Formal Verification of Data Provenance Records

Szymon Klarman[1], Stefan Schlobach[1], Luciano Serafini[2]

[1] VU University Amsterdam, The Netherlands,
{s.klarman, k.s.schlobach}@vu.nl
[2] Fondazione Bruno Kessler, Trento, Italy,
serafini@fbk.eu

Abstract. Data provenance is the history of derivation of a data artifact from its original sources. As the real-life provenance records can likely cover thousands of data items and derivation steps, one of the pressing challenges becomes development of formal frameworks for their automated verification.
In this paper, we consider data expressed in standard Semantic Web ontology languages, such as OWL, and define a novel verification formalism called *provenance specification logic*, building on dynamic logic. We validate our proposal by modeling the test queries presented in The First Provenance Challenge, and conclude that the logic core of such queries can be successfully captured in our formalism.

1 Introduction

In this paper, we propose and study a novel logic-based approach to formal verification of data provenance records in the Semantic Web environment.

Motivation Data provenance is the history of derivation of a data artifact from its original sources [1, 2]. A provenance record stores all the steps and contextual aspects of the entire derivation process, including the precise sequence of operations executed, their inputs, outputs, parameters, the supplementary data involved, etc., so that third parties can unambiguously interpret the final data product in its proper context. It has been broadly acknowledged that provenance information is crucial for facilitating reuse, management and reproducibility of published data [3, 1]. For instance, the ability of verifying whether past experiments conformed to some formal criteria is a key in the process of validation of eScientific results [4]. As provenance records can cover thousands of data items and derivation steps, one of the pressing challenges becomes development of formal frameworks and methods to automate verification. Such a logic back-end for practical reasoning tools could, e.g. be useful for provenance-driven data querying, or for validating conformance of provenance records to formal specifications.

Let us consider a concrete example taken from The First Provenance Challenge (FPC) — a community effort aimed at understanding the capabilities of available provenance systems [5]. In FPC, 17 teams competed in answering 9 queries over data provenance records (see Figure 1) obtained from executing a

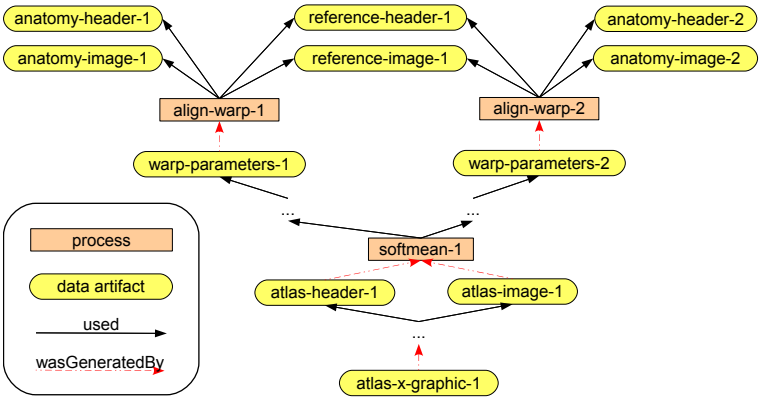

Fig. 1. A data provenance record describing a run of the FPC workflow [5].

real-life scientific workflow for creating population-based "brain atlases" of high resolution anatomical data. One representative task was to:

Q6. *Find all output averaged images of softmean (average) procedures, where the warped images taken as input were align warp'ed using a twelfth order nonlinear 1365 parameter model, i.e. where softmean was preceded in the workflow, directly or indirectly, by an align warp procedure with argument -m 12.*

A distinctive feature of this sort of queries is their inherent two-dimensionality: the domain data (here: image identifiers) is queried relative to its meta-level provenance description. To date all existing approaches to support such queries are based on ad hoc combinations of techniques and formalisms, dependent on the internal representation structures, and are procedural in nature. Given the semantic character of the task, and in light of the soon to be expected standardization of the Provenance vocabularies by W3C,[3] a logic-based language for querying and verifying provenance graphs, which could significantly improve reusability and generalisability, is critically missing. This paper closes this gap.

Methodology We introduce *provenance specification logic* (PSL^M) which, to the best of our knowledge, offers the first systematic view on the logical foundations of formal verification of data provenance records. Our focus is on data expressed in the Semantic Web ontology languages, such as OWL and RDF(S), whose formal core is essentially captured by Description Logics (DLs) [6], underpinning the Semantic Web architecture.

The basic idea is very intuitive. A data provenance record is represented as a directed graph, with certain nodes being treated as identifiers for datasets, containing the data involved in the respective stages of the computation. We construct the basic variant of our logic, called PSL, by substituting atoms of Propositional Dynamic Logic (PDL) with queries belonging to a selected query

[3] See `http://www.w3.org/2011/prov/wiki/Main_Page`.

language. The dynamic component, thus inherited from PDL, enables expressing complex provenance patterns, while the embedded queries support access to data artifacts. In the second step, we lift this approach to cater for scenarios in which provenance graphs are themselves described in dedicated provenance ontologies. This way, we obtain the target formalism PSL^M, which, on top of the functionalities offered by PSL, also facilitates the use of a rich metalanguage.

This mechanism is highly independent from the employed representation formalisms, and can be reused in a plug-and-play fashion for a number of combinations of ontology–query languages. Moreover, we demonstrate that PSL^M is computationally well-behaved. By separating the DL-level reasoning tasks from the pure model checking of provenance graphs, we obtain a PTIME-completeness result, carried over from the model checking problem in PDL, which remains invariant to the particular choice of the employed ontology/query languages.

Contents In this work we deliver three main contributions: 1) We introduce PSL^M, a declarative language for expressing complex constraints over data provenance records. 2) By systematically studying the collection of test queries from FPC, mentioned above, we show that PSL^M offers desired modeling capabilities. 3) Finally, we provide a computational analysis of the approach, and report on some satisfying results.

In the remainder of this paper, we first give a short overview of the related work (Section 2) and preliminary notions (Section 3). Next, we incrementally introduce PSL^M (Sections 4, 5) and validate it against the test queries (Section 6). Finally, we study the computational aspects of our framework (Section 7).

2 Related Work

Provenance is nowadays recognized as one of the critical problems to be addressed by the Semantic Web community, attracting increasing interest, e.g. [7]. Existing Semantic Web-based approaches to the problem of verification and querying, such as [8] are persistently technology-driven, and employ combinations of web services, ontologies, triple stores, SPARQL queries, etc. and fail to lay down systematic perspectives on the formal foundations of the problem. Noteworthy exceptions are [9] and [10] which provide, respectively: reproducibility semantics, which are executional in nature, and logic program-based framework for reasoning with a provenance-annotated linked data, where both annotations and data language are specifically restricted. Our paper goes beyond those proposals by providing a cohesive declarative semantic framework based on standard logic and ontology languages, and rich metamodels.

On the formal level, the problem of provenance verification bears a strong resemblance to the traditionally studied verification of transition systems, which in principle encourages the use of similar logic-based techniques [11]. This analogy, however, must be treated with caution. While in usual transition systems states represent complete, propositional abstractions of system's configurations, in the data provenance context states are effectively datasets, reflecting the knowledge

of the system in a certain configuration. This creates a need for more expressive verification formalisms, extending the basic program logics, such as PDL [12]. Even Dynamic DLs [13], which are capable of modeling transition systems with states corresponding to DL knowledge bases, are not flexible enough to express rich constraints on the data level. Some other verification formalisms, of a more suitable, data-oriented flavor, have been proposed for verification of data-driven systems [14], knowledge base programs [15], or workflow schemas [16]. However, the central motivation behind their design is to enable representation of all permissable data-altering operations over a fixed data language, with the aim of studying general properties of programs composed of such operations. Consequently, the considered representation languages are strongly restricted in order to ensure decidability of those properties. Such general problems, however, are not of a primary interest in our case, since a provenance record describes by definition a single, completed computation process, which one wants to study ex-post. Hence, rather than abstracting from the richness of a given system and focusing on its possible behaviors, we must enable reasoning machinery which can maximally utilize the available information about the system.

3 Preliminaries

For clarity of exposition, in this paper we consider data represented and managed within the framework of *Description Logics* (DLs), noting that all claims made in this context extend naturally to arbitrary fragments of OWL/RDF(S) languages. In what follows, we briefly recap the preliminaries regarding DLs, and further formalize the notion of data provenance record.

3.1 Description Logics

We use the standard nomenclature and notation for the syntax and semantics of DLs. We refer the reader to [6] for full details. A DL language $\mathcal{L}$ is specified by a vocabulary $\Sigma = (N_C, N_R, N_I)$, where N_C is a set of concept names, N_R a set of role names and N_I a set of individual names, and by a selection of logical operators enabling construction of complex concepts, roles and axioms. Different combinations of operators give rise to DLs of different expressiveness and computational complexity, from the highly expressive $\mathcal{SROIQ}$, underpinning the OWL 2 DL language, to the lightweight $\mathcal{EL}^{++}$ or the DL-*Lite* family, on which tractable OWL profiles are based [17]. For instance, the DL $\mathcal{ALCO}$, a substantial fragment of OWL 2 DL, permits the following constructors:

Concepts:

$$\top \mid \bot \mid A \in N_C \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists r.C \mid \forall r.C \mid \{a\}$$

Axioms:

$$C \sqsubseteq D \mid C(a) \mid r(a, b)$$

where $r \in N_R$, $a, b \in N_I$ and C, D are (possibly complex) concepts.

From the data management perspective, a set of concept inclusions $C \sqsubseteq D$ is typically considered an ontology which provides access to instance data,

represented as a set of assertions of the form $A(a), r(a,b)$ [18]. We implicitly abide by this distinction, but for simplicity refer rather to the general notion of a DL knowledge base.

Definition 1 (Knowledge base). *A* knowledge base $\mathcal{K}$ *over a DL language* $\mathcal{L}$ *is a finite set of axioms allowed by the syntax of* $\mathcal{L}$*. The set of all knowledge bases over* $\mathcal{L}$ *is denoted by* $\mathbf{K}(\mathcal{L})$*.*

The semantics of $\mathcal{L}$ is given in terms of the usual model-theoretic interpretations. An interpretation $\mathcal{I}$ is a model of a knowledge base *iff* it satisfies all its axioms. We say that an axiom ϕ is *entailed* by a knowledge base $\mathcal{K}$, denoted as $\mathcal{K} \models \phi$ *iff* ϕ is satisfied in every model $\mathcal{I}$ of $\mathcal{K}$.

Further, we recall the notion of conjunctive queries (CQs) — the most popular class of first-order queries studied in the context of DLs [19]. The problem of answering CQs is known to be decidable for most DLs, and is effectively handled by existing tools, such as DL reasoners (e.g. Pellet) or, as in case of DL-*Lite* family, relational database management systems (e.g. Mastro[4]). Let N_V be a countably infinite set of variables. A *conjunctive query* over a DL language $\mathcal{L}$ with the vocabulary $\Sigma = (N_C, N_R, N_I)$ is a first-order formula $\phi = \exists \vec{y}.q(\vec{x}, \vec{y})$, where $\vec{x}, \vec{y} \in N_V$ are sequences of variables and q is a conjunction of atoms over Σ. The free variables $\vec{x}$ occurring in ϕ are also called the answer variables and denoted as $\mathrm{avar}(\phi)$. For a CQ ϕ, a ϕ-substitution is a mapping $\mu : \mathrm{avar}(\phi) \mapsto N_I$. We write $\mu(\phi)$ to denote the formula resulting from applying μ to ϕ. We call μ a *certain answer* to ϕ w.r.t. a knowledge base $\mathcal{K}$, whenever $\mathcal{K} \models \mu(\phi)$, i.e. whenever $\mu(\phi)$ is satisfied in all models $\mathcal{I}$ of $\mathcal{K}$.

3.2 Data Provenance Records

The definition of a provenance record that we adopt here, and further refine in Section 5, is the simplest abstraction of the proposals currently discussed in the course of a W3C standardization effort. Those proposals, building largely on the specification of the Open Provenance Model [2], consider a provenance record to be a basic graph structure (such as presented in Figure 1) representing the whole documented history of interactions between processes and data artifacts during a certain computation, where data artifacts are in fact datasets (knowledge bases) expressed in DLs. The choice of the OPM foundations for our approach is motivated largely by the fact that OPM is suggested as the intended formalism for representing provenance in the expected W3C recommendation. In principle, however, the level of abstraction which we endorse here goes beyond particular, concrete encodings of provenance information, and builds only on generic provenance notions present also in other formalisms used for recording provenance, such as Proof Markup Language [20, 21]. Crucially, our approach generalizes over any (transition) graph-based representation of data provenance.

A *directed graph* is a pair (V, E), where V is a non-empty set of nodes and E is a set of ordered pairs from $V \times V$, called edges. A *bipartite graph* is a

[4] See `http://www.dis.uniroma1.it/quonto/`.

graph $(V \cup W, E)$, where $V \cup W$ is a set of nodes and E a set of edges such that $E \subseteq V \times W \cup W \times V$. An *edge-labeled graph* is a triple (V, E, l), such that (V, E) is a graph and $l : E \mapsto R$ assigns a relation name from a set R to every edge in E. A graph (V, E) is called *acyclic iff* for every node $v \in V$, there exists no sequence $w_1, \ldots, w_n \in V$, such that $(v, w_1), \ldots, (w_{n-1}, w_n), (w_n, v) \in E$.

Definition 2 (Provenance graph). *An* $\mathcal{L}$-provenance graph *is a tuple* $G = (P, D, E, l, k)$*, where* $(P \cup D, E, l)$ *is a bipartite, directed, acyclic, edge-labeled graph, and* k *is a function* $k : D \mapsto \mathbf{K}(\mathcal{L})$*. The nodes in* P *are called* processes *and in* D data artifacts*.*

By convention, we identify process nodes with unique process invocations that occurred during the recorded computation, and data artifact nodes with the corresponding DL knowledge bases $\{k(d) \mid d \in D\}$ that were involved. Note, that we do not presume any specific causal relationships between the represented entities. We are only interested in the formal properties of the graphs.

4 Provenance Specification Logic

Formal verification is the task of checking whether a certain formal structure satisfies the property described by a given formula of a dedicated specification language. The properties of data provenance records which we aim to capture here are essentially complex relationships between the structural patterns occurring in the provenance graphs and the contents of data artifacts. Three typical constraints, representative of most reasoning tasks requested from practical provenance systems [3–5], are e.g.:

1. $r(a, b)$ holds in data artifact d_1 and d_1 is reachable via edge *succeeds* from processes p_1 and p_2,
2. a data artifact in which $D(a)$ does not hold is reachable via a finite sequence of two-step edge compositions *wasGeneratedBy-used* from a data artifact in which $D(a)$ holds,
3. if $D(a)$ holds in any data artifact related to process p_1 via either $input_1$ or $input_2$, then p_1 must be related via *output* to some data artifact in which $r(a, y)$ holds, for some arbitrary y.

These informally stated properties are clearly satisfied by the respective provenance graphs, illustrated in Figure 2, where nodes p_1, p_2 represent process nodes, and d_1, d_2, d_3 data artifacts, whose contents are listed inside the nodes.

The ability of expressing constraints of this flavor is the key feature of a big family of program verification formalisms based on dynamic logics, in particular the prominent Propositional Dynamic Logic (PDL) [12]. The *provenance specification logic* (PSL), which we introduce below, is a data-oriented extension of PDL. Essentially, we substitute propositional letters of PDL formulas with queries belonging to a certain query language. The dynamic component of PSL enables explicit modeling of requested provenance patterns, while the queries allow for accessing the contents of data artifacts. The choice of an adequate query

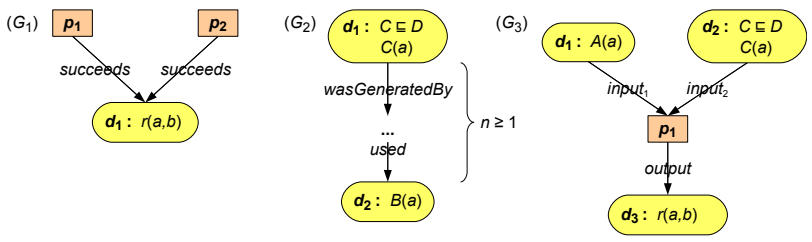

Fig. 2. Sample provenance graphs.

language is in principle an application-driven decision, depending strongly on the underlying data language. For instance, if data artifacts use RDF(S) representation, a natural candidate is SPARQL [22]. As our focus is on the general DL setup, we consider the class of conjunctive queries, as introduced in Section 3.

Definition 3 (PSL: syntax). *Let $G = (P, D, E, l, k)$ be an $\mathcal{L}$-provenance graph and R the set of relation names used in G. Then the* provenance specification language *over G is the smallest language induced by the following grammar:*

Object queries:

$$\phi \;:=\; \textit{CQs over } \mathcal{L}$$

Path expressions:

$$\pi \;:=\; r \mid \pi;\pi \mid \pi \cup \pi \mid \pi^{-} \mid \pi^{*} \mid v? \mid \alpha?$$

where $r \in R$ and $v \in P \cup D$,

Provenance formulas:

$$\alpha \;:=\; \{\phi\} \mid \top \mid \langle\pi\rangle\alpha \mid \alpha \wedge \alpha \mid \neg\alpha$$

Whenever convenient we use the usual abbreviations $\bot = \neg\top$, $[\pi] = \neg\langle\pi\rangle\neg$, $\alpha \vee \beta = \neg(\neg\alpha \wedge \neg\beta)$ and $\alpha \rightarrow \beta = \neg\alpha \vee \beta$.

Following the CQ notation, by $\mathrm{avar}(\alpha)$ we denote the set of all free (answer) variables occurring in a provenance formula α, i.e. the union of all answer variables from the CQs embedded in α. Note, that different CQs are allowed to share same answer variables. This way one can capture interesting data dependencies between the contents of data artifacts. An α-substitution is a mapping $\mu : \mathrm{avar}(\alpha) \mapsto N_I$ and $\mu(\alpha)$ denotes the result of applying μ to α. We say that $\mu(\alpha)$ is *satisfied* in an $\mathcal{L}$-provenance graph $G = (P, D, E, l, k)$ in a node $v \in P \cup D$ *iff* $G, v \Vdash \mu(\alpha)$, where the *satisfaction relation* $\Vdash$ is defined as follows.

Definition 4 (PSL: semantics). *The* satisfaction relation $\Vdash$ *for PSL formulas is given by a simultaneous induction over the structure of provenance formulas and path expressions,* w.r.t. *a substitution μ. For an $\mathcal{L}$-provenance graph $G = (P, D, E, l, k)$ and every $v, w \in P \cup D$:*

Provenance formulas:

$G, v \Vdash \{\phi\}$ iff $v \in D$ *and* $k(v) \models \mu(\phi)$,
$G, v \Vdash \top$,
$G, v \Vdash \langle\pi\rangle\alpha$ iff *there exists* $w \in P \cup D$, *s.t.* $G \Vdash v \xrightarrow{\pi} w$ *and* $G, w \Vdash \alpha$,

$G, v \Vdash \alpha \wedge \beta$ iff $G, v \Vdash \alpha$ *and* $G, v \Vdash \beta$,
$G, v \Vdash \neg\alpha$ iff $G, v \nVdash \alpha$,

Path expressions:

$G \Vdash v \xrightarrow{r} w$ iff $(v, w) \in E$ *and* $l(v, w) = r$,
$G \Vdash v \xrightarrow{\pi;\sigma} w$ iff *there is* $u \in P \cup D$ *s.t.* $G \Vdash v \xrightarrow{\pi} u$ *and* $G \Vdash u \xrightarrow{\sigma} w$,
$G \Vdash v \xrightarrow{\pi\cup\sigma} w$ iff $G \Vdash v \xrightarrow{\pi} w$ *or* $G \Vdash v \xrightarrow{\sigma} w$,
$G \Vdash v \xrightarrow{\pi^-} w$ iff $G \Vdash w \xrightarrow{\pi} v$,
$G \Vdash v \xrightarrow{\pi^*} w$ iff $v(\xrightarrow{\pi})^* w$, *where* $(\xrightarrow{\pi})^*$ *is the transitive reflexive closure of* $\xrightarrow{\pi}$ *on* G,
$G \Vdash v \xrightarrow{v?} v$,
$G \Vdash v \xrightarrow{\alpha?} v$ iff $G, v \Vdash \alpha$.

Observe, that unlike in typical transition systems, only selected nodes in provenance graphs — exactly the data artifacts in D — represent the states over which object queries can be evaluated. Irrespective of this deviation, the model checking problem, underlying formal verification tasks, is defined as usual.

Model Checking 1 (PSL formulas) *Given an $\mathcal{L}$-provenance graph $G = (P, D, E, l, k)$, a node $v \in P \cup D$, a PSL provenance formula α and an α-substitution μ, decide whether $G, v \Vdash \mu(\alpha)$.*

It is easy to check that the following PSL formulas express precisely the properties from the three examples presented in the opening of this section, and are satisfied by the specified graphs, nodes and substitutions (Figure 2):

1. $\alpha := \langle p_1?; \mathit{succeeds}; d_1?\rangle(\{r(x, y)\} \wedge \langle \mathit{succeeds}^-; p_2?\rangle\top)$,
   where $G_1, p_1 \Vdash \mu(\alpha)$ for $\mu = \{x \mapsto a, y \mapsto b\}$,
2. $\alpha := \{D(x)\} \wedge \langle(\mathit{wasGeneratedBy}; \mathit{used})^*\rangle\neg\{D(x)\}$,
   where $G_2, d_1 \Vdash \mu(\alpha)$ for $\mu = \{x \mapsto a\}$,
3. $\alpha := \langle p_1?\rangle(\langle(\mathit{input}_1 \cup \mathit{input}_2)^-\rangle\{D(x)\} \rightarrow \langle \mathit{output}\rangle\{\exists y.r(x, y)\})$,
   where $G_3, p_1 \Vdash \mu(\alpha)$ for $\mu = \{x \mapsto a\}$.

For a more practical illustration, we model two use-cases from the eScience domain. The first one illustrates a typical problem of provenance-based validation of an eScience experiment, reported in [4].

Example 1. *A bioinformatician, B, downloads a file containing sequence data from a remote database. B then processes the sequence using an* analysis service. *Later, a reviewer, R, suspects that the sequence may have been a* nucleotide *sequence but processed by a service that can only analyze meaningfully* amino acid *sequences. R determines whether this was the case.*

$$\alpha := \{\exists y.\text{Sequence}(x) \wedge \text{analysis-result}(x, y)\} \rightarrow$$
$$[\mathit{output}; \mathit{analysis\text{-}service}?; \mathit{input}](\{\text{Amino-acid}(x)\} \wedge \neg\{\text{Nucleotide}(x)\})$$

Solution: The requested property is satisfied by a graph G if $G, v \Vdash \mu(\alpha)$ for every $v \in P \cup D$ and α-substitution $\mu = \{x \mapsto a\}$, where a is the sequence in question. Naturally, we presume a certain underlying representation model, where e.g. *analysis-service* is the name of the cited service, the result of analysis is given via an axiom of type analysis-result(a, y), etc. For lack of space we do not sketch such models for any of the remaining examples, relying on a proper reconstruction by the reader.

As the second example, we formalize one of the queries from FPC [5].

Example 2. See Q6 in Section 1 (*cf.* Figure 1).

$$\alpha := \{\text{Image}(x)\} \wedge \langle \mathit{wasGeneretedBy}; \mathit{softmean}_{1\ldots n}; \mathit{used}\rangle(\{\exists y.\text{Image}(y)\} \wedge \langle(\mathit{wasGeneratedBy}; \mathit{used})^*; \mathit{wasGeneratedBy}; \mathit{align\text{-}warp}_{1\ldots m}\rangle\top)$$

where $\mathit{softmean}_{1\ldots n} := \mathit{softmean}_1? \cup \ldots \cup \mathit{softmean}_n?$ includes all invocations of *softmean* process in the graph, while $\mathit{align\text{-}warp}_{1\ldots m} := \mathit{align\text{-}warp}_1? \cup \ldots \cup \mathit{align\text{-}warp}_m?$ all invocations of *align warp* with the specified parameter value.

Solution: For every $v \in P \cup D$ and α-substitution μ, if $G, v \Vdash \mu(\alpha)$, then μ is a requested resource.

Observe, that in the latter example not all information requested in the query can be expressed is the PSL formula in a direct, declarative manner. Namely, the selection of *softmean* and *align warp* invocations has to be encoded by an exhaustive enumeration of all the nodes satisfying the specified description. This shortcoming, which affects the high-level modeling capabilities of our formalism, is exactly what motivates the extension introduced in the next section.

5 Provenance Metalanguage

In practice, the relevant provenance information can be much richer than reflected in our abstract notion of provenance graphs. Typically, provenance records account also for the execution context of all processes, including their parametrization, time, responsible actors, etc. [1–3]. Moreover, they use complex taxonomies for classifying all these resources. Consequently, the structure of a provenance graph, along the accompanying contextual information, is commonly expressed by means of another DL-based language, used orthogonally to that representing the contents of data artifacts [23, 3]. For instance, the provenance graph G implicitly referred to in Example 2, would be likely represented as a knowledge base containing, among others, the axioms listed in Table 1. Formally, we define such a meta-level representation of provenance graphs as follows.

Definition 5 (Metalanguage, metaknowledge base). *Let $G = (P, D, E, l, k)$ be an $\mathcal{L}$-provenance graph and R the set of relation names used in G. Let $\mathcal{L}_G$ be a DL language with the vocabulary $\Gamma = (M_C, M_R, M_I)$, such that $R \subseteq M_R$ and $P \cup D \subseteq M_I$, and $\mathcal{K}_G$ a knowledge base over $\mathcal{L}_G$. Then, $\mathcal{L}_G$ is called the* metalanguage *and $\mathfrak{K}_G = (\mathcal{K}_G, k)$ the* metaknowledge base *over G* iff *the following conditions are satisfied:*

Table 1. A DL knowledge base encoding (part of) a provenance graph.

$\mathit{Artifact} \sqsubseteq \neg \mathit{Process}$	$\mathit{Softmean} \sqsubseteq \mathit{Process}$
$\mathit{Artifact} \sqsubseteq \forall \mathit{wasGeneratedBy}.\mathit{Process}$	$\mathit{Align\text{-}warp} \sqsubseteq \mathit{Process}$
$\mathit{Process} \sqsubseteq \forall \mathit{used}.\mathit{Artifact}$	$\mathit{Align\text{-}warp} \sqsubseteq \exists \mathit{hasArgValue}.\texttt{String}$
$\mathit{Softmean}(\mathit{softmean}_i)$	- for every node $\mathit{softmean}_i$
$\mathit{Align\text{-}warp}(\mathit{align\text{-}warp}_i)$	- for every node $\mathit{align\text{-}warp}_i$
$\mathit{hasArgValue}(\mathit{align\text{-}warp}_i$, "-m 12")	- for every node $\mathit{align\text{-}warp}_i$ corresponding to an invocation of *align warp* with argument "-m 12"

1. $D = \{v \in M_I \mid \mathcal{K}_G \models \mathit{Artifact}(v)\}$, *for a designated concept* $\mathit{Artifact} \in M_C$,
2. *for every* $r \in R$ *and* $v, w \in M_I$, *it holds that* $(v, w) \in E$ *and* $l(v, w) = r$ iff $\mathcal{K}_G \models r(v, w)$.

Assuming the names in M_I are interpreted uniquely, it is easy to see that the structure of G is isomorphically encoded in the set of role assertions, over role names in R, entailed by $\mathcal{K}_G$. As the positions of data artifacts remain unaltered, one can immediately rephrase the definition of the satisfaction relation $\Vdash$, to show that for any PSL formula α, node $v \in P \cup D$, and an α-substitution μ it is the case that $G, v \Vdash \mu(\alpha)$ *iff* $\mathfrak{K}_G, v \Vdash \mu(\alpha)$. More interestingly, however, we can instead slightly extend the provenance specification language to make a vital use of the newly included meta-level information.

Definition 6 (PSL^M: syntax). *Let* $G = (P, D, E, l, k)$ *be an* $\mathcal{L}$*-provenance graph and* $\mathcal{L}_G$, $\mathfrak{K}_G$ *the metalanguage and the metaknowledge base over* G, *respectively. Then the* provenance specification language (with metalanguage) *over* $\mathfrak{K}_G$ *is the smallest language induced by the grammar of PSL over* G *(Definition 3), modulo the revision of path expressions:*

***Path expressions**:*

$$\pi \;:=\; r \mid \pi;\pi \mid \pi \cup \pi \mid \pi^- \mid \pi^* \mid v? \mid C? \mid \alpha?$$

where $r \in M_R$, $v \in M_I$ *and* C *is a concept in* $\mathcal{L}_G$,

For a PSL^M provenance formula α, and an α-substitution μ, we say that $\mu(\alpha)$ is *satisfied* a metaknowledge base $\mathfrak{K}_G$ in an instance $v \in M_I$ *iff* $\mathfrak{K}_G, v \Vdash \mu(\alpha)$, where the satisfaction relation $\Vdash$ is defined as follows.

Definition 7 (PSL^M: semantics). *The* satisfaction relation $\Vdash$ *for PSL^M formulas is given by a simultaneous induction over the structure of provenance formulas and path expressions,* w.r.t. *a substitution* μ. *Let* $\mathcal{L}_G$ *be the metalanguage with vocabulary* $\Gamma = (M_C, M_R, M_I)$ *and* $\mathfrak{K}_G = (\mathcal{K}_G, k)$ *the metaknowledge base over an* $\mathcal{L}$*-provenance graph* G. *For all individual names* $v, w \in M_I$*:*

***Provenance formulas**:*

$\mathfrak{K}_G, v \Vdash \{\phi\}$ iff $\mathcal{K}_G \models \mathit{Artifact}(v)$ *and* $k(v) \models \mu(\phi)$,
$\mathfrak{K}_G, v \Vdash \langle\pi\rangle\alpha$ iff *there exists* $w \in M_I$, *s.t.* $\mathfrak{K}_G \Vdash v \xrightarrow{\pi} w$ *and* $\mathfrak{K}_G, w \Vdash \alpha$,

Path expressions:

$\mathfrak{K}_G \Vdash v \xrightarrow{r} w$ iff $\mathcal{K}_G \models r(v, w)$,
$\mathfrak{K}_G \Vdash v \xrightarrow{\pi;\sigma} w$ iff *there is* $u \in M_I$ *s.t.* $\mathfrak{K}_G \Vdash v \xrightarrow{\pi} u$ *and* $\mathfrak{K}_G \Vdash u \xrightarrow{\sigma} w$,
$\mathfrak{K}_G \Vdash v \xrightarrow{C?} v$ iff $\mathcal{K}_G \models C(v)$,

where the remaining conditions are exactly as in Definition 4 (modulo $G/\mathfrak{K}_G$*).*

The model checking problem is rephrased accordingly.

Model Checking 2 (PSL^M formulas) *Given a metaknowledge base* $\mathfrak{K}_G$ *over an* $\mathcal{L}$*-provenance graph* G*, an instance* $v \in M_I$*, a PSL*^{M} *provenance formula* α*, and an* α*-substitution* μ*, decide whether* $\mathfrak{K}_G, v \Vdash \mu(\alpha)$*.*

The usefulness of the presented extension, in particular of the test operator $C?$, which allows for referring to graph nodes generically by their types, inferred from the metaknowledge base, can be observed in the following example.

Example 3. See Q6 in Section 1 and Example 2. We restate the formula α as:

$$\alpha := \{\text{Image}(x)\} \wedge \langle \mathit{wasGeneretedBy}; \mathit{Softmean}?; \mathit{used}\rangle$$
$$(\{\exists y.\text{Image}(y)\} \wedge \langle(\mathit{wasGeneratedBy}; \mathit{used})^*; \mathit{wasGeneratedBy};$$
$$(\mathit{Align\text{-}warp} \sqcap \exists \mathit{hasArgValue}.\{\text{“-m 12”}\})?\rangle \top)$$

where $\mathcal{K}_G$, in the metaknowledge base $\mathfrak{K}_G = (\mathcal{K}_G, k)$, contains (among others) the axioms from Table 1.

Solution: For every $v \in M_I$ and α-substitution μ, if $\mathfrak{K}_G, v \Vdash \mu(\alpha)$, then μ is a requested resource.

Compared to its PSL variant from Example 2, the PSL^M formula used in Example 3 is much more succinct and explicitly represents all requested information. More importantly, thanks to the use of a generic vocabulary for classifying nodes (here: concepts *Softmean*, $\mathit{Align\text{-}warp} \sqcap \exists \mathit{hasArgValue}.\{\text{“-m 12”}\}$), instead of their enumerations, the formula is also more input-independent, in the sense that it can be directly reused to verify/query alternative provenance records obtained from running the same workflows.

6 Evaluation

In order to validate our approach in practical scenarios, we have analyzed the complete list of test queries from The First Provenance Challenge, which to our knowledge constitutes a so far unique ‘golden standard’ for the provenance community. Below we model possible solutions using the logic PSL^M and elaborate on our findings.

Q1. *Find the process that led to Atlas X Graphic / everything that caused Atlas X Graphic to be as it is. This should tell us the new brain images from which the averaged atlas was generated, the warping performed etc.*

$$\alpha_1 := (\top \vee \{\top(x)\}) \wedge \langle (used^- \cup wasGeneratedBy^-)^*; Atlas\text{-}X\text{-}Graphic?\rangle \top$$

Solution: Every $v \in M_I$ and μ such that $\mathfrak{K}, v \Vdash \mu(\alpha_1)$ are requested resources.

Q2. *Find the process that led to Atlas X Graphic, excluding everything prior to the averaging of images with softmean.*

$$\alpha_2 := \alpha_1 \wedge [(used^- \cup wasGeneratedBy^-)^*] \neg \langle wasGeneratedBy^-; Softmean?\rangle \top$$

Solution: Every $v \in M_I$ and μ such that $\mathfrak{K}, v \Vdash \alpha_2$ are requested resources.

Q3. *Find the Stage 3, 4 and 5 details of the process that led to Atlas X Graphic.*

Comment: This is a complex search/verification task, whose reasoning parts can be accomplished by a mix of formulas used in Q1, Q2. Essentially, one must decide what the relevant details are and retrieve them by applying appropriate provenance formulas over the provenance graphs.

Q4. *Find all invocations of procedure align warp using a twelfth order nonlinear 1365 parameter model, i.e. align warp procedure with argument -m 12, that ran on a Monday.*

$$\alpha_4 := \langle (Align\text{-}warp \sqcap \exists hasArgValue.\{\text{“-m 12”}\} \sqcap \exists executedOn.Monday)?\rangle \top$$

Solution: Every $v \in M_I$ such that $\mathfrak{K}, v \Vdash \alpha_4$ is a requested resource.

Q5. *Find all Atlas Graphic images outputted from workflows where at least one of the input Anatomy Headers had an entry global maximum=4095. The contents of a header file can be extracted as text using the scanheader AIR utility.*

$$\alpha_5 := \langle AtlasGraphic?\rangle(\{\text{Image}(x)\} \wedge \langle (wasGeneratedBy; used)^*\rangle \\ \langle AnatomyHeader?\rangle\{\text{hasValue(global-maximum, “4095”)}\})$$

Solution: For every $v \in M_I$ and μ such that $\mathfrak{K}, v \Vdash \mu(\alpha_5)$, μ is a requested resource.

Q6. Example 3, discussed in the previous section.

Q7. *A user has run the workflow twice, in the second instance replacing each procedures (convert) in the final stage with two procedures: pgmtoppm, then pnmtojpeg. Find the differences between the two workflow runs. The exact level of detail in the difference that is detected by a system is up to each participant.*

Comment: This is a complex search/verification task, whose reasoning parts can be accomplished by posing a number of model checking problems. Essentially, for each relevant provenance formula one must verify it over both graphs and compare the obtained answers.

Q8. *A user has annotated some anatomy images with a key-value pair center=UChicago. Find the outputs of align warp where the inputs are annotated with center=UChicago.*

$$\alpha_8 := \{\top(x)\} \wedge \langle \mathit{wasGeneratedBy}; \mathit{Align\text{-}warp}?; \mathit{used}; \mathit{AnatomyImage}?\rangle \{\exists y.(\mathrm{Image}(y) \wedge \mathrm{center}(y, \mathrm{UChicago}))\}$$

Solution: For every $v \in M_I$ and μ such that $\mathfrak{K}, v \Vdash \mu(\alpha_8)$, μ is a requested resource.

Q9. *A user has annotated some atlas graphics with key-value pair where the key is studyModality. Find all the graphical atlas sets that have metadata annotation studyModality with values speech, visual or audio, and return all other annotations to these files.*

$$\alpha_9 := \langle \mathit{AtlasGraphic}?; \exists \mathit{studyModality}.\{\mathit{speech}\}? \cup \exists \mathit{studyModality}.\{\mathit{visual}\}? \cup \exists \mathit{studyModality}.\{\mathit{radio}\}?\rangle \top$$

Solution: Every $v \in M_I$ such that $\mathfrak{K}, v \Vdash \alpha_9$ is a requested resource. Finding other annotations can be accomplished by posing a number of model checking problems w.r.t. the identified resources.

The above analysis shows that typical reasoning tasks over data provenance records consist of two components: search and logical verification. As far as verification is concerned, the logic PSL^M proves well suited for modeling requested properties and queries. In particular, out of the 9 considered problems, at least 5 — Q1, Q2, Q5, Q6, Q8 — can be solved directly, using a combination of all distinctive features of PSL^M, namely: PDL-like path expressions, embedded CQs and the metalanguage. Queries Q4, Q9 can be answered without the use of embedded CQs. Problems Q3, Q7 and partially Q9 are in fact descriptions of complex search/verification tasks, which can be decomposed into a number of individual verification problems. Those, in turn, can be addressed using PSL^M in the same fashion as in the remaining cases.

7 Reasoning and Complexity

The close relationship of PSL^M to PDL can be conveniently exploited on the computational level. Crucially, PSL^M model checking can be decoupled into two separate problems:

1. construction of a finite-state transition system and a PDL formula (involving polynomially many CQ answering / DL entailment problems),
2. PDL model checking.

This technically unsurprising, but fully intended result has some significant theoretical and practical implications. From the theoretical perspective, it allows for identifying a complexity bound, invariant to the cost of reasoning with the particular DL languages used in the representation. From the practical viewpoint, it opens up a possibility of building simple, yet well-grounded and efficient reasoning architectures based on existing, highly optimized DL reasoners, query engines (e.g. Mastro), and PDL model checkers (e.g. MCPDL[5]).

[5] See `http://www2.tcs.ifi.lmu.de/~axelsson/veri_non_reg/pdlig_mc.html`.

In the following, we sketch the reduction procedure. Its full description and the missing proofs are presented in the online technical report [24]. First, recall the notion of a finite-state transition system.

Definition 8 (Transition system). *Let $\mathcal{P} = \{p, q, \ldots\}$ be a set of propositional letters and $\mathcal{A} = \{r, s, \ldots\}$ a set of atomic program names. Then a* finite-state transition system *is a tuple $\mathcal{S} = (W, \{\xrightarrow{r} | \ r \in \mathcal{A}\}, \mathcal{I})$, where:*

- *W is a finite, non-empty set of elements called* states*,*
- *$\xrightarrow{r} \subseteq W \times W$ is a* transition relation *corresponding to program r,*
- *$\mathcal{I} : W \mapsto 2^{\mathcal{P}}$ is a propositional* valuation *function.*

Let α be a PSL^M provenance formula, μ an α-substitution, $\mathcal{L}_G$ a metalanguage with vocabulary $\Gamma = (M_C, M_R, M_I)$, and $\mathfrak{K}_G = (\mathcal{K}_G, k)$ a metaknowledge base over an $\mathcal{L}$-provenance graph G. Given this input, we define a finite-state transition system $\mathcal{S}(\mathfrak{K}_G, \mu(\alpha)) = (W, \{\xrightarrow{r} | \ r \in \mathcal{A}\}, \mathcal{I})$, essentially, by turning individuals of $\mathcal{K}_G$ into states of the system, structuring the transition relations in the system isomorphically to the corresponding role relationships in $\mathcal{K}_G$, and by encoding all relevant information about the individuals in $\mathcal{K}_G$ in the valuation $\mathcal{I}$ over a designated set of propositional letters corresponding to concepts and CQs. This reduction step involves a polynomial number of decision problems of the form $\mathcal{K}_G \models C(v)$ and $k(v) \models \mu(\phi)$, where $v \in M_I$, C is a concept in $\mathcal{L}_G$ and ϕ is a CQ. Further, we transform the formula $\mu(\alpha)$ by consistently applying substitutions of designated propositions for the corresponding PSL^M subformulas in $\mu(\alpha)$. The resulting expression $\mu(\alpha)^{\text{PDL}}$ is a well-formed PDL formula. Thanks to such "propositionalization" of the input we obtain the following reduction result, where $\mathcal{S}, v \models \varphi$ denotes the model checking problem in PDL, i.e. the problem of deciding whether a PDL formula φ is satisfied in state v of the transition system $\mathcal{S}$.

Theorem 1 (PSL^M vs. PDL) $\mathfrak{K}_G, v \Vdash \mu(\alpha)$ iff $\mathcal{S}(\mathfrak{K}_G, \mu(\alpha)), v \models \mu(\alpha)^{\text{PDL}}$.

It is known that the complexity of model checking in PDL is PTIME-complete [12]. Moreover, the size of the transition system $\mathcal{S}(\mathfrak{K}_G, \mu(\alpha))$ and of the formula $\mu(\alpha)^{\text{PDL}}$ is polynomial in $\ell(\mathfrak{K}_G, \alpha, \mu)$, where $\ell(\mathfrak{K}_G, \alpha, \mu)$ is the total size of $\mathfrak{K}_G$, α and μ measured in the number of symbols used. This means, that by disregarding the cost of DL reasoning involved in the construction of $\mathcal{S}(\mathfrak{K}_G, \mu(\alpha))$, we obtain the following time bound.

Theorem 2 (PSL^M model checking: complexity) *Let $\mathfrak{K}_G$ be a metaknowledge base, expressed in $\mathcal{L}_G$, over an $\mathcal{L}$-provenance graph G. Model checking PSL^M formulas over $\mathfrak{K}_G$ is* PTIME^{DL}*-complete, where DL is an oracle answering CQs in $\mathcal{L}$ and deciding DL entailment in $\mathcal{L}_G$.*

Finally, we observe that for a given problem $\mathfrak{K}_G, v \Vdash \alpha$ there are at most $2^{\ell(\mathfrak{K}_G, \alpha)}$ different α-substitutions μ, and thus, maximum $2^{\ell(\mathfrak{K}_G, \alpha)}$ different possible pairs $\mathcal{S}(\mathfrak{K}_G, \alpha), \mu(\alpha)^{\text{PDL}}$ to be considered. In practice this number can be dramatically reduced by using smart heuristics to guess only potentially "promising" substitutions. Analogically, the described procedure of constructing the transition systems leaves a considerable space for practical optimizations.

8 Conclusion

In this paper we have introduced the provenance specification logic PSL^M— a dynamic logic-based formalism for verification of data provenance records. The validation, which we have conducted using the test queries of The First Provenance Challenge, shows that a typically requested reasoning task over a data provenance record consist of two components: search and logical verification. As far as the search aspect goes beyond the scope of this work and remains an interesting problem in its own right, requiring smart retrieval and heuristic techniques, we have demonstrated that the logical reasoning part can be successfully captured using the logic and the framework developed here. Moreover, we have shown that the computational cost of performing such tasks is very moderate, and depends mostly on the expressiveness of the languages used for representing the data and the provenance record.

With this contribution, we hope to pave the way towards more systematic and formal studies of the logical problems emerging on the intersection of data-level representations with their meta-level provenance/contextual descriptions, which, in our belief, are of rapidly growing importance in the large-scale, distributed, data- and semantics-rich environments of the Semantic Web and eScience.

Acknowledgments The authors thank Paul Groth for inspiring discussions on the ideas presented in this paper, and Davide Ceolin for helpful comments.

References

1. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. SIGMOD Rec. **34** (2005)
2. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., , den Bussche, J.V.: The open provenance model — core specification (v1.1). Future Generation Computer Systems **27** (2010)
3. Sahoo, S.S., Sheth, A., Henson, C.: Semantic Provenance for eScience: Managing the Deluge of Scientific Data. IEEE Internet Computing **12**(4) (2008)
4. Miles, S., Wong, S.C., Fang, W., Groth, P., Zauner, K.P., Moreau, L.: Provenance-based validation of e-science experiments. Web Semantics **5** (2007)
5. Moreau, L., et al.: Special issue: The first provenance challenge. Concurrency and Computation : Practice and Experience **20** (2008)
6. Baader, F., Calvanese, D., Mcguinness, D.L., Nardi, D., Patel-Schneider, P.F.: The description logic handbook: theory, implementation, and applications. Cambridge University Press (2003)
7. Groth, P., Gil, Y.: Editorial - using provenance in the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web **9**(2) (2011)
8. Golbeck, J., Hendler, J.: A semantic web approach to the provenance challenge. Concurrency and Computation: Practice and Experience **20**(5) (2008)
9. Moreau, L.: Provenance-based reproducibility in the semantic web. Web Semantics: Science, Services and Agents on the World Wide Web **9**(2) (2011)

10. Bonatti, P.A., Hogan, A., Polleres, A., Sauro, L.: Robust and scalable linked data reasoning incorporating provenance and trust annotations. Web Semantics: Science, Services and Agents on the World Wide Web **9**(2) (2011)
11. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. The MIT Press (2000)
12. Lange, M.: Model checking propositional dynamic logic with all extras. Journal of Applied Logic **4**(1) (2006)
13. Wolter, F., Zakharyaschev, M.: Dynamic description logics. In: Proceedings of AiML-98. (2000)
14. Vianu, V.: Automatic verification of database-driven systems: a new frontier. In: Proceedings of ICDT-09. (2009)
15. Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R.: Actions and programs over description logic knowledge bases: A functional approach. In Lakemeyer, G., McIlraith, S.A., eds.: Knowing, Reasoning, and Acting: Essays in Honour of Hector Levesque. College Publications (2011)
16. Karamanolis, C.T., Giannakopoulou, D., Magee, J., Wheater, S.M.: Model checking of workflow schemas. In: Proceedings of EDOC-00. (2000)
17. Motik, B.: OWL 2 web ontology language profiles. Technical report, W3C Recommendation: `http://www.w3.org/TR/owl2-profiles/` (2009)
18. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. In: Proceedings of KR-06. (2006)
19. Glimm, B., Horrocks, I., Lutz, C., Sattler, U.: Conjunctive query answering for the description logic SHIQ. In: Journal of Artificial Intelligence Research. (2007)
20. da Silva, P.P., McGuinness, D.L., Fikes, R.: A proof markup language for semantic web services. Journal of Information Systems - Special issue: The semantic web and web services **31**(4) (2006)
21. Mcguinness, D.L., Ding, L., Silva, P.P.D., Chang, C.: Pml2: A modular explanation interlingua. In: Proceedings of ExaCt-07. (2007)
22. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF. Technical report, W3C Recom.: `http://www.w3.org/TR/rdf-sparql-query/` (2008)
23. Belhajjame, K., et al.: The PROV Ontology: Model and formal semantics. Technical report, W3C Draft: `http://dvcs.w3.org/hg/prov/raw-file/default/ontology/ProvenanceFormalModel.html` (2011)
24. Klarman, S., Schlobach, S., Serafini, L.: Formal verification of data provenance records. Technical report, `http://klarman.synthasite.com/resources/KlaEtAlISWC12.pdf` (2012)