Analyzing Real-World SPARQL Queries in the Light of Probabilistic Data

Joerg Schoenfisch and Heiner Stuckenschmidt

Data and Web Science Group
University of Mannheim
B6 26, 68159 Mannheim, Germany
{joerg,heiner}@informatik.uni-mannheim.de

Abstract. Handling uncertain knowledge – like information extracted from unstructured text, with some probability of being correct – is crucial for modeling many real world domains. Ontologies and ontology-based data access (OBDA) have proven to be versatile methods to capture this knowledge. Multiple systems for OBDA have been developed and there is theoretical work towards probabilistic OBDA, namely identifying efficiently processable (safe) queries. However, there is no analysis on the safeness of probabilistic queries in real-world applications, or in other words the feasibility of fulfilling users' information needs over probabilistic data. In this paper we investigate queries collected from several public SPARQL endpoints and determine the distribution of safe and unsafe queries. This analysis shows that many queries in practice are safe, making probabilistic OBDA feasible and practical to fulfill real-world users' information needs.

Keywords: safeness of probabilistic queries, SPARQL, probabilistic data, linked data

1 Motivation

Ontology-based Data Access (ODBA) has received a lot of attention in the Semantic Web Community. In particular, results on light-weight description logics that allow efficient reasoning and query answering provide new possibilities for using ontologies in data access. One approach for ontology-based data access is to rewrite a given query based on the background ontology in such a way that the resulting – more complex – query can directly be executed on a relational database. This is possible for different light-weight ontology languages, in particular the $DL\text{-}Lite$ family [1].

At the same time, many applications in particular on the (Semantic) Web have to deal with uncertainty in the data. Examples are large-scale information extraction from text or the integration of heterogeneous information sources. To cope with uncertainty, the database community has investigated probabilistic databases where each tuple in the database is associated with a probability indicating the belief in the truth of the respective statement. Querying a probabilistic database requires not only to retrieve tuples that match the query, but also to compute a correct probability for each answer. Research in probabilistic databases has shown that there is a strict dichotomy of safe (data complexity in PTIME) and unsafe (in #P-hard) queries [15].

The goal of our work is to develop data access methods that can use background knowledge in terms of a light-weight ontology and also deal with uncertainty in the data. A promising idea for efficiently computing probabilistic query answers is to use existing approaches for OBDA based on query rewriting and pose the resulting query against a probabilistic database that computes answers with associated probabilities.

Jung et al. have shown that query rewriting for OBDA can directly be lifted to the probabilistic case [10]. Furthermore, they prove that the complexity results and the dichotomy of safe and unsafe queries also carry over. As we have shown in [14] this approach is well applicable in the real-world and scales well to datasets of the size of NELL [5] with millions of triples.

In this paper we address the question of how real-world queries, and consequently users' information needs, are distributed concerning query safeness. Therefore, we analyze the queries contained in the Linked SPARQL Queries Dataset (LSQ) [13], which consists of real-world queries collected from several public SPARQL endpoints.

The paper is structured as follows: In Section 2 we give the definition of extensional query processing and query safeness for probabilistic queries and the application to OBDA. The dataset and the analysis and its results are described in Section 3. In Section 4 we briefly discuss some related work analyzing SPARQL queries. We conclude and give directions for future work in Section 5.

2 Preliminaries

In this section we briefly introduce $DL\text{-}Lite$, the description logic underlying the OWL 2 QL profile. Next, we detail how the distribution semantics for probabilistic description logics [12] is applied to $DL\text{-}Lite_R$. Then, we explain extensional query processing – the key to efficient query processing in probabilistic database – and give the definition for query safeness. This lays the foundation for tractable ODBA on top of probabilistic data.

2.1 $DL\text{-}Lite_R$ and the Distribution Semantics

In $DL\text{-}Lite_R$ concepts and roles are formed in the following syntax [4]:

$$\begin{array}{ll} B \rightarrow A \mid \exists R & \qquad C \rightarrow B \mid \neg B \\ R \rightarrow P \mid P^- & \qquad E \rightarrow R \mid \neg R \end{array}$$

where A denotes an atomic concept, P an atomic role, and P^- the inverse of the atomic role P. B denotes a basic concept, i.e. either an atomic concept or a concept of the form $\exists R$, where R denotes a basic role, that is, either an atomic role or the inverse of an atomic role. C denotes a general concept, which can be a basic concept or its negation, and E denotes a general role, which can be a basic role or its negation.

A $DL\text{-}Lite_R$ knowledge base (KB) $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ models a domain in terms of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. A TBox is formed by a finite set of inclusion assertions of the form $B \sqsubseteq C$ or $R \sqsubseteq E$. An ABox is formed by a finite set of membership assertions on atomic concepts and on atomic roles, of the form $A(a)$ or $P(a,b)$ stating respectively that the object denoted by the constant a is an instance of A and that the pair of objects denoted by the pair of constants (a,b) is an instance of the role P.

The semantics of a DL is as an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, consisting of a nonempty interpretation domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ that assigns to each concept C a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, and to each role R a binary relation $R^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$:

$$\begin{aligned}
A^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}}\\
P^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}\\
(P)^{\mathcal{I}} &= \{(o_2, o_1) | (o_1, o_2) \in P^{\mathcal{I}}\}\\
(\exists R)^{\mathcal{I}} &= \{o \mid \exists o'.(o, o') \in R^{\mathcal{I}}\}\\
(\neg B)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus B^{\mathcal{I}}\\
(\neg R)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}}
\end{aligned}$$

An interpretation $\mathcal{I}$ is a model of $C_1 \sqsubseteq C_2$, where C_1,C_2 are general concepts if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$. Similarly, I is a model of $E_1 \sqsubseteq E_2$, where E_1, E_2 are general roles if $E_1^{\mathcal{I}} \subseteq E_2^{\mathcal{I}}$.

To specify the semantics of membership assertions, the interpretation function is extended to constants, by assigning to each constant a a distinct object $a^{\mathcal{I}} \in \Delta^{I}$. This enforces the unique name assumption on constants. An interpretation $\mathcal{I}$ is a model of a membership assertion $A(a)$ (resp., $P(a,b)$), if $a^{\mathcal{I}} \in A^{\mathcal{I}}$ (resp., $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$).

Given an assertion α and an interpretation $\mathcal{I}$, $\mathcal{I} \models \alpha$ denotes the fact that $\mathcal{I}$ is a model of α. Given a (finite) set of assertions λ, $\mathcal{I} \models \lambda$ denotes the fact that $\mathcal{I}$ is a model of every assertion in λ. A model of a KB $K = \langle \mathcal{T}, \mathcal{A} \rangle$ is an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$. A KB is satisfiable if it has at least one model. A KB K logically implies an assertion α, written $K \models \alpha$, if all models of K are also models of α. Similarly, a TBox T logically implies an assertion α, written $T \models \alpha$, if all models of T are also models of α.

A TIP-OWL (tuple-independent OWL) knowledge base $\mathcal{TKB} = \langle \mathcal{T}, \mathcal{A}, P \rangle$ consists of a *DL-Lite*$_R$ T-Box $\mathcal{T}$, an ABox $\mathcal{A}$, and a probability distribution $P : \mathcal{A} \rightarrow [0,1]$. Abusing terminology, we say that $\mathcal{KB} \subseteq \mathcal{TKB}$ if they have the same T-Box and the A-Box of $\mathcal{KB}$ is a subset of the A-Box of $\mathcal{TKB}$. We adopt the independent tuple semantics in the spirit of the probabilistic semantics for logic programs proposed in [7] and define the probabilistic semantics of a TIP-OWL knowledge base in terms of a distribution over possible knowledge bases as follows:

Definition 1. *Let $\mathcal{TKB} = \langle \mathcal{T}, \mathcal{A}, P \rangle$ be a TIP-OWL knowledge base. Then the probability of a DL-Lite$_R$ Knowledge Base $\mathcal{KB} = \langle \mathcal{T}, \mathcal{A}' \rangle \subseteq \mathcal{TKB}$ is given by:*

$$P(\mathcal{KB}|\mathcal{TKB}) = \prod_{a' \in \mathcal{A}'} P(a') \cdot \left(1 - \prod_{a \in \mathcal{A} \setminus \mathcal{A}'} P(a) \right)$$

Based on this semantics, we can now define the probability of existential queries over TIP-OWL knowledge bases as follows. First, the probability of a query over a possible knowledge base is one if the query follows from the knowledge base and zero otherwise:

$$P(Q|\mathcal{KB}) = \begin{cases} 1 & \mathcal{KB} \models Q \\ 0 & \text{otherwise} \end{cases}$$

Taking the probability of possible knowledge bases into account, the probability of an existential query over a possible knowledge base becomes the product of the probability of that possible knowledge base and the probability of the query given that knowledge base. By summing up these probabilities over all possible knowledge bases, we get the following probability for existential queries over TIP-OWL knowledge bases:

$$P(Q|\mathcal{TBK}) = \sum_{\mathcal{KB} \subseteq \mathcal{TKB}} P(Q|\mathcal{KB}) \cdot P(\mathcal{KB}|\mathcal{TKB})$$

This defines a complete probabilistic semantics for TIP-OWL knowledge bases and queries.

2.2 Implementing Reasoning on Top of Probabilistic Databases

In this section, we first briefly recall the idea of first-order rewritability of queries in $DL\text{-}Lite$ and then show that the query rewriting approach proposed by [4] can be used on top of tuple independent probabilistic databases for answering queries in TIP-OWL without changing the semantics of answers.

Query Rewriting Query processing in $DL\text{-}Lite_R$ is based on the idea of first-order reducibility. This means that for every conjunctive query q we can find a query q' that produces the same answers as q by just looking at the A-Box. Calvanese et al. also define a rewriting algorithm that computes a q' for every q by applying transformations that depend on the T-Box axioms.

Given a consistent T-Box $\mathcal{T}$ the algorithm takes a conjunctive query q_0 and expands it into a union of conjunctive queries U starting with $U = \{q_0\}$. The algorithm successively extends U by applying the following rule:

$$U = U \cup \{q[l/r(l, I)]\}$$

where q is a query in U, l is a literal in q, I is an inclusion axiom from $\mathcal{T}$ and r is a replacement function that is defined as follows:

l	**I**	$\mathbf{r}(l, I)$
$A(x)$	$A' \sqsubseteq A$	$A'(x)$
$A(x)$	$\exists P \sqsubseteq A$	$P(x, \_)$
$A(x)$	$\exists P^- \sqsubseteq A$	$P(\_, x)$
$P(x, \_)$	$A \sqsubseteq \exists P$	$A(x)$
$P(x, \_)$	$\exists P' \sqsubseteq \exists P$	$P'(x, \_)$
$P(x, \_)$	$\exists P'^- \sqsubseteq \exists P$	$P'(\_, x)$

l	**I**	$\mathbf{r}(l, I)$
$P(\_, x)$	$A \sqsubseteq \exists P^-$	$A(x)$
$P(\_, x)$	$\exists P' \sqsubseteq P^-$	$P'(x, \_)$
$P(\_, x)$	$\exists P'^- \sqsubseteq \exists P^-$	$P'(\_, x)$
$P(x, y)$	$P' \sqsubseteq P$	$P'(x, y)$
$P(x, y)$	$P'^- \sqsubseteq P^-$	$P'(x, y)$
$P(x, y)$	$P' \sqsubseteq P^-$	$P'(y, x)$
$P(x, y)$	$P'^- \sqsubseteq P$	$P'(y, x)$

Here '_' denotes an unbound variable, i.e., a variable that does not occur in any other literal of any of the queries.

Definition 2 (Derived Query). *Let $Q = L_1 \wedge \cdots \wedge L_m$ be a conjunctive query over a $DL\text{-}Lite$ terminology. We write $Q \overset{r}{\to} Q'$ if $Q' = Q \vee Q[L_i / r(L_i, I)]$ for some literal L_i from Q. Let $\overset{r}{\to}^*$ denote the transitive closure of $\overset{r}{\to}$, then we call every Q' with $Q \overset{r}{\to}^* Q'$ a derived query of Q. Q' is called maximal if there is no Q'' such that $Q \overset{r}{\to}^* Q''$ and $Q' \overset{r}{\to}^* Q''$.*

Using the notion of a maximal derived query, we can establish the FOL-reducibility of $DL\text{-}Lite$ by rephrasing the corresponding theorem from [4].

Theorem 1 (FOL-Reducibility of $DL\text{-}Lite$ (from [4])). *Query answering in $DL\text{-}Lite_R$ is FOL-reducible. In particular, for every query Q with maximal derived query Q' and every $DL\text{-}Lite_R$ T-Box $\mathcal{T}$ and non-empty A-Box $\mathcal{A}$ we have $\mathcal{T} \cup \mathcal{A} \models Q$ if and only if $\mathcal{A} \models Q'$.*

Correctness of Query Processing Implementing TIP-OWL on top of probabilistic databases can now be done in the following way. The A-Box is stored in the probabilistic database. It is easy to see that the probabilistic semantics of TIP-OWL A-Boxes and tuple independent databases coincide. What remains to be done is to show that the idea of FOL-reducibility carries over to our probabilistic model. In particular, we have to show that a rewritten query has the same probability given a knowledge base with empty T-Box as the original query given a complete knowledge base. This result is established in the following corollary.

Corollary 1. *Let $\mathcal{TKB} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ be a TIP-OWL knowledge base and $\mathcal{TKB}' = \langle \emptyset, \mathcal{A}, \mathcal{P} \rangle$ the same knowledge base, but with an empty T-Box. Let further Q be a conjunctive query and Q' a union of conjunctive queries obtained by rewriting Q on the basis of $\mathcal{T}$, then*

$$P(Q|\mathcal{TKB}) = P(Q'|\mathcal{TKB}')$$

We can easily see that the semantics of queries over a TIP-OWL A-Box with empty T-Box directly corresponds to the tuple-independence semantics used in probabilistic databases [15], thus queries posed to correctly constructed probabilistic database have the same probability as a TIP-OWL query with empty T-Box. From theorem 1 we get, that $P(Q|\mathcal{KB})$ does not change as $\mathcal{T}, \mathcal{A} \models Q$ if and only if $\mathcal{A} \models Q'$. Further, as $P(\mathcal{KB}|\mathcal{TKB})$ only depends on $\mathcal{A}$ this part also stays unchanged.

Over the past decade, the database community has developed efficient methods for querying uncertain information in probabilistic databases. Important results are the introduction of the independent tuple model for probabilistic data as well as a complete characterization of queries that can be computed in polynomial time. We briefly review these results in this section.

2.3 Extensional Query Processing

It has been shown that the key to efficient query processing in probabilistic databases is to avoid the computation of complex event descriptions and to directly compute the probability of a complex query from the probabilities of subqueries. This approach, referred to as extensional query processing, has been shown to correctly compute the probability of queries for the class of tractable queries using the following recursive algorithm:

Algorithm 1 Extensionally compute P(Q) (from [6])

```
Require: A conjunctive query Q in CNF, a tuple independent probabilistic database
Q is a conjunctive query in CNF: Compute symbol components $Q = Q_1 \wedge \cdots \wedge Q_m$
if $m \geq 2$ then
    return $\prod_{i=1,\cdots,m} P(Q_i)$
end if

$Q = Q_1 \wedge \cdots \wedge Q_k$ is a symbol-connected query in CNF:
if $k \geq 2$ then
    return $-\sum_{s \subseteq [n], s \neq \emptyset} (-1)^{|s|} P\left(\bigvee_{i \in s} Q_i\right)$
end if

Q is a disjunctive query: Compute symbol components $Q = Q_1 \vee \cdots \vee Q_m$
if $m \geq 2$ then
    return $1 - \left(\prod_{i=1,\cdots,n} 1 - P(Q_i)\right)$
end if

$Q = Q_1 \vee \cdots \vee Q_k$ is a symbol-connected disjunctive query:
if $Q = t$ has no variables then
    return $P(t)$
end if

if $Q$ has a separator variable z then
    return $1 - \left(\prod_{a \in ADom} 1 - P(Q[a/x])\right)$
else
    return false
end if
```

Symbol components are defined as follows [15]: Let $Q = d_1 \wedge \ldots \wedge d_k$ be a UCQ in CNF, and $K_1, \ldots, K_m$ the connected components for the set of queries $\{d_1, \ldots, d_k\}$. The symbol-components of Q are $Q1 = \bigwedge_{i \in K_1} d_i$, …, $Q_m = \bigwedge_{i \in K_m} d_i$. Then the probability $P(Q) = P(Q_1) \cdot \ldots \cdot P(Q_m)$. If $m = 1$, then Q is *symbol-connected.*

Each recursion of the algorithm processes a simpler subexpression, until the probability of an individual can be read directly from the database. Queries that can be completely processed using the steps above are called *safe*. It has been shown that safe queries exactly correspond to queries that can be computed in PTIME whereas queries that cannot be completely processed using these steps – in particular queries for which we cannot find a separator variable in line 19 – are in #P-hard. This means that we can use the notion of safeness and the processing steps above to analyze the general complexity of certain classes of queries.

Definition 3 (Safeness). *A query Q is called safe if and only if it can be computed by iteratively applying Algorithm 1.*

Theorem 2 (Dichotomy Theorem (adapted from [15])). *The probability of safe queries can be computed in PTIME. If a query is not safe computing its probability is in #P-hard.*

Jung et al. proved that Theorem 2 also holds for probabilistic OBDA in OWL 2 QL [10]. This essentially means that the same dichotomy of safe and unsafe queries still applies under the presence of reasoning through query rewriting.

3 Analysis of the SPARQL Dataset and Query Safeness

For our analysis we used the queries collected in the Linked SPARQL Queries Dataset (LSQ) [13]. LSQ contains queries which where posed against the public SPARQL endpoints of DBPedia[1], Linked Geo Data (LGD)[2], Semantic Web Dog Food (SWDF)[3], and the British Museum (BM)[4]. Although the queries in the dataset are not posed against probabilistic data, we argue that the large number of queries gives a good picture of the general information needs users have. In the case of DBpedia, there is actually some amount of uncertainty in its generation, as the data is automatically extracted from Wikipedia, which itself can uncertain/wrong information.

Table 1. Overview of queries in the LSQ dataset

	SELECT	ASK	DESCRIBE	CONSTRUCT	Total
SWDF	58 741	157	26 533	52	85 483
DBpedia	736 726	37 174	777	7 613	782 290
LGD	265 410	25 140	24	7 004	297 578
BM	29 073	0	0	0	29 073
Total	1 089 950	62 471	27 334	14 669	1 194 424

Table 1 shows the different query types and their distribution in the different datasets. Those numbers contain only queries that parse successfully; LSQ also lists queries with parse errors. They clearly show that the main amount of queries are SELECT queries. Interestingly, the dataset for the British Museum's endpoint only consists of SELECT queries; all being uniformly structured. This implies that those are not actual user queries but automatically created ones of some system. However, the queries from the British Museum make up only a small part of the whole LSQ dataset. @

[1] http://dbpedia.org/

[2] http://linkedgeodata.org/

[3] http://data.semanticweb.org/

[4] http://bm.rkbexplorer.com

Table 2. Used SPARQL features

	UNION	FILTER	DISTINCT	GROUP BY	EXISTS
BM	0	0	29073	0	0
DBpedia	36127	183882	144245	3	3
LGD	28827	92558	66206	11	0
SWDF	27982	818	39000	445	123
Total	92936	277258	278524	459	126

Looking at the used SPARQL features, FILTER and DISTINCT are the most prominent, with UNION coming in second. GROUP BY and EXISTS play only a minor role. However, note that DISTINCT is technically a GROUP BY over all variables in the SELECT clause.

The focus of our analysis is on SELECT and ASK queries. For DESCRIBE queries there is no formal definition of how a result is produced, thus making it impossible to determine if the process is safe; whereas CONSTRUCT queries do not directly translate to query answering as it generates a new graph and not a result set. Furthermore, our current implementation cannot analyze queries containing features, like GROUP BY, DISTINCT, EXISTS, OPTIONAL, and endpoint specific functions not part of the SPARQL recommendation (e.g. Oracle's version of COUNT). After removing those we are left with 507 241 queries for which we can determine query safeness.

We used parts of Ontop [3], a system for deterministic OBDA, to translate the SPARQL queries to (union of) conjunctive queries. Those queries were then processed by Algorithm 1 to determine their safeness. The implementation is written in Java and the experiments were run on an Intel Core i5@2.9 GHz with 4GB of RAM for the Java VM.

The distribution of safe and unsafe queries is shown in Table 3. With 99.68%, almost all of the *analyzed* queries are safe. Regarding the total number of 1 194 424 queries, 42.33% are definitly safe. Note that this does not imply that 57.67% of the queries are unsafe. Those queries need a more thorough analysis because of their usage of DISTINCT/GROUP BY, FILTER, HAVING, EXISTS, or OPTIONAL.

Looking at the numbers for the different data sources, the percent of unsafe queries sent to the Semantic Web Dog Food endpoint is significantly higher than for the other two. Analyzing those queries we found the major number being of the form `SELECT ?targetType WHERE { ?obj a <someURI>. ?obj <someOtherURI> ?targetObj. ?targetObj a ?targetType.}`. This suggest that those query are generate by some tool and not by a user. Overall, this shows that the users' general information needs can also be satisfied over probabilistic data in a tractable way.

Our implementation of the algorithm is still quite simple and we are not able to simplify all queries. Thus determining their safeness becomes sometimes unfeasible. This results in a timeout for 69 queries after 10 minutes. Overall, the average processing time for a query is 86ms.

Table 3. Analysis results for queries in the LSQ dataset

	DBpedia		LGD		SWDF		Total	
	#	%	#	%	#	%	#	%
Safe	287 487	99,72	199 636	99,79	18 240	97,93	505 563	99,68
Unsafe	809	0,28	414	0,21	386	2,07	1 609	0,32
Total	288 296		200 050		18 626		507 172	

4 Related Work

There is some other work analyzing the characteristics of deterministic SPARQL queries from different sources. To the best of our knowledge, there is no analysis of the safeness of real-world SPARQL queries for probabilistic data.

In [11] Picalausa et al. analyzed 3 million queries from DBpedia query logs. They come to the conclusion that the majority of the queries therein are tractable for the deterministic case.

Arias et al. [8] observe that most real-world SPARQL queries (USEWOD2011 dataset [2] containing queries from DBPedia and Semantic Web Dog Food) are star-shaped and do not contain property chains. As Jung et al. [10] have shown that tractable queries are generally star-shaped, their findings also coincide with our results.

Han et al. [9] also analyzed queries in LSQ. Similarly to Picalausa et al. they come to the conclusion that most queries are in PTIME. They also show that most queries have a simple structure consisting of three or less triple patterns, making it less probable for them having a structure that is unsafe. An unsafe query must at least consist of either two triple patterns with a self-join (the same predicate), or three triple patterns (cf. unsafe queries given in [15]).

5 Conclusion and Future Work

In this paper we analyzed half a million queries from the LSQ dataset; those which can directly be translate to (union of) conjunctive queries. Checking for probabilistic query safeness we found nearly all queries to be safe and thus tractable when processed over probabilistic data. Relating to the whole set of queries, about half of them are safe; for the other half there is no information in our analysis. This shows that the general information needs a user of the public SPARQL endpoints serving as sources for the LSQ dataset has are also feasible to be fulfilled on uncertain data. Note that there is no uncertain version of those dataset. However, we argue that the users' information needs would not change over uncertain data, and that it is possible to create such a version of DBpedia, e.g. by incorporating information about trustworthiness in general or knowledge about how commonly a certain property is accurately extracted. To the best of our knowledge this is the first work that analyzes real-world SPARQL queries in the light of probabilistic data.

For future work, we have two different directions. First, we want to further study queries using constructs like DISTINCT/GROUP BY or OPTIONAL and how they influence query safeness. Second, considering unsafe queries, we want to analyze their structure and investigate ways of handling those effectively, e.g. through approximation or simplification similarly to approaches present in probabilistic databases.

References

1. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite Family and Relations. Journal of Artificial Intelligence Research 36, 1–69 (2009)
2. Berendt, B., Hollink, L., Hollink, V., Luczak-Rösch, M., Möller, K., Vallet, D.: USEWOD2011 - 1st International Workshop on Usage Analysis and the Web of Data. Proceedings of the 20th International Conference Companion on World Wide Web, WWW 2011 pp. 305–306 (2011), http://www.scopus.com/inward/record.url?eid=2-s2.0-79955127036&partnerID=40&md5=6ec8c19cff2e223d891c92e7c6e236e7
3. Calvanese, D., Cogrel, B., Komla-ebri, S., Kontchakov, R., Lanti, D.: Ontop : Answering SPARQL Queries over Relational Databases. Semantic Web journal 0(0) (2015), http://www.semantic-web-journal.net/system/files/swj1004.pdf
4. Calvanese, D., Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. Journal of Automated Reasoning 39(3), 385–429 (jul 2007), http://link.springer.com/10.1007/s10817-007-9078-x
5. Carlson, A., Betteridge, J., Kisiel, B.: Toward an Architecture for Never-Ending Language Learning. In Proceedings of the Conference on Artificial Intelligence (AAAI) (2010) pp. 1306–1313 (2010), http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/download/1879/2201
6. Dalvi, N., Suciu, D.: The Dichotomy of Probabilistic Inference for Unions of Conjunctive Queries. Journal of the ACM 59(6), 1–87 (2012), http://dl.acm.org/citation.cfm?doid=2395116.2395119
7. De Raedt, L., Kimmig, A., Toivonen, H.: ProbLog: A Probabilistic Prolog and its Application in Link Discovery. In: IJCAI International Joint Conference on Artificial Intelligence. pp. 2468–2473 (2007)
8. Gallego, Mario Arias Fernández, Javier D. Martínez-Prieto, M.A., de la Fuente, P.: An Empirical Study of Real-World SPARQL Queries. In: 1st International Workshop on Usage Analysis and the Web of Data (USEWOD2011) at the 20th International World Wide Web Conference (WWW 2011). pp. 10–13 (2011), http://arxiv.org/abs/1103.5043
9. Han, X., Feng, Z., Zhang, X., Wang, X., Rao, G., Jiang, S.: On the Statistical Analysis of Practical SPARQL Queries. arXiv preprint arXiv:1603.06729 p. 6 (2016), http://arxiv.org/abs/1603.06729
10. Jung, J.C., Lutz, C.: Ontology-Based Access to Probabilistic Data with OWL QL. In: The Semantic Web - ISWC 2012. pp. 182—-197. Springer (2012)
11. Picalausa, F., Vansummeren, S.: What are Real SPARQL Queries Like? Proceedings of the International Workshop on Semantic Web Information Management pp. 1–6 (2011), http://portal.acm.org/citation.cfm?doid=1999299.1999306
12. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R.: Probabilistic Description Logics under the Distribution Semantics. Semantic Web journal (2014), http://www.semantic-web-journal.net/system/files/swj651.pdf
13. Saleem, M., Ali, M.I., Hogan, A., Mehmood, Q., Ngonga Ngomo, A.C.: LSQ: The Linked SPARQL Queries Dataset. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9367, 261–269 (2015)
14. Schoenfisch, J., Stuckenschmidt, H.: Towards Large-Scale Probabilistic OBDA. In: Beierle, C., Dekhtyar, A. (eds.) Scalable Uncertainty Management: 9th International Conference,

SUM 2015, Québec City, QC, Canada, September 16-18, 2015. Proceedings, pp. 106—-120. Springer International Publishing, Cham (2015), http://dx.doi.org/10.1007/978-3-319-23540-0_8

15. Suciu, D., Olteanu, D., Ré, C., Koch, C.: Probabilistic Databases. Morgan & Claypool Publishers (2011)