Combining Reasoning on Semantic Web Metadata

Technical Report
TR-FBK-DKM-2014-1

Loris Bozzato and Luciano Serafini

Abstract. As the amount of available linked data expand and the number of related applications increases, the management of aspects such as provenance and access control of such data begin to become an issue. Current approaches do not provide sufficient support for automatic reasoning over different metadata and their possible interdependencies. MetaReasons is a framework that supports the representation of metadata in a logical formalism and consequently to support automated reasoning on metadata. Different types of metadata, such as data-provenance and accessibility-restrictions are represented as distinct meta-theories, and dependencies between types of metadata are represented by rules between different meta-theories. In this paper we present the logic based definition of the MetaReasons framework and two examples of meta-theories for provenance and access control. Moreover, we propose a materialization calculus for concrete forward reasoning on the two aspects.

1 Introduction

As the production and the usage of Semantic Web (SW) / LOD data continuously expands, a flexible management of different aspects of these data become more and more urgent. Among the most urgent aspects for which a solution need to be found there are provenance, access control, privacy and trust relative to the use of combinations of multiple datasets and sources of information. A basic feature shared by all of these aspects is that they consist of meta-information that qualify a dataset or relate to each other different datasets. As such, in the meta-level data-sources can be represented as individuals of the domain of meta-theories; meta-data about dataset $d$ can be represented by stating properties of $i_d$ (the individual for $d$) and by relating $i_d$ with other individuals (e.g. representing user types, resource provenance, levels of trust).

Meta-reasoning takes clearly a major role in supporting a flexible management of such aspects: it allows to infer implicit relations among datasets or to automatic classify newly added datasets. The problem of reasoning about each single metadata information (as provenance, access control, privacy and trust) of data sources has already been considered in the area of SW/LOD data. Multiple alternative and complementary frameworks for representing and reasoning about such aspects have been developed (see for instance [7, 11]) and some of them were formalized in a web ontology language (e.g. [6]). Here we are agnostic with respect to the appropriateness of each single framework, and in general we believe that different frameworks for metadata may be useful in different situations. In this paper we focus on providing a methodology that allow to transparently adopt one of the alternative frameworks.

A further issue we want to address in this work derives from the observation that meta-data about different aspects are dependent one with the other. Accessibility and trust associated to a particular dataset might depend on its provenance (e.g. one may require to access a set of data with a specific trust and privacy criteria). Metadata, therefore, should not be considered as separate aspects in reasoning. Relations between each of the aspects may be defined by the modeller: intuitively, these relations define policies (i.e. global axioms) that connect different aspects. Policies provide the means for a combination of the meta-theories for all of the different aspects in a coherent metaknowledge.

Given the latter issues, we are interested in a flexible combination of reasoning and querying between the metalevel (i.e. about datasets) and the knowledge level (i.e. inside datasets). For this purpose we have to consider the relations between reasoning (and querying) about datasets in the metalevel and reasoning inside each dataset. Indeed, reasoning and query answering in single datasets can now be influenced by all of the reasoning at the meta level.

Main contributions described in this paper are:

- the definition of a framework, called MetaReasons, for the representation and reasoning over metadata such as provenance, data access level, privacy information and trust in an integrated metalevel structure.
- a concrete implementation of the MetaReasons framework that supports combined reasoning and querying between different meta-data types as well as on the contents of the data sources\(^2\).

The structure MetaReasons is depicted in Figure 1. MetaReasons is a two-layer framework, composed of an object layer containing the data-sources and a meta-level containing knowledge about metadata. This is similar to our previous works on the CKR contextual framework [12, 3].\(^1\) First of all, we note that we have a clear separation between the level representing the metainformation (Metaknowledge) associated to a set of information sources (viewed as atomic components) and the level (Object knowledge) respective to the actual dataset contents. Each of the meta-theories in the upper level separately encodes one of the aspects to be represented about datasets. Note that each of the meta-theories can use its own schema and possibly its own local reasoning formalism. It is intended, however, that the final representation of the meta-theories in the upper level has to be reencoded to a single formalism, possibly encoded as one of the frameworks.

\(^{2}\)The framework has been currently proposed inside the PlanetData Project (http://www.planet-data.eu/) for reasoning with provenance, access control, privacy and trust of SW data.

\(^{1}\)Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy. E-mail: \{bozzato,serafini\}@fbk.eu
OWL2 fragments. This will enable to adapt different existing models and readily available ontologies for different aspects under a single framework. This architecture implies that we can decouple reasoning on metaknowledge from the one on the object knowledge, thus allowing to use different languages and formalisms on the two parts (e.g., RDFS reasoning on data and OWL reasoning in metatheories).

Dependency axioms (i.e. policies) connecting metatheories can be defined in the part of the metaknowledge covering all of the single theories. Specification of policies should be general enough to connect any new metatheory that can be plugged in the metaknowledge.

On this architecture, unified queries over both the meta-knowledge and the object knowledge can be expressed in standard SPARQL (possibly extended with the primitives defined by the architecture) exploiting the features of named graphs. For example, if we want to retrieve “all of the facts about a certain event, from datasets which have a certain level of trust and for which we have access rights” we can express this in a SPARQL query that selects the datasets matching the meta-level requirements and then run the object-query “all of the facts about a certain event” on the selected datasets.

We can see different advantages in such integrated architecture:

- The possibility to combine all of the metainformation about datasets in a coherent view in which reasoning can be activated.
- The fact that we are agnostic on underlying (logic) formalisms inside metatheories give us the flexibility to integrate external reasoning methods and tools.
- The possibility to configure the upper layer in accordance to the intended application: a modeler can choose to combine any number of metatheories needed for the modeling of domain policies.
- The view of datasets as individuals in the metaknowledge permits to reason over large collections of data sources of arbitrary size.
- Metaknowledge can be used as a “filter” to direct queries to specific datasets, enabling a “divide and conquer” approach to query answering useful to limit processing only to relevant part of data.
- In the formalization we propose, object and meta knowledge are expressed in the same language, and object and meta inference is realized using the same reasoning engine. This provides the clear advantage of a seamless integration of object and meta reasoning.

This paper presents the first steps in the realization of the MetaReasons framework: after some preliminaries (Section 2), we provide a description logics based formalization of the intuitive schema of “unified knowledge base” discussed so far (Section 3). We then propose (Section 4) two initial examples of metatheories encoding provenance and access control information, providing a detailed example of use. In the following (Section 5), we provide a sound and complete materialization calculus for our framework. After a brief discussion on related works (Section 6), we conclude (Section 7) by giving some outlook on the following phases of implementation and metatheories development.

Complete proofs for all of the presented formal results and complete tables for calculus rules are provided in the Appendix.

2 Preliminaries

**SROIQ-RL language.** In the following we assume familiarity with the basic concepts of description logics [1] and with the description logic SROIQ [8]. We based MetaReasons framework on a restriction of SROIQ syntax corresponding to OWL RL [10]; we call such language SROIQ-RL. The language is obtained by restricting the form of SROIQ General Concept Inclusion axioms (GCIs) and concept equivalences as follows: the GCIs \( C \sqsubseteq D \) is admitted only when \( C \) and \( D \) are members of two classes of concept expressions called left-side concept and right-side concept, respectively; the equivalence assertion \( E \equiv F \) is admitted only if \( E \) and \( F \) are both members of the class of left- and right-side concepts. The classes of left- and right-side concepts are defined by the following grammar:

\[
C : = A | \{ a \} | C_1 \sqcap C_2 | C_1 \sqcup C_2 | \exists R.C_1 | \exists R.\{ a \} | \exists R.\top \\
D : = A | \neg C_1 | D_1 \sqcap D_2 | \exists R.\{ a \} \sqcap R.D_1 | \leq nR.C_1 | \leq nR.\top
\]

where \( A \) is a concept name, \( R \) is role name and \( n \in \{ 0, 1 \} \). A both-side concept \( E, F \) is a concept expression which is both a left- and right-side concept. TBox axioms can only take the form \( C \sqsubseteq D \) or \( E \equiv F \). The RBox can contain every role axiom of SROIQ except for Ref(\( R \)). ABox concept assertions can be only stated in the form \( D(a) \), where \( D \) is a right-side concept.

**Normal form.** To facilitate the definition of the calculus, we suppose that SROIQ-RL input axioms are in the following normal form:

\[
\begin{align*}
\text{Table 1. } & \quad \text{SROIQ-RL normal form axioms} \\
A(a) & \quad R(a,b) & \quad \neg R(a,b) & \quad a = b & \quad a \neq b \\
A \sqcap B & \quad \{ a \} \sqsubseteq B & \quad A \sqsubseteq \neg B & \quad A \sqcap B \sqsubseteq C \\
\exists R. A \sqsubseteq B & \quad A \sqsubseteq \exists R.\{ a \} & \quad A \sqsubseteq \forall R.B & \quad A \sqsubseteq \exists 1.R.B \\
R \sqsubseteq T & \quad R \circ S \sqsubseteq T & \quad \text{Dis}(R,S) & \quad \text{Inv}(R,S) & \quad \text{Irr}(R)
\end{align*}
\]

With \( A, B, C \in \text{NC} \) and \( R, S, T \in \text{NR} \).

As in [9], we assume that rule chain axioms in input are already decomposed in binary role chains. We can provide a set of rules (Table 2 in the Appendix) that transform any SROIQ-RL KB in an “equivalent” KB in normal form. The correctness of this translation can be shown by the following lemma.

**Lemma 1** For every SROIQ-RL knowledge base \( K \) over a signature \( \Sigma \), a knowledge base \( K' \) over an extended signature \( \Sigma' \) can be computed s.t. (i) all axioms in \( K \) are in normal form (ii) the size of \( K' \) is a linear factor of the size of \( K \), (iii) for all axioms \( \alpha \) in the signature \( \Sigma \), \( K \models \alpha \) if and only if \( K' \models \alpha \).

**Datalog rules.** In defining the reasoning of MetaReasons, we follow the approach described in [9], where the inference method is specified by a set of forward inference rules expressed in datalog. In the following we summarize their basic definitions.
A signature is a tuple \((C, P)\), with \(C\) a finite set of constants and \(P\) a finite set of predicates, each of which is associated with an arity. We assume a set \(V\) of variables and we call terms the elements of \(C \cup V\). A datalog atom over \((C, P)\) is in the form \(p(t_1, \ldots, t_n)\) with \(p \in P\) and every \(t_i \in C \cup V\) for \(i \in \{1, \ldots, n\}\), and \(n\) is the arity of \(p\). A rule is an expression in the form \(B_1, \ldots, B_m \rightarrow H\) where \(H\) and \(B_1, \ldots, B_m\) are datalog atoms (the head and body of the rule). A ground substitution \(\sigma\) for \((C, P)\) is a function \(\sigma : V \rightarrow C\). We define as usual substitutions on atoms and ground instances of atoms. A ground rule with empty body (\(\rightarrow H\) or simply \(H\)) is called a fact. A program \(P\) is a finite set of datalog rules and facts.

A proof tree for \(P\) is a structure \(\langle N, E, \lambda \rangle\) with \(\langle N, E \rangle\) a finite directed tree and \(\lambda\) a labelling function assigning a ground atom to each node, where: for each \(v \in N\), there exists a rule \(B_1, \ldots, B_m \rightarrow H\) in \(P\) and a ground substitution \(\sigma\) s.t. (i) \(\lambda(v) = \sigma(H)\) and (ii) \(v\) has \(m\) child nodes \(w_i\) in \(E\), with \(\lambda(w_i) = \sigma(B_i)\) for \(i \in \{1, \ldots, m\}\).

A ground atom \(H\) is a consequence of \(P\) (denoted \(P \models H\)) if there exists a proof tree for \(P\) with root node \(v\) and with \(\lambda(v) = H\).

### 3 Architecture definition

**Syntax.** As we previously discussed, in order to uniformly represent and reason over metalevel and knowledge level information, we want to model knowledge bases as two layered structures: we call such structures Unified Knowledge Bases (UKB), since they offer an unified model to reason on both levels.

Given a fixed set of datasets, with names in \(N\), the upper layer of an UKB is composed by a DL knowledge base \(\forall \Gamma\) containing a set of metatheories \(\forall MT_i\), each one encoding a specific type of meta-information of the datasets in \(N\), and a set \(P\) of axioms, that we call policies, spanning over (and possibly linking) elements of any metatheory. The lower layer of the UKB is composed by a set \(\forall \Gamma\) of datasets (i.e. DL knowledge bases) \(\forall DS_n\) for each dataset name \(n \in N\). Knowledge in a UKB is thus expressed in a DL language: the following definitions are parametric to any description logic language, while successively we instantiate them to \(\forall SROIQ\).

The knowledge at the meta-level of a UKB is expressed by a DL language containing symbols for referring to datasets and for expressing facts about the metatheories. Formally, a meta-vocabulary is a DL vocabulary \(\Gamma\) composed of a set of atomic concepts \(NC_{\Gamma}\), a set atomic roles \(NR_{\Gamma}\) and a set of individual constants \(NI_{\Gamma}\) that are mutually disjoint and such that: (i) \(N \subseteq NI_{\Gamma}\) is a set of dataset names; (ii) \(\Gamma \supseteq \bigcup_{i \in \{1, \ldots, m\}} NI_i\). Intuitively, each \(\Gamma_i\) represent the sub-vocabulary for each of the \(m\) metatheories. The meta-language denoted by \(\forall \Gamma\), is a DL language defined over meta-vocabulary \(\Gamma\). We may distinguish local languages \(\forall \Gamma_i\), based on vocabulary subset \(NI_i\).

On the other hand, each dataset at the knowledge level can be defined with its own vocabulary and language: we call, object vocabulary any DL vocabulary \(\Sigma = NC_{\Sigma} \cup NR_{\Sigma} \cup NI_{\Sigma}\). The object-language \(\forall \Gamma\) is the DL language defined from the \(\Sigma\) vocabulary.

Using these notions, we can now introduce the structure representing a unified knowledge base with the following definition.

**Definition 1 (Unified Knowledge Base, UKB)** An Unified Knowledge Base (UKB) is a structure \(\forall \Gamma = (\forall P, \forall \forall \exists)\) such that:

\[
\forall P = \langle P, \{MT_1, \ldots, MT_m\} \rangle
\]

\(\forall\) is a DL knowledge base over \(\forall \Gamma\); (policy knowledge base);

\(MT_1, \ldots, MT_m\) are DL knowledge bases (metatheories) with \(MT_i\) defined over \(\forall \Gamma_i\), with metavocabulary \(\Gamma_i \subseteq \Gamma\).

\[
\forall \exists = \{ \forall DS_n \mid n \in N \} \text{ where each } \forall DS_n \text{ for } n \in N \text{ is a DL knowledge base (dataset) with } \forall DS_n \text{ defined over } \forall \exists_n \text{ for an object vocabulary } \forall \exists_n.
\]

An UKB is a \(\forall SROIQ\) ukb if \(P\), every \(MT_i\) with \(i \in \{1, \ldots, m\}\) and every \(\forall DS_n\) in \(\forall \exists\) are \(\forall SROIQ\) knowledge bases.

**Semantics.** Interpretations for an unified knowledge base follow the two layered structure of its components:

**Definition 2 (UKB interpretation)** A UKB interpretation is a structure \(\forall I = \langle \forall M, \forall \forall \exists \rangle\) such that (i) \(\forall M\) is a DL interpretation of \(\forall \Gamma\) s.t. for any \(n, m \in N\) with \(n \neq m\), \(\forall M_n \neq \forall M_m^m\); (ii) for every \(n \in N\), \(\forall I(n) = \emptyset\) or is a DL interpretation over \(\forall \exists_n\).

Intuitively, an UKB interpretation is composed by an interpretation for the whole metaknowledge layer and a set of knowledge layer interpretations for each of the datasets.

**Definition 3 (UKB model)** An UKB interpretation \(\forall I = \langle \forall M, \forall \forall \exists \rangle\) is a model for an UKB \(\forall \Gamma = (\forall P, \forall \forall \exists)\) (denoted \(\forall I \models \forall \Gamma\)) if, (i) for every \(\alpha \in \forall \forall \exists, \forall M \models \alpha\); (ii) for every \(n \in N\) and \(\alpha \in \forall \exists_n\), if \(\forall I(n) \neq \emptyset\) then \(\forall I(n) \models \alpha\).

The classical reasoning problem of entailment can be adapted to UKB models intuitively by referring to a specific dataset (or to the metalevel). Given an UKB \(\forall \Gamma, \forall n \in N\) and an axiom \(\alpha \in \forall \forall \exists_n\), we say that \(\forall n\)-entailed by \(\forall \exists\) (denoted \(\forall n \models \forall \exists\)) if, for every UKB model \(\forall I = \langle \forall M, \forall \forall \exists \rangle\) of \(\forall \exists\), it holds that \(\forall I(n) \models \alpha\). Similarly, given \(\alpha \in \forall \forall \exists\), we say that \(\alpha\) is entailed by \(\forall \exists\) (denoted \(\forall \exists \models \alpha\)) if, for every UKB model \(\forall I = \langle \forall M, \forall \forall \exists \rangle\) of \(\forall \exists\), it holds that \(\forall M \models \alpha\).

### 4 Meta-theories

In this section we propose two metatheories that can be “imported” in the UKB architecture in order to represent notions of provenance and access control for datasets. The two metatheories encode in our framework the models presented in [11, 7]. For both metatheories we assume that the language of reference is \(\forall SROIQ\). Before introducing these two models, in the following section we provide a metatheory needed for representation of derivability across datasets.

**4.1 A meta-theory for derivability**

Logical reasoning allows to derive new facts from explicitly stated facts. Logical point of view, derived and asserted facts are indistinguishable: they all constitute what is known by an agent. However, considering meta-properties, the derivation of a fact from a set of facts having some meta-properties induces other meta-properties on the derived fact. For instance, the provenance of a fact \(\alpha\) derived from facts \(\beta\) and \(\gamma\) depends on the provenance of \(\beta\) and \(\gamma\). Thus, in order to reason with meta-properties of facts, we need to maintain information on how facts are derived from other facts: in other words, we need a meta-theory for derivability.

We call such meta-theory \(\forall MT_d\). The only predicate of the meta-theory is \(\text{derivedFrom}\), a role linking a dataset \(ds\), with the set of datasets \(ds_1, \ldots, ds_k\) containing the facts used to infer the facts in \(ds\), \(\text{derivedFrom}\) is declared to be transitive.

We impose the following semantic condition on the interpretation of \(\text{derivedFrom}\): we ask that all of the consequences derivable (at instance level) from the contents of combining datasets are contained in the interpretation of the derived dataset. An instantiation of an
axiom $\alpha \in \mathcal{L}_\Sigma$ with a tuple $\mathbf{e}$ of individuals in $\mathcal{N}_\Sigma$, written $\alpha(\mathbf{e})$, is the specialization of $\alpha$, viewed as its first order translation in an universal sentence $\forall \mathbf{x}. \phi_\alpha(\mathbf{x})$, to $\mathbf{e}$ (i.e., $\phi_\alpha(\mathbf{e})$); accordingly, $\mathbf{e}$ is void for assertions, a single element $e$ for GCIs, and a pair $e_1, e_2$ of elements for role axioms\(^3\). Given an UKB $\mathcal{A} = \langle \mathcal{M}, \mathcal{D} \rangle$ such that $\mathit{MT}_\mathcal{d} \in \mathcal{L}$, an UKB interpretation $\mathcal{I} = \langle \mathcal{M}, \mathcal{D} \rangle$ is a model for $\mathcal{A}$ if it is an UKB model and, if $\mathcal{I} \models \mathit{MT}_\mathcal{d} (d_{s_1}, d_{s_j})$ for $j \in \{1, \ldots, k\}$, then either $\mathcal{I}(d_{s_i}) = \emptyset$ or, for every $\alpha \in \mathcal{D}_\Sigma$ and $x^{\mathcal{I}(d_{s_i})} \in \Delta^{\mathcal{I}(d_{s_i})}$, $\mathcal{I}(x^{\mathcal{I}(d_{s_i})}) = \alpha(x)$.

4.2 A meta-theory for provenance

Provenance model. Before developing the meta-theory for provenance we introduce its reference model as proposed in [7]. The idea of this model is to associate single RDF triples to a color representing the source of information from which the triple has been obtained. The approach thus defines a model for the so-called “where provenance” [13]: the interest is to track the source of information and not (as in the “how” provenance) to determine how it has been computed. The color of a triple can be stored either as a fourth component of quadruples, or by grouping all the triples sharing the same color into a named-graph [4] and by associating the color as an attribute of the graph. This second alternative turns out to be more convenient, as information about color for different triples can be factorized in a unique property attached to their graph.

By combining triples coming from different data sources it is possible to logically infer new triples: the goal of the model is to automatically compute the color values for the triples inferred from initial colored triples. To this purpose the model allows to combine colors using the binary operator $+$, so that, for instance, the color of a triple derived from a triple colored with $c_1$ and one colored with $c_2$ is a new “derived color” denoted with $c_1 + c_2$. Formally, colors and combinations of colors are modeled as a color structure \( (\mathcal{I}, +) \) where: (i) $\mathcal{I}$ is a set of colors; (ii) $+$ is a binary operator of composition over $\mathcal{I}$ that is idempotent, commutative and associative. In our application we assume that $\mathcal{I}$ contains a set of “primitive” colors from which one is able to derive all the colors of $\mathcal{I}$. Intuitively, primitive colors are associated to initial data sources. These assumptions are acceptable in the provenance setting since the set of initial data sources is always bound, and inferred triples that need to be colored can be derived only from triples coming from initial data sources. As a consequence the set $\mathcal{I}$ is isomorphic to the set of non empty subsets of primitive colors. So if $c_1, \ldots, c_k$ is the set of primitive colors, any element of $\mathcal{I}$ can be denoted with $c_A$ where $A \neq \emptyset$ and $A \subseteq \{1, \ldots, k\}$.

The colors $c_1$ and $c_2$ that compose $c_{(1,2)} = c_1 + c_2$ are called defining colors for $c_{(1,2)}$. Formally, one triple is colored with the composed color $c_{(1,\ldots,n)}$ iff it has been derived from triples colored by the primitive colors $c_1, \ldots, c_n$.

Encoding provenance metatheory in $\mathit{MT}_\mathcal{pr}$. We encode the model for provenance described above in a $\mathcal{SROIQ}$-RL metatheory called $\mathit{MT}_\mathcal{pr}$. Differently from [7] we adapt the representation to extend the granularity from a single triple to datasets. By doing this we have to adapt the notion of composition of provenance information by inference: we do so by explicitly adding information about composition of datasets using the $\mathit{MT}_\mathcal{d}$ metatheory.

The basic schema of the encoding is shown in Figure 2. Nodes are resource names, squares are color names of type Color. Every dataset is associated with an identifier in $\mathcal{N}$: in the depicted case, the

\[ \begin{align*}
\mathcal{I}(\mathcal{D}_s) = \mathcal{D}_s, \\
\mathcal{I}(\mathcal{D}_{s_j}) = \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \cup \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \cup \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \cap \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \cap \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \setminus \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \setminus \mathcal{D}_{s_j}.
\end{align*} \]

\[ \begin{align*}
\mathcal{I}(\mathcal{D}_s) = \mathcal{D}_s, \\
\mathcal{I}(\mathcal{D}_{s_j}) = \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \cup \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \cup \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \cap \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \cap \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \setminus \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \setminus \mathcal{D}_{s_j}.
\end{align*} \]

\[ \begin{align*}
\mathcal{I}(\mathcal{D}_s) = \mathcal{D}_s, \\
\mathcal{I}(\mathcal{D}_{s_j}) = \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \cup \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \cup \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \cap \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \cap \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \setminus \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \setminus \mathcal{D}_{s_j}.
\end{align*} \]

\[ \begin{align*}
\mathcal{I}(\mathcal{D}_s) = \mathcal{D}_s, \\
\mathcal{I}(\mathcal{D}_{s_j}) = \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \cup \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \cup \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \cap \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \cap \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \setminus \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \setminus \mathcal{D}_{s_j}.
\end{align*} \]

\[ \begin{align*}
\mathcal{I}(\mathcal{D}_s) = \mathcal{D}_s, \\
\mathcal{I}(\mathcal{D}_{s_j}) = \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \cup \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \cup \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \cap \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \cap \mathcal{D}_{s_j}, \\
\mathcal{I}(\mathcal{D}_{s_1} \setminus \mathcal{D}_{s_j}) = \mathcal{D}_{s_1} \setminus \mathcal{D}_{s_j}.
\end{align*} \]

4.3 A meta-theory for access control

Access control model. The model we encode as our metatheory for access control is presented in [11]. The approach is similar to previous model for provenance: the idea is to add access control tokens to triples in order to identify their accessibility properties.

The model is said to represent an abstract access control model: tokens and operators on them are abstract, in the sense that they model the propagation of access control information and their properties without committing to specific representations of tokens and operations. The idea behind the abstract formulation of the model is that it is desirable to abstract from how access control information is computed and only model the formal properties of combination operations: the actual accessibility values can then be computed by defining a suitable mapping to a concrete implementation.

As in previous model, the goal of the model is to describe the propagation of access control information from explicit to inferred triples. The access control label is represented as the fourth component of quadruples. An initial assignment of access control tokens to triples in order to identify their accessibility properties.

The model is said to represent an abstract access control token: tokens and operators on them are abstract, in the sense that they model the propagation of access control information and their properties without committing to specific representations of tokens and operations. The idea behind the abstract formulation of the model is that it is desirable to abstract from how access control information is computed and only model the formal properties of combination operations: the actual accessibility values can then be computed by defining a suitable mapping to a concrete implementation.

As in previous model, the goal of the model is to describe the propagation of access control information from explicit to inferred triples. The access control label is represented as the fourth component of quadruples. An initial assignment of access control tokens to triples in order to identify their accessibility properties.
used to propagate the same label to triples representing e.g. instances of a class that are intended to have the same access control properties of the higher layer.

Formally, the abstract access control model is defined as a structure $\mathcal{M} = (\mathcal{L}, \perp, \ominus, \ominus)$ where: (i) $\mathcal{L}$ is a set of abstract access tokens including the default token $\perp$, a particular token that is associated to triples not in the scope of an authorization rule; (ii) $\ominus$ is the binary inference operator; (iii) $\ominus$ is the unary propagation operator. The inference operator $\ominus$ is commutative and associative, while the propagation operator $\ominus$ is idempotent since it might be useful to define the case in which $a_1 \ominus a_2 \neq a_3$.

Encoding access control metatheory in $\mathcal{M}_{ac}$. We propose a representation as a metatheory, that we denote with $\mathcal{M}_{ac}$, for this access control model. We note that the model can be defined almost analogously to the previous metatheory and it is closely related to the database representation suggested in the original paper. $\mathcal{M}_{ac}$ represents the abstract model from [11], but a concrete model can be defined from it, in particular by fixing concrete values for abstract tokens and mapping to access or deny values for specific classes of users: this information can be easily encoded in the policies knowledge base. In our setting, access control authorizations can be still realized as (SPARQL) queries: on the base of the contents of a dataset, an authorization can assign a particular token to the dataset individual at the metalevel. Moreover, the additional operator for knowledge propagation $\ominus$ can be omitted in our current formulation: this kind of propagation is implicit in the inference "local" to a single dataset.

Syntax. The meta-grammar $\Gamma_{ac} = NC_{ac} \uplus NR_{ac} \uplus NL_{ac}$ contains the following symbols: $\text{AccessToken} \in NC_{ac}$, class of access tokens with default access token $\perp \in NL_{ac}$; $\text{hasLabel} \in NR_{ac}$, role linking a dataset in N to its token in AccessToken; $\text{isComposedFrom} \in NR_{ac}$, role linking a dataset in N to the datasets in N of the tokens it is composed. Language $\mathcal{L}_{ac}$ is a DL language based on $\Gamma_{ac}$ such that: (i) the domains of $\text{hasLabel}$ and $\text{isComposedFrom}$ are mutually disjoint; (ii) $\text{hasLabel}$ is functional.

Semantics. Given an UKB $\mathcal{R} = (\mathcal{M}, \mathcal{D})$ such that $\mathcal{M}_{ac} \in \mathcal{M}$, an UKB interpretation $\mathcal{I} = (\mathcal{M}, \mathcal{I})$ is a model for $\mathcal{R}$ if it is an UKB model and $\mathcal{M} \models \text{isComposedFrom}(d_s, d_i)$ iff $\mathcal{M} \models \text{isComposedFrom}(d_s, d_i)$.

4.4 Example

In the following we detail an example of an UKB using the two presented metatheories: we show the syntactic model of the unified knowledge base and its interpretation.

Inspiring to the motivating example presented in [11], we model a system for the management of information about university staff. Suppose that we have three different datasets:

- $d_s$, containing the official schema of the university organization, with an assigned access token $\alpha_1$ and a provenance color $c_1$;
- $d_{st}$, containing data about one of the students (alice) from the secretary records, with an assigned access token $\alpha_2$ and a provenance color $c_2$;
- $d_{sp}$, containing data about one of the professors (bob) from the secretary records, with an assigned access token $\alpha_3$ and a provenance color $c_2$ (the same of previous dataset);

Moreover, we introduce a new dataset $d_t$ that is used to contain the inferences from the combined sources. Thus, in our case $\mathcal{N} = \{d_s, d_{st}, d_{sp}, d_t\}$.

In the policies, we want to model what is visible from the two users alice and bob. In particular we state that:

- **provenance**: all of the datasets are trusted by the two users.
- **access control**: bob has higher privileges than alice and he can access also $d_s$ dataset (while alice can not access the dataset $d_s$).
- **combination**: we exclude from visible datasets the ones that are untrusted or inaccessible.

We can model this situation (together with the information from the datasets) as a SROIQ-RL unified knowledge base $\mathcal{R}_{uni} = (\mathcal{M}, \mathcal{D})$.

The metalevel part $\mathcal{M} = (P, \{\mathcal{M}_{d}, \mathcal{M}_{ac}, \mathcal{M}_{prov}\})$ is a structure where metatheories are defined as:

$$\mathcal{M}_{ac} = \{ \text{hasLabel}(d_s, a_1), \text{hasLabel}(d_{st}, a_2), \text{AccessToken}(a_1), \\}$$

$$\mathcal{M}_{prov} = \{ \text{hasColor}(d_s, c_1), \text{hasColor}(d_{st}, c_2), \text{hasColor}(d_{sp}, c_2) \}$$

$$\mathcal{M} = \{ \text{derivedFrom}(d_s, d_s), \text{derivedFrom}(d_{st}, d_{st}), \text{derivedFrom}(d_{sp}, d_{sp}) \}$$

Then we can include the following facts and axioms in P to describe the policies for access control:

$$\text{isDeniedLabel}(a_3, alice) \cap \text{hasLabel}(\exists\text{isDeniedLabel}.(alice)) \sqsubseteq \text{DeniedAliceDataset} \cap \text{hasLabel}(\exists\text{isDeniedLabel}.(bob)) \sqsubseteq \text{DeniedBobDataset} \cap \text{isComposedFrom}.\text{DeniedAliceDataset} \sqsubseteq \text{DeniedAliceDataset} \cap \text{isComposedFrom}.\text{DeniedBobDataset} \sqsubseteq \text{DeniedBobDataset}$$

Intuitively, with the first assertion we state that $a_3$ is a denied label for the user alice. Any dataset that has a label that is denied by alice, is then classified as DeniedAliceDataset and similarly for the other user bob. Last two inclusion axioms allow to propagate to datasets that have been composed from denied datasets the property to be denied for that particular user. We remark that while in this example we are defining policies for single users, this modelling can be easily extended to classes of users and labels. A similar set of facts and axioms can be included to define the policies for provenance:

$$\text{hasColor}(\exists\text{isUntrustedColor}.(alice)) \sqsubseteq \text{UntrustedAliceDataset} \cap \text{hasColor}(\exists\text{isUntrustedColor}.(bob)) \sqsubseteq \text{UntrustedBobDataset} \cap \text{isDefiningColor}.\text{UntrustedAliceDataset} \sqsubseteq \text{UntrustedAliceDataset} \cap \text{isDefiningColor}.\text{UntrustedBobDataset} \sqsubseteq \text{UntrustedBobDataset}$$

The combination policy for both users can be stated as:

- DeniedAliceDataset $\sqsubseteq \neg \text{VisibleAliceDataset}$
- UntrustedAliceDataset $\sqsubseteq \neg \text{VisibleAliceDataset}$
- DeniedBobDataset $\sqsubseteq \neg \text{VisibleBobDataset}$
- UntrustedBobDataset $\sqsubseteq \neg \text{VisibleBobDataset}$

At the object layer, $\mathcal{D} = \{DS_s, DS_{st}, DS_{sp}, DS_t\}$ contain:

$$DS_s = \{ \text{Student} \sqsubseteq \text{Person}, \text{Professor} \sqsubseteq \text{Person}, \text{Person} \sqsubseteq \text{Agent}, \ldots \}$$

$$DS_{st} = \{ \text{Student}(alice), \ldots \}$$

$$DS_{sp} = \{ \text{Professor}(bob), \ldots \}$$

$$DS_t = \emptyset$$

This can be depicted as the two level structure in Figure 3. We depict contents of datasets in the knowledge level and relations across individuals in the metalevel, dividing these in the metatheories and policies. Empty diamonds represent dataset names (possibly appearing more than once in the picture for ease of representation).
dotted arrows in the metalevel and gray axioms in the knowledge level represent implicit knowledge that can be obtained by inference. We can now consider the formal interpretation of $\mathcal{R}_\text{ents}$. A model $\mathcal{I} = (\mathcal{M}, \mathcal{Z})$ for $\mathcal{R}_\text{ents}$ must respect the semantic conditions given for models of $\mathcal{MT}_\text{ac}$ and $\mathcal{MT}_\text{prov}$. Note that by these conditions we can derive the following facts about $d_i$ and its associated dataset $DS_i$:

- In the metalevel, for $\mathcal{MT}_\text{prov}$ we have that it holds:
  $$\mathcal{M} \models \text{hasDefiningColor}(d_i, d_a)$$
  $$\mathcal{M} \models \text{hasDefiningColor}(d_i, d_b)$$

- For $\mathcal{MT}_\text{ac}$ we have that:
  $$\mathcal{M} \models \text{id}(d_i, d_a)$$
  $$\mathcal{M} \models \text{id}(d_i, d_b)$$
  $$\mathcal{M} \models \text{id}(d_i, d_3)$$

With respect to policies, it is easy to verify that:

$$\mathcal{M} \models \text{DeniedAliceDataset}(d_b) \quad \mathcal{M} \models \text{DeniedAliceDataset}(d_b)$$
$$\mathcal{M} \models \text{VisibleAliceDataset}(d_a) \quad \mathcal{M} \models \text{VisibleAliceDataset}(d_3)$$

Note that such information can be used to obtain a “projection” of the data content of $\mathcal{R}_\text{ents}$ with respect to the visibility of the datasets for a specific user. Thus for example, the sets of datasets visible from alice can be defined as $\{DS_a \mid n \in \mathbb{N}, \mathcal{M} \models \text{VisibleAliceDataset}(n)\}$. We remark that this basically corresponds to a query to the metalevel to obtain the vision of the knowledge base for a specific user.

## 5 Materialization calculus

In this section we propose a formalization for an inference method in the presented architecture. We base our definitions on the materialization calculus proposed in [3] (which, in turn, is an adaptation to CKB of the calculus presented in [9]) in order to define a reasoning procedure for deciding instance checking in the structure of an UKB.

In our presentation, we consider SROIQ-RL UKBs in normal form that include all of the previously presented metatheories.

**Calculas definition.** We instantiate and adapt the general definition of materialization calculus given by [9] in order to meet the structure of UKBs. Basically, we have to extend the calculus with respect to the two layered structure of unified knowledge bases and introduce specific sets of rules encoding the presented metatheories.

The materialization calculus is composed by the sets of input translations $\mathcal{I}$, deduction rules $\mathcal{P}_\text{deriv}$, $\mathcal{P}_\text{ac-meta}$, $\mathcal{P}_\text{der-obj}$, and output translation $\mathcal{O}$, such that: (i) every input translation $I$ and output translation $O$ is a partial function (defined over axioms in normal form) while deduction rules $P$ are sets of datalog rules; (ii) given an axiom or signature symbol $\alpha$ and $n \in \mathbb{N}$, each $I(\alpha, n)$ is either undefined or a set of datalog facts; (iii) given an axiom $\alpha$ and $n \in \mathbb{N}$, $O(\alpha, n)$ is either undefined or a single datalog fact. We define the translation of input translations to knowledge bases (set of axioms) $S$ with their signature $\Sigma$, with $I(S, n) = \bigcup_{\alpha \in S} I(\alpha, n)$.

SROIQ-RL input $I$ and deduction $\mathcal{P}_\text{ac-meta}$ rules provide the translation and interpretation of SROIQ-RL axioms and signature in normal form. For example, the following rules provide the translation for atomic concept assertions and atomic concept inclusions and the corresponding deduction rule for atomic subsumptions:

$$A(\alpha) \rightarrow \{\text{inst}(\alpha, A, n)\} \quad A \subseteq B \rightarrow \{\text{subClass}(A, B, n)\}$$

Note that w.r.t. the translation of [9], we include an additional parameter identifying the knowledge base of reference. Moreover, ABox assertion predicates have to be divided in their “asserted” (insta, triplae) and “derived” (instd, tripled) form, in order to recognize explicit ABox assertions inside datasets that need to be propagated to their “derived” datasets.

The output rules $O$ provide the translation of “output” ABox assertions that can be proved by applying the rules of the final program. It is composed of the following rules:

$$\text{(o-concept)} \quad A(\alpha) \rightarrow \{\text{inst}(\alpha, A, n)\}$$
$$\text{(o-role)} \quad R(\alpha, b) \rightarrow \{\text{tripled}(\alpha, R, b, n)\}$$

The input rules $I_{\text{meta}}$ encode rules specifically needed for the translation of the metalevel layer: it includes only the following rule assuring the distinction in the interpretation of dataset identifiers.

$$\text{(im-names)} \quad n_1, n_2 \in \mathbb{N}, \text{ with } n_1 \neq n_2 \rightarrow \{\text{neq}(n_1, n_2)\}$$

We provide specific sets of rules for the metatheories $\mathcal{MT}_\text{prov}$, $\mathcal{MT}_\text{ac}$ and $\mathcal{MT}_d$: we have to distinguish the rules that affect the metaknowledge layer of the reasoning ($\mathcal{P}_\text{ac-meta}$ and $\mathcal{P}_\text{der-obj}$) from the ones affecting the propagation of information in the object layer ($\mathcal{P}_\text{der-obj}$). The former rules correspond to the semantic conditions imposed on $\mathcal{MT}_\text{prov}$, $\mathcal{MT}_\text{ac}$, while $\mathcal{P}_\text{der-obj}$ define the propagation of knowledge across “derived” datasets that is implied by derivedFrom assertions in the $\mathcal{MT}_d$ metatheory. These rules basically encode the same deduction rules from $\mathcal{P}_\text{der}$, considering in the body one axiom that might appear in a combined dataset with instances appearing in the target dataset: consequences are then propagated in the “composed” dataset. For instance the rule (ppo-subc) for atomic subsumptions reads as:

$$\text{tripled}(nx, \text{derivedFrom}, n_1, n_2), \text{subClass}(y, z, n_1), \text{instd}(x, y, nx) \rightarrow \text{instd}(x, z, nx)$$

The full set of rules for $I_{\text{en}}, \mathcal{P}_\text{en}, \mathcal{P}_\text{der-obj}, \mathcal{P}_\text{ac-meta}$ and $\mathcal{P}_\text{der-obj}$ are presented in Table 3 in the Appendix.

**Translation process.** We introduce the notion of entailment by defining the “translation process” to produce a program that represents the
complete input UKB. Let \( \mathcal{R} = (\mathcal{M}, \mathcal{D}) \) be an input UKB in normal form. Then, the meta program for its metalevel part \( \mathcal{M} \) is defined as:
\[
PM(\mathcal{M}) = I_{inst}(\mathcal{M}, m) \cup I_{meta}(\mathcal{M}, m) \cup P_{inst} \cup P_{prop-meta} \cup P_{desc-meta}
\]
with \( m \) a new constant identifying the metalevel knowledge base. Intuitively, the meta program translates the contents of policies and all of the metatheories in datalog atoms and includes all of the rule sets needed for reasoning at the meta level.

For every \( n \in \mathbb{N} \), we define the local program for \( DS_n \) as:
\[
PD(n) = I_{inst}(DS_n, m) \cup P_{inst} \cup P_{der-obj}
\]
This means that in every local program we include the translation for local axioms and the rules for local reasoning plus the rules for knowledge propagation to derived datasets. Finally, the UKB program for \( \mathcal{R} \) can be encoded as: \( PK(\mathcal{R}) = PM(\mathcal{M}) \cup \bigcup_{n \in \mathbb{N}} PD(n) \).

We say that \( \mathcal{M} \) entails an axiom \( \alpha \in \mathcal{L}_c \) (denoted \( \mathcal{M} \models \alpha \)) if \( PM(\mathcal{M}) \) and \( O(\alpha, m) \) are defined and \( PM(\mathcal{M}) \models O(\alpha, m) \). We say that \( \mathcal{R} \) entails an axiom \( \alpha \in \mathcal{L}_m \) in a dataset \( DS_n \) with \( n \in \mathbb{N} \) (denoted \( \mathcal{R} \models n : \alpha \)) if the elements of \( PK(\mathcal{R}) \) and \( O(\alpha, n) \) are defined and \( PK(\mathcal{R}) \models O(\alpha, n) \).

Correctness. We can show that the presented rules and translation procedure provide a sound and complete calculus for instance checking (with respect to \( n \)-entailment) in \( SROIQ \). UKB in normal form.

The result can be proved by applying the line of proof used in [9] and [3] to the structure of unified knowledge bases.

**Theorem 1** Given \( \mathcal{R} = (\mathcal{M}, \mathcal{D}) \) a consistent UKB in normal form, \( n \in \mathbb{N} \), and \( \alpha \in \mathcal{L}_m \), an atomic concept or role assertion, then \( \mathcal{R} \models n : \alpha \iff \mathcal{M} \models \alpha \).

## 6 Related works

Our approach has been defined from the need to generalize and unify the reasoning over different meta-level aspects of SW knowledge. By this abstraction, our work can be related to several lines of research in the area of knowledge representation and reasoning.

First of all, as already mentioned, MetaReasons originated from our previous works [12, 3] in contextual reasoning for SW knowledge. The main common point shared with this research area is the clear separation of the two levels of reasoning, about and inside pieces of knowledge. In this area, the reasoning at metalevel is needed to define the situation of validity for object level knowledge and control how knowledge propagates across such situations.

On a larger scope, our work can be also associated to the area of metareasoning [5]. In a nutshell, metareasoning regards the control, at a meta-level, over the strategy for inference at the object level. In our framework, in fact, metalevel information influences reasoning (and querying) and knowledge propagation across datasets in the knowledge layer. For example, as previously remarked (and shown in [2] for the CKR framework), reasoning at the metalevel allows to take a “divide and conquer” approach for querying at the knowledge level, by considering only the fraction of knowledge deemed relevant from its metalevel information. In the context of SW, this form of metareasoning has been also adopted by the CHAINSAW OWL reasoner [15]. CHAINSAW aims at reasoning with very large OWL ontologies by taking a divide and conquer approach w.r.t. the input inference task: intuitively, given an input query, CHAINSAW first extracts an entailment preserving module from the initial ontology; then, it chooses a reasoner that is suitable for treating the module, e.g. depending on module size and language expressivity.

A work closely related to our approach and aiming at the representation of multiple SW metadata is proposed in [14]. The goal is again to enhance query answering by reasoning on a clearly distinct metalevel to focus on parts of an object ontology. The basic observation, common to our work, is that knowledge and metaknowledge have distinct universes of discourse and thus they can be interpreted independently. In [14], metalevel statements are given as annotations over object knowledge axioms. A KB representing the metalevel is then obtained as a “metaview” of such annotations by applying a suitable transformation. Differently from our approach, this work concentrates on querying over a single knowledge level KB. Moreover, annotations are asserted inside the object layer and over statements, while in MetaReasons we consider multiple datasets and metadata information is completely separated from the underlying knowledge.

## 7 Conclusions

In this paper we presented the first steps in the definition of the MetaReasons framework, an approach for reasoning with multiple metadata aspects over SW / LOD datasets. After an overview for the intended structure of the framework, we presented a description logics based formalization for its basic architecture. We then encoded two known models for provenance [7] and access control [11] for SW data as metatheories (i.e. meta-level theories) inside the formal architecture. Finally, we provided a complete materialization calculus for reasoning over knowledge bases that include both metatheories.

One of the next steps in the framework development will consist of its implementation over state of the art tools for management of SW data. The presented materialization calculus will constitute the formal base for the implementation of a forward reasoning procedure over RDF data for our framework. Similarly to the CKR framework [3], the MetaReasons architecture can be implemented by representing the metalevel and the datasets as distinct RDF named graphs. Inference inside (and across) named graphs will be implemented as SPARQL based forward rules. In particular, we plan to use an extension of the Sesame framework that we developed, called SPRINGLES\(^*\), which provides methods to demand an inference materialization over multiple graphs: rules are encoded as SPARQL queries and it is possible to customize their evaluation strategy. In our case, the ruleset will encode the rules of the materialization calculus and the evaluation strategy will follow the calculus translation process. Another parallel activity concerns the definition of new metatheories covering different metalevel aspects. Such models may be already available as, e.g., OWL ontologies that can be readily “plugged in” our framework, or we may encode known models that still need an application inside SW context.

---

---

**References**


---

\(^*\)SPARQL-based Rule Inference over Named Graphs Layer Extending Sesame.
Lemma 1 For every SROIQ-RL knowledge base $K$ over a signature $\Sigma$, a knowledge base $K'$ over an extended signature $\Sigma'$ can be computed s.t. (i) all axioms in $K'$ are in normal form (ii) the size of $K'$ is a linear factor of the size of $K$, (iii) for all axioms $\alpha$ in the signature $\Sigma$, $K \models \alpha$ iff $K' \models \alpha$.

Proof: Intuitively, to prove point (i) we need to show that the rules for normal form transformation provided in Table 2 are complete with respect to the possible SROIQ-RL input axioms; to prove point (ii) one has to show that these translations produce at most a linear increase in the size of the output KB. Finally, to prove point (iii), we have to show that any interpretation satisfying the original axiom can be extended to an interpretation satisfying the transformed axiom set, and, conversely, that any interpretation satisfying the transformed axioms set satisfies the original axiom.

Let $\Sigma$ be the DL vocabulary of $K$ with $\Sigma = NC_2 \sqcup NR_2 \sqcup NI_2$; We extend such vocabulary to $\Sigma'$ in all of its components by adding a distinct set of new symbols: that is $\Sigma' = NC_2 \sqcup NR_2 \sqcup NI_2$ with $NC_2 = NC_2 \uplus X$, $NR_2 = NR_2 \uplus Y$, $NI_2 = NI_2 \uplus Z$. The extended vocabulary is only used to consider the new symbols added in the translation.

We first prove the completeness (i) of the presented normal form translation with respect to the general form, by proving the following claim:

(a) Given a knowledge base $K$ in SROIQ-RL over vocabulary $\Sigma$, then the KB $K'$ over extended vocabulary $\Sigma'$ that is computed by exhaustively applying the rules in Table 2 to axioms in $K$, is in SROIQ-RL and in normal form.

We prove Claim (a) by cases on the possible form of input axioms. Let $\alpha \in \mathcal{A}$ a SROIQ-RL axiom (i.e. following the grammar given in Section 2): we consider all of the possible cases in which $\alpha$ is not already in normal form and show how the rules can be applied to yield a normal form equivalent. In the following we use the same conventions on symbols used in Table 2 (e.g., $C, D$ represent complex concepts while $A, B$ are concept names).

- If $\alpha = C(a)$, by applying the corresponding rule in Table 2 we obtain $S = \{X(a), X \subseteq C\}$. We note that since $X$ is a new concept name, $X(a)$ is in normal form, while $X \subseteq C$ need further expansion (that will be shown as one of the cases below): we remark however that the latter axiom is in SROIQ-RL since by definition concept assertions can be only defined over right concept expressions.

- If $\alpha = C \subseteq D$, by applying the corresponding rule in Table 2 we obtain $S = \{C \subseteq X, X \subseteq D\}$. As in the case above, both $C \subseteq X$ and $X \subseteq D$ need further expansion (treated in the cases below), but the axioms are indeed in SROIQ-RL.

- If $\alpha = C \subseteq A$, then we can recognize the following cases:
  - If $C = A$, $\{\alpha\} \not\models \exists R.T$, then $\alpha$ is already in normal form.
  - If $C = C_1 \sqcup C_2$, then by applying the rule of Table 2 we obtain the set $\{C_1 \sqsubseteq Y_1, C_2 \sqsubseteq Y_2, Y_1 \sqsubseteq Y_2 \subseteq X\}$. All of the axioms are in SROIQ-RL and the first two axioms can be further expanded following the case for $\alpha = C \subseteq A$.
  - If $C = C_1 \sqcup C_2$, then by applying the rule of Table 2 we obtain the set $\{C_1 \not\subseteq A, C_2 \not\subseteq A\}$. All of the axioms are in SROIQ-RL and the two axioms can be then expanded following the case for $\alpha = C \subseteq A$. 

A Appendix: result proofs

A.1 Normal form
If \( C = \emptyset \) or, similarly, if \( C = \emptyset R.\{a\} \), then by applying the rule of Table 2 we obtain the set \( \{C_1 \subseteq X, \exists R.X \subseteq A\} \). The axioms are in \( SROIQ-RL \) and the second axiom is in normal form while the first can be expanded following the case for \( \alpha = C \subseteq A \).

If \( \alpha = A \subseteq D \), then we can recognize the following cases:

If \( D = A, \exists R.\{a\} \) or \( \leq n R.\top \), then \( \alpha \) is already in normal form.

If \( D = \neg C_1 \), then by applying the corresponding rule of Table 2 we obtain the set \( \{A \subseteq \neg X, C_1 \subseteq X\} \). Both axioms are in \( SROIQ-RL \) and the first axiom is in normal form: the second axiom can be expanded following the case for \( \alpha = C \subseteq A \).

If \( D = D_1 \cap D_2 \), then by applying the corresponding rule of Table 2 we obtain the set \( \{A \subseteq D_1, A \subseteq D_2\} \). Both axioms are in \( SROIQ-RL \) and can be further expanded following the case for \( \alpha = A \subseteq D \).

If \( D = \forall R.\top \), then by applying the corresponding rule of Table 2 we obtain the set \( \{A \subseteq \forall R.X, X \subseteq D_1\} \). Both axioms are in \( SROIQ-RL \) and the first is in normal form: the second axiom can be expanded following the case for \( \alpha = A \subseteq D \).

If \( D = \exists 0 R.\top C_1 \), then by consequitively applying the rule of Table 2 relative for axioms \( A \subseteq \exists 0 R.C \) and then the rule for \( A \subseteq \forall R.D \), we obtain the set \( \{X \subseteq \forall R.Y_1, Y_1 \subseteq \neg Y_2, C_1 \subseteq Y_2\} \). It is easy to check that all of the axioms are in \( SROIQ-RL \) and the first two are in normal form: moreover, the last axiom can be further expanded following the case for \( \alpha = C \subseteq A \).

If \( D = \forall 1 R.\top C_1 \), then by applying the corresponding rule of Table 2 we obtain the set \( \{A \subseteq \exists 1 R.X, X \subseteq C_1\} \). Both axioms are in \( SROIQ-RL \) and the first is in normal form while the second can be further expanded following the case for \( \alpha = C \subseteq A \).

If \( \alpha = \text{Sym}(P) \), then by applying the corresponding rule of Table 2 we obtain the set \( \{P \subseteq \forall W.\text{Inv}(P, W)\} \). The axioms are in \( SROIQ-RL \) and in normal form.

If \( \alpha = \text{Trans}(P) \), then by applying the corresponding rule of Table 2 we obtain the set \( \{P \circ P \subseteq P\} \). The axiom is in \( SROIQ-RL \) and in normal form.

If \( \alpha = \text{Asym}(P) \), then by applying the corresponding rule of Table 2 we obtain the set \( \{\text{Dis}(P, W), \text{Inv}(P, W)\} \). Both axioms are in \( SROIQ-RL \) and in normal form.

Point (ii) about the size of resulting knowledge base \( K' \) can be proved by introducing a measure on input axioms from \( K \). Given a concept \( C \) over \( \Sigma \), we define its size \(||C||\) as:

- \(||A|| = 0 \) for \( A \in \text{NC}_C \);  
- \(||\{a\}|| = 0 \) for \( a \in \text{N}_C \);  
- \(||\neg C|| = ||C|| + 1 \);  
- \(||C \circ C_2|| = ||C|| + ||C_2|| + 1 \), for \( C \in \{\neg, \forall\} \);  
- \(||\exists R.C \circ A|| = ||C|| + 1 \), for \( C \in \{\exists, \forall, \forall\} \);

We extend the definition to \( SROIQ-RL \) axioms \( \alpha \) in \( \Sigma_S \) as:

- \(||C \subseteq D|| = ||C|| + ||D|| + 1 \);  
- \(||C(a)|| = ||C|| \);  
- \(||\neg R(a, b)|| = 0 \);  
- \(||\text{char}_1(R)\| = ||\text{char}_2(R, S)|| = 0 \), for \( \text{char}_1 \in \{\text{Sym}, \text{Trans}, \text{Asym}, \text{Irr}\} \) and \( \text{char}_2 \in \{\text{Inv}, \text{Dis}\} \).

The size of sets of axioms is the sum of sizes of all their components. We can prove that every rule in Table 2 adds in the size \(||\Sigma||\) of the resulting set at most a linear increase w.r.t. the size \(||\alpha||\) of the input axiom. This can be easily proved by cases on the rules of Table 2: in the following we show only some examples cases.

- Let \( \alpha = C(a) \), then \(||\alpha|| = ||C|| + ||\Sigma|| = ||C|| + 1 \). Thus \(||\Sigma|| = ||\alpha|| + 1 \).
- Let \( \alpha = C \subseteq D \), then \(||\alpha|| = ||C|| + ||D|| + 1 \) and \(||\Sigma|| = ||C|| + 1 \). Thus \(||\Sigma|| = ||\alpha|| + 1 \).
- Let \( \alpha = \exists \neg C \), then \(||\alpha|| = ||\neg C|| + 1 = ||C|| + 2 \) and \(||\Sigma|| = ||\neg X|| + ||C|| + 1 = ||\Sigma|| + 3 \). Thus \(||\Sigma|| = ||\alpha|| + 1 \).
- Let \( \alpha = \exists C \cup D \cap \exists C \cap D \), then \(||\alpha|| = ||C \cup D|| + 1 = ||C|| + ||D|| + 1 \) and \(||\Sigma|| = ||C|| + 1 + ||D|| + 1 = ||\Sigma|| + ||C|| + ||D|| + 2 \). Thus \(||\Sigma|| = ||\alpha|| + 1 \).

We can now prove the remaining point (iii) of the assertion by showing the validity of the following two claims. For every axiom \( \alpha \) in \( \Sigma_S \), let \( \Sigma \) be the set of axioms obtainable from the application of the corresponding rule in Table 2 to \( \alpha \).

(b) For all axioms \( \alpha \) in \( \Sigma \), for every interpretation \( M \) s.t. \( M \models K \) and \( M \models \alpha \), then there exists an \( \overline{M} \), extension of \( M \) to \( \Sigma \) such that \( \overline{M} \models S \).

Let \( M \) be a DL interpretation on \( \Sigma \) such that \( M \models \Sigma \) and \( M \models \alpha \). We can extend this interpretation to the interpretation \( \overline{M} \) on \( \Sigma \) such that:

- Let \( A \in \text{NC}_S \). If \( A \in \text{NC}_C \) then \( A^\overline{M} = A^M \). Otherwise, if \( A \in \text{NC}_X \) then \( A^\overline{M} \) has been introduced in the translation in an axiom set \( S \); then \( A^\overline{M} \) is the least set of \( d \in \Delta^\overline{M} \) such that \( \overline{M} \models S \).
- Let \( R \in \text{NC}_S \). If \( R \in \text{NC}_X \) then \( R^\overline{M} = R^M \). Otherwise, if \( R \in \text{NC}_W \) then \( R^\overline{M} \) has been introduced in the translation in an axiom set \( S \); then \( R^\overline{M} \) is the least set of \( d \in \Delta^\overline{M} \) such that \( \overline{M} \models S \).

We can show the claim by induction on the form of \( \alpha \) and transformation rules, for example:

- If \( \alpha = C \subseteq D \), then \( S = \{C \subseteq X, X \subseteq D\} \). By hypothesis \( M \models C \subseteq D \), that is \( C^M \subseteq D^M \); by construction we have that \( X^\overline{M} \supseteq C^M \) and \( X^\overline{M} \subseteq D^M \). This implies that \( C^\overline{M} \subseteq D^\overline{M} \) and thus \( \overline{M} \models S \).
Finally, the converse in the condition of point (iii) (i.e. if $K' \models \alpha$ then $K \models \alpha$) is proved by showing the following claim.

(c) For all axioms $\alpha$ in $\Sigma$, for every interpretation $\overline{M}$ s.t. $\overline{M} \models K$ and $\overline{M} \models S$, then $\overline{M}$, the restriction of $\overline{M}$ to $\Sigma$, is such that $\overline{M} \models \alpha$.

We assume the correspondence for the interpretation of $M$ and $\overline{M}$ over the initial vocabulary $\Sigma$ as defined above. We can then again prove the claim proceeding by induction on the form of $\alpha$ and transformation rules, for example:

- If $\alpha = C \subseteq D$, then $S = \{C \subseteq X, X \subseteq D\}$. By hypothesis $\overline{M} \models S$, that is $C^{\overline{M}} \subseteq X^{\overline{M}} \subseteq D^{\overline{M}}$. This implies that $C^{\overline{M}} \subseteq D^{\overline{M}}$ and thus $C^{\overline{M}} \subseteq D^{\overline{M}}$ (since $C, D \in \Sigma$), that is $\overline{M} \models \alpha$.

- If $\alpha = C(a)$, then $S = \{X(a), X \subseteq D\}$. By hypothesis $\overline{M} \models S$, that is $\overline{A} \subseteq X^{\overline{M}}$ and $X^{\overline{M}} \subseteq C^{\overline{M}}$. This implies that $\overline{A} \subseteq C^{\overline{M}}$ and thus $\overline{M} \models \alpha$.

- If $\alpha = C \cap A \subseteq B$, then $S = \{C \subseteq X, X \subseteq A, A \subseteq B\}$. By hypothesis $\overline{M} \models S$, that is $C^{\overline{M}} \subseteq X^{\overline{M}}$ and $X^{\overline{M}} \subseteq A^{\overline{M}} \subseteq B^{\overline{M}}$. This implies that $C^{\overline{M}} \subseteq A^{\overline{M}} \subseteq B^{\overline{M}}$. Hence $C^{\overline{M}} \subseteq A^{\overline{M}} \subseteq B^{\overline{M}}$ (since $C, A, B \in \Sigma$), that is $\overline{M} \models \alpha$.

A.2 Materialization calculus: soundness

Lemma 2 Given $\mathcal{R} = (\mathcal{M}, \mathcal{D})$ an UKR in normal form, and $\alpha \in L_T$ with $\alpha$ an atomic concept or role assertion, then $\mathcal{M} \vdash \alpha$ implies $\mathcal{M} \models \alpha$.

Proof: We can basically follow the same proof schema used for proving soundness in [9], by adapting it to the rules of our calculus.

By definition, we have that $PM(\mathcal{M}) = I_{\text{inst}}(\mathcal{M}, m) \cup I_{\text{meta}}(\mathcal{M}, m) \cup P_{\text{inst}} \cup P_{\text{prov-meta}} \cup P_{\text{arc-meta}}$. We can assign an interpretation to the ground atoms derived from $PM(\mathcal{M})$ as follows:

- $\text{inst}(\alpha, A, m)$ with $a \in NI_T, A \in NC_T$, then $\mathcal{M} \models A(a)$;
- $\text{inst}(\alpha, \text{top}, m)$ with $a \in NI_T$, then $\mathcal{M} \models \top(a)$;
- $\text{inst}(\alpha, \text{bot}, m)$ with $a \in NI_T$, then $\mathcal{M} \models \bot(a)$;
- $\text{tripl}(\alpha, R, b, m)$ with $a, b \in NI_T, R \in NI_R$, then $\mathcal{M} \models R(a, b)$;
- $\text{eq}(\alpha, a, b, m)$ with $a, b \in NI_R$, then $\mathcal{M} \models a = b$;
- $\text{neq}(\alpha, a, b, m)$ with $a, b \in NI_R$, then $\mathcal{M} \models a \neq b$.

We claim that, for any ground atom $H$ of the above form with the corresponding semantic condition $C(H)$, $PM(\mathcal{M}) \models H$ implies $\mathcal{M} \models C(H)$. The proof proceeds by induction on the possible proof tree of the above atoms $H$.

- (prin-inst): then $H = \text{inst}(\alpha, A, m)$ and $PM(\mathcal{M}) \models \text{inst}(\alpha, A, m)$. By the definition of $\text{inst}$, we directly have that $A(a) \in \mathcal{M}$, directly implying that $\mathcal{M} \models A(a)$ as required. The case for (prin-triple) can be shown similarly.
- (prin-ntuple): then $H = \text{ntuple}(\alpha, b, m)$ and $PM(\mathcal{M}) \models \text{ntuple}(\alpha, b, m)$. We have that $\neg R(a, b) \in \mathcal{M}$ and, by the above interpretation of atoms, we obtain that $\mathcal{M} \models R(a, b)$, which is absurd, thus there cannot be an interpretation satisfying $\mathcal{M}$, which justifies the consequence $\mathcal{M} \models \bot(a)$.
- (prin-eq1): then $H = \text{eq}(\alpha, a, m)$ and, by $\text{inst}$ rules, $a \in NI_T$. For any model $\mathcal{M}$ of $\mathcal{M}$, for any $a \in NI_T$ it holds that $a^\mathcal{M} = a^\mathcal{M}$, thus this verifies $\mathcal{M} \models (a = a)$.
- (prin-eq2): then $H = \text{eq}(\alpha, b, m)$ and $PM(\mathcal{M}) \models \text{eq}(\alpha, b, m)$. By the above interpretation of atoms, $\mathcal{M} \models (a = b)$; by symmetry of equality relation this directly implies that $\mathcal{M} \models (b = a)$.
- (prin-ntrupe): then $H = \text{eq}(\alpha, b, m)$ and $PM(\mathcal{M}) \models \text{eq}(\alpha, b, m)$. By the above interpretation of atoms, $\mathcal{M} \models (a = b)$ and $\mathcal{M} \models B(b)$. This directly implies that $\mathcal{M} \models B(b)$, thus proving the assertion.
- (prin-neq1): then $H = \text{eq}(\alpha, a, m)$ and $PM(\mathcal{M}) \models \text{eq}(\alpha, a, m)$. By induction hypothesis and the above semantic conditions, we obtain $\mathcal{M} \models (a = b)$ and $\mathcal{M} \models (a \neq b)$. This is an absurd, thus there cannot be an interpretation satisfying $\mathcal{M}$, which justifies the consequence $\mathcal{M} \models \bot(a)$.
- (prin-top): then $H = \text{inst}(\alpha, \text{top}, m)$ and $PM(\mathcal{M}) \models \text{inst}(\alpha, \text{top}, m)$. By induction hypothesis, $\mathcal{M} \models B(a)$: for every model $\mathcal{M}$ of $\mathcal{M}$, it holds that $a^\mathcal{M} \in \Delta^\mathcal{M} = \top^\mathcal{M}$, thus it is verified that $\mathcal{M} \models \top(a)$.
- (prin-subc): then $H = \text{inst}(\alpha, A, m)$, $A \subseteq B \in \mathcal{M}$ and $PM(\mathcal{M}) \models \text{inst}(\alpha, A, m)$. By the above semantic conditions, $\mathcal{M} \models A(a)$; this directly implies that $\mathcal{M} \models B(a)$.
- (prin-neq1): then $H = \text{eq}(\alpha, a, m)$, $A \subseteq B \in \mathcal{M}$ and $PM(\mathcal{M}) \models \text{eq}(\alpha, a, m)$. By induction hypothesis, $\mathcal{M} \models B(a)$; this is an absurd, since the first consequence would imply that $\mathcal{M} \models (\neg B(a))$. There can not be an interpretation satisfying $\mathcal{M}$, which justifies the consequence $\mathcal{M} \models \bot(a)$.
- (prin-subc): then $H = \text{inst}(\alpha, A, m)$, $A \subseteq B \in \mathcal{M}$ and $PM(\mathcal{M}) \models \text{inst}(\alpha, A, m)$. By the above semantic conditions, $\mathcal{M} \models \top(a)$ and $\mathcal{M} \models C(a)$; this directly implies that $\mathcal{M} \models (\top \cap C)(a)$, and thus $\mathcal{M} \models B(a)$.
- (prin-supex): then $H = \text{tripl}(\alpha, R, b, m)$, $\exists R.A \subseteq B \in \mathcal{M}$ and $PM(\mathcal{M}) \models \text{tripl}(\alpha, R, b, m)$. By induction hypothesis, this implies that $\mathcal{M} \models R(a, b)$ and $\mathcal{M} \models A(b)$; by definition of the semantics, this proves that $\mathcal{M} \models (\exists R.A)(a)$ which implies $\mathcal{M} \models B(a)$.
- (prin-supex): then $H = \text{tripl}(\alpha, R, b, m)$, $\exists R.A \subseteq B \in \mathcal{M}$ and $PM(\mathcal{M}) \models \text{inst}(\alpha, A, m)$. By induction hypothesis, $\mathcal{M} \models A(a)$; this implies that, for every model $\mathcal{M}$ of $\mathcal{M}$, $a^\mathcal{M} \in (\exists R.b)^\mathcal{M}$, that is $(a^\mathcal{M}, b^\mathcal{M}) \in R^\mathcal{M}$. This proves that $\mathcal{M} \models R(a, b)$.
Table 3. Materialization calculus input and deduction rules

RL input translation $I_{\text{in}}(S, n)$

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(prl-nom)</td>
<td>$a \notin N \rightarrow {\text{nom}(a, n)}$</td>
</tr>
<tr>
<td>(prl-cla)</td>
<td>$A \subset NC \rightarrow {\text{cls}(A, n)}$</td>
</tr>
<tr>
<td>(prl-rel)</td>
<td>$R \in NR \rightarrow {\text{rol}(R, n)}$</td>
</tr>
<tr>
<td>(prl-inst)</td>
<td>$A(o) \rightarrow {\text{inst}(A, n)}$</td>
</tr>
<tr>
<td>(prl-triple)</td>
<td>$R(a, b) \rightarrow {\text{triple}(A, R, b, n)}$</td>
</tr>
<tr>
<td>(prl-negtriple)</td>
<td>$\neg R(a, b) \rightarrow {\text{negtriple}(A, R, b, n)}$</td>
</tr>
<tr>
<td>(prl-eq)</td>
<td>$a \rightarrow {\text{eq}(a, n)}$</td>
</tr>
<tr>
<td>(prl-not)</td>
<td>$\neg a \rightarrow {\text{neg}(a, n)}$</td>
</tr>
<tr>
<td>(prl-supex)</td>
<td>$\exists x. \neg a \rightarrow {\text{supEx}(x, a, n)}$</td>
</tr>
<tr>
<td>(prl-subr)</td>
<td>$\forall x. a \rightarrow {\text{subR}(x, a, n)}$</td>
</tr>
<tr>
<td>(prl-subcnj)</td>
<td>$x, y, z, n \rightarrow {\text{subConj}(x, y, z, n)}$</td>
</tr>
<tr>
<td>(prl-supall)</td>
<td>$\forall x. a \rightarrow {\text{supAll}(x, a, n)}$</td>
</tr>
<tr>
<td>(prl-suporall)</td>
<td>$\exists x. a \rightarrow {\text{supOrAll}(x, a, n)}$</td>
</tr>
<tr>
<td>(prl-imp)</td>
<td>$x \rightarrow {\text{imp}(x, n)}$</td>
</tr>
<tr>
<td>(prl-impl)</td>
<td>$\neg x \rightarrow {\text{impl}(x, n)}$</td>
</tr>
</tbody>
</table>

RL deduction rules $P_{\text{rl}}$

- (prl-inst) $\text{inst}(x, z, n) \rightarrow \text{inst}(x, z, n)$
- (prl-triple) $\text{triple}(x, v, y, n) \rightarrow \text{triple}(x, v, y, n)$
- (prl-negtriple) $\neg \text{triple}(x, v, y, n) \rightarrow \text{inst}(x, z, n)$
- (prl-eq) $\text{nom}(x, n) \rightarrow \text{eq}(x, n)$
- (prl-not) $(\neg x, y, n) \rightarrow \text{eq}(x, n)$
- (prl-supex) $(\exists x. \neg a, y, n) \rightarrow \text{eq}(x, n)$
- (prl-subr) $(\forall x. a, y, n) \rightarrow \text{eq}(x, n)$
- (prl-subcnj) $(x, y, z, n) \rightarrow \text{eq}(x, n)$
- (prl-supall) $(\forall x. a, y, z, n) \rightarrow \text{eq}(x, n)$
- (prl-suporall) $(\exists x. a, y, z, n) \rightarrow \text{eq}(x, n)$
- (prl-imp) $(x, y, n) \rightarrow \text{eq}(x, n)$
- (prl-impl) $(\neg x, y, n) \rightarrow \text{eq}(x, n)$

Derivability object deduction rules $P_{\text{obj}}$

- (poo-inst) $\text{triple}(x, y, n) \rightarrow \text{triple}(x, y, n)$
- (poo-triple) $\text{triple}(x, y, n) \rightarrow \text{triple}(x, y, n)$
- (poo-eq) $\text{eq}(x, y, n) \rightarrow \text{eq}(x, y, n)$
- (poo-not) $\neg \text{eq}(x, y, n) \rightarrow \text{eq}(x, y, n)$
- (poo-supex) $\exists x. \neg a \rightarrow \text{eq}(x, y, n)$
- (poo-subr) $\forall x. a \rightarrow \text{eq}(x, y, n)$
- (poo-supall) $\forall x. a \rightarrow \text{eq}(x, y, n)$
- (poo-suporall) $\exists x. a \rightarrow \text{eq}(x, y, n)$
- (poo-imp) $(x, y, n) \rightarrow \text{eq}(x, y, n)$
- (poo-impl) $(\neg x, y, n) \rightarrow \text{eq}(x, y, n)$

Provenance meta deduction rules $P_{\text{prov-meta}}$

- (pom-comp) $\text{triple}(x, y, n) \rightarrow \text{triple}(x, y, n)$
- (pom-comp) $\text{triple}(x, y, n) \rightarrow \text{triple}(x, y, n)$

Access control meta deduction rules $P_{\text{acc-meta}}$

- (pac-comp) $\text{triple}(x, y, n) \rightarrow \text{triple}(x, y, n)$
- (pac-comp) $\text{triple}(x, y, n) \rightarrow \text{triple}(x, y, n)$
– (prln-supforall): then \( H = \text{inst}(b, B, m), A \subseteq R.B \in \mathcal{M} \) and \( \mathcal{M}(\mathcal{P}_0) \models \text{tripled}(a, R, b, m) \). By induction hypothesis, \( \mathcal{M} \models A(a) \) and \( \mathcal{M} \models R(a, b, m) \); this implies that, for every model \( \mathcal{M} \) of \( \mathcal{M}, a^M \in (\forall R.B)^M \), and thus \( b^M \in B^M \). This proves that \( \mathcal{M} \models B(b) \).

– (prln-leqone): then \( H = \text{eq}(b, c, m), A \subseteq 1.R.B \in \mathcal{M} \). Moreover, \( \mathcal{M}(\mathcal{P}_0) \models \text{tripled}(a, R, b, m) \) with \( \mathcal{M}(\mathcal{P}_0) \models \text{instd}(b, B, m) \) and \( \mathcal{M}(\mathcal{P}_0) \models \text{triple}(a, R, c, m) \) with \( \mathcal{M}(\mathcal{P}_0) \models \text{instd}(c, B, m) \). By induction hypothesis, \( \mathcal{M} \models A(a) \) and thus \( \mathcal{M} \models (\leq 1.R.B)(a) \). Moreover, \( \mathcal{M} \models R(a, b) \) and \( \mathcal{M} \models R(a, c) \) with \( \mathcal{M} \models B(b) \) and \( \mathcal{M} \models B(c) \). By definition of the semantics, for every model \( \mathcal{M} \) of \( \mathcal{M} \), it holds that \( b^M = c^M \), which implies \( \mathcal{M} \models (b = c) \).

– (prln-subr): then \( H = \text{tripled}(a, S, b, m), R \subseteq S \in \mathcal{M} \) and \( \mathcal{M}(\mathcal{P}_0) \models \text{tripled}(a, R, b, m) \). By the above semantic constraints, \( \mathcal{M} \models R(a, b) \) which directly implies \( \mathcal{M} \models S(a, b) \).

– (prln-subc): then \( H = \text{tripled}(a, T, b, m), R \subseteq T \in \mathcal{M} \) and \( \mathcal{M}(\mathcal{P}_0) \models \text{tripled}(a, R, c, m) \). By the above semantic constraints, \( \mathcal{M} \models R(a, c) \) and \( \mathcal{M} \models S(c, b) \); by definition of the semantics, this implies that \( \mathcal{M} \models T(a, b) \).

– (prln-dis): then \( H = \text{instd}(a, b, m), \text{Dis}(R, S) \in \mathcal{M} \) and \( \mathcal{M}(\mathcal{P}_0) \models \text{tripled}(a, R, b, m) \). By the above semantic constraints, \( \mathcal{M} \models R(a, b) \) and \( \mathcal{M} \models S(a, b) \); this is an absurd, thus there cannot be an interpretation satisfying \( \mathcal{M} \), which justifies the consequence \( \mathcal{M} \models (\bot.a) \).

– (prln-inv1): then \( H = \text{instd}(a, b, m), \text{Inv}(R, S) \in \mathcal{M} \) and \( \mathcal{M}(\mathcal{P}_0) \models \text{tripled}(b, R, a, m) \). By the above semantic constraints, \( \mathcal{M} \models R(b, a) \); this directly implies that \( \mathcal{M} \models S(a, b) \). The case for (prln-inv2) can be proved similarly.

– (prln-irr): then \( H = \text{instd}(a, b, m), \text{Irr}(R) \in \mathcal{M} \) and \( \mathcal{M}(\mathcal{P}_0) \models \text{tripled}(a, R, a, m) \). By induction hypothesis this would imply that \( \mathcal{M} \models R(a, a) \), which is an absurd: thus there cannot be an interpretation satisfying \( \mathcal{M} \), which justifies the consequence \( \mathcal{M} \models (\bot.a) \).

For rules in \( P_{\text{pre-meta}} \) and \( P_{\text{der-meta}} \) the proof is immediate from the conditions defined by the semantics of meta-theories. For example, in the case of (ppp-comp1), we have \( H = \text{tripled}(dx, \text{hasDefiningColor}, dy, m) \) and \( \mathcal{M}(\mathcal{P}_0) \models \text{tripled}(dx, \text{derivedFrom}, dy, m) \); for the interpretation of atoms we obtain that \( \mathcal{M} \models \text{derivedFrom}(dx, dy) \) and thus, by the semantic condition defined by \( \mathcal{M}_{\text{pre}} \), it holds that \( \mathcal{M} \models \text{hasDefiningColor}(dx, dy) \) as required.

\[\]
PK(\hat{\alpha}) \models \text{inst}(b, A, n). This proves point (i) of the assertion. By rule (prln-eq2), (prln-eq6) and (prln-eq4), we can derive that, if a \cong n b, then PK(\hat{\alpha}) \models \text{triple}(a, R, d, n) iff PK(\hat{\alpha}) \models \text{triple}(b, R, d, n), proving point (ii). Point (iii) can be shown similarly by rules (prln-eq2), (prln-eq6) and (prln-eq5).

Note that the result holds also for the metalevel, consider n = m and \sum = \Gamma. Moreover, note that differently from [3], the case in which n \equiv_m n' for two dataset names n, n' \in N cannot hold. □

Lemma 4

Given \hat{\alpha} = (M, D) a consistent URB in normal form, and \alpha \in L_C with \alpha an atomic concept or role assertion, then M \models \alpha implies M \vdash \alpha.

Proof: The assertion basically can be obtained by adapting the completeness proof presented in [9]. We show by contrapositive that M \nvdash \alpha implies M \not\models \alpha. Assuming M \nvdash \alpha, then by definition it holds that PMA(M) \nmodels O(\alpha, m). Thus there exists an Herbrand model H of PMA(M) such that H \nmodels O(\alpha, m). We show that from H we can build a model M for (\alpha) meeting the semantic conditions for models of the metaknowledge that include M\text{triple}, M\text{sub}, and M\text{eq}\text{ac} such that M \nmodels \alpha, which allow us to derive that M \not\models \alpha.

Let us consider the equivalence relation \equiv_{M} as defined in Lemma 3. We define the equivalence classes \{c | d \equiv_{M} c\}, in order to define the domain of the built interpretation.

Then M = (\Delta^M, \mathcal{M}) is defined as follows:

- \Delta^M = \{e | e \in \text{Nr}\};
- For each e \in \Delta^M, we define the projection function \iota(e) such that, if e = \{c\}, then \iota(e) = b with a fixed b \in \{c\};
- e^M = \{c\} for every e \in \text{Nr};
- A^M = \{d \in \Delta^M | H \models \text{inst}(\iota(d), A, m)\}, for every A \in \text{NC}_C;
- R^M is the smallest set such that \langle d, d' \rangle \in R^M if one of the following holds:
  - H \models \text{triple}(\iota(d), R, \iota(d'), m);
  - S R R \in \text{M} and \langle d, d' \rangle \in S^M;
  - SOT R R \in \text{M} and \langle d, e \rangle \in S^M, \langle e, d' \rangle \in T^M for \langle e, e \rangle \in \Delta^M;
  - Inv(R, S) \in \text{M} or Inv(S, R) \in \text{M} and \langle d', d \rangle \in S^M.

Note that by Lemma 3, the definition of M does not depend on the choice of the \{c\} \in \{c\}. Moreover, note that for every \langle d, d' \rangle \in R^M it holds that H \models \text{triple}(\iota(d), R, \iota(d'), m); this can be shown by an induction on the last three cases of the definition for R^M and the rules (prln-subc), (prln-subrc), (prln-inv1) and (prln-inv2).

It is easy to see that given \alpha \in L_C with H \nmodels O(\alpha, m), then M \nmodels \alpha. For example, if \alpha = A(a), then H \models \text{inst}(a, A, m) which implies by definition that M \nmodels \text{inst}(A, A, m) remains to be shown that M is in fact a model for \text{M}, that is we have to prove that:

(a). for every axiom \beta \in \text{M}, M \models \beta;
(b). for any n, m \in \text{M} with n \neq m, n^M \neq m^M;
(c). M \models \text{inv}(d, S) \iff M \models \text{inv}(S, d);
(d). M \models \text{inv}(d, S).

Point (a) can be proved by cases, considering all of the possible forms of \beta \in \text{M}:

- Let \beta = A(a) \in \text{M}, then H \models \text{inst}(a, A, m) and, by rule (prln-inst), H \models \text{inst}(a, A, m). This directly implies that a^M = [a] \in A^M.
- Let \beta = R(a, b) \in \text{M}, then H \models \text{triple}(a, R, b, m) and, by rule (prln-triple), H \models \text{triple}(a, R, b, m). By definition, we directly have that \{a, b\} \in R^M.
- Let \beta = \neg(R(a, b)) \in \text{M}, then H \models \neg\text{triple}(a, R, b, m). Suppose that \{a, b\} \in R^M, then H \models \text{triple}(\iota(a), R, \iota(b), m). By rule (prln-triple) and Lemma 3, this would imply that H \models \text{inst}(\iota(a), \bot, m) contradicting our assumptions on the consistency of H. Thus \{a, b\} \notin R^M as required.
- Let \beta = (a = b) \in \text{M}, then H \models \text{eq}(a, b, m). By the definition of \cong, it holds that a \equiv_{M} b, thus \{a, b\} \subseteq \{a\} and a^M = b^M = [a].
- Let \beta = (a \neq b) \in \text{M}, then H \models \text{neq}(a, b, m). Suppose that a^M = b^M, then H \models \text{eq}(\iota(a), \iota(b), m). By rule (prln-neq) and Lemma 3, we would obtain that H \models \text{inst}(\iota(a), \bot, m). Again, this contradicts our assumptions on the consistency of H. Thus a^M \neq b^M as required.
- Let \beta = (a \in B) \in \text{M}, then H \models \text{inst}(a, B, m). This case can be proved similarly to the case \beta = A(a).
- Let \beta = A \subseteq B \in \text{M}, then H \models \text{subClass}(A, B, m). If d \in A^M, then by definition H \models \text{inst}(d, A, m) and thus d \in B^M.
- Let \beta = \bot \in \text{M}, then H \models \text{inst}(\bot, m). As in the case for \beta = A(a) (in which case this is subsumed), we directly obtain that a^M = [a] \in T^M.
- Let \beta = \bot \in \text{M}, assuming that M in input is consistent, this case can not subsist as we would directly have that H \models \text{inst}(a, \bot, m), showing the inconsistency of H.
- Let \beta = A \subseteq \neg B \in \text{M}, then H \models \text{supNot}(A, B, m). Suppose that d \in A^M, then H \models \text{inst}(d, A, m). Moreover, suppose that d \in B^M; this implies that H \models \text{inst}(d, B, m). By rule (prln-not) and Lemma 3, we would obtain that H \models \text{inst}(d, \bot, m). This contradicts our assumptions on the consistency of H, thus d \notin B^M as required.
- Let \beta = A_1 \cap A_2 \subseteq B \in \text{M}, then H \models \text{subConj}(A_1, A_2, B, m). If d \in A^M and d \in A_2^M, then by definition H \models \text{inst}(d, A_1, m) and H \models \text{inst}(d, A_2, m). By rule (prln-subc), we directly obtain that H \models \text{inst}(d, B, m); thus d \in B^M as required.
- Let \beta = A \subseteq \exists R.A \in \text{M}, then H \models \text{supEx}(R, A, B, m). Let d \in (\exists R.A)^M; by definition of the semantics this means that there exists d' \in \Delta^M such that \langle d', d \rangle \in R^M. Thus, H \models \text{inst}(d', A, m) and H \models \text{triple}(d', R, \iota(d), m). By rule (prln-supex), we obtain that H \models \text{inst}(d', B, m); thus d \in B^M as required.
- Let \beta = A \subseteq \forall R.B \in \text{M}, then H \models \text{supForall}(A, B, R, m). Let d \in A^M, then H \models \text{inst}(d, A, m). Supposing that there exists d' \in \Delta^M such that \langle d', d \rangle \in R^M, we have that H \models \text{triple}(d', R, \iota(d), m). By rule (prln-supforall) this implies that H \models \text{inst}(d', B, m), thus proving d' \in B^M as required.
- Let \beta = A \subseteq \exists R.B \in \text{M}, then H \models \text{supEqOne}(A, R, B, m). Let d \in A^M, then H \models \text{inst}(d, A, m). Suppose that there exist d_1, d_2 \in \Delta^M such that \langle d_1, d \rangle \in R^M and \langle d_2, d \rangle \in R^M, and \{d_1, d_2\} \subseteq B^M. Thus H \models \{\text{triple}(d, R, \iota(d), m), \text{triple}(d, R, \iota(d), m), \text{inst}(d_1, B, m), \text{inst}(d_2, B, m)\}. 

13
By (prln-leave) rule we obtain that $H \models \text{eq}(\iota(d_1), \iota(d_2), m)$. This implies that $\iota(d_1) \models_n \iota(d_2)$ and thus they are interpreted as the same domain element $d_1 = d_2$ in $M$.

- The cases for $\beta = R \subseteq S, R \circ S \subseteq T$ and Inv$(R, S)$ follow directly from the interpretation of roles in $M$.

- Let $\beta = \text{Dis}(R, S) \in \mathcal{M}$, then $H \models \text{dis}(R, S, m)$. Suppose that $\langle d, d' \rangle \in R^M$ and $\langle d, d' \rangle' \in S^M$. Then $H \models \text{tripled}(\iota(d), R, \iota(d'), m)$ and $H \models \text{tripled}(\iota(d), S, \iota(d'), m)$. By rule (prln-dis) and Lemma 3, we would obtain that $H \models \text{inst}(\iota(d), bot, m)$. This contradicts our assumptions on the consistency of $H$, thus there can not exist a pair $\langle d, d' \rangle \in R^M \cap S^M$ as required.

- Let $\beta = \text{Irr}(R) \in \mathcal{M}$, then $H \models \text{irr}(R, m)$. Suppose that $\langle d, d \rangle \in R^M$, then $H \models \text{tripled}(\iota(d), R, \iota(d), m)$. By rule (prln-irr) and Lemma 3, we would obtain that $H \models \text{inst}(\iota(d), bot, m)$. This contradicts our assumptions on the consistency of $H$, thus $\langle d, d \rangle \notin R^M$ as required.

Point (b) is immediate: considering that by the only rule in $\text{Irr}$ we have that, for every $n \neq n' \in \mathbb{N}$, $\text{eq}(n, n, n') \in H$ then, by the definition of $M$, it holds that $n_1^M \neq n_2^M$ (otherwise, by applying rule (prln-neq), we would obtain inconsistency of the input UKB).

Point (c) can be shown as follows: assuming that $M \models \text{derivedFrom}(ds_1, ds_2)$, we have that $H \models \text{tripled}(\iota(ds_1), \text{derivedFrom}, \iota(ds_2), m)$ and by rule (pmm-comp1), $H \models \text{tripled}(\iota(ds_1), \text{hasDefiningColor}, \iota(ds_2), m)$. By definition of $M$, this directly implies that $M \models \text{hasDefiningColor}(ds_1, ds_2)$. The converse can be proved similarly, considering rule (pmm-comp2).

Also, point (d) can be proved analogously considering rules (pam-comp1) and (pam-comp2).

**Theorem 2 (Completeness)** Given $\mathcal{R} = (\mathcal{M}, \mathcal{D})$ a consistent UKB in normal form, $n \in \mathbb{N}$, and $\alpha \in \mathcal{L}_{\alpha}$ an atomic concept or role assertion then $\mathcal{R} \models n \models \alpha$: implies $\mathcal{R} \models n \models \alpha$.

**Proof:** In the case of soundness, the assertion is proved by extending the previous construction for the metawhole to the structure of the input UKB.

We prove by contrapositive that $\mathcal{R} \not\models n \models \alpha$ implies $\mathcal{R} \not\models n \models \alpha$. If $\mathcal{R} \not\models n \models \alpha$, then we have by definition that $PK(\mathcal{R}) \not\models O(a, n)$. Then there exists an Herbrand model $H$ of $PK(\mathcal{R})$ such that $H \not\models O(a, n)$. As in the lemma above, from this model $H$ for $PK(\mathcal{R})$ we define an UKB model $\mathcal{I} = (\mathcal{M}, \mathcal{T})$ for $\mathcal{R}$ such that $\mathcal{I}(n) \not\models \alpha$, implying that $\mathcal{R} \not\models n \models \alpha$.

We consider again the equivalence relation $\equiv_n$ defined in Lemma 3 and the equivalence classes $[c] = \{d \mid d \equiv_n c\}$ as in the above lemma. Note that the equivalence classes are distinct by the dataset of property of the constants. Then we build $\mathcal{I} = (\mathcal{M}, \mathcal{T})$ as follows: the global interpretation $\mathcal{M} = (\Delta^M, \mathcal{M})$ is a structure defined as in Lemma 4; for each $c \in \Delta^M$, define the projection function $\iota(c)$ such that, if $c = [c']$, then $\iota(c) = b$ with a fixed $b \in [c]$.

For every $n \in \mathbb{N}$, we build $\mathcal{I}(n) = (\Delta_n, \mathcal{Z}(n))$ as follows:

- $\Delta_n = \{\langle d, n \rangle \mid d \in \text{Nk}(\alpha)\}$
- $\mathcal{Z}(n) = \{a \mid \text{for every } a \in \text{Nk}(\alpha)\}$
- $\mathcal{A}(n) = \{d \in \Delta_n \mid H \models \text{inst}(\iota(d), A, n)\}$, for every $A \in \text{NC}(\alpha)$
- $R^2(n)$ is the smallest set such that $(d, d') \in R^2(n)$ if one of the following conditions hold:
  - $H \models \text{tripled}(\iota(d), N, \iota(d'), n)$
  - $S \circ R \subseteq S_n$ and $(d, d') \in S^2(n)$
  - $S \circ R \subseteq S_n$ and $(d, e) \in S^2(n)$, $(e, d') \in T^2(n)$ for $e \in \Delta_n$.
  - Inv$(R, S) \subseteq S_n$ or Inv$(S, R) \subseteq S_n$ and $(d', d) \in S^2(n)$.

As in the above lemma, we can see that, given $H \not\models O(a, n)$, then $\mathcal{I}(n) \not\models \alpha$ as required.

To show the assertion, we have to prove that $\mathcal{I}$ meets the definition of UKB model and that $\mathcal{I} \models \alpha$. By Lemma 4 we directly obtain that the conditions on $\mathcal{M}$ are verified. Thus, it remains to show that, for every $n \in \mathbb{N}$:

(a) $\mathcal{I}(n) \models \beta$, for $\beta \in \text{DS}_n$;
(b) if $M \models \text{derivedFrom}(n, n')$, then, for every $\beta \in \text{DS}_n$ and $x^{(n)} \in \Delta^2(n)$, $\mathcal{I}(n) \models \beta(x)$

Point (a) can be proved by cases on the possible forms of $\beta \in \text{DS}_n$ analogously to the proof for point (a) of Lemma 4. Also point (b) can be verified similarly, considering the possible form of axioms $\beta$ appearing in $\text{DS}_n$ and proving that in all of the case $\mathcal{I}(n) \models \beta(x)$ for $x^{(n)} \in \Delta^2(n)$. In the following we show some of the relevant cases, while the others can be proved with similar reasoning.

- Let $\beta = A(a) \in \text{DS}_n$, then $H \models \text{inst}(a, A, n)$. By the rule (ppo-inst), we directly have that $H \models \text{inst}(a, A, n)$. By definition of $\mathcal{I}(n)$, this implies that $\alpha^{(n)} = a \in A^{(n)}$.
- Let $\beta = (a = b) \in \text{DS}_n$, then $H \models \text{eq}(a, b, n)$. By the rule (ppo-eq), we have that $H \models \text{eq}(a, b, n)$; by definition of $\equiv_n$, it holds that $a \equiv_n b$, thus $(a, b) \subseteq [a]$ and $\alpha^{(n)} = (a, b) = [a] = [b] = [a] = [b] = [a]$.
- Let $\beta = (a \neq b) \in \text{DS}_n$, then $H \models \text{neq}(a, b, n')$. By the rule (prln-neq), we have that $H \models \text{neq}(a, b, n)$; suppose that $\alpha^{(n)} = b^{(n)}$, then $H \models \text{eq}(a, b, n)$, by rule (prln-neq), we would obtain that $H \models \text{inst}(\iota([a]), bot, n)$. This contradicts our assumptions on the consistency of $H$ and thus $\alpha^{(n)} \neq b^{(n)}$ as required.

- Let $\beta = A \sqsubseteq B \in \text{DS}_n$, then $H \models \text{subClass}(A, B, n)$. Let $b^{(n)} \in A^{(n)}$, then by definition of $\mathcal{I}(n)$, $H \models \text{inst}(\iota(b), A, n)$. By the rule (ppo-subc), we obtain that $H \models \text{inst}(\iota(b), B, n)$ and thus $\alpha^{(n)} = b^{(n)}$, that is $\mathcal{I}(n) = B(b)$.
- Let $\beta = A \sqsubseteq \neg B \in \text{DS}_n$, then $H \models \text{supRole}(A, B, n')$. Suppose that $\alpha^{(n)} \in A^{(n)}$, then $H \models \text{inst}(\iota([a]), A, n)$. Moreover, suppose that $\alpha^{(n)} \in B^{(n)}$, this implies that $H \models \text{inst}(\iota([a]), B, n)$. By rule (prln-not), we would obtain that $H \models \text{inst}(\iota([b]), bot, n)$, contradicting our assumption on the consistency of $H$. Thus $\alpha^{(n)} \notin B^{(n)}$ as required.

- Let $\beta = A \sqsubseteq \exists R \in \text{DS}_n$, then $H \models \text{supEx}(A, R, a, n')$. Let $a^{(n)} \in A^{(n)}$, then $H \models \text{inst}(\iota([a]), B, n)$. By rule (ppo-subb), this implies that $H \models \text{tripled}(\iota([a]), B, n)$, by rule (ppo-subb), this implies that $H \models \text{tripled}(\iota([a]), S, \iota([b]), n)$, showing that $\alpha^{(n)} \in B^{(n)}$ as required.

\[\Box\]