Data Exchange between Relational Knowledge Bases in the Web of Data

Tadeusz Pankowski
Institute of Control and Information Engineering, Poznań University of Technology, Poznań, Poland

Keywords: DL Knowledge Bases, Data Integration, Integrity Constraints, Data Exchange, Databases vs. Knowledge Bases.

Abstract: An important component of the Web of Data is formed by data originally stored in relational databases. The relational data along with its schemes and integrity constraints is translated into a knowledge base, that we call a *relational knowledge base* (RKB), residing on the Web. It is important to preserve semantics in data-to-knowledge transformation, as well as in knowledge-to-knowledge exchange between two RKBs. We discuss these issues and propose an algorithm for checking whether a mapping between two RKBs is semantics preserving. The algorithm is based on the chase procedure.

1 INTRODUCTION

Technologies of the Semantic Web enables web-wide integration of data coming from various sources. In this way the Web of Data is created and can be also perceived as a giant knowledge base. The extensional layer of this knowledge base consists of an RDF graph (or a corresponding OWL specification), and the intensional layer is a set of axioms (in RDFS or OWL). Very often the data presented in the Web comes from relational databases. Thus, the similarities and differences between databases and knowledge bases, and combining these technologies in data integration activities, has been an important and attractive field of research since many years (Abiteboul et al., 1995; Reiter, 1982; Motik et al., 2009). Now, as a formal foundation of knowledge bases serve Description Logics (DLs) (Baader et al., 2003), and DL knowledge base (or DL ontology) is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a set of axioms modeling the intensional knowledge (the TBox axioms), and $\mathcal{A}$ is a set of assertions forming the extensional knowledge (the ABox assertions).

Some recent results of representing relational databases in the Semantic Web are surveyed in (Sequeda et al., 2011) and some solutions were proposed in (Sequeda et al., 2012; Arenas et al., 2012; Poggi et al., 2008; Pankowski, 2012b; Pankowski, 2013a). A relationship between relational databases and DL knowledge bases has been studied in (Motik et al., 2009; Pankowski, 2012a).

There are three main differences between databases and knowledge bases making the translation between them difficult: (a) databases are based on CWA (*Closed World Assumption*) while knowledge bases on OWA (*Open World Assumption*); (b) databases accept UNA (*Unique Name Assumption*) while knowledge bases usually do not accept it; (c) integrity constraints in databases are interpreted as checks while in knowledge bases all rules are deductive rules. It turns out that incorporating integrity constraints into knowledge bases is the most challenging issue.

In this paper, we follow the concept of an *extended DL knowledge base* (EKB), where the set $\mathcal{T}$ of TBox axioms is divided into *standard* TBox axioms, $\mathcal{S}$, and *integrity constraint* TBox axioms, $\mathcal{C}$ (Motik et al., 2009). We will use the notion of EKB to represent a relational database in DL. We define a *data-to-knowledge exchange* (*dk-exchange*) system that defines translation of relational database schema, its integrity constraints and instances into an EKD referred to as a *relational knowledge base* (RKB). The semantics of data should not be lost by the translation, i.e. consistent (inconsistent) databases are transformed into consistent (inconsistent) knowledge bases. We propose and discuss an algorithm for checking whether a mapping between two RKBs is semantics preserving.

In Section 2 we introduce a running example, and in Section 3 we review some basic notions of relational databases. Translation of databases into RKBs is discussed in Section 4. In Section 5 an algorithm

Pankowski T..
Data Exchange between Relational Knowledge Bases in the Web of Data.
DOI: 10.5220/0004555903040309
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 304-309
ISBN: 978-989-8565-59-4

for reasoning about data exchange between RKBs is proposed. Section 6 concludes the paper.

2 MOTIVATING SCENARIO

As the running example we will consider ER diagrams in Figure 1 describing students, courses and exams taken by students, in databases corresponding to two universities, named *a* and *b*, respectively. In *a* (Figure 1(a)) a student is a specialization of a person. Farther on, all names will be prefixed by the corresponding database name (e.g. *a:Student, b:SId*).

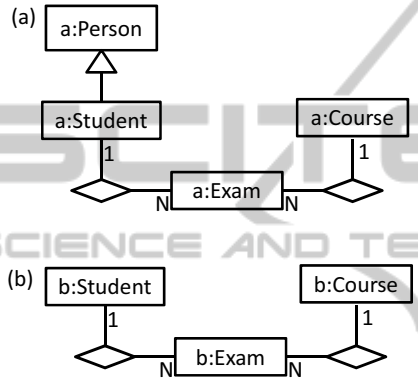

Figure 1: Two ER diagrams of two university domains.

Besides syntactic differences between *a* and *b*, there is also an important semantic difference between them: in *a*, *Faculty* is an attribute of *Student*, while in *b* – an attribute of *Exam* meaning that a student can be enrolled in many faculties. The corresponding relation schemes are listed in Figure 2.

```
a:Person(a:PId,a:Name)
a:Student(a:SId,a:Faculty)
a:Course(a:CId)
a:Exam(a:EId,a:ESId,a:Course,a:Grade)

b:Student(b:SId,b:Name)
b:Course(b:CId)
b:Exam(b:EId,b:ESId,b:Course,b:Faculty,b:Grade)
```

Figure 2: Relation schemes corresponding to ERDs.

There must be also some integrity constraints defined for these relation schemes, such as: *a:SId* is the primary key for *a:Student*, *a:SId* is also a foreign key referring to *a:PId* in *a:Person*, *a:Faculty* must be not `NULL`, *b:Name* can be `NULL`, etc.

In our scenario, we are interested in:

1. Creation of DL knowledge bases $\mathcal{K}_a$ and $\mathcal{K}_b$ representing databases DB_a and DB_b, i.e. creation of a *dk-exchange* system, that should be semantics preserving.
2. Reasoning about mappings between $\mathcal{K}_a$ and $\mathcal{K}_b$, and – in consequences – between DB_a and DB_b. The question is whether there exists a mapping and the corresponding data transformation that preserves information and semantics. Intuitively, we see that such a mapping can be defined from DB_a to DB_b but not inversely (because of differences in semantics of *Faculty* in both databases).

3 RELATIONAL DATABASES

A (relational) *database schema* (*db-schema*) is a pair $(\mathbf{R}, \mathbf{IC})$, where $\mathbf{R} = \{R_1, \ldots, R_n\}$ is a *relation schema* consisting of a set of *relation symbols*, and **IC** is a set of *integrity constraints* over **R**. Each *relation symbol* $R \in \mathbf{R}$ has a *type*, which is a nonempty finite set $att(R)$ of *attributes*. Without loss of generality, we can assume that types of relation symbols are pairwise disjoint.

Let **Const** be a countable infinite set of *constants*, and `NULL` be a reserved symbol not in **Const**. An *instance* I of **R** is a finite set of *facts* (or *atoms*) of the form $R(A_1 : c_1, \ldots, A_m : c_m)$, where $R \in \mathbf{R}$, $att(R) = \{A_1, \ldots, A_m\}$, and $c_i \in \mathbf{Const} \cup \{\texttt{NULL}\}$, $1 \leq i \leq m$.

Integrity constraints in databases play a dual role. They can be used in data reasoning tasks, such as checking the correctness of database data, as well as in schema reasoning tasks, such as computing query containment.

We assume that $\mathbf{IC} = \mathit{Unique} \cup \mathit{NotNull} \cup \mathit{PKey} \cup \mathit{FKey} \cup \mathit{Inherit}$, where:

1. *Unique* is a set of *unique integrity constraints*, i.e. expressions of the form $unique(R, A)$, where $R \in \mathbf{R}$, $A \in att(R)$. An instance I of **R** is consistent with $unique(R, A_k)$, if for every i, $0 \leq i \leq m$, I satisfies the formula
$$R(t_1) \wedge R(t_2) \wedge t_1.A_k = t_2.A_k \wedge t_1.A_k \neq \texttt{NULL} \wedge t_2.A_k \neq \texttt{NULL} \wedge t_1.A_i \neq \texttt{NULL} \wedge t_2.A_i \neq \texttt{NULL} \Rightarrow t_1.A_i = t_2.A_i.$$
2. *NotNull* is a set of *not-null integrity constraints*, i.e. expressions of the form $notnull(R, A_k)$. An instance I of **R** is consistent with $notnull(R, A_k)$, if for any fact $R(t) \in I$, $t.A_k$ is a constant, i.e. if I satisfies the formula
$$R(t) \Rightarrow t.A_k \neq \texttt{NULL}.$$
3. *PKey* is a set of *primary key integrity constraints*, i.e. expressions of the form $pkey(R, A_k)$. An instance I of **R** is consistent with $pkey(R, A_k)$ if it is consistent with $unique(R, A_k)$, and $notnull(R, A_k)$.
4. *FKey* is a set of *foreign key integrity constraints*. Let $R, R' \in \mathbf{R}$, $A \in att(R)$, and $A' \in att(R')$. A

foreign key integrity constraint is an expression of the form $fkey(R,A,R',A')$. An instance I of $\mathbf{R}$ is consistent with $fkey(R,A,R',A')$ if I satisfies $unique(R',A')$, and

$$R(t) \wedge t.A \neq \texttt{NULL} \Rightarrow \exists t'.(R'(t') \wedge t'.A' = t.A)$$

5. *Inherit* is a set of *inheritance integrity constraints*, i.e. pairs of the form $(pkey(R,A), fkey(R,A,R',A'))$. An instance I of $\mathbf{R}$ is consistent with $(pkey(R,A), fkey(R,A,R',A'))$, if I satisfies both $pkey(R,A)$ and $fkey(R,A,R',A')$.

Let $(\mathbf{R},\mathbf{IC})$ be a db-schema and I be an instance of $\mathbf{R}$. A database $\mathrm{DB} = (\mathbf{R},\mathbf{IC},I)$ is consistent, if I satisfies (is a model of) all integrity constraints, denoted $I \models \mathbf{IC}$. Otherwise we say that DB is inconsistent.

For the database DB_a with relation schema in Figure 2, we assume (the prefix a : is omitted):

$\mathbf{IC}_a = \{pkey(Person,PId), pkey(Student,SId), pkey($
$Course,CId), pkey(Exam,EId), fkey(Student,SId,$
$Person,PId), pkey(Exam,ESId,Student,SId),$
$pkey(Exam,Course,Course,CId), notnull(Student,$
$Faculty), notnull(Exam,Grade)\}$.

Analogously, for DB_b. Note, that *Name* can be NULL in both databases.

4 DK-EXCHANGE

4.1 Translation of a Database

While translating a relational database into a DL knowledge base, the following should be taken into account:

1. A traditional DL knowledge base understood as a pair $(\mathcal{T},\mathcal{A})$ is unable to model integrity constraints (Motik et al., 2009). The reason is two-fold: firstly, axioms in $\mathcal{T}$ are interpreted under the standard first-order semantics and are treated as deductive rules and not as checks, and secondly, the UNA is not accepted in general in DL knowledge bases, it means that two different individual names can denote the same individual.
2. In the translation, semantics of the database should be preserved, i.e. any consistent (inconsistent) database should be translated into a consistent (inconsistent) DL knowledge base.

Now, we define a *relational knowledge base* (RKB) that is a DL knowledge base adequately representing a relational database. RKB is based on the concept of EKB (Motik et al., 2009). We propose and discuss a system of TBox axioms, which properly represents a relational database defined in the previous section.

A *relational knowledge base* is a tuple $\mathrm{RKB} = (\mathsf{N},\mathcal{S},\mathcal{C},\mathcal{A})$, where:

1. N is the *vocabulary* of RKB, consisting of a set $\mathsf{N}_{\mathrm{Ind}}$ of *individual names*, a set N_{Cl} of *class names* (or *atomic concepts*), a set N_{OP} of *object property names* (or *atomic roles*).
2. $\mathcal{S}$ is a finite set of *standard* TBox axioms, which are treated as deductive rules and can infer new assertions.
3. $\mathcal{C}$ is a finite set of *integrity constraint* TBox axioms, which are treated as checks, and must be satisfied by any minimal Herbrand model of the set of assertions implied by $\mathcal{A}$ and $\mathcal{S}$. Axioms in $\mathcal{C}$ cannot imply new assertions.
4. $\mathcal{A}$ is a set of ABox *assertions*, i.e. class memberships and properties of individual objects.

The translation is made by a *data-to-knowledge exchange* (*dk-exchange*) system $\mathcal{M} = (\tau,\Sigma)$, such that for each db-schema $(\mathbf{R},\mathbf{IC})$ and every instance I of $\mathbf{R}$, $\mathcal{M}(\mathbf{R},\mathbf{IC},I) = (\tau(\mathbf{R},\mathbf{IC}),\Sigma(I)) = (\mathsf{N},\mathcal{S},\mathcal{C},\mathcal{A})$, where $\tau(\mathbf{R},\mathbf{IC}) = (\mathsf{N},\mathcal{S},\mathcal{C})$, and $\Sigma(I) = \mathcal{A}$.

Creating Vocabulary. Let Δ_{Var} be a countable infinite set of *labeled nulls* disjoint from the set of constants. Labeled nulls, denoted $X,V,X_1,V_1,...$, are used as "fresh" Skolem terms, which are placeholders for unknown values, and can thus be seen as *variables* (Fagin et al., 2005). The vocabulary $\mathsf{N} = \mathsf{N}_{\mathrm{Ind}} \cup \mathsf{N}_{\mathrm{Cl}} \cup \mathsf{N}_{\mathrm{OP}}$, is created as follows: (1) The set $\mathsf{N}_{\mathrm{Ind}}$ of individual names consists of the union of **Const** and Δ_{Var}. (2) There are predefined class names Tuple and Val of, respectively, individuals called *tuples* and individuals called *attribute values*. (3) For each relation symbol $R \in \mathbf{R}$, there is a class name $C_R \in \mathsf{N}_{\mathrm{Cl}}$, every individual in C_R is a tuple. (4) For each attribute $A \in att(R)$, there is a class name $C_A \in \mathsf{N}_{\mathrm{Cl}}$ (every individual in C_A is an attribute value), and an object property name $P_A \in \mathsf{N}_{\mathrm{OP}}$; the object property P_A connects tuples in C_R with attribute values in C_A.

Creating Standard TBox Axioms. The set $\mathcal{S}$ of standard TBox axioms is given in Table 1. All these axioms are deductive rules.

Table 1: Standard TBox axioms of relational knowledge base.

	Constraints of relational db	DL
S1	$R \in \mathbf{R}$	$C_R \sqsubseteq \mathsf{Tuple}$
S2	$A \in att(R), R \in \mathbf{R}$	$C_A \sqsubseteq \mathsf{Val}$
S3	range of P_A	$\exists P_A^- \sqsubseteq C_A$
S4	domain of P_A	$\exists P_A \sqsubseteq C_R$
S5	$unique(R,A)$	$(\mathtt{func}\ P_A^-)$
S6	$(pkey(R,A), fkey(R,A,R',A'))$	$P_A \sqsubseteq P_{A'}$

$(S1)$ and $(S2)$ belong to translation of facts that $R \in \mathbf{R}$ and $A \in att(R)$; they say that all tuple names in C_R,

and all attribute value names in C_A must be inserted into classes Tuple and Val, respectively. $(S3)$ and $(S4)$ belong to translation of the fact that $A \in att(R)$, where: $(S3)$ says that any individual belonging to the range of P_A must be inserted into C_A, and any individual belonging to the domain of P_A must be inserted into C_R. $(S5)$ is result of translation of a unique constraint $unique(R,A)$, and enforces equality between x_1 and x_2, if all $P_A(x_1,v_1)$, $P_A(x_2,v_2)$, and $v_1 = v_2$ hold. $(S6)$ results of the translation of an inheritance constraint $(pkey(R,A), fkey(R,A,R',A'))$, and says that extension of P_A must be inserted into the extension of $P_{A'}$.

Creating Integrity Constraint TBox Axioms. The set $\mathcal{C}$ of TBox ic-axioms is given in Table 2. Note that ic-axioms are checks, so we expect that the value of such an axiom is either `TRUE` or `FALSE`.

Table 2: Integrity constraint TBox axioms of relational knowledge base.

	Constraints of relational db	DL
C1	disjointness	$\mathsf{Tuple} \sqsubseteq \neg\mathsf{Val}$
C2	$A \in att(R), R \in \mathbf{R}$	$(\mathtt{func}\ P_A)$
C3	$notnull(R,A)$	$C_R \sqsubseteq \exists P_A$
C4	$fkey(R,A,R',A')$	$\exists P_A^- \sqsubseteq \exists P_{A'}^-$
C5	$(pkey(R,A), fkey(R,A,R',A'))$	$C_R \sqsubseteq C_{R'}$

$(C1)$ tests disjointness of Tuple and Val. $(C2)$ belongs to translation of $A \in att(R)$ and checks if P_A has the functional property. $(C3)$ is result of translation of $notnull(R,A)$ and tests if any tuple name in C_R is in domain of P_A. $(C4)$ is result of translation of $fkey(R,A,R',A')$, and tests the inclusion of the range of P_A in the range of $P_{A'}$. $(C5)$ belongs to the result of the translation of $(pkey(R,A), fkey(R,A,R',A'))$, and tests the inclusion of C_R in $C_{R'}$.

Creating ABox Assertions. ABox assertions are expressions of the form: $C(a)$, $P(a_1,a_2)$, and $a_1 = a_2$, where $C \in \mathsf{N}_{\mathrm{Cl}}$, $P \in \mathsf{N}_{\mathrm{OP}}$, and $a, a_1, a_2 \in \mathsf{N}_{\mathrm{Ind}}$. Translation of an instance I of $\mathbf{R}$ can be performed using Algorithm 1.

Algorithm 1: Creating ABox assertions.

Input: Instance I of $\mathbf{R}$, and an empty ABox $\mathcal{A}$.
Output: ABox assertions in $\mathcal{A}$ representing I.
for each $R(t) \in I$
  $U_{R,t} := \{A \in att(R) \mid t.A \neq \mathtt{NULL}\}$
  $X :=$ a fresh labeled null in Δ_{Var}
  if $U_{R,t} = \emptyset$ **then**
    $\mathcal{A} := \mathcal{A} \cup \{C_R(X)\}$
  else
  for each $A \in U_{R,t}$
    $\mathcal{A} := \mathcal{A} \cup \{P_A(X, t.A)\}$
end

4.2 Semantics Preservation

One of the most challenging issues in dk-exchange is to show that the semantics of the source data is not lost by the transformation into a knowledge base. The preservation of semantics of a dk-exchange system $\mathcal{M} = (\tau, \Sigma)$ can be understood in two ways:

1. *Soundness*. $\mathcal{M} = (\tau, \Sigma)$ is *sound* w.r.t. *semantics preservation* if every consistent database $(\mathbf{R}, \mathbf{IC}, I)$ is transformed into a consistent relational knowledge base $(\mathsf{N}, \mathcal{S}, \mathcal{C}, \mathcal{A})$, i.e.

$$I \models \mathbf{IC} \wedge \tau(\mathbf{R}, \mathbf{IC}) = (\mathsf{N}, \mathcal{S}, \mathcal{C}) \wedge \Sigma(I) = \mathcal{A} \Rightarrow \mathcal{A} \cup \mathcal{S} \models_{mHm} \mathcal{C}.$$

2. *Completeness*. $\mathcal{M} = (\tau, \Sigma)$ is *complete* w.r.t. *semantics preservation* if every inconsistent database $(\mathbf{R}, \mathbf{IC}, I)$ is transformed into an inconsistent relational knowledge base $(\mathsf{N}, \mathcal{S}, \mathcal{C}, \mathcal{A})$, i.e.
$I \not\models \mathbf{IC} \wedge \tau(\mathbf{R}, \mathbf{IC}) = (\mathsf{N}, \mathcal{S}, \mathcal{C}) \wedge \Sigma(I) = \mathcal{A} \Rightarrow \mathcal{A} \cup \mathcal{S} \not\models_{mHm} \mathcal{C}$.

It can be shown (Pankowski, 2013b) that the dk-exchange system $\mathcal{M} = (\tau, \Sigma)$, is both sound and complete w.r.t. semantics preservation.

5 REASONING ABOUT KBS-MAPPING

A *knowledge base schema mapping* (*kbs-mapping*) from a source kb-schema $\mathsf{R}_s = (\mathsf{N}_s, \mathcal{S}_s, \mathcal{C}_s)$ to a target kb-schema $\mathsf{R}_t = (\mathsf{N}_t, \mathcal{S}_t, \mathcal{C}_t)$, is defined by a finite set Γ_{st} of *source to target dependencies* (STDs) (Fagin et al., 2005), i.e. implications of the form

$$\forall \mathbf{x}, \mathbf{v}. (\varphi_s(\mathbf{x}, \mathbf{v}) \Rightarrow \exists \mathbf{x}', \mathbf{v}'. \varphi_t(\mathbf{x}, \mathbf{x}'\mathbf{v}, \mathbf{v}')),$$

where φ_s and φ_t are conjunctions of atomic formulas over N_s and N_t, respectively.

Example 5.1. *For the knowledge bases corresponding to databases in Figure 2, we can define the following kbs-mappings:*

$$\begin{aligned}
&\Gamma_{ab} = \{\\
&a : p(x,v) \Rightarrow b : p(x,v),\\
&\quad \text{for } p \in \{SId, CId, EId, ESId, Course, Grade\},\\
&a : Name(x,v) \wedge a : Student(x) \Rightarrow b : Name(x,v),\\
&a : SId(x,v_1) \wedge a : Faculty(x,v_2) \wedge a : EId(y,v_3) \wedge\\
&\quad a : ESId(y,v_1) \Rightarrow b : Faculty(y,v_2)\}
\end{aligned}$$

$$\begin{aligned}
&\Gamma_{ba} = \{\\
&b : p(x,v) \Rightarrow a : p(x,v), \ \text{ for } p \in \{SId,\\
&\quad Name, CId, EId, ESId, Course, Grade\},\\
&b : ESId(x,v_1) \wedge b : Faculty(x,v_2) \Rightarrow\\
&\quad \exists y. (a : SId(y,v_1) \wedge a : Faculty(y,v_2))\}
\end{aligned}$$

The crucial problem is if any *consistent source knowledge base* is transformed by the given set Γ of

STDs, into a *consistent target knowledge base*. It can be easily seen for our running example that R_a and R_b are not semantically equivalent – integrity constraints for R_a are more restrictive than those of R_b. Thus, we can expect that:

- any consistent knowledge base with schema R_a is transformed via Γ_{ab} into a consistent knowledge base with schema R_b;
- there is a consistent knowledge base with schema R_b that is transformed via Γ_{ba} into an inconsistent knowledge base with schema R_a.

In order to perform such reasoning, we use the *chase procedure* (Maier et al., 1979; Fagin et al., 2005). Input, output and steps of this procedure are as follows:

1. *Input.* A source kb-schema $\mathsf{R}_s = (\mathsf{N}_s, \mathcal{S}_s, \mathcal{C}_s)$ representing a db-schema $(\mathbf{R}_s, \mathbf{IC}_s)$, a target kb-schema $\mathsf{R}_t = (\mathsf{N}_t, \mathcal{S}_t, \mathcal{C}_t)$, and a set Γ_{st} of STDs from R_s to R_t.
2. *Output.* The decision whether Γ_{st} maps any consistent knowledge base with the kb-schema R_s into a consistent knowledge base with the schema R_t.
3. *Steps.* (1) Construct a tableau $\mathcal{A}_s$ of assertions such that $(\mathsf{N}_s, \mathcal{S}_s, \mathcal{C}_s, \mathcal{A}_s)$ forms a consistent knowledge base. Moreover, $\mathcal{A}_s$ should be a "well suited" source instance for the next steps in the chase procedure. (2) Proceed the chase from $\mathcal{A}_s$ to $\mathcal{A}_t$ using Γ_{st} and axioms from $\mathcal{S}_t$. (3) Verify consistency of $(\mathsf{N}_t, \mathcal{S}_t, \mathcal{C}_t, \mathcal{A}_t)$.

Algorithm 2: Constructing the tableau $\mathcal{A}_s$.

$\mathcal{A}_0 := \emptyset$;
for each $\gamma \in \Gamma_{st}$.
  let $\varphi_s(\mathbf{x}, \mathbf{v})$ be the left-hand side of γ;
  if $\varphi_s(\mathbf{x}, \mathbf{v})$ consists of one atom $P_A(x, v)$ **then**
    Modify $\mathcal{A}_0$ in such a way that the formula

$$\exists x_1, x_2, v. P_A(x_1, v) \wedge P_A(x_2, v)$$

    is satisfied in $\mathcal{A}_0$.
  if $\varphi_s(\mathbf{x}, \mathbf{v})$ consists of one atom $C_R(x)$ **then**
    Modify $\mathcal{A}_0$ in such a way that the formula

$$\exists x. C_R(x)$$

    is satisfied in $\mathcal{A}_0$.
  else // there are more atoms than one in $\varphi_s(\mathbf{x}, \mathbf{v})$
    Let $\mathbf{v} = (v_1, ..., v_n)$ and $\mathbf{v}' = (v'_1, ..., v'_n)$ be disjoint sets of variables. By $\omega = [v_1 \mapsto w_1, ..., v_n \mapsto w_n]$, where $w_i \in \{v_i, v'_i\}$ we denote a substitution replacing v_i either with itself or with v'_i. The set Ω of all such substitutions has 2^n elements. Then, from $\varphi_s(\mathbf{x}, \mathbf{v})$ we obtain the following formula consisting of 2^n conjunctions

$$\exists \mathbf{x}, \mathbf{v}, \mathbf{v}'. (\varphi_s(\mathbf{x}, \mathbf{v})[\omega_1] \wedge \cdots \wedge \varphi_s(\mathbf{x}, \mathbf{v})[\omega_{2^n}] \wedge \mathbf{v} \neq \mathbf{v}').$$

    For each $\omega \in \Omega$ determine a substitution ν_ω of variables in $\mathbf{x}$ with a newly invented variable names $\mathbf{x}_\omega$, denoted $\nu_\omega = [\mathbf{x} \mapsto \mathbf{x}_\omega]$. Then the following formula is created

$$\Phi \equiv \exists \mathbf{x}_{\omega_1}, \ldots, \mathbf{x}_{\omega_{2^n}}, \mathbf{v}, \mathbf{v}'. (\varphi_s(\mathbf{x}_{\omega_1}, \mathbf{v})[\omega_1] \wedge \cdots \wedge \varphi_s(\mathbf{x}_{\omega_{2^n}}, \mathbf{v})[\omega_{2^n}] \wedge \mathbf{v} \neq \mathbf{v}').$$

    Modify $\mathcal{A}_0$ so that $\mathcal{A}_0$ satisfies Φ.
end for each $R \in \mathbf{R}, A \in att(R)$
  if $notnull(R, A) \notin \mathbf{IC}$ **then**
    Modify $\mathcal{A}_0$ in such a way that the formula

$$\exists x. C_R(x) \wedge \neg \exists v. P_A(x, v)$$

    is satisfied in $\mathcal{A}_0$.
end

Closing. Chase with respect to $\mathcal{S}_s \cup \mathcal{C}_s$, i.e.

$$\mathcal{A}_0 \xrightarrow{\mathcal{S}_s \cup \mathcal{C}_s} \mathcal{A}_s$$

Transformation. Chase with respect to Γ_{st}, i.e.

$$\mathcal{A}_s \xrightarrow{\Gamma_{st}} \mathcal{A}_1$$

Repairing. Chase with respect to $\mathcal{S}_t$, i.e.

$$\mathcal{A}_1 \xrightarrow{\mathcal{S}_t} \mathcal{A}_t$$

Verifying. It must be checked if axioms in $\mathcal{C}_t$ are satisfied in $\mathcal{A}_t$, i.e. if the following entailment holds:

$$\mathcal{A}_t \models_{mHm} \mathcal{C}_t.$$

Example 5.2. *For Γ_{ba} in Example 5.1, we have the following formulas mentioned in Algorithm 2:*

$$\exists x_1, x_2, v. b : p(x_1, v) \wedge b : p(x_2, v),$$
$$\text{for } p \in \{SId, Name, CId, EId, ESId, Course, Grade\},$$
$$\exists x_1, x_2, x_3, x_4, v_1, v_2, v'_1, v'_2. (p : ESId(x_1, v_1) \wedge b : Faculty(x_1, v_2) \wedge p : ESId(x_2, v_1) \wedge b : Faculty(x_2, v'_2) \wedge p : ESId(x_3, v'_1) \wedge b : Faculty(x_3, v_2) \wedge p : ESId(x_4, v'_1) \wedge b : Faculty(x_4, v'_2) \wedge v_1 \neq v'_1 \wedge v_2 \neq v'_2),$$
$$\exists x. b : Student(x) \wedge \neg \exists v. b : Name(x, v).$$

We start the chase procedure with the first formula in Example 5.2. Then we have $b : SId(X_1, V_1)$ and $b : SId(X_2, V_1)$. Next, using $(S4)$ and $(S5)$, we obtain $b : Student(X_1)$ and $X_1 = X_2$. The final form of $\mathcal{A}_b$ is presented in Figure 3 (prefixes $b :$ are omitted), where additionally: $V_6 = V_1, V_{12} = V_3$, $V_8 = V_{15} = V_{16} = V_4$, and $V_6 \neq V_{12}, V_{10} \neq V_{11}$.

A fragment of the tableau $\mathcal{A}_a$, being the result of applying Γ_{ba} to $\mathcal{A}_b$, is presented in Figure 4 (again,

Student	*SId*	*Name*
X_1	V_1	V_2
X_3	V_3	V_2
X_{11}	V_{19}	

Course	*CId*
X_4	V_4

Exam	*EId*	*ESId*	*Course*	*Faculty*	*Grade*
X_6	V_5	V_6	V_8	V_{10}	V_9
X_8	V_7	V_6	V_8	V_{11}	V_9
X_9	V_{13}	V_{12}	V_{15}	V_{10}	V_{17}
X_{10}	V_{14}	V_{12}	V_{16}	V_{11}	V_{18}

Figure 3: Tabular representation of $\mathcal{A}_b$ as an input tableau to Γ_{ba}.

Person	*PId*	*Name*
X_1	V_1	V_2
X_3	V_3	V_2
X_{11}	V_{19}	

Student	*SId*	*Faculty*
X_1	V_1	V_{10}
X_1	V_1	V_{11}
X_3	V_3	V_{10}
X_3	V_3	V_{11}

Figure 4: Tabular representation of a fragment of $\mathcal{A}_a$ being the result of applying Γ_{ba} against $\mathcal{A}_b$ from Figure 3.

prefixes a : are omitted). We see that consistency of $\mathcal{A}_a$ requires that $V_{10} = V_{11}$. However, this equality contradicts the assumption in $\mathcal{A}_b$ (i.e. $V_{10} \neq V_{11}$). Thus, the target knowledge base is inconsistent. We see that Γ_{ba} does not preserve semantics, because a consistent knowledge base with kb-schema R_b is transformed into inconsistent knowledge base with kb-schema R_a. In this case, the reason is that kb-schemes R_a and R_b are not semantically equivalent.

6 CONCLUSIONS

In this paper we discuss the problem of semantics preservation in data exchange between two relational knowledge bases (RKBs) in the Web of Data. RKBs are important components of Web of Data since they arise as results of translation relational databases along with their integrity constraints into knowledge bases. In this paper we adapt the concept of DL extended knowledge bases (Motik et al., 2009). Data exchange between RKBs is a vital problem in data integration over the Web (Brzykcy et al., 2008). We sketch an algorithm that checks whether a given mapping between two RKBs is semantics preserving, that is whether it maps a consistent source RKB into a consistent target RKB.

REFERENCES

Abiteboul, S., Hull, R., and Vianu, V. (1995). *Foundations of Databases*. Addison-Wesley, Reading, Massachusetts.

Arenas, M., Bertails, A., Prud'hommeaux, E., and Sequeda, J. (2012). A Direct Mapping of Relational Data to RDF. http://www.w3.org/TR/rdb-direct-mapping.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Petel-Schneider, P., editors (2003). *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

Brzykcy, G., Bartoszek, J., and Pankowski, T. (2008). Schema Mappings and Agents' Actions in P2P Data Integration System. *Journal of Universal Computer Science*, 14(7):1048–1060.

Fagin, R., Kolaitis, P. G., Miller, R. J., and Popa, L. (2005). Data exchange: semantics and query answering. *Theor. Comput. Sci*, 336(1):89–124.

Maier, D., Mendelzon, A. O., and Sagiv, Y. (1979). Testing implications of data dependencies. *ACM Trans. Database Syst.*, 4(4):455–469.

Motik, B., Horrocks, I., and Sattler, U. (2009). Bridging the gap between OWL and relational databases. *Journal of Web Semantics*, 7(2):74–89.

Pankowski, T. (2012a). Using Data-to-Knowledge Exchange for Transforming Relational Databases to Knowledge Bases. In *RuleML 2012, LNCS 7438*, pages 256–263. Springer.

Pankowski, T. (2012b). Using OWL ontology for reasoning about schema mappings in data exchange systems, *KES 2012. Frontiers in Artificial Intelligence and Applications*, Vol. 243.

Pankowski, T. (2013a). Reasoning About Consistency Of Relational Knowledge Bases. In *ICCGI 2013, Nice, France*, pages 1–6. IARIA.

Pankowski, T. (2013b). Semantics Preservation In Data-To-Knowledge Exchange From Relational Databases To Knowledge Bases. *(submitted)*.

Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., and Rosati, R. (2008). Linking data to ontologies. In *Journal on Data Semantics X*, pages 133–173. Springer-Verlag.

Reiter, R. (1982). Towards a logical reconstruction of relational database theory. In *On Conceptual Modelling. Perspectives from Artificial Intelligence, Databases, and programming Languages*, pages 191–233.

Sequeda, J., Arenas, M., and Miranker, D. P. (2012). On Directly Mapping Relational Databases to RDF and OWL (Extended Version). *CoRR*, abs/1202.3667:1–17.

Sequeda, J., Tirmizi, S. H., Corcho, Ó., and Miranker, D. P. (2011). Survey of directly mapping SQL databases to the Semantic Web. *Knowledge Eng. Review*, 26(4):445–486.