Simple Reasoning for Contextualized RDF Knowledge[1]

Mathew Joseph [a,b] and Luciano Serafini [b]
[a] *FBK-IRST, Via Sommarive 18, 38050 Trento, Italy*
[b] *DISI, University of Trento, Italy*

Abstract. The standard semantic web languages and reasoning tools do not explicitly take into account the contextual dimension of knowledge, i.e., the fact that a certain statement (RDF triple) is not universally true, but true only in certain circumstances. Such contextual information, which includes for instance, the time interval, the spatial region, or the sub-domain in which a certain statement holds, are of foremost importance to determine correct answers for a user query. Rather than proposing a new standard, in this work, we introduce a framework for contextual knowledge representation and reasoning based on current RDF(S) standards, and we provide a sound and complete set of forward inference rules that support reasoning in and across contexts. The approach proposed in this paper has been implemented in a prototype, which will be briefly described.

1. Introduction

Although, RDF Knowledge that is available on the web from multiple distributed sources and servers, pertains to multiple heterogeneous domains, topics and time frames, the current search engines or RDF stores do not take into account this "contextual information" of the RDF data. Moreover, these systems answer the user queries forming a union/merge of all the knowledge they have. This approach first of all leads to "inefficiency" because of having to search the whole knowledge store for every query. Another more fatal consequence that these systems ignore is the "inconsistencies" that result from the merge of knowledge that pertain to different/inconsistent domains. Where as, a Contextualized approach to knowledge representation brings *modularity* to the approach by segregating the relevant sources of knowledge for a query from irrelevant ones by the knowledge of contextual information of the query. Inconsistent sources of knowledge can be separated among different contexts and hence be reasoned separately.

One of the significant approaches that shed light in the direction of context based modeling is that of the one in [1]. The authors provide a context based semantics and a framework based on OWL2 and show by a sample use case, how a domain such as football world cup, can be modeled using some of the constructs like context classes, qualified predicates etc. Although OWL2 has become popular recently, some of the hin-

[1] This work has been partially supported by the LiveMemories project (Active Digital Memories of Collective Life). Many thanks to Andrei Tamilin for all the support recieved in implementation of the CKR system and Martin Homola, Francesco Corcoglionitti for all valuable suggestions.

drances for a system based on OWL to be implementable are (i) decidability of conjunctive query answering is unknown [2] (ii) efficient forward chained repositories cannot be built because of a proliferation of inferred statements. The proliferation effect in the latter point is caused due to its "if and only if" semantics, RDFS is the most popular semantic web language and it is based on the paradigm called "if" semantics [3]. Also, most of the knowledge currently available in the semantic web is in much weaker languages such as RDF, RDF(S) and OWL-Horst, which support efficient forward chaining reasoning and query answering. The applicability of the approach proposed in [1] should therefore pass through its adaptation to weaker languages.

In this paper, we following the basic principles in [1], adapt their framework based on OWL to RDFS, provide an RDFS based semantics suitable for contexts. Moreover, we provide sound/complete set of forward inference rules for computing logical consequence and provide a proof of completeness for the same. In developing our approach, we take into account its concrete implementability in state of the art RDF(S) knowledge stores, where contexts can be represented by named graphs, and contextual information as statements about graph names. Furthermore, contexts can be organized in broader/narrower hierarchical relation which supports the propagation of information across different contexts. The main impact of this work is that, using our framework one could create a semantic repository in which RDF statements can be contextualized with attributes such as time, topic, and location, and (s)he can query both on a single context or on a set of contexts. One could also state rules that propagate statements around different contexts. To check the feasibility of our theory we have implemented a prototype which will be shortly described in the final section. With respect of previous works, the novel contribution of this paper is on the axiomatization of Contextualized knowledge repository on the bases of RDF(S) semantics and using the limited language of RDF, and the proof of soundness and completeness of such an axiomatization.

2. RDF(S) Background

In this section we briefly introduce RDF(S) constructs, but we assume familiarity with RDFS concepts and semantics as described in [4,5]. An RDF vocabulary $\mathbf{V}$ is composed of three mutually disjoint sets $\mathbf{U}, \mathbf{B}, \mathbf{L}$ representing respectively URI references, blank nodes, and literals. An *RDF triple* is any tuple $(s, p, o) \in (\mathbf{U} \cup \mathbf{B}) \times (\mathbf{U}) \times (\mathbf{U} \cup \mathbf{B} \cup \mathbf{L})$. Any element of the set $\mathbf{U} \cup \mathbf{L}$ is called a *name*.

Definition 1 (RDF graph) *An RDF graph is a set of RDF triples. The* universe*, $U(G)$, of a graph G is the set $\{s, p, o | (s, p, o) \in G\}$. The* names*, $name(G)$, of a graph G is the set $U(G) \setminus \mathbf{B}$. If $S = \{G_i\}_{i \in I}$ is a family of rdf graphs on the set of indices I, we define $U(S) = \bigcup_{i \in I} U(G_i)$ and $name(S) = \bigcup_{i \in I} name(G_i)$. A* ground graph *is an rdf graph without any blank node occurrences, i.e., a graph G such that $U(G) = name(G)$.*

Definition 2 (Simple interpretation) *A* simple interpretation*, of the vocabulary $\mathbf{V} = (\mathbf{U}, \mathbf{B}, \mathbf{L})$, is a tuple*

$$\mathcal{I} = \langle IR, IP, IC, IEXT, ICEXT, LV, IS \rangle$$

where (1) IR is a nonempty set of objects, called the domain or universe of $\mathcal{I}$; (2) $IP \subseteq IR$, is a set of objects denoting properties; (3) $IC \subseteq IR$, is a distinguished subset of IR

denoting classes (4) IEXT : IP $\rightarrow 2^{IR \times IR}$, a mapping that assigns an extension to each property name; (5) ICEXT : $IC \rightarrow 2^{IR}$, a mapping that assigns a set of objects to every object denoting a class; (6) LV $\subseteq$ IR, the set of literal values, LV contains all plain literals; (7) IS : $\mathbf{U} \cup \mathbf{L} \rightarrow$ IR $\cup$ IP, the interpretation mapping, a mapping that assigns an object in IR to each element in $\mathbf{U} \cup \mathbf{L}$, such that IS is the identity function for plain literals and only assigns objects in LV to lexically valid literals in $\mathbf{L}$.

The class of RDF (resp. RDFS) interpretations is a subclass of the class of simple interpretations that satisfy additional constraints on the interpretation of the RDF (resp. RDFS) primitives. For instance an RDF interpretation should satisfy the condition:

$$x \in \text{IP} \iff \langle x, \text{IS}(\texttt{rdf:Property}) \rangle \in \text{IEXT}(\text{IP}(\texttt{rdf:type}))$$

For lack of space, we don't list all the conditions associated to RDF (resp. RDFS) interpretations and we refer the reader to [5]. Entailment between RDF(S) graphs can be defined in the usual model theoretic way

Definition 3 *A simple (resp. RDF and RDFS) interpretation is a model of a graph G, in symbols $\mathcal{I} \models_{simple} G$ (resp. $\mathcal{I} \models_{rdf} G$ $\mathcal{I} \models_{rdfs} G$) if there is an assignment A to the blank nodes of G, such that for every triple $(s, p, o) \in G$, we have that $\langle IS+A(s), IS+A(o) \rangle \in IEXT(IS(p))$, where $IS + A(x)$ returns $IS(x)$ if $x \in \mathbf{U}$ and $A(x)$ if $x \in \mathbf{B}$.*

Definition 4 *A graph G simply (resp. RDF, RDFS) entails a graph H, in symbols $G \models_{simple} H$ (resp, $G \models_{rdf} H$, $G \models_{rdfs} H$) if for every simple (resp, RDF, RDFS) interpretation $\mathcal{I}$ that satisfies G, $\mathcal{I}$ also satisfies H.*

The provability relation is denoted by $\vdash_{rdfs}, \vdash_{rdf}, \vdash_{simple}$. An RDFS *class* is denoted by symbols C, D etc; a *property* by R, S. X can be either a property or a class; *individuals* (denoted by symbols a, b etc.) are symbols that are instances of properties and classes.

3. Contextualized Knowledge Bases: A Formal Model

In a contextualized knowledge base (CKB), each statement is qualified with a set of contextual parameters. Statements qualified with the same parameters are grouped as a unique graph which constitute a context. We call such contextual parameters as *dimensions*.

Definition 5 (Contextual dimension) *A contextual dimension $\mathcal{D}_i$, is a ground RDFS Graph defined over a set of names $ValSet(\mathcal{D}_i) \subseteq U(\mathcal{D}_i)$ and $U(\mathcal{D}_i)$ contains a property $\prec_i$ called cover, defined over $ValSet(\mathcal{D}_i)$.*

Intuitively the relation $cover$ provides a structure among the values of $ValSet(\mathcal{D}_i)$. An example of a contextual dimension is the Location dimension, whose $\mathcal{D}_{\text{location}}$ contains values of geographic locations, for instance milan, rome, italy etc. The relation $\prec_{\text{Location}}$ represent the coverage between geographical regions, for instance milan $\prec_{\text{Location}}$ italy, rome $\prec_{\text{Location}}$ italy. Other examples of dimensions are Time, Topic whose $\prec_{\text{time}}$ represents temporal interval containment (e.g., jan-2011 $\prec_{\text{Time}}$ 2011) and the relation $\prec_{\text{topic}}$ represents broader/narrower relation between domains

(e.g., biology $\prec_{\text{Topic}}$ science). In the literature of the semantic web there are approaches that propose examples of dimension to qualify statements to encode for instance information about provenance, ownership and trust [6], or about time [7] and space [8]. In the literature of knowledge representation [9] proposes a set of 12 dimensions to model context dependent knowledge. The formal framework developed is general and admits an arbitrary, but fixed number $n \geq 0$ of contextual dimensions. A zero dimensional CKB can be thought of as a single RDFS knowledge repository without contextual information.

Definition 6 (Context) *A* context*, defined over the n dimensions $\{\mathcal{D}_i\}_{1 \leq i \leq n}$, is a triple $\langle \mathcal{C}, \mathbf{d}(\mathcal{C}), \mathrm{Graph}(\mathcal{C}) \rangle$, where*

1. *$\mathcal{C}$ is the* context identifier*;*
2. *$\mathbf{d}(\mathcal{C}) = \langle d_1, \ldots, d_n \rangle$ where each $d_i \in ValSet(\mathcal{D}_i), i = 1, ..., n$;*
3. *$\mathrm{Graph}(\mathcal{C})$ is an RDF graph*

Example 1 *The following are two sample contexts in the 2-dimensional CKB in the dimension of* Time *and* Topic*.*

2010, uefa_champions_legue
Winner(inter)
Defeats(inter, bayern_munich)
Best-player(diego_milito)
Happy(maurinho)

2010, french_open_tennis
Winner(rafael_nadal)
Defeats(rafael_nadal, robin_soderling)
...
...

It can be noted from the figure that Time and Topic are the two dimensions. For the first context, the values taken by the context for these dimensions are respectively 2010 and uefa_champions_legue, where as in the second context values taken by these are respectively 2010 and french_open_tennis. Now consider the statement Winner(inter) which is true in the first context. We can denote this fact by $\langle$2010, uefa_champions_legue$\rangle$: Winner(inter), the same statement is false in a context about $\langle$2010, french_open_tennis$\rangle$ where as the statement Winner(nadal) is true. We occasionally use the symbol $\mathcal{C}$ to refer to a whole context whose context identifier is $\mathcal{C}$. An important component of a CKB is a knowledge base called *meta knowledge* where knowledge about dimensions and how their values are related to context identifiers are defined. We define *dimension space*, $\mathcal{D}$, as the set $\{ValSet(\mathcal{D}_1) \times ... \times ValSet(\mathcal{D}_n)\}$

Definition 7 (Meta Knowledge) *A* Meta Knowledge $\mathfrak{M}$*, is an RDF graph composed of:*

1. *n rdf graphs $\{\mathcal{D}_i\}_{1 \leq i \leq n}$: the context dimensions;*
2. *a set $\mathfrak{C}$ of URIs: the context identifiers;*
3. *for each $\mathcal{C} \in \mathfrak{C}$, and each $1 \leq i \leq n$, the assertion $\mathcal{D}_i$-Of$(\mathcal{C}, d_i)$, for some $d_i \in ValSet(\mathcal{D}_i)$.*

Intuitively, $\mathcal{D}_i$-Of(.) is a function to associate a context $\mathcal{C}$ to a value in $\mathcal{D}_i$. For any two contexts $\mathcal{C}_1$ and $\mathcal{C}_2$, we use the notation $\mathcal{C}_1 \prec C_2$ if $\mathcal{D}_i$-Of$(\mathcal{C}_1) \prec_i \mathcal{D}_i$-Of$(\mathcal{C}_2)$ for $i = 1, ..., n$. We also assume in the above definition that $ValSet(\mathcal{D}_i)$s are pairwise disjoint.

One of the most important feature of contexts is that the same symbol may have different extensions in different contexts. For example, the class Winner in the context of 2010 UEFA champions legue denotes a different set w.r.t., the literally equal class in

the context of 2010 French open tennis competition. This assumption however requires to have the possibility to distinguish the two classes when for instance one wants to integrate information contained in two or more contexts in a broader context. For this reason we introduce the mechanism of *qualification* of a symbol w.r.t., a (set of) contextual dimensions. For instance, in a broader context about sport, the qualified symbols $\mathsf{Winner}_{\mathsf{2010,uefa\_champions\_legue}}$ and $\mathsf{Winner}_{\mathsf{2010,french\_open\_tennis}}$ are used to denote Winner in the two different contexts described above. Apart from resolving inconsistencies during data integration, qualification can also be used in circumstances like:

External Referencing To talk about classes and properties in a context from another context. For instance, one could use statements like $\langle \mathsf{2010}, \mathsf{sports} \rangle : \mathsf{Winner}_{\mathsf{2010,uefa\_champions\_legue}}(\mathsf{inter})$ to state that the fact $\mathsf{Winner}(\mathsf{inter})$ in the context of $\langle \mathsf{2010}, \mathsf{uefa\_champions\_legue} \rangle$ from a context of $\langle \mathsf{2010}, \mathsf{sports} \rangle$.

Lifting We can use this construct to define data transfer rules across contexts, for instance consider the two rules below stated in the context below:

$$\langle \mathsf{2010}, \mathsf{sports} \rangle : \begin{array}{l} \mathsf{Winner}_{\mathsf{2010,uefa\_champions\_legue}} \ \texttt{rdfs:subClassOf}\ \mathsf{Winner} \\ \mathsf{Winner}_{\mathsf{2010,french\_open\_tennis}} \ \texttt{rdfs:subClassOf}\ \mathsf{Winner} \end{array}$$

Such rules can effectuate the movement of instances from the source class Winner, respectively from source contexts, $\langle \mathsf{2010}, \mathsf{uefa\_champions\_legue} \rangle$ and $\langle \mathsf{2010}, \mathsf{french\_open\_tennis} \rangle$ to destination class, Winner in destination context, $\langle \mathsf{2010}, \mathsf{sports} \rangle$

A qualified class (property) in syntax is of the form $C_{\mathbf{d}}$ ($R_{\mathbf{d}}$), where $\mathbf{d} = \langle d_1, \ldots, d_n \rangle$ is an n-dimensional vector representing a context. We call C (R), the *base class (base property)* of the qualified class $C_{\mathbf{d}}$ (qualified property $R_{\mathbf{d}}$). A description about how qualified class and properties are encoded using URIs is given in the section 6. We now formally define a system of multiple contexts which we call a *contextualized knowledge repository*.

Definition 8 (Contextualized Knowledge Repository) *A* Contextualized Knowledge Repository (CKR) *is a pair* $\mathfrak{K} = \langle \mathfrak{M}, \mathcal{G} \rangle$ *where*

1. $\mathfrak{M}$ *is a meta knowledge on the set of contexts* $\mathfrak{C}$
2. $\mathcal{G} = \{G_{\mathcal{C}}\}_{\mathcal{C} \in \mathfrak{C}}$ *is a family of rdfs graphs, one for each* $\mathcal{C} \in \mathfrak{C}$

Before we define the model of a CKR, we state some of our assumptions and notations below. We put a restriction on our system that for any two distinct contexts $\mathcal{C}_1, \mathcal{C}_2, \mathbf{d}(\mathcal{C}_1) \neq \mathbf{d}(\mathcal{C}_2)$. As a result of this contexts are uniquely determined by their dimension vectors and since each context has exactly one associated RDF graph (possibly empty). Hence from now on, for ease of notation and whenever there is no ambiguity, we use symbols $\mathbf{d}, \mathbf{f}, \mathbf{g}, \mathbf{h}$ to also denote the corresponding graphs in $\mathcal{G}$. Readers should note that whenever these symbols are used in association with $\prec$ operator (for instance $\mathbf{d} \prec \mathbf{g}$) or when these are used as suffixes of qualified symbols (for instance $C_{\mathbf{d}}$) they should be taken as dimension vectors.

Definition 9 (Model of a CKR) *model of a CKR* $\mathfrak{K} = \langle \mathfrak{M}, \mathcal{G} \rangle$ *is a pair* $\langle \mathcal{I}_{\mathfrak{M}}, \mathcal{I}_{\mathcal{G}} \rangle$ *where*

- *$\mathcal{I}_{\mathfrak{M}}$ is an rdfs-interpretation for a meta knowledge space $\mathfrak{M}$, with the additional semantic conditions $\mathcal{I}_{\mathfrak{M}} \models_{rdfs} \mathfrak{M}$.*
- *$\mathcal{I}_{\mathcal{G}} = \{\mathcal{I}_{\mathbf{g}}\}_{\mathbf{g}\in\mathcal{D}}$ is a family of RDFS Interpretations,*
  $\mathcal{I}_{\mathbf{g}} = \langle IR_{\mathbf{g}}, IP_{\mathbf{g}}, IC_{\mathbf{g}}, IEXT_{\mathbf{g}}, ICEXT_{\mathbf{g}}, LV_{\mathbf{g}}, IS_{\mathbf{g}}\rangle$ when the following conditions are satisfied:

  1. $IR_{\mathbf{g}} \subseteq IR_{\mathbf{h}}$ *if* $\mathbf{g} \prec \mathbf{h}$;
  2. $IS_{\mathbf{g}}(a) = IS_{\mathbf{h}}(a)$ *if* $IS_{\mathbf{h}}(a) \in IR_{\mathbf{g}}, \mathbf{g} \prec \mathbf{h}$, *for every* $a \in \Sigma_{\mathbf{U}}$;
  3. $IS_{\mathbf{g}}(a) = IS_{\mathbf{h}}(a)$ *if* $IS_{\mathbf{g}}(a) \in IR_{\mathbf{h}}, \mathbf{g} \prec \mathbf{h}$, *for every* $a \in \Sigma_{\mathbf{U}}$;
  4. $ICEXT_{\mathbf{g}}(IS_{\mathbf{g}}(C)) = ICEXT_{\mathbf{g}}(IS_{\mathbf{g}}(C_{\mathbf{g}}))$;
  5. $ICEXT_{\mathbf{g}}(IS_{\mathbf{g}}(R)) = ICEXT_{\mathbf{g}}(IS_{\mathbf{g}}(R_{\mathbf{g}}))$;
  6. $ICEXT_{\mathbf{h}}(IS_{\mathbf{h}}(C_{\mathbf{g}})) = ICEXT_{\mathbf{g}}(IS_{\mathbf{g}}(C_{\mathbf{g}}))$ *if* $\mathbf{g} \prec \mathbf{h}$;
  7. $IEXT_{\mathbf{h}}(IS_{\mathbf{h}}(R_{\mathbf{g}})) = IEXT_{\mathbf{g}}(IS_{\mathbf{g}}(R_{\mathbf{g}}))$ *if* $\mathbf{g} \prec \mathbf{h}$;
  8. $ICEXT_{\mathbf{g}}(IS_{\mathbf{g}}(C_{\mathbf{d}})) = ICEXT_{\mathbf{h}}(IS_{\mathbf{h}}(C_{\mathbf{d}})) \cap IR_{\mathbf{g}}$, *if* $\mathbf{g} \prec \mathbf{h}$;
  9. $IEXT_{\mathbf{g}}(IS_{\mathbf{g}}(R_{\mathbf{d}})) = IEXT_{\mathbf{h}}(IS_{\mathbf{h}}(R_{\mathbf{d}})) \cap IR_{\mathbf{g}}^2$, *if* $\mathbf{g} \prec \mathbf{h}$;
  10. *For every* $\mathbf{g} \in \mathcal{D}$, $\mathcal{I}_{\mathbf{g}} \models_{rdfs} G_{\mathbf{g}}$

Let's analyze $\mathcal{I}_{\mathcal{G}}$ of the definition 9. Condition 1 imposes that domains increase when the contexts get broader. Condition 2 and 3 imposes that URIs and literals referred in different contexts denotes the same resource, this assumption has been made mainly in motivation with the initiatives regarding standardization of use of URIs. Also in classical rdfs entailment between any two graphs, URIs and Literals in these graphs are treated as denoting the same resources. Conditions 4,5 constraints the denotation of a class (property) in a context to be equivalent to the qualified class (property) obtained by making explicit the dimensional values of the context in which it appears. Conditions 6,7 imposes that the extension of a qualified class (property) qualified with the parameters of the context in which it appears (say $\mathbf{d}$) remain the same in all the broader contexts (i.e., with dimensional values equal to $\mathbf{e} \succ \mathbf{d}$). Conditions 8,9 impose that the interpretation of a qualified predicate in a narrower context (i.e., with dimensional values $\mathbf{d} \prec \mathbf{e}$) is obtained by restricting the interpretation of the same predicate in the broader context $\mathbf{e}$. Condition 10 enforces that every local model of a CKR interpretation satisfies that relative context.

4. Reasoning in CKR

The objective of this section is to define a set of inference rules (IR), along with RDF, RDFS inference rules that are sound and complete w.r.t. the semantics described above. The initial objective of this axiomatization is to provide a naive forward decision procedure for for query answering for the CKR system. Reasoning in CKR is performed at two distinct layers, at the meta level and at the object level. Meta level reasoning is performed in the metaknowledge and has the main objective of inferring coverage relation between contexts. Object level reasoning is performed within and across the set of contexts and has the main objective of inferring knowledge within a context and propagating knowledge across contexts connected via coverage relation.

In a CKR model $\langle \mathcal{I}_{\mathfrak{M}}, \mathcal{I}_{\mathcal{G}} \rangle$, the interpretation of the meta-knowledge $\mathcal{I}_{\mathfrak{M}}$ is a standard RDF(S) interpretation with the additional semantic conditions imposing functionality on the property $\mathcal{D}_i$-Of. An RDFS semantic extension and sound/complete axiomatization with such a construct is already available thanks to OWL-Horst [3]. For lack of

space we don't report these set of rules. For object knowledge, there is principally two kinds of reasoning *local entailment* and *cross-context entailment*. The former happens by virtue of the fact that $\mathcal{I}_{\mathbf{g}}, \mathbf{g} \in \mathcal{D}$ in definition 9 is also an RDFS interpretation and hence, we suppose to have the following "macro rule" for every rdf graph G.

$$\frac{\mathbf{g} : (s_1, p_1, o_1), \ldots, \mathbf{g} : (s_n, p_n, o_n) \text{ and } (s_1, p_1, o_1), \ldots, (s_n, p_n, o_n) \vdash_{\text{rdfs}} (s, p, o)}{\mathbf{g} : (s, p, o)} \mathsf{LR}$$

As an implication of local reasoning, our axiomatization must be an extension of the set of RDFS inference rules. The inference rules for Blank nodes, RDF and RDFS, given in [4], are hence assumed to be included in the axiomatization in addition to the inference rules which capture the additional conditions on RDFS interpretation given in the definition of CKR model. Cross-context entailment allows to deduce conclusions in one of the contexts based on the evidence from other contexts. This set of inference rules for this type of entailment has different context symbols in premises and conclusion (for example, see table 2).

In the following, p, q denotes URIs for properties; u, v denotes a URIs or a blank node, i.e any subject of an rdf triple; a, b denote URIs or Literals, and x, y denote URIs, blank nodes or Literals, i.e they can any possible objects. l stands for literals. $\_ : k, \_ : l, \_ : m, \_ : n$ are blank node identifiers. `alloc(`x`,_:`b`)` , as in standard RDFS inference rules, is used with inference rules that introduce new blank nodes for existing names or blank nodes. Intuitively it means that the blank node $\_ : b$ must have been created by an earlier rule application on the blank node for symbol x, or if there is no such blank node, $\_ : b$ is a "fresh" node which does not occur in the graph. Also the relation `allocated_to` between blank nodes follow transitivity across contexts i.e, if a blank node $\_ : l$ is allocated in a context $\mathbf{d}$ for another blank node $\_ : m$ in context $\mathbf{e}$, which it self was allocated for another blank node $\_ : n$ in context $\mathbf{f}$, then $\_ : l$ is allocated for $\_ : n$.

Table 1 reports the set of rules to infer that a qualified class (property) have the same extension as a base class (property) in their home context. These set of inference rules make sure that condition 4 and 5 of definition 9 is satisfied. The next sets of rule

Table 1. Local completion rules

2a:	$\mathbf{g} : C$ `rdf:type rdfs:Class`	$\Rightarrow$	$\mathbf{g} : C_{\mathbf{g}}$ `rdfs:subClassOf` C
2b:	$\mathbf{g} : C_{\mathbf{g}}$ `rdf:type rdfs:Class`	$\Rightarrow$	$\mathbf{g} : C$ `rdfs:subClassOf` $C_{\mathbf{g}}$
3a:	$\mathbf{g} : R$ `rdf:type rdf:Property`	$\Rightarrow$	$\mathbf{g} : R_{\mathbf{g}}$ `rdfs:subPropertyOf` R
3b:	$\mathbf{g} : R_{\mathbf{g}}$ `rdf:type rdf:Property`	$\Rightarrow$	$\mathbf{g} : R$ `rdfs:subPropertyOf` $R_{\mathbf{g}}$

formalize how instances of various classes and properties and their qualified counterparts move around between various contexts of the system. For this we introduce the notation `presentIn`$(\mathbf{g}, a)$ that holds when the name a, occurs in a triple (s, p, o) that can be derived in context $\mathbf{g}$, i.e $\mathbf{g} : (s, p, o)$ and $s = a$ or $p = a$ or $o = a$. Conventionally for literals we assume that `presentIn`$(\mathbf{g}, l)$ is always true.

Domain expansion

The rules reported in Table 2 formalize the fact that if a qualified triple, i.e triple of the form $(a$ `rdf:type` $C_{\mathbf{d}})$ or $(a\ R_{\mathbf{d}}\ b)$ holds in a narrower context it should hold also in

the broader context. This is by virtue of the fact that domain of contexts expand as one moves to broader contexts. A special remark is necessary for blank nodes. Since in RDF

Table 2. Domain expansion rules

4a:	$\mathbf{g} : (a \ \texttt{rdf:type} \ C_{\mathbf{d}}) \ \& \ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\mathbf{h} : (a \ \texttt{rdf:type} \ C_{\mathbf{d}})$
4b:	$\mathbf{g} : (a \ R_{\mathbf{d}} \ b) \ \& \ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\mathbf{h} : (a \ R_{\mathbf{d}} \ b)$
5:	$\mathbf{g} : (\_ : m \ \texttt{rdf:type} \ C_{\mathbf{d}}) \ \& \ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\begin{cases} \mathbf{h} : (\_ : n \ \texttt{rdf:type} \ C_{\mathbf{d}}) \\ \texttt{alloc}(\_ : m, \_ : n) \end{cases}$
6a:	$\mathbf{g} : (\_ : m \ R_{\mathbf{d}} \ b) \ \& \ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\mathbf{h} : (\_ : n \ R_{\mathbf{d}} \ b) \ (\texttt{alloc}(\_ : m, \_ : n))$
6b:	$\mathbf{g} : (a \ R_{\mathbf{d}} \ \_ : m) \ \& \ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\mathbf{h} : (a \ R_{\mathbf{d}} \ \_ : n) \ (\texttt{alloc}(\_ : m, \_ : n))$
7:	$\mathbf{g} : (\_ : m \ R_{\mathbf{d}} \ \_ : n) \ \& \ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\begin{cases} \mathbf{h} : (\_ : k \ R_{\mathbf{d}} \ \_ : l) \\ \texttt{alloc}(\_ : m, \_ : k), \ \texttt{alloc}(\_ : n, \_ : l) \end{cases}$

blank nodes are treated like existential variables, and therefore, they cannot be shared across contexts. This implies that when we shift up a triple with a blank node, we have to allocate a new fresh node.

Example 2 *An illustration of a derivation using the above mentioned inference rules:*
(i) $\mathbf{g} : (a \ \texttt{rdf} : \texttt{type} \ C), \ \mathfrak{M} : \mathbf{g} \prec \mathbf{h} \vdash \mathbf{h} : (a \ \texttt{rdf} : \texttt{type} \ C_{\mathbf{g}})$

$$\dfrac{\dfrac{\mathbf{g} : (a \ \texttt{rdf} : \texttt{type} \ C)}{\mathbf{g} : (a \ \texttt{rdf} : \texttt{type} \ C_{\mathbf{g}})} \textit{ IR 2b, rdfs 9} \qquad \mathfrak{M} : \mathbf{g} \prec \mathbf{h}}{\mathbf{h} : (a \ \texttt{rdf} : \texttt{type} \ C_{\mathbf{g}})} \textit{ IR 4a}$$

(ii) $\mathbf{g} : (a \ \texttt{rdf} : \texttt{type} \ C_{\mathbf{h}}), \mathfrak{M} : \mathbf{g} \prec \mathbf{h} \vdash \mathbf{h} : (a \ \texttt{rdf} : \texttt{type} \ C)$

$$\dfrac{\dfrac{\mathbf{g} : (a \ \texttt{rdf} : \texttt{type} \ C_{\mathbf{h}}) \qquad \mathfrak{M} : \mathbf{g} \prec \mathbf{h}}{\mathbf{h} : (a \ \texttt{rdf} : \texttt{type} \ C_{\mathbf{h}})} \textit{ IR 4a}}{\mathbf{h} : (a \ \texttt{rdf} : \texttt{type} \ C)} \textit{ IR 2a, rdfs 9}$$

Domain contraction

Table 3 reports the set of rules that allow to propagate triple from broader contexts into narrower context whenever the property of the triple is qualified with dimensions smaller than or equal to the dimensions of the narrower context, i.e to derive $\mathbf{d} : C_{\mathbf{d}}(a)$ from $\mathbf{e} : C_{\mathbf{d}}(a)$ given $\mathbf{d} \prec \mathbf{e}$. This kind of reasoning that happens as a consequence of semantic condition 6 and 7 of the definition 9. Also in this case a remark on blank nodes is necessary. The presence of a blank node in a broader context might not correspond to a resource in the narrower context, as there could be resources in a larger domain that are not present in smaller domains. Hence the conditional execution of 10, 13a, 13b and 14. They only operate on blank nodes that are allocated in the larger domain corresponding to blank nodes in smaller domain and hence sure of their presence in smaller domain.

Table 3. Domain contraction rules

8a:	$\begin{cases} \mathbf{h} : (a\ \texttt{rdf:type}\ C_{\mathbf{d}}) \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h} \\ \texttt{presentIn}(\mathbf{g}, a) \end{cases}$	$\Rightarrow$	$\mathbf{g} : (a\ \texttt{rdf:type}\ \ C_{\mathbf{d}})$
8b:	$\begin{cases} \mathbf{h} : (a\ R_{\mathbf{d}}\ b) \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h} \\ \texttt{presentIn}(\mathbf{g}, a), \texttt{presentIn}(\mathbf{g}, b) \end{cases}$	$\Rightarrow$	$\mathbf{g} : (a\ R_{\mathbf{d}}\ b)$
9a:	$\mathbf{h} : (a\ \texttt{rdf:type}\ \ C_{\mathbf{g}})\ \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\mathbf{g} : (a\ \texttt{rdf:type}\ \ C)$
9b:	$\mathbf{h} : (a\ R_{\mathbf{g}}\ b)\ \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\mathbf{g} : (a\ R\ b)$
9:	$\mathbf{h} : (\_ : m\ \texttt{rdf:type}\ C_{\mathbf{g}}) \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\begin{cases} \mathbf{g} : (\_ : n\ \texttt{rdf:type}\ C) \\ \texttt{alloc}(\_ : m, \_ : n) \end{cases}$
10:	$\begin{cases} \mathbf{h} : (\_ : m\ \texttt{rdf:type}\ C_{\mathbf{d}}) \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h} \\ \texttt{alloc}(\_ : n, \_ : m) \end{cases}$	$\Rightarrow$	$\mathbf{g} : (\_ : n\ \ \texttt{rdf:type}\ C_{\mathbf{d}})$
11a:	$\mathbf{h} : (\_ : m\ R_{\mathbf{g}}\ b) \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\mathbf{g} : (\_ : n\ R\ b)\ (\texttt{alloc}(\_ : m, \_ : n))$
11b:	$\mathbf{h} : (a\ R_{\mathbf{g}}\ \_ : m) \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\mathbf{g} : (a\ R\ \_ : n)\ (\texttt{alloc}(\_ : m, \_ : n))$
12:	$\mathbf{h} : (\_ : m\ R_{\mathbf{g}}\ \_ : n) \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h}$	$\Rightarrow$	$\begin{cases} \mathbf{g} : (\_ : k\ R\ \_ : l) \\ \texttt{alloc}(\_ : m, \_ : k) \\ \texttt{alloc}(\_ : n, \_ : l) \end{cases}$
13a:	$\begin{cases} \mathbf{h} : (\_ : m\ R_{\mathbf{d}}\ b) \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h} \\ \texttt{alloc}(\_ : n, \_ : m), \texttt{presentIn}(\mathbf{g}, b) \end{cases}$	$\Rightarrow$	$\mathbf{g} : (\_ : n\ R_{\mathbf{d}}\ b)$
13b:	$\begin{cases} \mathbf{h} : (a\ R_{\mathbf{d}}\ \_ : m) \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h} \\ \texttt{alloc}(\_ : n, \_ : m), \texttt{presentIn}(\mathbf{g}, a) \end{cases}$	$\Rightarrow$	$\mathbf{g} : (a\ R_{\mathbf{d}}\ \_ : n)$
14:	$\begin{cases} \mathbf{h} : (\_ : m\ R_{\mathbf{d}}\ \_ : n) \ \&\ \mathfrak{M} : \mathbf{g} \prec \mathbf{h} \\ \texttt{alloc}(\_ : k, \_ : m), \texttt{alloc}(\_ : l, \_ : n) \end{cases}$	$\Rightarrow$	$\mathbf{g} : (\_ : k\ R_{\mathbf{d}}\ \_ : l)$

5. Completeness of CKR entailment

In this section, we will prove soundness and completeness of the inference rules w.r.t., consequence relation in CKR. We say that a triple (s, p, o) logically follows from $\mathfrak{K}$ in the context $\mathbf{g}$, in symbols $\mathfrak{K} \models \mathbf{g} : (s, p, o)$ if, for all models $(\mathcal{I}_{\mathfrak{M}}, \mathcal{I}_{\mathcal{G}})$ of $\mathfrak{K}$, $\mathcal{I}_{\mathbf{g}} \models_{\text{rdfs}} (s, p, o)$. The completeness proof, relies on the construction of an interpretation called *herbrand interpretation* as is done in the RDFS completeness proof in [4]. It is an interpretation constructed from the lexical items in $\mathcal{G}$. The strategy, we will adopt for the completeness proof, is similar to that of [4]. In brief the steps are as follows

- form the closure(described further) of the CKR $\mathfrak{K}$ w.r.t to the inference rules.
- construct a herbrand interpretation from this closure in which there is a reversible mapping from it's vocabulary to the elements in it's domain.
- prove that the herbrand interpretation is a model for the CKR $\mathfrak{K}$

Before proving completeness, we need to introduce some technicalities. In RDF inferencing, the following two inference rules are used for graph entailement. We also need them for completeness proof.

lg : s p o	$\Rightarrow$ s p _:n(alloc(o,_:n))
gl : s p _:n(alloc(o,_:n))	$\Rightarrow$ s p o

which allocate a blank node for every object element of a triple, only when it is a literal. We also suppose, w.l.o.g that, the set of blank nodes of the graphs of a CKR are pairwise disjoint. On the basis of the above rule, we define the mapping sur, which maps every URI and blank node into itself, and every literal l of $U(G)$ into the blank node allocated via the rule "lg". Finally we define the operator, closure of a graph. Given a CKR $\mathfrak{K}$, the closure operator in symbols, $\mathcal{D}(\mathfrak{K})$, is a set of labeled triples $\mathbf{g} : (s, p, o)$ defined as follows:

1. if $(s, p, o) \in G_{\mathbf{g}}$ then $\mathbf{g} : (s, p, o) \in \mathcal{D}(\mathfrak{K})$;
2. If (s, p, o) is an RDF(S) axiomatic triple, then $\mathbf{g} : (s, p, o) \in \mathcal{D}(\mathfrak{K})$ for any $\mathbf{g}$;
3. $\mathcal{D}(\mathfrak{K})$ is closed under LR, the rules of table 1, 2 and 3 and the "lg" and "gl" rules.

The set $cl(\mathfrak{K}, \mathbf{g})$ is defined as $\{(s, p.o) \mid \mathbf{g} : (s, p.o) \in \mathcal{D}(\mathfrak{K})\}$. The *herbrand interpretation* $\mathcal{I}^H_{\mathcal{G}}$, for the object knowledge base, $\mathcal{G}$ of a CKR, is given by $\mathcal{I}^H_{\mathcal{G}} = \{\mathcal{I}^H_{\mathbf{g}}\}_{\mathbf{g}\in\mathcal{G}}$ where each $\mathcal{I}^H_{\mathbf{g}} = \langle \text{IR}^H_{\mathbf{g}}, \text{IP}^H_{\mathbf{g}}, IC^H_{\mathbf{g}}, \text{IEXT}^H_{\mathbf{g}}, \text{ICEXT}^H_{\mathbf{g}}, \text{LV}^H_{\mathbf{g}}, \text{IS}^H_{\mathbf{g}} \rangle$ for the graph $\mathbf{g}$ is given by the following construction. For the construction of herbrand interpretation and for the completeness proof, the approach is an adaptation of the one in [4].

- $\text{LV}^H_{\mathbf{g}} = \{l | cl(\mathfrak{K}, \mathbf{g})$ contains $sur(l)$ `rdf:type rdfs:Literal`$\}$
- $\text{IR}^H_{\mathbf{g}} = U(cl(\mathfrak{K}, \mathbf{g}))$
- $\text{IP}^H_{\mathbf{g}} = \{p \mid (sur(p), \texttt{rdf:type}, \texttt{rdf:Property}) \in cl(\mathfrak{K}, \mathbf{g})\}$
- $IC^H_{\mathbf{g}} = \{s \mid (sur(s), \texttt{rdf:type}, \texttt{rdfs:Class}) \in cl(\mathfrak{K}, \mathbf{g})\}$
- for every $p \in \text{IP}^H_{\mathbf{g}}$, $\text{IEXT}^H_{\mathbf{g}}(p) = \{(s, o) \mid (sur(s), p, sur(o)) \in cl(\mathfrak{K}, \mathbf{g})\}$
- for every $o \in IC^H_{\mathbf{g}}$, $\text{ICEXT}^H_{\mathbf{g}}(o) = \{s | (sur(s), \texttt{rdf:type}, sur(o)) \in cl(\mathfrak{K}, \mathbf{g})\}$
- $\text{IS}^H_{\mathbf{g}}(x) = \begin{cases} xml(x), \text{ if } x \text{ is a well typed XML literal in } cl(\mathfrak{K}, \mathbf{g}), \\ x \text{ otherwise} \end{cases}$
  where $xml(.)$ is the xml datatype function, that maps valid xml literals to xml value space.

Let B be a function blank node mapping such that if x is blank node allocated to an xml literal l then $B(x) = xml(l)$; if it is allocated to a plain literal l, then $B(x) = l$ otherwise $B(x) = x$.

Lemma 1 *If CKR does not have an XML clash*[2]*, then* $\mathcal{I}^H_{\mathcal{G}} \models \mathcal{G}$.

Proof By hypothesis, for every $\mathbf{g} \in \mathcal{G}$ does not have an XML Clash. Since each $\mathcal{I}^H_{\mathbf{g}}$ is closed with respect to RDF, RDFS inference rules, $\mathcal{I}^H_{\mathbf{g}} \models_{rdfs} \mathbf{g}$ [4]. Now in order to prove that $\mathcal{I}^H_{\mathcal{G}} \models \mathcal{G}$, we need to prove that $\mathcal{I}^H_{\mathcal{G}} = \{\mathcal{I}^H_{\mathbf{g}}\}_{\mathbf{g}\in\mathcal{G}}$ satisfies each of the semantic conditions of $\mathcal{I}_{\mathcal{G}}$ in definition 9. Below we enumerate each condition and prove that they are actually satisfied

1. Suppose if $\mathbf{g} \prec \mathbf{h}$ and if $\text{IR}_{\mathbf{g}} \not\subseteq \text{IR}_{\mathbf{h}}$, this means that there exists a element $v \in \text{IR}_{\mathbf{g}}$ and $v \notin \text{IR}_{\mathbf{h}}$. This means that there exists a triple $(s, p, o) \in cl(\mathbf{g})$ where $s = v$ or $p = v$ or $o = v$ and $(s, p, o) \notin cl(\mathbf{h})$. But this cannot be the case by the virtue of execution of inference rules (4,5,6,7) to completion as these rules make sure that there is a triple (s, p, o) in $cl(\mathbf{h})$ for every (s, p, o) in $cl(\mathbf{g})$.

[2] a graph has an XML clash, when it contains the statement, l `rdf:type rdf:XMLLiteral`, and l is a lexically invalid XML literal

2. Since every $v \in U(cl(\mathbf{g}))$ is interpreted to it self by the $\text{IS}^H_{\mathbf{g}}$ except the valid XML literals that are interpreted to the XML value space, this condition is trivially satisfied.
3. Similar to the above
4. Taken care by inference rules 2a,2b
5. Taken care by inference rules 3a,3b
6. In order to prove the condition $\text{ICEXT}_{\mathbf{g}}(\text{IS}_{\mathbf{g}}(C_{\mathbf{g}})) = \text{ICEXT}_{\mathbf{h}}(\text{IS}_{\mathbf{h}}(C_{\mathbf{g}}))$ if $\mathbf{g} \prec \mathbf{h}$. We could prove this by showing that (i) $\text{ICEXT}_{\mathbf{g}}(\text{IS}_{\mathbf{g}}(C_{\mathbf{g}})) \subseteq \text{ICEXT}_{\mathbf{h}}(\text{IS}_{\mathbf{h}}(C_{\mathbf{g}}))$ and (ii) $\text{ICEXT}_{\mathbf{h}}(\text{IS}_{\mathbf{h}}(C_{\mathbf{g}})) \subseteq \text{ICEXT}_{\mathbf{g}}(\text{IS}_{\mathbf{g}}(C_{\mathbf{g}}))$ holds
   case (i) Suppose an element $v \in \text{ICEXT}_{\mathbf{g}}(\text{IS}_{\mathbf{g}}(C_{\mathbf{g}}))$ then there exists a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{g}}))$ in $cl(\mathbf{g})$. Suppose $sur(v)$ and $sur(C_{\mathbf{g}})$ are URIs then by virtue of inference rule 4a, we have a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{g}}))$ in $cl(\mathbf{h})$. Suppose if they are blank nodes (or one of them) then by rules 5,6a,6b,7 then there exists a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{g}}))$ in $cl(\mathbf{h})$
   case (ii) Suppose an element $v \in \text{ICEXT}_{\mathbf{h}}(IS_{\mathbf{h}}(C_{\mathbf{g}}))$ then there exists a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{g}}))$ in $cl(\mathbf{h})$. Suppose $sur(v)$ and $sur(C_{\mathbf{g}})$ are URIs then by virtue of inference rule 8a, we have a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{g}}))$ in $cl(\mathbf{g})$. Suppose if they are blank nodes (or one of them) then by rules 9,11a,11b then there exists a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{g}}))$ in $cl(\mathbf{g})$
7. Similar to the above
8. In order to prove the condition $\text{ICEXT}_{\mathbf{g}}(\text{IS}_{\mathbf{g}}(C_{\mathbf{d}})) = \text{ICEXT}_{\mathbf{h}}(\text{IS}_{\mathbf{h}}(C_{\mathbf{d}})) \cap \text{IR}_{\mathbf{g}}$ if $\mathbf{g} \prec \mathbf{h}$. We prove this by showing that both (i) $\text{ICEXT}_{\mathbf{g}}(\text{IS}_{\mathbf{g}}(C_{\mathbf{d}})) \subseteq \text{ICEXT}_{\mathbf{h}}(\text{IS}_{\mathbf{h}}(C_{\mathbf{d}})) \cap \text{IR}_{\mathbf{g}}$ and (ii)$\text{ICEXT}_{\mathbf{h}}(\text{IS}_{\mathbf{h}}(C_{\mathbf{d}})) \cap \text{IR}_{\mathbf{g}} \subseteq \text{ICEXT}_{\mathbf{g}}(\text{IS}_{\mathbf{g}}(C_{\mathbf{d}}))$ holds
   case(i) Suppose an element $v \in \text{ICEXT}_{\mathbf{g}}(\text{IS}_{\mathbf{g}}(C_{\mathbf{d}}))$ then there exists a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{d}}))$ in $cl(\mathbf{g})$. Suppose $sur(v)$ and $sur(C_{\mathbf{g}})$ are URIS then by virtue of inference rule 4a, we have a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{g}}))$ in $cl(\mathbf{h})$. Suppose if they are blank nodes (or one of them) then by rules 5,6a,6b,7 then there exists a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{g}}))$ in $cl(\mathbf{h})$
   case(ii) We prove this by contradiction
   Suppose if $v \notin \text{ICEXT}_{\mathbf{g}}(\text{IS}_{\mathbf{g}}(C_{\mathbf{d}}))$ and if there exists a element $v \in \text{IR}_{\mathbf{g}}$ and $v \in \text{IEXT}_{\mathbf{h}}(\text{IS}_{\mathbf{h}}(C_{\mathbf{d}}))$ then there exists a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{d}}))$ in $cl(\mathbf{h})$. suppose $sur(v)$ and $sur(C_{\mathbf{d}}))$ are uris then there exists a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{d}}))$ by inference rules 8a. Suppose if they are blank nodes (or one of them) then by rule 10,13a,13b,14 there exists a triple $(sur(v)\ \texttt{rdf:type}\ sur(C_{\mathbf{g}}))$ in $cl(\mathbf{g})$ since $v \in \text{IR}_{\mathbf{g}}$(i.e $sur(v)$ is allocated in $cl(\mathbf{h})$ for $sur(v)$ in $cl(\mathbf{g})$. Hence this implies that $v \in \text{IEXT}_{\mathbf{g}}(\text{IS}_{\mathbf{g}}(C_{\mathbf{d}}))$ (contradiction)
9. Similar to the above
10. By hypothesis we have that g does not have an XML clash for every $\mathbf{g} \in \mathcal{G}$, each $\mathcal{I}^H_{\mathbf{g}}$ is closed with respect to RDF, RDFS inference rules and is an rdfs model for g[4].

Theorem 1 (completeness) *Suppose if CKR does not have an XML clash and CKR* $\models \mathbf{g} : (s, p, o)$ *then CKR* $\vdash \mathbf{g} : (\sigma(s), p, \sigma(o))$ *where* σ *is a renaming function for blank nodes, for every* $\mathbf{g} \in \mathcal{G}$ *in CKR.*

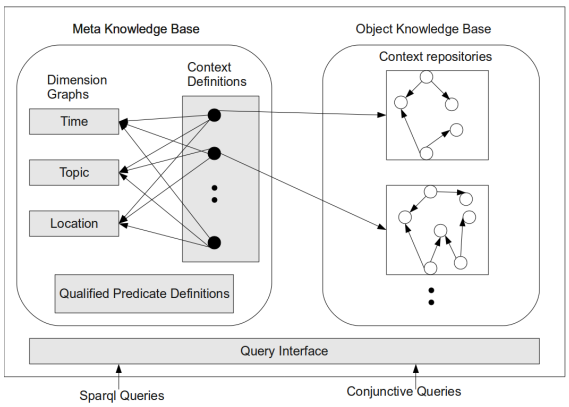

Figure 1. Implementation architecture.

Proof suppose CKR $\models \mathbf{g} : (s, p, o)$ then this means that $\mathcal{I}_{\mathbf{g}}^{H} \models (s, p, o)$. This means that there exists a mapping A from blank nodes to $\text{IR}_{\mathbf{g}}^{H}$ such that $\mathcal{I}_{\mathbf{g}}^{H} + A \models (s, p, o)$. This implies that $\mathcal{I}_{\mathbf{g}}^{H} + A \models_{rdfs} (s, p, o)$. This implies that $\mathcal{I}_{\mathbf{g}}^{H} + A \models_{rdf} (s, p, o)$. This implies that $\mathcal{I}_{\mathbf{g}}^{H} + A \models_{simple} (s, p, o)$ (Since semantic conditions of simple interpretations are contained in RDF interpretations which themselves are contained in RDFS interpretations). Let us now extend A such that $A(l) = xml(l)$ for any valid xml literal l else $A(x) = x$ for any other x that is not a blank node or xml literal. now $\mathcal{I}_{\mathbf{g}}^{H} + B \models (A(s), p, A(o))$. where B is the identity mapping for blank nodes. This implies that there exists a triple $(sur(A(s))\ p\ sur(A(o))$ in $cl(\mathfrak{K}, \mathbf{g})$ which is obtained only by the set of inference rules for rdf, rdfs, lg, gl and ckr inference rules.

6. Implementation

The formal framework described previously is implemented as an extension of Sesame RDF store with OWLIM plugin[3] In the system, contexts are represented as named graph (NG). The NG identifier is used for attaching dimension values to the context that corresponds to dimension-value pairs outside the box. The actual data in the graph pertains to the knowledge about the domain corresponding to what is inside the box. The standard approach of taking the union of all the graphs in the repository and interpreting as a single RDF interpretation is not applicable since data in different contexts can have inconsistencies mentioned before. Hence, we store and reason data in each context separately from the data in other contexts. The overall architecture of the contextualized knowledge repository is graphically depicted in Fig. 1. The repository is principally divided into two components:

Meta Knowledge Base Meta-knowledge base, as is a unique graph that contains all the information about dimensions, their values, cover relations between these values and definition of qualified classes/properties, it is implemented as a single Sesame/OWLLim Repository. In the current prototype, we limit to the following three main contextual di-

[3]See [4] and `http://www.ontotext.com/owlim`

mensions:
Time Dimensions values are time intervals of the form (start-time, end-time). Cover relation is the standard containment relation between intervals. There is presently no serialization for this structure, because, the relation between two values can be determined by the start time and end time values of the interval.
Location dimension determines the geographical region in which the set of statements in a context is true. This structure is represented using OWL-Horst serialization.
Topic dimension determines the subject topic that a context pertains to. Similarly to locations, the current structure of this dimension is also is represented as an OWL-Horst ontology. Qualified Classes and Properties are specified to the system by means of the following sample syntax and is part of the meta knowledge space.

```
<rdf:Description rdf:about="qualified:Class">
   <rdfs:subClassOf rdf:resource="rdfs:Class"/>
</rdf:Description>
 <qualified:Class rdf:about="http://ns#Winner_In_FrenchOpen_2010">
   <qualified:BaseClass rdf:resource="http://ns#Winner"/>
   <dimension:Time rdf:resource="2010"/>
   <dimension:Topic rdf:resource="French\_Open\_Tennis"/>
 </qualified:Class>
```

Qualified Classes and Properties are instances of two designated meta classes called `qualified:Class` and `qualified:Property` as shown in the above rdf snippet. In this snippet, `http://ns#Winner_In_FrenchOpen_2010` is a qualified class whose base class is `http://ns#Winner` and intuitively represents an abstract concept of Winner in context of $\langle$2010, French_Open_Tennis$\rangle$.

Object Knowledge Base Actual RDFS content of each context defined in meta knowledge base is materialized as an individual Sesame/OWLim repository. The repositories are logically organized as hierarchies based on cover relation of their corresponding dimension values. We currently have implemented conjunctive query answering on the whole CKR that is sound and complete with respect to it's semantics. Moreover, individual contexts can by queried with SPARQL.

7. Related Work

One of the approaches that tried to add a single temporal dimension to standard RDFS was the work in [7]. The authors define the notion of a 'temporal rdf graph' in which is triple is augmented of the form $(s, p, o) : t$, where t is a time point. A temporal graph $g1$ temporally entails another temporal graph $g2$ if at every instant t, $g1(t)$ entails g2(t), where $g(t) = \{(s, p, o)|(s, p, o) : t \in g\}$. The authors provide a deductive system and query language that motivates the usefulness of temporal RDF graphs. although the semantics and inference rules are suitable to time dimension, it cannot be applied to any of the other dimension in the literature of contexts like locations, topics, provenance etc.

Annotated RDF [10] is a theoretical framework and a concrete proposal to extend RDF triples with a set of annotations taken from an n-tuple of partially ordered set. Considering partially ordered meta-attributes is what makes the two approaches quite similar. The main difference however is that here we do the meta-tagging at the level of an RDF graph, rather for each single triple. For efficiency reasons, graph level annotation is

preferable w.r.t., triple level annotation in formalizing context, since usually the context is fixed for a group of statements.

In [11], the authors, also give a semantics for graphs annotated with values from a lattice structure. The authors give a semantics based on interpretation structure which is popularly used in many-valued logics. The main difference from our work is that the semantics in [11] impose a restriction on dimensions used, so that it is a complete lattice with join($\vee$), meet($\wedge$) operators with also $\top$ and $\bot$ values, where as, we do not have such restrictions as, we assume the use of existing ontologies of time, topic, locations etc. for dimensions which necessarily are not in lattice structure. Another difference is that, the extension of the top class (`rdfs:Resource`) is the same across all contexts, where as for us, it's extension is context-dependent and moreover, for any contexts $\mathbf{d}$ and $\mathbf{e}$ with $\mathbf{d} \prec \mathbf{e}$, then extension of `rdfs:Resource` in context $\mathbf{d}$ is a subset of `rdfs:Resource` in $\mathbf{e}$. Also any symbol a is interpreted as the same resource across different contexts in [11], where as for us, it is context dependent. We also have the mechanism of qualified predicates, which mainly used to refer to other classes and properties in other contexts and can be used to state lifting rules as described in section 3, such constructs are absent in [11].

Jie Bao et.al. [12] provide a more concrete formalization of theory of context by McCarthy. They extend/modify the theory described in [13] with a predicate $\mathsf{isin}(c, \phi)$ representing the fact that the triple ϕ is in the context c. This approach is particularly suited for manipulating contexts as first class objects. A considerable number of constructs were introduced for combining contexts (like $c_1 \wedge c_1$, $c_1 \vee c_2$ and $\neg c$) and for relating contexts (like $c \Rightarrow c_2$, and $c \rightarrow c_2$). The work, however, does not give an axiomatization of all these operators. the same family of the one proposed here. The main difference between the two are that we provide a set of propagation patterns, a sound and complete axiomatization and an implementation based on Sesame 2 named graphs.

The authors in [14] describe a detailed architecture of Yars, a semantic repository and with a search/query engine which stores rdf data in the form of $((s, p, o), c)$ where (s, p, o) is an RDF triple in context c. The architecture contains modules that includes the crawler, index manager and indexer, query processing and a query distribution module. The indexing module contains an keyword based indexer and a quad index that is distributed over multiple servers that also includes a context identifier as an index key. Although the work motivates the use of context in semantic repository, it does not provide a semantics for various contexts or do not show how contexts have be related to one another or how queries can be forwarded to the right context or how to find answers that are distributed across multiple contexts.

CKR is a special case of quantified Multi Context Systems (a.k.a distributed first order logic) [15] in which contexts interfere only via coverage relation. In CKR, bridge rules are not listed explicitly as in MCS, but they are "automatically" determined by the coverage relation formalized in the metaknowledege. The bridge rules "generated" by the coverage statements are described in the section about inference rules (section 4).

8. Conclusion

We have presented a framework for context based knowledge representation that is based on RDFS. We introduce a mechanism called qualification that is useful in a data integration scenario. We provide a semantics that we found intuitive for contexts. We pro-

vide a set of inference rules that can be used to construct a forward chained semantic repository and provide completeness proof for the inference rules. We further describe the architecture of our implemented system that is built by extending the Sesame RDF store.

References

[1] M.Homola, L.Serafini, and A.Tamilin, "Modeling contextualized knowledge," in *Procs. of the 2nd Workshop on Context, Information and Ontologies* (V. Ermolayev, J. M. Gomez-Perez, P. Haase, and P. Warren, eds.), vol. 626 of *CEUR-WS*, 2010.

[2] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, "OWL 2 Web Ontology Language Primer," W3C Recommendation, World Wide Web Consortium, October 2009.

[3] H. J. ter Horst, "Completeness, decidability and complexity of entailment for rdf schema and a semantic extension involving the owl vocabulary," *Web Semant.*, vol. 3, no. 2-3, pp. 79–115, 2005.

[4] P. Hayes, "Rdf semantics," Feb. 2004.

[5] P. Hitzler, R. Sebastian, and M. Krötzsch, *Foundations of Semantic Web Technologies*. London: Chapman & Hall/CRC, 2009.

[6] R. Q. Dividino, S. Sizov, S. Staab, and B. Schueler, "Querying for provenance, trust, uncertainty and other meta knowledge in rdf," *J. Web Sem.*, vol. 7, no. 3, pp. 204–219, 2009.

[7] C. Gutierrez, C. A. Hurtado, and A. A. Vaisman, "Temporal rdf," in *ESWC*, pp. 93–107, 2005.

[8] M. Stocker and E. Sirin, "Pelletspatial: A hybrid rcc-8 and rdf/owl reasoning and query engine," in *Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2009), Chantilly, VA, United States, October 23-24, 2009*, vol. 529 of *CEUR Workshop Proceedings*, 2009.

[9] D.Lenat, "The Dimensions of Context Space," tech. rep., CYCorp, 1998.

[10] O. Udrea, D. R. Recupero, and V. S. Subrahmanian, "Annotated rdf," *ACM Trans. Comput. Logic*, vol. 11, no. 2, pp. 1–41, 2010.

[11] U. Straccia, N. Lopes, G. Lukacsy, and A. Polleres, "A general framework for representing and reasoning with annotated semantic web data," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010), Special Track on Artificial Intelligence and the Web*, July 2010.

[12] J. Bao, J. Tao, and D. McGuinness, "Context representation for the semantic web." In Web Science Conference. Online at `http://www.websci10.org/`, 2010.

[13] J. McCarthy, "Notes on formalizing context," in *IJCAI'93*, 1993.

[14] A. Harth, J. Umbrich, A. Hogan, and S. Decker, "Yars2: a federated repository for querying graph structured data from the web," in *Proceedings of the ISWC/ASWC-2007*, 2007.

[15] C. Ghidini and L. Serafini, "Distributed first order logics," in *Frontiers of Combining Systems*, 1998.