)$.
Assume to the contrary
$\mathcal{K} \models \mathcal{T}(R_1 \ldots R_n \sqsubseteq R)$
$\Longrightarrow \mathcal{K} \models R_1(x,y_1) \wedge \ldots R_n(y_{n-1},y_n) \wedge \textbf{not}\, R(x,y_n)$
$[\texttt{by def. of } \mathcal{T}(R_1 \ldots R_n \sqsubseteq R)]$
$\Longrightarrow \exists \sigma : x \rightarrow a, y_1 \rightarrow b_1, \ldots, y_n \rightarrow b_n, s.t.$
$\mathcal{K} \models R_1(a,b_1), \ldots, \mathcal{K} \models R_n(b_{n-1},b_n), \mathcal{K} \not\models R(a,b_n)$
$\Longrightarrow \forall \mathcal{I} \in Mod(\mathcal{K}), \langle a^{\mathcal{I}}, {b_1}^{\mathcal{I}} \rangle \in R_1^{\mathcal{I}},$
$\ldots,$
$\forall \mathcal{I} \in Mod(\mathcal{K}), \langle {b_{n-1}}^{\mathcal{I}}, {b_n}^{\mathcal{I}} \rangle \in R_n^{\mathcal{I}},$
$\exists \mathcal{J} \in Mod(\mathcal{K}), \langle a^{\mathcal{J}}, {b_n}^{\mathcal{J}} \rangle \notin R^{\mathcal{I}}$
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I}}, {b_1}^{\mathcal{I}} \rangle \in R_1^{\mathcal{I}},$
$\ldots,$
$\forall \mathcal{I} \in \mathcal{U}, \langle {b_{n-1}}^{\mathcal{I}}, {b_n}^{\mathcal{I}} \rangle \in R_n^{\mathcal{I}},$
$\exists \mathcal{J}' \in \mathcal{U}, \langle a^{\mathcal{J}'}, {b_n}^{\mathcal{J}'} \rangle \notin R^{\mathcal{J}'}$
$[\texttt{by Lemma 2}]$
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I},\mathcal{U}}, {b_1}^{\mathcal{I},\mathcal{U}} \rangle \in R_1^{\mathcal{I},\mathcal{U}},$
$\ldots,$
$\forall \mathcal{I} \in \mathcal{U}, \langle {b_{n-1}}^{\mathcal{I},\mathcal{U}}, {b_n}^{\mathcal{I},\mathcal{U}} \rangle \in R_n^{\mathcal{I},\mathcal{U}},$
$\exists \mathcal{J}' \in \mathcal{U}, \langle a^{\mathcal{J}',\mathcal{U}}, {b_n}^{\mathcal{J}',\mathcal{U}} \rangle \notin R^{\mathcal{J}',\mathcal{U}}$
$\Longrightarrow \texttt{It is false that}$
$\forall \mathcal{I} \in \mathcal{U}, R_1^{\mathcal{I},\mathcal{U}} \circ \ldots \circ R_n^{\mathcal{I},\mathcal{U}} \subseteq R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \not\models_{IC} R_1 \ldots R_n \sqsubseteq R$
Which is a contradiction.

($\Longleftarrow$): if $\mathcal{K} \not\models \mathcal{T}(R_1 \ldots R_n \sqsubseteq R)$ then $\mathcal{K} \models_{IC} R_1 \ldots R_n \sqsubseteq R$.

$\mathcal{K} \not\models \mathcal{T}(R_1 \ldots R_n \sqsubseteq R)$
$\Longrightarrow \mathcal{K} \not\models R_1(x,y_1) \wedge \ldots R_n(y_{n-1},y_n) \wedge \textbf{not}\, R(x,y_n)$
$[\texttt{by def. of } \mathcal{T}(R_1 \ldots R_n \sqsubseteq R)]$
$\Longrightarrow \nexists \sigma : x \rightarrow a, y_1 \rightarrow b_1, \ldots, y_n \rightarrow b_n, s.t.$
$K \models R_1(a,b), \ldots, K \models R_n(b_{n-1},b_n), \mathcal{K} \not\models R(a,b_n)$
$\Longrightarrow \forall \sigma : x \rightarrow a, y_1 \rightarrow b_1, \ldots, y_n \rightarrow b_n,$
$\texttt{if } \mathcal{K} \models R_1(a,b), \ldots, K \models R_n(b_{n-1},b_n),$
$\texttt{then } \mathcal{K} \models R(a,b_n)$
$\Longrightarrow \forall x, y_1, \ldots, y_n \in N_I,$
$\texttt{if } \forall \mathcal{I} \in Mod(\mathcal{K}), \langle x^{\mathcal{I}}, {y_1}^{\mathcal{I}} \rangle \in R_1^{\mathcal{I}}$
$\ldots,$
$\forall \mathcal{I} \in Mod(\mathcal{K}), \langle {y_{n-1}}^{\mathcal{I}}, {y_n}^{\mathcal{I}} \rangle \in R_n^{\mathcal{I}}$
$\texttt{then } \forall \mathcal{I} \in Mod(\mathcal{K}), \langle x^{\mathcal{I}}, {y_n}^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
$\Longrightarrow \forall x, y_1, \ldots, y_n \in N_I,$
$\texttt{if } \forall \mathcal{I} \in \mathcal{U}, \langle x^{\mathcal{I}}, {y_1}^{\mathcal{I}} \rangle \in R_1^{\mathcal{I}}$
$\ldots,$
$\forall \mathcal{I} \in \mathcal{U}, \langle {y_{n-1}}^{\mathcal{I}}, {y_n}^{\mathcal{I}} \rangle \in R_n^{\mathcal{I}}$
$\texttt{then } \forall \mathcal{I} \in \mathcal{U}, \langle x^{\mathcal{I}}, {y_n}^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
$[\texttt{by Lemma 2}]$
$\Longrightarrow \forall x, y_1, \ldots, y_n \in N_I,$
$\texttt{if } \forall \mathcal{I} \in \mathcal{U}, \langle x^{\mathcal{I},\mathcal{U}}, {y_1}^{\mathcal{I},\mathcal{U}} \rangle \in R_1^{\mathcal{I},\mathcal{U}}$
$\ldots,$
$\forall \mathcal{I} \in \mathcal{U}, \langle {y_{n-1}}^{\mathcal{I},\mathcal{U}}, {y_n}^{\mathcal{I},\mathcal{U}} \rangle \in R_n^{\mathcal{I},\mathcal{U}}$
$\texttt{then } \forall \mathcal{I} \in \mathcal{U}, \langle x^{\mathcal{I},\mathcal{U}}, {y_n}^{\mathcal{I},\mathcal{U}} \rangle \in R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, R_1^{\mathcal{I},\mathcal{U}} \circ \ldots \circ R_n^{\mathcal{I},\mathcal{U}} \subseteq R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \models_{IC} R_1 \ldots R_n \sqsubseteq R$

Case 4: $\mathcal{K} \models_{IC} \texttt{Ref}(\mathrm{R})$ iff $\mathcal{K} \not\models \mathcal{T}(\texttt{Ref}(\mathrm{R}))$

($\Longrightarrow$): if $\mathcal{K} \models_{IC} \texttt{Ref}(\mathrm{R})$ then $\mathcal{K} \not\models \mathcal{T}(\texttt{Ref}(\mathrm{R}))$.
Assume to the contrary
$\mathcal{K} \models \mathcal{T}(\texttt{Ref}(R))$
$\Longrightarrow \mathcal{K} \models \textbf{not}\, R(x,x)$
$[\texttt{by def. of } \mathcal{T}(\texttt{Ref}(R))]$
$\Longrightarrow \forall x \in N_I, \exists \mathcal{I} \in Mod(\mathcal{K}), s.t. \langle x^{\mathcal{I}}, x^{\mathcal{I}} \rangle \notin R^{\mathcal{I}}.$
$\Longrightarrow \forall x \in N_I, \exists \mathcal{I}' \in \mathcal{U}, s.t. \langle x^{\mathcal{I}'}, x^{\mathcal{I}'} \rangle \notin R^{\mathcal{I}'}.$
$[\texttt{by Lemma 2}]$
$\Longrightarrow \forall x \in N_I, \exists \mathcal{I}' \in \mathcal{U}, s.t. \langle x^{\mathcal{I}',\mathcal{U}}, x^{\mathcal{I}',\mathcal{U}} \rangle \notin R^{\mathcal{I}',\mathcal{U}}.$
$\Longrightarrow \texttt{It is false that}$
$\forall \mathcal{I} \in \mathcal{U}, \forall x \in N_I, \langle x^{\mathcal{I},\mathcal{U}}, x^{\mathcal{I},\mathcal{U}} \rangle \in R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \not\models_{IC} \texttt{Ref}(R)$
Which is a contradiction.

($\Longleftarrow$): if $\mathcal{K} \not\models \mathcal{T}(\texttt{Ref}(\mathrm{R}))$ then $\mathcal{K} \models_{IC} \texttt{Ref}(\mathrm{R})$.

$\mathcal{K} \not\models \mathcal{T}(\texttt{Ref}(R))$
$\Longrightarrow \mathcal{K} \not\models \textbf{not}\, R(x,x)$
`[by def. of` $\mathcal{T}(\texttt{Ref}(R))$`]`
$\Longrightarrow \nexists \sigma : x \to a, s.t. \mathcal{K} \models \textbf{not}\, R(a,a)$
$\Longrightarrow \forall x \in N_I, \mathcal{K} \models R(x,x)$
$\Longrightarrow \forall x \in N_I, \forall \mathcal{I} \in Mod(\mathcal{K}), \langle x^{\mathcal{I}}, x^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
$\Longrightarrow \forall x \in N_I, \forall \mathcal{I} \in \mathcal{U}, \langle x^{\mathcal{I}}, x^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
`[by Lemma 2]`
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \forall x \in N_I, \langle x^{\mathcal{I},\mathcal{U}}, x^{\mathcal{I},\mathcal{U}} \rangle \in R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \models_{IC} \texttt{Ref}(R)$

Case 5: $\mathcal{K} \models_{IC} \texttt{Irr}(\text{R})$ iff $\mathcal{K} \not\models \mathcal{T}(\texttt{Irr}(\text{R}))$

($\Longrightarrow$): if $\mathcal{K} \models_{IC} \texttt{Irr}(\text{R})$ then $\mathcal{K} \not\models \mathcal{T}(\texttt{Irr}(\text{R}))$.
Assume to the contrary
$\mathcal{K} \models \mathcal{T}(\texttt{Irr}(R))$
$\Longrightarrow \mathcal{K} \models R(x,x)$
`[by def. of` $\mathcal{T}(\texttt{Irr}(R))$`]`
$\Longrightarrow \exists \sigma : x \to a, s.t. \mathcal{K} \models R(a,a)$
$\Longrightarrow \exists a \in N_I, s.t. \forall \mathcal{I} \in Mod(\mathcal{K}), \langle a^{\mathcal{I}}, a^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
$\Longrightarrow \exists a \in N_I, s.t. \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I}}, a^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
`[by Lemma 2]`
$\Longrightarrow \exists a \in N_I, s.t. \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I},\mathcal{U}}, a^{\mathcal{I},\mathcal{U}} \rangle \in R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow$ `It is false that`
$\forall \mathcal{I} \in \mathcal{U}, \forall x \in N_I, \langle x^{\mathcal{I},\mathcal{U}}, x^{\mathcal{I},\mathcal{U}} \rangle \notin R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \not\models_{IC} \texttt{Irr}(R)$
Which is a contradiction.

($\Longleftarrow$): if $\mathcal{K} \not\models \mathcal{T}(\texttt{Irr}(\text{R}))$ then $\mathcal{K} \models_{IC} \texttt{Irr}(\text{R})$.

$\mathcal{K} \not\models \mathcal{T}(\texttt{Irr}(R))$
$\Longrightarrow \mathcal{K} \not\models R(x,x)$
`[by def. of` $\mathcal{T}(\texttt{Irr}(R))$`]`
$\Longrightarrow \forall x \in N_I, \exists \mathcal{I} \in Mod(\mathcal{K}), s.t. \langle x^{\mathcal{I}}, x^{\mathcal{I}} \rangle \notin R^{\mathcal{I}}$
$\Longrightarrow \forall x \in N_I, \exists \mathcal{I}' \in \mathcal{U}, s.t. \langle x^{\mathcal{I}'}, x^{\mathcal{I}'} \rangle \notin R^{\mathcal{I}'}$
`[by Lemma 2]`
$\Longrightarrow \forall x \in N_I, \forall \mathcal{I} \in \mathcal{U}, \langle x^{\mathcal{I},\mathcal{U}}, x^{\mathcal{I},\mathcal{U}} \rangle \notin R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \models_{IC} \texttt{Irr}(R)$

Case 6: $\mathcal{K} \models_{IC} \texttt{Dis}(R_1, R_2)$ iff $\mathcal{K} \not\models \mathcal{T}(\texttt{Dis}(R_1, R_2))$

($\Longrightarrow$): if $\mathcal{K} \models_{IC} \texttt{Dis}(R_1, R_2)$ then $\mathcal{K} \not\models \mathcal{T}(\texttt{Dis}(R_1, R_2))$.
Assume to the contrary
$\mathcal{K} \models \mathcal{T}(\texttt{Dis}(R_1, R_2))$
$\Longrightarrow \mathcal{K} \models R_1(x,y) \wedge R_2(x,y)$
`[by def. of` $\mathcal{T}(\texttt{Dis}(R_1, R_2))$`]`
$\Longrightarrow \exists \sigma : x \to a, y \to b, s.t. \mathcal{K} \models R_1(a,b), \mathcal{K} \models R_2(a,b)$
$\Longrightarrow \forall \mathcal{I} \in Mod(\mathcal{K}), \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in {R_1}^{\mathcal{I}}, \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in {R_2}^{\mathcal{I}}$
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in {R_1}^{\mathcal{I}}, \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in {R_2}^{\mathcal{I}}$
`[by Lemma 2]`
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I},\mathcal{U}}, b^{\mathcal{I},\mathcal{U}} \rangle \in {R_1}^{\mathcal{I},\mathcal{U}}, \langle a^{\mathcal{I},\mathcal{U}}, b^{\mathcal{I},\mathcal{U}} \rangle \in {R_2}^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I},\mathcal{U}}, b^{\mathcal{I},\mathcal{U}} \rangle \in {R_1}^{\mathcal{I},\mathcal{U}} \cap {R_2}^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow$ `It is false that`
$\forall \mathcal{I} \in \mathcal{U}, R_1^{\mathcal{I},\mathcal{U}} \cap R_2^{\mathcal{I},\mathcal{U}} = \emptyset$
$\Longrightarrow \mathcal{K} \not\models_{IC} \texttt{Dis}(R_1, R_2)$
Which is a contradiction.
($\Longleftarrow$): if $\mathcal{K} \not\models \mathcal{T}(\texttt{Dis}(R_1, R_2))$ then $\mathcal{K} \models_{IC} \texttt{Dis}(R_1, R_2)$.

$\mathcal{K} \not\models \mathcal{T}(\texttt{Dis}(R_1, R_2))$
$\Longrightarrow \mathcal{K} \not\models R_1(x,y) \wedge R_2(x,y)$
`[by def. of` $\mathcal{T}(\texttt{Dis}(R_1, R_2))$`]`
$\Longrightarrow \nexists a, b \in N_I, s.t. \mathcal{K} \models R_1(a,b), \mathcal{K} \models R_2(a,b)$
$\Longrightarrow \nexists a, b \in N_I, s.t.$
$\forall \mathcal{I} \in Mod(\mathcal{K}), \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in {R_1}^{\mathcal{I}}, \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in {R_2}^{\mathcal{I}}$
$\Longrightarrow \nexists a, b \in N_I, s.t. \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in {R_1}^{\mathcal{I}}, \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in {R_2}^{\mathcal{I}}$
`[by Lemma 2]`
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \nexists a, b \in N_I, s.t. \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R_1^{\mathcal{I},\mathcal{U}} \cap R_2^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, R_1^{\mathcal{I},\mathcal{U}} \cap R_2^{\mathcal{I},\mathcal{U}} = \emptyset$
$\Longrightarrow \mathcal{K} \models_{IC} \texttt{Dis}(R_1, R_2)$
Case 7: $\mathcal{K} \models_{IC} C(a)$ iff $\mathcal{K} \not\models \mathcal{T}(C(a))$
($\Longrightarrow$): if $\mathcal{K} \models_{IC} C(a)$ then $\mathcal{K} \not\models \mathcal{T}(C(a))$.
Assume to the contrary
$\mathcal{K} \models \mathcal{T}(C(a))$
$\Longrightarrow \mathcal{K} \models \textbf{not}\, \mathcal{T}_c(C, a)$
`[by def. of` $\mathcal{T}(C(a))$`]`
$\Longrightarrow \mathcal{K} \not\models \mathcal{T}_c(C, a)$
$\Longrightarrow \mathcal{K} \not\models_{IC} C(a)$
`[by Lemma 4]`
Which is a contradiction.
($\Longleftarrow$): if $\mathcal{K} \not\models \mathcal{T}(C(a))$ then $\mathcal{K} \models_{IC} C(a)$.

$\mathcal{K} \not\models \mathcal{T}(C(a))$
$\Longrightarrow \mathcal{K} \not\models \textbf{not}\, \mathcal{T}_c(C, a)$
`[by def. of` $\mathcal{T}(C(a))$`]`
$\Longrightarrow \mathcal{K} \models \mathcal{T}_c(C, a)$
$\Longrightarrow \mathcal{K} \models_{IC} C(a)$
`[by Lemma 4]`
Case 8: $\mathcal{K} \models_{IC} R(a,b)$ iff $\mathcal{K} \not\models \mathcal{T}(R(a,b))$

($\Longrightarrow$): if $\mathcal{K} \models_{IC} R(a,b)$ then $\mathcal{K} \not\models \mathcal{T}(R(a,b))$.
Assume to the contrary
$\mathcal{K} \models \mathcal{T}(R(a,b))$
$\Longrightarrow \mathcal{K} \models$ **not** $R(a,b)$
[`by def. of` $\mathcal{T}(R(a,b))$]
$\Longrightarrow \exists \mathcal{I} \in Mod(\mathcal{K}), s.t. \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \notin R^{\mathcal{I}}$
$\Longrightarrow \exists \mathcal{I}' \in \mathcal{U}, s.t. \langle a^{\mathcal{I}'}, b^{\mathcal{I}'} \rangle \notin R^{\mathcal{I}'}$
[`by Lemma 2`]
$\Longrightarrow \exists \mathcal{I}' \in \mathcal{U}, s.t. \langle a^{\mathcal{I}',\mathcal{U}}, b^{\mathcal{I}',\mathcal{U}} \rangle \notin R^{\mathcal{I}',\mathcal{U}}$
$\Longrightarrow$ `It is false that`
$\forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I},\mathcal{U}}, b^{\mathcal{I},\mathcal{U}} \rangle \in R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \not\models_{IC} R(a,b)$
Which is a contradiction.

($\Longleftarrow$): if $\mathcal{K} \not\models \mathcal{T}(R(a,b))$ then $\mathcal{K} \models_{IC} R(a,b)$.

$\mathcal{K} \not\models \mathcal{T}(R(a,b))$
$\Longrightarrow \mathcal{K} \not\models$ **not** $R(a,b)$
[`by def. of` $\mathcal{T}(R(a,b))$]
$\Longrightarrow \mathcal{K} \models R(a,b)$
$\Longrightarrow \forall \mathcal{I} \in Mod(\mathcal{K}), \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
[`by Lemma 2`]
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I},\mathcal{U}}, b^{\mathcal{I},\mathcal{U}} \rangle \in R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \models_{IC} R(a,b)$

Case 9: $\mathcal{K} \models_{IC} \neg R(a,b)$ iff $\mathcal{K} \not\models \mathcal{T}(\neg R(a,b))$

($\Longrightarrow$): if $\mathcal{K} \models_{IC} \neg R(a,b)$ then $\mathcal{K} \not\models \mathcal{T}(\neg R(a,b))$.
Assume to the contrary
$\mathcal{K} \models \mathcal{T}(\neg R(a,b))$
$\Longrightarrow \mathcal{K} \models R(a,b)$
[`by def. of` $\mathcal{T}(\neg R(a,b))$]
$\Longrightarrow \forall \mathcal{I} \in Mod(\mathcal{K}), \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
[`by Lemma 2`]
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I},\mathcal{U}}, b^{\mathcal{I},\mathcal{U}} \rangle \in R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow$ `It is false that`
$\forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I},\mathcal{U}}, b^{\mathcal{I},\mathcal{U}} \rangle \notin R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \not\models_{IC} \neg R(a,b)$
Which is a contradiction.

($\Longleftarrow$): if $\mathcal{K} \not\models \mathcal{T}(\neg R(a,b))$ then $\mathcal{K} \models_{IC} \neg R(a,b)$.

$\mathcal{K} \not\models \mathcal{T}(\neg R(a,b))$
$\Longrightarrow \mathcal{K} \not\models R(a,b)$
[`by def. of` $\mathcal{T}(\neg R(a,b))$]
$\Longrightarrow \exists \mathcal{I} \in Mod(\mathcal{K}), \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \notin R^{\mathcal{I}}$
$\Longrightarrow \exists \mathcal{I}' \in \mathcal{U}, \langle a^{\mathcal{I}'}, b^{\mathcal{I}'} \rangle \notin R^{\mathcal{I}'}$
[`by Lemma 2`]
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, \langle a^{\mathcal{I},\mathcal{U}}, b^{\mathcal{I},\mathcal{U}} \rangle \notin R^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \models_{IC} \neg R(a,b)$

Case 10: $\mathcal{K} \models_{IC} a = b$ iff $\mathcal{K} \not\models \mathcal{T}(a = b)$

($\Longrightarrow$): if $\mathcal{K} \models_{IC} a = b$ then $\mathcal{K} \not\models \mathcal{T}(a = b)$.
Assume to the contrary
$\mathcal{K} \models \mathcal{T}(a = b)$
$\Longrightarrow \mathcal{K} \models$ **not** $(a = b)$
[`by def. of` $\mathcal{T}(a = b)$]
$\Longrightarrow \exists \mathcal{I} \in Mod(\mathcal{K}), a^{\mathcal{I}} \neq b^{\mathcal{I}}$
$\Longrightarrow \exists \mathcal{I}' \in \mathcal{U}, a^{\mathcal{I}'} \neq b^{\mathcal{I}'}$
[`by Lemma 2`]
$\Longrightarrow \exists \mathcal{I}' \in \mathcal{U}, a^{\mathcal{I}',\mathcal{U}} \neq b^{\mathcal{I}',\mathcal{U}}$
$\Longrightarrow$ `It is false that`
$\forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I},\mathcal{U}} = b^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \not\models_{IC} a = b$
Which is a contradiction.

($\Longleftarrow$): if $\mathcal{K} \not\models \mathcal{T}(a = b)$ then $\mathcal{K} \models_{IC} a = b$.

$\mathcal{K} \not\models \mathcal{T}(a = b)$
$\Longrightarrow \mathcal{K} \not\models$ **not** $(a = b)$
[`by def. of` $\mathcal{T}(a = b)$]
$\Longrightarrow \mathcal{K} \models (a = b)$
$\Longrightarrow \forall \mathcal{I} \in Mod(\mathcal{K}), a^{\mathcal{I}} = b^{\mathcal{I}}$
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I}} = b^{\mathcal{I}}$
[`by Lemma 2`]
$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I},\mathcal{U}} = b^{\mathcal{I},\mathcal{U}}$
$\Longrightarrow \mathcal{K} \models_{IC} a = b$

Case 11: $\mathcal{K} \models_{IC} a \neq b$ iff $\mathcal{K} \not\models \mathcal{T}(a \neq b)$

($\Longrightarrow$): if $\mathcal{K} \models_{IC} a \neq b$ then $\mathcal{K} \not\models \mathcal{T}(a \neq b)$.

Assume to the contrary

$\mathcal{K} \models \mathcal{T}(a \neq b)$

$\Longrightarrow \mathcal{K} \models (a = b)$

[`by def. of` $\mathcal{T}(a \neq b)$]

$\Longrightarrow \forall \mathcal{I} \in Mod(\mathcal{K}), a^{\mathcal{I}} = b^{\mathcal{I}}$

$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I}} = b^{\mathcal{I}}$

[`by Lemma 2`]

$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I},\mathcal{U}} = b^{\mathcal{I},\mathcal{U}}$

$\Longrightarrow$ `It is false that`

$\forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I},\mathcal{U}} \neq b^{\mathcal{I},\mathcal{U}}$

$\Longrightarrow \mathcal{K} \not\models_{IC} a \neq b$

Which is a contradiction.

($\Longleftarrow$): if $\mathcal{K} \not\models \mathcal{T}(a \neq b)$ then $\mathcal{K} \models_{IC} a \neq b$.

$\mathcal{K} \not\models \mathcal{T}(a \neq b)$

$\Longrightarrow \mathcal{K} \not\models (a = b)$

[`by def. of` $\mathcal{T}(a \neq b)$]

$\Longrightarrow \exists \mathcal{I} \in Mod(\mathcal{K}), a^{\mathcal{I}} \neq b^{\mathcal{I}}$

$\Longrightarrow \exists \mathcal{I}' \in \mathcal{U}, a^{\mathcal{I}'} \neq b^{\mathcal{I}'}$

[`by Lemma 2`]

$\Longrightarrow \forall \mathcal{I} \in \mathcal{U}, a^{\mathcal{I},\mathcal{U}} \neq b^{\mathcal{I},\mathcal{U}}$

$\Longrightarrow \mathcal{K} \models_{IC} a \neq b$

Note that, when the expressivity of the extended KB $\langle \mathcal{K}, \mathcal{C} \rangle$ is $\langle \mathcal{SROIQ}, \mathcal{SROI} \rangle$, the concepts used in α will not include cardinality restrictions. □

6. Implementation

The query language for OWL ontologies is SPARQL (Prud'hommeaux and Seaborne 2008) which is known to have the same expressive power as nonrecursive Datalog programs (Angles and Gutierrez 2008) and can express DCQ**not** queries. Therefore, based on the results from Section 5.3we can reduce IC validation to SPARQL query answering if the KB is $\mathcal{SRI}$ or the ICs do not contain cardinality restrictions.

In Table 3, we present a mapping $\mathcal{M} : Q \rightarrow P$ from a DCQ**not** query Q to a SPARQL graph pattern P, such that $\mathbf{A}(Q, \mathcal{K})$ will be equivalent to the result set obtained by evaluating $\mathcal{M}(Q)$ over the corresponding OWL ontology using OWL entailment regime. Note that, in our mapping we use NOT EXISTS pattern which is being added in SPARQL 1.1[4] and can also be encoded in SPARQL 1.0 using a combination of other operators (Angles and Gutierrez 2008).

Based on this mapping, the SPARQL representation of the query in Example 10 becomes:

```
ASK WHERE {
    ?x rdf:type Product.
    NOT EXISTS { ?x hasProducer ?y.
                 ?y rdf:type Producer.}
}
```

[4]http://www.w3.org/TR/sparql11-query/#negation

Q	**P** = $\mathcal{M}$(**Q**)
$C(x)$	`x rdf:type C`
$R(x,y)$	`x R y`
$x = y$	`x owl:sameAs y`
$x \neq y$	`x owl:differentFrom y`
$Q_1 \wedge Q_2$	JOIN($\mathcal{M}(Q_1), \mathcal{M}(Q_2)$)
not Q	NOT EXISTS($\mathcal{M}(Q)$)

Table 3: Mapping DCQ**not** to SPARQL queries

We have built a prototype IC validator[5] by extending the OWL 2 DL reasoner Pellet[6]. The prototype reads ICs expressed as OWL axioms and translates each IC first to a DCQ**not** query and then to a SPARQL query. The resulting query is executed by the SPARQL engine in Pellet where a non-empty result indicates a constraint violation. Since the translation algorithm is reasoner independent this prototype can be used in conjunction with any OWL reasoner that supports SPARQL query answering.

We have used this proof-of-concept prototype to validate ICs with several large ontologies such as the LUBM dataset.[7] For testing, we removed several axioms from the LUBM ontology and declared them as ICs instead. The dataset is logically consistent but turning axioms into ICs caused some violations to be detected. Since each constraint is turned into a separate query there is no dependence between the validation time of different constraints. We have not performed extensive performance analysis for IC validation but as a simple comparison we looked at logical consistency checking time vs. IC validation time. For LUBM(5), which has 100K individuals and 800K ABox axioms, logical consistency checking was in average 10s whereas validating a single IC took in average 2s. The naive approach in our prototype to execute each query separately would not scale well as the number of ICs increase. However, there are many improvement possibilities ranging from combining similar queries into a single query to running multiple queries in parallel.

7. Conclusions and Future Work

In this paper, we described how to provide an IC semantics for OWL axioms that can be used for data validation purposes. Our IC semantics provide intuitive results for various different use cases we examined. We presented translation rules from IC axioms to DCQ**not** queries, showing that IC validation can be reduced to query answering when the KB expressivity is $\mathcal{SRI}$ or constraint expressivity is $\mathcal{SROI}$.

Our preliminary results with a prototype IC validator implementation shows that existing OWL reasoners can be used for IC validation efficiently with little effort. Using SPARQL queries for IC validation makes our approach applicable to a wide range of reasoners. In the future, we will be looking at the performance of IC validation in realistic datasets and will be exploring the IC validation algorithms for the full expressivity of $\mathcal{SROIQ}$.

[5]http://clarkparsia.com/pellet/oicv-0.1.2.zip

[6]http://clarkparsia.com/pellet

[7]http://swat.cse.lehigh.edu/projects/lubm/

References

Angles, R., and Gutierrez, C. 2008. The expressive power of sparql. In *ISWC2008*, 114–129.

Baader, F., and Hanschke, P. 1991. A schema for integrating concrete domains into concept languages. In *Proceedings of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI91)*, 452–457.

Calvanese, D.; Giacomo, G. D.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. EQL-Lite: Effective first-order query processing in description logics. In *IJCAI2007*, 274–279.

Donini, F. M.; Lenzerini, M.; Nardi, D.; Nutt, W.; and Schaerf, A. 1998. An epistemic operator for description logics. *AI* 100(1–2):225–274.

Eiter, T.; Ianni, G.; Lukasiewicz, T.; Schindlauer, R.; and Tompits, H. 2008. Combining answer set programming with description logics for the semantic web. *AI* 172(12-13):1495–1539.

Horrocks, I.; Kutz, O.; and Sattler, U. 2006. The even more irresistible sroiq. In *KR2006*, 57–67. AAAI Press.

Lloyd, J. W. 1987. *Foundations of logic programming*.

Motik, B.; Horrocks, I.; and Sattler, U. 2007. Bridging the gap between owl and relational databases. In *WWW2007*, 807–816. ACM Press.

Motik, B.; Patel-Schneider, P. F.; and Grau, B. C. 2009. Owl 2 web ontology language direct semantics.

Motik, B. 2007. A faithful integration of description logics with logic programming. In *IJCAI2007*, 477–482.

Prud'hommeaux, E., and Seaborne, A. 2008. Sparql query language for rdf.

Reiter, R. 1988. On integrity constraints. In *TARK1988*, 97–111. Morgan Kaufmann.

Sirin, E., and Parsia, B. 2006. Optimizations for answering conjunctive abox queries. In *Proceedings of the 12th International Workshop on Description Logics (DL2006)*.