SPARQL with Property Paths on the Web

Editor(s): Fabien Gandon, INRIA, France; Marta Sabou, Technische Universität Wien, Austria; Harald Sack, Hasso Plattner Institute, Germany
Solicited review(s): Pedro Szekely, University of Southern California, USA; Jérôme Euzenat, INRIA Grenoble Rhône-Alpes, France; Oscar Corcho, Universidad Politécnica de Madrid, Spain

Olaf Hartig [a,b,*], Giuseppe Pirrò [c]
[a] *Hasso Plattner Institute, Universität Potsdam, Germany*
[b] *Department of Computer and Information Science (IDA), Linköping University, Sweden*
E-mail: olaf.hartig@liu.se
[c] *Italian National Research Council (ICAR-CNR), Rende(CS), Italy*
E-mail: pirro@icar.cnr.it

Abstract. Linked Data on the Web represents an immense source of knowledge suitable to be automatically processed and queried. In this respect, there are different approaches for Linked Data querying that differ on the degree of centralization adopted. On one hand, the SPARQL query language, originally defined for querying single datasets, has been enhanced with features to query federations of datasets; however, this attempt is not sufficient to cope with the distributed nature of data sources available as Linked Data. On the other hand, extensions or variations of SPARQL aim to find trade-offs between centralized and fully distributed querying. The idea is to partially move the computational load from the servers to the clients. Despite the variety and the relative merits of these approaches, as of today, there is no standard language for querying Linked Data on the Web. A specific requirement for such a language to capture the distributed, graph-like nature of Linked Data sources on the Web is a support of graph navigation. Recently, SPARQL has been extended with a navigational feature called *property paths* (PPs). However, the semantics of SPARQL restricts the scope of navigation via PPs to *single* RDF graphs. This restriction limits the applicability of PPs for querying distributed Linked Data sources on the Web. To fill this gap, in this paper we provide formal foundations for evaluating PPs on the Web, thus contributing to the definition of a query language for Linked Data. We first introduce a family of reachability-based query semantics for PPs that distinguish between navigation on the Web and navigation at the data level. Thereafter, we consider another, alternative query semantics that couples Web graph navigation and data level navigation; we call it context-based semantics. Given these semantics, we find that for some PP-based SPARQL queries a complete evaluation on the Web is not possible. To study this phenomenon we introduce a notion of Web-safeness of queries, and prove a decidable syntactic property that enables systems to identify queries that are Web-safe. In addition to establishing these formal foundations, we conducted an experimental comparison of the context-based semantics and a reachability-based semantics. Our experiments show that when evaluating a PP-based query under the context-based semantics one experiences a significantly smaller number of dereferencing operations, but the computed query result may contain less solutions.

Keywords: Property paths, Web navigational language, Web safeness, SPARQL

1. Introduction

The increasing trend in sharing and interlinking pieces of structured data on the World Wide Web (WWW) is evolving the classical Web—which is focused on hypertext documents and syntactic links among them—into a Web of Linked Data. The Linked Data principles [5] present an approach to extend the scope of Uniform Resource Identifiers (URIs) to new types of resources (e.g., people, places) and represent their descriptions and interlinks by using the Resource Description Framework (RDF) [8] as standard data format. RDF adopts a graph-based data model, which can be queried by using the SPARQL query language [15]. When it comes to Linked Data on the WWW, the common way to provide query-based ac-

[*] Corresponding author, e-mail: olaf.hartig@liu.se

cess is via SPARQL endpoints; that is, services that usually answer SPARQL queries over a single dataset. Recently, the original core of SPARQL has been extended with features supporting query federation; it is now possible, within a single query, to target multiple endpoints (via the SERVICE operator). However, such an extension is not enough to cope with an unbounded and a priori unknown space of data sources such as the WWW. Moreover, not all Linked Data on the WWW is accessible via SPARQL endpoints. More recent proposals are based on the idea of Linked Data Fragments [39,40] and aim at moving part of the computational load from Web servers to clients.

However, as of today, there exists no standard query language for Linked Data on the WWW, although SPARQL is clearly a candidate. A key feature that such a language should provide is navigation across the unbound, a priori unknown, graph-like environment represented by distributed Linked Data sources.

While earlier research on using SPARQL for Linked Data is limited to fragments of the first version of the language [6,16,18,38], the version 1.1 of SPARQL introduces a feature called *property paths* (PPs) that equips the language with navigational capabilities [15]. However, the standard definition of PPs is limited to single RDF graphs and, thus, not directly applicable to Linked Data that is distributed over the WWW.

Therefore, toward the definition of a language for accessing Linked Data live on the WWW, the following questions emerge naturally:

How can PPs be defined over the WWW?

and

What are the implications of such a definition?

Answering these questions is the broad objective of this paper. In particular, we focus on *Linked Data on the WWW*, by which we mean RDF data that is made available on the WWW as per the Linked Data principles [5] and, thus, can be accessed by looking up HTTP scheme based URIs. In this context we make the following main contributions:

1. We formalize a family of *reachability-based* query semantics of PP-based SPARQL queries that are meant to be evaluated over Linked Data on the WWW. This formalization approach treats navigation on the Web separate from navigation on the level of data.
2. We also formalize an alternative, *context-based* query semantics that intertwines Web graph navigation and data level navigation.
3. We study the feasibility of evaluating queries under these semantics. For this study we assume that query engines do not have complete information about the queried Web of Linked Data (as it is the case for the WWW). Our study shows that query evaluation under any reachability-based semantics is possible in practice and that a similarly general statement cannot be made for the context-based semantics; that is, there exist cases in which query evaluation under the context-based semantics is not possible.
4. We establish a decidable syntactic property of queries for which an evaluation under the context-based semantics is possible.
5. We provide an experimental comparison of the context-based and a reachability-based semantics. For this comparison we executed queries directly over the WWW. As its main result, our experiment shows that when evaluating a PP-based query under the context-based semantics, one experiences a significantly smaller number of dereferencing operations, but the computed query result may contain less solutions.

This article extends a preliminary version that appeared in the proceedings of the ESWC 2015 conference [21]. The extension includes: (i) the definition and analysis of a family of reachability-based query semantics for Property Paths on the Web; (ii) an experimental analysis and comparison of the different semantics; (iii) a more detailed description of the main technical results; (iv) further examples to better clarify the terminology and the main concepts of the paper; (v) a more comprehensive discussion of related work.

The paper is organized as follows. Section 2 provides an overview on related work. In Section 3 we introduce the formal framework for this paper, including a data model that captures the notion of Linked Data on the WWW. Section 4 focuses on PPs, isolated from other SPARQL operators. In Section 5 we broaden our view to define PP-based SPARQL graph patterns. In Section 6 we characterize a class of *Web-safe* patterns and prove their feasibility. Section 7 discusses the experimental evaluation. Finally, in Section 8 we conclude.

2. Related Work

There is an extensive body of research on the foundations of querying RDF data. An important work in this context is the investigation of SPARQL provided by Peréz et al. [30]. Other authors focused on the foundations of SPARQL query optimization [34,26].

From the perspective of graphs, languages for the *navigation and specification* of vertices in graphs have a long tradition (see Wood's survey [41]). For RDF, extensions of SPARQL such as PSPARQL [2], nSPARQL [31], and SPARQLeR [23] introduced navigational features since those were missing in the first version of SPARQL. Only recently, with the addition of *property paths* (PPs) in version 1.1 [15], SPARQL has been enhanced officially with such features. The final definition of PPs has been influenced by research that studied the computational complexity of an early draft version of PPs [3,27]. There also already exists a proposal to extend the expressive power of PPs [11]. Other strands of research focus on studying properties of PPs such as containment [25] or supporting recursion in SPARQL [32]. However, the main assumption of all these navigational extensions of SPARQL is to work on a single, centralized RDF graph.

The idea of querying the WWW as a database is not new (see Florescu et al.'s survey [13]). Perhaps the most notable early works in this context are by Konopnicki and Shmueli [24], Abiteboul and Vianu [1], and Mendelzon et al. [28], all of which tackled the problem of evaluating SQL-like queries on the hypertext Web. While such queries included navigational features, the focus was on retrieving specific Web pages, particular attributes of specific pages, or content within them.

Our departure point is different: *We aim at defining semantics of SPARQL queries (including property paths) over Linked Data on the WWW;* this involves dealing with two graphs of different type; namely, an RDF graph that is distributed over an unbounded number of documents on the WWW and the Web graph in which these documents are interlinked with each other.

To express queries over Linked Data on the WWW, two main strands of research can be identified. The first studies how to extend the scope of SPARQL queries to the WWW, with existing work focusing on basic graph patterns [6,16,38] or a more expressive fragment that includes AND, OPT, UNION and FILTER [18]. The second strand of research focuses on emphasizing navigational features, which resulted in new languages such as NautiLOD [10,12], LDPath [33], and LDQL [20].

These two strands have different departure points. The former employs navigation over the WWW to collect data for answering a given SPARQL query; here navigation is a means to discover query-relevant data. The latter provides explicit navigational features and uses querying capabilities to filter data sources of interest; here navigation (not querying) is the main focus. The context-based query semantics proposed in this paper combines both approaches.

Another line of research slightly related to our proposal is that of *focused crawling*. The idea is to enhance the behavior of classical Web crawlers, that consider all pages reachable from a given page, to be more selective; selectivity is obtained by considering e.g., a set of predefined topics [36] or meta data within HTML pages [29]. A more recent line of related research looks into building (domain-specific) knowledge graphs by exploiting semantic technologies to reconcile the data continuously crawled from diverse sources [35]. In a way, these approaches mimic the process of filtering performed by our approach but on a less expressive scale due to the limited expressiveness of the filtering mechanism as compared to our language. Nevertheless, our approach could be used to enable a finer-grained information filtering.

3. Formal Framework

This section provides a formal framework for defining semantics of PPs over Linked Data. In particular, we first recall the definition of PPs as per the SPARQL standard [15]. Thereafter, we introduce a data model that captures the notion of Linked Data on the WWW.

3.1. Preliminaries

We assume four pairwise disjoint, countably infinite sets $\mathcal{I}$ (IRIs), $\mathcal{B}$ (blank nodes), $\mathcal{L}$ (literals), and $\mathcal{V}$ (variables, denoted by a leading '?' symbol). An *RDF triple* (or simply *triple*) is a tuple from the set $\mathcal{T} = (\mathcal{I} \cup \mathcal{B}) \times \mathcal{I} \times (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$. For any such triple $t = \langle s, p, o \rangle$ we call s the *subject*, p the *predicate*, and o the *object*, and we write $\mathrm{iris}(t)$ to denote the set of all IRIs in the triple; i.e., $\mathrm{iris}(t) = \{s, p, o\} \cap \mathcal{I}$. A set of triples is called an *RDF graph*.

A *property path pattern* (or *PP pattern* for short) is a tuple $P = \langle \alpha, \mathtt{path}, \beta \rangle$ with $\alpha \in (\mathcal{I} \cup \mathcal{L} \cup \mathcal{V})$, $\beta \in (\mathcal{I} \cup \mathcal{L} \cup \mathcal{V})$, and $\mathtt{path}$ is a *property path expression* (*PP expression*) that is defined by the following grammar (where $u, u_1, \ldots, u_n \in \mathcal{I}$):

$$\begin{aligned} \mathtt{path} = u \;\big|\; !(u_1 \,|\, \ldots \,|\, u_n) \;\big|\; \mathtt{path/path} \;\big|\\ (\mathtt{path}\,|\,\mathtt{path}) \;\big|\; (\mathtt{path})^* \;\big|\; \hat{}\mathtt{path} \end{aligned}$$

As can be seen from this grammar, we have two base cases for PP expressions, namely, arbitrary IRIs

$M_1 \bowtie M_2 = \langle \Omega, \mathit{card} \rangle$ such that $\Omega = \{\, \mu_1 \cup \mu_2 \mid (\mu_1, \mu_2) \in \Omega_1 \times \Omega_2$ and $\mu_1 \sim \mu_2 \}$ and for every solution mapping $\mu \in \Omega$ we have $\mathit{card}(\mu) = \sum_{(\mu_1,\, \mu_2)\, \in\, \Omega_1 \times \Omega_2 \text{ s.t. } \mu = \mu_1 \cup \mu_2} (\mathit{card}(\mu_1) \cdot \mathit{card}(\mu_2))$.

$M_1 \setminus M_2 = \langle \Omega, \mathit{card} \rangle$ such that $\Omega = \{\, \mu_1 \in \Omega_1 \mid \nexists \mu_2 \in \Omega_2 : \mu_1 \sim \mu_2 \}$ and for every solution mapping $\mu \in \Omega$ we have $\mathit{card}(\mu) = \mathit{card}_1(\mu)$.

$M_1 \sqcup M_2 = \langle \Omega, \mathit{card} \rangle$ such that $\Omega = \Omega_1 \cup \Omega_2$ and (i) $\mathit{card}(\mu) = \mathit{card}_1(\mu)$ for all solution mappings $\mu \in \Omega \setminus \Omega_2$, (ii) $\mathit{card}(\mu) = \mathit{card}_2(\mu)$ for all $\mu \in \Omega \setminus \Omega_1$, and (iii) $\mathit{card}(\mu) = \mathit{card}_1(\mu) + \mathit{card}_2(\mu)$ for all $\mu \in \Omega_1 \cap \Omega_2$.

$\pi_V(M_1) = \langle \Omega, \mathit{card} \rangle$ such that $\Omega = \{\mu \mid \exists \mu' \in \Omega_1 : \mu \sim \mu'$ and $\mathrm{dom}(\mu) = V \cap \mathrm{dom}(\mu')\}$ and for every solution mapping $\mu \in \Omega$ we have $\mathit{card}(\mu) = \sum_{\mu' \in \Omega_1 \text{ s.t. } \mu \sim \mu'} \mathit{card}_1(\mu')$.

Fig. 1. SPARQL algebra operators over multisets of solution mappings, $M_1 = \langle \Omega_1, \mathit{card}_1 \rangle$ and $M_2 = \langle \Omega_2, \mathit{card}_2 \rangle$.

and expressions of the form $!(u_1 \mid \ldots \mid u_n)$. PP patterns based on the former are ordinary triple patterns, which, in the context of PPs, represent single navigation steps from the subject to the object of any triple whose predicate is the given IRI. The second base case captures a form of negation that represents a navigation step along any triple whose predicate is *not* among the IRIs listed. Given these base types of PP expressions, users may combine them via the classical regular expression operators: concatenation $/$, disjunction $|$, and recursive concatenation $(\cdot)^*$; additionally, $\hat{}\texttt{path}$ represents the inverse of `path` (a formal semantics of PP patterns and PP expressions follows shortly).

The SPARQL standard introduces additional types of PP expressions [15]. Since these are merely syntactic sugar (they are defined in terms of expressions covered by the grammar given above), we ignore them in this paper. As another slight deviation from the standard, we do not permit blank nodes in PP patterns (i.e., $\alpha, \beta \notin \mathcal{B}$). However, standard PP patterns with blank nodes can be simulated using fresh variables.

Example 1. *As an example of a PP pattern consider* $\langle \mathsf{Tim}, (\mathsf{knows})^*/\mathsf{name}, ?n \rangle$ *where* $?n \in \mathcal{V}$ *and* $\mathsf{Tim}, \mathsf{knows}, \mathsf{name} \in \mathcal{I}$*. This pattern retrieves the names of persons that can be reached from* Tim *by an arbitrarily long path of* knows *relationships (which includes* Tim*). Another example are the two PP patterns* $\langle ?p, \mathsf{knows}, \mathsf{Tim} \rangle$ *and* $\langle \mathsf{Tim}, \hat{}\mathsf{knows}, ?p \rangle$*, both of which retrieve persons that know* Tim*. For further examples we refer to the SPARQL specification [15, Section 9.2].*

In addition to a syntax for the queries of interest, we have to introduce the standard semantics of these queries. The SPARQL specification defines this semantics by an evaluation function (see below) that returns multisets of so called *solution mapping*s; such a mapping is a partial function $\mu : \mathcal{V} \to (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$.

To refer to the domain of a solution mapping μ (i.e., the set of variables for which μ is defined) we write $\mathrm{dom}(\mu)$. If, for two solution mappings, say μ_1 and μ_2, we have $\mu_1(?v) = \mu_2(?v)$ for every variable $?v \in \big(\mathrm{dom}(\mu_1) \cap \mathrm{dom}(\mu_2)\big)$, then we say that μ_1 and μ_2 are *compatible* ($\mu_1 {\sim} \mu_2$). In this case, μ_1 and μ_2 can be combined into a solution mapping $\mu = \mu_1 \cup \mu_2$ such that $\mathrm{dom}(\mu) = \big(\mathrm{dom}(\mu_1) \cup \mathrm{dom}(\mu_2)\big)$, $\mu \sim \mu_1$, and $\mu \sim \mu_2$. Given a solution mapping μ and a PP pattern P, we write $\mu[P]$ to denote the PP pattern obtained by replacing the variables in P according to μ (where variables for which μ is not defined are not replaced).

We represent a *multiset* of solution mappings by a pair $M = \langle \Omega, \mathit{card} \rangle$ where Ω is the underlying set (of solution mappings) and *card* is the corresponding *cardinality function*; i.e., $\mathit{card} : \Omega \to \{1, 2, ...\}$. By abusing notation slightly, we write $\mu \in M$ for every $\mu \in \Omega$. Furthermore, to simplify the following definitions we introduce a family of special, parameterized cardinality functions for multisets in which every solution mapping has a cardinality of 1. That is, for any set of solution mappings Ω, let $\mathsf{card1}^{(\Omega)} : \Omega \to \{1, 2, ...\}$ be the *constant-1 cardinality function* that is defined by $\mathsf{card1}^{(\Omega)}(\mu) = 1$ for all $\mu \in \Omega$.

To define the aforementioned evaluation function we also need to introduce several operators of the SPARQL algebra, which is defined over multisets of solution mappings. That is, for two such multisets, $M_1 = \langle \Omega_1, \mathit{card}_1 \rangle$ and $M_2 = \langle \Omega_2, \mathit{card}_2 \rangle$, we define the join ($\bowtie$), the difference ($\setminus$), the multiset union ($\sqcup$), and projection ($\pi_V$, where $V \subseteq \mathcal{V}$ is a finite set of variables) as given in Figure 1. In addition to these algebra operators, the SPARQL standard introduces auxiliary functions to define the semantics of PP patterns of the form $\langle \alpha, \mathtt{path}^*, \beta \rangle$. Figure 2 provides these functions—which we call `ALP1` and `ALP2`—adapted to our formalism (we need a variable $?x$ in line 6 since PP patterns in our formalism do not have blank nodes).

We are now ready to define the evaluation function that formalizes the standard semantics of PP patterns.

```
Function ALP1(γ, path, G)
Input: $\gamma \in (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$,
       path is a PP expression,
       $G$ is an RDF graph.
Visited := ∅
ALP2(γ, path, Visited, G)
return Visited
```

```
Function ALP2(γ, path, Visited, G)
Input: $\gamma \in (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$,  path is a PP expression,
       $Visited \subseteq (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$,  $G$ is an RDF graph.
if $\gamma \notin Visited$ then
    add $\gamma$ to Visited
    for all $\mu \in [\![\langle ?x, \mathtt{path}, ?y \rangle]\!]_G$ such that $\mu(?x) = \gamma$ and $?x, ?y \in \mathcal{V}$ do
        ALP2($\mu(?y)$, path, Visited, $G$)
```

Fig. 2. Auxiliary functions used for defining the semantics of PP expressions of the form path*.

$$
\begin{aligned}
[\![\langle \alpha, u, \beta \rangle]\!]_G &= \Big\langle \big\{ \mu \mid \mathrm{dom}(\mu) = (\{\alpha, \beta\} \cap \mathcal{V}) \text{ and } \mu[\langle \alpha, u, \beta \rangle] \in G \big\}, \ \ \mathsf{card1}^{(\Omega)} \Big\rangle \\
[\![\langle \alpha, !(u_1 \mid \ldots \mid u_n), \beta \rangle]\!]_G &= \Big\langle \big\{ \mu \mid \mathrm{dom}(\mu) = \big(\{\alpha, \beta\} \cap \mathcal{V}\big) \text{ and there exists an IRI} \\
&\qquad u \in \mathcal{I} \text{ such that } u \notin \{u_1, \ldots, u_n\} \text{ and } \mu[\langle \alpha, u, \beta \rangle] \in G \big\}, \ \ \mathsf{card1}^{(\Omega)} \Big\rangle \\
[\![\langle \alpha, \hat{\ }\mathtt{path}, \beta \rangle]\!]_G &= [\![\langle \beta, \mathtt{path}, \alpha \rangle]\!]_G \\
[\![\langle \alpha, \mathtt{path}_1/\mathtt{path}_2, \beta \rangle]\!]_G &= \pi_{\{\alpha,\beta\} \cap \mathcal{V}}\Big([\![\langle \alpha, \mathtt{path}_1, ?v \rangle]\!]_G \bowtie [\![\langle ?v, \mathtt{path}_2, \beta \rangle]\!]_G \Big) \\
[\![\langle \alpha, (\mathtt{path}_1 | \mathtt{path}_2), \beta \rangle]\!]_G &= [\![\langle \alpha, \mathtt{path}_1, \beta \rangle]\!]_G \sqcup [\![\langle \alpha, \mathtt{path}_2, \beta \rangle]\!]_G \\
[\![\langle x_{\mathrm{L}}, (\mathtt{path})^*, ?v_{\mathrm{R}} \rangle]\!]_G &= \Big\langle \big\{ \mu \mid \mathrm{dom}(\mu) = \{?v_{\mathrm{R}}\} \text{ and } \mu(?v_{\mathrm{R}}) \in \mathtt{ALP1}(x_{\mathrm{L}}, \mathtt{path}, G) \big\}, \ \ \mathsf{card1}^{(\Omega)} \Big\rangle \\
[\![\langle ?v_{\mathrm{L}}, (\mathtt{path})^*, ?v_{\mathrm{R}} \rangle]\!]_G &= \Big\langle \big\{ \mu \mid \mathrm{dom}(\mu) = \{?v_{\mathrm{L}}, ?v_{\mathrm{R}}\} \text{ and } \mu(?v_{\mathrm{L}}) \in \mathrm{terms}(G) \\
&\qquad \text{and } \mu(?v_{\mathrm{R}}) \in \mathtt{ALP1}(\mu(?v_{\mathrm{L}}), \mathtt{path}, G) \big\}, \ \ \mathsf{card1}^{(\Omega)} \Big\rangle \\
[\![\langle ?v_{\mathrm{L}}, (\mathtt{path})^*, x_{\mathrm{R}} \rangle]\!]_G &= [\![\langle x_{\mathrm{R}}, (\hat{\ }\mathtt{path})^*, ?v_{\mathrm{L}} \rangle]\!]_G \\
[\![\langle x_{\mathrm{L}}, (\mathtt{path})^*, x_{\mathrm{R}} \rangle]\!]_G &= \Big\langle \begin{cases} \{\mu_\emptyset\} & \text{if } \exists\, \mu \in [\![\langle x_{\mathrm{L}}, (\mathtt{path})^*, ?v \rangle]\!]_G : \mu(?v) = x_{\mathrm{R}}, \\ \emptyset & \text{else} \end{cases}, \ \ \mathsf{card1}^{(\Omega)} \Big\rangle
\end{aligned}
$$

Fig. 3. Standard query semantics of SPARQL Property Paths, where $\alpha, \beta \in (\mathcal{I} \cup \mathcal{L} \cup \mathcal{V})$; $u, u_1, ..., u_n \in \mathcal{I}$; $x_{\mathrm{L}}, x_{\mathrm{R}} \in (\mathcal{I} \cup \mathcal{L})$; $?v_{\mathrm{L}}, ?v_{\mathrm{R}} \in \mathcal{V}$; $?v \in \mathcal{V}$ is a fresh variable; and $\mu_\emptyset$ is the empty solution mapping with $\mathrm{dom}(\mu_\emptyset) = \emptyset$.

Definition 2. *Let P be a PP pattern and let G be an RDF graph. The* evaluation *of P over G, denoted by $[\![P]\!]_G$, is a multiset of solution mappings $\langle \Omega, card \rangle$ that is defined recursively as given in Figure 3.*

Example 3. *Consider the following RDF graph:*

$$
\begin{aligned}
G_{\mathsf{ex}} = \big\{ &\langle \mathsf{Suzi}, \mathsf{knows}, \mathsf{Eve} \rangle, \ \langle \mathsf{Eve}, \mathsf{knows}, \mathsf{Charlie} \rangle, \\
&\langle \mathsf{Suzi}, \mathsf{knows}, \mathsf{Alice} \rangle, \langle \mathsf{Alice}, \mathsf{knows}, \mathsf{Charlie} \rangle, \\
&\langle \mathsf{Alice}, \mathsf{knows}, \mathsf{Eve} \rangle \big\}.
\end{aligned}
$$

Then, for the PP pattern $P_a = \langle \mathsf{Suzi}, \mathsf{knows/knows}, ?x \rangle$ we have $[\![P_a]\!]_{G_{\mathsf{ex}}} = \langle \Omega_a, card_a \rangle$ with $\Omega_a = \{\mu_{a1}, \mu_{a2}\}$,

$$
\begin{aligned}
\mu_{a1}(?x) &= \mathsf{Charlie} && \textit{where } card_a(\mu_{a1}) = 2, \textit{ and} \\
\mu_{a2}(?x) &= \mathsf{Eve} && \textit{where } card_a(\mu_{a2}) = 1.
\end{aligned}
$$

Note that the result contains the solution mapping μ_{a1} twice because Charlie *can be reached from* Suzi *by two different paths that match the PP expression* knows/knows *(namely, one via* Eve*, the other via* Alice*).*

Example 4. *As another example, consider PP pattern $P_b = \langle \mathsf{Suzi}, (\mathsf{knows})^*, ?x \rangle$, for which we have:*

$$
[\![P_b]\!]_{G_{\mathsf{ex}}} = \langle \{\mu_{b1}, \mu_{b2}, \mu_{b3}, \mu_{b4}\}, card_b \rangle, \textit{ where}
$$

$$
\begin{aligned}
\mu_{b1}(?x) &= \mathsf{Suzi}, & \mu_{b2}(?x) &= \mathsf{Eve}, \\
\mu_{b3}(?x) &= \mathsf{Alice}, & \mu_{b4}(?x) &= \mathsf{Charlie},
\end{aligned}
$$

and $card_b(\mu_{bi}) = 1$ for all $i \in \{1, 2, 3, 4\}$. The latter may be surprising at first. However, for the PP pattern P_b, as for every PP pattern whose PP expression is of the form (path)**, *the SPARQL specification digresses from the standard bag semantics of other PP patterns* *to an existential semantics where every solution mapping is counted only once, even if there exist multiple matching paths with the same target node (the procedural definition represented by function* ALP2 *achieves this effect by ignoring already visited elements; cf. line 4 in Figure 2).*

3.2. Data Model

The standard query semantics of PP patterns—as introduced in the SPARQL specification and presented in the previous section—defines the result expected from evaluating such a pattern over a (single) RDF graph. Since the WWW is not an RDF graph, this standard definition is insufficient as a formal foundation for evaluating PP patterns over Linked Data on the WWW. As a basis for providing a suitable definition we need a data model that captures the notion of a Web of Linked Data. To this end, we adopt the data model introduced in our earlier work [18].

For this model we assume an infinite set $\mathcal{D}$ that is disjoint from the aforementioned sets $\mathcal{I}$ (IRIs), $\mathcal{B}$ (blank nodes), $\mathcal{L}$ (literals), and $\mathcal{V}$ (variables). Elements in this set $\mathcal{D}$ represent the concept of Web documents from which Linked Data can be extracted; hereafter, we call each $d \in \mathcal{D}$ a *Linked Data document*, or *document* for short. Moreover, we assume a function $data : \mathcal{D} \rightarrow 2^{\mathcal{T}}$ that maps every document $d \in \mathcal{D}$ to a finite set of triples $data(d) \subseteq \mathcal{T}$. As prescribed by the RDF data model [8], we require that the triples of each document use a unique set of blank nodes; i.e., for any pair of distinct documents $d, d' \in \mathcal{D}$, there does not exist two triples $t = \langle s, p, o \rangle$ and $t' = \langle s', p', o' \rangle$ such that $t \in data(d)$, $t' \in data(d')$, and $\{s, p, o\} \cap \{s', p', o'\} \cap \mathcal{B} \neq \emptyset$. Given these preliminaries, we define a Web of Linked Data as follows.

Definition 5. *Assume a special symbol $\perp$ such that $\perp \notin (\mathcal{D} \cup \mathcal{I} \cup \mathcal{B} \cup \mathcal{L} \cup \mathcal{V})$. A* Web of Linked Data *is a tuple $W = \langle D, adoc \rangle$ with the following two elements:*

- *$D \subseteq \mathcal{D}$ is a set of documents; and*
- *$adoc$ is a function that maps every IRI $u \in \mathcal{I}$ either to a document in D or to the symbol $\perp$ (i.e., $adoc \colon \mathcal{I} \rightarrow D \cup \{\perp\}$) such that for every $d \in D$, there exists an IRI $u \in \mathcal{I}$ with $adoc(u) = d$.*

Observe that the function $adoc$ captures the concept of obtaining documents by looking up (HTTP) IRIs on the WWW (also referred to as *dereferencing*). IRIs that cannot be looked up, or whose look up does not result in retrieving a document (even after following HTTP-based redirection pointers) are mapped to the special symbol $\perp$. In this paper we assume that in any Web of Linked Data $W = \langle D, adoc \rangle$ the set of documents D is finite, in which case we say W is *finite* (for a discussion of infiniteness refer to our earlier work [18]).

For the subsequent discussion we introduce a few additional concepts: Given a Web of Linked Data $W = \langle D, adoc \rangle$, we write $\mathrm{dom}^{\not\perp}(adoc)$ to denote the set of IRIs that function $adoc$ maps to a document; i.e., $\mathrm{dom}^{\not\perp}(adoc) = \{u \in \mathcal{I} \,|\, adoc(u) \neq \perp\}$ (hence, this set corresponds to what is also referred to as *"dereferencable IRIs"*). Moreover, for any two documents $d, d' \in D$ in W, we say that document d has a *data link* to d' if there exists some triple $t = \langle s, p, o \rangle$ in the data of d (i.e., $t \in data(d)$) such that t contains an IRI that can be used to obtain d', i.e., $adoc(u) = d'$ for some $u \in \{s, p, o\}$. Such data links establish the *link graph* of the Web of Linked Data W, that is, a directed graph $\langle D, E \rangle$ in which the edges E are all pairs $\langle d, d' \rangle \in D \times D$ for which d has a data link to d'. We emphasize that the link graph of W is a different type of graph than the RDF "graph" whose triples are distributed over the documents in W.

Example 6. *As a running example for the remainder of this paper, we assume a small Web of Linked Data $W_{\mathsf{ex}} = \langle D_{\mathsf{ex}}, adoc_{\mathsf{ex}} \rangle$ consisting of seven documents, $D_{\mathsf{ex}} = \{d_{\mathsf{A}}, d_{\mathsf{B}}, d_{\mathsf{C}}, d_{\mathsf{D}}, d_{\mathsf{E}}, d_{\mathsf{S}}, d_{\mathsf{P}}\}$, with data that describes a project, denoted by IRI* PrjX $\in \mathcal{I}$*, and people, denoted by* Alice, Bob, Charlie, Dody, Eve, Suzi $\in \mathcal{I}$. *Figure 4 presents this data and illustrates the link graph of W_{ex}, assuming function $adoc_{\mathsf{ex}}$ is given as follows:*

$$\begin{aligned} adoc_{\mathsf{ex}}(\mathsf{Alice}) &= d_{\mathsf{A}}, & adoc_{\mathsf{ex}}(\mathsf{Eve}) &= d_{\mathsf{E}}, \\ adoc_{\mathsf{ex}}(\mathsf{Bob}) &= d_{\mathsf{B}}, & adoc_{\mathsf{ex}}(\mathsf{Suzi}) &= d_{\mathsf{S}}, \\ adoc_{\mathsf{ex}}(\mathsf{Charlie}) &= d_{\mathsf{C}}, & adoc_{\mathsf{ex}}(\mathsf{PrjX}) &= d_{\mathsf{P}}, \\ adoc_{\mathsf{ex}}(\mathsf{Dody}) &= d_{\mathsf{D}}, & \textit{and } adoc_{\mathsf{ex}}(u) &= \perp \\ & & \textit{for every other IRI } u. & \end{aligned}$$

We emphasize that the link graph, as well as the two elements D and $adoc$, typically are not available directly to systems that aim to compute queries over the Web of Linked Data captured by $W = \langle D, adoc \rangle$. In particular, the set $\mathrm{dom}^{\not\perp}(adoc)$—i.e., all IRIs that can be used to retrieve some document—is unknown to such systems and can only be disclosed partially (by trying to look up IRIs). This inherent lack of complete information about a queried Web of Linked Data has an impact on the feasibility of answering specific types of queries completely as we shall see in Section 6.

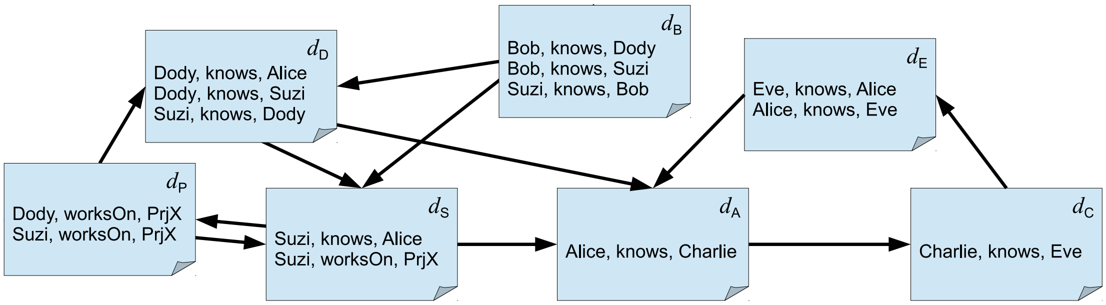


Fig. 4. The link graph of our example Web of Linked Data W_{ex} (self-edges are omitted).

We are now ready to formalize query semantics that define PP patterns as queries over a Web of Linked Data (and, thus, over Linked Data on the WWW).

4. Web-aware Semantics of Property Paths

This section introduces three alternative query semantics, each of which defines an expected query result for any PP pattern over any Web of Linked Data.

4.1. Full-Web Query Semantics

As a first approach we may assume a semantics that is based on the standard evaluation function for PP patterns (cf. Definition 2) and defines expected query results in terms of *all data* in a queried Web of Linked Data. The following definition captures this approach, which we call a *"full-Web query semantics"* [18].

Definition 7. *Let P be a PP pattern, $W = \langle D, adoc \rangle$ be a Web of Linked Data, and G_{all} be the RDF graph for which it holds that $G_{\mathsf{all}} = \bigcup_{d \in D} data(d)$. The* evaluation *of P over W under* full-Web semantics*, denoted by $[\![P]\!]^{\mathit{fw}}_{W}$, is defined by $[\![P]\!]^{\mathit{fw}}_{W} = [\![P]\!]_{G_{\mathsf{all}}}$.*

Example 8. *Recall our example Web W_{ex} (cf. Example 6 and Figure 4). The expected result of evaluating PP pattern $P_a = \langle \mathsf{Suzi}, \mathsf{knows}/\mathsf{knows}, ?x \rangle$ over W_{ex} under full-Web semantics is the multiset of solution mappings $[\![P_a]\!]^{\mathit{fw}}_{W_{\mathsf{ex}}} = \langle \{\mu_{a1}, \mu_{a2}, \mu_{a3}, \mu_{a4}, \mu_{a5}\}, \mathit{card}^{\mathit{fw}}_{a} \rangle$ for which the following properties hold:*

- $\mu_{a1}(?x) = \mathsf{Charlie}$ *and* $\mathit{card}^{\mathit{fw}}_{a}(\mu_{a1}) = 1$ *(because* Suzi *has a "*knows/knows *connection" to* Charlie *via* Alice *by using triples from documents* d_{S} *and* d_{A}*);*
- $\mu_{a2}(?x) = \mathsf{Eve}$ *and* $\mathit{card}^{\mathit{fw}}_{a}(\mu_{a2}) = 1$ *(connection via* Alice *with triples from* d_{S} *and* d_{E}*);*
- $\mu_{a3}(?x) = \mathsf{Alice}$ *and* $\mathit{card}^{\mathit{fw}}_{a}(\mu_{a3}) = 1$ *(via* Dody *by using only triples from* d_{D}*);*
- $\mu_{a4}(?x) = \mathsf{Suzi}$ *and* $\mathit{card}^{\mathit{fw}}_{a}(\mu_{a4}) = 2$ *(connections via* Dody*, see* d_{D}*, and* Bob*, see* d_{B}*);*
- $\mu_{a5}(?x) = \mathsf{Dody}$ *and* $\mathit{card}^{\mathit{fw}}_{a}(\mu_{a5}) = 1$ *(via* Bob*).*

We emphasize that the full-Web query semantics is mostly of theoretical interest. In practice, that is, for a Web of Linked Data $W^* = \langle D^*, adoc^* \rangle$ that represents the "real" WWW (as deployed on the Internet), there cannot exist any system that guarantees to compute the given evaluation function $[\![\cdot]\!]^{\mathit{fw}}_{\cdot}$ over W^* using an algorithm that both terminates and returns complete query results. Our earlier work provides a formal proof of such a limitation of a full-Web query semantics for other types of SPARQL graph patterns, including triple patterns [18]. It is trivial to carry this result over to the full-Web semantics of PP patterns (i.e., Definition 7) because any PP pattern $P = \langle \alpha, \mathtt{path}, \beta \rangle$ with PP expression $\mathtt{path}$ being an IRI $u \in \mathcal{I}$ is a triple pattern $\langle \alpha, u, \beta \rangle$. Informally, we explain this negative result by the fact that the two structures D^* and $adoc^*$ that capture the queried Web formally, are not available for the WWW. Consequently, to enumerate the set of all triples in W^* (denoted by G_{all} in Definition 7), a query execution system would have to discover all documents of the set D^*; given that mapping $adoc^*$ is not available to such a system (in particular, $\mathrm{dom}^{\not\perp}(adoc^*)$—the set of all IRIs whose lookup retrieves a document—is, at best, partially known), the only guarantee to discover all documents is to look up any possible (HTTP) IRI. Since these are infinitely many [9], the enumeration process cannot terminate.

4.2. Reachability-Based Query Semantics

Given the limited practical applicability of the full-Web semantics, our earlier work introduces reachability-based semantics that restrict the scope of queries and expected results to *"reachable"* documents [18]. In the following, we adapt this idea for PP patterns.

Informally, a set of reachable documents of a Web of Linked Data W contains all the documents that can be reached by traversing recursively a well-defined set of data links in the link graph of W. To specify what data links belong to such a set, we introduce the notion of a *reachability criterion* [18], which we define formally as a function $c : \mathcal{T} \times \mathcal{I} \times \mathcal{P} \rightarrow \{\text{true}, \text{false}\}$ where $\mathcal{P}$ denotes the infinite set of all PP patterns (and, as introduced before, $\mathcal{T}$ and $\mathcal{I}$ are the sets of all triples and all IRIs, respectively). Then, given such a reachability criterion, we define reachability of documents as follows.

Definition 9. *Let P be a PP pattern, let $S \subseteq \mathcal{I}$ be a finite set of IRIs (which serve as a seed), let c be a reachability criterion, and let $W = \langle D, adoc \rangle$ be a Web of Linked Data. A document $d \in D$ is (S, c, P)*-reachable *in W if any of the following two conditions holds:*

1. *There exists an IRI $u \in S$ such that $adoc(u) = d$ (in which case we call d a* "seed document"*); or*
2. *there exist (another) document $d' \in D$, a triple t, and an IRI u such that*
   (a) *d' is (S, c, P)-reachable in W,*
   (b) $t \in data(d')$,
   (c) $u \in \text{iris}(t)$,
   (d) $c(t, u, P) = \text{true}$, *and*
   (e) $adoc(u) = d$.

Notice how the second condition restricts the notion of reachability by ignoring any data link that does not satisfy the given reachability criterion. In earlier work we define several concrete reachability criteria [18], including c_{All} that, for each tuple $\langle t, u, P \rangle \in \mathcal{T} \times \mathcal{I} \times \mathcal{P}$, is defined by $c_{\mathsf{All}}(t, u, P) = \text{true}$; hence, c_{All} does not place any restrictions on data links.

Another, more restrictive criterion that is commonly used in practice [19,38], is c_{Match} [18]; this criterion ignores all data links that do not match any triple pattern contained in the given SPARQL query. While our earlier formal definition of c_{Match} assumes that SPARQL queries are constructed from triple patterns [18], we may adapt the idea of this criterion for the PP-based patterns in this paper and define a corresponding reachability criterion that we call c_{PPMatch}.

Definition 10. *For any triple $t = \langle s, p, o \rangle$, IRI u, and PP pattern P, $c_{\mathsf{PPMatch}}(t, u, P) = \text{true}$ if and only if p is an IRI that is mentioned in the PP expression of PP pattern P except for those IRIs that appear* only *in subexpressions of the forms $!(u_1 \mid \ldots \mid u_n)$.*

Example 11. *By using our previous example pattern $P_a = \langle \mathsf{Suzi}, \mathsf{knows/knows}, ?x \rangle$ and $S_{\mathsf{ex}} = \{\mathsf{Suzi}\}$, the following documents are $(S_{\mathsf{ex}}, c_{\mathsf{PPMatch}}, P_a)$-reachable in our example Web W_{ex} (cf. Example 6 and Figure 4): d_{S}, d_{A}, d_{C}, and d_{E}. If we consider the less restrictive reachability criterion c_{All} instead, then we have these four documents and, additionally, d_{P} and d_{D} as being $(S_{\mathsf{ex}}, c_{\mathsf{All}}, P_a)$-reachable in W_{ex} (i.e., all but d_{B}).*

Given the notion of reachability criteria, we define a family of reachability-based semantics for PP patterns:

Definition 12. *Let P be a PP pattern, let $S \subseteq \mathcal{I}$ be a finite set of IRIs, and let c be a reachability criterion. Furthermore, let W be a Web of Linked Data, let D_{R} be the set of all documents that are (S, c, P)-reachable in W, and let G_{R} be the RDF graph for which it holds that $G_{\mathsf{R}} = \bigcup_{d \in D_{\mathsf{R}}} data(d)$. Then, the S-*seeded evaluation *of P over W under c*-semantics, *denoted by $[\![P]\!]_W^{rw(c,S)}$, is defined by $[\![P]\!]_W^{rw(c,S)} = [\![P]\!]_{G_{\mathsf{R}}}$ where $[\![P]\!]_{G_{\mathsf{R}}}$ uses the standard evaluation function for PP patterns (cf. Definition 2).*

Example 13. *Consider $P_a = \langle \mathsf{Suzi}, \mathsf{knows/knows}, ?x \rangle$ and $S_{\mathsf{ex}} = \{\mathsf{Suzi}\}$, then, under c_{All}-semantics, we have $[\![P_a]\!]_{W_{\mathsf{ex}}}^{rw(c_{\mathsf{All}},S_{\mathsf{ex}})} = \langle \{\mu_{a1}, \mu_{a2}, \mu_{a3}, \mu_{a4}\}, card_a^{rw(c_{\mathsf{All}},S_{\mathsf{ex}})} \rangle$ with the solution mappings μ_{a1}–μ_{a4} as in Example 8 and $card_a^{rw(c_{\mathsf{All}},S_{\mathsf{ex}})}(\mu_{ai}) = 1$ for all $i \in \{1, 2, 3, 4\}$. Note that solution mapping μ_{a5} (cf. Example 8) is not a solution in this case because computing it requires triples from document d_{B}, but d_{B} is not $(S_{\mathsf{ex}}, c_{\mathsf{All}}, P_a)$-reachable in W_{ex} (cf. Example 11); due to the same reason we have $card_a^{rw(c_{\mathsf{All}},S_{\mathsf{ex}})}(\mu_{a4}) = 1$ (under full-Web semantics it is $card_a^{fw}(\mu_{a4}) = 2$; cf. Example 8).*

Example 14. *Under c_{PPMatch}-semantics, we only expect the following result for P_a (and S_{ex}) over W_{ex}: $[\![P_a]\!]_{W_{\mathsf{ex}}}^{rw(c_{\mathsf{PPMatch}},S_{\mathsf{ex}})} = \langle \{\mu_{a1}, \mu_{a2}\}, card_a^{rw(c_{\mathsf{PPMatch}},S_{\mathsf{ex}})} \rangle$. As mentioned in Example 8, solution mapping μ_{a3} requires document d_{D}, which is is not $(S_{\mathsf{ex}}, c_{\mathsf{PPMatch}}, P_a)$-reachable in W_{ex} (cf. Example 11); similarly, for μ_{a4}.*

4.3. Context-Based Query Semantics

Reachability-based query semantics as introduced in the previous section impose a clear conceptual separation between navigation over the link graph of a queried Web of Linked Data—which serves the purpose of discovering and retrieving reachable documents—and standard PP-based navigation over the data obtained from all reachable documents. That is, there exists no correlation between paths of triples that match PP expressions and paths of data links that connect reachable documents to seed documents.

At this point it is interesting to also explore an alternative approach in which navigation on the link graph correlates with PP patterns in queries. To this end, we introduce another semantics that interprets PP patterns as a language for navigation over Linked Data on the WWW (i.e., along the lines of earlier navigational languages for Linked Data such as NautiLOD [10]). We refer to this semantics as *context-based*.

The main idea of this query semantics is to restrict the scope of searching for any next triple of a potentially matching path to specific data within specific documents on the queried Web of Linked Data.

To formalize these restrictions we introduce the notion of a *context selector*. Informally, for each IRI that can be used to retrieve a document, the context selector returns a specific subset of the data within that document; this subset contains only those triples that have the given IRI as their subject (such a subset of triples resembles Harth and Speiser's notion of *"subject authoritative triples"* [16]). Formally, for any Web of Linked Data $W = \langle D, adoc\rangle$, the context selector of W is a function $\mathrm{C}^W\colon (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L} \cup \mathcal{V}) \to 2^{\mathcal{T}}$ that, for every IRI $u \in \mathcal{I}$ with $u \in \mathrm{dom}^{\not\perp}(adoc)$, is defined by

$$\mathrm{C}^W(u) = \big\{ \langle s, p, o\rangle \in data\big(adoc(u)\big) \,\big|\, u = s \big\},$$

and for any other $\gamma \in (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L} \cup \mathcal{V}) \setminus \mathrm{dom}^{\not\perp}(adoc)$ we have $\mathrm{C}^W(\gamma) = \emptyset$ (by extending the definition of C^W to handle any such γ, we can simplify the following formalization of the context-based query semantics).

Informally, the context-based semantics uses the notion of a context selector to restrict the scope of PP patterns over a Web of Linked Data as follows. Assume a sequence of triples $\langle s_1, p_1, o_1\rangle$, ..., $\langle s_k, p_k, o_k\rangle$ that presents a path that already matches a sub-expression of a given PP expression. Under the previously defined reachability-based query semantics, the next triple for such a path can be searched for in any reachable document in the queried Web of Linked Data W. By contrast, under the context-based query semantics that we formalize in the following Definition 15, the next triple has to be searched for only in $\mathrm{C}^W(o_k)$.

Definition 15. *Given a PP pattern P and a Web of Linked Data $W = \langle D, adoc\rangle$, the* evaluation *of P over W under* context-based semantics, *denoted by $[\![P]\!]^{ctx}_W$, is a multiset of solution mappings $\langle \Omega, card\rangle$ that is defined recursively as given in Figure 5.*

Note how Definition 15 uses the context selector to restrict the data that has to be searched to find matching triples (e.g., consider the first line in Figure 5).

Example 16. *Coming back to the example PP pattern $P_a = \langle \mathsf{Suzi}, \mathsf{knows/knows}, ?x\rangle$, and W_{ex} (cf. Example 6 and Figure 4), under the context-based semantics we obtain $[\![P_a]\!]^{ctx}_{W_{\mathsf{ex}}} = \langle \{\mu_{a1}\}, card^{ctx}_a\rangle$ with μ_{a1} as before (cf. Example 8) and $card^{ctx}_a(\mu_{a1}) = 1$.*

There are two points worth emphasizing regarding Definition 15: First, we define the context-based semantics such that it resembles the standard semantics of PP patterns in Section 3.1 as close as possible. To this end, the part of our definition that covers PP patterns of the form $\langle \alpha, \mathtt{path}^*, \beta\rangle$ also uses auxiliary functions, namely, ALPW1 and ALPW2 (cf. Figure 6). These functions evaluate the sub-expression `path` recursively over the queried Web of Linked Data (instead of using a fixed RDF graph as done in the standard semantics in Figure 2). Second, the two base cases with a variable in the subject position (i.e., the third and the sixth case in Figure 5) require an enumeration of all IRIs. Such a requirement is necessary to both, remain consistent with the standard semantics and preserve commutativity of operators that can be defined on top of PP patterns (such as the AND operator in SPARQL; cf. Section 5).

However, due to this requirement, there exist PP patterns whose (complete) evaluation under context-based semantics is infeasible when querying the WWW. The following example describes such a case.

Example 17. *Consider the following PP pattern P_{E17}, which retrieves the IRIs of people that know Tim:*

$$P_{\mathsf{E17}} = \langle ?v, \mathsf{knows}, \mathsf{Tim}\rangle.$$

Under context-based semantics, any IRI u' can be used to generate a correct solution mapping for the pattern as long as a lookup of that IRI results in retrieving a document whose data contains the triple $\langle u', \mathsf{knows}, \mathsf{Tim}\rangle$. While, for any Web of Linked Data that is finite, there exists only a finite number of such IRIs, determining these IRIs and guaranteeing completeness requires enumerating the infinite set of all possible IRIs and checking each of them—unless one knows the complete (and finite) subset of all IRIs that can be used to retrieve some document, which, due to the infiniteness of possible HTTP-scheme IRIs, cannot be achieved for the WWW.

It is not difficult to see that the issue illustrated in the example exists for any triple pattern that has a variable in the subject position. On the other hand, triple patterns whose subject is an IRI do not have this issue. However, having an IRI in the subject position is

$$
\begin{aligned}
[\![\langle u_{\mathrm{L}}, p, \beta\rangle]\!]^{\mathtt{ctx}}_{W} &= \Big\langle \,\big\{\, \mu \,|\, \mathrm{dom}(\mu) = (\{\beta\} \cap \mathcal{V}) \text{ and } \mu[\langle u_{\mathrm{L}}, p, \beta\rangle] \in \mathrm{C}^{W}(u_{\mathrm{L}}) \,\big\}\,, \;\; \mathsf{card1}^{(\Omega)} \,\Big\rangle \\
[\![\langle l_{\mathrm{L}}, p, \beta\rangle]\!]^{\mathtt{ctx}}_{W} &= \Big\langle \,\emptyset\,, \;\; \mathsf{card1}^{(\emptyset)} \,\Big\rangle \\
[\![\langle ?v_{\mathrm{L}}, p, \beta\rangle]\!]^{\mathtt{ctx}}_{W} &= \Big\langle \,\big\{\, \mu \,|\, \mathrm{dom}(\mu) = (\{?v_{\mathrm{L}}, \beta\} \cap \mathcal{V}) \\
&\qquad\qquad \text{and } \mu[\langle ?v_{\mathrm{L}}, p, \beta\rangle] \in \bigcup_{u \in \mathcal{I}} \mathrm{C}^{W}(u) \,\big\}\,, \;\; \mathsf{card1}^{(\Omega)} \,\Big\rangle \\
[\![\langle u_{\mathrm{L}}, !(u_1 \mid \cdots \mid u_n), \beta\rangle]\!]^{\mathtt{ctx}}_{W} &= \Big\langle \,\big\{\, \mu \,|\, \mathrm{dom}(\mu) = (\{\beta\} \cap \mathcal{V}) \text{ and there exists an IRI } p \in \mathcal{I} \\
&\qquad\qquad \text{s.t. } p \notin \{u_1, \ldots, u_n\} \text{ and } \mu[\langle u_{\mathrm{L}}, p, \beta\rangle] \in \mathrm{C}^{W}(u_{\mathrm{L}}) \,\big\}\,, \;\; \mathsf{card1}^{(\Omega)} \,\Big\rangle \\
[\![\langle l_{\mathrm{L}}, !(u_1 \mid \cdots \mid u_n), \beta\rangle]\!]^{\mathtt{ctx}}_{W} &= \Big\langle \,\emptyset\,, \;\; \mathsf{card1}^{(\emptyset)} \,\Big\rangle \\
[\![\langle ?v_{\mathrm{L}}, !(u_1 \mid \cdots \mid u_n), \beta\rangle]\!]^{\mathtt{ctx}}_{W} &= \Big\langle \,\big\{\, \mu \,|\, \mathrm{dom}(\mu) = (\{?v_{\mathrm{L}}, \beta\} \cap \mathcal{V}) \text{ and there exists an IRI } p \in \mathcal{I} \\
&\qquad\qquad \text{s.t. } p \notin \{u_1, \ldots, u_n\} \text{ and } \mu[\langle ?v_{\mathrm{L}}, p, \beta\rangle] \in \bigcup_{u \in \mathcal{I}} \mathrm{C}^{W}(u) \,\big\}\,, \mathsf{card1}^{(\Omega)} \,\Big\rangle \\
[\![\langle \alpha, {}^\wedge\mathtt{path}, \beta\rangle]\!]^{\mathtt{ctx}}_{W} &= [\![\langle \beta, \mathtt{path}, \alpha\rangle]\!]^{\mathtt{ctx}}_{W} \\
[\![\langle \alpha, \mathtt{path}_1/\mathtt{path}_2, \beta\rangle]\!]^{\mathtt{ctx}}_{W} &= \pi_{\{\alpha,\beta\} \cap \mathcal{V}}\Big([\![\langle \alpha, \mathtt{path}_1, ?v\rangle]\!]^{\mathtt{ctx}}_{W} \bowtie [\![\langle ?v, \mathtt{path}_2, \beta\rangle]\!]^{\mathtt{ctx}}_{W} \Big) \\
[\![\langle \alpha, \mathtt{path}_1 \,|\, \mathtt{path}_2, \beta\rangle]\!]^{\mathtt{ctx}}_{W} &= [\![\langle \alpha, \mathtt{path}_1, \beta\rangle]\!]^{\mathtt{ctx}}_{W} \sqcup [\![\langle \alpha, \mathtt{path}_2, \beta\rangle]\!]^{\mathtt{ctx}}_{W} \\
[\![\langle x_{\mathrm{L}}, (\mathtt{path})^{*}, ?v_{\mathrm{R}}\rangle]\!]^{\mathtt{ctx}}_{W} &= \Big\langle \,\big\{\, \mu \,|\, \mathrm{dom}(\mu) = \{?v_{\mathrm{R}}\} \text{ and } \mu(?v_{\mathrm{R}}) \in \mathtt{ALPW1}(x_{\mathrm{L}}, \mathtt{path}, W) \big\}\,, \;\; \mathsf{card1}^{(\Omega)} \,\Big\rangle \\
[\![\langle ?v_{\mathrm{L}}, (\mathtt{path})^{*}, ?v_{\mathrm{R}}\rangle]\!]^{\mathtt{ctx}}_{W} &= \Big\langle \,\big\{\, \mu \,|\, \mathrm{dom}(\mu) = \{?v_{\mathrm{L}}, ?v_{\mathrm{R}}\} \text{ and } \mu(?v_{\mathrm{L}}) \in \mathrm{terms}(W) \\
&\qquad\qquad \text{and } \mu(?v_{\mathrm{R}}) \in \mathtt{ALWP1}(\mu(?v_{\mathrm{L}}), \mathtt{path}, W) \,\big\}\,, \;\; \mathsf{card1}^{(\Omega)} \,\Big\rangle \\
[\![\langle ?v_{\mathrm{L}}, (\mathtt{path})^{*}, x_{\mathrm{R}}\rangle]\!]^{\mathtt{ctx}}_{W} &= [\![\langle x_{\mathrm{R}}, ({}^\wedge\mathtt{path})^{*}, ?v_{\mathrm{L}}\rangle]\!]^{\mathtt{ctx}}_{W} \\
[\![\langle x_{\mathrm{L}}, (\mathtt{path})^{*}, x_{\mathrm{R}}\rangle]\!]^{\mathtt{ctx}}_{W} &= \Big\langle \begin{cases} \{\mu_{\emptyset}\} & \text{if } \exists\, \mu \in [\![\langle x_{\mathrm{L}}, (\mathtt{path})^{*}, ?v\rangle]\!]^{\mathtt{ctx}}_{W} : \mu(?v) = x_{\mathrm{R}}, \\ \emptyset & \text{else} \end{cases} , \;\; \mathsf{card1}^{(\Omega)} \,\Big\rangle
\end{aligned}
$$

Fig. 5. Context-based semantics of property paths over a Web of Linked Data; $\alpha, \beta \in (\mathcal{I} \cup \mathcal{L} \cup \mathcal{V})$; $u_{\mathrm{L}}, p, u_1, \ldots, u_n \in \mathcal{I}$; $x_{\mathrm{L}}, x_{\mathrm{R}} \in (\mathcal{I} \cup \mathcal{L})$; $?v_{\mathrm{L}}, ?v_{\mathrm{R}} \in \mathcal{V}$; $?v \in \mathcal{V}$ is a fresh variable; $\mu_{\emptyset}$ is the empty solution mapping with $\mathrm{dom}(\mu_{\emptyset}) = \emptyset$; and function ALPW1 is given in Figure 6.

```
Function ALPW1($(\gamma, \mathtt{path}, W)$)
Input: $\gamma \in (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$,
       path is a PP expression,
       $W$ is a Web of Linked Data.
Visited := $\emptyset$
ALPW2$(\gamma, \mathtt{path}, \mathit{Visited}, W)$
return Visited
```

```
Function ALPW2$(\gamma, \mathtt{path}, \mathit{Visited}, W)$
Input: $\gamma \in (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$,  path is a PP expression,
       $\mathit{Visited} \subseteq (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$,  $W$ is a Web of Linked Data.
if $\gamma \notin \mathit{Visited}$ then
    add $\gamma$ to Visited
    for all $\mu \in [\![\langle ?x, \mathtt{path}, ?y\rangle]\!]^{\mathtt{ctx}}_{W}$ s.t. $\mu(?x) = \gamma$ and $?x, ?y \in \mathcal{V}$ do
        ALPW2$(\mu(?y), \mathtt{path}, \mathit{Visited}, W)$
```

Fig. 6. Auxiliary functions used for defining context-based query semantics.

not a sufficient condition in general. For instance, the PP pattern $\langle \mathsf{Tim}, {}^\wedge\mathsf{knows}, ?v\rangle$ has the same issue as the pattern in Example 17 (in fact, both patterns are semantically equivalent under context-based semantics as can be observed from the seventh case in Figure 5).

A question that arises is whether there exists a (decidable) property of PP patterns that can be used to distinguish between patterns that do not have this issue (i.e., evaluating them over any Web of Linked Data is feasible under the context-based semantics) and those that do. Another question is whether any of the aforementioned reachability-based semantics has a similar problem, and, more generally, how do these semantics compare to the context-based semantics?

We come back to these questions in Sections 6 and 7, after introducing the more general case of PP-based SPARQL queries in the next section.

5. PP-based SPARQL Queries for the Web

After considering PP patterns in isolation, we now turn to a more expressive fragment of SPARQL that embeds PP patterns as the basic building block and uses additional operators on top. In this section, we define the resulting PP-based SPARQL queries; we specify their syntax and formalize Web-aware semantics that extend the above defined semantics of PP patterns.

By using the algebraic syntax of SPARQL [30], we define a *graph pattern* recursively as follows:[1]

- Any PP pattern $\langle \alpha, \mathtt{path}, \beta \rangle$ is a graph pattern.
- If P_1 and P_2 are graph patterns, then so are $(P_1 \;\mathsf{AND}\; P_2)$, $(P_1 \;\mathsf{UNION}\; P_2)$, and $(P_1 \;\mathsf{OPT}\; P_2)$.

For any graph pattern P, we write $\mathrm{vars}(P)$ to denote the set of *all variables* in P; that is, if P is a PP pattern $\langle \alpha, \mathtt{path}, \beta \rangle$, we have $\mathrm{vars}(P) = \{\alpha, \beta\} \cap \mathcal{V}$, and if P is of the form $(P_1 \;\mathsf{AND}\; P_2)$, $(P_1 \;\mathsf{UNION}\; P_2)$, or $(P_1 \;\mathsf{OPT}\; P_2)$, we have $\mathrm{vars}(P) = \mathrm{vars}(P_1) \cup \mathrm{vars}(P_2)$.

Example 18. *An example of a graph pattern that combines two PP patterns using the* OPT *operator is given as follows:* $\big(\langle \mathsf{Tim}, \mathsf{knows/knows}, ?p \rangle \;\mathsf{OPT}\; \langle ?p, \mathsf{name}, ?n \rangle\big)$ *This pattern retrieves persons known by acquaintances of* Tim *and, if available, the names of these persons.*

By using PP patterns as the basic building block of graph patterns, we can readily carry over any of the above defined query semantics to graph patterns. To this end, let $\mathcal{S}$ be a set of symbols that denote these semantics; in particular, we have $\mathtt{fw} \in \mathcal{S}$ that denotes the full-Web semantics (cf. Section 4.1), $\mathtt{rw}(c, S) \in \mathcal{S}$ denotes the (reachability-based) c-semantics with a set S of seed IRIs (cf. Section 4.2), and $\mathtt{ctx} \in \mathcal{S}$ denotes the context-based semantics (cf. Section 4.3). We extend these semantics to cover graph patterns as follows.

Definition 19. *Let P be a graph pattern and let W be a Web of Linked Data. For any $\varphi \in \mathcal{S}$, the* evaluation of P over W under the semantics denoted by φ *is a multiset of solution mappings, denoted by $[\![P]\!]^{\varphi}_{W}$, that is defined recursively as follows:*[2]

- *If P is a PP pattern $\langle \alpha, \mathtt{path}, \beta \rangle$, then $[\![P]\!]^{\varphi}_{W}$ is defined in the φ-specific subsection of Section 4.*
- *If P is of the form $(P_1 \;\mathsf{AND}\; P_2)$, then* $$[\![P]\!]^{\varphi}_{W} = [\![P_1]\!]^{\varphi}_{W} \bowtie [\![P_2]\!]^{\varphi}_{W}.$$
- *If P is of the form $(P_1 \;\mathsf{UNION}\; P_2)$, then* $$[\![P]\!]^{\varphi}_{W} = [\![P_1]\!]^{\varphi}_{W} \sqcup [\![P_2]\!]^{\varphi}_{W}.$$
- *If P is of the form $(P_1 \;\mathsf{OPT}\; P_2)$, then* $$[\![P]\!]^{\varphi}_{W} = \big([\![P_1]\!]^{\varphi}_{W} \bowtie [\![P_2]\!]^{\varphi}_{W}\big) \sqcup \big([\![P_1]\!]^{\varphi}_{W} \setminus [\![P_2]\!]^{\varphi}_{W}\big).$$

6. Web-Safeness

Given the different semantics for evaluating (PP-based) graph patterns over a Web of Linked Data, we now study formally whether such evaluations are possible in practice over Linked Data on the WWW.

To this end, we first recall from Section 4.1 that, under full-Web semantics, evaluating PP patterns over the WWW is not possible in practice because, for the tuple $W = \langle D, adoc \rangle$ with which we formalize the notion of Linked Data on the WWW, the sets D and $\mathrm{dom}^{\not\perp}(adoc)$ cannot be assumed to be available completely to any algorithm [18]. Without complete knowledge of these two sets, an algorithm designed to answer PP patterns completely under full-Web semantics would have to enumerate the infinite set of all possible (HTTP-scheme) IRIs and look up each of them.

Based on this observation, we define a notion of Web-safeness of graph patterns; with this notion we capture whether it is possible for a graph pattern to be evaluated completely over Linked Data on the WWW under a given semantics.

Definition 20. *For any $\varphi \in \mathcal{S}$, a graph pattern P under the semantics denoted by φ is* Web-safe *if there exists an algorithm that, for any finite Web of Linked Data $W = \langle D, adoc \rangle$, has the following properties:*

1. *The algorithm computes $[\![P]\!]^{\varphi}_{W}$.*
2. *During its execution, the algorithm looks up only a finite number of IRIs (that is, conceptually, the algorithm invokes function $adoc$ only a finite number of times).*

[1] For this paper we leave out other types of SPARQL graph patterns such as filters, subqueries, assignments (BIND), aggregation. Adding them is an exercise that would not have any significant implication on the results in this paper.

[2] Note that the definition uses the algebra defined in Figure 1.

3. *Neither the set D nor the set $\mathrm{dom}^{\not\checkmark}(adoc)$ is required as input for the algorithm (hence, the algorithm does not require any a priori information about W).*

Unsurprisingly, as already discussed in Section 4.1, it follows from the results in our earlier work [18] that, under full-Web semantics, none of the graph patterns considered in this paper is Web-safe.

In the following, we study Web-safeness of graph patterns under the other Web-aware query semantics.

6.1. Web-Safeness of Reachability-Based Semantics

Independent of what reachability criterion (and seed IRIs) one chooses, for *every* reachability-based semantics we can show the following positive result.

Theorem 21. *Given an arbitrary reachability criterion c and any finite set $S \subseteq \mathcal{I}$ of IRIs, every graph pattern is Web-safe under c-semantics with S as seed IRIs.*

As a basis to prove Theorem 21, we first focus on PP patterns, for which we show the following lemma.

Lemma 22. *Given an arbitrary reachability criterion c and any finite set $S \subseteq \mathcal{I}$ of IRIs, every PP pattern is Web-safe under c-semantics with S as seed IRIs.*

Proof (Lemma 22). We prove the lemma by providing Algorithm 1. It is easily verified that this algorithm has the desired properties (as listed in Definition 20). Note that the execution of this algorithm consists of two consecutive phases: a data retrieval phase (lines 1 to 12) and a standard result computation phase (line 13). During the data retrieval phase the algorithm incrementally discovers all documents that are (S, c, P)-reachable in the queried Web, and collects their data in RDF graph G_{R}. The second condition in line 11 ensures that any other document is ignored during the data retrieval phase. Hence, when the execution of the algorithm reaches line 13, we have $G_{\mathsf{R}} = \bigcup_{d \in D_{\mathsf{R}}} data(d)$ where D_{R} is the set of all (S, c, P)-reachable documents. Due to the finiteness of the queried Web of Linked Data, both D_{R} and G_{R} are finite. Therefore, there exists a finite upper bound on the number of different IRIs that the algorithm has to look up; in the worst case this upper bound is the number of all IRIs in the final version of G_{R} (in practice, the upper bound may be smaller depending on the reachability criterion c). The existence of this upper bound and the first condition in line 11 ensure that the data retrieval phase terminates. □

Given Lemma 22, it is trivial to prove Theorem 21.

Algorithm 1 Computation of the S-seeded evaluation of a PP pattern P over any Web of Linked Data under c-semantics (where $S \subseteq \mathcal{I}$ is a finite set of IRIs and c is a reachability criterion).

```
 1: $G_{\mathsf{R}}$:= $\emptyset$    // an initially empty RDF graph
 2: Visited := $\emptyset$    // an initially empty set of IRIs
 3: Create a list of IRIs called Open and add every IRI $u \in S$
    to this list (in an arbitrary order)
 4: while Open is not empty do
 5:    Remove the first IRI, say $u$, from Open, add this IRI
       to Visited, and look up this IRI
 6:    if the lookup of IRI $u$ results in retrieving a document,
       say $d$, and $d$ contains triples then
 7:       $G$ := the set of triples in $d$  (use a fresh set of blank
          node identifiers when parsing $d$)
 8:       Add $G$ to $G_{\mathsf{R}}$ (i.e., $G_{\mathsf{R}}$:= $G_{\mathsf{R}} \cup G$)
 9:       for all $t \in G$ do
10:          for all $u' \in \mathrm{iris}(t)$ do
11:             if $u' \notin$ Visited and $c(t, u', P) = \mathrm{true}$ then
12:                Add $u'$ to Open
13: Compute the query result $[\![P]\!]_{G_{\mathsf{R}}}$ (by using an arbitrary
    algorithm that implements the standard SPARQL evalu-
    ation function for PP patterns)
14: return $[\![P]\!]_{G_{\mathsf{R}}}$
```

Proof (Theorem 21). Theorem 21 is a direct consequence of Definition 19 and Lemma 22. That is, given multisets of solution mappings computed for PP patterns, combining such multisets as per the algebra operators does not require any more URI lookups (or any other kind of access to the queried Web of Linked Data) and can be done by any algorithm that implements these algebra operators. □

We emphasize that, while Algorithm 1 is sufficient for proving Lemma 22 and, thus, Theorem 21, it is perhaps not a very efficient algorithm to use in practice. Systems might instead implement traversal-based execution approaches to evaluate PP patterns under reachability-based semantics [19,38]; the processing of IRIs from the *Open* list (used in the algorithm) can be parallelized by a multi-threaded implementation; additionally, assuming a suitable invalidation policy, documents may be cached and reused for later queries [17].

6.2. Web-Safeness of Context-Based Semantics

After finding that under any reachability-based semantics all graph patterns are Web-safe, we now come back to the context-based semantics for which we know from Example 17 that Web-safeness cannot be assumed in general. We begin our analysis by providing the following example, which extends Example 17.

Example 23. *Consider the following graph pattern:*

$$P_{\mathsf{E23}} = \big(\langle \mathsf{Bob}, \mathsf{knows}, ?v\rangle \ \mathsf{AND}\ \langle ?v, \mathsf{knows}, \mathsf{Tim}\rangle\big).$$

The right sub-pattern $P_{\mathsf{E17}} = \langle ?v, \mathsf{knows}, \mathsf{Tim}\rangle$ is not *Web-safe because evaluating it completely over the WWW is not possible under context-based semantics (cf. Example 17). However, the larger pattern P_{E23} is Web-safe under context-based semantics: A possible algorithm may first evaluate the left sub-pattern, $\langle \mathsf{Bob}, \mathsf{knows}, ?v\rangle$, which is possible because it requires the lookup of a single IRI only (the IRI* Bob*). Thereafter, the evaluation of the right sub-pattern P_{E17} can be reduced to looking up a finite number of IRIs only, namely the IRIs bound to variable $?v$ in solution mappings obtained in the first step for the left sub-pattern. Although any other IRI, say u^*, might also be used to discover triples for P_{E17}, each of these triples has IRI u^* as its subject (which is a consequence of restricting retrieved data based on the context selector introduced in Section 4.3). Therefore, possible solution mappings resulting from such triples cannot be compatible with any solution for the left sub-pattern and, thus, do not satisfy the join condition established by the semantics of* AND *in pattern P_{E23}.*

The example illustrates that some graph patterns are Web-safe under context-based semantics even if some of their sub-patterns are not. Consequently, we are interested in a *decidable* property that enables us to identify Web-safe patterns under context-based semantics, including those whose sub-patterns are not Web-safe.

Buil-Aranda et al. study a similar problem in the context of SPARQL federation where graph patterns of the form $(\mathsf{SERVICE}\ ?v\ P)$ are allowed [7]. For such a pattern $P_S = (\mathsf{SERVICE}\ ?v\ P)$, variable $?v$ ranges over a possibly large set of IRIs, each of which represents the address of a (remote) SPARQL service that needs to be called to assemble the complete result of P_S. However, many service calls may be avoided if P_S is embedded in a larger graph pattern that allows for an evaluation during which $?v$ can be bound before evaluating P_S. To identify such cases, Buil-Aranda et al. introduce a notion of *strong boundedness* of variables in graph patterns and use it to show a notion of safeness for the evaluation of patterns like P_S within larger graph patterns. The idea behind the notion of strongly bound variables has already been used in earlier work (e.g., *"certain variables"* [34], *"output variables"* [37]), and it is tempting to adopt it for our problem. To this end, we first define the notion of strongly bound variables for our PP-based graph patterns:

Definition 24. *The set of* strongly bound variables *in a graph pattern P, denoted by* $\mathrm{sbvars}(P)$*, is defined recursively as follows (recall that* $\mathrm{vars}(P)$ *is the set of all variables in P):*

- *If P is a PP pattern, then* $\mathrm{sbvars}(P) = \mathrm{vars}(P)$.
- *If P is of the form $(P_1\ \mathsf{AND}\ P_2)$, then* $\mathrm{sbvars}(P) = \mathrm{sbvars}(P_1) \cup \mathrm{sbvars}(P_2)$.
- *If P is of the form $(P_1\ \mathsf{UNION}\ P_2)$, then* $\mathrm{sbvars}(P) = \mathrm{sbvars}(P_1) \cap \mathrm{sbvars}(P_2)$.
- *If P is of the form $(P_1\ \mathsf{OPT}\ P_2)$, then* $\mathrm{sbvars}(P) = \mathrm{sbvars}(P_1)$.

Given the definition of strongly bound variables, we observe that one cannot identify Web-safe graph patterns by using *only* this notion of strong boundedness.

Example 25. *Consider graph pattern P_{E23} from Example 23. We know that (i) P_{E23} is Web-safe and that (ii)* $\mathrm{vars}(P_{\mathsf{E23}}) = \{?v\}$ *and also* $\mathrm{sbvars}(P_{\mathsf{E23}}) = \{?v\}$*. Then, one might hypothesize that a graph pattern P is Web-safe if* $\mathrm{sbvars}(P) = \mathrm{vars}(P)$*. However, the PP pattern $P_{\mathsf{E17}} = \langle ?v, \mathrm{knows}, \mathrm{Tim}\rangle$ disproves such a hypothesis because, even if* $\mathrm{sbvars}(P_{\mathsf{E17}}) = \mathrm{vars}(P_{\mathsf{E17}})$*, pattern P_{E17} is not Web-safe (cf. Example 17). Alternatively, one might also hypothesize that if a graph pattern P is Web-safe, then* $\mathrm{sbvars}(P) = \mathrm{vars}(P)$*. However, this hypothesis can be disproved by using pattern $P_{\mathsf{E25}} = \big(\langle \mathsf{Bob}, \mathsf{knows}, ?x\rangle\ \mathsf{OPT}\ \langle ?x, \mathsf{knows}, ?y\rangle\big)$. It can easily be verified that P_{E25} is Web-safe (e.g., it is not difficult to adjust the algorithm for pattern P_{E23} in Example 23 accordingly). However, in contradiction to the hypothesis we have* $\mathrm{sbvars}(P_{\mathsf{E25}}) \neq \mathrm{vars}(P_{\mathsf{E25}})$.

We conjecture the following reason why strong boundedness cannot be used directly for our problem. Consider the types of graph patterns that combine two sub-patterns (by using operators such as AND). For such a pattern, the sets of strongly bound variables of its sub-patterns are defined *independent* from each other, whereas the algorithm outlined in Example 23 leverages a specific relationship between sub-patterns. More precisely, the algorithm leverages the fact that the same variable that is the subject of the right sub-pattern is also the object of the left sub-pattern.

Based on this observation, we introduce the notion of *conditionally bound variables*, which is based on particular relationships between sub-patterns due to which the result of one sub-pattern may be used to evaluate another sub-pattern in a more well-behaved manner (along the lines of Example 23). This notion shall turn out to be suitable for our case.

Definition 26. *Let $X \subseteq \mathcal{V}$ be a set of variables. The* conditionally bound variables *in a graph pattern P w.r.t. X, denoted by* $\mathrm{cbvars}(P \,|\, X)$*, is a subset of the variables in P (i.e.,* $\mathrm{cbvars}(P \,|\, X) \subseteq \mathrm{vars}(P)$*) that is defined recursively as given in Table 1.*

Example 27. *The conditionally bound variables in the PP pattern $P_{\mathsf{E17}} = \langle ?v, \mathsf{knows}, \mathsf{Tim} \rangle$ w.r.t. the* empty *set of variables can be determined based on line 2 in Table 1, and we obtain:* $\mathrm{cbvars}(P_{\mathsf{E17}} \,|\, \emptyset) = \emptyset$*. However, if we use the set $\{?v\}$ instead, then, by line 1 in Table 1, we obtain:* $\mathrm{cbvars}\big(P_{\mathsf{E17}} \,\big|\, \{?v\}\big) = \{?v\}$.

Example 28. *As another example consider the graph pattern $P_{\mathsf{E23}} = \big(\langle \mathsf{Bob}, \mathsf{knows}, ?v \rangle \text{ AND } \langle ?v, \mathsf{knows}, \mathsf{Tim} \rangle\big)$ for which we obtain* $\mathrm{cbvars}(P_{\mathsf{E23}} \,|\, \emptyset) = \{?v\}$ *by using line 10 in Table 1 and the following facts:*

1. $\mathrm{cbvars}\big(\langle \mathsf{Bob}, \mathsf{knows}, ?v \rangle \,\big|\, \emptyset\big) = \{?v\}$*,*
2. $\mathrm{sbvars}\big(\langle \mathsf{Bob}, \mathsf{knows}, ?v \rangle\big) = \{?v\}$*,*
3. $\mathrm{cbvars}\big(\langle ?v, \mathsf{knows}, \mathsf{Tim} \rangle \,\big|\, \{?v\}\big) = \{?v\}$*.*

We note that for the pattern P_{E17}, which is *not* Web-safe under context-based semantics (as discussed in Example 17), we have $\mathrm{cbvars}(P_{\mathsf{E17}} \,|\, \emptyset) \neq \mathrm{vars}(P_{\mathsf{E17}})$, whereas for the pattern P_{E23}, which is Web-safe under context-based semantics (cf. Example 23), we have $\mathrm{cbvars}(P_{\mathsf{E23}} \,|\, \emptyset) = \mathrm{vars}(P_{\mathsf{E23}})$. This example seems to suggest that, if *all* variables of a graph pattern are conditionally bound w.r.t. the empty set of variables, then the graph pattern is Web-safe under context-based semantics. The following result verifies this hypothesis.

Theorem 29. *A graph pattern P is Web-safe under context-based semantics if* $\mathrm{cbvars}(P \,|\, \emptyset) = \mathrm{vars}(P)$.

Before proving Theorem 29 in the remainder of this section, we emphasize the following observation.

Note 30. *Due to the recursive nature of Definition 26, the condition* $\mathrm{cbvars}(P \,|\, \emptyset) = \mathrm{vars}(P)$ *(as used in Theorem 29) is decidable for any graph pattern P.*

To prove Theorem 29 we aim to provide an algorithm that evaluates graph patterns recursively by passing (intermediate) solution mappings to recursive calls. To capture the desired results of each recursive call formally, we introduce a special evaluation function for a graph pattern P over a Web of Linked Data W that takes a solution mapping μ as input and returns only the solutions of P over W that are compatible with μ (recall from Section 3.1 that the compatibility of two solution mappings, μ_1 and μ_2, is denoted by $\mu_1 \sim \mu_2$).

Definition 31. *Let P be a graph pattern, let W be a Web of Linked Data, and let $\langle \Omega, card \rangle = [\![P]\!]^{\mathit{ctx}}_{W}$. Given a solution mapping μ, the μ-*restricted evaluation *of P over W under* context-based semantics*, denoted by $[\![P \,|\, \mu\,]\!]^{\mathit{ctx}}_{W}$, is the multiset of solution mappings $\langle \Omega', card' \rangle$ with $\Omega' = \big\{\mu' \in \Omega \,\big|\, \mu' \sim \mu\big\}$ and $card'$ is the restriction of card to Ω', i.e., for every solution mapping $\mu' \in \Omega'$ we have $card'(\mu') = card(\mu')$.*

The following lemma shows the existence of the aforementioned recursive algorithm.

Lemma 32. *Let P be a graph pattern and μ_{in} be a solution mapping. If* $\mathrm{cbvars}\big(P \,\big|\, \mathrm{dom}(\mu_{\mathsf{in}})\big) = \mathrm{vars}(P)$*, then there exists an algorithm that, for any finite Web of Linked Data $W = \langle D, adoc \rangle$, has the following three properties:*

1. *The algorithm computes $[\![P \,|\, \mu_{\mathsf{in}}\,]\!]^{\mathit{ctx}}_{W}$.*
2. *During its execution, the algorithm looks up only a finite number of IRIs (that is, conceptually, the algorithm invokes function adoc only a finite number of times).*
3. *Neither the set D nor the set* $\mathrm{dom}^{\not\perp}(adoc)$ *is required as input for the algorithm (hence, the algorithm does not require any a priori information about W).*

Before proving the lemma (and Theorem 29), we point out two important properties of Definition 31. First, it is easily seen that, for any graph pattern P and Web of Linked Data W, $[\![P \,|\, \mu_{\emptyset}\,]\!]^{\mathtt{ctx}}_{W} = [\![P]\!]^{\mathtt{ctx}}_{W}$, where $\mu_{\emptyset}$ is the empty solution mapping with $\mathrm{dom}(\mu_{\emptyset}) = \emptyset$. Consequently, given an algorithm, say A, that, for P and $\mu_{\emptyset}$, has the properties of the algorithm described by Lemma 32, a trivial algorithm that can be used to prove Theorem 29 may simply call algorithm A and return the result of this call (a more detailed discussion of this approach follows in the proof of Theorem 29 below). Second, for any PP pattern $\langle \alpha, \mathtt{path}, \beta \rangle$ and Web of Linked Data W, if α is a variable and $\mathtt{path}$ is a PP expression that corresponds to one of the first two cases in the grammar in Section 3.1 (i.e., the two base cases), then $[\![P \,|\, \mu\,]\!]^{\mathtt{ctx}}_{W}$ is empty for every solution mapping μ that binds (variable) α to a literal or a blank node. Formally, we show the latter as follows.

Lemma 33. *Let $?v \in \mathcal{V}$ be a variable, P be a PP pattern of the form $\langle ?v, u, \beta \rangle$ or $\langle ?v, !(u_1 \,|\, \ldots \,|\, u_n), \beta \rangle$ with $u, u_1, \ldots, u_n \in \mathcal{I}$, and μ be a solution mapping. If $?v \in \mathrm{dom}(\mu)$ and $\mu(?v) \in (\mathcal{B} \cup \mathcal{L})$, then, for any Web of Linked Data W, $[\![P \,|\, \mu\,]\!]^{\mathtt{ctx}}_{W}$ is the empty multiset (of solution mappings).*

If P is:	then $\mathrm{cbvars}(P \,\vert\, X)$ is:
1) $\langle \alpha, u, \beta\rangle$ or $\langle \alpha, !(u_1 \,\vert\, ... \,\vert\, u_n), \beta\rangle$ such that $\alpha \in (\mathcal{I} \cup \mathcal{L})$ or $\alpha \in X$	$\mathrm{vars}(P)$
2) $\langle \alpha, u, \beta\rangle$ or $\langle \alpha, !(u_1 \,\vert\, ... \,\vert\, u_n), \beta\rangle$ such that $\alpha \notin (\mathcal{I} \cup \mathcal{L})$ and $\alpha \notin X$	$\emptyset$
3) $\langle \alpha, (\texttt{path})^*, \beta\rangle$ such that $\alpha \in \mathcal{V}$ and $\beta \notin \mathcal{V}$	$\mathrm{cbvars}\big(\langle \beta, ({}^\wedge\texttt{path})^*, \alpha\rangle \,\vert\, X\big)$
4) $\langle \alpha, (\texttt{path})^*, \beta\rangle$ such that $\alpha \notin \mathcal{V}$ or $\beta \in \mathcal{V}$, and for any two variables $?x, ?y \in \mathcal{V}$ it holds that $\mathrm{cbvars}\big(\langle ?x, \texttt{path}, ?y\rangle \,\vert\, \{?x\}\big) = \{?x, ?y\}$	$\mathrm{cbvars}\big(\langle \alpha, \texttt{path}, \beta\rangle \,\vert\, X\big)$
5) $\langle \alpha, (\texttt{path})^*, \beta\rangle$ such that none of the above	$\emptyset$
6) $\langle \alpha, {}^\wedge\texttt{path}, \beta\rangle$ with $P' = \langle \beta, \texttt{path}, \alpha\rangle$	$\mathrm{cbvars}(P' \,\vert\, X)$
7) $\langle \alpha, (\texttt{path}_1 \vert \texttt{path}_2), \beta\rangle$ with $P' = \big(\langle \alpha, \texttt{path}_1, \beta\rangle \text{ UNION } \langle \alpha, \texttt{path}_2, \beta\rangle\big)$	$\mathrm{cbvars}(P' \,\vert\, X)$
8) $\langle \alpha, \texttt{path}_1/\texttt{path}_2, \beta\rangle$ such that for any $?v \in \mathcal{V} \setminus (X \cup \{\alpha, \beta\})$ we have $?v \in \mathrm{cbvars}(P' \vert\, X)$ where $P' = \big(\langle \alpha, \texttt{path}_1, ?v\rangle \text{ AND } \langle ?v, \texttt{path}_2, \beta\rangle\big)$	$\mathrm{cbvars}(P' \,\vert\, X) \setminus \{?v\}$
9) $\langle \alpha, \texttt{path}_1/\texttt{path}_2, \beta\rangle$ such that none of the above	$\emptyset$
10) $(P_1 \text{ AND } P_2)$ s.t. $\mathrm{cbvars}(P_1 \,\vert\, X) = \mathrm{vars}(P_1)$ and $\mathrm{cbvars}(P_2 \,\vert\, X \cup \mathrm{sbvars}(P_1)) = \mathrm{vars}(P_2)$	$\mathrm{vars}(P)$
11) $(P_1 \text{ AND } P_2)$ s.t. $\mathrm{cbvars}(P_2 \,\vert\, X) = \mathrm{vars}(P_2)$ and $\mathrm{cbvars}(P_1 \,\vert\, X \cup \mathrm{sbvars}(P_2)) = \mathrm{vars}(P_1)$	$\mathrm{vars}(P)$
12) $(P_1 \text{ AND } P_2)$ such that none of the above	$\emptyset$
13) $(P_1 \text{ UNION } P_2)$	$\mathrm{cbvars}(P_1 \,\vert\, X) \cap \mathrm{cbvars}(P_2 \,\vert\, X)$
14) $(P_1 \text{ OPT } P_2)$ s.t. $\mathrm{cbvars}(P_1 \,\vert\, X) = \mathrm{vars}(P_1)$ and $\mathrm{cbvars}(P_2 \,\vert\, X \cup \mathrm{sbvars}(P_1)) = \mathrm{vars}(P_2)$	$\mathrm{vars}(P)$
15) $(P_1 \text{ OPT } P_2)$ such that none of the above	$\emptyset$

Table 1

Cases of the recursive definition of the conditionally bound variables of a graph pattern P w.r.t. a set of variables $X \subseteq \mathcal{V}$.

Proof (Lemma 33). Recall that for any IRI u and any Web of Linked Data W, every triple in the context $\mathrm{C}^W(u)$ has IRI u as its subject. As a consequence, for any Web of Linked Data W, every solution mapping in $[\![P]\!]^{\mathtt{ctx}}_W$ binds variable $?v$ to some IRI (and not to a literal or a blank node); that is, formally, for every $\mu' \in [\![P]\!]^{\mathtt{ctx}}_W$ we have $\mu'(?v) \in \mathcal{I}$. Therefore, if $?v \in \mathrm{dom}(\mu)$ and $\mu(?v) \in (\mathcal{B} \cup \mathcal{L})$, then none of the solution mappings in $[\![P]\!]^{\mathtt{ctx}}_W$ is compatible with μ, and, thus, $[\![P \,\vert\, \mu\,]\!]^{\mathtt{ctx}}_W$ is empty. □

We use Lemma 33 to prove Lemma 32 as follows.

Proof idea (Lemma 32). We prove Lemma 32 by induction on the possible structure of graph pattern P. To this end, we provide Algorithm 2 and show that this (recursive) algorithm has the desired properties for any possible graph pattern (i.e., any case of the induction, including the base case). In this paper we focus on a fragment of the algorithm and highlight essential properties thereof. This fragment covers the base case (lines 1-11) and one pivotal case of the induction step, namely, graph patterns of the form $(P_1 \text{ AND } P_2)$. The complete version of the algorithm and the full proof can be found in our technical report [22].

For the base case (i.e., PP patterns of the form $\langle \alpha, u, \beta\rangle$ or $\langle \alpha, !(u_1 \,\vert\, \dots \,\vert\, u_n), \beta\rangle$), Algorithm 2 looks up at most one IRI (cf. lines 2-5). The crux of showing that the returned result is sound and complete is Lemma 33 and the fact that a triple $\langle s, p, o\rangle$ with $s \in \mathcal{I}$ can be found only in the context $\mathrm{C}^W(s)$.

For PP patterns of the form $(P_1 \text{ AND } P_2)$ consider lines 57-72. For sub-patterns P_i and P_j as used in this part of the algorithm, we may use Definition 26 to show that (i) $\mathrm{cbvars}\big(P_i \,\vert\, \mathrm{dom}(\mu_{\mathsf{in}})\big) = \mathrm{vars}(P_i)$ and (ii) $\mathrm{cbvars}\big(P_j \,\big\vert\, \mathrm{dom}(\mu_{\mathsf{in}}) \cup \mathrm{dom}(\mu)\big) = \mathrm{vars}(P_j)$ for all $\mu \in \Omega^{P_i}$. Therefore, by induction, any recursive call of the algorithm in line 61 and line 63 looks up a finite number of IRIs and returns the expected (sound and complete) result; that is, $\langle \Omega^{P_i}, \mathit{card}^{P_i}\rangle = [\![P_i \,\vert\, \mu_{\mathsf{in}}\,]\!]^{\mathtt{ctx}}_W$ and $\langle \Omega^{\mu}, \mathit{card}^{\mu}\rangle = [\![P_j \,\vert\, \mu_{\mathsf{in}} \cup \mu\,]\!]^{\mathtt{ctx}}_W$ for all $\mu \in \Omega^{P_i}$. Then, since every $\mu \in \Omega^{P_i}$ is compatible with every $\mu' \in \Omega^{\mu}$ and all processed solution mappings are compatible with μ_{in}, it is easily verified that the computed result is $[\![(P_1 \text{ AND } P_2) \,\vert\, \mu_{\mathsf{in}}\,]\!]^{\mathtt{ctx}}_W$. □

We are now ready to prove Theorem 29.

Proof (Theorem 29). Suppose P is a graph pattern such that $\mathrm{cbvars}(P \,\vert\, \emptyset) = \mathrm{vars}(P)$. Then, by using the empty solution mapping $\mu_\emptyset$ with $\mathrm{dom}(\mu_\emptyset) = \emptyset$, we have $\mathrm{cbvars}\big(P \,\big\vert\, \mathrm{dom}(\mu_\emptyset)\big) = \mathrm{vars}(P)$. Therefore, by Lemma 32, there exists an algorithm, say A, that, for any finite Web of Linked Data $W = \langle D, \mathit{adoc}\rangle$, computes $[\![P \,\vert\, \mu_\emptyset\,]\!]^{\mathtt{ctx}}_W$ by looking up a finite number of IRIs only without using the set D or the set

Algorithm 2 *EvalCtxBased*(P, μ_{in}), which computes $[\![P \mid \mu_{\mathsf{in}}]\!]^{\mathtt{ctx}}_W$ for a Web of Linked Data W.

```
if $P$ is $\langle \alpha, u, \beta \rangle$ or $\langle \alpha, !(u_1 \mid \dots \mid u_n), \beta \rangle$ then
    if $\alpha \in \mathcal{I}$ then $u' := \alpha$
    else if $\alpha \in \mathrm{dom}(\mu_{\mathsf{in}})$ and $\mu_{\mathsf{in}}(\alpha) \in \mathcal{I}$ then $u' := \mu_{\mathsf{in}}(\alpha)$
    else $u' :=$ null

    if $u'$ is an IRI and looking it up results in retrieving a document, say $d$ then
        $G :=$ the set of triples in $d$ (use a fresh set of blank node identifiers when parsing $d$)
        $G' := \{\langle s, p, o \rangle \in G \mid s = u'\}$
        $\langle \Omega, card \rangle := [\![P]\!]_{G'}$   ($[\![P]\!]_{G'}$ can be computed by using any algorithm that implements the standard SPARQL evaluation function)
        return a new multiset $\langle \Omega', card' \rangle$ with
               $\Omega' = \{\mu' \in \Omega \mid \mu' \sim \mu_{\mathsf{in}}\}$ and
               $card'(\mu') = card(\mu')$ for all $\mu' \in \Omega'$
    else
        return a new empty multiset $\langle \Omega, card \rangle$ with
               $\Omega = \emptyset$ and $\mathrm{dom}(card) = \emptyset$

else if $P$ is ...
...
else if $P$ is of the form $(P_1$ AND $P_2)$ then
    if $\mathrm{cbvars}(P_1 | \mathrm{dom}(\mu_{\mathsf{in}})) = \mathrm{vars}(P_1)$ then $i$:=1; $j$:=2
    else $i$:=2; $j$:=1

    Create a new empty multiset $M = \langle \Omega, card \rangle$
    $\langle \Omega^{P_i}, card^{P_i} \rangle :=$ EvalCtxBased$(P_i, \mu_{\mathsf{in}})$
    for all $\mu \in \Omega^{P_i}$ do
        $\langle \Omega^{\mu}, card^{\mu} \rangle :=$ EvalCtxBased$(P_j, \mu_{\mathsf{in}} \cup \mu)$
        for all $\mu' \in \Omega^{\mu}$ do
            $\mu^* := \mu \cup \mu'$
            $k := card^{P_i}(\mu) \cdot card^{\mu}(\mu')$
            if $\mu^* \in \Omega$ then
                $old := card(\mu^*)$
                Set $card(\mu^*) = k + old$
            else
                Set $card(\mu^*) = k$, and add $\mu^*$ to $\Omega$
    return $M$

else if $P$ is ...
```

$\mathrm{dom}^{\not\perp}(adoc)$ as input. We also know that the empty solution mapping $\mu_\emptyset$ is compatible with any solution mapping. Consequently, by Definition 31, we have $[\![P \mid \mu_\emptyset]\!]^{\mathtt{ctx}}_W = [\![P]\!]^{\mathtt{ctx}}_W$ for any Web of Linked Data W. Hence, algorithm A can be used to compute $[\![P]\!]^{\mathtt{ctx}}_W$ for any finite Web of Linked Data W (and during this computation the algorithm looks up a finite number of IRIs only without using D or $\mathrm{dom}^{\not\perp}(adoc)$ as input). □

While the condition given in Theorem 29 is sufficient to identify graph patterns that are Web-safe under context-based semantics, the question that remains is whether it is a necessary condition (i.e., whether it can be used to decide Web-safeness of *all* graph patterns under context-based semantics). Unfortunately, the answer is *no* as the following example shows.

Example 34. *For the graph pattern* $P = (P_1$ UNION $P_2)$ *with* $P_1 = \langle u_1, p_1, ?x \rangle$ *and* $P_2 = \langle u_2, p_2, ?y \rangle$ *we note that* $\mathrm{cbvars}(P_1 \mid \emptyset) = \{?x\}$ *and* $\mathrm{cbvars}(P_2 \mid \emptyset) = \{?y\}$, *and, thus,* $\mathrm{cbvars}(P \mid \emptyset) = \emptyset$. *Hence, the pattern does not satisfy the condition in Theorem 29. Nonetheless, it is easy to see that there exists a (sound and complete) algorithm that, for any finite Web of Linked Data W, computes* $[\![P]\!]^{\mathtt{ctx}}_W$ *by looking up a finite number of IRIs only. For instance, such an algorithm, say A, may first use two other algorithms that compute* $[\![P_1]\!]^{\mathtt{ctx}}_W$ *and* $[\![P_2]\!]^{\mathtt{ctx}}_W$ *by looking up a finite number of IRIs, respectively. Such algorithms exist by Theorem 29, because* $\mathrm{cbvars}(P_1 \mid \emptyset) = \mathrm{vars}(P_1)$ *and* $\mathrm{cbvars}(P_2 \mid \emptyset) = \mathrm{vars}(P_2)$. *Finally, algorithm A can generate the (sound and complete) query result* $[\![P]\!]^{\mathtt{ctx}}_W$ *by computing the multiset union* $[\![P_1]\!]^{\mathtt{ctx}}_W \sqcup [\![P_2]\!]^{\mathtt{ctx}}_W$, *which requires no additional IRI lookups.*

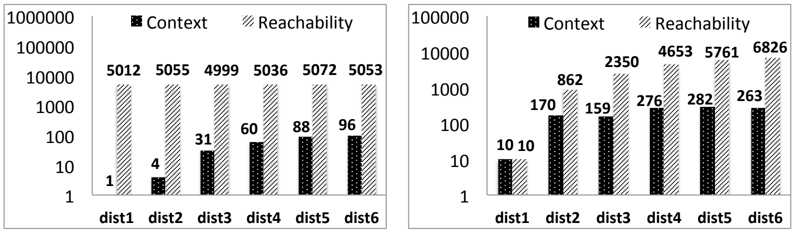
The example illustrates that "only if" cannot be shown in Theorem 29. It remains an open question whether there exists an alternative condition for Web-safeness that is both sufficient and necessary (and decidable) and, thus, can be used to decide Web-safeness of all graph patterns under context-based semantics.

7. Experimental Comparison

In the previous section we have shown that, when querying Linked Data on the WWW, it is possible for PP-based graph patterns to be evaluated completely under any reachability-based semantics, and, similarly, under the context-based semantics (assuming, for the latter, we use only patterns that have been identified to be Web-safe). Hence, we have shown that—based on these semantics—one can build a system that answers PP-based SPARQL queries over the WWW *in a well-defined manner*. At this point, a natural question that arises is:

How do these query semantics compare when actually used in practice?

To achieve empirical insights related to this question we conducted an experimental comparison of the



(a) # Dereferencing Operations (logscale)

(b) # Solutions (logscale)

Fig. 7. Comparison between context-based semantics and (reachability-based) c_{PPMatch}-semantics on D1.

context-based semantics and a reachability-based semantics. For this comparison we selected c_{PPMatch}-semantics as an exemplar of the family of reachability-based semantics; as argued in Section 4.2, c_{PPMatch} is very close in nature to the reachability criterion c_{Match} [18] which is commonly used in the literature on Linked Data query execution approaches [19,38] (note that c_{Match} is defined for SPARQL queries constructed from triple patterns, instead of PP patterns).

In the remainder of this section, we specify the experimental setup, describe the experiments, present the measurements, and discuss the experimental results.

7.1. Metrics and Experimental Setup

The objective of the experimental comparison is to identify the differences between the studied semantics in terms of (i) number of dereferencing operations performed to evaluate a query and (ii) number of solutions in the respective query results, including duplicates (which are possible in our bag semantics as Example 13 illustrates). Hereafter, we refer to these metrics as (i) `nderef` and (ii) `ressize`, respectively. Since this paper focuses on possible query semantics rather than on efficient techniques to implement such semantics, performance-related metrics such as query execution time are out of scope of our study.

For the experiments, which we conducted during the days of November 16–28, 2015, we used a prototypical implementation of the studied semantics to execute PP-based SPARQL queries *directly on the WWW*. To avoid overloading Web servers we introduced a delay of 3 seconds between dereferencing operations. While we did not use any client-side caching of retrieved documents, there may have been Web caches (proxy servers) between our prototypical query clients and the Web servers that host the data discovered and retrieved during the execution of our test queries. Measurements reported in the following are the average of five executions with rounding to the next integer.

7.2. Experiments and Measurements

We conducted two different experiments considering two different topical domains of Linked Data on the WWW, namely, distributed social network data (D1) and encyclopedic data about influence relationships between people (D2). Within these domains we focus on navigational queries that we express using PP patterns. The particular queries used for the experiments can be found in Appendix A. In the following, we describe the experiments and the queries in more detail, and we present the measurements.

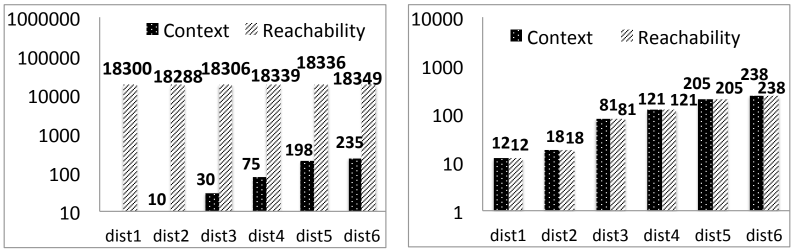
7.2.1. Experiment on D1

In our first experiment we considered the distributed social network of FOAF profiles [14]. Such FOAF profiles typically are RDF documents that people make available online to provide Linked Data that describes themselves in terms of their interests, their works, and, most important for our experiment, references to other people they know. Such references are expressed using triples with the IRI[3] `foaf:knows` as predicate and the persons' IRIs as subject and object (i.e., along the lines of our example Web in Figure 4). Hence, such triples establish data links between different people's FOAF profiles. The resulting network of such "`foaf:knows` links" is thus a part of the Web of Linked Data, and it is the focus of our first experiment. We point out that this experiment is particularly significant due to the truly distributed nature of the FOAF profiles, which typically reside (and get updated) on different servers. Indeed, there is no SPARQL endpoint to query (the live version of) this kind of distributed social network.

In this experiment we use the IRI of Nuno Lopes[4] in his FOAF profile as a starting point for six queries that

[3] For the compact representation of IRIs in this section we use the following two prefixes. foaf: <http://xmlns.com/foaf/> and dbo: <http://dbpedia.org/ontology/>

[4] http://nunolopes.org/foaf.rdf#me



(a) # Dereferencing Operations (logscale)

(b) # Solutions (logscale)

Fig. 8. Comparison between context-based semantics and (reachability-based) c_{PPMatch}-semantics on D2.

retrieve Lopes' acquaintances from distances 1 to 6, respectively. The measurements obtained by executing these queries under both the context-based semantics and the (reachability-based) c_{PPMatch}-semantics are reported in the charts in Figure 7; the x-axes list the six queries and the y-axes represent our metrics, `nderef` and `ressize`, respectively (reported in log-scale).

By looking at Figure 7 (a), we notice that, under the context-based semantics, `nderef` is increasing steadily with the (increasing) distance selected in the queries. In contrast, under c_{PPMatch}-semantics, `nderef` is almost the same for all six queries, and it is significantly higher than under the context-based semantics, even for the distance-6 query. Considering the definition of the reachability criterion c_{PPMatch} (cf. Definition 10), this observation is not unexpected: While the PP patterns of the six queries differ, they all mention the same IRI in their PP expressions, namely, `foaf:knows`. Consequently, in all six cases, the same set of documents is reachable by applying c_{PPMatch} as reachability criterion (and using the same seed IRI). Essentially, this set of documents represents the complete strongly connected component of FOAF profiles that contains the profile of the seed IRI. Recall that the data of all these documents must be retrieved to compute query results that are guaranteed to be complete under c_{PPMatch}-semantics; this explains the comparable high number of dereferencing operations. The slight variations of these numbers across the six queries are due to occasional timeouts of dereferencing operations and Web servers that did not always respond during each query execution.

The effect of taking into account more data can be observed by looking at our `ressize` measurements in Figure 7 (b). Clearly, under c_{PPMatch}-semantics we obtain query results that have a much greater size than the results under the context-based semantics, in particular, for the higher distance queries. This effect, again, is not unexpected. Instead, it can also be seen, on a much smaller scale, in the examples in Section 4 (compare in particular Examples 14 and 16). However, we note that the greater `ressize` per query under c_{PPMatch}-semantics is due not only to finding paths to additional persons in the data retrieved under c_{PPMatch}-semantics, but also to a greater number of duplicates, which result from finding a greater number of alternative paths to some persons (cf. Example 3).

The only exception, where the query result under both semantics is the same, is the distance-1 query. This query consists only of a single triple pattern with the seed IRI as subject, `foaf:knows` as predicate, and a variable as object. In the given case of using Nuno Lopes' IRI as seed, all triples that match this pattern happen to be in the same document (Lopes' FOAF profile) and, thus, all other documents retrieved under c_{PPMatch}-semantics turn out to not contribute to the query result (which may be different for other seeds).

7.2.2. *Experiment on D2*

For our second experiment we considered influence relationships between people described in Linked Data that is made available by the DBpedia project [4]. In particular, we focused on the relationships expressed by triples with the IRI `dbo:influencedBy` as predicate, and we used the IRI of Veno Taufer[5] as starting point for six queries that obtain influences of Taufer at distance 1 to 6, respectively. These queries are of the same form as the queries used in the first experiment. However, the main difference w.r.t. the first experiment is that the "`dbo:influencedBy` links" point only to data in DBpedia. In other words, every document that is reachable according to the reachability criterion c_{PPMatch} (and, thus, has to be retrieved under c_{PPMatch}-semantics) comes from the DBpedia Linked Data server. Hence, with this second experiment we wanted to capture a more dataset-centric scenario,

[5] http://dbpedia.org/resource/Veno_Taufer

while the first experiment has captured a scenario in which the data to be discovered during query execution is truly distributed all over the WWW. Another important difference is that the `dbo:influencedBy` links are bidirectional; that is, any triple with predicate `dbo:influencedBy` can be found in both the document for the subject IRI of the triple and the document for the object IRI.

Due to the availability of these bidirectional data links, the query results under both semantics are the same for each of the six queries (cf. Figure 8). In contrast, the `nderef` measurements differ significantly and present the same pattern as observed in the first experiment. In fact, the number of dereferencing operations necessary to guarantee complete results under c_{PPMatch}-semantics is even higher in the second experiment. We explain this observation by the fact that the strongly connected component established by the `dbo:influencedBy` links is bigger than the component of FOAF profiles in the first experiment. Apparently, this "fact" is known *only after* the corresponding traversal processes have been performed.

7.3. Discussion of the Experimental Results

Our experiments indicate that choosing one of the two tested query semantics over the other may have a significant impact in practice. Considering the size of query results first, our experiments show that there are cases in which the query result computed under the context-based semantics is smaller than under the (reachability-based) c_{PPMatch}-semantics. We explain this finding by two important properties that distinguish the context-based semantics from reachability-based semantics such as the c_{PPMatch}-semantics.

First, since it is based on the context selector (cf. Section 4.3), the context-based semantics ignores *all* the triples from any given document that have a subject IRI different from the IRI whose lookup resulted in retrieving the document. Ignoring such triples significantly decreases the number of paths (of triples) that can be found to match a given PP expression.

Second, the context-based semantics is designed to be very selective in the way the queried Web of Linked Data has to be traversed. More precisely, every traversal step is the result of first discovering a triple in the data of the current context document such that this triple can be used as a next step along a path that eventually may match the given PP expression. As a consequence of enforcing such a behavior, the traversal may not reach some documents that are reached under the c_{PPMatch}-semantics, and some of these documents may happen to contain triples that can be used to compute additional solutions under the c_{PPMatch}-semantics.

Our first experiment shows that this may happen in particular if the region of the Web that a query focuses on has a very heterogeneous link structure with many unidirectional links. On the other hand, if the link structure is more homogeneous, with mostly bidirectional links, then the query results under both semantics are more likely to coincide. Our second experiment presents an extreme case of such a scenario.

The downside of potentially larger query results that may be expected under c_{PPMatch}-semantics is a greater number of dereferencing operations, which implies longer execution times and more network traffic generated. Our experiments provide remarkable evidence that this problem is not negligible. That is, for every query in our experiments the difference w.r.t. the corresponding number of dereferencing operations under the context-based semantics is substantial (up to two orders of magnitude). The fact that we made this observation in both experiments also shows that a greater number of dereferencing operations under c_{PPMatch}-semantics is not a peculiarity of traversing an either more homogeneous or more heterogeneous link structure.

The significantly smaller number of dereferencing operations may be seen as a crucial advantage of the context-based semantics over the c_{PPMatch}-semantics. The flip side of course is that users of systems that implement the context-based semantics may see query results with less solutions. Hence, choosing among the two semantics is a question of whether a user is willing to accept the price of possibly having to retrieve many more documents (and, thus, longer execution times) for the chance of seeing a greater number of solutions.

8. Concluding Remarks

This paper studies the problem of extending the scope of the Property Paths feature in SPARQL to query Linked Data that is distributed on the WWW. We have investigated reachability-based query semantics, which decouple navigation from querying. Additionally, we have proposed a different interpretation for PPs over the Web via the context-based query semantics. An interesting finding regarding this latter semantics is that there exist queries whose evaluation over the WWW is not possible in practice. We studied this aspect using a notion of Web-safeness and introduced a decidable syntactic property for identifying queries that are Web-safe under the context-based semantics. Moreover, we have presented an experimental evaluation that compares the two semantics on different datasets showing that the context-based semantics incurs in a lower number of dereferencing operations that will have an impact on the running time.

We believe that the presented work provides valuable input to a wider discussion about defining how the SPARQL language can be used for accessing Linked Data on the WWW. There are several directions for future research including an investigation of the relationships between navigational queries and SPARQL federation, as well as an exploration of techniques based on which query execution systems may implement efficiently the machinery developed in this paper.

Acknowledgements

We thank the ESWC reviewers and the SWJ reviewers for their valuable feedback. Olaf Hartig's work has been funded by the German Government, Federal Ministry of Education and Research under the project number 03WKCJ4D. Giuseppe Pirrò's work has been funded by the Cyber Security Technological District financed by the Italian MIUR.

References

[1] S. Abiteboul and V. Vianu. Queries and computation on the web. *Theor. Comput. Sci.*, 239(2):231–255, 2000. doi:10.1016/S0304-3975(99)00221-2.

[2] F. Alkhateeb, J. Baget, and J. Euzenat. Extending SPARQL with regular expression patterns (for querying RDF). *J. Web Sem.*, 7(2):57–73, 2009. doi:10.1016/j.websem.2009.02.002.

[3] M. Arenas, S. Conca, and J. Pérez. Counting beyond a yottabyte, or how SPARQL 1.1 property paths will prevent adoption of the standard. In A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, and S. Staab, editors, *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 629–638. ACM, 2012. doi:10.1145/2187836.2187922.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. DBpedia: A nucleus for a web of open data. In K. Aberer, K. Choi, N. F. Noy, D. Allemang, K. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer, 2007. doi:10.1007/978-3-540-76298-0_52.

[5] T. Berners-Lee. Design Issues: Linked Data. Online at http://www.w3.org/DesignIssues/LinkedData.html, July 2006.

[6] P. Bouquet, C. Ghidini, and L. Serafini. Querying the Web of Data: A formal approach. In A. Gómez-Pérez, Y. Yu, and Y. Ding, editors, *The Semantic Web, Fourth Asian Conference, ASWC 2009, Shanghai, China, December 6-9, 2009. Proceedings*, volume 5926 of *Lecture Notes in Computer Science*, pages 291–305. Springer, 2009. doi:10.1007/978-3-642-10871-6_20.

[7] C. Buil-Aranda, M. Arenas, Ó. Corcho, and A. Polleres. Federating queries in SPARQL 1.1: Syntax, semantics and evaluation. *J. Web Sem.*, 18(1):1–17, 2013. doi:10.1016/j.websem.2012.10.001.

[8] R. Cyganiak, D. Wood, and M. Lanthaler, editors. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation, 25 February 2014. https://www.w3.org/TR/rdf11-concepts/.

[9] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. J. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, RFC Editor, June 1999. http://www.rfc-editor.org/rfc/rfc2616.txt.

[10] V. Fionda, C. Gutierrez, and G. Pirrò. Semantic navigation on the Web of Data: Specification of routes, web fragments and actions. In A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, and S. Staab, editors, *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 281–290. ACM, 2012. doi:10.1145/2187836.2187875.

[11] V. Fionda, G. Pirrò, and M. P. Consens. Extended property paths: Writing more SPARQL queries in a succinct way. In B. Bonet and S. Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 102–108. AAAI Press, 2015. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9661.

[12] V. Fionda, G. Pirrò, and C. Gutierrez. NautiLOD: A formal language for the Web of Data graph. *TWEB*, 9(1):5:1–5:43, 2015. doi:10.1145/2697393.

[13] D. Florescu, A. Y. Levy, and A. O. Mendelzon. Database techniques for the World-Wide Web: A survey. *SIGMOD Record*, 27(3):59–74, 1998. doi:10.1145/290593.290605.

[14] J. Golbeck and M. Rothstein. Linking social networks on the web with FOAF: A Semantic Web case study. In D. Fox and C. P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 1138–1143. AAAI Press, 2008. http://www.aaai.org/Library/AAAI/2008/aaai08-180.php.

[15] S. Harris and A. Seaborne, editors. *SPARQL 1.1 Query Language*. W3C Recommendation, 21 March 2013. https://www.w3.org/TR/sparql11-query/.

[16] A. Harth and S. Speiser. On completeness classes for query evaluation on linked data. In J. Hoffmann and B. Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.* AAAI Press, 2012. http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/5114.

[17] O. Hartig. How caching improves efficiency and result completeness for querying linked data. In C. Bizer, T. Heath, T. Berners-Lee, and M. Hausenblas, editors, *WWW2011 Workshop on Linked Data on the Web, Hyderabad, India, March 29, 2011*, volume 813 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011. http://ceur-ws.org/Vol-813/ldow2011-paper05.pdf.

[18] O. Hartig. SPARQL for a web of linked data: Semantics and computability. In E. Simperl, P. Cimiano, A. Polleres, Ó. Corcho, and V. Presutti, editors, *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*, volume 7295 of *Lecture Notes in Computer Science*, pages 8–23. Springer, 2012. doi:10.1007/978-3-642-30284-8_8.

[19] O. Hartig, C. Bizer, and J. C. Freytag. Executing SPARQL queries over the web of linked data. In A. Bernstein, D. R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, and K. Thirunarayan, editors, *The Semantic Web - ISWC 2009, 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings*, volume 5823 of *Lecture Notes in Computer Science*, pages 293–309. Springer, 2009. doi:10.1007/978-3-642-04930-9_19.

[20] O. Hartig and J. Pérez. LDQL: A query language for the web of linked data. In M. Arenas, Ó. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. T. Groth, M. Dumontier, J. Heflin, K. Thirunarayan, and S. Staab, editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, volume 9366 of *Lecture Notes in Computer Science*, pages 73–91. Springer, 2015. doi:10.1007/978-3-319-25007-6_5.

[21] O. Hartig and G. Pirrò. A context-based semantics for SPARQL property paths over the web. In F. Gandon, M. Sabou, H. Sack, C. d'Amato, P. Cudré-Mauroux, and A. Zimmermann, editors, *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings*, volume 9088 of *Lecture Notes in Computer Science*, pages 71–87. Springer, 2015. doi:10.1007/978-3-319-18818-8_5.

[22] O. Hartig and G. Pirrò. A context-based semantics for SPARQL property paths over the web (extended version). *CoRR*, abs/1503.04831, 2015. http://arxiv.org/abs/1503.04831.

[23] K. Kochut and M. Janik. SPARQLeR: Extended SPARQL for semantic association discovery. In E. Franconi, M. Kifer, and W. May, editors, *The Semantic Web: Research and Applications, 4th European Semantic Web Conference, ESWC 2007, Innsbruck, Austria, June 3-7, 2007, Proceedings*, volume 4519 of *Lecture Notes in Computer Science*, pages 145–159. Springer, 2007. doi:10.1007/978-3-540-72667-8_12.

[24] D. Konopnicki and O. Shmueli. Information gathering in the world-wide web: The W3QL query language and the W3QS system. *ACM Trans. Database Syst.*, 23(4):369–410, 1998. doi:10.1145/296854.277639.

[25] E. V. Kostylev, J. L. Reutter, M. Romero, and D. Vrgoc. SPARQL with property paths. In M. Arenas, Ó. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. T. Groth, M. Dumontier, J. Heflin, K. Thirunarayan, and S. Staab, editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, volume 9366 of *Lecture Notes in Computer Science*, pages 3–18. Springer, 2015. doi:10.1007/978-3-319-25007-6_1.

[26] A. Letelier, J. Pérez, R. Pichler, and S. Skritek. Static analysis and optimization of Semantic Web queries. *ACM Trans. Database Syst.*, 38(4):25, 2013. doi:10.1145/2500130.

[27] K. Losemann and W. Martens. The complexity of evaluating path expressions in SPARQL. In M. Benedikt, M. Krötzsch, and M. Lenzerini, editors, *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 101–112. ACM, 2012. doi:10.1145/2213556.2213573.

[28] A. O. Mendelzon, G. A. Mihaila, and T. Milo. Querying the World Wide Web. *Int. J. on Digital Libraries*, 1(1):54–67, 1997. doi:10.1007/s007990050004.

[29] R. Meusel, P. Mika, and R. Blanco. Focused crawling for structured data. In J. Li, X. S. Wang, M. N. Garofalakis, I. Soboroff, T. Suel, and M. Wang, editors, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 1039–1048. ACM, 2014. doi:10.1145/2661829.2661902.

[30] J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. *ACM Trans. Database Syst.*, 34(3), 2009. doi:10.1145/1567274.1567278.

[31] J. Pérez, M. Arenas, and C. Gutierrez. nSPARQL: A navigational language for RDF. *J. Web Sem.*, 8(4):255–270, 2010. doi:10.1016/j.websem.2010.01.002.

[32] J. L. Reutter, A. Soto, and D. Vrgoc. Recursion in SPARQL. In M. Arenas, Ó. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. T. Groth, M. Dumontier, J. Heflin, K. Thirunarayan, and S. Staab, editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, volume 9366 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2015. doi:10.1007/978-3-319-25007-6_2.

[33] S. Schaffert, C. Bauer, T. Kurz, F. Dorschel, D. Glachs, and M. Fernandez. The Linked Media Framework: Integrating and interlinking enterprise media content and data. In V. Presutti and H. S. Pinto, editors, *I-SEMANTICS 2012 - 8th International Conference on Semantic Systems, I-SEMANTICS '12, Graz, Austria, September 5-7, 2012*, pages 25–32. ACM, 2012. doi:10.1145/2362499.2362504.

[34] M. Schmidt, M. Meier, and G. Lausen. Foundations of SPARQL query optimization. In L. Segoufin, editor, *Database Theory - ICDT 2010, 13th International Conference, Lausanne, Switzerland, March 23-25, 2010, Proceedings*, ACM International Conference Proceeding Series, pages 4–33. ACM, 2010. doi:10.1145/1804669.1804675.

[35] P. A. Szekely, C. A. Knoblock, J. Slepicka, A. Philpot, A. Singh, C. Yin, D. Kapoor, P. Natarajan, D. Marcu, K. Knight, D. Stallard, S. S. Karunamoorthy, R. Bojanapalli, S. Minton, B. Amanatullah, T. Hughes, M. Tamayo, D. Flynt, R. Artiss, S. Chang, T. Chen, G. Hiebel, and L. Ferreira. Building and using a knowledge graph to combat human trafficking. In M. Arenas, Ó. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. T. Groth, M. Dumontier, J. Heflin, K. Thirunarayan, and S. Staab, editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*, volume 9367 of *Lecture Notes in Computer Science*, pages 205–221. Springer, 2015. doi:10.1007/978-3-319-25010-6_12.

[36] T. T. Tang, D. Hawking, N. Craswell, and K. Griffiths. Focused crawling for both topical relevance and quality of medical information. In O. Herzog, H. Schek, N. Fuhr, A. Chowdhury, and W. Teiken, editors, *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005*,

pages 147–154. ACM, 2005. doi:10.1145/1099554.1099583.

[37] D. Toman and G. E. Weddell. *Fundamentals of Physical Design and Query Compilation*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011. doi:10.2200/S00363ED1V01Y201105DTM018.

[38] J. Umbrich, A. Hogan, A. Polleres, and S. Decker. Link traversal querying for a diverse web of data. *Semantic Web*, 6(6):585–624, 2015. doi:10.3233/SW-140164.

[39] R. Verborgh, O. Hartig, B. De Meester, G. Haesendonck, L. De Vocht, M. Vander Sande, R. Cyganiak, P. Colpaert, E. Mannens, and R. Van de Walle. Querying datasets on the Web with high availability. In P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. A. Knoblock, D. Vrandecic, P. T. Groth, N. F. Noy, K. Janowicz, and C. A. Goble, editors, *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, volume 8796 of *Lecture Notes in Computer Science*, pages 180–196. Springer, 2014. doi:10.1007/978-3-319-11964-9_12.

[40] R. Verborgh, M. Vander Sande, O. Hartig, J. Van Herwegen, L. De Vocht, B. De Meester, G. Haesendonck, and P. Colpaert. Triple pattern fragments: A low-cost knowledge graph interface for the web. *J. Web Sem.*, 37–38:184–206, 2016. doi:10.1016/j.websem.2016.03.003.

[41] P. T. Wood. Query languages for graph databases. *SIGMOD Record*, 41(1):50–60, 2012. doi:10.1145/2206869.2206879.

Appendix A: Queries used in the Evaluation

This appendix provides the queries used in our experiment. These queries use the following prefixes:

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/>
```

8.1. Distance 1 Query for the Experiment on D1

```
SELECT ?end WHERE {
   <http://nunolopes.org/foaf.rdf\#me> foaf:knows ?end
}
```

8.2. Distance 2 Query for the Experiment on D1

```
SELECT ?end WHERE {
   <http://nunolopes.org/foaf.rdf\#me> foaf:knows/foaf:knows ?end
}
```

8.3. Distance 3 Query for the Experiment on D1

```
SELECT ?end WHERE {
   <http://nunolopes.org/foaf.rdf\#me>
        foaf:knows/foaf:knows/foaf:knows ?end
}
```

8.4. Distance 4 Query for the Experiment on D1

```
SELECT ?end WHERE {
   <http://nunolopes.org/foaf.rdf\#me>
        foaf:knows/foaf:knows/foaf:knows/foaf:knows ?end
}
```

8.5. Distance 5 Query for the Experiment on D1

```
SELECT ?end WHERE {
   <http://nunolopes.org/foaf.rdf\#me>
        foaf:knows/foaf:knows/foaf:knows/foaf:knows/foaf:knows ?end
}
```

8.6. Distance 6 Query for the Experiment on D1

```
SELECT ?end WHERE {
   <http://nunolopes.org/foaf.rdf\#me>
        foaf:knows/foaf:knows/foaf:knows/foaf:knows/foaf:knows/foaf:knows ?end
}
```

8.7. Distance 1 Query for the Experiment on D2

```
SELECT ?end WHERE {
   <http://dbpedia.org/resource/Veno_Taufer> dbo:influencedBy ?end
}
```

8.8. Distance 2 Query for the Experiment on D2

```
SELECT ?end WHERE {
   <http://dbpedia.org/resource/Veno_Taufer> dbo:influencedBy/dbo:influencedBy ?end
}
```

8.9. Distance 3 Query for the Experiment on D2

```
SELECT ?end WHERE {
   <http://dbpedia.org/resource/Veno_Taufer>
        dbo:influencedBy/dbo:influencedBy/dbo:influencedBy ?end
}
```

8.10. Distance 4 Query for the Experiment on D2

```
SELECT ?end WHERE {
   <http://dbpedia.org/resource/Veno_Taufer>
        dbo:influencedBy/dbo:influencedBy/dbo:influencedBy/dbo:influencedBy ?end
}
```

8.11. Distance 5 Query for the Experiment on D2

```
SELECT ?end WHERE {
   <http://dbpedia.org/resource/Veno_Taufer>
        dbo:influencedBy/dbo:influencedBy/dbo:influencedBy/
                               dbo:influencedBy/dbo:influencedBy ?end
}
```

8.12. Distance 6 Query for the Experiment on D2

```
SELECT ?end WHERE {
   <http://dbpedia.org/resource/Veno_Taufer>
        dbo:influencedBy/dbo:influencedBy/dbo:influencedBy/
                               dbo:influencedBy/dbo:influencedBy/dbo:influencedBy ?end
}
```