Why-provenance Information for RDF, Rules, and Negation

Anastasia Analyti[1], Carlos V. Damásio[2], Grigoris Antoniou[1,3], and Ioannis Pachoulakis[4]

[1] Institute of Computer Science, FORTH-ICS, Greece
[2] CENTRIA, Departamento de Informatica, Faculdade de Ciencias e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[3] Department of Informatics, University of Huddersfield, Huddersfield, UK
[4] Dept. of Informatics Engineering, TEI of Crete, Greece
analyti@ics.forth.gr, cd@fct.unl.pt, G.Antoniou@hud.ac.uk, ip@ie.teicrete.gr

Abstract. The provenance (i.e., origins) of derived information on the Web is crucial in many applications to allow information quality assessment, trust judgments, accountability, as well as understanding the temporal and spatial status of the information. On the other hand, the inclusion of negative information in knowledge representation both in the form of negation-as-failure and explicit negation is also important to allow various forms of reasoning, provided that weakly negated information is associated with the sources (contexts) in which it holds. In this work, we consider collections of g-RDF ontologies, distributed over the web, along with a set of conflict statements expressing that information within a pair of g-RDF ontologies cannot be combined together for deriving new information. A g-RDF ontology is the combination of (i) a g-RDF graph G (i.e., a set of positive and strongly negated RDF triples, called g-RDF triples) and (ii) a g-RDF program P containing derivation rules with possibly both explicit and scoped weak negation. Information can be inferred through the g-RDF graphs or the derivation rules of the g-RDF ontologies, or through the RDFS derivation rules. We associate each derived grounded g-RDF triple $[\neg]p(s, o)$ with the set of names S of the g-RDF ontologies that contributed to its derivation. To achieve this, we define the provenance stable models of a g-RDF ontology collection. We show that our provenance g-RDF semantics faithfully extends RDFS semantics. Finally, we provide an algorithm based on Answer Set Programming that computes all provenance stable models of a g-RDF ontology collection and provides the answer to various kinds of queries. Various complexity results are provided.
Keywords: why-provenance, RDF graphs, program rules, negation, query answering, complexity.

1 Introduction

Ontologies and automated reasoning are the building blocks of the Semantic Web initiative. During the recent years an increasing number of data providers have adopted a set of best practices for publishing and connecting RDF data on the Web, leading to the creation of distributed dataspaces. These dataspaces hold an enormous potential in terms of combining RDF data about the same resource and deriving new information based on the RDFS inference rules that cannot be inferred from single sources alone. However, such combinations and derivations pose questions in terms of information quality, trustworthiness, temporal and spatial status of information, etc. On the other hand, additional derivation rules can be included in an ontology to define derived concepts based on base concepts. For example, rules allow defining the extension of a property based on a complex relation between the extensions of the same or other properties. Additionally, the inclusion of negative information both in the form of negation-as-failure (weak negation $\sim$) and explicit negative information (strong negation $\neg$) is also needed to enable various forms of reasoning, provided that weakly negated information is associated with the sources (contexts) in which it holds [4]. For example, asking for the jobs that are not heavy and unhealthy in the World Wide Web is meaningless unless the answers are accompanied by the sets of sources that this information holds. Scoped negation as failure means that something cannot be derived from a set of sources, while explicit negative information is information that either is explicitly declared to be false or derived to be false. Thus, scoped weak negation is non-monotonic and is essential to reason with absent (incomplete) information.

In this paper, we consider collections of g-RDF ontologies, distributed over the web. A g-RDF[5] ontology is the combination of (i) a g-RDF graph G (i.e., a set of positive and strongly negated RDF triples, whose subject can be a literal and its property can be a variable, called *g-RDF triples*) and (ii) a g-RDF program P containing derivation rules whose body is a conjunction of g-RDF triples and scoped weakly negated g-RDF triples and their head is a g-RDF triple. We would like to note that any form of recursion is allowed in the g-RDF programs. For provenance control, each g-RDF ontology is associated with a name [12, 11]. It is useful that derived information based on the distributed g-RDF ontologies is associated with the set of the sets of names of the g-RDF ontologies that contributed to its derivation[6]. This is the kind of why-provenance information supported in this work, called *RDF why-provenance*. In other words, RDF why-provenance keeps track of the origins of the derived RDF triples. Each RDF ontology is possibly associated with metadata information (like using the PROV-O ontology under development by the W3C [6]), expressing, for example, its quality, certainty, authority, temporal and spatial status. Thus, RDF why-provenance information derived together with a g-RDF triple can help clarify the conditions under which the g-RDF triple is derived. In particular, any linked open dataset can be seen as a g-RDF ontology, and thus the present work is immediately applicable to Linked Open Data.

The model of why-provenance for relational data first appeared in [9] (for a survey, see [14]), where why-provenance distinguishes the set of source tuples that are involved in the derivation of a tuple (and, thus, can be seen as a set of sets of contributing source tuples). In our approach, we consider instead of contributed source tuples, contributed g-RDF ontologies whose metadata information can be useful in validating derived g-RDF triples. This granularity level of provenance for RDF data has been proposed before for instance in [26]. However, our model can be easily adapted to obtain tuple-level provenance information at the expense of extra complexity. In fact, as it will be demonstrated by the examples provided in this paper, RDF why-provenance can have several applications in the government, biomedical, medical, and cultural domain.

The Rule Interchange Format (RIF) [43] is an activity within the World Wide Web Consortium aimed at the introduction of rule based technologies into the mainstream of knowledge representation and information processing on the Web. One of the use cases of RIF [53] is rule set integration for medical decision support. Medical decision support systems might use rules from pharmaceutical knowledge bases, laboratory knowledge bases, patient databases, and medical ontologies, such as SNOMED Clinical Terminology[7], Gene Ontology (GO)[8], Foundational Model of Anatomy (FMA)[9], and DailyMed[10]. Such systems can be used to provide therapeutic recommendations, personalized drug

[5] g-RDF stands for *general* RDF.

[6] The outer set captures all alternative derivations and each inner set corresponds to one of those alternative derivations.

[7] http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html

[8] http://www.geneontology.org/

[9] http://sig.biostr.washington.edu/projects/fm/

[10] http://dailymed.nlm.nih.gov/dailymed/

descriptions, and assessment of the effectiveness of a treatment. Of course, such conclusions can only be assessed if they are accompanied by the set of sources that contributed to their derivation. Note that drug prescription is a complex task due to possible interactions and contraindications based on a large number of parameters, including patient's overall condition.

Our theory can also be applied to the cultural domain. For example, information about an ancient artifact may be derived from the place where it was found, its artistic features, other artifacts found at the same place, famous persons that used the artifact, etc. Since archaeological information is many times incomplete and ambiguous, knowing the sources from which information about the artifact is derived is very important to archaeologists.

As another example, consider different labs working on the Alzheimer Disease using terms from the SWAN Ontology [15] for interoperability. Collaborations between the different labs allows the derivation of new information. Certainly knowing the labs that contributed to the derivation of this information if very important.

Note that in all of the three previous examples, why-provenance information becomes even more important if the collaboration of a set of sources derives information that is in conflict with the information derived from the collaboration of a different set of sources.

A *g-RDF information base* $\mathcal{C}$ is a set of g-RDF ontologies associated with their names and a set of conflict statements *conflict*(c, c'), where c, c' are the names of the g-RDF ontologies in $\mathcal{C}$, expressing that g-RDF ontology c is in conflict with g-RDF ontology c' and that their information cannot be combined together for deriving new information. Thus, we extend the RDF theory by (i) considering g-RDF triples, instead of RDF triples, (ii) considering g-RDF programs within g-RDF ontologies with possibly explicitly negated and scoped weakly negated g-RDF triples, (ii) considering a collection of g-RDF ontologies within which inferences are made, (iii) considering conflict statements between the g-RDF ontologies of a collection, and (iv) considering provenance quads $[\neg]H(s, p, o, S)$,[11] where S is the set of names of the g-RDF ontologies that contributed to the derivation of the ground g-RDF triple $[\neg]p(s, o)$.

Conflicts can naturally occur between rule bases on the web because of different views, different opinions, incomplete data, erroneous data, different temporal and spatial status. The problem of conflicting sources has been identified in [24, 62]. For example, there may exist :

- conflicts between sources about the way dinosaurs disappeared,
- conflicts between sources about the real-world setting for the legendary island Atlantis,
- conflicts between sources about the time and reason gypsies left India,
- conflicts between outdated and more recent information about a gene,
- conflicts on the regional division of Middle East among different scholars and between different periods.

In particular, in this paper:

1. We extend the simple interpretations of the RDF theory [39] to the provenance simple interpretations, where each property p is associated not only with a property-truth but also with a property-falsity extension, each being not a set of pair of resources $\langle s, o\rangle$ but a set of triples $\langle s, o, S\rangle$, where S is the set of names of the g-RDF ontologies that contributed to derive $p(s, o)$ or $\neg p(s, o)$.

[11] Note that the symbol $[\neg]$ denotes that strong negation is optional and the symbol $[\sim]$ denotes that weak negation is optional.

2. We extend RDFS interpretations to provenance RDFS interpretations of a g-RDF information base $\mathcal{C}$ by extending the RDFS interpretation conditions with property-falsity information, why-provenance information, and conflict information among the g-RDF ontologies of $\mathcal{C}$.
3. We define the provenance Herbrand interpretations and provenance stable models of a g-RDF information base $\mathcal{C}$. We show that our provenance stable model semantics faithfully extends RDFS semantics [39], yet due to the infinite in number rdf:_i terms of RDF theory, it is undecidable. Therefore, we propose the $\#n$-provenance stable model semantics which also extends RDFS semantics but eliminates all rdf:_i terms with $i > n$.
4. We define various kinds of queries, such as (i) which are the g-RDF triples or weakly negated g-RDF triples that are derived from a set of g-RDF ontologies that is subset, superset, or equal to a provided set of g-RDF ontologies, (ii) which are the sets of g-RDF ontologies that contribute to the derivation of a g-RDF triple or a weakly negated g-RDF triple, and (iii) which are the sets of g-RDF ontologies that contribute to the derivation of a g-RDF triple t and which are the sets of g-RDF ontologies that contribute to the derivation of $\neg t$. In general, all kinds of queries of an extended logic programming language are allowed together with provenance information and associated with subset, superset, or equal conditions on RDF-why provenance. We show that the answer to these queries can be provided through Answer Set Programming on a particular extended logic program [32]. Additionally, we show that the complexity of query answering is co-NP$^{\text{NP}}$-complete w.r.t. the size of $\mathcal{C}$, and co-NP-complete w.r.t. the sizes of the g-RDF graphs appearing in $\mathcal{C}$.
5. Finally, we consider the case of g-RDF information bases, without negation, called *positive g-RDF information bases*. We show that the complexity of query answering for positive g-RDF information bases is NP-complete w.r.t. the size of $\mathcal{C}$ and P-complete w.r.t. the sizes of the g-RDF graphs appearing in $\mathcal{C}$. We provide a translation into XSB-PROLOG resulting in a sound and complete implementation that can be used to provide the answers of all kinds of defined queries over a positive g-RDF information base, using the XSB logic programming system [58].

The idea of adding a fourth element to an RDF triple is not new and appears in [50] as the context in which the RDF triple is true or, similarly to our work, in [26] as a set of source names. However, all the works in the literature we are aware of [26, 63, 10] do not consider any general form of rules and most of them restrict themselves to the case of RDFS semantics for the ρ-df fragment [51]. In particular, in [26], this why-provenance information is derived through a subset of the RDFS inference rules, without considering a model theory and conflicts between sources. Additionally, in [26], only RDF graphs (and not program rules, weak and strong negation) are considered.

In summary, and to the best of our knowledge this is the first work that considers RDF why-provenance for g-RDF ontologies where derivation rules, explicit and scoped weak negation, and conflicts among sources are permitted. Additionally, no safety condition is imposed, while any kind of recursion is allowed.

The rest of the paper is organized as follows: In Section 2, we define g-RDF information bases and provide an example. In Section 3, we define the provenance RDFS interpretations. In Section 4, we define the provenance stable model semantics of a g-RDF information base. In Section 5, we define the $\#n$-provenance stable model semantics of a g-RDF information base, eliminating the infinite rdf:_i terms with $i > n$. In Section 6, we provide an algorithm based on Answer Set Programming that computes all the entailed provenance quads $[\sim][\neg]H(s,p,o,S)$ of a g-RDF information base and provides the answer of various kinds of queries. In Section 7, we consider the case of positive g-RDF ontologies. Section 8 provides a discussion and further applications. Section 9 reviews related work, in particular the relationship to other provenance models for RDF. Finally, Section 10 provides concluding remarks and directions for further research. Appendix A provides the proof of all propositions and Appendix B provides a list of symbols.

2 General RDF Information Bases

The Resource Description Framework is the standard language for specifying information in the Semantic Web. In this section, we define *general* RDF (*g-RDF*) *ontologies* and *general RDF* (*g-RDF*) *information bases* which extend RDF with mechanisms to infer and obtain new information, as well as to organize it in a modular way. Additionally, we provide an example of a *g*-RDF information base. In the subsequent sections, the provenance stable models of a *g*-RDF information base will be defined.

A (Web) *vocabulary* V is a set of URI references and/or literals (plain or typed) [39]. We denote the set of all URI references by URI, the set of all plain literals by $\mathcal{PL}$, the set of all typed literals by $\mathcal{TL}$, and the set of all literals by $\mathcal{LIT}$. We consider a set Var of variable symbols, such that the sets Var, URI, $\mathcal{LIT}$ are pairwise disjoint. In our examples, variable symbols are prefixed by the question mark symbol “?”.

Below, we define a *g*-RDF graph, as an extension of an RDF graph.

Definition 1 **(*g*-RDF triple, *g*-RDF graph).** Let V be a vocabulary. A *g-RDF triple* over V is an expression of the form $[\neg]p(s, o)$, where $s, o \in V \cup \mathit{Var}$ are called *subject* and *object*, respectively, and $p \in (V \cap \mathit{URI}) \cup \mathit{Var}$ is called *property*. A *g-RDF graph* G is a set of *g*-RDF triples. The set of URI references and literals appearing in G is denoted by V_G. Additionally, the set of variables appearing in G is denoted by $\mathit{Var}(G)$. $\square$

Obviously, the notion of a *g*-RDF graph extends the notion of an RDF graph. Note that an RDF triple is a positive *g*-RDF triple with the constraint that the subject of the triple is not a literal and the property is not a variable. For example, `ex:URI_of`(“`Ioannis`”, `ex:Ioannis`) is a valid *g*-RDF triple but not a valid RDF triple. Our choice of allowing literals appearing in the subject position is based on our intuition that this case can naturally appear in knowledge representation (as in the previous example). SPARQL [55] and de Bruijn et al. [20] also consider literals in the subject position of RDF triples. Our choice of allowing variables in the property position of a *g*-RDF triple is justified by the fact that, in RDF, properties are resources on which various statements can be made. For example, we may want to identify the properties that relate two resources and have specific characteristics. Ter Horst [61] also allows variables in the property position of RDF triples.

Another extension provided by *g*-RDF graphs, is the possibility to declare negative information. A negative triple can be used to declare that Peter does not know Mary, i.e., $\neg$`foaf:knows`(`ex:Peter`, `ex:Mary`). Note that a *g*-RDF graph does not contain weakly negated *g*-RDF triples, as it contains only explicitly declared information.

Besides the raw data, mechanisms are required to extract information from RDF graphs that can be obtained from different places in the Semantic Web, for instance via SPARQL endpoints. Based now on the notion of a *g*-RDF triple, we define a *g*-RDF program and a *g*-RDF ontology. A general RDF ontology is used to extract more information from a provided RDF graph using a program formed from declarative logic programming rules, which can query the given RDF graph as well as any available external general RDF ontology. Weakly negated queries are restricted to a scope (context) identified by a set of Uniform Resource Identifiers, which identify other general RDF ontologies. The limitation on negative queries avoids indiscriminate non-monotonic queries to ontologies, by forcing the knowledge engineer to explicitly specify the intended ontologies.

Definition 2 (g-RDF program, g-RDF ontology). A *g-RDF rule* is a derivation rule of the form: $t_0 \leftarrow t_1, ..., t_l, {\sim}t_{l+1}@N_{l+1}, ..., {\sim}t_k@N_k$, where each t_i, for $i = 0, ..., k$, is a g-RDF triple and $N_i \subseteq \mathit{URI}$, for $i = l + 1, ..., k$.[12] A *g-RDF program* is a finite set of *g-RDF rules.* A *g-RDF ontology* O is a pair $\langle G_O, P_O \rangle$, where G_O is a g-RDF graph and P_O is a g-RDF program. The set of URI references and literals appearing in a g-RDF program P is denoted by V_P. $\square$

The g-RDF graph G_O of a general RDF ontology O provides the local data of the ontology. The local data can be used to infer new information via the g-RDF program P_O. A rule of the form $t_0 \leftarrow t_1, ..., t_l, {\sim}t_{l+1}@N_{l+1}, ..., {\sim}t_k@N_k$ in P_O of ontology O states that t_0 holds in O whenever each t_i $(1 \leq i \leq l)$ can be derived by the available ontologies (including itself), and each t_j $(l + 1 \leq j \leq k)$ cannot be derived by the ontologies identified by a URI in N_j. The set of available ontologies is formally captured by a g-RDF information base $\mathcal{C}$ consisting of a collection of g-RDF ontologies, associated with their names, and a set of *conflict* statements between pairs of g-RDF ontologies. A *conflict* statement between two g-RDF ontologies O, O' of $\mathcal{C}$ indicates that information derived through O, O' cannot be combined together.

Note that scoped negation as failure has also been considered in [59, 54, 8][13]. But there, weakly negated literals ${\sim}L$ are qualified by a single source, i.e., ${\sim}L@O$, indicating that L cannot be derived by source O. In our case, weakly negated g-RDF triples ${\sim}t$ are qualified by a set of sources N, i.e., ${\sim}t@N$, indicating that t cannot be derived by the *combination* of the knowledge of the sources in N.

Definition 3 (g-RDF information base). A *g-RDF information base* $\mathcal{C}$ is (i) a set of pairs $\langle Nam_{O_i}, O_i \rangle$, where $i = 1, ..., m$, O_i is a g-RDF ontology, and $Nam_{O_i} \in \mathit{URI}$ (called the *name* of O_i) and (ii) a set of statements $\mathit{conflict}(c, c')$, where $c, c' \in \mathit{names}_{\mathcal{C}}$ and $\mathit{names}_{\mathcal{C}} = \{Nam_{O_i} \mid i = 1, ..., m\}$. In compact form, $\mathcal{C} = \langle \{\langle Nam_{O_i}, O_i \rangle \mid i = 1, ..., m\}, \mathit{conflict} \rangle$. For each scoped weakly negated g-RDF triple ${\sim}t@N$ appearing in $\mathcal{C}$, it should hold $N \subseteq \mathit{names}_{\mathcal{C}}$. $\square$

Let $\mathcal{C}$ be a g-RDF information base and let $c \in \mathit{names}_{\mathcal{C}}$. We denote the set of URI references and literals appearing in c by V_c.

Example 1. In Figure 1, we present a g-RDF information base $\mathcal{C}$. The g-RDF ontology with name $Nam_{O_1} = \langle \mathit{http://general\_info.int} \rangle$ contains general information about persons and jobs. The g-RDF ontology with name $Nam_{O_2} = \langle \mathit{http://persons\_info.gr} \rangle$ contains information about the jobs of greek persons. The g-RDF ontology with name $Nam_{O_3} = \langle \mathit{http://jobs\_info\_2000\_to\_2009.gr} \rangle$ contains information regarding the heavy and unhealthy jobs in Greece for the period [2000,2009]. The g-RDF ontology with name $Nam_{O_4} = \langle \mathit{http://jobs\_info\_2010\_to\_2013.gr} \rangle$ contains information regarding the heavy and unhealthy jobs in Greece for the period [2010,2013]. The g-RDF ontology with name $Nam_{O_5} = \langle \mathit{http://pensions\_info\_2000\_to\_2009.gr} \rangle$ contains rules determining, for the period [2000,2009], the age that a person gets pension depending on the kind of his/her job. The g-RDF ontology with name $Nam_{O_6} = \langle \mathit{http://pensions\_info\_2010\_to\_2013.gr} \rangle$ contains rules determining, for the period [2010,2013], the age that a person gets pension depending on the kind of his/her job. The g-RDF ontology with name $Nam_{O_7} = \langle \mathit{http://sameAs\_info.com} \rangle$ contains rules determining the values of the *owl*:*sameAs* property. Note that in the head of the last rule of ontology O_7, there is a variable in the property position of the corresponding g-RDF triple.

[12] Note that N_i can be a set that is not necessarily a singleton, as in our examples.
[13] These works are reviewed in the Related Work section.

Note that the g-RDF ontologies $O_1, ..., O_4$ contain just g-RDF triples (in their corresponding g-RDF graphs), whereas the rest of the g-RDF ontologies in $\mathcal{C}$ contain just g-RDF program rules.

The g-RDF information base $\mathcal{C}$ also contains a set of *conflict* statements, indicating that O_3 is in conflict with O_4, O_3 is in conflict with O_6, O_4 is in conflict with O_5, and O_5 is in conflict with O_6. In this example, conflicts are due to the different temporal status of the ontologies. Thus, we cannot combine information from O_2, O_3, and O_6 and derive that $\texttt{ex:gets\_pension}(\texttt{ex:Mary}, \text{"62"}\hat{}\hat{}\,xsd{:}integer)$, since ontologies O_3 and O_6 are in conflict.

Note that the g-RDF ontologies O_1 to O_6 may be associated with meta-information regarding their temporal and spatial status, topic, authorship, certainty, etc. not considered in this work. This meta-information helps understanding the status of the derived g-RDF triples. □

Possible queries that can be imposed to $\mathcal{C}$ is (i) what is the age that Mary gets pension, as derived from a set of g-RDF ontologies that is superset of $\{Nam_{O_2}\}$ and subset of $\{Nam_{O_1}, Nam_{O_2}, Nam_{O_3}, Nam_{O_5}\}$ and (ii) what is age that Mary gets pension and the associated RDF why-provenance information.

3 Provenance RDFS Interpretations

Since the data is open and used freely, some form of annotating the provenance of the data, but at the same time respecting the RDF and RDF schema semantics, is required. The details are complex but the intuition can be summarized as follows. First, RDF and RDF schema provide a basic vocabulary and semantics that is capable to express the existing classes and properties, membership of entities to classes, domains and range of properties, and inheritance. We complement the semantic with annotations taken from the why-provenance semiring [9] to track the (alternative) sets of ontologies involved in the derivation of a particular triple. In this section, we define the provenance simple interpretations and provenance RDFS interpretations w.r.t. a g-RDF information base $\mathcal{C}$. Additionally, we define satisfaction of a provenance graph by a provenance simple interpretation. These notions will be used in defining the provenance stable models of $\mathcal{C}$ that takes care of the semantics of rules and weak negation.

Convention: In the rest of the paper, we consider a g-RDF information base $\mathcal{C}$.

Below, we review the definition of a provenance simple interpretation of a vocabulary V and $\mathcal{C}$, which is an extension of the definition of a simple interpretation of V [39], such that each property p is associated with a property-truth extension that is a set of triples $\langle s, o, S\rangle$, where $S \subseteq names_{\mathcal{C}}$ is the set of g-RDF ontologies that contributed to the derivation that $\langle s, o\rangle$ belongs to the property-truth extension of p. Additionally, each property p is associated with a property-falsity extension that is a set of triples $\langle s, o, S\rangle$, where $S \subseteq names_{\mathcal{C}}$ is the set of g-RDF ontologies that contributed to the derivation that $\langle s, o\rangle$ belongs to the property-falsity extension of p. We refer to S as *set of provenance names*.

Definition 4 (Provenance simple interpretation). A *provenance simple interpretation I* of a vocabulary V_I and $\mathcal{C}$ consists of:

– A non-empty set of resources Res_I called the *domain* or *universe* of I.
– A set of properties $Prop_I$.
– A vocabulary interpretation mapping $I_{\mathtt{V}} : V_I \cap URI \rightarrow Res_I \cup Prop_I$.
– A property-truth extension mapping $PT_I : Prop_I \rightarrow \mathcal{P}(Res_I \times Res_I \times \mathcal{P}(names_{\mathcal{C}}))$.
– A property-falsity extension mapping $PF_I : Prop_I \rightarrow \mathcal{P}(Res_I \times Res_I \times \mathcal{P}(names_{\mathcal{C}}))$.

Ontology O_1

$\langle http://general\_info.int \rangle$

$G_{O_1} =$
rdfs:subClassOf(ex:Adult,ex:Person).
rdfs:subClassOf(ex:Heavy_Job,ex:Job).
owl:sameAs(ex:gets_pension,
ex:pension_age).

Ontology O_2

$\langle http://persons\_info.gr \rangle$

$G_{O_2} =$
rdfs:domain(ex:has_job,ex:Adult).
rdfs:range(ex:has_job,ex:Job).
ex:has_job(ex:Mary,ex:hairdresser).
ex:has_job(ex:Peter,ex:garbage_collector).

Ontology O_3

$\langle http://jobs\_info\_2000\_to\_2009.gr \rangle$

$G_{O_3} =$
rdf:type(ex:hairdresser,
ex:Heavy_Job).
rdf:type(ex:garbage_collector,
ex:Heavy_Job).

Ontology O_4

$\langle http://jobs\_info\_2010\_to\_2013.gr \rangle$

$G_{O_4} =$
rdf:type(ex:garbage_collector,
ex:Heavy_Job).

Ontology O_5

$\langle http://pensions\_info\_2000\_to\_2009.gr \rangle$

$P_{O_5} =$
ex:gets_pension($?x$, "55"^^$xsd{:}integer$) $\leftarrow$ ex:has_job($?x, ?y$), rdf:type($?y$,ex:Heavy_Job).
ex:gets_pension($?x$, "60"^^$xsd{:}integer$) $\leftarrow$ ex:has_job($?x, ?y$),
$\sim$rdf:type($?y$,ex:Heavy_Job)@$\{Nam_{O_3}\}$.

Ontology O_6

$\langle http://pensions\_info\_2010\_to\_2013.gr \rangle$

$P_{O_6} =$
ex:gets_pension($?x$, "62"^^$xsd{:}integer$) $\leftarrow$ ex:has_job($?x, ?y$), rdf:type($?y$,ex:Heavy_Job).
ex:gets_pension($?x$, "65"^^$xsd{:}integer$) $\leftarrow$ ex:has_job($?x, ?y$),
$\sim$rdf:type($?y$,ex:Heavy_Job)@$\{Nam_{O_4}\}$.

Ontology O_7

$\langle http://sameAs\_info.com \rangle$

$P_{O_7} =$
owl:sameAs($?x, ?x$) $\leftarrow$.
owl:sameAs($?x, ?y$) $\leftarrow$ owl:sameAs($?y, ?x$).
owl:sameAs($?x, ?z$) $\leftarrow$ owl:sameAs($?x, ?y$), owl:sameAs($?y, ?z$).
$?p(?x, ?y)$ $\leftarrow$ owl:sameAs($?p, ?p'$), owl:sameAs($?x, ?x'$), owl:sameAs($?y, ?y'$), $?p'(?x', ?y')$.

Conflicts

conflict($http://jobs\_info\_2000\_to\_2009.gr, http://jobs\_info\_2010\_to\_2013.gr$).
conflict($http://jobs\_info\_2000\_to\_2009.gr, http://pensions\_info\_2010\_to\_2013.gr$).
conflict($http://jobs\_info\_2010\_to\_2013.gr, http://pensions\_info\_2000\_to\_2009.gr$).
conflict($http://pensions\_info\_2000\_to\_2009.gr, http://pensions\_info\_2010\_to\_2013.gr$).

Fig. 1. A g-RDF information base

– A mapping $IL_I : V_I \cap \mathcal{TL} \to Res_I$.
– A set of literal values $LV_I \subseteq Res_I$, which contains $V \cap \mathcal{PL}$.

We define the mapping: $I : V_I \to Res_I \cup Prop_I$ such that: (i) $I(x) = I_V(x),\ \forall x \in V_I \cap URI$, (ii) $I(x) = x,\ \forall\, x \in V_I \cap \mathcal{PL}$, and (iii) $I(x) = IL_I(x),\ \forall\, x \in V_I \cap \mathcal{TL}$. □

The differences to the original definition of RDF interpretation is the inclusion of the property-falsity extension mapping, and the inclusion of $\mathcal{P}(names_{\mathcal{C}})$ on both extension mappings collecting the provenance information. Note that according to RDF semantics [39], (i) the mapping IL_I maps typed literals in the vocabulary of I to the resources of I and (ii) LV_I is the set of literal values of I (a subset of the resources of I), which includes all plain literals in the vocabulary of I.

Example 2. Consider the simple g-RDF information base $\mathcal{C}$ of Figure 2. Let:

$V = \{$`ex:Mary`, `ex:gets_pension`, `ex:has_job`, `ex:hairdresser`, `ex:garbage_collector`, `rdf:type`, `ex:Heavy_Job`, "65"^^*xsd:integer*$\}$

and consider a structure I that consists of:

– A set of resources $Res_I = \{M, g, h, h2, g2, t, He, 65\}$.
– A set of properties $Prop_I = \{g, h, t\}$.
– A vocabulary interpretation mapping $I_V : V \cap URI \to Res_I \cup Prop_I$ such that: $I_V($`ex:Mary`$) = M$, $I_V($`ex:gets_pension`$) = g$, $I_V($`ex:has_job`$) = h$, $I_V($`ex:hairdresser`$) = h2$, $I_V($`ex:garbage_collector`$) = g2$, $I_V($`rdf:type`$) = t$, and $I_V($`ex:Heavy_Job`$) = He$.
– A property-truth extension mapping $PT_I : Prop_I \to \mathcal{P}(Res_I \times Res_I \times \mathcal{P}(names_{\mathcal{C}}))$ such that:
$PT_I(h) = \{\langle M, h2, \{Nam_{O_1}\}\rangle\}$,
$PT_I(t) = \{\langle g2, He, \{Nam_{O_2}\}\rangle\}$, and
$PT_I(g) = \{\langle M, 65, \{Nam_{O_1}, Nam_{O_3}\}\rangle\}$.
– A property-falsity extension mapping $PT_I : Prop_I \to \mathcal{P}(Res_I \times Res_I \times \mathcal{P}(names_{\mathcal{C}}))$ such that: $PF_I(h) = \{\langle M, g2, \{Nam_{O_1}\}\rangle\}$.
– A mapping $IL_I : V \cap \mathcal{TL} \to Res_I$ such that: $IL_I($"65"^^*xsd:integer*$) = 65$.
– A set of literal values $LV_I = \{$"65"^^*xsd:integer*$\}$.

It is easy to see that I is a provenance simple interpretation of V and $\mathcal{C}$, expressing that: (i) Mary is a hairdresser according to g-RDF ontology O_1, (ii) Mary is not a garbage collector according to g-RDF ontology O_1, (iii) garbage collector is a heavy job according to g-RDF ontology O_2, and (iv) Mary will get pension at age 65 according to g-RDF ontologies O_1 and O_3. □

Below, we define a coherent provenance simple interpretation of a vocabulary V and $\mathcal{C}$ as a provenance simple interpretation for which the property-truth extension and the falsity-truth extension of each property are disjoint.

Definition 5 (Coherent provenance simple interpretation). Let I be a provenance simple interpretation of a vocabulary V and $\mathcal{C}$. I is a *coherent provenance simple interpretation* if for all $p \in Prop_I$, $PT_I(p) \cap PF_I(p) = \emptyset$. □

Now, we define a provenance quad as $[\neg]H(s, p, o, S)$, where S is a set of provenance names. Note that H is a new predicate symbol for holding quadruples, and extend the notion of quads used in RDF triplestores by annotating a triple with a set of ontology names instead of a single one. We require this extra structure since rules allow variable property names (see ontology O_7 in Fig. 1), not needing in this way to resort to second order logic.

Ontology O_1

$\langle \mathit{http://persons\_info.gr} \rangle$

$G_{O_1} =$
`ex:has_job(ex:Mary,ex:hairdresser).`
$\neg$ `ex:has_job(ex:Mary,ex:garbage_collector).`

Ontology O_2

$\langle \mathit{http://jobs\_info.gr} \rangle$

$G_{O_2} =$
`rdf:type(ex:garbage_collector,`
`ex:Heavy_Job).`

Ontology O_3

$\langle \mathit{http://pensions\_info.gr} \rangle$

$P_{O_3} =$
`ex:gets_pension`$(?x, \text{"65"\textasciicircum\textasciicircum}\mathit{xsd{:}integer}) \leftarrow$ `ex:has_job`$(?x, ?y)$,
$\sim$`rdf:type`$(?y,$`ex:Heavy_Job`$)@\{\langle \mathit{http://jobs\_info.gr} \rangle\}$.

Conflicts$=\emptyset$.

Fig. 2. A simple g-RDF information base

Definition 6 (Provenance quad and provenance graph). A *provenance (RDF) quad* of $\mathcal{C}$ is a quadruple $[\neg]H(s,p,o,S)$, where $p(s,o)$ is a g-RDF triple and $S \subseteq \mathit{names}_{\mathcal{C}}$.

A *provenance (RDF) graph* of $\mathcal{C}$ is a set of provenance quads of $\mathcal{C}$. Let G be a provenance graph or provenance quad, we denote by V_G the set of URI references and literals appearing in G. $\square$

Note that the definition of a provenance graph stays close to the definition of g-RDF graph, where g-RDF triples have been replaced by provenance quads.

Example 3. Consider the g-RDF information base $\mathcal{C}$ of Example 2. Then, $H($`ex:Mary`, `ex:gets_pension`, $\text{"65"\textasciicircum\textasciicircum}\mathit{xsd{:}integer}, \{\mathit{Nam}_{O_1}, \mathit{Nam}_{O_3}\})$ is a provenance quad of $\mathcal{C}$. $\square$

It is important to realize that a triple can occur in several different quadruples, capturing alternative different ontologies supporting the derivation of the triple. This is different from the approaches used in the literature where a triple has a single annotation, usually represented as $(s,p,o){:}a$, following the principles of annotated relational databases. The use of quadruples is essential to simplify the definitions of provenance stable models later on. So, the set of quadruples $H(s,p,o,S_1),\ldots,H(s,p,o,S_m)$ corresponds to a RDF triple $(s,p,o){:}\{S_1,\ldots,S_m\}$ annotated with elements of the why-provenance semiring, and vice-versa.

To define satisfaction of a provenance quad or provenance graph w.r.t. a provenance simple interpretation, we need first the following auxiliary definition of valuation that interprets variables.

Definition 7 (Composition of a provenance simple interpretation and a valuation). Let I be a provenance simple interpretation of a vocabulary V and $\mathcal{C}$ and let v be a partial function $v : \mathit{Var} \rightarrow \mathit{Res}_I$ (called *valuation*). We define: (i) $[I+v](x) = v(x)$, if $x \in \mathit{Var}$, and (ii) $[I+v](x) = I(x)$, if $x \in V$. $\square$

Definition 8. (Satisfaction of a provenance quad w.r.t. a provenance simple interpretation and a valuation) Let t be a provenance quad and let I be a provenance simple interpretation of a vocabulary V and $\mathcal{C}$. Additionally, let v be a mapping $v : \mathit{Var}(t) \rightarrow \mathit{Res}_I$.

- If $t = H(s,p,o,S)$ then $I, v \models t$ iff $p \in (V \cap \mathit{URI}) \cup \mathit{Var}$, $s,o \in V \cup \mathit{Var}$, $[I+v](p) \in \mathit{Prop}_I$, and $\langle [I+v](s), [I+v](o), S \rangle \in \mathit{PT}_I([I+v](p))$.
- $t = \neg H(s,p,o,S)$ then $I, v \models t$ iff $p \in (V \cap \mathit{URI}) \cup \mathit{Var}$, $s,o \in V \cup \mathit{Var}$, $[I+v](p) \in \mathit{Prop}_I$, and $\langle [I+v](s), [I+v](o), S \rangle \in \mathit{PF}_I([I+v](p))$.

– $I, v \models {\sim}t$ iff $I, v \not\models t$. □

Definition 8 essentially says that a provenance simple interpretation I and a valuation v satisfy a provenance quad $H(s, p, o, S)$ if the pair of the mapping of s and o according to $I + v$ belongs to the property-truth of the mapping of p according to $I + v$. Similarly, I and v satisfy a provenance quad $\neg H(s, p, o, S)$ if the pair of the mapping of s and o according to $I + v$ belongs to the property-falsity of the mapping of p according to $I + v$. Now, I and v satisfy a weakly negated provenance quad ${\sim}t$ if (similarly to first-order logic) I and v do not satisfy t.

Below, we define satisfaction of a provenance graph w.r.t. a provenance simple interpretation and a valuation in order to be compatible with the RDFS semantics that defines satisfaction of an RDF graph w.r.t. a simple interpretation and a valuation.

Definition 9. (Satisfaction of a provenance graph w.r.t. a provenance simple interpretation and a valuation) Let G be a provenance graph and let I be a provenance simple interpretation of a vocabulary V and $\mathcal{C}$. Additionally, let v be a mapping $v : \mathit{Var}(G) \to \mathit{Res}_I$. If $G = \{t_1, ..., t_n\}$ then $I, v \models G$ iff $I, v \models t_i$, for all $i = 1, ..., n$. □

Example 4. Continuing Example 2, let $v : \{?x, ?y\} \to \mathit{Res}_I$ such that $v(?x) = M$ and $v(?y) = h2$. It holds that:

$I, v \models \{H(?x, \texttt{ex:has\_job}, ?y, \{\mathit{Nam}_{O_1}\}),$
$\neg H(?x, \texttt{ex:has\_job}, \texttt{ex:garbage\_collector}, \{\mathit{Nam}_{O_1}\}),$
$H(?x, \texttt{ex:gets\_pension}, "65"\hat{}\hat{}\mathit{xsd{:}integer}, \{\mathit{Nam}_{O_1}, \mathit{Nam}_{O_3}\})\}$. □

Note that variables in RDF graphs, according to RDFS semantics, are existentially quantified [39]. Similar is the case, in our theory, for variables in provenance graphs.

Definition 10 (Satisfaction of a provenance quad and provenance graph w.r.t. a provenance simple interpretation). Let G be a provenance quad or the weak negation of a provenance quad, or a provenance graph. Additionally, let I be a provenance simple interpretation of a vocabulary V and $\mathcal{C}$. We say that I *satisfies* G, denoted by $I \models G$, iff there exists a mapping $v : \mathit{Var}(G) \to \mathit{Res}_I$ s.t. $I, v \models G$. □

Let I be a coherent provenance simple interpretation of a vocabulary V and $\mathcal{C}$. It is easy to see that, based on Definition 5, there are no $s, p, o \in V$ and $S \subseteq \mathit{names}_{\mathcal{C}}$ s.t. $I \models H(s, p, o, S)$ and $I \models \neg H(s, p, o, S)$. This implies that we cannot derive conflicting statements from the same set of sources. However, as we will see in Section 4, it is possible that there exist S, S' s.t. $I \models H(s, p, o, S)$ and $I \models \neg H(s, p, o, S')$, if there exists $c \in S$ and $c' \in S'$ s.t. $\mathit{conflict}(c, c')$ holds. In particular, Example 9 of Section 4 provides an example of opposite conclusions from conflicting namespaces.

Example 5. Consider the g-RDF information base $\mathcal{C}$ of Example 2. Then, there is no coherent provenance simple interpretation of a vocabulary V and $\mathcal{C}$ s.t.
$I \models H(\texttt{ex:Mary}, \texttt{ex:gets\_pension}, "65"\hat{}\hat{}\mathit{xsd{:}integer}, \{\mathit{Nam}_{O_1}, \mathit{Nam}_{O_3}\})$ and
$I \models \neg H(\texttt{ex:Mary}, \texttt{ex:gets\_pension}, "65"\hat{}\hat{}\mathit{xsd{:}integer}, \{\mathit{Nam}_{O_1}, \mathit{Nam}_{O_3}\})$. □

The semantics of RDF Schema vocabulary, which includes `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdfs:domain`, and `rdfs:range`, requires additional constraints on simple interpretations. Technically, this is achieved by enforcing a large number of extra semantic conditions. To capture provenance we have also to reflect the propagation of annotations in this definition as well as their compatibility. Below we review the definition of a provenance RDFS interpretation of a vocabulary V and $\mathcal{C}$, which is an extension of the definition of an RDFS interpretation of V [39], such that sets of provenance names and the predicate $\mathit{compatible}(.)$ appear in all provenance RDFS conditions. Let $S \subseteq \mathit{names}_{\mathcal{C}}$ then $\mathit{compatible}(S)$ means that there are not $c, c' \in S$ s.t. $\mathit{conflict}(c, c')$. We denote by $\mathcal{V}_{RDF}$ the vocabulary of RDF and by $\mathcal{V}_{RDFS}$ the vocabulary of RDFS [39].

Definition 11 (provenance RDFS interpretation). A *provenance* RDFS *interpretation* I of a vocabulary V and $\mathcal{C}$ is a coherent provenance simple interpretation of $V \cup \mathcal{V}_{RDF} \cup \mathcal{V}_{RDFS}$ and $\mathcal{C}$ that satisfies the following *provenance RDFS conditions*:

1. $Prop_I = \{x \mid \exists S \subseteq names_{\mathcal{C}}, \langle x, I(\texttt{rdf:Property}), S\rangle\rangle \in PT_I(I(\texttt{rdf:type}))\}$.
   $Res_I = \{x \mid \exists S \subseteq names_{\mathcal{C}}, \langle x, I(\texttt{rdfs:Resource}), S\rangle\rangle \in PT_I(I(\texttt{rdf:type}))\}$.
   $LV_I = \{x \mid \exists S \subseteq names_{\mathcal{C}}, \langle x, I(\texttt{rdfs:Literal}), S\rangle\rangle \in PT_I(I(\texttt{rdf:type}))\}$.
2. If $c \in names_{\mathcal{C}}$ and $x \in Res_I \cap I(V_c)$ then $\langle x, I(\texttt{rdfs:Resource}), \{c\}\rangle \in PT_I(I(\texttt{rdf:type}))$.
3. If $c \in names_{\mathcal{C}}$ and $x \in \mathcal{PL} \cap I(V_c)$ then $\langle x, I(\texttt{rdfs:Literal}), \{c\}\rangle \in PT_I(I(\texttt{rdf:type}))$.
4. If $\langle x, y, S\rangle \in PT_I(z)$ then $\langle z, I(\texttt{rdf:Property}), S\rangle \in PT_I(I(\texttt{rdf:type}))$.
5. If $\langle x, y, S_1\rangle \in PT_I(I(\texttt{rdfs:domain}))$, $\langle z, w, S_2\rangle \in PT_I(x)$, and $compatible(S_1 \cup S_2)$ then $\langle z, y, S_1 \cup S_2\rangle \in PT_I(I(\texttt{rdf:type}))$.
6. If $\langle x, y, S_1\rangle \in PT_I(I(\texttt{rdfs:range}))$, $\langle z, w, S_2\rangle \in PT_I(x)$, and $compatible(S_1 \cup S_2)$ then $\langle w, y, S_1 \cup S_2\rangle \in PT_I(I(\texttt{rdf:type}))$.
7. If $\langle x, y, S\rangle \in PT_I(I(\texttt{rdf:type}))$ then $\langle x, I(\texttt{rdfs:Resource}), S\rangle \in PT_I(I(\texttt{rdfs:subClassOf}))$.
8. If $\langle x, y, S_1\rangle \in PT_I(I(\texttt{rdfs:subClassOf}))$, $\langle z, x, S_2\rangle \in PT_I(I(\texttt{rdf:type}))$, and $compatible(S_1 \cup S_2)$ then $\langle z, y, S_1 \cup S_2\rangle \in PT_I(I(\texttt{rdf:type}))$.
9. If $\langle x, y, S_1\rangle \in PT_I(I(\texttt{rdfs:subClassOf}))$, $\langle z, y, S_2\rangle \in PF_I(I(\texttt{rdf:type}))$, and $compatible(S_1 \cup S_2)$ then $\langle z, x, S_1 \cup S_2\rangle \in PF_I(I(\texttt{rdf:type}))$.
10. If $\langle x, \texttt{rdfs:Class}, S\rangle \in PT_I(I(\texttt{rdf:type}))$ then $\langle x, x, S\rangle \in PT_I(I(\texttt{rdfs:subClassOf}))$.
11. If $\langle x, y, S_1\rangle \in PT_I(I(\texttt{rdfs:subClassOf}))$, $\langle y, z, S_2\rangle \in PT_I(I(\texttt{rdfs:subClassOf}))$, and $compatible(S_1 \cup S_2)$ then $\langle x, z, S_1 \cup S_2\rangle \in PT_I(I(\texttt{rdfs:subClassOf}))$.
12. If $\langle x, y, S_1\rangle \in PT_I(I(\texttt{rdfs:subPropertyOf}))$, $\langle z, w, S_2\rangle \in PT_I(I(x))$, and $compatible(S_1 \cup S_2)$ then $\langle z, w, S_1 \cup S_2\rangle \in PT_I(I(y))$.
13. If $\langle x, y, S_1\rangle \in PT_I(I(\texttt{rdfs:subPropertyOf}))$, $\langle z, w, S_2\rangle \in PF_I(I(y))$, and $compatible(S_1 \cup S_2)$ then $\langle z, w, S_1 \cup S_2\rangle \in PF_I(I(x))$.
14. If $\langle x, I(\texttt{rdf:Property}), S\rangle \in PT_I(I(\texttt{rdf:type}))$ then $\langle x, x, S\rangle \in PT_I(I(\texttt{rdfs:subPropertyOf}))$.
15. If $\langle x, y, S_1\rangle \in PT_I(I(\texttt{rdfs:subProperyOf}))$, $\langle y, z, S_2\rangle \in PT_I(I(\texttt{rdfs:subPropertyOf}))$, and $compatible(S_1 \cup S_2)$ then $\langle x, z, S_1 \cup S_2\rangle \in PT_I(I(\texttt{rdfs:subProperyOf}))$.
16. If $\langle x, I(\texttt{rdfs:Datatype}), S\rangle \in PT_I(I(\texttt{rdf:type}))$ then $\langle x, I(\texttt{rdfs:Literal}), S\rangle \in PT_I(I(\texttt{rdfs:subClassOf}))$.
17. If $\langle x, I(\texttt{rdfs:ContainerMembershipProperty}), S\rangle \in PT_I(I(\texttt{rdf:type}))$ then $\langle x, I(\texttt{rdfs:member}), S\rangle \in PT_I(I(\texttt{rdfs:subPropertyOf}))$.
18. If $"s"\hat{}\hat{}rdf{:}XMLLiteral \in V$ and s is a well-typed XML literal then $IL_I("s"\hat{}\hat{}rdf{:}XMLLiteral)$ is the XML value of s.
19. If $c \in names_{\mathcal{C}}$, $"s"\hat{}\hat{}rdf{:}XMLLiteral \in V \cap V_c$, and s is a well-typed XML literal then $\langle IL_I("s"\hat{}\hat{}rdf{:}XMLLiteral), I(\texttt{rdf:XMLLiteral}), \{c\}\rangle \in PT_I(I(\texttt{rdf:type}))$.
20. If $"s"\hat{}\hat{}rdf{:}XMLLiteral \in V$ and s is an ill-typed XML literal then $IL_I("s"\hat{}\hat{}rdf{:}XMLLiteral) \in Res_I - LV_I$.
21. If $p(s, o)$ is an RDF or RDFS axiomatic triple with terms in $\mathcal{V}_{RDF} \cup \mathcal{V}_{RDFS}$ [39] then $I \models H(s, p, o, \{\})$.
22. If $\langle x, y, S'\rangle \in PT_I(z)$, $S' \subseteq S$, and $compatible(S)$ holds then $\langle x, y, S\rangle \in PT_I(z)$.
23. If $\langle x, y, S'\rangle \in PF_I(z)$, $S' \subseteq S$, and $compatible(S)$ holds then $\langle x, y, S\rangle \in PF_I(z)$. $\square$

In the definition of a provenance RDFS interpretation, for simplicity, we do not consider extensions of classes as in the definition of an RDFS interpretation. This does not result in any loss of generality, as the extension of classes is defined based on the extension of the property `rdf:type`. Note that in the provenance RDFS conditions 2,3, and 19 above, the set of provenance names is a singleton. Additionally, in the provenance RDFS condition 21 above, the set of provenance names is the empty set. Further, note that the provenance RDFS conditions 9, 13, and 23 refer to the falsity-extension of properties. Additionally, note that semantic conditions 2,3, and 22 (and of course conditions 9, 13, and 23) have no corresponding RDFS condition.

Semantic conditions 2, 3, and 19 are due to the fact that the set of provenance names $\{c\}$ should be associated to each resource, plain literal, and well-typed XML literal appearing in a g-RDF ontology c. Semantic condition 4 expresses that if a triple $\langle x, y, S\rangle$ belongs to the property-truth extension of a resource then this resource should be of type property according to the set of provenance names S. Semantic condition 5 expresses that if y is the domain of a property x according to a set of provenance names S_1, $\langle z, w, S_2\rangle$ belongs to the property-truth extension of x, and $\mathit{compatible}(S_1, S_2)$ holds then z is of type y according to the set of provenance names $S_1 \cup S_2$. Semantic condition 9 expresses that if x is a subclass of y according to a set of provenance names S_1, z is not of type y according to a set of provenance names S_2, and $\mathit{compatible}(S_1, S_2)$ holds then z is not of type x according to a set of provenance names $S_1 \cup S_2$. Semantic conditions 18 and 20 are the same as corresponding conditions in the definition of an RDFS interpretation. Finally, semantic conditions 22 and 23 are based on the intuition that if $[\neg]p(s, o)$ is derived from a set of g-RDF ontologies S', $S' \subseteq S$, and $\mathit{compatible}(S)$ holds then $[\neg]p(s, o)$ is also derived from S. Now, the rest of the semantic conditions should be easily understandable.

Note that if $"s"\hat{}\hat{}\mathit{rdf{:}XMLLiteral} \in V$, where s is an ill-typed XML literal, and it exists $S \subseteq \mathit{names}_{\mathcal{C}}$ s.t. $\langle I("s"\hat{}\hat{}\mathit{rdf{:}XMLLiteral}),\ I(\texttt{rdfs:Literal}),\ S\rangle \in P_I(I(\texttt{rdf:type}))$ then an *XML clash* is caused due to provenance RDFS conditions 1 and 20. This notion of XML clash extends the notion of XML clash of RDFS semantics, which is the only case that an RDF graph does not have an RDFS model [39].

4 Provenance Stable Model Semantics

In this section, we define the provenance Herbrand interpretations and provenance stable models of a g-RDF information base $\mathcal{C}$.

Variables in general RDF graphs correspond to blank nodes, which denote anonymous resources that are implicitly existentially quantified. We handle existentially quantified variables in g-RDF graphs by *skolemization*, a syntactic transformation commonly used in automatic inference systems for removing existentially quantified variables.

Definition 12 (Skolemization of a g-RDF graph). Let G be a g-RDF graph. The *skolemization function* of G is an 1:1 mapping $sk_G : \mathit{Var}(G) \rightarrow \mathit{URI}$, where for each $x \in \mathit{Var}(G)$, $sk_G(x)$ is an artificial URI. The *skolemization* of G, denoted by $sk(G)$, is the ground g-RDF graph derived from G after replacing each variable $x \in \mathit{Var}(G)$ by $sk_G(x)$. $\square$

Intuitively, the Skolem vocabulary of G (that is, $sk_G(\mathit{Var}(G))$) contains artificial URIs giving "arbitrary" names to the anonymous entities whose existence was asserted by the use of variables in G.

Example 6. Let: $G = \{\texttt{rdf:type}(?x, \texttt{ex:EuropeanCountry}), \texttt{rdf:type}(?x, \texttt{ex:EUmember})\}$. Then,

$sk(G) = \{\texttt{rdf:type}(sk_G(?x), \texttt{ex:EuropeanCountry}), \texttt{rdf:type}(sk_G(?x), \texttt{ex:EUmember})\}$. □

It is well-known [39] that if G, G' are RDF graphs and $V_{G'} \cap sk_G(Var(G)) = \emptyset$ then it holds that: $G \models^{RDFS} G'$ iff $sk(G) \models^{RDFS} G'$.

Below we define the vocabulary of a g-RDF information base $\mathcal{C}$.

Definition 13 (Vocabulary of a g-RDF information base). Consider a g-RDF information base $\mathcal{C}$, which contains the g-RDF ontologies $O_i = \langle G_i, P_i \rangle$, for $i = 1, ..., m$. The *vocabulary* of $\mathcal{C}$ is defined as $V_{\mathcal{C}} = \bigcup_{i=1,...,m}(V_{sk(G_i)} \cup V_{P_i}) \cup \mathcal{V}_{RDF} \cup \mathcal{V}_{RDFS}$. □

Note that the vocabulary of a g-RDF information base $\mathcal{C}$ contains the skolemization vocabulary of each g-RDF graph appearing in $\mathcal{C}$, the vocabulary of each g-RDF program appearing and $\mathcal{C}$, and the RDF and RDFS vocabularies.

We denote by $Res^H_{\mathcal{C}}$ the union of $V_{\mathcal{C}}$ and the set of XML values of the well-typed XML literals in $V_{\mathcal{C}}$ minus the well-typed XML literals. In other words, the set of Herbrand resources $Res^H_{\mathcal{C}}$ is $V_{\mathcal{C}}$ with the well-typed XML literals substituted by their XML values.

The following definition defines the provenance Herbrand interpretations of $\mathcal{C}$. This definition is needed for defining the provenance stable models of $\mathcal{C}$.

Definition 14 (Provenance Herbrand interpretation). Let I be a provenance RDFS interpretation of $V_{\mathcal{C}}$ and $\mathcal{C}$. We say that I is a *provenance Herbrand interpretation* of $\mathcal{C}$ iff[14]:

– $Res_I = Res^H_{\mathcal{C}}$.
– $I_V(x) = x$, for all $x \in V_{\mathcal{C}} \cap URI$.
– $IL_I(x) = x$, if[15] x is a typed literal in $V_{\mathcal{C}}$ other than a well-typed XML literal, and $IL_I(x)$ is the XML value of x, if x is a well-typed XML literal in $V_{\mathcal{C}}$.

We denote the set of provenance Herbrand interpretations of $\mathcal{C}$ by $\mathcal{I}^H(\mathcal{C})$. □

Note that we first defined the notion of a provenance RDFS interpretation and then the notion of a provenance Herbrand interpretation, in order to faithfully extend the RDFS semantics [39]. As usual in the construction of Herbrand interpretations, we map every constant to itself except for the predefined XML literals that have a fixed interpretation in the original RDFS semantics. Accordingly, every provenance Herbrand interpretation of $\mathcal{C}$ is uniquely identified by (i) its set of properties and (ii) its property-truth and property-falsity extension mappings.

Definition 15 (Provenance rule). Let r be a g-RDF rule of a g-RDF ontology O of the form:

$$[\neg]p_0(s_0, o_0) \leftarrow [\neg]p_1(s_1, o_1), ..., [\neg]p_l(s_l, o_l),$$
$$\sim[\neg]p_{l+1}(s_{l+1}, o_{l+1})@N_{l+1}, ..., \sim[\neg]p_k(s_k, o_k)@N_k.$$

Based on r, we generate the *provenance (RDF) rule*, denoted by $prov(r)$, as follows:[16]

$$[\neg]H(s_0, p_0, o_0, ?S) \leftarrow [\neg]H(s_1, p_1, o_1, ?S_1),\ ...,\ [\neg]H(s_l, p_l, o_l, ?S_l),$$
$$\sim[\neg]H(s_{l+1}, p_{l+1}, o_{l+1}, N_{l+1}),\ ...,\ \sim[\neg]H(s_k, p_k, o_k, N_k),$$
$$union(\{?S_1, ..., ?S_l,\ \{Nam_O\}\},\ ?S),\ compatible(?S).$$

Let P be the g-RDF program of a g-RDF ontology O. We define the *provenance (RDF) program* $prov(P) = \bigcup_{r \in P}\{prov(r)\}$. □

[14] Note that due to semantic condition 18 of Definition 11, any provenance RDFS interpretation I maps well-typed XML literals to their XML values. This is because the same is done by any RDFS interpretation. The notion of XML value is fully described in Section 5.1 of [45].

[15] Note that this item is in accordance with semantic conditions 18, 19 and 20 of Definition 11.

[16] Note that $union(SS, S)$ indicates that S is the union of the sets in SS.

Let r be a g-RDF rule of a g-RDF ontology O of the form: $[\neg]p_0(s_0, o_0) \leftarrow [\neg]p_1(s_1, o_1)$, ..., $[\neg]p_l(s_l, o_l)$, $\sim[\neg]p_{l+1}(s_{l+1}, o_{l+1})@N_{l+1}$, ..., $\sim[\neg]p_k(s_k, o_k)@N_k$. The generation of $prov(r)$ is based on the intuition that:

- If (i) $[\neg]p_i(s_i, o_i)$ is derived from a set of g-RDF ontologies S_i, for $i = 1, ..., l$, and (ii) $[\neg]p_i(s_i, o_i)$ is not derived from the set of g-RDF ontologies N_i, for $i = l+1, ..., k$, then $[\neg]p_0(s_0, o_0)$ is derived from $S = S_1 \cup ... \cup S_l \cup \{Nam_O\}$, provided that $compatible(S)$ holds.

Example 7. The rule $H(?x$, `ex:gets_pension`, "65"^^*xsd:integer*, $?S) \leftarrow H(?x$, `ex:has_job`, $?y, ?S_1)$, $\sim H(?y$, `rdf:type`, `ex:HeavyJob`, $\{Nam_{O_4}\})$, $union(\{?S_1,\ \{Nam_{O_6}\}\},\ ?S)$, $compatible(?S)$ is a provenance rule generated by the second rule of the g-RDF ontology O_6 in Figure 1. □

To define the provenance models of a g-RDF information base $\mathcal{C}$, we need first to define grounding of provenance rules. Grounding a provenance rule r, we replace (i) every variable appearing in a property position by a URI in $V_{\mathcal{C}}$, (ii) every variable appearing in a subject or object position by a constant in $V_{\mathcal{C}}$, and (iii) every variable appearing in a provenance position by a subset of $names_{\mathcal{C}}$.

Definition 16 (Grounding of a provenance program). Let r be a provenance rule:

$$\begin{aligned}[\neg]H(s_0, p_0, o_0, ?S) \leftarrow\ & [\neg]H(s_1, p_1, o_1, ?S_1),\ ...,\ [\neg]H(s_l, p_l, o_l, ?S_l),\\ & \sim[\neg]H(s_{l+1}, p_{l+1}, o_{l+1}, N_{l+1}),\ ...,\ \sim[\neg]H(s_k, p_k, o_k, N_k),\\ & union(\{?S_1, ..., ?S_l,\ \{Nam_O\}\},\ ?S),\ compatible(?S).\end{aligned}$$

We denote by $[r]_{\mathcal{C}}$ the set of rules that result from r if (i) we replace each variable in $\{p_0, ..., p_k\}$ by a constant in $V_{\mathcal{C}} \cap URI$, (ii) we replace each variable in $\{s_0, o_0, ..., s_k, o_k\}$ by a constant in $V_{\mathcal{C}}$, (iii) we replace each $?S_1, ..., ?S_k, ?S$ by $u(?S_1), ...u(?S_k), u(?S)$, which are subsets of $names_{\mathcal{C}}$, (iv) $u(?S) = u(?S_1) \cup ... \cup u(?S_l) \cup \{Nam_O\}$,[17] (v) $compatible(u(?S))$ holds, and (vi) we eliminate the $union(.)$ and $compatible(.)$ predicates.

Let P be the g-RDF program of a g-RDF ontology O. We define $[prov(P)]_{\mathcal{C}} = \bigcup_{r \in P}[prov(r)]_{\mathcal{C}}$. □

Note that a rule variable can naturally appear in the subject position of a g-RDF triple. Since variables can be instantiated by a literal, a literal can naturally appear in the subject position s_i of a provenance quad $[\neg]H(s_i, p_i, o_i, S_i)$ in the body or head of a rule in $[prov(P)]_{\mathcal{C}}$. This case further supports our choice of allowing literals in the subject position of a g-RDF triple.

Example 8. Consider the provenance rule r of example 7. Then, the rule $H($`ex:Mary`, `ex:gets_pension`, "65"^^*xsd:integer*, $\{Nam_{O_2}, Nam_{O_6}\}) \leftarrow H($`ex:Mary`, `ex:has_job`, `ex:hairdresser`, $\{Nam_{O_2}\})$, $\sim H($`ex:hairdresser`, `rdf:type`, `ex:HeavyJob`, $\{Nam_{O_4}\})$ belongs to $[r]_{\mathcal{C}}$. □

Before we define the provenance stable models of $\mathcal{C}$, we need to define a partial ordering on the provenance Herbrand interpretations of $\mathcal{C}$.

Before, we present our semantics, we first need a way of comparing interpretations in order to be able to minimize information, i.e., to define the semantics of weak negation according to the principle of negation by default.

Definition 17 (Provenance Herbrand interpretation ordering). Let $I, J \in \mathcal{I}^H(\mathcal{C})$. We say that J *extends* I, denoted by $I \leq J$ (or $J \geq I$), iff $Prop_I \subseteq Prop_J$, and for all $p \in Prop_I$, it holds that $PT_I(p) \subseteq PT_J(p)$ and $PF_I(p) \subseteq PF_J(p)$. □

[17] Obviously, $u(?s) \subseteq names_{\mathcal{C}}$.

It is easy to verify that the relation $\leq$ is reflexive, transitive, and antisymmetric. Thus, it is a partial ordering on $\mathcal{I}^H(\mathcal{C})$. The idea is that we prefer models which minimize the property-truth extensions of properties (i.e., what we know that is true) and the property-falsity extensions of properties (i.e., what we know that is false).

Definition 18 (Provenance stable model). Let $M \in \mathcal{I}^H(\mathcal{C})$. We say that M is a *provenance stable model* of $\mathcal{C}$ iff there is a chain of provenance Herbrand interpretations of $\mathcal{C}$, $I_0 \leq ... \leq I_{m+1}$ s.t. $I_m = I_{m+1} = M$ and[18]:

1. $I_0 = \mathit{minimal}(\{I \in \mathcal{I}^H(\mathcal{C}) \mid I \models [\neg]H(s,p,o,\{\mathit{Nam}_O\})$, for each $[\neg]p(s,o)$ appearing in the skolemization of a g-RDF graph G_O in $\mathcal{C}\})$.
2. For natural numbers α with $0 < \alpha \leq m+1$:
$I_\alpha = \mathit{minimal}(\{I \in \mathcal{I}^H(\mathcal{C}) \mid I \geq I_{\alpha-1}$ and for all $q_0 \leftarrow q_1, ..., q_l, {\sim}q_{l+1}, ..., {\sim}q_k \in [\mathit{prov}(P)]_\mathcal{C}$, for P be a g-RDF program appearing in $\mathcal{C}$, it holds that if (i) $I_{\alpha-1} \models q_i$, for $i = 1, ..., l$ and (ii) $M \models {\sim}q_i$, for $i = l+1, ..., k$ then $I \models q_0\})$.

We denote the provenance stable models of $\mathcal{C}$ by $\mathcal{M}(\mathcal{C})$. □

Note that I_0 is a minimal provenance Herbrand interpretation of $\mathcal{C}$ that satisfies $[\neg]H(s,p,o,\{\mathit{Nam}_O\})$, for each $[\neg]p(s,o)$ appearing in the skolemization of a g-RDF graph G_O in $\mathcal{C}$, while Herbrand interpretations $I_1, ..., I_{m+1}$ correspond to a stratified sequence of rule applications, where all applied rules are also applicable by M. In other words, a stable model is generated bottom-up by the iterative application of the rules in $[\mathit{prov}(P)]_\mathcal{C}$, for a g-RDF program P appearing in $\mathcal{C}$, starting from the information in the g-RDF graphs in $\mathcal{C}$. Thus, provenance stable model semantics captures the intuition that provenance quads $H(s,p,o,S)$ and $\neg H(s,p,o,S)$ should only be accepted if $\mathcal{C}$ contains some direct support for them in the form of an acceptable rule sequence.

Note that Definition 18 is purely model-theoretic, based on $\mathcal{I}^H(\mathcal{C})$. In the following section, we will show that satisfiability and entailment under the provenance stable model semantics are in general undecidable.

Below, we define entailment of a provenance quad, weak negation of a provenance quad, or provenance graph w.r.t. a g-RDF information base. Entailment of a provenance graph by a g-RDF information base under the provenance stable model semantics is defined in order to stay compatible with the RDFS semantics that define RDFS-entailment between RDF graphs.

Definition 19 (Provenance entailment of a provenance quad or provenance graph). Let $\mathcal{C}$ be a g-RDF information base and let G be a provenance quad or the weak negation of a provenance quad, or a provenance graph. We say that $\mathcal{C}$ entails G under the provenance stable model semantics, denoted by $\mathcal{C} \models^{pr} G$, if for all $M \in \mathcal{M}(\mathcal{C})$, it holds that $M \models G$. □

Note that it is not possible for an $M \in \mathcal{M}(\mathcal{C})$ to satisfy both $M \models H(s,p,o,S)$ and $M \models \neg H(s,p,o,S)$, due to the coherence condition imposed to all provenance Herbrand interpretations of $\mathcal{C}$. However, it is possible for an $M \in \mathcal{M}(\mathcal{C})$ to satisfy both $M \models H(s,p,o,S)$ and $M \models \neg H(s,p,o,S')$, as long as there is $c \in S$ and $c' \in S'$ s.t. $\mathit{conflict}(c,c')$ holds.

Example 9. Consider the g-RDF information base $\mathcal{C}$ of Figure 1. Then, $\mathcal{C}$ has a unique provenance stable model M s.t. $M \models {\sim}H($`ex:hairdresser`, `rdf:type`, `ex:Heavy_Job`, $\{O_4\})$, generated by the sequence $I_0 \leq I_1 \leq M$. In Figure 3, we show a few provenance

[18] Minimality is taken w.r.t. the relation $\leq$. The condition $I_m = I_{m+1} = M$ actually states that two successive iterations, one computing I_m and the next computing I_{m+1} result to the same provenance Herbrand interpretation of $\mathcal{C}$, which is equal to M.

quads satisfied by the provenance Herbrand interpretations I_0, I_1, and M. In particular, if $I_i \models H(s,p,o,S')$ in Figure 3 then $I_i \models H(s,p,o,S)$, for each $S \supseteq S'$ s.t. *compatible*(S) holds. Similar is the case for M.

Provenance Herbrand interpretation I_0

$I_0 \models H(s,p,o,\{Nam_{O_1}\})$, for all $p(s,o) \in G_{O_1}$
$I_0 \models H(s,p,o,\{Nam_{O_2}\})$, for all $p(s,o) \in G_{O_2}$
$I_0 \models H(s,p,o,\{Nam_{O_3}\})$, for all $p(s,o) \in G_{O_3}$
$I_0 \models H(s,p,o,\{Nam_{O_4}\})$, for all $p(s,o) \in G_{O_4}$
$I_0 \models H(\texttt{ex:Mary}, \texttt{rdf:type}, \texttt{ex:Adult}, \{Nam_{O_2}\})$
$I_0 \models H(\texttt{ex:hairdresser}, \texttt{rdf:type}, \texttt{ex:Job}, \{Nam_{O_2}\})$
$I_0 \models H(\texttt{ex:garbage\_collector}, \texttt{rdf:type}, \texttt{ex:Job}, \{Nam_{O_2}\})$
$I_0 \models H(\texttt{ex:Mary}, \texttt{rdf:type}, \texttt{ex:Person}, \{Nam_{O_1}, Nam_{O_2}\})$

Provenance Herbrand interpretation $I_1 \geq I_0$

$I_1 \models H(\texttt{ex:Mary}, \texttt{ex:gets\_pension}, "55"\hat{}\hat{}xsd{:}integer, \{Nam_{O_2}, Nam_{O_3}, Nam_{O_5}\}$
$I_1 \models H(\texttt{ex:Mary}, \texttt{ex:gets\_pension}, "65"\hat{}\hat{}xsd{:}integer, \{Nam_{O_2}, Nam_{O_6}\}$
$I_1 \models H(x, \texttt{owl:sameAs}, x, \{Nam_{O_7}\})$, for all $x \in V_{\mathcal{C}}$
$I_1 \models H(\texttt{ex:pension\_age}, \texttt{owl:sameAs}, \texttt{ex:gets\_pension}, \{Nam_{O_1}, Nam_{O_7}\})$

Provenance Herbrand interpretation $M \geq I_1$

$M \models H(\texttt{ex:Mary}, \texttt{ex:pension\_age}, "55"\hat{}\hat{}xsd{:}integer, \{Nam_{O_2}, Nam_{O_3}, Nam_{O_5}, Nam_{O_7}\}$
$M \models H(\texttt{ex:Mary}, \texttt{ex:pension\_age}, "65"\hat{}\hat{}xsd{:}integer, \{Nam_{O_2}, Nam_{O_6}, Nam_{O_7}\}$

Fig. 3. g-RDF triples satisfied by the provenance Herbrand interpretations I_0, I_1, and M

Thus, $\mathcal{C} \models^{pr} H(\texttt{ex:Mary}, \texttt{rdf:type}, \texttt{ex:Person}, \{Nam_{O_1}, Nam_{O_2}\})$, based on the first g-RDF triple of O_1, the first and third g-RDF triple of O_2, and the fifth provenance RDFS condition (Definition 11). Additionally, $\mathcal{C} \models^{pr} H(\texttt{ex:Mary}, \texttt{ex:gets\_pension}$, $"55"\hat{}\hat{}xsd{:}integer$, $\{Nam_{O_2}, Nam_{O_3}, Nam_{O_5}\})$, based on the third g-RDF triple of O_2, the first rule of O_3, and the first rule of O_5. Further, $\mathcal{C} \models^{pr} H(\texttt{ex:Mary}, \texttt{ex:gets\_pension}$, $"65"\hat{}\hat{}xsd{:}integer$, $\{Nam_{O_2}, Nam_{O_6}\})$, based on the third g-RDF triple of O_2, the fact that $\texttt{rdf:type}(\texttt{ex:hairdresser}, \texttt{ex:Heavy\_Job}) \notin G_{O_4}$, and the second rule of O_6. Note that though the RDF triples $\texttt{ex:gets\_pension}(\texttt{ex:Mary}, "55"\hat{}\hat{}xsd{:}integer)$ and $\texttt{ex:gets\_pension}(\texttt{ex:Mary}, "65"\hat{}\hat{}xsd{:}integer)$ are conflicting, this is due to the different temporal status of the results. The RDF triple $\texttt{ex:gets\_pension}(\texttt{ex:Mary}, "55"\hat{}\hat{}xsd{:}integer)$ holds for the years 2000 to 2009, while the RDF triple $\texttt{ex:gets\_pension}(\texttt{ex:Mary}, "65"\hat{}\hat{}xsd{:}integer)$ holds for the years 2010 to 2013. This information can be derived by checking the metadata regarding the temporal status of the g-RDF ontologies in $\mathcal{C}$. Note that it does not hold $\mathcal{C} \models^{pr} H(\texttt{ex:Mary}, \texttt{ex:gets\_pension}, "62"\hat{}\hat{}xsd{:}integer$, $\{Nam_{O_2}, Nam_{O_3}, Nam_{O_6}\})$, since the g-RDF ontologies O_3 and O_6 are conflicting. $\square$

The following proposition shows that our provenance stable model semantics extends RDFS semantics.

Proposition 1. Let $\mathcal{C} = \langle \{\langle c, O\rangle\}, \textit{conflict}\rangle$ be a g-RDF information base s.t. $O = \langle G, \emptyset\rangle$ is a g-RDF ontology with G being an RDF graph and $\textit{conflict} = \{\}$. Additionally, let $G' = \{p_1(s_1, o_1), ..., p_m(s_m, o_m)\}$ be an RDF graph. Then,
$\mathcal{C} \models^{pr} \{H(s_1, p_1, o_1, \{c\}), ..., H(s_m, p_m, o_m, \{c\})\}$ iff $G \models^{RDFS} G'$. $\square$

5 #*n*-Provenance Stable Model Semantics

In this section, we show that satisfiability and entailment under the provenance stable model semantics are in general undecidable. Therefore, we define the $\#n$-provenance stable model semantics of a g-RDF information base $\mathcal{C}$ which is decidable and also faithfully extends the RDFS semantics.

The proof of undecidability exploits a relationship between the stable model semantics of a simple ERDF ontology [5] and the provenance stable model semantics of a g-RDF information base with a single g-RDF ontology. Intuitively, an ERDF ontology is the combination of (i) an ERDF graph G containing (implicitly existentially quantified) positive and negative information, in the form of ERDF triples[19] and (ii) an ERDF program P containing derivation rules, with possibly all connectives $\sim$, $\neg$, $\supset$, $\wedge$, $\vee$, $\forall$, $\exists$ in the body of a rule, and strong negation $\neg$ in the head of a rule. A *simple ERDF ontology* is an ERDF ontology whose program rules contain only the connectives $\sim$, $\neg$, and $\wedge$. In addition to the RDF and RDFS vocabulary terms, the stable model semantics of ERDF ontologies support (i) the vocabulary term `erdf:TotalPropery` which is a class whose instances are properties p that satisfy the *Open-World Assumption (OWA)* and (ii) the vocabulary term `erdf:TotalClass` which is a class whose instances are classes that satisfy the *Open-World Assumption (OWA)*. Note that the ERDF theory does not allow variables in property positions of ERDF triples. The stable model semantics of an ERDF ontology are defined in [5], showing that even in the case of simple ERDF ontologies, satisfiability and entailment are, in general, undecidable.

In particular, the following proposition holds:

Proposition 2. Let $O = \langle G, P\rangle$ be a simple ERDF ontology and $\mathcal{C} = \langle\{(c, O')\}, \emptyset\rangle$ be a g-RDF information base s.t. $O' = \langle G', P'\rangle$, where G' and P' are defined as follows:

$$\begin{aligned} G' = G \cup \{&\texttt{rdfs:subClassOf}(\texttt{erdf:TotalProperty}, \texttt{rdfs:Class}),\\ &\texttt{rdfs:subClassOf}(\texttt{erdf:TotalClass}, \texttt{rdfs:Class})\}\end{aligned}$$

$$\begin{aligned} P' = P \cup \{&\neg ?p(?s, ?o) \leftarrow \texttt{rdf:type}(?p, \texttt{erdf:TotalProperty}), \sim ?p(?s, ?o).\\ &?p(?s, ?o) \leftarrow \texttt{rdf:type}(?p, \texttt{erdf:TotalProperty}), \sim\neg ?p(?s, ?o).\\ &\neg\texttt{rdf:type}(?o, ?c) \leftarrow \texttt{rdf:type}(?c, \texttt{erdf:TotalClass}), \sim\texttt{rdf:type}(?o, ?c).\\ &\texttt{rdf:type}(?o, ?c) \leftarrow \texttt{rdf:type}(?c, \texttt{erdf:TotalClass}), \sim\neg\texttt{rdf:type}(?o, ?c).\}\end{aligned}$$

Let $p(s, o)$ be a ground g-RDF triple over $V_{\mathcal{C}}$. Then, it holds that $O \models^{st} [\sim][\neg]p(s, o)$ iff $\mathcal{C} \models^{pr} [\sim][\neg]H(s, p, o, \{c\})$, where $\models^{st}$ is entailment from an ERDF ontology under the stable model semantics[20]. $\square$

Based on the fact the stable model semantics on simple ERDF ontologies is undecidable, it follows that the provenance stable model semantics on g-RDF information bases is also undecidable. The source of undecidability of the provenance stable model semantics of $\mathcal{C}$ is the fact that $\mathcal{V}_{RDF}$ is infinite, because of the container membership properties. Thus, the vocabulary of $\mathcal{C}$ is also infinite (note that $\{rdf{:}\_i \mid i \geq 1\} \subseteq \mathcal{V}_{RDF} \subseteq V_{\mathcal{C}}$). Consequently, in this section, we slightly modify the definition of the provenance stable model semantics, based on a redefinition of the vocabulary of a g-RDF information base, which now becomes finite by limiting the maximum number of container membership properties. We call the modified semantics, the *#n-provenance stable model semantics* (for $n \in I\!N$).

In order to define the $\#n$-provenance stable model semantics, we need to modify several of the definitions on which the provenance stable model semantics is based. Actually, what we do is that we remove from the vocabularies the infinite in number terms $\{rdf{:}\_i \in \mathcal{V}_{RDF} \mid i > n\}$. Specifically:

[19] An *ERDF triple* is a g-RDF triple with no variables allowed in the property position.

[20] Note that this is a compact form of expressing that (i) $O \models^{st} p(s, o)$ iff $\mathcal{C} \models^{st} p(s, o)$, (ii) $O \models^{st} \neg p(s, o)$ iff $\mathcal{C} \models^{st} \neg p(s, o)$, (iii) $O \models^{st} \sim p(s, o)$ iff $\mathcal{C} \models^{st} \sim p(s, o)$, and (iv) $O \models^{st} \sim\neg p(s, o)$ iff $\mathcal{C} \models^{st} \sim\neg p(s, o)$.

– Let $\mathcal{V}_{RDF}^{\#n} = \mathcal{V}_{RDF} - \{rdf{:}\_i \in \mathcal{V}_{RDF} \mid i > n\}$.
– A *#n-provenance RDFS interpretation* is defined as a provenance RDFS interpretation in Definition 11, where $\mathcal{V}_{RDF}$ is replaced by $\mathcal{V}_{RDF}^{\#n}$.
– We define: $V_{\mathcal{C}}^{\#n} = V_{\mathcal{C}} - \{rdf{:}\_i \in \mathcal{V}_{RDF} \mid i > n\}$, and $Res_{\mathcal{C}}^{H\#n} = Res_{\mathcal{C}}^{H} - \{rdf{:}\_i \in \mathcal{V}_{RDF} \mid i > n\}$.
– We define a *#n-provenance Herbrand interpretation* as a provenance Herbrand interpretation in Definition 14, where we replace $Res_{\mathcal{C}}^{H}$ by $Res_{\mathcal{C}}^{H\#n}$ and $V_{\mathcal{C}}$ by $V_{\mathcal{C}}^{\#n}$. We denote the set of #n-provenance Herbrand interpretations of $\mathcal{C}$ by $\mathcal{I}^{H_{\#n}}(\mathcal{C})$.
– We define $[r]_{\mathcal{C}}^{\#n}$ as $[r]_{\mathcal{C}}$ in Definition 16, where we replace $V_{\mathcal{C}}$ by $V_{\mathcal{C}}^{\#n}$. Let P be the g-RDF program of a g-RDF ontology O. We define $[prov(P)]_{\mathcal{C}}^{\#n} = \bigcup_{r \in P} [prov(r)]_{\mathcal{C}}^{\#n}$.
– We define a *#n-provenance stable model* of $\mathcal{C}$ as a provenance stable model of $\mathcal{C}$ in Definition 18, where we replace $\mathcal{I}^{H}(\mathcal{C})$ by $\mathcal{I}^{H_{\#n}}(\mathcal{C})$ and $[prov(P)]_{\mathcal{C}}$ by $[prov(P)]_{\mathcal{C}}^{\#n}$. We denote the #n-provenance stable models of $\mathcal{C}$ by $\mathcal{M}^{\#n}(\mathcal{C})$.

Note that the definition of a #n-provenance stable model of $\mathcal{C}$ is purely-model theoretic and it depends on the definition of $\mathcal{I}^{H_{\#n}}(\mathcal{C})$. Though in the next Section, we show that it can implemented through Answer Set Programming [32], there are other possible implementations of this semantics such as those provided in the proofs of Propositions 9(1), 10(1), and 11(1).

Definition 20 (#n-Provenance entailment of a provenance quad and provenance graph). Let $\mathcal{C}$ be a g-RDF information base and let G be a provenance quad or the weak negation of a provenance quad, or a provenance graph. We say that $\mathcal{C}$ entails G under the #n-provenance semantics, denoted by $\mathcal{C} \models^{pr_{\#n}} G$, if for all $M \in \mathcal{M}^{\#n}(\mathcal{C})$, it holds that $M \models G$. □

The following proposition states that provenance entailment of a provenance graph coincides with #n-provenance entailment, provided that the queried g-RDF ontology $\mathcal{C}$ does not contain weak negation and n is large enough to include all *rdf:_i* terms appearing in $\mathcal{C}$ and the query. The concrete value n is determined resorting to the following auxiliary definition.

$$n_{\mathcal{C}} = \begin{cases} 1, & \text{if no } rdf{:}\_i \text{ term appears in } \mathcal{C} \\ max(\{i \in I\!N \mid rdf{:}\_i \text{ appears in } \mathcal{C}\}), & \text{otherwise} \end{cases}$$

For example, if $\mathcal{C}$ is the g-RDF information base of Example 1 then $n_{\mathcal{C}} = 1$.

Proposition 3. Let $\mathcal{C}$ be a g-RDF information base that does not contain $\sim$ and let $n \geq n_{\mathcal{C}}$. Let G be a provenance graph s.t. $max(\{i \in I\!N \mid rdf{:}\_i \in V_G\}) \leq n$. It holds that: $\mathcal{C} \models^{pr} G$ iff $\mathcal{C} \models^{pr_{\#n}} G$. □

The following proposition follows directly from Proposition 1 and Proposition 3, showing that #n-provenance semantics not only faithfully extends the RDFS model theory but also faithfully extends RDFS entailment.

Proposition 4. Let $\mathcal{C} = \langle \{\langle c, O\rangle\}, \mathit{conflict}\rangle$ be a g-RDF information base s.t. $O = \langle G, \emptyset\rangle$ is a g-RDF ontology, where G is an RDF graph and $\mathit{conflict} = \{\}$. Let $n \geq n_{\mathcal{C}}$. Additionally, let $G' = \{p_1(s_1, o_1), ..., p_m(s_m, o_m)\}$ be an RDF graph s.t. $max(\{i \in I\!N \mid rdf{:}\_i \in V_{G'}\}) \leq n$. Then, $\mathcal{C} \models^{pr_{\#n}} \{H(s_1, p_1, o_1, \{c\}), ..., H(s_m, p_m, o_m, \{c\})\}$ iff $G \models^{RDFS} G'$. □

Faithfully extending the RDFS model theory is of particular importance because it shows how the two model theories are related and what are the extensions.

6 Computing the Derived Provenance Quads and Answers to Provenance Queries

Up to now, we have defined g-RDF information bases $\mathcal{C}$ and their provenance stable model semantics, showing that satisfiability and entailment under this semantics are, in general, undecidable. Therefore, by removing the infinite in number $rdf\!:\_i$ terms, where $i > n$ and keeping the rest of the definitions the same, we defined the $\#n$-provenance stable model semantics, which are decidable. Both semantics extend the RDFS semantics with strong negation, scoped weak negation, derivation rules, and why-provenance information.

In this section, we define the provenance queries over a g-RDF information base $\mathcal{C}$ and their answers according to $\#n$-provenance stable model semantics. Additionally, we present an algorithm based on Answer Set Programming [32] for computing all provenance quads that are entailed from $\mathcal{C}$.

Intuitively, a simple provenance query returns, through its variables, the g-RDF triples that are derived (or cannot be derived) (i) from a specified set of sources, (ii) from a superset of a specified set of sources, or (iii) from a subset of a specified set of sources. Provenance information is also returned. The combination of simple provenance queries, where variables can be shared, result to conjunctive provenance queries.

Formally, *simple provenance query SQ* of a g-RDF information base $\mathcal{C}$ is a formula of the form $\langle[\sim][\neg]H(s,p,o,?S),L^{\subseteq}\rangle$ or $\langle[\sim][\neg]H(s,p,o,?S),L^{\supseteq}\rangle$ or $\langle[\sim][\neg]H(s,p,o,?S),L^{=}\rangle$, where $s,o \in \mathit{URI} \cup \mathcal{LIT} \cup \mathit{Var}$, $p \in \mathit{URI} \cup \mathit{Var}$, and $L \subseteq \mathit{names}_{\mathcal{C}}$.

Let $SQ = \langle[\sim][\neg]H(s,p,o,?S),L^{R}\rangle$, for $R \in \{\subseteq,\supseteq,=\}$, be a simple provenance query over $\mathcal{C}$. We define the *answers* of SQ over $\mathcal{C}$ w.r.t. the $\#n$-provenance stable model semantics, denoted by $\mathit{Answers}^{\#n}_{\mathcal{C}}(SQ)$, as the set of mappings v (i) from the variables in s,p,o to $V^{\#n}_{\mathcal{C}}$ and (ii) from $?S$ to $\mathcal{P}(\mathit{names}_{\mathcal{C}})$ s.t. $\mathcal{C} \models^{pr\#n} [\sim][\neg]v(H(s,p,o,?S))$ and $L\ R\ v(?S)$.

Obviously, to derive the set of positive g-RDF triples, and their accompanied sets of provenance names, entailed by $\mathcal{C}$, we pose to $\mathcal{C}$ the query $SQ = \langle H(?s,?p,?o,?S),\emptyset^{\subseteq}\rangle$. Additionally, to derive the sets of provenance names of a g-RDF triple $p(s,o)$ that are (possibly) entailed by $\mathcal{C}$, we pose to $\mathcal{C}$ the query $SQ = \langle H(s,p,o,?S),\emptyset^{\subseteq}\rangle$.

A *conjunctive provenance query CQ* of $\mathcal{C}$ has the form $SQ_1 \wedge ... \wedge SQ_k$, where each $SQ_i = \langle[\sim][\neg]H(s_i,p_i,o_i,?S_i),L_i^{R_i}\rangle$ is a simple provenance query of $\mathcal{C}$[21]. Now, we define the *answers* of CQ over $\mathcal{C}$ w.r.t. the $\#n$-provenance stable model semantics, denoted by $\mathit{Answers}^{\#n}_{\mathcal{C}}(CQ)$, as the set of mappings v (i) from the variables in s_i,p_i,o_i to $V^{\#n}_{\mathcal{C}}$ and (ii) from $?S_i$ to $\mathcal{P}(\mathit{names}_{\mathcal{C}})$ s.t. $\mathcal{C} \models^{pr\#n} [\sim][\neg]v(H(s_i,p_i,o_i,?S_i))$ and $L_i\ R_i\ v(?S_i)$, for $i = 1,...,k$.

Example 10. Consider the g-RDF information base $\mathcal{C}$ of Figure 1. Then,

$$\begin{aligned}CQ = {} & \langle H(\texttt{ex:Mary},\texttt{ex:gets\_pension},?x,?S),\{\mathit{Nam}_{O_2}\}^{\subseteq})\rangle \wedge \\ & \langle H(\texttt{ex:Mary},\texttt{ex:gets\_pension},?x,?S),\{\mathit{Nam}_{O_1},\mathit{Nam}_{O_2},\mathit{Nam}_{O_3},\mathit{Nam}_{O_5}\}^{\supseteq})\rangle\end{aligned}$$

is a conjunctive provenance query of $\mathcal{C}$, requesting the age that Mary gets pension, as derived from a set of g-RDF ontologies that is superset of $\{\mathit{Nam}_{O_2}\}$ and subset of $\{\mathit{Nam}_{O_1},\ \mathit{Nam}_{O_2},\ \mathit{Nam}_{O_3},\mathit{Nam}_{O_5}\}$. Then, $\mathit{Answers}^{\#1}_{\mathcal{C}}(CQ)$ is (i) the mapping v s.t. $v(?x) =$ "55"^^*xsd:integer* and $v(?S) = \{\mathit{Nam}_{O_2},\ \mathit{Nam}_{O_3},\mathit{Nam}_{O_5}\}$ and (ii) the mapping u s.t. $u(?x) =$ "55"^^*xsd:integer* and $u(?S) = \{\mathit{Nam}_{O_1},\mathit{Nam}_{O_2},\ \mathit{Nam}_{O_3},\mathit{Nam}_{O_5}\}$. □

Example 11. Consider the g-RDF information base $\mathcal{C}$ of Figure 1. Then,

$$\begin{aligned}CQ = {} & \langle H(\texttt{ex:Mary},\ \texttt{rdf:type},?x,?S_1),\{\}^{\subseteq})\rangle \wedge \\ & \langle H(\texttt{ex:Mary},\texttt{ex:gets\_pension},?y,?S_2),\{\mathit{Nam}_{O_1},\mathit{Nam}_{O_2},\ \mathit{Nam}_{O_3},\mathit{Nam}_{O_5}\}^{\supseteq})\rangle\end{aligned}$$

[21] Provenance variables $?S_i$ can be shared.

is a conjunctive provenance query of $\mathcal{C}$, requesting the type of Mary, as derived from a set of g-RDF ontologies that is superset of $\{\}$, and the age that Mary gets pension, as derived from a set of g-RDF ontologies that is subset of $\{Nam_{O_1},\ Nam_{O_2},\ Nam_{O_3}, Nam_{O_5}\}$. Then, $Answers^{\#1}_{\mathcal{C}}(CQ)$ includes the mapping v s.t. $v(?x) = \texttt{ex:Person}, v(?y) =$ "55"^^ $xsd{:}integer$, $v(?S_1) = \{Nam_{O_1}\}$, and $v(?S_2) = \{Nam_{O_2},\ Nam_{O_3}, Nam_{O_5}\}$. $\square$

Before we provide an algorithm for computing all provenance quads that are entailed from $\mathcal{C}$ according to $\#n$-provenance stable model semantics, we provide a few definitions defining the logic program $\Pi^{\#n}_{\mathcal{C}}$. We consider a g-RDF information base:

$$\mathcal{C} = \langle\{\langle Nam_{O_i}, O_i\rangle \mid i = 1, ..., m\}, \mathit{conflict}\rangle,$$

where each $O_i = \langle G_i, P_i\rangle$ is a g-RDF ontology.

We define:

$$\Pi^{\mathtt{G}}_{\mathcal{C}} = \{H(s, p, o, \{Nam_{O_i}\}) \leftarrow .\ \mid p(s, o) \in sk(G_i) \text{ and } i = 1, ..., m\}.$$

Additionally, we define:

$$\Pi^{\mathtt{P}}_{\mathcal{C}} = \bigcup\{prov(P_i) \mid i = 1, ..., m\}$$

Further, we denote by $\Pi^{\mathtt{H}\#n}_{\mathcal{C}}$ the following set of provenance rules[22]:

Provenance Interpretation Rules

For all $x \in V_{Nam_{O_i}} \cup V^{\#n}_{RDF} \cup V_{RDFS}$:

(1) $H(x, \texttt{type}, \texttt{Resource}, \{Nam_{O_i}\}) \leftarrow.$

For all $x \in V_{Nam_{O_i}} \cap \mathcal{PL}$:

(2) $H(x, \texttt{type}, \texttt{Literal}, \{Nam_{O_i}\}) \leftarrow.$

(3) $H(?z, \texttt{type}, \texttt{Property}, ?S\}) \leftarrow H(?x, ?z, ?y, ?S).$

(4) $H(?z, \texttt{type}, ?y, ?S) \leftarrow H(?x, \texttt{domain}, ?y, ?S_1), H(?z, ?x, ?w, ?S_2), union(\{?S_1, ?S_2\}, ?S), compatible(?S).$

(5) $H(?w, \texttt{type}, ?y, ?S) \leftarrow H(?x, \texttt{range}, ?y, ?S_1), H(?z, ?x, ?w, ?S_2), union(\{?S_1, ?S_2\}, ?S). compatible(?S).$

(6) $H(?x, \texttt{subClassOf}, \texttt{Resource}, ?S) \leftarrow H(?x, \texttt{type}, \texttt{Class}, ?S).$

(7) $H(?z, \texttt{type}, ?y, ?S) \leftarrow H(?x, \texttt{subClassOf}, ?y, ?S_1), H(?z, \texttt{type}, ?x, ?S_2), union(\{?S_1, ?S_2\}, ?S), compatible(?S).$

(8) $\neg H(?z, \texttt{type}, ?x, ?S) \leftarrow H(?x, \texttt{subClassOf}, ?y, ?S_1), \neg H(?z, \texttt{type}, ?y, ?S_2), union(\{?S_1, ?S_2\}, ?S), compatible(?S).$

(9) $H(?x, \texttt{subClassOf}, ?x, ?S) \leftarrow H(?x, \texttt{type}, \texttt{Class}, ?S).$

(10) $H(?x, \texttt{subClassOf}, ?z, ?S) \leftarrow H(?x, \texttt{subClassOf}, ?y, ?S_1), H(?y, \texttt{subClassOf}, ?z, ?S_2), union(\{?S_1, ?S_2\}, ?S), compatible(?S).$

(11) $H(?z_1, ?y, ?z_2, ?S) \leftarrow H(?x, \texttt{subPropertyOf}, ?y, ?S_1), H(?z_1, ?x, ?z_2, ?S_2), union(\{?S_1, ?S_2\}, ?S), compatible(?S).$

(12) $\neg H(?z_1, ?x, ?z_2, ?S) \leftarrow H(?x, \texttt{subPropertyOf}, ?y, ?S_1), \neg H(?z_1, ?y, ?z_2, ?S_2), union(\{?S_1, ?S_2\}, ?S), compatible(?S).$

(13) $H(?x, \texttt{subPropertyOf}, ?x, ?S) \leftarrow H(?x, \texttt{type}, \texttt{Property}, ?S).$

(14) $H(?x, \texttt{subPropertyOf}, ?z, ?S) \leftarrow H(?x, \texttt{subPropertyOf}, ?y, ?S_1), H(?y, \texttt{subPropertyOf}, ?z, S_2),$

[22] For simplicity, we have eliminated the namespace from the URIs in $\mathcal{V}_{RDF} \cup \mathcal{V}_{RDFS}$.

$union(\{?S_1, ?S_2\}, ?S), compatible(?S).$

(15) $H(?x, \texttt{subClassOf}, \texttt{Literal}, ?S) \leftarrow H(?x, \texttt{type}, \texttt{Datatype}, ?S).$
(16) $H(?x, \texttt{subPropertyOf}, \texttt{member}, ?S) \leftarrow H(?x, \texttt{type}, \texttt{ContainerMembershipProperty}, ?S).$

For all $S, S' \subseteq names_{\mathcal{C}}$ and $S \subseteq S'$:
(17) $subset(S, S') \leftarrow .$

(18) $H(?s, ?p, ?o, ?S) \leftarrow H(?s, ?p, ?o, ?S'), subset(?S', ?S), compatible(?S).$
(19) $\neg H(?s, ?p, ?o, ?S) \leftarrow \neg H(?s, ?p, ?o, ?S'), subset(?S', ?S), compatible(?S).$

For each $"s"\hat{}\hat{}rdf{:}XMLLiteral \in V_{Nam_{O_i}}$ s.t. s is a well-typed XML literal:
(20) $H("s"\hat{}\hat{}rdf{:}XMLLiteral, \texttt{type}, \texttt{XMLLiteral}, \{Nam_{O_i}\}) \leftarrow.$

For each $"s"\hat{}\hat{}rdf{:}XMLLiteral \in V_{\mathcal{C}}^{\#n}$ s.t. s is an ill-typed XML literal:
(21) $false \leftarrow H("s"\hat{}\hat{}rdf{:}XMLLiteral, \texttt{type}, \texttt{Literal}, ?S).$

For each RDF and RDFS axiomatic triple $p(s, o)$ s.t. $p, s, o \in \mathcal{V}_{RDF}^{\#n} \cup \mathcal{V}_{RDFS}$:
(22) $H(s, p, o, \{\}) \leftarrow.$

Intuitively, the provenance interpretation rules of a g-RDF information base $\mathcal{C}$ represent the semantic conditions in Definition 4 (*Provenance Simple Interpretation*) and the semantic conditions in Definition 11 (*Provenance RDFS Interpretation*) that a $\#n$-provenance Herbrand interpretation of $\mathcal{C}$ (for $n \in I\!N$) satisfies. In particular, Rule (1) is derived from the provenance RDFS condition 2 of Definition 11 and the definition a $\#n$-provenance Herbrand interpretation of $\mathcal{C}$. Rule (2) is derived from the provenance RDFS condition 3 and the definition of a $\#n$-provenance Herbrand interpretation of $\mathcal{C}$. Additionally, Rule (21) is derived the provenance RDFS condition 20 and the third equality of the provenance RDFS condition 1. The rest of the provenance interpretation rules are derived immediately from corresponding semantic conditions in Definition 11 and the definition of $\#n$-provenance Herbrand interpretation of $\mathcal{C}$.

The intended logic program $\Pi_{\mathcal{C}}^{\#n}$ is obtained by translating the triples in the g-RDF graphs of $\mathcal{C}$, the g-RDF program rules of $\mathcal{C}$, and adding the previous provenance interpretation rules. In particular, we define:

$$\Pi_{\mathcal{C}}^{\#n} = \Pi_{\mathcal{C}}^{\texttt{G}} \cup \Pi_{\mathcal{C}}^{\texttt{P}} \cup \Pi_{\mathcal{C}}^{\texttt{H}\#n}.$$

To proceed in computing the $\#n$-provenance stable models of $\mathcal{C}$, we need the following auxiliary definitions. A *ground instance* of a rule $r \in \Pi_{\mathcal{C}}^{\#n}$ is derived from r by replacing all variables except $?S_i$ and $?S, ?S'$ by a term in $V_{\mathcal{C}}^{\#n}$ and and each variable $?S_i$ and $?S, ?S'$ by a subset of $names_{\mathcal{C}}$. A *ground provenance quad* of $\mathcal{C}$ is a provenance quad of $\mathcal{C}$, $[\neg]H(s, p, o, S)$, with no variables. We denote the set of ground provenance quads of $\mathcal{C}$ by $Q_{\mathcal{C}}$.

Let A be a set with 2^m constants. We define a one-to-one mapping $f_{\mathcal{C}}$ from $\mathcal{P}(names_C)$ to A. Let $[\neg]H(s, p, o, S) \in Q_{\mathcal{C}}$. We denote by $[\neg]f_{\mathcal{C}}(H(s, p, o, S))$ the extended logic programming (ELP) literal $[\neg]H(s, p, o, f_{\mathcal{C}}(S))$. Note that the mapping $f_{\mathcal{C}}$ is needed because ELP literals do not support sets in their arguments.

Below, we define the ground extended logic program $P_{\mathcal{C}}^{\#n}$ defined from $\Pi_{\mathcal{C}}^{\#n}$ as follows:

1. Generate all ground instances of the rules $r \in \Pi_{\mathcal{C}}^{\#n}$.
2. From the rules generated in Step 1, remove those that (i) contain in their bodies $union(SS, S), compatible(S)$ and (ii) S is not the union of the sets in SS or $compatible(S)$ does not hold.
3. From the rules generated in Step 2, remove those that (i) contain in their bodies $subset(S', S), compatible(S)$ and (ii) $S' \not\subseteq S$ or $compatible(S)$ does not hold.
4. In the remaining rules, replace each $[\neg]H(s, p, o, S) \in Q_{\mathcal{C}}$ by $[\neg]f_{\mathcal{C}}(H(s, p, o, S))$.

Intuitively, a $\#n$-provenance semi-Herbrand interpretation of $\mathcal{C}$ is a provenance simple interpretation of $V_{\mathcal{C}}$ and $\mathcal{C}$ that satisfies the three conditions of a $\#n$-provenance Herbrand interpretation, regarding the interpretation of the vocabulary, while the rest of the conditions are just the definitions of the ontological categories present in the definition of $\#n$-provenance RDFS interpretation. The definition of a $\#n$-provenance semi-Herbrand interpretation is needed for mapping $\#n$-provenance Herbrand interpretations to sets of ELP literals $[\neg]H(s,p,o,S) \in f_{\mathcal{C}}(Q_{\mathcal{C}})$, where $s,p,o \in V_{\mathcal{C}}^{\#n}$, and vice-versa.

Definition 21 ($\#n$-Provenance semi-Herbrand interpretation). A *$\#n$-provenance semi-Herbrand interpretation* I of $\mathcal{C}$ is a provenance simple interpretation of $V_{\mathcal{C}}^{\#n}$ and $\mathcal{C}$ without the condition $\mathcal{PL} \cap V_{\mathcal{C}}^{\#n} \subseteq LV_I$ such that:

1. $Res_I = Res_{\mathcal{C}}^{H_{\#n}}$.
2. $I_V(x) = x$, for all $x \in V_{\mathcal{C}}^{\#n} \cap URI$.
3. $IL_I(x) = x$, if x is a typed literal in $V_{\mathcal{C}}^{\#n}$ other than a well-typed XML literal, and $IL_I(x)$ is the XML value of x, if x is a well-typed XML literal in $V_{\mathcal{C}}^{\#n}$.

Additionally:
$Prop_I = \{x \mid \exists S \subseteq names_{\mathcal{C}} \text{ s.t. } \langle x, \texttt{rdf:Property}, S\rangle \in PT_I(\texttt{rdf:type})\}$.
$Res_I = \{x \mid \exists S \subseteq names_{\mathcal{C}} \text{ s.t. } \langle x, \texttt{rdfs:Resource}, S\rangle \in PT_I(\texttt{rdf:type})\}$.
$LV_I = \{x \mid \exists S \subseteq names_{\mathcal{C}} \text{ s.t. } \langle x, \texttt{rdfs:Literal}, S\rangle \in PT_I(\texttt{rdf:type})\}$. □

Let I be a $\#n$-provenance semi-Herbrand interpretation of $\mathcal{C}$. We define:

$$\begin{aligned} ELP(I) = \; & \{H(s,p,o,f_{\mathcal{C}}(S)) \mid s,p,o \in V_{\mathcal{C}}^{\#n} \text{ and } \langle I(s), I(o), S\rangle \in PT_I(p)\} \cup \\ & \{\neg H(s,p,o,f_{\mathcal{C}}(S)) \mid s,p,o \in V_{\mathcal{C}}^{\#n} \text{ and } \langle I(s), I(o), S\rangle \in PF_I(p)\} \end{aligned}$$

Note that the function $ELP(.)$ mapping a $\#n$-provenance semi-Herbrand interpretations of $\mathcal{C}$ to a set of ELP literals $[\neg]H(s,p,o,S) \in f_{\mathcal{C}}(Q_{\mathcal{C}})$, where $s,p,o \in V_{\mathcal{C}}^{\#n}$, is a bijective mapping.

Obviously, if M is a $\#n$-provenance Herbrand interpretation of $\mathcal{C}$ then M is a $\#n$-provenance semi-Herbrand interpretation of $\mathcal{C}$.

Proposition 5. Let $\mathcal{C}$ be a g-RDF information base. Let M be a $\#n$-provenance semi-Herbrand interpretation of $\mathcal{C}$. It is the case that: $M \in \mathcal{M}^{\#n}(\mathcal{C})$ iff $ELP(M)$ is a consistent answer set of $P_{\mathcal{C}}^{\#n}$. □

This result shows that the $\#n$-provenance stable model conditions can be captured by extended logic programming rules under answer set semantics [32]. Based on Proposition 5, we present the Algorithm 6.1 *All-$\#n$-ProvenanceStableModels*$(\mathcal{C}, n)$ that takes as input a g-RDF information base $\mathcal{C}$ and computes the set of all $\#n$-provenance stable models of $\mathcal{C}$.

Algorithm 6.1 *All-$\#n$-ProvenanceStableModels*$(\mathcal{C}, n)$

```
Input: (i) a g-RDF information base $\mathcal{C}$ and (ii) an $n \in \mathbb{N}$
Output: the set of all #n-provenance stable models of $\mathcal{C}$

(1)  $S = \{ELP^{-1}(N) \mid N$ is a consistent answer set of $P_{\mathcal{C}}^{\#n}\}$ ;
(2)  return($S$);
```

Using again Proposition 5, it follows directly that entailment of a ground provenance quad $Q \in Q_{\mathcal{C}}$ from $\mathcal{C}$ under the $\#n$-provenance stable model semantics can be achieved through Answer Set Programming (ASP) [32] on $P_{\mathcal{C}}^{\#n}$.

Proposition 6. Let $\mathcal{C}$ be a g-RDF information base and $Q \in Q_{\mathcal{C}}$. It holds that $\mathcal{C} \models^{pr_{\#n}} [\sim]Q$ iff $P_{\mathcal{C}}^{\#n} \models^{\mathtt{ASP}} [\sim]f_{\mathcal{C}}(Q)$. $\square$

Note that instead of the extended logic program $P_{\mathcal{C}}^{\#n}$, we could have used a Hilog program evaluated under answer set semantics [57], where each quad $H(s,p,o,f_{\mathcal{C}}(S))$ is replaced by $p(s,o,f_{\mathcal{C}}(S))$.

We now provide a few definitions. Let Π be an ELP. By $[\Pi]$, we denote the grounded version of Π and by $EHB(\Pi)$, we denote the *Extended Herbrand Base* of Π.[23] Let Π be a ground ELP. Let $r = L_0 \leftarrow L_1, ..., L_m, \sim L_{m+1}, ..., \sim L_n \in \Pi$. We define: $Head(r) = L_0$, $Body(r)^+ = \{L_1, ..., L_m\}$ and $Body(r)^- = \{L_{m+1}, ..., L_n\}$. Let $N \subseteq EHB(\Pi)$. We define: $\Pi^N = \{Head(r) \leftarrow Body(r)^+ \mid r \in \Pi \text{ and } Body(r)^- \cap N = \emptyset\}$.

Let Π be a ground ELP such that for any $r \in \Pi$, it holds $Body(r)^- = \emptyset$. We define the mapping $T_\Pi : \mathcal{P}(EHB(\Pi)) \to \mathcal{P}(EHB(\Pi))$, where[24] $T_\Pi(N) = N \cup \{Head(r) \mid r \in \Pi$ and $Body^+(r) \subseteq N\}$.

Based on Proposition 5, if M is a $\#n$-provenance semi-Herbrand interpretation of $\mathcal{C}$ then $M \in \mathcal{M}^{\#n}(\mathcal{C})$ iff $ELP(M) = T^{\uparrow\omega}_{(P_{\mathcal{C}}^{\#n})^{ELP(M)}}(\emptyset)$.

Let P be the set of rules of $P_{\mathcal{C}}^{\#n}$ generated by the rules (18) and (19) of $\Pi_{\mathcal{C}}^{\mathtt{H}\#n}$ and let P' be the rest of the rules of $P_{\mathcal{C}}^{\#n}$. It is easy to see that if M is a $\#n$-provenance semi-Herbrand interpretation of $\mathcal{C}$ then $M \in \mathcal{M}^{\#n}(\mathcal{C})$ iff $ELP(M) = T_P(T^{\uparrow\omega}_{P'^{ELP(M)}}(\emptyset))$. Thus, $M \in \mathcal{M}^{\#n}(\mathcal{C})$ iff for each $H(s,p,o,f_{\mathcal{C}}(S)) \in ELP(M)$, *compatible*$(S)$ holds and there exists $S' \subseteq S$ s.t. $H(s,p,o,f_{\mathcal{C}}(S')) \in T^{\uparrow\omega}_{P'^{ELP(M)}}$. This property allows for more efficient $\#n$-provenance stable model checking.

Before we provide the complexity of the $\#n$-provenance stable model semantics, we will show two reductions to the satisfiability problem of $\mathcal{C}$. The first reduction concerns the satisfiability problem[25] of a simple ERDF ontology under the $\#n$-stable model semantics[26] which in Proposition 13(1) of [3] is shown to be $\mathrm{NP}^{\mathrm{NP}}$-complete.

Proposition 7. Let $O = \langle G, P\rangle$ be a simple ERDF ontology and $\mathcal{C} = \langle\{(c, O')\}, \emptyset\rangle$ be a g-RDF information base s.t. $O' = \langle G', P'\rangle$, where G' and P' are as defined in Proposition 2. Let $p(s,o)$ be a ground g-RDF triple over $V_{\mathcal{C}}$. Then, it holds that $O \models^{st_{\#n}} [\sim][\neg]p(s,o)$ iff $\mathcal{C} \models^{pr} [\sim][\neg]p(s,o)$, where $\models^{st_{\#n}}$ is entailment from an ERDF ontology under the $\#n$-stable model semantics. $\square$

The proof of Proposition 7 is similar to the proof of Proposition 2.

Now, we will show a reduction from the 3-colorability problem[27]. Let $D = (V,E)$ be a graph. We say that D is *3-colorable*, if there exists a mapping *color* from the vertices V to colors $\{Red, Green, Blue\}$ such that if $(v,u) \in E$ then $color(v) \neq color(u)$. The

[23] The *Extended Herbrand Base* of an ELP Π is the set $\{p(c_1, ..., c_n) \mid p$ is a predicate appearing in Π of arity n and $c_1, ..., c_n$ are constants appearing in $\Pi\} \cup \{\neg p(c_1, ..., c_n) \mid p$ is a predicate appearing in Π of arity n and $c_1, ..., c_n$ are constants appearing in $\Pi\}$.

[24] $T_P(.)$ is the *immediate consequence operator* on a definite logic program P and is defined in [48]. $T_P^{\uparrow\omega}(\emptyset)$ denotes that the operator T_P is applied from the empty set until closure.

[25] In fact, we prove a stronger result.

[26] The $\#n$-stable model semantics of an ERDF ontology O are defined in [3], similarly to the stable model semantics of O by removing the infinite in number terms $\{rdf{:}\_i \in \mathcal{V}_{RDF} \mid i > n\}$.

[27] A similar reduction appears in [5] but we fully included the adapted reduction here because the terms `erdf:TotalProperty` and *false* used in [5] are not included in the present theory.

3-colorability problem is the following: given a graph D, decide if it is 3-colorable. From D, we will construct a g-RDF information base $\mathcal{C}_D$ such that D is *3-colorable* iff $\mathcal{C}_D$ has a $\#n$-provenance stable model, for $n \in I\!N$. The *3-colorability* problem is a classical NP-complete problem. Thus, based on this reduction, we will show that the satisfiability problem of a g-RDF information base, under the $\#n$-provenance stable model semantics is NP-hard w.r.t. the sizes of the g-RDF graphs in $\mathcal{C}$. We will construct a g-RDF ontology, denoted by $O_D = \langle G_D, P_D \rangle$, and based on this, we will construct the g-RDF information base $C_D = \langle \{\langle c, O_D \rangle\}, \mathit{conflict} \rangle$, where c is the name of O_D and $\mathit{conflict} = \emptyset$.

Consider (i) a property $\texttt{ex:edge}(v,u)$, indicating that there is an edge in D from vertex v to vertex u, (ii) a class $\texttt{ex:Red}$ whose instances are vertices of color *Red*, (iii) a class $\texttt{ex:Green}$ whose instances are vertices of color *Green*, (iv) a class $\texttt{ex:Blue}$ whose instances are vertices of color *Blue*, (v) a property $\texttt{ex:edgeR}(v,u)$, indicating that in edge $\langle v,u \rangle$, the color of v is *Red*, (vi) a property $\texttt{ex:edgeG}(v,u)$, indicating that in edge $\langle v,u \rangle$, the color of v is *Green*, (vii) a property $\texttt{ex:edgeB}(v,u)$, indicating that in edge $\langle v,u \rangle$, the color of v is *Blue*. Let $s, p, o \in \mathit{URI}$.

Let G_D be the g-RDF graph:

$$\begin{aligned} G_D = \ & \{\texttt{ex:edge}(v,u) \mid \langle v,u \rangle \in E\} \cup \\ & \{\texttt{rdfs:domain}(\texttt{ex:edgeR}, \texttt{ex:Red}), \texttt{rdfs:range}(\texttt{ex:edgeR}, \texttt{ex:NotRed}), \\ & \ \texttt{rdfs:domain}(\texttt{ex:edgeG}, \texttt{ex:Green}), \texttt{rdfs:range}(\texttt{ex:edgeG}, \texttt{ex:NotGreen}), \\ & \ \texttt{rdfs:domain}(\texttt{ex:edgeB}, \texttt{ex:Blue}), \texttt{rdfs:range}(\texttt{ex:edgeB}, \texttt{ex:NotBlue}), \\ & \ \texttt{rdfs:subClassOf}(\texttt{ex:Red}, \texttt{ex:NotGreen}), \texttt{rdfs:subClassOf}(\texttt{ex:Red}, \texttt{ex:NotBlue}), \\ & \ \texttt{rdfs:domain}(\texttt{ex:Blue}, \texttt{ex:NotGreen}), \texttt{rdfs:subClassOf}(\texttt{ex:Blue}, \texttt{ex:NotRed}), \\ & \ \texttt{rdfs:subClassOf}(\texttt{ex:Green}, \texttt{ex:NotRed}), \texttt{rdfs:subClassOf}(\texttt{ex:Green}, \texttt{ex:NotBlue}), \\ & \ p(s,o)\} \end{aligned}$$

Let P_D be the g-RDF program, containing the following rules:

$$\begin{aligned} \neg\texttt{rdf:type}(?x, \texttt{ex:edgeR}) & \leftarrow \sim\texttt{rdf:type}(?x, \texttt{ex:edgeR})@\{c\}. \\ \texttt{rdf:type}(?x, \texttt{ex:edgeR}) & \leftarrow \sim\neg\texttt{rdf:type}(?x, \texttt{ex:edgeR})@\{c\}. \\ \neg\texttt{rdf:type}(?x, \texttt{ex:edgeG}) & \leftarrow \sim\texttt{rdf:type}(?x, \texttt{ex:edgeG})@\{c\}. \\ \texttt{rdf:type}(?x, \texttt{ex:edgeG}) & \leftarrow \sim\neg\texttt{rdf:type}(?x, \texttt{ex:edgeG})@\{c\}. \\ \neg\texttt{rdf:type}(?x, \texttt{ex:edgeB}) & \leftarrow \sim\texttt{rdf:type}(?x, \texttt{ex:edgeB})@\{c\}. \\ \texttt{rdf:type}(?x, \texttt{ex:edgeB}) & \leftarrow \sim\neg\texttt{rdf:type}(?x, \texttt{ex:edgeB})@\{c\}. \end{aligned}$$

and the following rules (which are actually constraints since $p(s,o) \in G_D$);

$$\begin{aligned} \neg p(s,o) & \leftarrow \texttt{ex:edge}(?x,?y) \wedge \neg\texttt{ex:edgeR}(?x,?y) \wedge \neg\texttt{ex:edgeG}(?x,?y) \wedge \\ & \quad\ \neg\texttt{ex:edgeB}(x,y). \\ \neg p(s,o) & \leftarrow \texttt{rdf:type}(?x, \texttt{ex:Red}) \wedge \texttt{rdf:type}(?x, \texttt{ex:NotRed}). \\ \neg p(s,o) & \leftarrow \texttt{rdf:type}(?x, \texttt{ex:Green}) \wedge \texttt{rdf:type}(?x, \texttt{ex:NotGreen}). \\ \neg p(s,o) & \leftarrow \texttt{rdf:type}(?x, \texttt{ex:Blue}) \wedge \texttt{rdf:type}(?x, \texttt{ex:NotBlue}). \end{aligned}$$

Proposition 8. Let $D = (V, E)$ be a graph. D is *3-colorable* iff $\mathcal{C}_D$ has a $\#n$-provenance stable model, for $n \in I\!N$. $\square$

The proof of proposition 8 is similar to the proof of Proposition 8 in [3].

Below, we provide the following satisfiability complexity results.

Proposition 9. Let $\mathcal{C} = \langle \{\langle \mathit{Nam}_{O_i}, O_i \rangle \mid i = 1, ..., m\}, \mathit{conflict} \rangle$ be a g-RDF information base. Deciding if $\mathcal{C}$ has a $\#n$-provenance stable model is:

1. $\mathrm{NP}^{\mathrm{NP}}$-complete w.r.t. the size[28] of $\mathcal{C}$ and

[28] The size of $\mathcal{C}$ is $\mathit{size}(O_1) + ... + \mathit{size}(O_m) + m + \mathit{size}(\mathit{conflict})$.

2. NP-complete w.r.t. $graph\_size(\mathcal{C}) = size(G_1) + ... + size(G_m)$. □

Based on Proposition 9, we provide the following query answering complexity results.

Proposition 10. Let $\mathcal{C} = \langle \{\langle Nam_{O_i}, O_i\rangle \mid i = 1, ..., m\}, \mathit{conflict}\rangle$ be a g-RDF information base and let $CQ = SQ_1 \wedge ... \wedge SQ_k$, where each $SQ_i = \langle [\sim][\neg]H(s_i, p_i, o_i, ?S_i), L_i^{R_i}\rangle$ is a simple provenance query of $\mathcal{C}$. Let v be a mapping (i) from the variables in s_i, p_i, o_i to $V_{\mathcal{C}}^{\#n}$ and (ii) from $?S_i$ to $\mathcal{P}(names_{\mathcal{C}})$. The time complexity of checking if $v \in Answers_{\mathcal{C}}^{\#n}(CQ)$ is:

1. co-NP$^{\mathrm{NP}}$-complete w.r.t. the size of $\mathcal{C}$,
2. co-NP-complete w.r.t. $graph\_size(\mathcal{C}) = size(G_1) + ... + size(G_m)$. □

7 Positive *g*-RDF Information Bases

In this section, we consider g-RDF information bases without negation, called *positive g-RDF information bases* and provenance queries without negation, called *positive provenance queries*. This corresponds to the strictly monotonic fragment of the language and it is adequate for the users desiring that the knowledge extracted from g-RDF information bases will never be retracted by subsequent additions to the existing information bases. Obviously, there many applications and languages that do not use negation. For example, RDF [39] and RIF-CORE (the CORE dialect of the Rule Interchange Format) [7] do not use negation. Note that RIF-CORE corresponds to the language of definite Horn rules without function symbols. Additionally, [26, 63][29], though they are concerned with the provenance of RDF triples from a set of named RDF graphs, do not support neither rules nor negation. Of course, provenance of g-RDF triples from positive g-RDF rule bases are of great importance since they allow to annotated the conclusions with the intervening information bases.

Let $\mathcal{C}$ be a positive g-RDF information base, it follows from Proposition 5 that if $\mathcal{M}^{\#n}(\mathcal{C}) \neq \emptyset$ then $\mathcal{C}$ has a unique $\#n$-provenance stable model, denoted by $M_{\mathcal{C}}^{\#n}$, such that $M_{\mathcal{C}}^{\#n} = ELP^{-1}(T^{\uparrow\omega}_{P_{\mathcal{C}}^{\#n}}(\emptyset))$. Additionally, $\mathcal{M}^{\#n}(\mathcal{C}) = \emptyset$ iff $false \in T^{\uparrow\omega}_{P_{\mathcal{C}}^{\#n}}(\emptyset)$. Note that if $false \in T^{\uparrow\omega}_{P_{\mathcal{C}}^{\#n}}(\emptyset)$ then there is an ill-typed XML literal $x \in V_{\mathcal{C}}^{\#n}$, for which it holds that $x \in LV_M$, for $M = ELP^{-1}(T^{\uparrow\omega}_{P_{\mathcal{C}}^{\#n}}(\emptyset) - \{false\})$. Thus, an XML clash is caused and this is the only case that a positive g-RDF information base does not have a $\#n$-provenance stable model.

Query answering for positive g-RDF information bases can be readily implemented by resorting to a state-of-the-art XSB-PROLOG system [58]. The advantage of XSB is that it provides tabling mechanisms that support both minimal model semantics for definite logic programming as well as well-founded semantics for the case of weak negation, and no complex control is required by the programmer. Usually, the produced coded is very similar to the declarative theoretical specification providing additional guarantees of the correctness of the implementation (as the reader will see). Contrary to answer-set systems, the grounding of all rules may not be necessary beforehand and thus the size of the program can be substantially reduced and no preprocessing is required. Moreover, the evaluation mechanism is goal-oriented avoiding blow-up in the extracted conclusions. Finally, the tabling mechanisms ensures termination for mutually recursive queries, and has optimal complexity [60]; therefore no other techniques are necessary to perform transitive closure of the `rdfs:subClassOf` and `rdfs:subPropertyOf` relations that are essential to guarantee completeness with respect to the original RDFS semantics.

[29] These works are reviewed in the Related Work section.

Below, we define the XSB-PROLOG program $W_{\mathcal{C}}^{\#n}$, which can be used to answer positive conjunctive provenance queries, by resorting to the XSB system. First, the declaration `:- table 'H'/4.` is added to the beginning of program $W_{\mathcal{C}}^{\#n}$.

Next, each logic rule in $\Pi_{\mathcal{C}}^{\#n}$, except rules in lines (18) and (19), is added to $W_{\mathcal{C}}^{\#n}$, after applying the following translation:

– Substituting the preceding question mark "?" by an underscore "_". This is because variables in Prolog either start with capital letter or underscore.
– Substituting the leftarrow rule symbol "$\leftarrow$" in facts by a dot "`.`";
– Replacing the leftarrow rule symbol "$\leftarrow$" by "`:-`";
– Replacing the predicate symbol `H` by `'H'`;
– Replacing each set $\{?S_1, ?S_2\}$ by the list $[\_S_1, \_S_2]$ and each set $\{?S_1, ..., ?S_k, \{Nam_O\}\}$ by the list $[\_S_1, ..., \_S_k,\ [Nam_O]]$;
– Replacing each set of constants S by an ordered list of constants $[t_1, ..., t_k]$ s.t. $S = \{t_1, ..., t_k\}$.

Additionally, we add to $W_{\mathcal{C}}^{\#n}$, the following rules supporting set operations. We use the XSB-Prolog library `ordsets`, which represents sets as ordered lists of elements with no repetitions resulting in linear time set operations. In particular, the call `ord_union( SetOfSets, Union)` returns true if `Union` is the union of all elements in the set of sets `SetOfSets`. Each member of `SetOfSets` must be an ordered set, while the sets need not be ordered in any way. The call `ord_member(Element, Set)` returns true if `Element` is a member of the ordered set `Set`. The call `ord_subset(Set1, Set2)` returns true when every element of the ordered set `Set1` appears in the ordered set `Set2`. Finally, the call `ord_seteq(Set1, Set2)` returns true when the two ordered sets `Set1` and `Set2` are equal.

Required library predicates for ordered sets representation
```
:- import ord_union/2, ord_member/2, ord_subset/2, ord_seteq/2 from ordsets.
```

Deciding if a set of *g*-RDF ontologies is compatible
```
:- table compatible/1. % For efficiency of compatibility.
compatible(S) :- \+ incompatible(S).
incompatible(S) :- conflict(X,Y), ord_member(X,S), ord_member(Y,S).
```

Computing the union of a list of lists
```
union(SetOfSets,Result) :- ord_union(SetOfSets, Aux), sort(Aux,Result).
```

Let SQ_i be a simple provenance query of $\mathcal{C}$ of the form $\langle H(s,p,o,?S), \{a_1, ..., a_k\}^{\subseteq}\rangle$. Then, $Ans_{\mathcal{C}}^{\#n}(SQ_i)$ can be computed by adding to $W_{\mathcal{C}}^{\#n}$ the rule:

```
answeri(s,p,o,S) :- sort([a1,...,ak], A), 'H'(s,p,o,S'), subset(S',S)
                    compatible(S), ord_subset(A,S).
```

Let SQ_i be a simple provenance query of $\mathcal{C}$ of the form $\langle H(s,p,o,?S), \{a_1, ..., a_k\}^{\supseteq}\rangle$. Then, $Ans_{\mathcal{C}}^{\#n}(SQ_i)$ can be computed by adding to $W_{\mathcal{C}}^{\#n}$ the rule:

```
answeri(s,p,o,S) :- sort([a1,...,ak], A), 'H'(s,p,o,S'), subset(S',S)
                    compatible(S), ord_subset(S,A).
```

Let SQ_i be a simple provenance query of $\mathcal{C}$ of the form $\langle H(s,p,o,?S), \{a_1, ..., a_k\}^{=}\rangle$. Then, $Ans_{\mathcal{C}}^{\#n}(SQ_i)$ can be computed by adding to $W_{\mathcal{C}}^{\#n}$ the rule:

```
answeri(s,p,o,S) :- sort([a1,...,ak], A), compatible(A), 'H'(s,p,o,S'),
                    ord_subset(S',A), S=A.
```

If the list `[a`$_1$`,...,a`$_k$`]` is already ordered then the `sort` call in the body of the rules is redundant, and can be removed by substituting the variable `A` with the list.

Let CQ be a positive conjunctive provenance query of $\mathcal{C}$ of the form $SQ_1 \wedge ... \wedge SQ_k$, where each $SQ_i = \langle H(s_i, p_i,\ o_i, ?S_i), L^{R_i} \rangle$. Then, $Ans^{\#n}_{\mathcal{C}}(SQ_i)$ can be computed by adding the rule:

```
answer(s1,p1,o1,S1,...,sk,pk,ok,Sk) :-
      answer1(s1,p1,o1,S1),...,answerk(sk,pk,ok,Sk).
```

In [13], Chen and Warren proved that a query Q with a normal logic program P terminates under XSB's tabled evaluation if P has the *bounded-term-size property*. An even stronger requirement is that there exists an upper bound on the size of the arguments of all goals generated during query answering. If this requirement is satisfied, only a finite number of goals will be generated and thus, query evaluation terminates. This is the case with $W^{\#n}_{\mathcal{C}}$, since the largest size of each list appearing during the evaluation of a conjunctive provenance query is bounded by $l * m + 1$, where l is the maximum number of g-RDF triples appearing in the body of a rule in $\mathcal{C}$ (see the first argument of *union*(.) in Definition 15).

Below, we provide query answering complexity results for positive g-RDF information bases.

Proposition 11. Let $\mathcal{C} = \langle \{\langle Nam_{O_i}, O_i \rangle \mid i = 1, ..., m\}, \mathit{conflict} \rangle$ be a positive g-RDF information base and let $CQ = SQ_1 \wedge ... \wedge SQ_k$, where each $SQ_i = \langle H(s_i, p_i,\ o_i, ?S_i), L_i^{R_i} \rangle$ is a simple provenance query of $\mathcal{C}$. Let v be a mapping (i) from the variables in s_i, p_i, o_i to $V^{\#n}_{\mathcal{C}}$ and (ii) from $?S_i$ to $\mathcal{P}(names_{\mathcal{C}})$. The time complexity of checking if $v \in Answers^{\#n}_{\mathcal{C}}($ $CQ)$ is:

1. NP-complete w.r.t. the size of $\mathcal{C}$,
2. P-complete w.r.t. $graph\_size(\mathcal{C}) = size(G_1) + ... + size(G_m)$. $\square$

8 Discussion and Applications

The reader may argue that the simple example of Figure 1 can be easily broken into two parts: one containing the g-RDF ontologies O_1, O_2, O_3, O_5, and O_7 and the other containing the g-RDF ontologies O_1, O_2, O_4, O_6, and O_7. However, this way, why-provenance information of derived g-RDF triples is lost. For example, the g-RDF triple `rdf:type`(`ex:Mary`, `ex:Person`) is derived based on only the g-RDF ontologies O_1 and O_2. Additionally, the g-RDF triple `ex:gets_pension`(`ex:Mary`,"65"^^*xsd:integer*) is derived based only on the g-RDF ontologies O_2 and O_6. Further, consider a g-RDF information base $\mathcal{C}$ containing n g-RDF ontologies $O_1, ..., O_n$ such that a scoped weakly negated g-RDF triple $\sim p_i(s_i, o_i)$ @$\{Nam_{O_{i+1}}\}$ appears in each ontology $O_1, ..., O_{n-1}$, respectively, and $\sim p_n(s_n, o_n)$@ $\{Nam_{O_1}\}$ appears in O_n. Then, it is not possible to break $\mathcal{C}$ into sets of g-RDF ontologies. Even in the case of positive g-RDF information bases with a set of n conflict statements, in order to derive the same why-provenance results derived by our theory, one needs to consider an exponential (w.r.t. n) number of sets of g-RDF ontologies.

We now briefly review two applications of our framework, one from the biomedical and one from medical domain. In Figure 4, a g-RDF information base is displayed, where ontology O_1 contains the information that `ex:Mary` has breast cancer and that certain genes are expressed or not expressed in `ex:Mary`. Here, the use of explicit negation is important since the list of expressed genes cannot be complete. Ontology O_2 expresses the opinion of an expert on breast cancer that if a woman has breast cancer and gene1 is expressed on her, while gene2 is not expressed then her breast cancer is metastatic. Ontology O_3 expresses the opinion of an expert on breast cancer that if a woman has breast cancer, the status of the Estrogen Receptor (ER) marker is not positive according to O_1, and gene3 is expressed on her, while gene4 is not expressed then her breast cancer is not metastatic. Ontology O_4 contains the general information that if somebody has metastatic cancer then he/she should examine metastatic cancer markers, while the opposite holds if he/she does not have metastatic cancer. Additionally, $\mathcal{C}$ expresses that ontologies O_2 and O_3 are in conflict. Note that $\mathcal{C} \models^{pr\#_1}$ H(`ex:Mary, ex:should_do, ex:metastatic_exams`, $\{Nam_{O_1}, Nam_{O_2}, Nam_{O_4}\}$) and $\mathcal{C} \models^{pr\#_1} \neg H$(`ex:Mary, ex:should_do, ex:metastatic_exams`, $\{Nam_{O_1}, Nam_{O_3},$ $Nam_{O_4}\}$). Thus, from $\mathcal{C}$, contradictory information is derived and it is important to know the sources contributing to the derivation of each contradictory fact.

Ontology O_1

$\langle$*http://personal_biomedical_info.gr*$\rangle$

G_{O_1} =
```
rdf:type(ex:Mary,ex:Breast_Cancer_Patient).
has_gene_expressed(ex:Mary, ex:gene1).
¬ has_gene_expressed(ex:Mary, ex:gene2).
has_gene_expressed(ex:Mary, ex:gene3).
¬ has_gene_expressed(ex:Mary, ex:gene4).
```

Ontology O_2

$\langle$*http://breast_cancer_expert1.gr*$\rangle$

P_{O_2} =
```
rdf:type(?x,Metastatic_Cancer)← rdf:type(?x,ex:Breast_Cancer_Patient),
   has_gene_expressed(?x, ex:gene1), ¬ has_gene_expressed(?x, ex:gene2).
```

Ontology O_3

$\langle$*http://breast_cancer_expert2.gr*$\rangle$

P_{O_3} =
```
¬ rdf:type(?x,ex:Has_Metastatic_Cancer)← rdf:type(?x,ex:Breast_Cancer_Patient),
   ∼ rdf:type( ?x, ER_positive)@{Nam_O1}, has_gene_expressed( ?x, ex:gene3)
   ¬ has_gene_expressed( ?x, ex:gene4).
```

Ontology O_4

$\langle$*http://medical.gr*$\rangle$

P_{O_4} =
```
ex:should_do(?x, ex:metastatic_exams) ← rdf:type(?x,ex:Has_Metastatic_Cancer)
¬ ex:should_do(?x, ex:metastatic_exams) ← ¬ rdf:type(?x,ex:Has_Metastatic_Cancer)
```

Conflicts

`conflict`(*http://breast_cancer_expert1.gr*, *http://breast_cancer_expert1.gr*).

Fig. 4. A biomedical g-RDF information base

Consider now the g-RDF information base $\mathcal{C}$ of Figure 5. Ontology O_1 expresses that `ex:Manos` has multiple allergies, including allergies to dust mites, pollen, and dust. He is a 6 year old boy and he suffers from excessive eye blinking. Ontology O_2 expresses the opinion of a doctor that if a boy between 5 and 10 years old presents excessive eye blinking without having red eyes (according to ontology O_1) then he suffers from psychological eye tic. Ontology O_3 expresses that if somebody presents excessive eye blinking and has multiple allergies then he suffers from allergic conjunctivitis. Ontology O_4 expresses that if somebody suffers from psychological eye tic then he can play with flowers but the opposite holds if he suffers from allergic conjunctivitis. Additionally, $\mathcal{C}$ expresses that ontologies O_2 and O_3 are in conflict. Note that $\mathcal{C} \models^{pr\#3} H($`ex:Manos, ex:can_play_with, ex:flowers`$, \{Nam_{O_1}, Nam_{O_2}, Nam_{O_4}\})$ and $\mathcal{C} \models^{pr\#3} \neg H($`ex:Manos, ex:can_play_with, ex:flowers`$, \{Nam_{O_1}, Nam_{O_3}, Nam_{O_4}\})$. Thus, similarly to the previous example, from $\mathcal{C}$, contradictory information is derived and it is important to know the sources contributing to the derivation of each contradictory fact.

Ontology O_1

$\langle http://personal\_medical\_info.gr \rangle$

$G_{O_1} =$
```
rdf:type(ex:Manos,ex:Myltiple_Allergic).
has_allergies(ex:Manos, ex:Manos_allergies).
rdf:_1(ex:Manos_allergies, ex:dust_mites).
rdf:_2(ex:Manos_allergies, ex:pollen).
rdf:_3(ex:Manos_allergies, ex:dust).
ex:suffers(ex:Manos, ex:excessive_eye_blinking).
ex:age(ex:Manos, "6"^^xsd:integer).
ex:sex(ex:Manos, ex:male)
```

Ontology O_2

$\langle http://doctor2.gr \rangle$

$P_{O_2} =$
```
ex:suffers(?x,ex:psychological_eye_tic)← ex:has_condition(?x,excessive_eye_blinking),
    ∼ has_condition(?x, ex:red_eye)@{Nam_O1}, ex:age(?x, ?y),
   ex:greater_than(?y, "5"^^xsd:integer), ex:less_than(?y, "10"^^xsd:integer),
   ex:sex(?x,ex:male).
```

Ontology O_3

$\langle http://doctor1.gr \rangle$

$P_{O_3} =$
```
ex:suffers(?x,ex:allergic_conjunctivitis)← ex:has_condition(?x,excessive_eye_blinking),
    rdf:type(?x, ex:Myltiple_Allergic).
```

Ontology O_4

$\langle http://medical.gr \rangle$

$P_{O_4} =$
```
ex:can_play_with(?x, ex:flowers) ← ex:suffers(?x,ex:psychological_eye_tic).
¬ ex:can_play_with(?x, ex:flowers) ← rdf:type(?x,ex:allergic_conjunctivitis).
```

Conflicts

`conflict`$(http://doctor1.gr, http://doctor2.gr)$.

Fig. 5. A medical g-RDF information base

9 Related Work

In this section, we review several related works. We divide them into two parts: those that are concerned with provenance information in RDF and those that are concerned with modular RDF rule bases.

Carroll et al. [12, 11] identify the problem of provenance of RDF triples. They propose *named graphs*, as an entity denoting a collection of RDF triples, which can be annotated with relevant provenance information. In particular, they do not specify how provenance itself should be represented, but they simply offer a placeholder for its representation. This approach is in contrast to earlier attempts for provenance, such as RDF reification. Similarly to [12, 11], we use a URI for naming g-RDF ontologies and this URI can be used for adding metadata, associated with the particular g-RDF ontology, such as temporal and spatial status, and trust.

Guha and Fikes [35] provide contexts, aggregate contexts, lifting rules, selective importing, preference rules, etc. They modify the RDF model theory to have additional context parameters both in the abstract syntax being interpreted and in the considered simple interpretations. However, the work in [35] does not consider g-RDF programs and why-provenance information derived based on the (named) g-RDF ontologies of a g-RDF information base and the RDFS entailment rules.

MacGregor and Ko [50], replace each RDF triple by a quadruple, where the fourth element is the name of the context that the RDF triple is true. In our case, extending [50], we derive why-provenance of derived g-RDF triples.

Flouris et al. [26] argue that named graphs alone cannot capture provenance information in the presence of RDFS reasoning and updates. Given two triples belonging to separate named graphs, which graph does any tuple inferred from these belong to? In other words, shared origin cannot be captured by named graphs alone. To remedy this problem, they extend RDF triples to RDF quadruples, where the fourth element is the set of graph names that participated in the derivation of the RDF triple through a limited subset of the RDFS entailment rules. They also discuss atomic update operations (i.e., inserts and deletes) of the RDF quadruples. In our work, we extend [26] regarding the inference part (as we do not consider update operations). We add to RDF graphs, g-RDF programs building named g-RDF ontologies and we consider g-RDF triples derived based on (non-conflicting) g-RDF ontologies and the provenance inference rules (which extend the RDFS inference rules). We associate derived g-RDF triples t with the set of names of g-RDF ontologies that contributed to the derivation of t. In contrast to [26], we present a provenance model theory that faithfully extends the RDFS model theory and presented a complete set of provenance inference rules that are used to implement our provenance stable model theory through Answer Set Programming. In particular, [26] does not present a model theory and from our provenance inference rules, they consider just the rules 7, 9, 10, 11, 13, and 14 (without the *compatible* predicate). Additionally, in contrast to [26], we support explicit negation, scoped weak negation, and conflicts between sources.

Zimmermann et al. [63] also consider provenance information in the presence of RDFS reasoning. In particular, the authors present a generic framework for representing and reasoning with annotated Semantic Web data. They formalize the annotated language, present a corresponding deductive system, and address the query answering problem. The authors show that their framework can be instantiated for the temporal, fuzzy, and provenance annotation domain. The authors present a model theory that, from our provenance RDFS conditions in Definition 11 (that concern property-truth extensions), ignore conditions 1,2,3,4,7,10,14, 16,17,18,19, 20, and 21. In fact the authors extend ρdf [51] that covers the essential features of RDFS. Additionally, the work in [63] does not consider program rules, negation, and conflict statements between sources. Here, we would like to state that the same annotated RDFS model theory and the same annotated RDFS inference rules as that of [63] are presented in [10]. In general, we can say that we do not focus in the RDFS inference with general annotations but on arbitrary rules with a specific and significant model for why-provenance.

Ding et al. [22] introduce the notion of *RDF molecule*, a "lossless" RDF graph partition, which offers a level of granularity between an RDF graph and an RDF triple. RDF molecules are the smallest meaning-preserving subgraphs for which we might seek independent evidence. Thus, they provide a useful granularity for tracking the provenance or evidence of the information found in a RDF graph. Yet, this work is not concerned with why-provenance information of derived RDF triples.

Hartig and Zhao [38, 36] present a provenance vocabulary that assists providers of Linked Data [40] to describe the provenance of their data using RDF. Their provenance model includes two dimensions: data creation and data access. They explain how this vocabulary can be used to describe data items and how metadata with such provenance descriptions can be an integral part of the Web of Data. Furthermore, they discuss how such metadata can be consumed in order to support a scenario of identifying outdated information from the Web of Data. In [37], the authors show how their proposed provenance vocabulary can be seen as a Web data specific specialization of the W3C PROV Ontology (PROV-O) [6], such that proposed classes and properties are domain specific extensions of the more general concepts introduced in PROV-O. The W7 model [56] is an ontological model created to describe the semantics of data provenance, and uses seven fundamental elements: *what, when, where, how, who, which,* and *why.* It has been built with general and extensible principles, hence it is possible to capture provenance semantics for data in different domains. Our work is complementary to the work in [38, 36, 56], as named g-RDF ontologies can possibly be annotated using terms from the proposed provenance vocabularies. This information can be used in identifying the characteristics of derived g-RDF triples based on their associating sets of provenance names. For example, if each g-RDF ontology in $\mathcal{C}$ is associated with a certainty degree, the certainty degree of a derived g-RDF triple $[\neg]p(s,o)$ is equal to $max(\{min(\{certainty(O) \mid O \in S\}) \mid \mathcal{C} \models^{pr} [\neg]H(s,p,o,S)\})$. If each g-RDF ontology O in $\mathcal{C}$ is associated with a set of authors, the set (of sets) of authors of a derived g-RDF triple $p(s,o)$ is equal to $\{\bigcup\{authors(O) \mid O \in S\} \mid \mathcal{C} \models^{pr} [\neg]H(s,p,o,S)$ and $\mathcal{C} \not\models^{pr} [\neg]H(s,p,o,S')$, for $S' \subseteq S\}$.

Cui [16] was among the first to formalize a notion of why-provenance of data in the context of relational databases, called *lineage*. She associated each tuple t present in the output of a query with the set of tuples, called the *lineage* of t, that helped to produce t. Green et al. [34], extending [16], formalized a notion of how-provenance for relational algebra in terms of an appropriate "provenance semiring". How-provenance contains information about how an output tuple is derived according to a query. The idea behind the semiring framework is to distinguish two basic transformations that source tuples undergo as a result of applying a relational query to a source database: they can be either joined together as an effect of a join, or merged together, via union or projection. Green et al. [34] generalized the provenance semirings framework to consider "pure" datalog programs, where the body of each datalog rule consists of only relational atoms. A difficulty that arises in extending the semiring model in this context has to do with handling recursion. Karvounarakis et al. [42] extended semiring provenance [34] and developed a query language for provenance which can express queries such that "Is this tuple derivable from trusted sources?", "What tuples are derived from this relationship", "What score should this answer receive, given initial scores of the base tuples?". Though works [34, 42] are very interesting, they are not concerned with RDF ontologies and negation.

The difficulty in handling negation, or more generally, difference in the relational provenance models is well-known and several alternative models have been proposed [29, 33, 1]. In [17], Damasio et al. proposed a first-time provenance model for a significant fragment of SPARQL [55], based on the relational annotated provenance models (or K-relations). Quite recently, the use of semiring models for SPARQL has been shown not adequate to handle the `OPTIONAL` construct, and new algebraic structures have been proposed [28]. Dividino et al. [23] presented a generic and formalized model for the management of many dimensions of meta knowledge, like source, authorship, certainty, and others, for RDF repositories. In particular, they extended SPARQL query processing in such a way that given a SPARQL query for data, one may request meta knowledge. Yet, [29, 33, 1] are concerned with the provenance of relational queries over relational data, while [17, 28, 23] are concerned with the provenance of SPARQL queries over RDF data. In our case, we are concerned with the provenance of g-RDF triples or weakly negated g-RDF triples from a g-RDF information base that supports negation and recursion.

Below, we review works that consider program rules distributed in the Semantic Web and RDF. Yet, these works do not consider provenance.

Analyti et al. [2] defined a modular ERDF ontology $\mathcal{M}$ as a collection of $\mathtt{r}$-ERDF ontologies, where an $\mathtt{r}$-ERDF ontology is the combination of (i) an ERDF graph G containing (implicitly existentially quantified) positive and negative information, and (ii) an $\mathtt{r}$-ERDF program P containing derivation rules. An $\mathtt{r}$-ERDF program P can possibly have (i) all connectives $\sim$, $\neg$, $\supset$, $\wedge$, $\vee$, $\forall$, $\exists$ in the body of a rule, (ii) strong negation $\neg$ in the head of a rule, and (iii) qualified literals in the body of a rule, for interaction with the other $\mathtt{r}$-ERDF ontologies in $\mathcal{M}$. The authors defined the modular stable model semantics of $\mathcal{M}$, model-theoretically, based on partial logic [41] and they showed that modular stable model entailment on modular ERDF ontologies extends RDFS entailment on RDF graphs.

TRIPLE [59] is a rule language for the Semantic Web that supports modules (called, *models* there), qualified literals, and dynamic module transformation. Its syntax is based on F-Logic [44]. Arbitrary formulas can be used in the body of a rule, handled through the Lloyd-Topor transformations [49]. Module indicators in qualified literals can be module names, variables, or skolem functions, as well as conjunction and difference of module indicators. Part of the semantics of the RDF(S) vocabulary is represented as pre-defined rules (and not as semantic conditions on interpretations), which are grouped together in a module. The semantics of a modular rule base is defined, based on the *well-founded semantics* (`WFS`) [30] of an equivalent logic program. Yet, the model-theoretic semantics of TRIPLE [21] does not faithfully extend RDFS semantics [39] and is not, in general, equivalent to its transformational semantics.

A modularity framework for RDF rule bases is proposed in [54]. There, RDFS semantics are partially represented through a normal logic program, associated with a special context/module c_{RDFS}. The *contextually closed* `AS` and *contextually closed* `WFS` semantics of such a modular RDF rule base $\mathcal{R}$ are defined, through the `AS` [32] and `WFS` [30] semantics of a normal logic program $\mathcal{R}_{CC}$, generated from $\mathcal{R}$, respectively. In particular, all simple atoms appearing in a rule base s are qualified by the name of s. The resulting rules union the original rules of each rule base $s \in \mathcal{R}$ form the logic program $\mathcal{R}_{CC}$. Yet, this framework does not have a model-theoretic semantics that faithfully extends RDFS model theory.

Finally, we would like to mention a general framework for multi-context reasoning, proposed in [8], that allows to combine arbitrary monotonic and non-monotonic logics. Information flow between the different contexts is achieved through a set of non-monotonic bridge rules. In particular, in [8], knowledge bases of different logics (such as Default Logic, normal logic programs under answer set semantics, propositional logics under closed world semantics) are associated with a set of bridge rules that possibly contain weakly negated elements qualified over other knowledge bases. Then, the ground equilibriums for such a system are defined.

Note that in all of the above four frameworks, provenance information is ignored and literals can be possibly qualified by a *single* module (the same is also true for TRIPLE under its transformational semantics). Yet, in [2], modules are importing properties and classes from specified other modules. In our case, such importing is not needed because g-RDF ontologies of a g-RDF information base are assumed to freely exchange information among themselves.

In [47], an alternative definition of the answer sets for normal logic programs [31] is proposed. In particular, let P be a normal logic program. A set X is an answer set of P if and only X is the result of a series of interpretations of P, $\langle X_i \rangle_0^\infty$ that satisfy the following properties:

1. *Revision*: each successive element in the computation $\langle X_i \rangle_0^\infty$ must be grounded in the preceding one and the program.
2. *Persistence of beliefs:* each next element in the computation $\langle X_i \rangle_0^\infty$ must contain the previous one.
3. *Convergence*: a computation $\langle X_i \rangle_0^\infty$ continues until it stabilizes.
4. *Persistence of reasons*: for every $\alpha \in X^\infty$ there is a rule $r_\alpha \in P$ (called the reason for α) whose head is α and whose body holds in every X_i, $i \geq i_\alpha - 1$, where i_α is the least integer such that $\alpha \in X_{i_\alpha}$.

Our definition of $\#n$-provenance stable model of $\mathcal{C}$ satisfies all four above properties (of course, properly adapted).

10 Conclusions

A g-RDF ontology is the combination of (i) a g-RDF graph G (i.e., a set of positive and strongly negated RDF triples, called g-RDF triples) and (ii) a g-RDF program P containing derivation rules with possibly both explicit and scoped weak negation. In this work, we considered collections of g-RDF ontologies, distributed over the web, along with a set of *conflict* statements, expressing that information within a pair of g-RDF ontologies cannot be combined together for deriving new information. We name such a collection, as *g-RDF information base*. Information can be inferred through the g-RDF ontologies of a g-RDF information base or through the provenance RDFS inference rules. We associate each derived grounded g-RDF triple $[\neg]p(s, o)$ with the set of names S of the g-RDF ontologies that contributed to its derivation. To achieve this, we define the provenance models of a g-RDF information base. We showed that our provenance semantics faithfully extends RDFS semantics. We provided an algorithm based on Answer Set Programming that computes all entailed provenance quads $[\neg]H(s, p, o, S)$ of a g-RDF information base and provides the answer to various kinds of queries. We showed that the complexity of query answering is we show that the complexity of query answering is co-NP$^{\text{NP}}$-complete w.r.t. the size of $\mathcal{C}$, and co-NP-complete w.r.t. the sizes of the g-RDF graphs appearing in $\mathcal{C}$. In particular, $\#n$-provenance stable model semantics can be implemented using any answer set solver like `smodels` [52], `dlv` [46], or `clasp` [27].

Finally, we considered the case of positive g-RDF information bases. We show that the complexity of query answering for positive g-RDF information bases is NP-complete w.r.t. the sizes of $\mathcal{C}$ and P-complete w.r.t. the sizes of the g-RDF graphs appearing in $\mathcal{C}$. We provided a translation into XSB-PROLOG that can be used to provide the answers of all kinds of defined queries over a positive g-RDF information base, using the XSB logic programming system [58]. This approach is complementary to query answering through answer set programming.

Note that we can easily modify our query language to support queries as the following: (i) give me the minimum sets (w.r.t. $\subseteq$) of provenance names of a derived g-RDF triple $[\neg]p(s,o)$ and (ii) give me the maximum sets (w.r.t. $\subseteq$) of provenance names of a derived weakly negated g-RDF triple $\sim[\neg]p(s,o)$.

Since no one can guarantee that every derived g-RDF triple is error free or globally accepted, why-provenance information of derived g-RDF trples is a good heuristic for evaluating trustworthiness and identifying different points of view, authorship, or temporal and spatial contexts. With why-provenance information, i.e., "why a derived g-RDF triple is true", one can propagate his/her trust in g-RDF ontologies to trustworthiness judgments against the derived g-RDF triples. Additionally, based on these sets, different points of view, authorship, temporal, and spatial information can be associated with a derived g-RDF triple.

The use of non-monotonic negation is not consensual in the Semantic Web and therefore a scoped negation as failure is provided as an approach to reach compromise. This requires the user to explicitly specify the g-RDF ontologies (context) where she wishes to evaluate the query, since an unrestricted use of the weak negation operator might result in unexpected interactions among the several ontologies in information bases. The sources used to evaluate weak negation queries are not included in the provenance information of quads because they might convey erroneous information to the user. As they are now, the set of source names are monotonic; therefore, the user knows if more ontologies are added to the information base the conclusions would not be retracted. If source names used for evaluating weakly negated queries are also included, then "holes" would appear in this hierarchy making the interpretation of the results more difficult to the user. Since the evaluation of weak negations are fixed in the ontology programs, the approach followed factors out this information from the provenance in quads. Moreover, the propagation of provenance information over weakly negated literals in logic programs has only been addressed recently in the literature (see our work in [18]), but requires even more complex representations of the annotations.

Note that both the extended logic program $P_{\mathcal{C}}^{\#n}$ for general g-RDF information based and the XSB prolog program $W_{\mathcal{C}}^{\#n}$ for positive g-RDF information bases are exponential in size with respect to the number of sources. Thus, they will not scale up well with respect to the number of sources. However, this is inevitable as provenance is a subset of the set of sources on the g-RDF information base. Note that the exact complexity results are provided in propositions 9, 10, and 11. Regarding implementation we have just implemented the XSB-Prolog program for verification proposes. Future work concerns implementing the extended logic program $P_{\mathcal{C}}^{\#n}$ and providing experimental results.

As future work, we also plan to devise a query language that queries not only the derived g-RDF triples and their accompanied sets of provenance names but also the provenance metadata of the g-RDF ontologies. For example, return all entailed g-RDF triples that are supported by at least four minimal (w.r.t. $\subseteq$) sets of provenance names S, all co-authored by Mary, i.e., $\forall S\ \exists O \in S$ s.t. $\texttt{ex:author}(\texttt{ex:Mary}, Nam_O)$ is true.

Acknowledgments: The authors would like to thank the anonymous referees for their valuable comments. Carlos Viegas Damsio was partially supported by FCT Project ERRO PTDC/EIACCO/121823/2010.

Appendix A: Proofs

In this appendix, we prove the propositions presented in the main paper. For simplicity, we have eliminated all namespaces from the considered URIs. We provide a few definitions that will be used throughout the proofs.

Let I be a provenance RDFS interpretation of a vocabulary V and a g-RDF information base $\mathcal{C}$. Let $x, y \in Res_I$ and $S \subseteq names_{\mathcal{C}}$. We define a mapping $CT_I : Res_I \rightarrow \mathcal{P}(Res_I \times \mathcal{P}(names_{\mathcal{C}}))$ s.t. $\langle x, S\rangle \in CT_I(y)$ iff $\langle x, y, S\rangle \in PT_I(I(type))$. Additionally, define a mapping $CF_I : Res_I \rightarrow \mathcal{P}(Res_I \times \mathcal{P}(names_{\mathcal{C}}))$ s.t. $\langle x, S\rangle \in CF_I(y)$ iff $\langle x, y, S\rangle \in PF_I(I(type))$.

Proposition 1. Let $\mathcal{C} = \langle\{\langle c, O\rangle\}, conflict\rangle$ be a g-RDF information base s.t. $O = \langle G, \emptyset\rangle$ is a g-RDF ontology with G be an RDF graph and $conflict = \{\}$. Additionally, let $G' = \{p_1(s_1, o_1), ..., p_m(s_m, o_m)\}$ being an RDF graph. Then,
$\mathcal{C} \models^{pr} \{H(s_1, p_1, o_1, \{c\}), ..., H(s_m, p_m, o_m, \{c\})\}$ iff $G \models^{RDFS} G'$.
Proof:
$\Rightarrow$) Let $\mathcal{C} \models^{pr} \{H(s_1, p_1, o_1, \{c\}), ..., H(s_m, p_m, o_m, \{c\})\}$. We will show that $sk(G) \models^{RDFS} G'$. Let I be an RDFS interpretation of a vocabulary V s.t. $I \models sk(G)$. We will show that $I \models G'$.

Based on I, we construct a provenance simple interpretation J of $V_{\mathcal{C}}$ and $\mathcal{C}$ as follows:

- $Res_J = Res^H_{\mathcal{C}}$.
- $J_V(x) = x$, for all $x \in V_{\mathcal{C}} \cap URI$.
- We define the mapping: $IL_J : V_{\mathcal{C}} \cap \mathcal{TL} \rightarrow Res_J$ such that:
  $IL_J(x) = x$, if x is a typed literal in $V_{\mathcal{C}}$ other than a well-typed XML literal, and $IL_I(x)$ is the XML value of x, if x is a well-typed XML literal in $V_{\mathcal{C}}$.
- We define the mapping: $J : V_{\mathcal{C}} \rightarrow Res_J$ such that:
  - $J(x) = J_V(x),\ \forall x \in V_{\mathcal{C}} \cap URI$.
  - $J(x) = x,\ \forall\, x \in V_{\mathcal{C}} \cap \mathcal{PL}$.
  - $J(x) = IL_J(x),\ \forall\, x \in V_{\mathcal{C}} \cap \mathcal{TL}$.
- $Prop_J = \{x \in Res_J \mid \exists x' \in V_{\mathcal{C}}$ s.t. $J(x') = x$ and $I(x') \in Prop_I\}$.
- The mapping $PT_J : Prop_J \rightarrow \mathcal{P}(Res_J \times Res_J \times \{\{c\}\})$ is defined as follows:
  $\forall x, y, z \in V_{\mathcal{C}}$, it holds that:
  $\langle J(x), J(y), \{c\}\rangle \in PT_J(J(z))$ iff $\langle I(x), I(y)\rangle \in PT_I(I(z))$.
- The mapping $PF_J : Prop_J \rightarrow \mathcal{P}(Res_J \times Res_J \times \{\{c\}\})$ is defined as follows:
  $\forall p \in Prop_J$, $PF_J(p) = \emptyset$.
- $LV_J = \{x \in Res_J \mid \langle x, J(Literal), \{c\}\rangle \in PT_J(J(type))\}$.

To show that J is a provenance simple interpretation of $V_{\mathcal{C}}$ and $\mathcal{C}$, it is enough to show that $V_{\mathcal{C}} \cap \mathcal{PL} \subseteq LV_J$. Let $x \in V_{\mathcal{C}} \cap \mathcal{PL}$. Then, $x \in LV_I$. Thus, $\langle x, I(Literal)\rangle \in PT_I(I(type))$. This implies that $\langle x, J(Literal), \{c\}\rangle \in PT_J(J(type))$. Thus, $x \in LV_J$.

We will now show that J is a provenance RDFS interpretation of $V_{\mathcal{C}}$ and $\mathcal{C}$. First, we will show that J satisfies semantic condition 1 of Definition 11 (provenance RDFS interpretation), in a number of steps:
Step 1: Here, we prove that $Res_J = \{x \mid \langle x, J(Resource), \{c\}\rangle \in PT_J(J(type))\}$. Obviously, $\{x \mid \langle x, J(Resource), \{c\}\rangle \in PT_J(J(type))\} \subseteq Res_J$. We will show that $Res_J \subseteq \{x \mid \langle x, J(Resource), \{c\}\rangle \in PT_J(J(type))\}$. Let $x \in Res_J$. Then, there is $x' \in V_{\mathcal{C}}$ such that $J(x') = x$. We want to show that $\langle J(x'), J(Resource), \{c\}\rangle \in PT_J(J(type))\}$. It holds that: $\langle J(x'), J(Resource), \{c\}\rangle \in PT_J(J(type))$ iff $\langle I(x'), I(Resource)\rangle \in PT_I(I(type))$, which is true, since I is an RDFS interpretation that satisfies $sk(G)$ and $I(x') \in Res_I$. Thus, $x = J(x') \in \{x \mid \langle x, J(Resource), \{c\}\rangle \in PT_J(J(type))\}$.
Therefore, $Res_J = \{x \mid \langle x, J(Resource), \{c\}\rangle \in PT_J(J(type))\}$.

Step 2: Here, we prove that $Prop_J = \{x \mid \langle x, J(\mathit{Property}), \{c\}\rangle \in PT_J(J(\mathit{type}))\}$. We will show that $Prop_J \subseteq \{x \mid \langle x, J(\mathit{Property}), \{c\}\rangle \in PT_J(J(\mathit{type}))\}$. Let $x \in Prop_J$. Then, there is $x' \in V_{\mathcal{C}}$ such that $J(x') = x$ and $I(x') \in Prop_I$. We want to show that $\langle J(x'), J(\mathit{Property})\rangle \in PT_J(J(\mathit{type}))$. It holds that: $\langle J(x'), J(\mathit{Property}), \{c\}\rangle \in PT_J(J(\mathit{type}))$ iff $\langle I(x'), I(\mathit{Property})\rangle \in PT_I(I(\mathit{type}))$, which is true, since $I(x') \in Prop_I$. Thus, $x = J(x') \in \{x \mid \langle x, J(\mathit{Property}), \{c\}\rangle \in PT_J(J(\mathit{type}))\}$. Therefore, $Prop_J \subseteq \{x \mid \langle x, J(\mathit{Property}), \{c\}\rangle \in PT_J(J(\mathit{type}))\}$.

We will now show that $\{x \mid \langle x, J(\mathit{Property}), \{c\}\rangle \in PT_J(J(\mathit{type}))\} \subseteq Prop_J$. Let $x \in \{x \mid \langle x, J(\mathit{Property}), \{c\}\rangle \in PT_J(J(\mathit{type}))\}$. Then, $\exists x' \in V_{\mathcal{C}}$ such that $J(x') = x$. It holds that $\langle J(x'), J(\mathit{Property}), \{c\}\rangle \in PT_J(J(\mathit{type}))$, which implies that $\langle I(x'), I(\mathit{Property})\rangle \in PT_I(I(\mathit{type}))$. Thus, $I(x') \in Prop_I$ and $x \in Prop_J$.
Therefore, $Prop_J = \{x \mid \langle x, J(\mathit{Property}), \{c\}\rangle \in PT_J(J(\mathit{type}))\}$.

Step 3: By definition, it holds that $LV_J = \{x \in Res_J \mid \langle x, J(\mathit{Literal}), \{c\}\rangle \in PT_J(J(\mathit{type}))\}$.

Semantic conditions 2 and 3 of Definition 11 are trivially satisfied based on Steps 1 and 3, above and the fact that $V_{\mathcal{C}} \cap \mathcal{PL} \subseteq LV_J$.

We will now show that J satisfies semantic condition 4 of Definition 11. Let $\langle x, y, \{c\}\rangle \in PT_J(z)$. Thus, $z \in Prop_J$. Therefore, $\langle z, J(\mathit{Property}), \{c\}\rangle \in PT_J(J(\mathit{type}))$.

We will now show that J satisfies semantic condition 5 of Definition 11. Let $\langle x, y, \{c\}\rangle \in PT_J(J(\mathit{domain}))$ and $\langle z, w, \{c\}\rangle \in PT_J(x)$. We will show that $\langle z, y, \{c\}\rangle \in PT_J(J(\mathit{type}))$. There are $x', y' \in V_{\mathcal{C}}$ such that $J(x') = x,\ J(y') = y$. Thus, $\langle J(x'), J(y'), \{c\}\rangle \in PT_J(J(\mathit{domain}))$. Additionally, there are $z', w' \in V_{\mathcal{C}}$ such that $J(z') = z,\ J(w') = w$. Thus, $\langle J(z'), J(w'), \{c\}\rangle \in PT_J(J(x'))$. Then, $\langle I(x'), I(y')\rangle \in PT_I(I(\mathit{domain}))$ and $\langle I(z'), I(w')\rangle \in PT_I(I(x'))$. Since I is an RDFS interpretation, $\langle I(z'), I(y')\rangle \in PT_I(I(\mathit{type}))$. Thus, $\langle J(z'), J(y'), \{c\}\rangle \in PT_J(J(\mathit{type}))$ and $\langle z, y, \{c\}\rangle \in PT_J(J(\mathit{type}))$.

In a similar manner, we can prove that J also satisfies the rest of the semantic conditions of Definition 11. Thus, J is a provenance RDFS interpretation of $V_{\mathcal{C}}$ and $\mathcal{C}$. Thus, $J \in \mathcal{I}^H(\mathcal{C})$.

Let $p(s, o) \in sk(G)$. We will now show that $J \models H(s, p, o, \{c\})$. It holds that $p, s, o \in V_{\mathcal{C}}$. Since $I \models sk(G)$, it holds that $I(p) \in Prop_I$. Thus, $\langle I(p), I(\mathit{Property})\rangle \in PT_I(I(\mathit{type}))$, which implies that $\langle J(p), J(\mathit{Property}), \{c\}\rangle \in PT_J(J(\mathit{type}))$. From this, it follows that $J(p) \in Prop_J$. It holds that: $\langle J(s), J(o), \{c\}\rangle \in PT_J(J(p))$ iff $\langle I(s), I(o)\rangle \in PT_I(I(p))$. The last statement is true since $I \models sk(G)$. Let $v : \{\} \rightarrow Res^H_{\mathcal{C}}$. Then, $J, v \models H(s, p, o, \{c\})$. Thus, $J \models H(s, p, o, \{c\})$. Therefore, J is a provenance model of $\mathcal{C}$, i.e., $J \in \mathcal{M}(\mathcal{C})$.

Since $\mathcal{C} \models^{pr} \{H(s_1, p_1, o_1, \{c\}), ..., H(s_m, p_m, o_m, \{c\})\}$, it follows that $J \models \{H(s_1, p_1, o_1, \{c\}), ..., H(s_m, p_m, o_m, \{c\})\}$. Therefore, there exists $u : Var(G') \rightarrow Res_J$ s.t. $J, u \models \{H(s_1, p_1, o_1, \{c\}), ..., H(s_m, p_m, o_m, \{c\})\}$.

We now define a total function $u' : V_{G'} \cup Var(G') \rightarrow V_{\mathcal{C}}$, as follows:

$$u'(x) = \begin{cases} u(x) \text{ if } x \in Var(G') \text{ and} \\ \qquad u(x) \text{ is not the xml value of a well-typed XML literal in } V_{\mathcal{C}} \\ t \quad\ \text{if } x \in Var(G') \text{ and} \\ \qquad u(x) \text{ is the xml value of a well-typed XML literal } t \text{ in } V_{\mathcal{C}} \\ x \quad\ \text{otherwise} \end{cases}$$

Moreover, we define a total function $u'' : Var(G') \rightarrow Res_I$ s.t. $u''(x) = I(u'(x))$.

Let $V' = V \cup \mathcal{V}_{RDF} \cup \mathcal{V}_{RDFS}$. We will show that $I, u'' \models G'$.
Let $p(s, o) \in G'$. Then, $p \in V_{G'}$ and $s, o \in V_{G'} \cup Var$. Since $J, u \models \{H(s_1, p_1, o_1, \{c\}), ..., H(s_m, p_m, o_m, \{c\})\}$, it follows that $V_{G'} \subseteq V_{\mathcal{C}}$. Therefore, $V_{G'} \subseteq V_{sk(G)} \cup \mathcal{V}_{RDF} \cup \mathcal{V}_{RDFS} \subseteq V'$. Thus, $p \in V'$ and $s, o \in V' \cup Var$.

We will now show that $I(p) \in Prop_I$. It holds that:
$\langle I(p), I(\mathit{Property})\rangle \in PT_I(I(\mathit{type}))$ iff

$\langle J(p), J(Property), \{c\}\rangle \in PT_J(J(type))$, which holds since $J, u \models \{H(s_1, p_1, o_1, \{c\})$, $..., H(s_m, p_m, o_m, \{c\})\}$.

We want to show that $\langle [I + v''](s), [I + v''](o)\rangle \in PT_I(I(p))$. Note that $\forall x \in V_{G'}$, it holds that: $[I + u''](x) = I(u'(x)) = I(x)$ and $J(u'(x)) = [J + u](x) = J(x)$. Moreover, $\forall x \in Var(G')$, it holds that: $[I + u''](x) = I(u'(x))$ and $J(u'(x)) = [J + u](x)$ (recall the definition of $J(.)$). Therefore, it holds that:
$\langle [I + u''](s), [I + u''](o)\rangle \in PT_I(I(p))$ iff
$\langle I(u'(s)), I(u'(o))\rangle \in PT_I(I(p))$ iff
$\langle J(u'(s)), J(u'(o)), \{c\}\rangle \in PT_J(J(p))$ iff
$\langle [J + u](s), [J + u](o), \{c\}\rangle \in PT_J(J(p))$, which is true since $J, u \models \{H(s_1, p_1, o_1, \{c\})$, $..., H(s_m, p_m, o_m, \{c\})\}$. Thus, $I, u'' \models p(s, o)$. Therefore, $I \models G'$.

Thus, $sk(G) \models^{RDFS} G'$. Therefore, $G \models^{RDFS} G'$.

$\Leftarrow$) Let $G \models^{RDFS} G'$. Thus, $sk(G) \models^{RDFS} G'$. We will show that $\mathcal{C} \models^{pr} \{H(s_1, p_1, o_1, \{c\})$, $..., H(s_m, p_m, o_m, \{c\})\}$. Let $I \in \mathcal{M}(\mathcal{C})$. Note that I is a provenance RDFS interpretation of $V_\mathcal{C}$ and $\mathcal{C}$, such that $I \models H(s, p, o, \{c\})$, for each $p(s, o) \in sk(G)$. Based on I, we will construct an RDFS interpretation J of $V_\mathcal{C}$.

First, based on I, we construct a simple interpretation J of $V_\mathcal{C}$ as follows:

- $Res_J = Res^H_\mathcal{C}$.
- $J_V(x) = x$, for all $x \in V_\mathcal{C} \cap URI$.
- We define the mapping: $IL_J : V_\mathcal{C} \cap \mathcal{TL} \rightarrow Res_J$ such that:
  $IL_J(x) = x$, if x is a typed literal in $V_\mathcal{C}$ other than a well-typed XML literal, and $IL_I(x)$ is the XML value of x, if x is a well-typed XML literal in $V_\mathcal{C}$.
- We define the mapping: $J : V_\mathcal{C} \rightarrow Res_J$ such that:
  - $J(x) = J_V(x),\ \forall x \in V_\mathcal{C} \cap URI$.
  - $J(x) = x,\ \forall\, x \in V_\mathcal{C} \cap \mathcal{PL}$.
  - $J(x) = IL_J(x),\ \forall\, x \in V_\mathcal{C} \cap \mathcal{TL}$.
- $Prop_J = Prop_I$.
- The mapping $PT_J : Prop_J \rightarrow \mathcal{P}(Res_J \times Res_J)$ is defined as follows:
  $\forall x, y, z \in Res_J$, it holds that:
  $\langle x, y\rangle \in PT_J(z)$ iff $\langle x, y, \{c\}\rangle \in PT_I(z)$.
- $LV_J = \{x \in Res_J \mid \langle x, J(Literal)\rangle \in PT_J(J(type))\}$.

As in the proof of $\Rightarrow$), we can show that: (i) J is an RDFS interpretation $V_\mathcal{C}$, (ii) $J \models G'$, and (iii) $I \models \{H(s_1, p_1, o_1, \{c\}), ..., H(s_m, p_m, o_m, \{c\})\}$. Therefore, $\mathcal{C} \models^{pr}$ $\{H(s_1, p_1, o_1, \{c\}), ..., H(s_m, p_m, o_m, \{c\})\}$. $\square$

Proposition 2. Let $O = \langle G, P\rangle$ be a simple ERDF ontology and $\mathcal{C} = \langle\{(c, O')\}, \emptyset\rangle$ be a g-RDF information base s.t. $O' = \langle G', P'\rangle$, where G' and P' are defined as follows:

$$\begin{aligned} G' = G \cup \{&\texttt{rdfs:subClassOf}(\texttt{erdf:TotalProperty}, \texttt{rdfs:Class}),\\ &\texttt{rdfs:subClassOf}(\texttt{erdf:TotalClass}, \texttt{rdfs:Class})\end{aligned}$$

$$\begin{aligned} P' = P \cup \{&\neg ?p(?s, ?o) \leftarrow \texttt{rdf:type}(?p, \texttt{erdf:TotalProperty}), \sim ?p(?s, ?o).\\ &?p(?s, ?o) \leftarrow \texttt{rdf:type}(?p, \texttt{erdf:TotalProperty}), \sim\neg ?p(?s, ?o).\\ &\neg\texttt{rdf:type}(?o, ?c) \leftarrow \texttt{rdf:type}(?c, \texttt{erdf:TotalClass}), \sim\texttt{rdf:type}(?o, ?c).\\ &\texttt{rdf:type}(?o, ?c) \leftarrow \texttt{rdf:type}(?c, \texttt{erdf:TotalClass}), \sim\neg\texttt{rdf:type}(?o, ?c).\}\end{aligned}$$

Let $p(s, o)$ be a ground g-RDF triple over $V_\mathcal{C}$. Then, it holds that $O \models^{st} [\sim][\neg]p(s, o)$ iff $\mathcal{C} \models^{pr} [\sim][\neg]H(s, p, o, \{c\})$, where $\models^{st}$ is entailment from an ERDF ontology under the stable model semantics.

Proof:
$\Rightarrow$) Assume that $O \models^{st} [\sim][\neg]p(s,o)$. We will show that iff $\mathcal{C} \models^{pr} [\sim][\neg]H(s,p,o,\{c\})$.

Let $M \in \mathcal{M}(\mathcal{C})$. Then, there is a chain of provenance Herbrand interpretations of $\mathcal{C}$, $I_0 \leq ... \leq I_{m+1}$ s.t. $I_m = I_{m+1} = M$ and:

1. $I_0 = minimal(\{I \in \mathcal{I}^H(\mathcal{C}) \mid I \models [\neg]H(s,p,o,\{c\})$, for each $[\neg]p(s,o)$ appearing in the $sk(G')\})$.
2. For natural numbers α with $0 < \alpha \leq m+1$:
   $I_\alpha = minimal(\{I \in \mathcal{I}^H(\mathcal{C}) \mid I \geq I_{\alpha-1}$ and for all $q_0 \leftarrow q_1, ..., q_l, \sim q_{l+1}, ..., \sim q_k \in [prov(P')]_{\mathcal{C}}$, it holds that if (i) $I_{\alpha-1} \models q_i$, for $i = 1, ..., l$ and (ii) $M \models \sim q_i$, for $i = l+1, ..., k$ then $I \models q_0\})$.

Now, let[30] $M' \in \mathcal{I}^H(O)$ s.t. $M' \models [\sim][\neg]p'(s',o')$ iff $M \models [\sim][\neg]H(s',p',o',\{c\})$, for any ground g-RDF triple $p'(s',o')$. It holds that there is a chain of Herbrand interpretations of O, $I'_0 \leq ... \leq I'_{m+1}$, such that $I'_m = I'_{m+1} = M'$ and:

1. $I'_0 \in minimal(\{I \in \mathcal{I}^H(O) \mid I \models sk(G)\})$.
2. For successor ordinals α with $0 < \alpha \leq k+1$:
   $I'_\alpha \in minimal(\{I \in \mathcal{I}^H(O) \mid I \geq I'_{\alpha-1}$ and it holds that:
   $\forall\, r \in [P]_{V_O}$, if $J \models Body(r)$, $\forall J \in \mathcal{I}^H(O)$ s.t. $I'_{\alpha-1} \leq J \leq M'$, then $I \models Head(r)\}$.

Therefore, $M' \in \mathcal{M}^{st}(O)$. Since $O \models^{st} [\sim][\neg]p(s,o)$, it follows that $M' \models [\sim][\neg]p(s,o)$. Therefore, $M \models [\sim][\neg]H(s,p,o,\{c\})$. Thus, $\mathcal{C} \models^{pr} [\sim][\neg]H(s,p,o,\{c\})$.

$\Leftarrow$) Assume that $\mathcal{C} \models^{pr} [\sim][\neg]H(s,p,o,\{c\})$. We will show that iff $O \models^{st} [\sim][\neg]p(s,o)$.

Let $M \in \mathcal{M}^{st}(O))$. Then, there is a chain of Herbrand interpretations of O, $I_0 \leq ... \leq I_{m+1}$, such that $I_m = I_{m+1} = M$ and:

1. $I_0 \in minimal(\{I \in \mathcal{I}^H(O) \mid I \models sk(G)\})$.
2. For successor ordinals α with $0 < \alpha \leq k+1$:
   $I_\alpha \in minimal(\{I \in \mathcal{I}^H(O) \mid I \geq I_{\alpha-1}$ and it holds that:
   $\forall\, r \in [P]_{V_O}$, if $J \models Body(r)$, $\forall J \in \mathcal{I}^H(O)$ s.t. $I_{\alpha-1} \leq J \leq M$, then $I \models Head(r)\}$.

Now, let $M' \in \mathcal{I}^H(\mathcal{C})$ s.t. $M' \models [\sim][\neg]H(s',p',o',\{c\})$ iff $M \models [\sim][\neg]p'(s',o')$, for any ground g-RDF triple $p'(s',o')$. It holds that there is a chain of provenance Herbrand interpretations of $\mathcal{C}$, $I'_0 \leq ... \leq I'_{m+1}$ s.t. $I'_m = I'_{m+1} = M'$ and:

1. $I'_0 = minimal(\{I \in \mathcal{I}^H(\mathcal{C}) \mid I \models [\neg]H(s,p,o,\{c\})$, for each $[\neg]p(s,o)$ appearing in $sk(G')\})$.
2. For natural numbers α with $0 < \alpha \leq m+1$:
   $I'_\alpha = minimal(\{I \in \mathcal{I}^H(\mathcal{C}) \mid I \geq I'_{\alpha-1}$ and for all $q_0 \leftarrow q_1, ..., q_l, \sim q_{l+1}, ..., \sim q_k \in [prov(P')]_{\mathcal{C}}$, it holds that if (i) $I'_{\alpha-1} \models q_i$, for $i = 1, ..., l$ and (ii) $M' \models \sim q_i$, for $i = l+1, ..., k$ then $I \models q_0\})$.

Therefore, $M' \in \mathcal{M}(\mathcal{C})$. Since $\mathcal{C} \models^{pr} [\sim][\neg]H(s,p,o,\{c\})$, it follows that $M' \models [\sim][\neg]H(s,p,o,\{c\})$. Therefore, $M \models [\sim][\neg]p(s,o)$. Thus, $O \models^{st} [\sim][\neg]p(s,o)$. $\square$

Proposition 3. Let $\mathcal{C}$ be a g-RDF information base that does not contain $\sim$ and let $n \geq n_{\mathcal{C}}$. Let G be a provenance graph s.t. $max(\{i \in I\!N \mid rdf{:}\_i \in V_G\}) \leq n$. It holds that: $\mathcal{C} \models^{pr} G$ iff $\mathcal{C} \models^{pr\#_n} G$.
Proof:
$\Rightarrow$) Let $\mathcal{C} = \langle\{\langle Nam_{O_i}, O_i\rangle \mid i = 1, ..., m\}, \mathit{conflict}\rangle$, where each $O_i = \langle G_i, P_i\rangle$ is a g-RDF ontology. Assume that $\mathcal{C} \models^{pr} G$. We will show that $\mathcal{C} \models^{pr\#_n} G$. Specifically, let $I \in \mathcal{M}^{\#_n}(\mathcal{C})$, we will show that $I \models G$.

[30] Note that here we use notation defined in [5].

First, we define a mapping $m : Res_{\mathcal{C}}^{H} \to Res_{\mathcal{C}}^{H_{\#n}}$ as follows:

$$m(x) = \begin{cases} rdf\text{:\_}n \text{ if } x \in \{rdf\text{:\_}i \mid i > n\}, \text{ and} \\ x \qquad \text{otherwise} \end{cases}$$

Based on I, we construct a provenance simple interpretation J of $V_{\mathcal{C}}$ and $\mathcal{C}$ as follows:

– $Res_J = Res_{\mathcal{C}}^{H}$.
– $J_V(x) = x$, for all $x \in V_{\mathcal{C}} \cap URI$.
– We define the mapping: $IL_J : V_{\mathcal{C}} \cap \mathcal{TL} \to Res_J$ such that:
  $IL_J(x) = x$, if x is a typed literal in $V_{\mathcal{C}}$ other than a well-typed XML literal, and $IL_I(x)$ is the XML value of x, if x is a well-typed XML literal in $V_{\mathcal{C}}$.
– We define the mapping: $J : V_{\mathcal{C}} \to Res_J$ such that:
  - $J(x) = J_V(x),\ \forall x \in V_{\mathcal{C}} \cap URI$.
  - $J(x) = x,\ \forall\, x \in V_{\mathcal{C}} \cap \mathcal{PL}$.
  - $J(x) = IL_J(x),\ \forall\, x \in V_{\mathcal{C}} \cap \mathcal{TL}$.
– $Prop_J = \{x \in Res_J \mid m(x) \in Prop_I\}$.
– The mapping $PT_J : Prop_J \to \mathcal{P}(Res_J \times Res_J \times \mathcal{P}(names_{\mathcal{C}}))$ is defined as follows:
  $\forall x, y, z \in Res_J$ and $S \subseteq names_{\mathcal{C}}$, it holds that:
  $\langle x, y, S\rangle \in PT_J(z)$ iff $\langle m(x), m(y), S\rangle \in PT_I(m(z))$.
– The mapping $PF_J : Prop_J \to \mathcal{P}(Res_J \times Res_J \times \mathcal{P}(names_{\mathcal{C}}))$ is defined as follows:
  $\forall x, y, z \in Res_J$ and $S \subseteq names_{\mathcal{C}}$, it holds that:
  $\langle x, y, S\rangle \in PF_J(z)$ iff $\langle m(x), m(y), S\rangle \in PF_I(m(z))$.
– $LV_J = \{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Literal), S\rangle \in PT_J(J(type))\}$.

To show that J is a provenance simple interpretation of $V_{\mathcal{C}}$ and $\mathcal{C}$, it is enough to show that $V_{\mathcal{C}} \cap \mathcal{PL} \subseteq LV_J$. Let $x \in V_{\mathcal{C}} \cap \mathcal{PL}$. Then, $x \in LV_I$. Thus, $\exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, I(Literal), S\rangle \in PT_I(I(type))$. This implies that $\langle x, J(Literal), S\rangle \in PT_J(J(type))$. Thus, $x \in LV_J$.

We will now show that J is a provenance RDFS interpretation of $V_{\mathcal{C}}$ and $\mathcal{C}$. Note that $\forall x \in V_{\mathcal{C}}^{\#n}$, it is the case that $m(J(x)) = J(x) = I(x)$.

First, we will show that J satisfies semantic condition 1 of Definition 11 (provenance RDFS interpretation), in a number of steps:
Step 1: Here, we prove that $Res_J = \{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Resource), S\rangle \in PT_J(J(type))\}$. Obviously, $\{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Resource), S\rangle \in PT_J(J(type))\}$ $\subseteq Res_J$. We will show that $Res_J \subseteq \{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Resource), S\rangle \in PT_J(J(type))\}$. Let $x \in Res_J$. We want to show that $\exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Resource), S\rangle$ $\in PT_J(J(type))$. It holds that: $\langle x, J(Resource), S\rangle \in PT_J(J(type))$ iff $\langle m(x), I(Resource),$ $S\rangle \in PT_I(I(type))$. Since I is a $\#n$-provenance RDFS interpretation of $V_{\mathcal{C}}^{\#n}$ and $\mathcal{C}$, it holds that $m(x) \in Res_I$. Thus, $\exists S$ s.t. $\langle m(x), I(Resource), S\rangle \in PT_I(I(type))$. Therefore, $x \in \{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Resource), S\rangle \in PT_J(J(type))\}$. Thus, $Res_J = \{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Resource), S\rangle \in PT_J(J(\ type))\}$.

Step 2: Here, we prove that $Prop_J = \{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Property), S\rangle \in PT_J(J(type))\}$. We will show that $Prop_J \subseteq \{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Property), S\rangle \in PT_J(J(type))\}$. Let $x \in Prop_J$. We want to show that $\exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Property),$ $S\rangle \in PT_J(J(type))$. It holds that: $\langle x, J(Property), S\rangle \in PT_J(J(type))$ iff $\langle m(x),$ $I(Property),\ S\rangle \in PT_I(I(type))$. Since $m(x) \in Prop_I$, it follows that $\exists S$ s.t. $\langle m(x),$ $I(Property), S\rangle \in PT_I(I(type))$. Thus, $x \in \{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Property), S\rangle \in PT_J(J(type))\}$. Therefore, $Prop_J \subseteq \{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Property), S\rangle \in PT_J(J(type))\}$.

We will now show that $\{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Property), S\rangle \in PT_J(J(type))\} \subseteq Prop_J$. Let $x \in \{x \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Property), S\rangle \in PT_J(J(type))\}$. It holds that $\exists S \subseteq \mathit{names}_{\mathcal{C}}$ s.t. $\langle x, J(\mathit{Property}), S\rangle \in PT_J(J(\mathit{type}))$, which implies that $\langle m(x), I(\mathit{Property}), S\rangle \in PT_I(I(\mathit{type}))$. Thus, $m(x) \in \mathit{Prop}_I$ and $x \in \mathit{Prop}_J$. Therefore, $\{x \mid \exists S \subseteq \mathit{names}_{\mathcal{C}}$ s.t. $\langle x, J(\mathit{Property}), S\rangle \in PT_J(J(\mathit{type}))\} \subseteq \mathit{Prop}_J$.

Thus, $\mathit{Prop}_J = \{x \mid \exists S \subseteq \mathit{names}_{\mathcal{C}}$ s.t. $\langle x, J(\mathit{Property}), S\rangle \in PT_J(J(\mathit{type}))\}$.

Step 3: By definition, it holds that $LV_J = \{x \mid \exists S \subseteq \mathit{names}_{\mathcal{C}}$ s.t. $\langle x, J(\mathit{Property}), S\rangle \in PT_J(J(\mathit{type}))\}$.

We will now show that J satisfies semantic condition 2 of Definition 11. Let $c \in \mathit{names}_{\mathcal{C}}$ and $x \in \mathit{Res}_J \cap J(V_c)$. We will show that $\langle x, J(\mathit{Resource}), \{c\}\rangle \in PT_J(J(\mathit{type}))$. It holds that $m(x) \in \mathit{Res}_I \cap I(V_c)$. Therefore, $\langle m(x), I(\mathit{Resource}), \{c\}\rangle \in PT_I(I(\mathit{type}))$. Thus, $\langle x, J(\mathit{Resource}), \{c\}\rangle \in PT_J(J(\mathit{type}))$.

Similarly, we can show that J satisfies semantic condition 3 of Definition 11.

We will now show that J satisfies semantic condition 4 of Definition 11. Let $\langle x, y, S\rangle \in PT_J(z)$. Then, $\langle m(x), m(y), S\rangle \in PT_I(m(z))$. Since I is a $\#n$-provenance RDFS interpretation, it holds that $\langle m(z), I(\mathit{Property}), S\rangle \in PT_I(I(\mathit{type}))$. Thus, $\langle z, J(\mathit{Property}), S\rangle \in PT_J(J(\mathit{type}))$.

We will now show that J satisfies semantic condition 5 of Definition 11. Let $\langle x, y, S_1\rangle \in PT_J(J(\mathit{domain}))$ and $\langle z, w, S_2\rangle \in PT_J(x)$ s.t. $\mathit{compatible}(S_1 \cup S_2)$. We will show that $\langle z, y, S_1 \cup S_2\rangle \in PT_J(J(\mathit{type}))$. It holds that $\langle m(x), m(y), S_1\rangle \in PT_I(I(\mathit{domain}))$ and $\langle m(z), m(w), S_2\rangle \in PT_I(m(x))$. Since I is a $\#n$-provenance RDFS interpretation, it holds that $\langle m(z), m(y), S_1 \cup S_2\rangle \in PT_I(I(\mathit{type}))$. Thus, $\langle z, y, S_1 \cup S_2\rangle \in PT_J(J(\mathit{type}))$.

In a similar manner, we can prove that J also satisfies the rest of the semantic conditions of Definition 11. Thus, J is a provenance RDFS interpretation of $V_{\mathcal{C}}$ and $\mathcal{C}$. Thus, $J \in \mathcal{I}^H(\mathcal{C})$.

Let $p(s, o) \in sk(G_i)$, for $i = 1, ..., m$. It holds that $p, s, o \in V_{\mathcal{C}}^{\#n}$. Since $I \models H(s, p, o, \{\mathit{Nam}_{O_i}\})$, it holds that $I(p) \in \mathit{Prop}_I$. Thus, $\exists S$ s.t. $\langle I(p), I(\mathit{Property}), S\rangle \in PT_I(I(\mathit{type}))$. Note that $m(J(p)) = J(p) = I(p)$, $m(J(\mathit{Property})) = J(\mathit{Property}) = I(\mathit{Property})$, and $m(J(\mathit{type})) = J(\mathit{type}) = I(\mathit{type})$. Therefore, $\langle J(p), J(\mathit{Property}), S\rangle \in PT_J(J(\mathit{type}))$ and thus, $J(p) \in \mathit{Prop}_J$. It holds that: $\langle J(s), J(o), \{\mathit{Nam}_{O_i}\}\rangle \in PT_J(J(p))$ iff $\langle m(J(s)), m(J(o)), \{\mathit{Nam}_{O_i}\}\rangle \in PT_I(m(J(p)))$ iff $\langle I(s), I(o), \{\mathit{Nam}_{O_i}\}\rangle \in PT_I(I(p))$. The last statement is true since $I \models H(s, p, o, \{\mathit{Nam}_{O_i}\})$. Thus, $J \models H(s, p, o, \{\mathit{Nam}_{O_i}\})$.

Let $r = H(s_0, p_0, o_0, S) \leftarrow H(s_1, p_1, o_1, S_1), ..., H(s_l, p_l, o_l, S_l), \neg H(s_{l+1}, p_{l+1}, o_{l+1}, S_{l+1}), ..., \neg H(s_k, p_k, o_k, S_k) \in [\mathit{prov}(P_i)]_{\mathcal{C}}$, for an $i = 1, ..., m$. Assume that $J \models H(s_i, p_i, o_i, S_i)$, for all $i = 1, ..., l$ and $J \models \neg H(s_i, p_i, o_i, S_i)$, for all $i = l+1, ..., k$. We will show that $J \models H(s_0, p_0, o_0, S)$. First, we define a total function $m' : V_{\mathcal{C}} \to V_{\mathcal{C}}^{\#n}$, as follows:

$$m'(x) = \begin{cases} x & \text{if } x \text{ is a well-typed XML literal} \\ m(x) & \text{otherwise} \end{cases}$$

Note that $p_i \in V_{\mathcal{C}}^{\#n}$, for $i = 0, ..., k$. Additionally, note that s_i, o_i are either terms in $V_{\mathcal{C}}^{\#n}$ or the instantiation of variables. Additionally, it holds that $\langle J(s_i), J(o_i), S_i\rangle \in PT_J(J(p_i))$, for $i = 1, ..., l$, and $\langle J(s_i), J(o_i), S_i\rangle \in PF_J(J(p_i))$, for $i = l+1, ..., k$. Thus, $\langle m(J(s_i)), m(J(o_i)), S_i\rangle \in PT_I(m(J(p_i)))$, for $i = 1, ..., l$, and $\langle m(J(s_i)), m(J(o_i)), S_i\rangle \in PF_I(m(J(p_i)))$, for $i = l+1, ..., k$. Note that $I(m'(s_i)) = m(J(s_i))$, $I(m'(o_i))) = m(J(o_i))$, and $I(m'(p_i))) = m(J(p_i))$. Therefore, $\langle I(m'(s_i)), I(m'(o_i)), S_i\rangle \in PT_I(I(m'(p_i)))$, for $i = 1, ..., l$ and $\langle I(m'(s_i)), I(m'(o_i)), S_i\rangle \in PF_I(I(m'(p_i)))$, for $i = l+1, ..., k$. Therefore, $I \models H(m'(s_i), m'(p_i), m'(o_i), S_i)$, for $i = 1, ..., l$, and $I \models \neg H(m'(s_i), m'(p_i), m'(o_i), S_i)$, for $i = l+1, ..., k$. This implies that $I \models H(m'(s_0), m'(p_0), m'(o_0), S_i)$. Thus, $\langle I(m'(s_0)), I(m'(o_0)), S\rangle \in PT_I(m'(p_0))$. Therefore, $\langle m(J(s_0)), m(J(o_0)), S\rangle \in PT_I(m(J(p_0)))$. Thus, $\langle J(s_0), J(o_0), S\rangle \in PT_J(J(p_0))$. It follows from this that $J \models H(s_0, p_0, o_0, S)$.

Let $r = \neg H(s_0, p_0, o_0, S) \leftarrow H(s_1, p_1, o_1, S_1), ..., H(s_l, p_l, o_l, S_l), \neg H(s_{l+1}, p_{l+1}, o_{l+1}, S_{l+1}), ..., \neg H(s_k, p_k, o_k, S_k) \in [\mathit{prov}(P_i)]_{\mathcal{C}}$, for an $i = 1, ..., m$. Assume that $J \models H(s_i, p_i,$

$o_i, S_i)$, for all $i = 1, ..., l$ and $J \models \neg H(s_i, p_i, o_i, S_i)$, for all $i = l+1, ..., k$. Similarly, we can show that $J \models \neg H(s_0, p_0, o_0, S)$.

Let:

1. $K_0 = \mathit{minimal}(\{I \in \mathcal{I}^H(\mathcal{C}) \mid I \leq J$ and $I \models [\neg]H(s, p, o, \{\mathit{Nam}_O\})$, for each $[\neg]p(s, o)$ appearing in the skolemization of a g-RDF graph G_O in $\mathcal{C}\})$.
2. For natural numbers α with $0 < \alpha$:
$K_\alpha = \mathit{minimal}(\{I \in \mathcal{I}^H(\mathcal{C}) \mid I \leq J,\ I \geq I_{\alpha-1}$ and for all $q_0 \leftarrow q_1, ..., q_l, \sim q_{l+1}, ..., \sim q_k \in [\mathit{prov}(P)]_\mathcal{C}$, for P be a g-RDF program appearing in $\mathcal{C}$, it holds that if (i) $I_{\alpha-1} \models q_i$, for $i = 1, ..., l$ and (ii) $M \models \sim q_i$, for $i = l+1, ..., k$ then $I \models q_0\})$.

Note that K_0 and K_α are well defined and it exists a λ s.t. $K_\lambda = K_{\lambda+1}$. Let $K = K_\lambda$. Since $\mathcal{C}$ does not contain $\sim$, it follows that:

1. $K_0 = \mathit{minimal}(\{I \in \mathcal{I}^H(\mathcal{C}) \mid I \models [\neg]H(s, p, o, \{\mathit{Nam}_O\})$, for each $[\neg]p(s, o)$ appearing in the skolemization of a g-RDF graph G_O in $\mathcal{C}\})$.
2. For natural numbers α with $0 < \alpha$:
$K_\alpha = \mathit{minimal}(\{I \in \mathcal{I}^H(\mathcal{C}) \mid I \geq I_{\alpha-1}$ and for all $q_0 \leftarrow q_1, ..., q_l, \sim q_{l+1}, ..., \sim q_k \in [\mathit{prov}(P)]_\mathcal{C}$, for P be a g-RDF program appearing in $\mathcal{C}$, it holds that if (i) $I_{\alpha-1} \models q_i$, for $i = 1, ..., l$ and (ii) $M \models \sim q_i$, for $i = l+1, ..., k$ then $I \models q_0\})$.

Thus, $K \in \mathcal{M}(\mathcal{C})$. Therefore, $K \models G$. Thus, there exists $u : \mathit{Var}(G) \to \mathit{Res}^H_\mathcal{C}$ s.t. $K, u \models G$. We will show that $J, u \models G$. Let $H(s, p, o, S) \in G$. Then, $p \in (V_\mathcal{C} \cap \mathit{URI}) \cup \mathit{Var}$, $s, o \in V_G \cup \mathit{Var}$, and $S \subseteq \mathit{names}_\mathcal{C}$. It holds that $[J+u](p) = [K+u](p) \in \mathit{Prop}_K \subseteq \mathit{Prop}_J$. Additionally, $\langle [K+u](s), [K+u](o), S\rangle \in PT_K([K+u](p))$. Since $\langle [K+u](s), [K+u](o), S\rangle = \langle [J+u](s), [J+u](o), S\rangle$ and $PT_K([K+u](p)) \subseteq PT_J([J+u](p))$, it follows that $\langle [J+u](s), [J+u](o), S\rangle \in PT_J([J+u](p))$. Thus, $J, u \models H(s, p, o, S)$. Similarly, let $\neg H(s, p, o, S) \in G$. We can show that $J, u \models \neg H(s, p, o, S)$.

Now, we define a total function $u' : \mathit{Var}(G) \to \mathit{Res}_I$ s.t. $u'(x) = m(u(x))$. We will show that $I, u' \models G$.

Let $H(s, p, o, S) \in G$. Then, $p \in (V_G \cap \mathit{URI}) \cup \mathit{Var}$, $s, o \in V_G \cup \mathit{Var}$, and $S \subseteq \mathit{names}_\mathcal{C}$. Since $\mathit{max}(\{i \in I\!N \mid \mathit{rdf}{:}\_i \in V_G\}) \leq n$, it follows that $V_G \subseteq V_\mathcal{C}^{\#n}$. Thus, $p \in (V_\mathcal{C}^{\#n} \cap \mathit{URI}) \cup \mathit{Var}$ and $s, o \in V_\mathcal{C}^{\#n} \cup \mathit{Var}$.

Note that $\forall x \in V_G$, it holds that: $I(x) = m(J(x))$ and $[I+u'](x) = m([J+u](x))$. Moreover, $\forall x \in \mathit{Var}(G)$, it holds that: $[I+u'](x) = m(u(x)) = m([J+u](x))$.

We will now show that $[I+u'](p) \in \mathit{Prop}_I$. It holds that:
$\langle [I+u'](p), I(\mathit{Property}), S\rangle \in PT_I(I(\mathit{type}))$ iff
$\langle m([J+u](p)), J(\mathit{Property}), S\rangle \in PT_I(J(\mathit{type}))$ iff
$\langle [J+u](p), J(\mathit{Property}), S\rangle \in PT_J(J(\mathit{type}))$. Since $J, u \models G$, it holds that $[J+u](p) \in \mathit{Prop}_J$. Therefore, it exists $S \subseteq \mathit{names}_\mathcal{C}$ s.t. $\langle [J+u](p), J(\mathit{Property}), S\rangle \in PT_J(J(\mathit{type}))$. Thus, $\exists S \subseteq \mathit{names}_\mathcal{C}$ s.t. $\langle [I+u'](p), I(\mathit{Property}), S\rangle \in PT_I(I(\mathit{type}))$. Therefore, $[I+u'](p) \in \mathit{Prop}_I$.

We want to show that $\langle [I+u'](s), [I+u'](o), S\rangle \in PT_I([I+u'](p))$. It holds that:
$\langle [I+u'](s), [I+u'](o), S\rangle \in PT_I([I+u'](p))$ iff
$\langle m([J+u](s)), m([J+u](o)), S\rangle \in PT_I(m([J+u](p)))$ iff
$\langle [J+u](s), [J+u](o), S\rangle \in PT_J([J+u](p))$, which is true since $J, u \models G$. Thus, $I, u' \models H(s, p, o, S)$.

Similarly, let $\neg H(s, p, o, S) \in G$. We can show that $I, u' \models \neg H(s, p, o, S)$.

Thus, $I, u' \models G$, which implies that $I \models G$. Therefore, $\mathcal{C} \models^{pr\#n} G$.

$\Leftarrow$) Let $\mathcal{C} \models^{pr\#n} G$. We will show that $\mathcal{C} \models^{pr} G$. Let $I \in \mathcal{M}(\mathcal{C})$. We will show that $I \models G$.

Note that I is a provenance RDFS interpretation of $V_\mathcal{C}$ and $\mathcal{C}$. Based on I, we construct a provenance simple interpretation J of $V_\mathcal{C}^{\#n}$ and $\mathcal{C}$ as follows:

- $Res_J = Res_{\mathcal{C}}^{H_{\#n}}$.
- $J_V(x) = x$, for all $x \in V_{\mathcal{C}}^{\#n} \cap URI$.
- We define the mapping: $IL_J : V_{\mathcal{C}}^{\#n} \cap \mathcal{TL} \to Res_J$ such that:
  $IL_J(x) = x$, if x is a typed literal in $V_{\mathcal{C}}^{\#n}$ other than a well-typed XML literal, and $IL_I(x)$ is the XML value of x, if x is a well-typed XML literal in $V_{\mathcal{C}}^{\#n}$.
- We define the mapping: $J : V_{\mathcal{C}}^{\#n} \to Res_J$ such that:
  - $J(x) = J_V(x),\ \forall x \in V_{\mathcal{C}}^{\#n} \cap URI$.
  - $J(x) = x,\ \forall\, x \in V_{\mathcal{C}}^{\#n} \cap \mathcal{PL}$.
  - $J(x) = IL_J(x),\ \forall\, x \in V_{\mathcal{C}}^{\#n} \cap \mathcal{TL}$.
- $Prop_J = Prop_I \cap Res_J$.
- $\forall x \in Prop_J, \quad PT_J(x) = PT_I(x) \cap (Res_J \times Res_J \times \mathcal{P}(names_{\mathcal{C}}))$.
- $\forall x \in Prop_J, \quad PF_J(x) = PF_I(x) \cap (Res_J \times Res_J \times \mathcal{P}(names_{\mathcal{C}}))$.
- $LV_J = \{x \in Res_J \mid \exists S \subseteq names_{\mathcal{C}}$ s.t. $\langle x, J(Literal), S\rangle \in PT_J(J(type))\}$.

As in the proof of $\Rightarrow$), we can show that: (i) J is a $\#n$-provenance RDFS interpretation of $V_{\mathcal{C}}^{\#n}$ and $\mathcal{C}$, (ii) $J \models G$, and (iii) $I \models G$. □

Proposition 5. Let $\mathcal{C}$ be a g-RDF information base. Let M be a $\#n$-provenance semi-Herbrand interpretation of $\mathcal{C}$. It is the case that: $M \in \mathcal{M}^{\#n}(\mathcal{C})$ iff $ELP(M)$ is a consistent answer set of $P_{\mathcal{C}}^{\#n}$.

Proof:

$\Rightarrow$) Let $M \in \mathcal{M}^{\#n}(\mathcal{C})$. We will show that $N = ELP(M)$ is a consistent answer set of $P_{\mathcal{C}}^{\#n}$. Since $M \in \mathcal{M}^{\#n}(O)$, it follows that there is a sequence of $\#n$-provenance Herbrand interpretations $I_0 \leq ... \leq I_{m+1}$ such that $I_m = I_{m+1} = M$ and:

1. $I_0 = minimal(\{I \in \mathcal{I}^{H_{\#n}}(\mathcal{C}) \mid I \models [\neg]H(s,p,o,\{Nam_O\})$, for each $[\neg]p(s,o)$ appearing in the skolemization of a g-RDF graph G_O in $\mathcal{C}\})$.
2. For natural numbers α with $0 < \alpha \leq m+1$:
   $I_\alpha = minimal(\{I \in \mathcal{I}^{H_{\#n}}(\mathcal{C}) \mid I \geq I_{\alpha-1}$ and for all $q_0 \leftarrow q_1, ..., q_l, \sim q_{l+1}, ..., \sim q_k \in [prov(P)]_{\mathcal{C}}^{\#n}$, for P be a g-RDF program appearing in $\mathcal{C}$, it holds that if (i) $I_{\alpha-1} \models q_i$, for $i = 1, ..., l$ and (ii) $M \models \sim q_i$, for $i = l+1, ..., k$ then $I \models q_0\})$.

Let $P^{\mathtt{G}}$ be the rules of $P_{\mathcal{C}}^{\#n}$ that are generated from the rules in $\Pi_{\mathcal{C}}^{\mathtt{G}}$. Let $P^{\mathtt{P}}$ be the rules of $P_{\mathcal{C}}^{\#n}$ that are generated from the rules in $\Pi_{\mathcal{C}}^{\mathtt{P}}$. Let $P^{\mathtt{H}}$ be the rules of $P_{\mathcal{C}}^{\#n}$ that are generated from the rules in $\Pi_{\mathcal{C}}^{\mathtt{H}_{\#n}}$.

Now, we define a sequence $N_0 \subseteq .... \subseteq N_{m+1} \subseteq EHB(P_{\mathcal{C}}^{\#n})$, as follows:

$N_0 = T_{P^{\mathtt{H}}}^{\uparrow\omega}(T_{P^{\mathtt{G}}}(\emptyset))$.

$N_\alpha = T_{P^{\mathtt{H}}}^{\uparrow\omega}(T_{(P^{\mathtt{P}})^N}(N_{\alpha-1}))$, where $1 \leq \alpha \leq m+1$.

Lemma: It holds $N_\alpha = ELP(I_\alpha)$, for $\alpha = 0, ..., m+1$.
Proof: We will prove the Lemma, by induction.

First, we will show that $N_0 \subseteq ELP(I_0)$. Since $I_0 \models [\neg]H(s,p,o,\{Nam_O\})$, for each $[\neg]p(s,o)$ appearing in the skolemization of a g-RDF graph G_O in $\mathcal{C}$, it follows that $T_{P^{\mathtt{G}}}(\emptyset) \subseteq ELP(I_0)$. As $I_0 \in \mathcal{I}^{H_{\#n}}(\mathcal{C})$, it follows that $ELP(I_0)$ satisfies all rules in $P^{\mathtt{H}}$. Now, as $T_{P^{\mathtt{G}}}(\emptyset) \subseteq ELP(I_0)$, it follows that $T_{P^{\mathtt{H}}}^{\uparrow\omega}(T_{P^{\mathtt{G}}}(\emptyset)) \subseteq ELP(I_0)$. Therefore, $N_0 \subseteq ELP(I_0)$.

By definition, N_0 satisfies all rules in $P^{\mathtt{H}}$. Therefore, $ELP^{-1}(N_0)$ satisfies all semantic conditions of a $\#n$-provenance Herbrand interpretation of $\mathcal{C}$. Thus, $ELP^{-1}(N_0) \in \mathcal{I}^{H_{\#n}}(\mathcal{C})$. Moreover, $ELP^{-1}(N_0) \models [\neg]H(s,p,o,\{Nam_O\})$, for each $[\neg]p(s,o)$ appearing in the skolemization of a g-RDF graph G_O in $\mathcal{C}$. Therefore, $ELP^{-1}(N_0) \in \{I \in$

$\mathcal{I}^{H_{\#n}}(\mathcal{C}) \mid I \models [\neg]H(s,p,o,\{Nam_O\})$, for each $[\neg]p(s,o)$ appearing in the skolemization of a g-RDF graph G_O in $\mathcal{C}\}$. Now, as $I_0 = minimal(\{I \in \mathcal{I}^{H_{\#n}}(\mathcal{C}) \mid I \models [\neg]H(s,p,o,\{Nam_O\})$, for each $[\neg]p(s,o)$ appearing in the skolemization of a g-RDF graph G_O in $\mathcal{C}\})$ and $N_0 \subseteq ELP(I_0)$, it follows that $N_0 = ELP(I_0)$.

Assumption: We assume that $N_{\alpha-1} = ELP(I_{\alpha-1})$, for an $\alpha \leq m$.

We will show that $N_\alpha = ELP(I_\alpha)$. First, we will show that $N_\alpha \subseteq ELP(I_\alpha)$. Due to assumption $N_{\alpha-1} = ELP(I_{\alpha-1})$ and the fact $I_{\alpha-1} \leq I_\alpha$, it follows that $N_{\alpha-1} \subseteq ELP(I_\alpha)$. Note that if (i) $q_0 \leftarrow q_1,...,q_l,\sim q_{l+1},...,\sim q_k \in [prov(P)]^{\#n}_{\mathcal{C}}$, for P being a g-RDF program appearing in $\mathcal{C}$, (ii) $I_{\alpha-1} \models q_i$, for $i = 1,...,l$, and (iii) $M \models \sim q_i$, for $i = l+1,...,k$, then $I_\alpha \models q_0$. Therefore, $T_{(P^{\mathtt{P}})^N}(N_{\alpha-1}) \subseteq ELP(I_\alpha)$.

As $I_\alpha \in \mathcal{I}^{H_{\#n}}(\mathcal{C})$, it follows that $ELP(I_\alpha)$ satisfies all rules in $P^{\mathtt{H}}$. Now, as $T_{(P^{\mathtt{P}})^N}(N_{\alpha-1}) \subseteq ELP(I_\alpha)$, it follows that $T^{\uparrow\omega}_{P^{\mathtt{H}}}(T_{(P^{\mathtt{P}})^N}(N_{\alpha-1})) \subseteq ELP(I_\alpha)$. Therefore, $N_\alpha \subseteq ELP(I_\alpha)$.

By definition, N_α satisfies all rules in $P^{\mathtt{H}}$. Therefore, $ELP^{-1}(N_\alpha)$ satisfies all semantic conditions of a $\#n$-provenance Herbrand interpretation of $\mathcal{C}$. Thus, $ELP^{-1}(N_\alpha) \in \mathcal{I}^{H_{\#n}}(\mathcal{C})$. Moreover, $ELP^{-1}(N_\alpha) \geq ELP^{-1}(N_{\alpha-1}) = I_{\alpha-1}$. Now, based on the assumption that $N_{\alpha-1} = ELP(I_{\alpha-1}) \subseteq N$ and the fact that $T_{(P^{\mathtt{P}})^N}(N_{\alpha-1}) \subseteq N_\alpha$, it follows that if (i) $q_0 \leftarrow q_1,...,q_l,\sim q_{l+1},...,\sim q_k \in [prov(P)]^{\#n}_{\mathcal{C}}$, for P be a g-RDF program appearing in $\mathcal{C}$, (ii) $ELP^{-1}(N_{\alpha-1}) \models q_i$, for $i = 1,...,l$, and (iii) $M \models \sim q_i$, for $i = l+1,...,k$, then $ELP^{-1}(N_\alpha) \models q_0$. Therefore, $ELP^{-1}(N_\alpha) \in \{I \in \mathcal{I}^{H_{\#n}}(\mathcal{C}) \mid I \geq I_{\alpha-1}$ and for all $q_0 \leftarrow q_1,...,q_l,\sim q_{l+1},...,\sim q_k \in [prov(P)]^{\#n}_{\mathcal{C}}$, for P be a g-RDF program appearing in $\mathcal{C}$, it holds that if (i) $I_{\alpha-1} \models q_i$, for $i = 1,...,l$ and (ii) $M \models \sim q_i$, for $i = l+1,...,k$ then $I \models q_0\}$). Now, as $I_\alpha = minimal(\{I \in \mathcal{I}^{H_{\#n}}(\mathcal{C}) \mid I \geq I_{\alpha-1}$ and for all $q_0 \leftarrow q_1,...,q_l,\sim q_{l+1},...,\sim q_k \in [prov(P)]^{\#n}_{\mathcal{C}}$, for P be a g-RDF program appearing in $\mathcal{C}$, it holds that if (i) $I_{\alpha-1} \models q_i$, for $i = 1,...,l$ and (ii) $M \models \sim q_i$, for $i = l+1,...,k$ then $I \models q_0\})$ and $N_\alpha \subseteq ELP(I_\alpha)$, it follows that $N_\alpha = ELP(I_\alpha)$.
End of Lemma

Therefore, $N_m = N_{m+1} = ELP(M)$. Since M is a $\#n$-provenance Herbrand interpretation, it follows that $N = ELP(M)$ is consistent. Moreover, since $(P^{\#n}_{\mathcal{C}})^N = P^{\mathtt{G}} \cup (P^{\mathtt{P}})^N \cup P^{\mathtt{H}}$, it follows that $T^{\uparrow\omega}_{(P^{\#n}_{\mathcal{C}})^N}(\emptyset) = N$. Therefore, $N = ELP(M)$ is a consistent answer set of $P^{\#n}_{\mathcal{C}}$.

$\Leftarrow$) Let $N = ELP(M)$ be a consistent answer set of $P^{\#n}_{\mathcal{C}}$. We will show that $M \in \mathcal{M}^{\#n}(\mathcal{C})$. Since N is a consistent answer set of $P^{\#n}_{\mathcal{C}}$, it follows that $T^{\uparrow\omega}_{(P^{\#n}_{\mathcal{C}})^N} = N$. We define a sequence $N_\alpha \subseteq EHB(P^{\#n}_{\mathcal{C}})$, $\alpha \in \{0,1,...\}$, as follows:

$N_0 = T^{\uparrow\omega}_{P^{\mathtt{H}}}(T_{P^{\mathtt{G}}}(\emptyset))$.

$N_\alpha = T^{\uparrow\omega}_{P^{\mathtt{H}}}(T_{(P^{\mathtt{P}})^N}(N_{\alpha-1}))$, where $1 \leq \alpha$.

Since $EHB(P^{\#n}_{\mathcal{C}})$ is a finite set and $(P^{\#n}_{\mathcal{C}})^N = P^{\mathtt{G}} \cup (P^{\mathtt{P}})^N \cup P^{\mathtt{H}}$, it follows that there is $m \in \{0,1,...\}$ such that $N_m = N_{m+1} = N$.

Note that N_0 is the smallest subset of $EHB(P^{\#n}_{\mathcal{C}})$ that satisfies all rules in $P^{\mathtt{G}} \cup P^{\mathtt{H}}$. Thus, $ELP^{-1}(N_0) = minimal(\{I \in \mathcal{I}^{H_{\#n}}(\mathcal{C}) \mid I \models [\neg]H(s,p,o,\{Nam_O\})$, for each $[\neg]p(s,o)$ appearing in the skolemization of a g-RDF graph G_O in $\mathcal{C}\})$.

Let α such that $1 \leq \alpha \leq m{+}1$. Note that N_α is the smallest subset of $EHB(\Pi^{\#n}_O)$ such that $N_\alpha \supseteq N_{\alpha-1}$, (ii) satisfies all rules in $P^{\mathtt{H}}$, and (iii) $Head(r) \in N_\alpha$, for all $r \in (P^{\mathtt{P}})^N$ such that $Body(r) \subseteq N_{\alpha-1}$. Therefore, $ELP^{-1}(N_\alpha) = minimal(\{I \in \mathcal{I}^{H_{\#n}}(\mathcal{C}) \mid I \geq I_{\alpha-1}$

and for all $q_0 \leftarrow q_1, ..., q_l, {\sim}q_{l+1}, ..., {\sim}q_k \in [prov(P)]_{\mathcal{C}}^{\#n}$, for P be a g-RDF program appearing in $\mathcal{C}$, it holds that if (i) $ELP^{-1}(N_{\alpha-1}) \models q_i$, for $i = 1, ..., l$ and (ii) $M \models {\sim}q_i$, for $i = l+1, ..., k$ then $I \models q_0\}$).

Now as $ELP^{-1}(N_0) \leq ... \leq ELP^{-1}(N_{m+1})$ and $ELP^{-1}(N_m) = ELP^{-1}(N_{m+1}) = ELP^{-1}(N) = M$, it follows that $M \in \mathcal{M}^{\#n}(\mathcal{C})$. □

Proposition 9. Let $\mathcal{C} = \langle \{\langle Nam_{O_i}, O_i\rangle \mid i = 1, ..., m\}, \mathit{conflict}\rangle$ be a g-RDF information base. Deciding if $\mathcal{C}$ has a $\#n$-provenance stable model:

1. $\mathrm{NP}^{\mathrm{NP}}$-complete w.r.t. the size of $\mathcal{C}$ and
2. NP-complete w.r.t. $graph\_size(\mathcal{C}) = size(G_1) + ... + size(G_m)$.

Proof:
1)
Hardness) Let O be a simple ERDF ontology and $\mathcal{C}$ be the g-RDF information base generated from O as defined in Proposition 7. Note that to generate $\mathcal{C}$ from O, it takes polynomial time w.r.t. the size of O. Now, from Proposition 7, and since the satisfiability problem of a simple ERDF ontology under the $\#n$-stable model semantics is an $\mathrm{NP}^{\mathrm{NP}}$-complete problem (see Proposition 13(1) of [3]), it follows that the problem of establishing whether $\mathcal{C}$ has a $\#n$-provenance stable model is $\Sigma_2^P = \mathrm{NP}^{\mathrm{NP}}$-hard w.r.t. the size of $\mathcal{C}$.

Membership)
Let $P^{\mathtt{RDFS}}$ be the program derived from $\Pi_{\mathcal{C}}^{\mathtt{H}\#n}$ by removing the rules (17), (18), and (19) and replacing $H(s, p, o, S)$ by $p(s, o)$, in the rest of the rules. Let $N \subseteq names_{\mathcal{C}}$. We define $P_N = \{[\neg]p(s, o) \leftarrow \ \mid [\neg]p(s, o)$ appears in the scolemization of a g-RDF graph appearing in the g-RDF ontologies whose name belongs to $N\} \cup \bigcup\{P \mid P$ is a g-RDF program appearing in the g-RDF ontologies whose name belongs to $N\} \cup P^{\mathtt{RDFS}}$. Collect now all the scoped weakly negated g-RDF triples appearing in $\mathcal{C}$ into a set W'. Ground these scoped weakly negated g-RDF triples in W using the vocabulary $V_{\mathcal{C}}$ and placing a URI at the property position, creating the set W. Now guess (Guess 1) a partition of W into the two sets W_+ and W_-, after removing the weak negation s.t. (i) if $p(s, o)@N \in W_+$, there is no $p(s, o)@N' \in W_-$ s.t. $N \subseteq N'$ and (ii) if $p(s, o)@N \in W_-$, there is no $p(s, o)@N' \in W_+$ s.t. $N \supseteq N'$. Now, for each $N \subseteq names_{\mathcal{C}}$ appearing in W_-, guess (Guess 2) a subset S_N of the $p(s, o)$ appearing in P_N, grounded over $V_{\mathcal{C}}$, s.t. $\{p(s, o) \mid p(s, o)@N \in W_+\} \subseteq S_N$ and $\{p(s, o) \mid p(s, o)@N \in W_-\} \cap S_N = \emptyset$. Note that both Guess 1 and Guess 2 can be done in one step. Our goal is to see if these can lead to the derivation of a $\#n$-provenance stable model for $\mathcal{C}$. To achieve this, 3 checks have to be made.

Check 1: If $r \in P_N$ then for any substitution θ of the variables of r over the vocabulary $V_{\mathcal{C}}$ and placing a URI at the property position of the g-RDF triples, it should hold that: If $Body^+(r\theta) \subseteq S_N$ and $Body^-(r\theta) \cap W_- = \emptyset$ then $Head(r) \in S_N$.

Check 2: If $L \in S_N$ then there is a *well-founded proof* of L w.r.t. P_N and W_-, defined as follows: there is a sequence of grounded rules $r_i\theta_i$, $i = 1, ..., k$, where $r_i \in P_N$ s.t. $Head(r_k\theta_k) = L$ and if $L_i = Head(r_i\theta_i)$, it should hold that $Body^+(r_i\theta_i) \subseteq \{L_1, ..., L_{i-1}\}$ and $Body^-(r_i\theta_i) \cap W_- = \emptyset$, for $i = 1, ..., k$.

Check 3: For all $N' \in maximal_{\subseteq}(\{N \subseteq names_{\mathcal{C}} \mid$ there are no $c, c' \in N$ s.t. $\mathit{conflict}(c, c')\})$, there is no well-founded proof of *false* w.r.t. $P_{N'} \cup \{\mathit{false} \leftarrow ?p(?s, ?o), \neg?p(?s, ?o).\}$ and W_-.

To compute the complexity of the complement of Check 1, guess an $r \in P_N$ and substitution θ and check if $Body^+(r\theta) \subseteq S_N$ and $Body^-(r\theta) \cap W_- = \emptyset$, while $Head(r) \notin S_N$. The complexity of the complement of Check 1 is in NP. Thus, the complexity of Check 1 is in co-NP.

To compute the complexity of Check 2, for each $L \in S_N$, guess a sequence of grounded rules $r_i\theta_i$, $i = 1, ..., n$, where $r_i \in P_N$ and check if it is indeed a well-founded proof of L w.r.t. P_N and W_- . Since S_N is polynomial w.r.t. the size of C and the size of each well-founded proof along with the time for its checks is polynomial w.r.t. the size of C, the complexity of Check 2 is in P^{NP}.

To compute the complexity of the complement of Step 3. Guess an $N' \subseteq \mathit{names}_{\mathcal{C}}$ and a sequence of grounded rules $r_i\theta_i$, $i = 1, ..., k$, where $r_i \in P_{N'} \cup \{\mathit{false} \leftarrow ?p(?s, ?o), \neg ?p(?s, ?o).\}$ and check (i) if $N' \in \mathit{maximal}_{\subseteq}(\{N \subseteq \mathit{names}_{\mathcal{C}} \mid$ these are no $c, c' \in N$ s.t. $\mathit{conflict}(c, c')\})$ and (i) $r_i\theta_i$, $i = 1, ..., k$ is a well-founded proof of *false* w.r.t. $P_{N'} \cup \{\mathit{false} \leftarrow ?p(?s, ?o), \neg ?p(?s, ?o).\}$ and W_-. Both (i) and (ii) can be checked in polynomial time w.r.t. the size of $\mathcal{C}$. Therefore, the complexity of the complement of Check 3 is in NP. Thus, the complexity of Check 3 is in co-NP.

Now if Checks 1,2, and 3 hold then based on W_-, we can build a $\#n$-provenance interpretation of $\mathcal{C}$ as follows: For each $N \subseteq \mathit{names}_{\mathcal{C}}$ s.t. these are no $c, c' \in N$ with $\mathit{conflict}(c, c')$ holding, take the grounding of all rules in P_N over the vocabulary $V_{\mathcal{C}}$ (ensuring a URI at the property position of g-RDF triples) resulting in a program $[P_N]$. Then, for each $r \in [P_N]$ if (i) $\mathit{Body}^-(r) \subseteq W_-$ then remove $\mathit{Body}^-(r)$ from the rule and (ii) otherwise, remove the rule r from $[P_N]$. Afterwards, apply the remaining rules in $[P_N]$, until a fixpoint D_N is reached. Consider now the set of ELP literals $\{H(s, p, o, f_{\mathcal{C}}(N)) \mid p(s, o) \in D_N\}$. Taking now the union of all these ELP literals and applying the function ELP^{-1}, we get a $\#n$-provenance stable model of $\mathcal{C}$.

Therefore, the problem of establishing whether $\mathcal{C}$ has a $\#n$-provenance stable model is in $\Sigma_2^P = \mathrm{NP}^{\mathrm{NP}}$ w.r.t. the size of $\mathcal{C}$, since an P^{NP} oracle and two co-NP oracles are called[31].

2)
Hardness) Let $D = (V, E)$ be a graph. Note that $\mathcal{C}_D = \langle\{\langle c, O_D\rangle\}, \emptyset\rangle$ is a g-RDF information base containing a single g-RDF ontology $O_D = \langle G_D, P_D\rangle$ and the size of P_D is fixed. Additionally, note that to generate O_D from D, it takes polynomial time w.r.t. the size of D. Now, from Proposition 8, and since *3-colorability* is an NP-complete problem, it follows that the problem of establishing whether O has a $\#n$-provenance stable model is NP-hard w.r.t. $\mathit{graph\_size}(\mathcal{C})$.

Membership) Let k be the maximum number of g-RDF triples in the body of a rule in $\mathcal{C}$. Additionally, let k' be the maximum number of scoped weakly negated g-RDF triples in the body of a rule in $\mathcal{C}$. Let l be the number of g-RDF rules appearing in C. The time complexity of taking the union of k sets of at most m elements is in $O(k^2 * m^2)$ and the time complexity of checking $\mathit{compatible}(S)$, for $S \subseteq \mathit{names}_C$, is in $O(m^3)$. Additionally, the time complexity for checking $S \subseteq S'$, where $S, S' \subseteq \mathit{names}_C$, is in $O(m^2)$. Thus, the time complexity for generating $\Pi^{\mathrm{p}}_{\mathcal{C}}$ is in $O(l * (|V^{\#n}_{\mathcal{C}}|^3 * 2^m)^{k+1} * |V^{\#n}_{\mathcal{C}}|^{3*k'})$. Therefore, the time complexity for generating $P^{\#n}_{\mathcal{C}}$ is in $O(l * (|V^{\#n}_{\mathcal{C}}|^3 * 2^m)^{k+1} * |V^{\#n}_{\mathcal{C}}|^{3*k'})$. Similarly, the size of $P^{\#n}_{\mathcal{C}}$ is in $O(l * (|V^{\#n}_{\mathcal{C}}|^3 * 2^m)^{k+1} * |V^{\#n}_{\mathcal{C}}|^{3*k'})$. Now guess a $\#n$-provenance semi-Herbrand interpretation of $\mathcal{C}$. Checking if $ELP(M)$ is an consistent answer set of $P^{\#n}_{\mathcal{C}}$, it takes polynomial time w.r.t. the size of $P^{\#n}_{\mathcal{C}}$. Thus, deciding if $\mathcal{C}$ has a $\#n$-provenance stable model is in NP w.r.t. $\mathit{graph\_size}(\mathcal{C})$, since k, k', and m are fixed.

Proposition 10. Let $\mathcal{C} = \langle\{\langle \mathit{Nam}_{O_i}, O_i\rangle \mid i = 1, ..., m\}, \mathit{conflict}\rangle$ be a g-RDF information base and let $CQ = SQ_1 \wedge ... \wedge SQ_k$, where each $SQ_i = \langle[\sim][\neg]H(s_i, p_i, o_i, ?S_i), L_i^{R_i}\rangle$ is a simple provenance query of $\mathcal{C}$. Let v be a mapping (i) from the variables in s_i, p_i, o_i to

[31] Note that any co-NP oracle can be changed to NP oracle by changing its reply "yes" to "no" and its reply "no" to "yes". Additionally, $\mathrm{NP}^{\mathrm{P}^{\mathrm{NP}}} = \mathrm{NP}^{\mathrm{NP}}$

$V_{\mathcal{C}}^{\#n}$ and (ii) from $?S_i$ to $\mathcal{P}(names_{\mathcal{C}})$. The time complexity of checking if $v \in Answers_{\mathcal{C}}^{\#n}(CQ)$ is:

1. co-NP$^{\text{NP}}$-complete w.r.t. the size of $\mathcal{C}$ and
2. co-NP-complete w.r.t. $graph\_size(\mathcal{C}) = size(G_1) + ... + size(G_m)$.

Proof:
1)
Hardness) Let $CQ = \langle H(s,p,o,?S), \emptyset^{\supseteq}\rangle \wedge \langle \neg H(s,p,o,?S), \emptyset^{\supseteq}\rangle$. Then, $v \in Answers_{\mathcal{C}}^{\#n}(CQ)$ iff $\mathcal{C}$ has no $\#n$-provenance stable model. It follows from Proposition 9(1) that deciding if $\mathcal{C}$ has a $\#n$-provenance stable model is NP$^{\text{NP}}$-hard w.r.t. the size of $\mathcal{C}$. Therefore, the complexity of checking if $v \in Answers_{\mathcal{C}}^{\#n}(CQ)$ is co-NP$^{\text{NP}}$-hard w.r.t. the size of $\mathcal{C}$.

Membership) Assume that we want to verify if $v \notin Answers_{\mathcal{C}}^{\#n}(CQ)$. Make Guesses 1 and 2 as in the proof of the membership part of Proposition 9(1) and see if these can lead to the derivation of a $\#n$-provenance stable model M of $\mathcal{C}$. This can be done in P$^{\text{NP}}$ time. Now, test if it does not hold $M \models [\sim][\neg]v(H(s_i,p_i,\, o_i,?S_i))$ and $L_i\ R_i\ v(?S_i)$, for $i = 1,...,k$. We will check the complement problem, that is if $M \models [\sim][\neg]v(H(s_i,p_i,$ $o_i,?S_i))$ and $L_i\ R_i\ v(?S_i)$, for $i = 1,...,k$.

It holds that $M \models [\neg]v(H(s_i,p_i,\, o_i,?S_i))$ and $L_i\ R_i\ v(?S_i)$ if (i) there are no c, c' in $v(?S_i)$ s.t. *conflict*(c,c') holds, (ii) there is a well-founded proof of $[\neg]v(p_i)(v(s_i),v(o_i))$ w.r.t. $P_{v(S_i)}$ and W_{-} (as in the membership part of Proposition 9(1)), and (iii) $L_i\ R_i\ v(?S_i)$ holds. The complexity of these steps are in P$^{\text{NP}}$. It holds that $M \models \sim[\neg]v(H(s_i,p_i,$ $o_i,?S_i))$ and $L_i\ R_i\ v(?S_i)$ if (i) there are c, c' in $v(?S_i)$ s.t. *conflict*(c,c') holds or there is no well-founded proof of $[\neg]v(p_i)(v(s_i),v(o_i))$ w.r.t. $P_{v(S_i)}$ and W_{-} and (ii) $L_i\ R_i\ v(?S_i)$ holds. The complexity of these steps are in P$^{\text{co-NP}}$. Therefore, the complexity of testing if it does not hold $M \models [\sim][\neg]v(H(s_i,p_i,\, o_i,?S_i))$ and $L_i\ R_i\ v(?S_i)$, for $i = 1,...,k$, is in co-P$^{\text{NP}}$. Thus, the complexity to verify if $v \notin Answers_{\mathcal{C}}^{\#n}(CQ)$ is in NP$^{\text{P}^{\text{NP}}}$=NP$^{\text{NP}}$. Thus, checking if $v \in Answers_{\mathcal{C}}^{\#n}(CQ)$ is in co-NP$^{\text{NP}}$ w.r.t. the size of $\mathcal{C}$.

2)
Hardness) Let $CQ = \langle H(s,p,o,?S), \emptyset^{\supseteq}\rangle \wedge \langle \neg H(s,p,o,?S), \emptyset^{\supseteq}\rangle$. Then, $v \in Answers_{\mathcal{C}}^{\#n}(CQ)$ iff $\mathcal{C}$ has no $\#n$-provenance stable model. It follows from Proposition 9(2) that deciding if $\mathcal{C}$ has $\#n$-provenance stable model is NP-hard w.r.t. $graph\_size(\mathcal{C})$. Therefore, the complexity of checking if $v \in Answers_{\mathcal{C}}^{\#n}(CQ)$ is co-NP-hard w.r.t. $graph\_size(\mathcal{C})$.

Membership) Assume that we want to verify if $v \notin Answers_{\mathcal{C}}^{\#n}(CQ)$. Guess a $\#n$-provenance semi-Herbrand interpretation I of C. Then, test if $I \in \mathcal{M}^{\#n}(\mathcal{C})$, as in the membership part of 2) of Proposition 9 (Step 1). Now, test if it does not hold $I \models$ $[\sim][\neg]v(H(s_i,p_i,\, o_i,?S_i))$ and $L_i\ R_i\ v(?S_i)$, for $i = 1,...,k$ (Step 2). Now the complexity of Step 1 is in P w.r.t. $graph\_size(\mathcal{C})$, while the complexity of Step 2 is in P w.r.t. $graph\_size(\mathcal{C})$. Therefore, checking if $v \notin Answers_{\mathcal{C}}^{\#n}(CQ)$ is in NP w.r.t. $graph\_size(\mathcal{C})$. Thus, checking if $v \in Answers_{\mathcal{C}}^{\#n}(CQ)$ is in co-NP w.r.t. $graph\_size(\mathcal{C})$. □

Proposition 11. Let $\mathcal{C} = \langle \{\langle Nam_{O_i}, O_i\rangle \mid i = 1,...,m\}, \mathit{conflict}\rangle$ be a positive g-RDF information base and let $CQ = SQ_1 \wedge ... \wedge SQ_k$, where each $SQ_i = \langle H(s_i,p_i,\, o_i,?S_i), L_i^{R_i}\rangle$ is a simple provenance query of $\mathcal{C}$. Let v be a mapping (i) from the variables in s_i, p_i, o_i to $V_{\mathcal{C}}^{\#n}$ and (ii) from $?S_i$ to $\mathcal{P}(names_{\mathcal{C}})$. The time complexity of checking if $v \in Answers_{\mathcal{C}}^{\#n}(CQ)$ is:

1. NP-complete w.r.t. the size of $\mathcal{C}$ and
2. P-complete w.r.t. $graph\_size(\mathcal{C}) = size(G_1) + ... + size(G_m)$.

Proof: 1)
Hardness) Let P be a definite logic program whose predicates have arity 2. Let $\mathcal{C}_P$ be the g-RDF information base which contains a single g-RDF ontology $O = \langle G_O, P_O \rangle$ with name c s.t. $G_O = \emptyset$ and P_O is derived from P by replacing each predicate and constant s in P by a unique URI $uri(s) \notin \mathcal{V}_{RDF} \cup \mathcal{V}_{RDFS}$. Obviously, $\mathcal{C}_P$ is derived from P in polynomial time w.r.t. the size of P. Let $p(s, o)$ be a ground atom in the Herbrand Universe of P. It holds that $P \models p(s, o)$ iff $\mathcal{C}_P \models^{pr_{\#n}} H(uri(s), uri(p), uri(o), \{c\})$. In [25], it is proved that deciding if P entails a ground atom in the Herbrand Universe of P is NP-complete w.r.t. the size of P. Thus, the time complexity for computing $Answers^{\#n}_{\mathcal{C}}(CQ)$ is NP-hard w.r.t. the size of $\mathcal{C}$.

Membership) For all $i = 1, ..., k$, check if (i) there are no $c, c' \in v(?S_i)$ s.t. $conflict(c, c')$ holds, (ii) $L_i\ R_i\ v(?S_i)$ holds, and (iii) there is a well-founded proof of $v(p_i)(v(s_i), v(o_i))$ w.r.t. $P_{v(?S_i)}$ as defined in the proof of membership part of Proposition 9(1). These checks take NP time with respect to the size of $\mathcal{C}$. Thus, the time complexity for checking if $v \in Answers^{\#n}_{\mathcal{C}}(CQ)$ is in NP w.r.t. the size of $\mathcal{C}$.

2)
Hardness) Let P be a definite propositional logic program whose atoms in the bodies of the rules are at most 2. Let $\mathcal{C}_P$ be the g-RDF information base which contains a single g-RDF ontology $O = \langle G_O, P_O \rangle$ with name c, where G_O is constructed as follows:

(i) For each j^{th} rule of the form $p_0 \leftarrow . \in P$, G_O contains the g-RDF triple $ex{:}c_{0,0}(ex{:}p_0, ex{:}a_j)$.
(ii) For each j^{th} rule of the form $p_0 \leftarrow p_1. \in P$, G_O contains the g-RDF triples $ex{:}c_{1,0}(ex{:}p_0, ex{:}a_j)$ and $ex{:}c_{1,1}(ex{:}p_1, ex{:}a_j)$.
(iii) For each j^{th} rule of the form $p_0 \leftarrow p_1, p_2. \in P$, G_O contains the g-RDF triples $ex{:}c_{2,0}(ex{:}p_0, ex{:}a_j)$, $ex{:}c_{2,1}(ex{:}p_1, ex{:}a_j)$, and $ex{:}c_{2,2}(ex{:}p_2, ex{:}a_j)$.

Now, P_O contains the following three rules:

(i) $rdf{:}type(?x, ex{:}true) \leftarrow ex{:}c_{0,0}(?x, ?w)$.
(ii) $rdf{:}type(?x, ex{:}true) \leftarrow rdf{:}type(?y, ex{:}true), ex{:}c_{1,0}(?x, ?w), ex{:}c_{1,1}(?y, ?w)$.
(iii) $rdf{:}type(?x, ex{:}true) \leftarrow rdf{:}type(?y, ex{:}true), rdf{:}type(?z, ex{:}true), ex{:}c_{2,0}(?x, ?w), ex{:}c_{2,1}(?y, ?w), ex{:}c_{2,2}(?z, ?w)$.

Then, $P \models p$, for a propositional atom p, iff $\mathcal{C}_P \models rdf{:}type(ex{:}p, ex{:}true)$. Obviously, $\mathcal{C}_P$ can be constructed in polynomial time w.r.t. the size of P. In [19], it is proved that deciding if P entails a propositional atom is P-complete w.r.t. the size of P. Thus, the time complexity for computing $Answers^{\#n}_{\mathcal{C}}(CQ)$ is P-hard w.r.t. $graph\_size(\mathcal{C})$.

Membership) Let k' be the maximum number of g-RDF triples in the body of a rule in $\mathcal{C}$. We assume that $k' \geq 2$. Let l be the number of g-RDF rules appearing in $\mathcal{C}$. The number of ground rules in $\Pi^{\mathtt{G}}_{\mathcal{C}}$ is at most $m * |V^{\#n}_{\mathcal{C}}|^3$. The number of ground rules in $\Pi^{\mathtt{P}}_{\mathcal{C}}$ is at most $l * (|V^{\#n}_{\mathcal{C}}|^3 * 2^m)^{k'+1}$, since the number of provenance quads appearing in a rule of $\Pi^{\mathtt{P}}_{\mathcal{C}}$ is at most $k' + 1$. The number of ground rules in $\Pi^{\mathtt{H}\#n}_{\mathcal{C}}$ is in $O((|V^{\#n}_{\mathcal{C}}|^3 * 2^m)^3)$. Thus, the number of rules in $P^{\#n}_{\mathcal{C}}$ is in $O(l * (|V^{\#n}_{\mathcal{C}}|^3 * 2^m)^{k'+1})$. The number of provenance quads in the body of a rule in $P^{\#n}_{\mathcal{C}}$ is at most k'. The time complexity of taking the union of k' sets of at most m elements is in $O(k'^2 * m^2)$ and the time complexity of checking $compatible(S)$, for $S \subseteq names_C$, is in $O(m^3)$. The time complexity for checking if $S \subseteq S'$, for $S, S' \subseteq names_{\mathcal{C}}$, is in $O(m^2)$. Finally, the size of $T^{\uparrow\omega}_{P^{\#n}_{\mathcal{C}}}(\emptyset)$ is at most $|V^{\#n}_{\mathcal{C}}|^3 * 2^m$. Therefore, the time complexity of checking if $v \in Answers^{\#n}_{\mathcal{C}}(CQ)$ is in P w.r.t. $graph\_size(\mathcal{C})$, since k' and m are fixed. $\square$

Appendix A: Table of Symbols

List of Symbols	
Symbol	*Description*
V_G	the vocabulary of g-RDF graph G
V_P	the vocabulary of g-RDF program P
$\mathcal{C}$	a g-RDF information base
Nam_O	the name of a g-RDF ontology O
$names_{\mathcal{C}}$	the names of all the g-RDF ontologies appearing in $\mathcal{C}$
$\mathit{conflict}(c, c')$	indicates that there exist conflict between g-RDF ontologies c and c'
$\mathit{compatible}(S)$	means that there are no g-RDF ontologies $c, c' \in S$ s.t. $\mathit{conflict}(c, c')$
$\mathit{union}(SS, S)$	means that S is the union of the sets appearing in SS
$sk(G)$	the skolemization of g-RDF graph G
$V_{\mathcal{C}}$	the vocabulary of $\mathcal{C}$
$Res_{\mathcal{C}}^H$	the set of resources of a provenance Herbrand interpretation of $\mathcal{C}$
$\mathcal{I}_{\mathcal{C}}^H$	the set provenance Herbrand interpretation of $\mathcal{C}$
$prov(P)$	the set of the provenance rules generated fron the g-RDF rules in P
$[prov(P)]_{\mathcal{C}}$	the grounding of $prov(P)$ according to vocabulary $V_{\mathcal{C}}$
$\mathcal{M}(\mathcal{C})$	the set of provenance stable models of $\mathcal{C}$
$\mathcal{C} \models^{pr} G$	for all $M \in \mathcal{M}(\mathcal{C})$, it holds that $M \models G$
$V_{\mathcal{C}}^{\#n}$	$V_{\mathcal{C}} - \{rdf{:}\_i \in \mathcal{V}_{RDF} \mid i > n\}$
$Res_{\mathcal{C}}^{H\#n}$	$Res_{\mathcal{C}}^H - \{rdf{:}\_i \in \mathcal{V}_{RDF} \mid i > n\}$
$\mathcal{I}^{H\#n}(\mathcal{C})$	the $\#n$-provenance Herbrand interpretations of $\mathcal{C}$
$[prov(P)]_{\mathcal{C}}^{\#n}$	the grounding of $prov(P)$ according to vocabulary $V_{\mathcal{C}}^{\#n}$
$\mathcal{M}^{\#n}(\mathcal{C})$	the set of $\#n$-provenance stable models of $\mathcal{C}$
$\Pi_{\mathcal{C}}^{\mathtt{G}}$	$\{H(s,p,o,\{Nam_{O_i}\}) \leftarrow . \mid p(s,o) \in sk(G_i) \text{ and } i = 1,...,m\}$, where $O_i = \langle G_i, P_i \rangle$ are the g-RDF ontologies appearing in $\mathcal{C}$
$\Pi_{\mathcal{C}}^{\mathtt{P}}$	$\bigcup\{prov(P_i) \mid i = 1,...,m\}$, where P_i are the g-RDF programs appearing in $\mathcal{C}$
$\Pi_{\mathcal{C}}^{\mathtt{H}\#n}$	the set of provenance interpretation rules
$\Pi_{\mathcal{C}}^{\#n}$	$\Pi_{\mathcal{C}}^{\mathtt{G}} \cup \Pi_{\mathcal{C}}^{\mathtt{P}} \cup \Pi_{\mathcal{C}}^{\mathtt{H}\#n}$
$P_{\mathcal{C}}^{\#n}$	a ground extended logic program defined from $\Pi_{\mathcal{C}}^{\#n}$
$f_{\mathcal{C}}$	a mapping from $\mathcal{P}(names_{\mathcal{C}})$ to a set of 2^m constants, where m is the number of g-RDF ontologies appearing in $\mathcal{C}$
$ELP(I) =$	$\{H(s,p,o,f_{\mathcal{C}}(S)) \mid s,p,o \in V_{\mathcal{C}}^{\#n} \text{ and } \langle I(s), I(o), S \rangle \in PT_I(p)\} \cup$ $\{\neg H(s,p,o,f_{\mathcal{C}}(S)) \mid s,p,o \in V_{\mathcal{C}}^{\#n} \text{ and } \langle I(s), I(o), S \rangle \in PF_I(p)\}$

Table 1. Symbols and Description

References

1. Y. Amsterdamer, D. Deutch, and V. Tannen. Provenance for aggregate queries. In *13th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS-2011)*, pages 153–164, 2011.
2. A. Analyti, G. Antoniou, and C. V. Damasio. A Formal Theory for Modular ERDF Ontologies. In *3rd International Conference Web Reasoning and Rule Systems (RR 2009)*, pages 212–226, 2009.
3. A. Analyti, G. Antoniou, and C. V. Damásio. Computability and Complexity Issues of ERDF. In *Technical Report, FORTH-ICS, submitted for publication*, 2010. Available at `http://www.ics.forth.gr/~analyti/Papers_2/ERDF_Complexity.pdf`.

4. A. Analyti, G. Antoniou, C. V. Damásio, and G. Wagner. Negation and Negative Information in the W3C Resource Description Framework. *Annals of Mathematics, Computing & Teleinformatics (AMCT)*, 1(2):25–34, 2004.
5. A. Analyti, G. Antoniou, C. V. Damásio, and G. Wagner. Extended RDF as a Semantic Foundation of Rule Markup Languages. *Journal of Artificial Intelligence Research (JAIR)*, 32:37–94, 2008.
6. K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, and J. Zhao. PROV-O: The PROV Ontology, 2012. editors: Timothy Lebo, Satya Sahoo, and Deborah McGuinness, W3C Working Draft. Consulted `http://www.w3.org/TR/2012/WD-prov-o-20120724/`.
7. H. Boley, G. Hallmark, M. Kifer, A. Paschke, A. Polleres, and D. Reynolds. RIF Core Dialect (Second Edition)). W3C Recommendation 5 February 2013. Latest version available at `http://www.w3.org/TR/rif-core/`.
8. G. Brewka and T. Eiter. Equilibria in Heterogeneous Nonmonotonic Multi-Context Systems. In *22nd AAAI Conference on Artificial Intelligence (AAAI-2007)*, pages 385–390, 2007.
9. P. Buneman, S. Khanna, and W. C. Tan. Why and Where: A Characterization of Data Provenance. In *8th International Conference on Database Theory (ICDT-2001)*, pages 316–330, 2001.
10. P. Buneman and E. V. Kostylev. Annotation algebras for RDFS. In *2nd International Workshop on the role of Semantic Web in Provenance Management (SWPM-2010)*, pages 316–330, 2010.
11. J. J. Carroll, C. Bizer, P. J. Hayes, and P. Stickler. Named graphs. *Journal of Web Semantics*, 3(4), 2005.
12. J. J. Carroll, C. Bizer, P. J. Hayes, and P. Stickler. Named graphs, Provenance and Trust. In *14th International Conference on World Wide Web (WWW-2005)*, pages 613–622, 2005.
13. W. Chen and D. S. Warren. Tabled Evaluation With Delaying for General Logic Programs. *Journal of the ACM*, 43(1), 1996.
14. J. Cheney, L. Chiticariu, and W. C. Tan. Provenance in Databases: Why, How, and Where. *Foundations and Trends in Databases*, 1(4):379–474, 2009.
15. P. Ciccarese, E. Wu, G. Wong, M. Ocana, J. Kinoshita, A. Ruttenberg, and T. Clark. The SWAN biomedical discourse ontology. *J Biomed Inform.*, 41(5):739–751, 2008.
16. Y. Cui. Lineage Tracing in Data Warehouses. Ph.D., Stanford InfoLab, December 2001.
17. C. V. Damásio, A. Analyti, and G. Antoniou. Provenance for SPARQL Queries. In *11th International Semantic Web Conference (ISWC-2012)*, 2012. To appear.
18. C. V. Damasio, A. Analyti, and G. Antoniou. Justifications for Logic Programming. In *12th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-2013)*, pages 530–542, 2013.
19. E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and Expressive power of Logic Programming. *ACM Computing Surveys*, 33(3):374–425, 2001.
20. J. de Bruijn, E. Franconi, and S. Tessaris. Logical Reconstruction of Normative RDF. In *OWL: Experiences and Directions Workshop (OWLED-2005)*, Galway, Ireland, November 2005.
21. S. Decker, M. Sintek, and W. Nejdl. The Model-Theoretic Semantics of TRIPLE. Technical Report, 2002.
22. L. Ding, T. Finin, Y. Peng, A. Joshi, P. P. da Silva, and D. L. McGuinness. Tracking RDF Graph Provenance using RDF Molecules. Technical Report, UMBC TR-CS-05-06, 2005.
23. R. Q. Dividino, S. Sizov, S. Staab, and B. Schueler. Querying for provenance, trust, uncertainty and other meta knowledge in RDF. *Journal of Web Semantics*, 7(3), 2009.
24. X. L. Dong and F. Naumann. Data fusion - Resolving Data Conflicts for Integration. *Proceedings of the VLDB Endowment*, 2(2):1654–1655, 2009.
25. T. Eiter, W. Faber, M. Fink, and S. Woltran. Complexity Results for Answer Set Programming with Bounded Predicate Arities and Implications. *Annals of Mathematics and Artificial Intelligence*, 51(2-4), 2007.
26. G. Flouris, I. Fundulaki, P. Pediaditis, Y. Theoharis, and V. Christophides. Coloring RDF Triples to Capture Provenance. In *8th International Semantic Web Conference (ISWC-2009)*, pages 196–212, 2009.

27. M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. Conflict-Driven Answer Set Solving. In *20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pages 386–392, 2007.
28. F. Geerts, G. Karvounarakis, V. Christophides, and I. Fundulaki. Algebraic Structures for Capturing the Provenance of SPARQL Queries. In *16th International Conference on Database Theory (ICDT-2013)*, 2013.
29. F. Geerts and A. Poggi. On database query languages for K-relations. *Journal of Applied Logic*, 8(2):173–185, 2010.
30. A. V. Gelder, K. A. Ross, and J. S. Schlipf. The Well-Founded Semantics for General Logic Programs. *Journal of the ACM*, 38(3):620–650, 1991.
31. M. Gelfond and V. Lifschitz. The Stable Model Semantics for Logic Programming. In R. Kowalski and K. A. Bowen, editors, *5th International Conference on Logic Programming*, pages 1070–1080. MIT Press, 1988.
32. M. Gelfond and V. Lifschitz. Logic programs with Classical Negation. In *7th International Conference on Logic Programming*, pages 579–597, 1990.
33. T. J. Green, Z. G. Ives, and V. Tannen. Reconcilable Differences. *Theory of Computing Systems*, 49(2):460–488, 2011.
34. T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *26th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS-2007)*, pages 31–40, 2007.
35. R. V. Guha, R. McCool, and R. Fikes. Contexts for the Semantic Web. In *3rd International Semantic Web Conference (ISWC-2004)*, pages 32–46, 2004.
36. O. Hartig. Provenance Information in the Web of Data. In *WWW2008 Workshop on Linked Data on the Web (LDOW-2009)*, 2009.
37. O. Hartig and J. Zhao. Provenance Vocabulary Core Ontology Specification. 14 March 2012. Latest version available at `http://trdf.sourceforge.net/provenance/ns.html`.
38. O. Hartig and J. Zhao. Publishing and Consuming Provenance Metadata on the Web of Linked Data. In *3rd International Provenance and Annotation Workshop (IPAW-2010)*, pages 78–90, 2010.
39. P. Hayes. RDF Semantics. W3C Recommendation, 10 February 2004. Available at `http://www.w3.org/TR/2004/REC-rdf-mt-20040210/`.
40. T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011.
41. H. Herre, J. Jaspars, and G. Wagner. Partial Logics with Two Kinds of Negation as a Foundation of Knowledge-Based Reasoning. In D. M. Gabbay and H. Wansing, editors, *What Is Negation?* Kluwer Academic Publishers, 1999.
42. G. Karvounarakis, Z. G. Ives, and V. Tannen. Querying data provenance. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-2010)*, pages 951–962, 2010.
43. M. Kifer and H. Boley. RIF Overview (Second Edition). W3C Working Group Note 5 February 2013. Latest version available at `http://www.w3.org/TR/rif-overview/`.
44. M. Kifer, G. Lausen, and J. Wu. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM*, 42(4):741–843, 1995.
45. G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, 10 February 2004. Available at `http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/`.
46. N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, and F. Scarcello. The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic*, 7(3):499–562, 2006.
47. L. Liu, E. Pontelli, T. C. Son, and M. Truszczynski. Logic programs with abstract constraint atoms: The role of computations. *Artificial Intelligence*, 174(3-4):295–315, 2010.
48. J. W. Lloyd. *Foundations of Logic Programming (2nd edition)*. Springer-Verlag, 1987.
49. J. W. Lloyd and R. W. Topor. Making Prolog more Expressive. *Journal of Logic Programming*, 1(3):225–240, 1984.
50. R. M. MacGregor and I.-Y. Ko. Representing Contextualized Data using Semantic Web Tools. In *First International Workshop on Practical and Scalable Semantic Systems (PSSS-2003)*, 2003.

51. S. Muñoz, J. Pérez, and C. Gutiérrez. Minimal Deductive Systems for RDF. In *4th European Semantic Web Conference (ESWC 2007)*, pages 53–67, 2007.
52. I. Niemelä and P. Simons. Smodels - An Implementation of the Stable Model and Well-Founded Semantics for Normal LP. In *4th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-1997)*, pages 421–430, 1997.
53. A. Paschke, L. Morgenstern, D. Hirtle, A. Ginsberg, P.-L. Patranjan, and F. McCabe. RIF Use Cases and Requirements (Second Edition). W3C Working Group Note 5 February 2013. Latest version available at `http://www.w3.org/TR/rif-ucr/`.
54. A. Polleres, C. Feier, and A. Harth. Rules with Contextually Scoped Negation. In *3rd European Semantic Web Conference (ESWC-2006)*, pages 332–347, 2006.
55. E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. W3C Recommendation, 15 January 2008. Available at `http://www.w3.org/TR/rdf-sparql-query/`.
56. S. Ram and J. Liu. Understanding the Semantics of Data Provenance to Support Active Conceptual Modeling. In *1st International ACM-L Workshop on Active Conceptual Modeling of Learning (ACM-L-2006)*, pages 17–29, 2006.
57. K. A. Ross. On Negation in HiLog. *Journal of Logic Programming*, 18(1):27–53, 1994.
58. K. F. Sagonas, T. Swift, and D. S. Warren. XSB as an Efficient Deductive Database Engine. In R. T. Snodgrass and M. Winslett, editors, *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, pages 442–453. ACM Press, 1994. Available at `http://xsb.sourceforge.net/`.
59. M. Sintek and S. Decker. TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web. In *1st International Semantic Web Conference (ISWC-2002)*, pages 364–378. Springer-Verlag, 2002.
60. T. Swift and D. S. Warren. XSB: Extending Prolog with Tabled Logic Programming. *Theory and Practice of Logic Programming (TPLP)*, 12(1-2):157–187, 2012.
61. H. J. ter Horst. Completeness, Decidability and Complexity of Entailment for RDF Schema and a Semantic Extension Involving the OWL Vocabulary. *Journal of Web Semantics*, 3(2-3):79–115, 2005.
62. X. Yin, J. Han, and P. S. Yu. Truth Discovery with Multiple Conflicting Information Providers on the Web. In *13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2007)*, pages 1048–1052, 2007.
63. A. Zimmermann, N. Lopes, A. Polleres, and U. Straccia. A General Framework for Representing, Reasoning and Querying with Annotated Semantic Web Data. *Journal of Web Semantics*, 11:72–95, 2012.