Updating Wikipedia via DBpedia Mappings and SPARQL

Albin Ahmeti[1], Javier D. Fernández[1,2], Axel Polleres[1,2], and Vadim Savenkov[1]
[1] Vienna University of Economics and Business, Vienna, Austria
[firstname.lastname]@wu.ac.at
[2] Complexity Science Hub Vienna, Vienna, Austria

Abstract. DBpedia crystallized most of the concepts of the Semantic Web using simple mappings to convert Wikipedia articles (i.e., infoboxes and tables) to RDF data. This "semantic view" of wiki content has rapidly become the focal point of the Linked Open Data cloud, but its impact on the original Wikipedia source is limited. In particular, little attention has been paid to the benefits that the semantic infrastructure can bring to maintain the wiki content, for instance to ensure that the effects of a wiki edit are consistent across infoboxes. In this paper, we present an approach to allow ontology-based updates of wiki content. Starting from DBpedia-like mappings converting infoboxes to a fragment of `OWL 2 RL` ontology, we discuss various issues associated with translating SPARQL updates on top of semantic data to the underlying Wiki content. On the one hand, we provide a formalization of DBpedia as an Ontology-Based Data Management framework and study its computational properties. On the other hand, we provide a novel approach to the inherently intractable update translation problem, leveraging the pre-existent data for disambiguating updates.

1 Introduction

DBpedia [25] is a community effort that has created the most important cross-domain dataset in RDF [8] in the focal point of the Linked Open Data (LOD) cloud [4]. At its core is a set of declarative mappings extracting data from Wikipedia *infoboxes* and tables into RDF. However, DBpedia makes knowledge machine readable only, rather than also *machine writable*. This not only restricts the possibilities of automatic curation of the DBpedia data that could be semi-automatically propagated back to Wikipedia, but also prevents maintainers from evaluating the impact of their edits on the consistency of knowledge; indeed, previous work confirms that there are such inconsistencies discoverable in DBpedia [7, 12] arising most likely from inconsistent content in Wikipedia itself with respect to the mappings and the DBpedia ontology. Excluding the DBpedia taxonomy from the editing cycle is thus a — as we will show, unnecessary — drawback, but rather can be turned into an advantage for helping editors to create and maintain consistent content inside infoboxes, which we aim to address.

To this end, in this paper we want to make a case for DBpedia as a practical, real-world benchmark for Ontology-Based Data Management (OBDM) [26]. Although based on fairly restricted mappings—which we cast as a variant of so-called nested tuple-generating dependencies (tgds) herein—and minimalistic TBox language, accommodating DBpedia updates is intricate from different perspectives. The challenges are both conceptual (what is an adequate semantics for DBpedia SPARQL updates?) and

practical, when having to cope with high ambiguity of update resolutions. While general updates in OBDM remain largely infeasible, we still arrive at reasons to believe, that for certain use cases within DBpedia updates, reasonable and practically usable conflict resolution policies could be defined; we present the first serious attempt with DBpedia as a potential benchmark use case in this area.

Pushing towards the vision of a "Read/Write" Semantic Web,[3] the unifying capabilities of SPARQL extend beyond the mere querying of heterogeneous data. Indeed, the standardization of update functionality introduced in SPARQL1.1 renders SPARQL as a strong candidate for the role of web data manipulation language. For a concrete motivation example consider Listing 1, where a simple SPARQL Update request would reflect a recent merger of French administrative regions: for each settlement belonging to either Upper or Lower Normandy, we set the corresponding administrative attribution property to be just Normandy. In our scenario, the user should have means to write this update in SPARQL and let it be reflected in the underlying Wikipedia data.

Despite clear motivation, updates in the information integration setting abound with all sorts of challenges, starting from obvious data security concerns, to performance, data quality issues and, last but not least the technical issues of side effects and lack of unique semantics, demonstrated already in the classical scenarios such as database views and deductive databases [5, 10]. Although based on a very special join-free mapping language, the DBpedia setting is no different in this respect. With a high-quality curated data source at the backend, we set our goal not at ultimate transparency and automatic translation of updates, but rather at maximally support users in choosing the most economic and *typical* way of accommodating an update while maintaining (or at least, not degrading) consistency and not losing information inadvertently. As for DBpedia, if such RDF frontend systems have their own taxonomy (TBox) with also class and property disjointness assertions as well as functionality of properties, updates can result in inconsistencies with the data already present. In particular, we make the following contributions in this paper:

- we formalize the actual ontology language used by DBpedia as an `OWL 2 RL` fragment, and DBpedia mappings as a variant of so-called nested tuple-generating dependencies (tgds); based on this formalization
- we propose a semantics of OBDA updates for DBpedia and its Wikipedia mappings
- we discuss how such updates can be practically accommodated by suitable conflict resolution policies: the number of consistent revisions are in the worst case exponential in the size of the mappings and the TBox, so we investigate policies for choosing the "most reasonable" ones, e.g. following existing patterns in the data, that is choosing most popular fields in the present data to be filled upon inserts.

Note that, since neither the SPARQL Update language [15] nor the SPARQL Entailment regimes [16] specification covers the behaviour of updates in the presence of TBox axioms, the choice of semantics in such cases remains up to application designers. In [2, 3] we have discussed how SPARQL updates relating to the ABox can be implemented with TBoxes allowing no or limited form of inconsistency (class disjointness), a work we partially build upon herein: as a requirement from this prior work (as a consequence

[3] cf. https://www.w3.org/community/rww/

```
DELETE { ?X :region :Upper_Normandy . ?Y :region :Lower_Normandy .}
INSERT { ?X :region :Normandy . ?Y :region :Normandy}
WHERE { {?X :region :Upper_Normandy} UNION {?Y :region :Lower_Normandy} }
```

Listing 1: SPARQL 1.1. Update that merges two regions in France.

of the common postulates for updates in belief revision), such an update semantics needs to ensure that no mutually inconsistent pairs of triples are inserted in the ABox. In order to achieve this, a policy of conflict resolution between the new and the old knowledge is needed. To this end, in our earlier work [3] we defined *brave*, *cautious* and *fainthearted* semantics of updates. Brave semantics removes from the knowledge base all facts clashing with the inserted data. Cautious semantics discards entirely an update if it is inconsistent w.r.t. knowledge base, otherwise brave semantics is applied. Fainthearted semantics is in-between the two, amounts to adding an additional filter to the WHERE clause of SPARQL update in order to discard variable bindings which make inserted facts contradict prior knowledge. In the present work, we stick to these three basic cases, extending them to the OWL fragment used by DBpedia. However, since our goal is to accommodate updates as Wiki infobox revisions for which no batch update language exists, we restrict our considerations to *grounded updates* (u^+, u^-) of triples over URIs and literals that are to be inserted or, respectively, deleted (instead of considering the whole general SPARQL Update language).[4]

The rest of the paper is organized as follows. Section 2 provides our formalization on the DBpedia ontology and mapping language, defining the translation of Wiki updates to DBpedia updates and their (local) consistency. Section 3 outlines the main sources of worst-case complexity for automatic update translation that cannot be mitigated by syntactic restrictions of the mapping language. Section 4 discusses our pragmatic approach to OBDM in the DBpedia setting including our specific update conflict resolution strategies for DBpedia. Section 5 gives an overview of related work, and finally Section 6 provides concluding remarks.

2 The DBpedia OBDM Setting

We define the declarative **WikiDBpedia framework** (WDF) $\mathcal{F}$ as a triple $(\mathbf{W}, \mathcal{M}, \mathcal{T})$ where $\mathbf{W}$ is a relational schema encoding the infoboxes, $\mathcal{M}$ is a set of rules transforming it into RDF triples (the DBpedia ABox), and $\mathcal{T}$ is a TBox. The rules in $\mathcal{M}$ are given by a custom-designed declarative DBpedia mapping language [20]. This language can be captured by the language of *nested tuple generating dependences* (nested tgds) [13, 23], enhanced with negation in the rule bodies and interpreted functions for arithmetics, date, string and geocoordinate processing.

A *WDF instance* of a WDF $(\mathbf{W}, \mathcal{M}, \mathcal{T})$ is an infobox instance I satisfying $\mathbf{W}$. We now specify the language used to formalize the TBox $\mathcal{T}$, the tgds language of $\mathcal{M}$ and the infobox schema $\mathbf{W}$.

[4] We emphasize though that such an extension is a fairly straightforward extension of the discussions in [3], since general SPARQL Updates can be viewed as templates which are instantiated into exactly such sets of INSERTed and DELETed triples.

TYPE OF MAPPINGS	DECLARED	DESCRIPTION
Template	958	Map Wiki templates to DBpedia classes.
Property	19,972	Map Wiki template properties to DBpedia properties.
IntermediateNode	107	Generate a blank node with a URI.
Conditional	31	Depend on template properties and their values.
Calculate	23	Compute a function over two properties.
Date	106	Mappings that generate a starting and ending date.

Table 1: Description of DBpedia (English) mappings.

DBpedia ontology language. DBpedia uses a fragment of OWL 2 RL profile, which we call DBP. It includes the RDF keywords `subClassOf` (which we abbreviate as sc), `subPropertyOf` (sp), `domain` (dom) and `range` (rng), `disjointWith` (dw), `propertyDisjointWith` (pdw), `inversePropertyOf` (inv) as well as `functionalProperty` (func). At present, functional properties in DBpedia are limited to data properties, and inverse functional roles are not used.

Many concepts in the actual DBpedia are copied from external ontologies like Yago [31] and UMBEL[5]. All DBpedia resources also instantiate the concepts in DBpedia ontology, with the namespace http://dbpedia.org/ontology, to which we refer as DBP. They can be listed by the following SPARQL query:

```
SELECT DISTINCT ?x WHERE {{?x a owl:Class}
   UNION {?x a owl:ObjectProperty}
   UNION {?x a owl:DatatypeProperty}
   FILTER(strstarts(str(?x), "http://dbpedia.org/"))}
```

As of December 2016, this query retrieves 758 concepts, 1104 object and 1756 datatype properties for the English Live DBpedia[6]. Herewith, we only consider the facts from this core vocabulary set instantiated with the set of DBpedia mappings $\mathcal{M}$, and not the imported assertions from the external ontologies. We denote this vocabulary by $\mathbf{T}$ and, analogously to the infobox part of the system, call it "schema".

Infobox schema $\mathbf{W}$. Each Wiki page is identified by a URI which translates to a subject IRI in DBpedia. A page can contain several *infoboxes* of distinct *types*. We model this semistructured data store using a relational schema $\mathbf{W}$ with two ternary relations $W_i = \mathsf{UTI}$ and $W_d = \mathsf{IPV}$, attribute I storing infobox identifiers, U page URI, T infobox type, and P and V being respectively property names and values. That is, unlike the real Wiki where infoboxes may belong to different pages or be separate tables of distinct types, we use an auxiliary surrogate key I to horizontally partition the single key-value store W_d. Our schema $\mathbf{W}$ assumes key constraints $\mathsf{UT} \to \mathsf{I}$, $\mathsf{IP} \to \mathsf{V}$ and the inclusion dependency $W_d[\mathsf{I}] \subseteq W_i[\mathsf{I}]$ which we encode as the set of rules $\mathcal{W}$:

$$\begin{aligned}\mathcal{W} = \{&\forall i \forall p \forall v \big(W_d(i,p,v) \to \exists u \exists t\, W_i(u,t,i)\big),\\ &\forall u \forall t \forall i_1 \forall i_2 \big(W_i(u,t,i_1) \wedge W_i(u,t,i_2) \wedge i_1 \neq i_2 \to \bot\big),\\ &\forall i \forall p \forall v_1 \forall v_2 \big(W_d(i,p,v_1) \wedge W_d(i,p,v_2) \wedge v_1 \neq v_2 \to \bot\big)\}.\end{aligned}$$

Mapping constraints $\mathcal{M}$. The specification [20] distinguishes several types of DBpedia mappings summarized in Table 1 along with their figures in the English DBpedia.

[5] http://techwiki.umbel.org/index.php/UMBEL_Vocabulary

[6] http://live.dbpedia.org/sparql

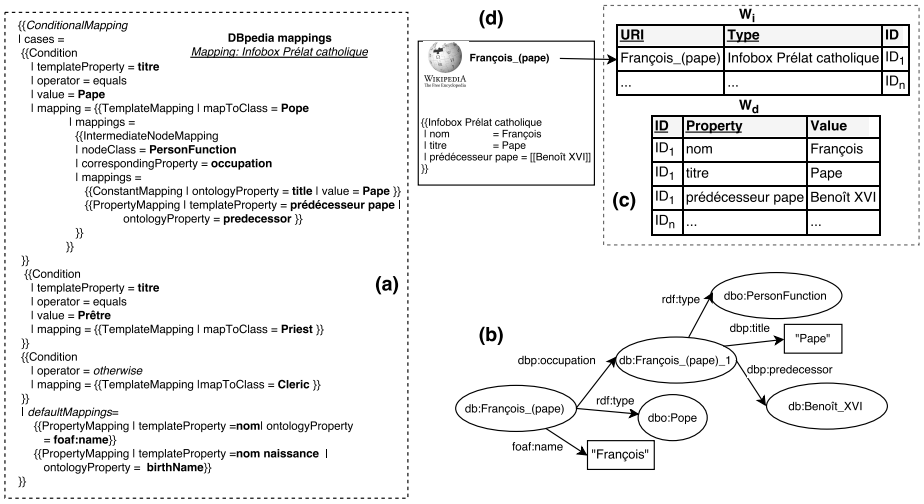


Fig. 1: (a) DBpedia mappings, (b) the RDF graph, and the Infobox as an instance of the schema $\mathbf{W}$ (c) and in the native format (d).

All these mappings can be represented as *nested tgds* [13, 23] extended with *negation and constraints* in the antecedents for capturing the conditional mappings and interpreted functions in the conclusions of implications, in the case of calculated mappings handling, e.g., dates or geo-coordinates. A crucial limitation of the mapping language (which we call *DBpedia tgds*) is the *impossibility of comparisons between infobox property values*. Infobox type $W_i.\mathsf{T}$ and property names $W_d.\mathsf{P}$ must be specified explicitly.

Example 1. Fig.1(a) shows a conditional mapping transferring the information about clerics from French wiki pages with an infobox *Prélat catholique* (d). Under these conditions, the except shown in Fig.1(c) as an instance over the schema $\mathbf{W}$ gives rise to the triples depicted in Fig.1(b). A tgd formalizing a French DBpedia mapping for clergy is given below:

$$
\begin{aligned}
&\forall U \forall I \big(W_i(U, \text{'fr:Prélat catholique'}, I) \rightarrow \\
&\quad \big(W_d(I, \text{'titre'}, \text{'Pape'}) \rightarrow \\
&\qquad \exists Y \big(\mathsf{Pope}(U) \wedge \mathsf{occupation}(U, Y) \wedge \mathsf{PersonFunction}(Y) \\
&\qquad\quad \wedge \mathsf{title}(Y, \text{'Pape'})) \quad \textit{// ``Intermediate node mapping''} \\
&\qquad\quad \wedge \ldots \\
&\qquad\quad \wedge \forall X (W_d(I, \text{'prédécesseur pape'}, X) \rightarrow \mathsf{predecessor}(Y, X)) \big) \\
&\quad \ldots \\
&\quad \wedge (W_d(I, \text{'titre'}, \text{'Prêtre'}) \rightarrow \mathsf{Priest}(U)) \\
&\quad \textit{// The ``otherwise'' branch:} \\
&\quad \wedge (\neg W_d(I, \text{'titre'}, \text{'Pape'}) \wedge \ldots \wedge \neg W_d(I, \text{'titre'}, \text{'Prêtre'}) \rightarrow \mathsf{Cleric}(U)) \\
&\quad \wedge \forall X (W_d(I, \text{'nom'}, X) \rightarrow \mathsf{foaf{:}name}(U, X)) \\
&\quad \ldots \\
&\quad \wedge \forall X (W_d(I, \text{'nom naissance'}, X) \rightarrow \mathsf{birthName}(U, X))) \big)
\end{aligned}
$$

The specification stipulates that conditions are evaluated in the natural order, and thus every next condition has to include the negation of all preceding conditions. In our case, this is only illustrated by the last, default ("otherwise") case, since the conditions are mutually exclusive. Note also that no universally quantified variable besides the page URI U and the technical infobox identifier I) — i.e., no X variable representing an infobox property — can occur more than once on the left-hand side of an implication, due to the lack of support for comparisons between infobox properties.

One further particularity of the chase with tgds is the handling of existentially quantified variables that represent so-called "intermediate nodes" (e.g., Y in Example 1). A usual approach is to instantiate such variables by null values, which could become blank nodes on the RDF storage side. The strategy currently followed by DBpedia is different: instead of blank nodes, the chase produces fresh IRIs. By appending an incremented number to the Wiki page address it avoids clashes with existing page URIs. We name it *constant inventing chase*.

Updates. We consider updates that can be specified on both the infobox and the DBpedia sides. Since DBpedia is a materialized extension constructed based on the contents of infoboxes, persistent modifications must be represented as infobox updates. We consider updates based on ground facts to be inserted or deleted, each update being limited to exactly one schema, the infobox $\mathbf{W}$ or DBpedia $\mathbf{T}$.

Definition 1. *Let* $\mathbf{S}$ *be a schema and* J *an instance of* $\mathbf{S}$. *An* update u *of* J *is a pair* (u^-, u^+) *of sets of ground atoms over* $\mathbf{S}$ *in which* u^+ *signifies facts to be inserted to* I *and* u^- *facts to be removed from* I. *Deletions are applied prior to insertions.*

Since WDF includes the mapping and TBox rules, special care is needed to make update effective and enforce or maintain the consistency of the affected WDF instance apply a minimal necessary modifications. Our formalization is close to the usual definition of formula based belief revision operators. A WDF instance I is identified with a conjunctive formula over $\mathbf{W}$ closed under the integrity constraints $\mathcal{W}$ of the infobox schema. The notation $u(I)$ is understood as $(I \setminus u^-) \cup u^+$ where $I \setminus u^-$ denotes the removal of all conjuncts occurring in u^- from I, and $I \cup u^+$ is the same as the conjunction $I \wedge u^+$.

We define a partial order $\preceq$ relation between updates as follows $u \preceq e$ iff $u^- \subseteq e^-$ and $u^+ \subseteq e^+$. One can as well consider other, e.g. cardinality based, partial orders.

Definition 2. *Let* $\boldsymbol{\mathcal{F}}$ *be a WDF* $(\mathbf{W}, \mathcal{M}, \mathcal{T})$, I *be an* $\boldsymbol{\mathcal{F}}$*-instance and let* u *be an update over* $\boldsymbol{\mathcal{F}}$. *The consistency-oblivious semantics* $\{\![u]\!\}$ *of* u *is the set of smallest (w.r.t.* $\preceq$*) updates* $[u]$ *over the infobox schema* $\mathbf{W}$ *such that the conditions* $[u](I) \cup \mathcal{W} \cup \mathcal{M} \cup \mathcal{T} \not\models u^-$, $[u](I) \cup \mathcal{W} \cup \mathcal{M} \cup \mathcal{T} \models u^+$ *and* $I \cup \mathcal{W} \not\models \bot$ *hold.*

The former two conditions ensure the effectiveness of the update, that is, that all desired insertions and deletions are performed. The conformance with $\mathbf{W}$ ensures that the update can be accommodated in the physical infobox storage model, which the constraints $\mathcal{W}$ simulate. The following definition of the semantics $\{\!\{\![u]\!\}\!\}$ restricts the semantics $\{\![u]\!\}$ in order to ensure that the DBpedia instance can be used under entailment w.r.t. $\mathcal{T}$, denoted as closure $cl(I, \mathcal{M})$. Note that both semantics $\{\![u]\!\}$, $\{\!\{\![u]\!\}\!\}$ depend on $\preceq$, $\boldsymbol{\mathcal{F}}$ and on I—which is not explicit in our notation for the sake of readability.

Definition 3. *Let $\mathcal{F}$ be a WDF $(\mathbf{W}, \mathcal{M}, \mathcal{T})$, I be an $\mathcal{F}$-instance and let u be an update over $\mathcal{F}$. The consistency-aware semantics $\{\!\![u]\!\!\}$ of u is the set of smallest (w.r.t. $\preceq$) updates $[\![u]\!]$ such that $[\![u]\!] \in \{\!\![u]\!\!\}$ and $[\![u]\!](I) \cup \mathcal{W} \cup \mathcal{M} \cup \mathcal{T} \not\models \bot$.*

3 Challenges of DBpedia OBDM

We consider the EXISTENCE OF SOLUTIONS problem and show that it is in general intractable even for the consistency-oblivious semantics.

Problem EXSOL-OBL. Parameter: WDF $\mathcal{F} = (\mathbf{W}, \mathcal{M}, \mathcal{T})$. Input: $\mathcal{F}$-instance I, update u. Test if $\{\!\![u]\!\!\} \neq \emptyset$.

Proposition 1. EXSOL-OBL *is NP-complete.*

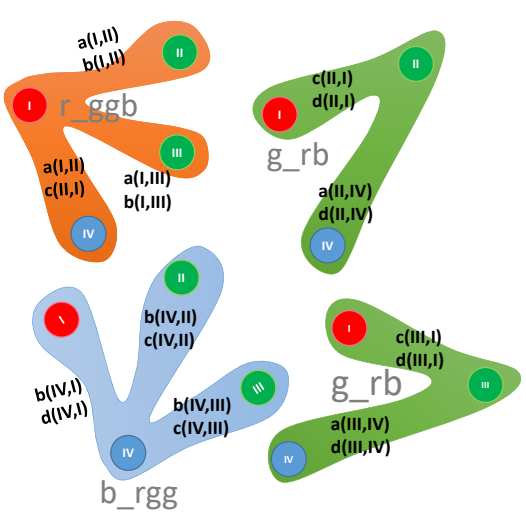
Proof (Sketch). Consider a DBpedia update u, and the WDF instance I. For the membership in NP, observe that enforcing the constraints in $\mathcal{M}$ and in $\mathcal{T}$ (e.g., via chase) terminates in polynomial time for every fixed WDF $\mathcal{F}$, which gives a bound on the size of the infobox instance witnessing $\{\!\![u]\!\!\} \neq \emptyset$ for an instance I. For each condition in the mapping $\mathcal{M}$ (limited to comparing a single infobox value with a fixed constant), we can define a canonical way of satisfying it, and thus defining *canonical witnesses*, whose size and active domain is determined by u, I and $\mathcal{F}$. As a result, the test comes down to guessing a canonical witness and checking it by the chase with constraints, that u^+ is inserted and u^- deleted, which is feasible in poly time for the constraints in DBP.

For the hardness, consider the following reduction from the 3-COLORABILITY problem. Let I be empty and let the set of atoms A that the DBpedia update $u = (\emptyset, A)$ inserts represents an undirected graph $G = (V, E)$ of degree at most 4 (for which 3COL is intractable [14]). A represents the vertices V as IRIs and each edge $(x, y) \in E$ for the IRIs x, y is represented by a collection of 8 atoms of the form $\mathsf{a}(x, y)$, $\mathsf{a}(y, x)$, $\mathsf{b}(x, y)$, $\mathsf{b}(y, x)$, $\mathsf{c}(x, y)$, $\mathsf{c}(y, x)$, $\mathsf{d}(x, y)$, and $\mathsf{d}(y, x)$, for which the assertions $\mathsf{a} = \mathsf{a}^{-1}$, $\mathsf{b} = \mathsf{b}^{-1}$, $\mathsf{c} = \mathsf{c}^{-1}$ and $\mathsf{d} = \mathsf{d}^{-1}$ are defined in $\mathcal{T}$.

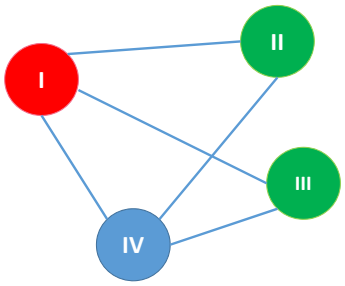
Each infobox encodes a single vertex of the graph, together with all its adjacent vertices (at most four direct neighbors). Together these 1-neighborhoods cover the graph. The encoding ensures that the only way to obtain the regular DBpedia representation of the graph, with exactly eight property assertions for each pair of vertices, is only possible if every vertex is assigned the same color in each infobox. This is achieved by distributing the a, b, c and d between each pair of adjacent nodes depending on the node color. The rules for that are given in Fig. 2(c). For instance, an edge between a red I and a green vertex II is composed from the properties **a**(I,II) **b**(I,II) whose creation is triggered by the infobox of page I, and the other two properties b, c are created by chasing the infobox I: **c**(II,I), **d**(II,I). Due to symmetry, this results in the eight property assertions.

The excerpt of the mapping for the neighborhood types 'r_ggb', 'b_rgg', 'g_rb' illustrated by a graph in Fig. 2 (b) is shown below.

$$\forall U \forall I \big(W_i(U, \text{'vertex'}, I) \rightarrow$$
$$\big(W_d(I, \text{'n-type'}, \text{'r\_ggb'}) \rightarrow \big(\mathsf{Node}(U) \wedge \forall X\, (W_d(I, \text{'n1'}, X) \rightarrow \mathsf{a}(U, X) \wedge \mathsf{b}(U, X))$$
$$\wedge \forall X\, (W_d(I, \text{'n2'}, X) \rightarrow \mathsf{a}(U, X) \wedge \mathsf{b}(U, X)) \wedge \forall X\, (W_d(I, \text{'n3'}, X) \rightarrow \mathsf{a}(U, X) \wedge \mathsf{c}(U, X))\big)$$



(a) The 1-neighborhoods of G



(b) The graph G

	red	*green*	*blue*
red		*ab*	*ac*
green	*cd*		*ad*
blue	*bd*	*bc*	

(c) Rules for TBox property generation

Fig. 2: Concepts of the proof of Proposition 1

$$\wedge\ (W_d(I, \text{'n-type'}, \text{'b\_rgg'}) \rightarrow (\mathsf{Node}(U) \wedge \forall X\ (W_d(I, \text{'n1'}, X) \rightarrow \mathsf{b}(U, X) \wedge \mathsf{d}(U, X))$$
$$\wedge \forall X\ (W_d(I, \text{'n2'}, X) \rightarrow \mathsf{b}(U, X) \wedge \mathsf{b}(U, X)) \wedge \forall X\ (W_d(I, \text{'n3'}, X) \rightarrow \mathsf{b}(U, X) \wedge \mathsf{c}(U, X)))$$
$$\wedge\ (W_d(I, \text{'n-type'}, \text{'g\_rb'}) \rightarrow (\mathsf{Node}(U) \wedge \forall X\ (W_d(I, \text{'n1'}, X) \rightarrow \mathsf{c}(U, X) \wedge \mathsf{d}(U, X))$$
$$\wedge \forall X\ (W_d(I, \text{'n2'}, X) \rightarrow \mathsf{a}(U, X) \wedge \mathsf{d}(U, X)))$$
$$\wedge \ldots \text{etc for other 1-neighborhood types} \ldots) \quad \square$$

If we bring the TBox and infobox schema constraints along with non-monotonicity of mapping rules into the picture, the potential challenges of accommodating updates start piling up quickly. An interplay of the following features of the framework can make update translation unwieldy: (i) inconsistencies due to the TBox assertions, namely the class and role disjointness and functional properties; (ii) many-to-many relationships between infobox and ontology properties defined by the mappings, and (iii) infobox schema constraints.

Example 2. **Deletions due to infobox constraints**. Consider the update u_1 inserting an alternative foaf:name value for an existing cleric (cf. the mapping in Example 1). The infobox key IP $\rightarrow$ V would deprecate this, since there is only one infobox property matching foaf:name. Therefore, all updates in $\{[u_1]\}$ will extend u_1 with the deletion of the old name.

Insertions and many-to-many property matches. Several Wiki properties are mapped to the same DBpedia property and the insertion cannot be uniquely resolved. E.g., infoboxes of football players in the English Wikipedia have the properties 'full name', 'name' and 'short name' all mapped to foaf:name.

Deletions with conditional mappings. Triples generated by a conditional mapping can be deleted either by removing the corresponding Wiki property or by modifying the Infobox property so that the condition is no longer satisfied. E.g., in Example 1, deleting

the triple predecessor $(\mathsf{Nicholas\_II}, \mathsf{Alexander\_II})$ can be done either by unsetting the infobox property 'prédécesseur pape' or the property 'titre' used in the condition.

The above considerations suggest that despite the syntactic restrictions of the DBpedia mappings, the problem of update translation is hard in the worst case. Furthermore, numerous translations of an ABox update often exist (exponentially many in the size of the mapping: e.g., each n-to-m property match increases the total size of possible translations by the factor of mn). Due to the interplay between the mapping conditions and TBox axioms a complete solution of the OBDM problem, presenting and explaining to the user *all possible ways* of accommodating an arbitrary update is not practical. Our pragmatic approach to the problem is described next.

4 Pragmatic DBpedia OBDM

Updates in the presence of constraints and mappings over a curated data source such as DBpedia are not likely to happen in a fully atomatic mode. Thus, rather than striving to define a set of formal principles to compare particular update implementations (akin, e.g. belief revision postulates) we focus on another aspect of update translation, especially important in collaborative and community-oriented settings, where adhering to standard practices and rules is crucial. Namely, we look for *most customary* ways to accommodate a change. For insertions, data evidence can be obtained from the actual data, whereas for deletions, additional logs are typically required. For all kinds of updates, we use a special kind of log, which we call *update resolution pattern*, recording the "shape" of each update command (e.g. *inserting a* birthPlace *DBpedia property of a* Pope *instance, where the Infobox property 'lieu de naissance' is alredy present. Delete the existing property and add the property 'lieu naissance' with the new value*).

To decide on the update pattern, when several alternatives are possible, we try to derive most customary ways of mapping objects of same classes from the existing data, rather than applying some principled belief revision semantics. E.g., when updating the birth place, we look at the usage statistics of the Infobox properties 'lieu naissance' and 'lieu de naissance' and choose the one used most often. If most infoboxes have both, we will not delete the already existing property but just add a second one. This way, we might resolve a DBpedia's foaf:name as two infobox properties (e.g., 'name' and 'full name') at once if most existing records of a given type follow this pattern, even if it would contradict the minimal change principle which typically governs belief revision.

A translation procedure we discuss next proceeds essentially on the best effort basis, exploring the most likely update accommodations and facilitating reuse of standard practices through update resolution patterns. It takes a SPARQL update and transforms it into a set of Infobox updates for the user to apply and save as an update reolution pattern. The source code of our system is openly available[7].

4.1 Update translation steps

From the very beginning, we turn our SPARQL update into a set of ground atoms, which are then grouped by subject (corresponding to the Wikipedia page). The idea of

[7] https://github.com/aahmeti/DBpedia-SUE, a screencast is available at https://goo.gl/BQhDYf

our update translation procedure is to create or to re-use existing update patterns for each grouped update extracted from the user input. A user update request related to a particular Wikipedia page (DBpedia entries grouped by a commond subject) becomes a core pattern, which gives rise to a number of possible translations as a wiki insert.

For each translation, the mapping and the TBox constraints are applied, in order to see which further atoms have to be added and if there are inconsistencies with the pre-existing facts. All such inconsistencies are removed, resulting in a further update, giving rise to an update resolution pattern nested within the root one, and the translation process procedes recursively.

Pruning is essential in this process, since resolution patterns can sprout actively (e.g., some DBpedia properties are mapped to tens of Wikipedia ones). Potentially non-terminating, with the current DBpedia mappings inconsistencies can typically be resolved within the scope of one or two subjects (Wikipedia pages), and thus pattern trees resulting from this process are not deep. The reason is that functionality is currently only used for data properties, and only very few properties are declared disjoint.

4.2 Update Resolution Policies

Given the large number of possible translations of an update, potentially resulting in different clash patterns, an update can be translated in various ways, from which the user must select one. The crucial issue here is that the number of choices can be too large even for a very simple update, and that updates can cause side effects outlined in the previous section.

Here, we consider *update resolution policies* aimed at reducing the number of options for the user in the specific case of *n*-to-1 alternatives to insert. We currently consider two different alternatives in accordance with some concise principles, namely *infobox-frequency-first* and *similar-subject-first*.

We exemplify the application of such policies looking at the ambiguities in the top 10 most used Infoboxes[8]. In particular, we find and inspect the ambiguities in 'Settlement', 'Taxobox', 'Person', 'Football biography' and 'Film'. For the sake of clarity, we show a selection of the most representative ambiguities in Table 2, while other ambiguities in the infoboxes follow the same patterns. For instance, all 'name', 'fullname' and 'player name' in a 'football biography' infobox map to a foaf:name property. Table 2 also reports the number of subjects (i.e., wikipedia pages) of each infobox type, converted from the English Wikipedia.

Infobox-frequency-first. This policy considers that, for an insertion in a subject with an infobox W, resulting in a *n*-to-1 alternatives, we infer that the most likely accommodation would be the most frequent property in all the subjects with such infobox W, among all the alternatives not fulfilled in the subject we are currently updating. Statistics on frequent properties can be computed seamless, concurrently to the DBpedia conversion. Overall, this approximation could help users to inspect frequent properties for the update, so that rare or infrequent properties can be quickly discarded. In contrast, the approach may fail to guess the concrete purpose or real users, who may choose to accommodate different alternatives.

[8] http://mappings.dbpedia.org/server/statistics/en/.

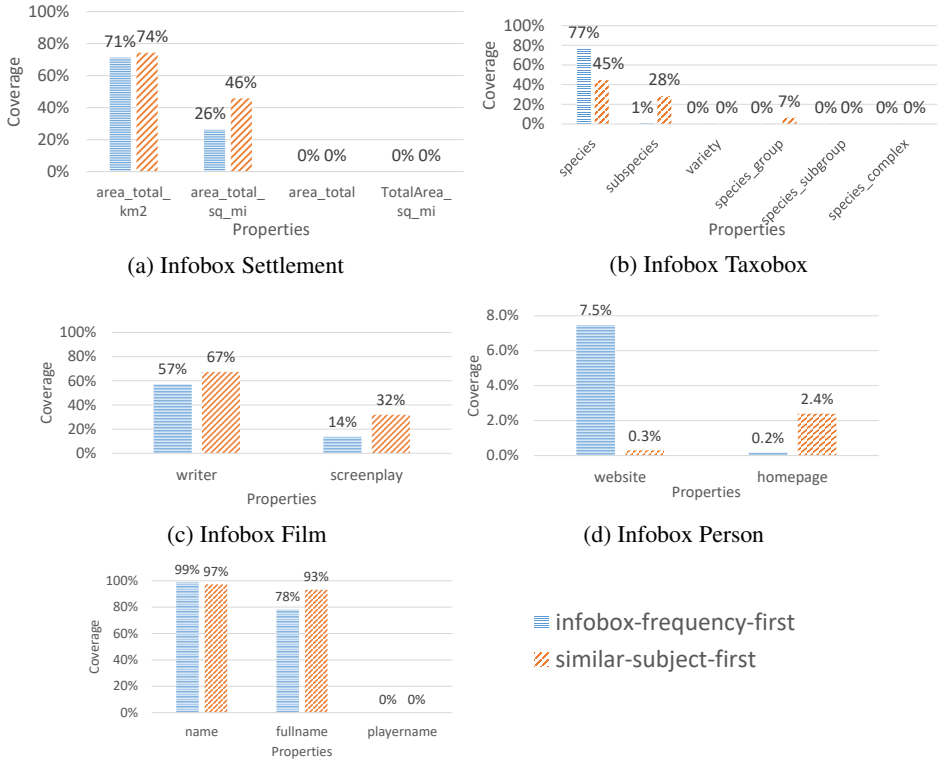
INFOBOX	SUBJECTS	AMBIGUOUS n-1 MAPPING	
		WIKIPEDIA PROP.	DBPEDIA PROP.
Settlement	369,024	*area_total_km2* *area_total_sq_mi* *area_total* *TotalArea_sq_mi*	dbp:areaTotal
Taxobox	293,715	*species* *subspecies* *variety* *species_group* *species_subgroup* *species_complex*	dbp:species
Person	168,372	*website* *homepage*	foaf:homepage
Football biography	128,602	*name* *fullname* *playername*	foaf:name
Film	106,254	*screenplay* *writer*	dbp:writer

Table 2: Examples of n-to-1 alternatives in DBpedia (English) mappings.

Figure 3 evaluates the distribution of frequencies of the Wikipedia properties involved in n-1 mappings from Table 2, considering all the subjects in the infobox (series *Infobox-frequency-first*). Results show that the application of this policy can certainly filter out infrequent property candidates, but it may require further elaboration for a more informed recommendation, specially in those cases in which the property is not extensively used in the infoboxes. For instance, all properties with no or marginal presence can be discarded, such as 'area_total' and 'TotalArea_sq_mi' in 'Settlement' (Figure 3 (a)), 'variety', 'species_group', 'species_subgroup' and 'species_complex' in 'Taxobox' (Figure 3 (b)), 'homepage' in 'Person' and 'playername' in 'Football biography' (Figure 3 (d)). In turn, some properties are much more represented than others, and shall be the first ranked suggestion when inserting an ambiguous mapping. This is the case of most of the infoboxes, such as the frequent 'area_total_km2' property in 'Settlement', 'species' in 'Taxobox', 'website' in 'Person', and 'writer' in 'Film'. In contrast, only one case, 'Football biography', showed two properties that are almost equally distributed, with 'name' slightly more used than 'fullname'.

Similar-subject-first. The objective of this strategy is to refine the previous *Infobox-frequency-first* policy by delimiting a set of similar subjects for which the frequent properties are inspected. The reason of this strategy is that most of the properties in infoboxes are optional, so that different Wikipedia resources can, and often are, described with different levels of detail. Thus, finding "similar" subjects could effectively recommend more frequent patterns. For finding similar entities, we focus for the moment on a simple approach on sampling m subjects described with the same target infobox W and described with the same DBpedia property as the update u.

Figure 3 evaluates the distribution of property frequencies in such scenario (series *Similar-subject-first*), sampling $m = 1,000$ subjects of each infobox described the DBpedia property to be inserted (dbp:areaTotal, dbp:species, foaf:homepage, foaf:name or dbp:writer respectively). Results show that this policy allows the system to perform more informed decisions. For instance, in the 'Person' use case (Figure 3 (c)), the 'homepage' property cannot be discarded (as suggested by the *Infobox-frequency-first* ap-



(a) Infobox Settlement (b) Infobox Taxobox

(c) Infobox Film (d) Infobox Person

(e) Infobox Football bio. (f) Legend for the charts (a)–(e).

Fig. 3: Statistics obtained by *infobox-frequency-first* and *similar-subject-first* policies on four different infoboxes.

proach), given that a particular type of persons are more frequently associated with homepage instead of websites (e.g., those who are not related to a company). Similarly, in 'Taxobox' (Figure 3 (b)), some particular species also include 'subspecies' and 'subsepecies_group', hence they should be included and ranked as potential accommodations for the user query.

5 Related work

The problem of knowledge base update and belief revision has been extensively studied in the literature for various description logics, cf. e.g. [9, 11, 6, 22, 17]. Both semantics strongly enforcing the new knowledge to be accepted and those deliberating between accepting and discarding the change have been studied [18]. Particularly, belief revision with user interaction has been considered, e.g., in [27]. In the same spirit, another recent study [33] considers repairs with user interaction. In both cases, an informed choice between alternatives is difficult as it requires understanding of such complex

update semantics. Our ultimate aim is not to compare our approach with such belief revision operators, but rather using/developing statistics (pre-existing data) and patterns (pre-existing interactions) as a means for helping users in making a meaningful choice, complementing work on belief revision with practical guidelines.

The majority of existing OBDM approaches (e.g., [29, 24, 30]) consider the problem of query answering only rather than updates, using different fragments of OWL. The emphasis in those approaches is in algorithms for query rewriting considering one-to-many, many-to-many mappings, where queries consist of also variables (without an instantiation step as in our case).

As for updates and tgds, the approach [21] addresses a quite different setting of a peer data network in which data and updates are propagated via tgds. The peers in the network do not impose additional schema constraints (like the DBpedia TBox), features like class disjointness are not part of the setting, the focus is on combining the external data with local updates in a peer network.

We mentioned work reporting about inconsistencies in DBpedia in the introduction already [7, 12]. In another work about detecting inconsistencies within DBpedia [28] have considered mappings to the DOLCE upper ontology to detect even more inconsistencies, operating in a more complex ontology language using a full OWL DL reasoner (HermiT). Their approach is orthogonal in the sense that they focus on detecting and resolving systematic errors in the DBpedia ontology itself, rather than automatically fixing the assertions, leave alone the data in Wikipedia itself. Nonetheless, it would be an interesting future direction to combine these two orthogonal approaches.

It is also worth mentioning work in the domain of applying statistical methods for disambiguating updates, e.g., [32], namely for enriching the TBox based on the data, which is actually not our scope, as we do not modify the TBox here.

Recently Wikipedia partially shifted to another, structured datasource than infoboxes, namely, Wikidata. We note that the model of Wikidata is different to DBpedia; different possible representations in plain RDF or Relational models have been recently suggested/discussed [19]. Our approach could potentially help in bridging between the two, which we leave to future work.

6 Conclusion

Little attention has been paid to the benefits that the semantic infrastructure can bring to maintain the wiki content, for instance to ensure that the effects of a wiki edit are consistent across infoboxes. In this paper, we present first insights to allow ontology-based updates of wiki content.

Various worst-case scenarios of update translation, especially those exhibiting the intractability of update handling, can be hardly realized in the current DBpedia version (mappings and ontology). From the practical point of view, the following aspects of OBDM appear crucial for the DBpedia case. Firstly, it is the *inherent ambiguity of update translation*; mappings often create a many-to-one or many-to-many relationships between infobox and DBpedia properties. Second, concisely presenting a large number of options to the user is a challenge, hence an *automatic selection of most likely update translations* is likely required. Finally, being a curated system, Wiki also requires

curated updates. Thus, *splitting a SPARQL update into small independent pieces* to be verified by the Wiki maintainers is needed as well. Note that human intervention is often unavoidable, since calculating mappings involve non-invertible functions.

The main distinguishing characteristic of our approach is the DBpedia OBDM setting, and the focus on update accommodation strategies which are simple, comprehensible for the user and can draw from pre-existing meta-knowledge, such as already existing mapping patterns resp. usage frequencies of certain infobox fields, to decide update ambiguities upon similar, prototypical objects in the underlying data, estimating probabilities of alternative update translations. Our goal in this work was twofold: on the one hand, to understand and to formalize the DBpedia setting from the OBDM perspective, and on the other hand, to explore more pragmatic approaches to OBDM. To the best of our knowledge, it is the first attempt to study DBpedia mappings from the formal point of view. We found out that although the worst-case complexity of OBDM can be prohibitively high (even with low expressivity ontology and mapping languages), the real data, mappings and ontology found in DBpedia do not necessarily hit this full potential complexity; indeed, we conclude that the study and development of best-effort pragmatic approaches — some of which we have explored — is worthwhile.

Our early practical experiments with a DBpedia-based OBDM prototype shows that high worst case complexity of update translation can have little to do with actual challenges of OBDM for curated data. Rather, simple and comprehensible update resolution policies, reliable methods of confidence estimation and the ability to automatically learn and use best practices should be considered.

Acknowledgements The work of Fernández was supported by the Austrian Science Fund (FWF): M1720-G11. Ahmeti was supported by the Vienna Science and Technology Fund (WWTF), project SEE: ICT12-15. Savenkov was supported by the Austrian Research Promotion Agency (FFG), project 855407.

References

1. S. Abiteboul, O. M. Duschka. Complexity of answering queries using materialized views. In *Proc. PODS '98*, pages 254–263, ACM, 1998.
2. A. Ahmeti, D. Calvanese, A. Polleres. Updating RDFS aboxes and tboxes in SPARQL. In *Proc. ISWC'14*, p. 441–456, Springer, 2014.
3. A. Ahmeti, D. Calvanese, A. Polleres, V. Savenkov. Handling inconsistencies due to class disjointness in SPARQL updates. In *Proc. ESWC'16*, p. 387–404, Springer, 2016.
4. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives. Dbpedia: A nucleus for a web of open data. In *Proc. ISWC'07*, Springer, 2007.
5. F. Bancilhon and N. Spyratos. Update semantics of relational views. *ACM Trans. Database Syst.*, 6(4):557–575, Dec. 1981.
6. S. Benferhat, Z. Bouraoui, O. Papini, E. Würbel. A prioritized assertional-based revision for *DL-Lite* knowledge bases. In *Proc. JELIA'14*, p. 442–456, Springer, 2014.
7. S. Bischof, M. Krötzsch, A. Polleres, S. Rudolph. Schema-agnostic query rewriting in SPARQL 1.1. In *Proc. ISWC'14)*, Springer, 2014.
8. D. Brickley, R. Guha (eds.). RDF Vocabulary Description Language 1.0: RDF Schema. W3C Rec., 2004.

9. D. Calvanese, E. Kharlamov, W. Nutt, D. Zheleznyakov. Evolution of *DL-Lite* knowledge bases. In *Proc. ISWC'10)*, p. 112–128, Springer, 2010.
10. G. Cong, W. Fan, F. Geerts, J. Li, J. Luo. On the complexity of view update analysis and its application to annotation propagation. *IEEE TKDE* 24(3):506–519, March 2012.
11. G. De Giacomo, M. Lenzerini, A. Poggi, R. Rosati. On instance-level update and erasure in description logic ontologies. *J. Log. Comput.* 19(5):745–770, 2009.
12. A. Dimou, D. Kontokostas, M. Freudenberg, R. Verborgh, J. Lehmann, E. Mannens, S. Hellmann, R. V. de Walle. Assessing and refining mappings to RDF to improve dataset quality. In *Proc. ISWC'15, Part II*, p. 133–149. Springer, 2015.
13. A. Fuxman, M. A. Hernández, C. T. H. Ho, R. J. Miller, P. Papotti, L. Popa. Nested mappings: Schema mapping reloaded. In *Proc. VLDB*, p. 67–78, 2006.
14. M. Garey, D. Johnson, L. Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science* 1(3):237 – 267, 1976.
15. P. Gearon, A. Passant, A. Polleres. SPARQL 1.1 update. W3C Rec., 2013.
16. B. Glimm, C. Ogbuji. SPARQL 1.1 entailment regimes. W3C Rec., 2013.
17. B. C. Grau, E. Jiménez-Ruiz, E. Kharlamov, D. Zheleznyakov. Ontology evolution under semantic constraints. In *Proc.(KR 2012)*, p. 137-147, AAAI Press, 2012.
18. S.O. Hansson. A survey of non-prioritized belief revision. *Erkenntnis* 50(2-3):413-427, 1999.
19. D. Hernández, A. Hogan, C. Riveros, C. Rojas, E. Zerega. Querying Wikidata: Comparing SPARQL, Relational and Graph Databases. Proc. ISWC'16 part 2, p. 88-103, Springer 2016.
20. A. Jentzsch, C. Sahnwaldt, R. Isele, C. Bizer. Dbpedia mapping language. Tech. report, http://mappings.dbpedia.org, 2010.
21. G. Karvounarakis, T. J. Green, Z. G. Ives, V. Tannen. Collaborative data sharing via update exchange and provenance. *ACM ToDS* 38(3):19:1-19:42, Sept. 2013.
22. E. Kharlamov, D. Zheleznyakov, D. Calvanese. Capturing model-based ontology evolution at the instance level: The case of *DL-Lite*. *J. Comput. Syst. Sci.* 79(6):835-872, Sept. 2013.
23. P. G. Kolaitis, R. Pichler, E. Sallinger, V. Savenkov. Nested dependencies: structure and reasoning. In *Proc, PODS'14*, p. 176-187, 2014.
24. R. Kontchakov, M. Rezk, M. Rodríguez-Muro, G. Xiao, M. Zakharyaschev. Answering SPARQL queries over databases under OWL 2 QL entailment regime. In *Proc. ISWC'14 - Part I*, ISWC '14, p. 552-567, Springer, 2014.
25. J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *SWJ* 6(2):167-195, 2015.
26. M. Lenzerini. Ontology-based data management. In *Proc. of the 20th ACM Int'l Conf. on Information and Knowledge Management*, CIKM '11, p. 5-6. ACM, 2011.
27. N. Nikitina, S. Rudolph, B. Glimm. Interactive ontology revision. *JWS* 12–13:118-130, 2012.
28. H. Paulheim, A. Gangemi. Serving DBpedia with DOLCE–more than just adding a cherry on top. In *Proc. ISWC'15*, p. 180-196, Springer, 2015.
29. F. Priyatna, O. Corcho, J. Sequeda. Formalisation and experiences of r2rml-based sparql to sql query translation using morph. In *Proc. WWW '14*, p. 479-490. ACM, 2014.
30. M. Rodríguez-Muro, M. Rezk. Efficient SPARQL-to-SQL with R2RML mappings. *JWS* 33(1), 2015.
31. F. M. Suchanek, G. Kasneci, G. Weikum. Yago: A core of semantic knowledge. In *Proc. WWW '07*, p. 697-706. ACM, 2007.
32. G. Töpper, M. Knuth, H. Sack. Dbpedia ontology enrichment for inconsistency detection. In *Proc. I-SEMANTICS '12*, p. 33-40. ACM, 2012.
33. M. Bienvenu, C. Bourgaux, F. Goasdoué. Query-Driven Repairing of Inconsistent DL-Lite Knowledge Bases. In *Proc. IJCAI '16*, p. 957–964. IJCAI/AAAI Press, 2016.