Logical Foundations for Data Interlinking with Keys and Link Keys

Manuel Atencia [*], Jérôme David and Jérôme Euzenat
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, F-38000 Grenoble France
E-mails: Manuel.Atencia@inria.fr, Jerome.David@inria.fr, Jerome.Euzenat@inria.fr

Abstract. Both keys and their generalisation, link keys, have been proposed as a means to perform data interlinking, i.e. finding identical resources in different RDF datasets. However, the usage of keys and link keys for data interlinking has not been formalised yet. This is necessary to ensure the correctness of data interlinking tools based on keys or link keys. Furthermore, such a formalisation allows to understand the differences between keys and link keys and to pin down the conditions under which keys and link keys are equivalent. In this paper, we first formalise how keys can be combined with ontology alignments for data interlinking. Then, we extend the definition of a link key by giving the formal semantics of six kinds of link keys: weak, plain and strong link keys, and their in- and eq-variants. Moreover, we establish the conditions under which link keys are equivalent to keys. Finally, we logically ground the usage of these link keys for data interlinking and show that data interlinking with keys and alignments can be reduced to data interlinking with link keys, but not the other way around.

Keywords: data interlinking, keys, ontology alignments, link keys

1. Introduction

Interoperability in linked data largely relies on links between RDF data sets and especially owl:sameAs links asserting the identity of resources bearing different IRIs [1]. Automatically finding such links from RDF data sets is an important and challenging task [2, 3]. Two approaches for doing so are based on finding keys on RDF datasets [4, 5] and link keys across them [6].

Two different kinds of keys have been proposed for interlinking RDF datasets [4, 5]. They are called S-keys and F-keys in [7], which also defines their formal semantics using logical rules. The difference lies in the fact that RDF properties are multivalued. If a set of properties form an S-key for a class, it is enough that two instances of the class share *one* value for each of the properties of the key to infer that they are the same (e.g. email property for AssistantProfessor class). But if the properties form an F-key, the instances must share *all* values (e.g. hasPoem property for PoemAnthology class, since two different poem anthologies may have a poem in common but will unlikely contain exactly the same poems). Therefore, S-keys behave like owl:HasKey statements, while F-keys as keys in relational databases.

If datasets are described using different ontologies, discovered keys need to be combined with ontology alignments, possibly computed by ontology matching tools [8], for interlinking the datasets (e.g. email should be matched with courrierÉlectronique, and AssistantProfessor with MaîtreDeConférences for interlinking datasets of a French university and a British university). How to combine S-keys and F-keys with alignments for data interlinking, though, has not been formalised yet. This is necessary to

[*] Corresponding author.

ensure the correctness of fully integrated data interlinking methods based on key discovery and ontology matching.

A first contribution of this paper is a complete formalisation of the usage of keys and alignments for data interlinking. We do so in the formalism of description logics, which are the base of semantic web languages such as OWL 2. We also make a minor change in the semantics of F-keys and rename S-keys and F-keys as in-keys and eq-keys, respectively. We prove that, given two datasets $\mathcal{D}$ and $\mathcal{D}'$, if a set of properties $\{p_i\}_{i=1}^n$ is an in-key for a class C in $\mathcal{D}$, and we know, thanks to an alignment between the vocabularies of $\mathcal{D}$ and $\mathcal{D}'$, that C subsumes a class D in $\mathcal{D}'$ and each p_i subsumes a property q_i of $\mathcal{D}'$, then it is possible to link instances of C with instances of D by comparing the values of the instances for p_i and q_i. If $\{p_i\}_{i=1}^n$ is an eq-key, though, the properties must be equivalent and a local closed-world assumption made.

Link keys have been proposed as a generalisation of keys and alignments for performing data interlinking [6]. An example of a link key is

$$(\{\langle \mathsf{auteur}, \mathsf{creator}\rangle, \langle \mathsf{titre}, \mathsf{title}\rangle\} \text{ linkkey } \langle \mathsf{Livre}, \mathsf{Book}\rangle)$$

stating that whenever an instance of the class Livre has the same values for properties auteur and titre as an instance of class Book has for properties creator and title, then they denote the same entity. This link key could be used to find links between the datasets of a French and a British libraries.

The formal semantics of link keys has been given in [9]. This semantics is a generalisation of the semantics of in-keys to different RDF datasets. In this paper, we go beyond this definition of a link key, that we now call weak in-link key, and formally define weak eq-link keys. Weak link keys are weak because they are not necessarily made up of keys. We define strong link keys (with their in- and eq-variants) as weak link keys composed of keys. From a practical point of view, both strong and weak link keys allow finding links between entities of different datasets, but strong link keys also allow finding links within each dataset, i.e. identifying duplicates. Additionally, we define plain link keys, which allow finding links between entities of different datasets and duplicates between linked entities only. We show examples of these kinds of link keys in real datasets. We also establish the relations between them: any strong link key is, by definition, a plain link key and any plain link key is a weak link key, and we prove that the classes of eq-plain link keys and eq-weak link keys are the same.

This paper also gives the conditions under which link keys are equivalent to keys. We prove that, when a weak in-link key for a pair of classes is composed of pairs of properties related by a subsumption then the subsumed properties constitute an in-key for the intersection of the classes. In the case of weak eq-link keys, the properties must be equivalent. The same holds for strong link keys and the union of classes.

Finally, we logically ground data interlinking with link keys, and prove that data interlinking with keys and ontology alignments can be reduced to data interlinking with link keys, but not the other way around.

In the remainder, Section 2 presents related work. Section 3 introduces the notations used throughout the paper. Section 4 presents the semantics of in-keys and eq-keys, and Section 5 logically grounds their uses for data interlinking. Section 6 defines weak, plain and strong link keys. The relations between keys and link keys are established in Section 7. Section 8 logically grounds the use of link keys for data interlinking and it relates it to the use of keys. Section 9 concludes the paper and discusses future work.

2. Related Work

2.1. On data interlinking with keys and link keys

Data interlinking refers to the process of finding pairs of IRIs of different RDF datasets representing the same entity [2, 3]. The result of this process is a set of same-as links, specified by owl:sameAs property. Whether to decide that two IRIs represent the same entity or not is mainly based on comparing their values for a selected number of properties. Data interlinking is closely related to the task of record linkage in databases [10].

Link discovery frameworks such as SILK [11, 12] and LIMES [13] enable users to execute — each implementing a different time-efficient execution technique — link specifications to do data interlinking. Link specifications are employed to express the properties to be used for generating owl:sameAs links between two RDF datasets. They also specify the similarity measures to be used for comparing datatype property values, aggregation functions to combine similarity values, and the similarity thresholds beyond which two values are considered equal. Link specifications may be directly set by users or built (semi-)automatically, for example, using machine learning techniques [14, 15]. This paper deals with keys and link keys, which can both be used to build link specifications.

Key-based approaches to data interlinking extract key candidates from RDF datasets and select the most accurate key candidates according to key quality measures [4, 16–18]. Keys can be used to build link specifications or translated into logical rules — which allows to take advantage of reasoning [19, 20] — to perform data interlinking. Key extraction algorithms discover either in-keys [16–18] or eq-keys [4, 21]. When datasets are described with different ontologies then alignments are used, either during the key extraction process or later when doing data interlinking.

The combination of keys and alignments for data interlinking has not received enough attention by the above-cited approaches. For example, the approach proposed in [16] searches in a source dataset for in-keys over classes which are equivalent to classes in a target dataset, and then selects among the discovered in-keys those ones composed of properties which are equivalent to properties of the target dataset. It happens, nonetheless, as we will show in Section 5, that the classes and properties of the source dataset are not required to be equivalent to classes and properties of the target dataset but just to subsume them. This may increase the range of discovered in-key candidates that can be used for data interlinking.

The approaches proposed in [17, 18] are different from [16] and closer to link keys [6]. Indeed, their goal is to discover in-keys that hold in both source and target datasets. It is assumed that both datasets are described using the same vocabulary, possibly resulting from merging different ontologies using an alignment again composed of equivalence correspondences only. In this case, discovered in-keys, although not equal, mostly correspond to strong in-link keys (link keys made up of keys), firstly defined in this paper, and not to weak in-link keys, which are the kind of link keys used in [6]. Also, link keys are not necessarily made up of equivalent properties, as exemplified in Section 6.

Finally, as remarked in [7], and formalised in this paper in Section 4, when an eq-key is used for data interlinking, local completeness for the properties of the eq-key must be assumed. This is not stressed in [4], nor in [21].

2.2. On formalising keys and link keys

The formal semantics of S-keys and F-keys have been given in [7] using logical rules, but the combination of S-keys and F-keys with alignments for data interlinking is not formally addressed. In this paper, we address this using description logics, which are the basis for semantic web languages such as OWL 2. We slightly modify the semantics of F-keys and rename S-keys and F-keys as in-keys and eq-keys, respectively. This enables having a direct correspondence between them: every eq-key is an in-key (but not every F-key is an S-key).

Different approaches to incorporate keys and functional dependencies in description logics have been proposed. Keys may be treated as a new concept constructor [22, 23], or as global constraints in a specific and separate key box (KBox) [24–27], which is the option that will be followed here. The goal of these approaches is to study decidability of reasoning with keys or functional dependencies in description logics. Instead, we use description logics to fully understand the relations between keys, ontology alignments and link keys in the context of a data interlinking task.

The formal semantics of weak in-link keys has been given [9], which introduces a tableau-based algorithm for reasoning with link keys in $\mathcal{ALC}$ description logic. In this paper, we extend this semantics to cover strong link keys, plain link keys, and their in- and eq- variants, and weak eq-link keys.

3. Preliminaries

This section introduces basic notions and notations that will be used throughout the entire paper. We assume that the reader is familiar with the basics of description logics (DLs) [28].

In this paper, ontologies will be the combination of a schema and a dataset, and they will be modelled as knowledge bases in DLs.

Definition 1. *An* ontology *is a knowledge base $\mathcal{O} = \langle \mathcal{S}, \mathcal{D} \rangle$ made up of a terminological box (TBox) $\mathcal{S}$ and an assertional box (ABox) $\mathcal{D}$. $\mathcal{S}$ and $\mathcal{D}$ will be referred to as the* schema *and* dataset *of $\mathcal{O}$, respectively.*

A schema is, therefore, modelled as a set of terminological axioms, i.e. a set of subsumption, equivalence, and disjointness axioms between classes and properties: $C_1 \mathcal{R} C_2$ and $p_1 \mathcal{R} q_2$ where $\mathcal{R} \in \{\sqsubseteq, \sqsupseteq, \equiv, \bot\}$. A dataset is modelled as a set of assertional axioms between individuals: $C(a)$ and $p(a_1, a_2)$. The semantics of ontologies is inherited from the model-theoretic semantics of knowledge bases using DL interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$.

Alignments relate entities — classes, properties and individuals — that belong to different ontologies [8]. Alignment relations between classes and properties are usually subsumption, equivalence and disjointness. In the case of individuals, they are typically related by owl:sameAs property, which express equality of individuals. Alignment statements between classes and properties are referred to as correspondences, whereas equality statements between individuals are called links.

We will also model alignments as knowledge bases. The difference with ontologies is that, in the case of an alignment, the TBox and ABox use two ontologies' vocabularies. In addition, the ABox contains equality assertions only, denoted $a \approx b$.

Definition 2. *Let $\mathcal{O} = \langle \mathcal{S}, \mathcal{D} \rangle$ and $\mathcal{O}' = \langle \mathcal{S}', \mathcal{D}' \rangle$ be two ontologies. An alignment between $\mathcal{O}$ and $\mathcal{O}'$ is a knowledge base $\mathcal{A}_{\mathcal{O},\mathcal{O}'} = \langle \mathcal{C}_{\mathcal{O},\mathcal{O}'}, \mathcal{L}_{\mathcal{O},\mathcal{O}'} \rangle$ where $\mathcal{C}_{\mathcal{O},\mathcal{O}'}$ is composed of class and property axioms $C \mathcal{R} D$ and $p \mathcal{R} q$ where $\mathcal{R} \in \{\sqsubseteq, \sqsupseteq, \equiv, \bot\}$, C and p are class and property expressions in $\mathcal{O}$'s vocabulary and D and q are class and property expressions in $\mathcal{O}'$'s vocabulary, and $\mathcal{L}_{\mathcal{O},\mathcal{O}'}$ is composed of equality assertions $a \approx b$ where a is an individual name in $\mathcal{O}$'s vocabulary and b an individual name in $\mathcal{O}'$'s* *vocabulary. The axioms in $\mathcal{C}_{\mathcal{O},\mathcal{O}'}$ will be referred to as* correspondences *and the axioms in $\mathcal{L}_{\mathcal{O},\mathcal{O}'}$* links. *If no confusion arises, $\mathcal{A}_{\mathcal{O},\mathcal{O}'}$, $\mathcal{C}_{\mathcal{O},\mathcal{O}'}$ and $\mathcal{L}_{\mathcal{O},\mathcal{O}'}$ will be replaced by $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{L}$.*

Different semantics for ontology alignments may be found in the literature [29, 30]. In this paper, though, we will consider the axioms of two ontologies and the correspondences and links of an alignment between them to be part of one single global ontology in which reasoning will be done. Without loss of generality, we can assume that the vocabularies of $\mathcal{O}$ and $\mathcal{O}'$ are disjoint.[1]

Given an ontology $\mathcal{O}$, in what follows, we will use the letters C, p, a and c (possibly with sub- or super-scripts) to denote class and property expressions and individual names of $\mathcal{O}$, respectively, and, in case another ontology $\mathcal{O}'$ is considered, we will use D, q, b and d for $\mathcal{O}'$. In this way, $C_1 \,\mathcal{R}\, C_2$ and $p_1 \,\mathcal{R}\, p_2$ will be used as general axioms in $\mathcal{O}$, while $C \,\mathcal{R}\, D$ and $p \,\mathcal{R}\, q$ as general correspondences in an alignment between $\mathcal{O}$ and $\mathcal{O}'$ ($\mathcal{R} \in \{\sqsubseteq, \sqsupseteq, \equiv, \perp\}$).

4. Two Kinds of Keys in Description Logics

S-keys and F-keys are defined in [7] in terms of rules. Here we reformulate them in the formalism of description logics and deal with keys as axioms to be included in knowledge bases. Instead of S-keys and F-keys, we will speak of in-keys and eq-keys, respectively. The prefixes in- and eq- are shortened forms of intersection and equality. These notations become apparent with the conditions (1) and (2) in Definitions 3 and 4 given below.

In what follows, given a DL interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, a property p, and a domain individual $\delta \in \Delta^{\mathcal{I}}$, $p^{\mathcal{I}}(\delta)$ will denote the set of individuals related to δ through p, i.e. $p^{\mathcal{I}}(\delta) = \{\eta \in \Delta^{\mathcal{I}} : (\delta, \eta) \in p^{\mathcal{I}}\}$.

Definition 3. *An* in-key assertion*, or simply an in-key, has the form*

$$(\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\; C)$$

where $p_1, \ldots, p_k$ are properties and C is a class.

An interpretation $\mathcal{I}$ satisfies $(\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\; C)$ iff, for any $\delta, \delta' \in C^{\mathcal{I}}$,

$$p_1^{\mathcal{I}}(\delta) \cap p_1^{\mathcal{I}}(\delta') \neq \emptyset, \ldots, p_k^{\mathcal{I}}(\delta) \cap p_k^{\mathcal{I}}(\delta') \neq \emptyset \textit{ implies } \delta = \delta'. \tag{1}$$

Definition 4. *An* eq-key assertion*, or simply an eq-key, has the form*

$$(\{p_1, \ldots, p_k\} \text{ key}_{\text{eq}}\; C)$$

where $p_1, \ldots, p_k$ are properties and C is a class.

An interpretation $\mathcal{I}$ satisfies $(\{p_1, \ldots, p_k\} \text{ key}_{\text{eq}}\; C)$ iff, for any $\delta, \delta' \in C^{\mathcal{I}}$,

$$p_1^{\mathcal{I}}(\delta) = p_1^{\mathcal{I}}(\delta') \neq \emptyset, \ldots, p_k^{\mathcal{I}}(\delta) = p_k^{\mathcal{I}}(\delta') \neq \emptyset \textit{ implies } \delta = \delta'. \tag{2}$$

[1] If not, we can consider the disjoint union of the vocabularies.

According to Definition 3, if two instances of a class share one value for each of the properties of an in-key for the class, then we can infer that they are the same instance. More formally,

$$\begin{aligned}
&C(a), \{p_i(a, c_i)\}_{i=1}^k \\
&C(b), \{p_i(b, d_i)\}_{i=1}^k \\
&(\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C) \\
&\qquad \{c_i \approx d_i\}_{i=1}^k \models a \approx b
\end{aligned} \tag{3}$$

On the other hand, according to Definition 4, given an eq-key for a class and two instances of the class, we can infer that they are the same instance if they share *all* values for each of the properties of the key. However, we need to be sure that *all known values* indeed are *all values* the instances may have. More formally,

$$\begin{aligned}
&C(a), \{p_i(a, c_i^1), \ldots, p_i(a, c_i^{r_i})\}_{i=1}^k \\
&\qquad \{\{a\} \sqsubseteq \forall p_i.\{c_i^1, \ldots, c_i^{r_i}\}\}_{i=1}^k \\
&C(b), \{p_i(b, d_i^1), \ldots, p_i(b, d_i^{r_i})\}_{i=1}^k \\
&\qquad \{\{b\} \sqsubseteq \forall p_i.\{d_i^1, \ldots, d_i^{r_i}\}\}_{i=1}^k \\
&\qquad\qquad (\{p_1, \ldots, p_k\} \text{ key}_{\text{eq}}\ C) \\
&\qquad \{c_i^1 \approx d_i^1, \ldots, c_i^{r_i} \approx d_i^{r_i}\}_{i=1}^k \models a \approx b
\end{aligned} \tag{4}$$

Therefore, in contrast to in-keys, eq-keys require some sort of local closed world assumption, which, even though it is generally advised to avoid in the context of the semantic web, it is also expected to be made in certain controlled scenarios. This is illustrated in Example 1.

Compared to [7], the semantics of an in-key corresponds directly to that one of an S-key, but the semantics of an eq-key would correspond to that one of an F-key if condition (2) in Definition 4 was replaced by

$$p_1^{\mathcal{I}}(\delta) = p_1^{\mathcal{I}}(\delta'), \ldots, p_k^{\mathcal{I}}(\delta) = p_k^{\mathcal{I}}(\delta') \text{ implies } \delta = \delta'.$$

Every eq-key is then an F-key, but not the other way around. The prerequisite that the sets of property values must be non-empty allows considering in-keys as a subset of eq-keys (which does not hold between S-keys and F-keys). This result is stated in Proposition 1.

Proposition 1. $(\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C) \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{eq}}\ C)$

Proof. This follows directly from Definitions 3 and 4. □

Conversely, an eq-key is an in-key if it is made up of functional properties. Notice that it is possible to define a functional property as a property p such that for any interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and any $\delta \in \Delta^{\mathcal{I}}$ then $|p^{\mathcal{I}}(\delta)| \leqslant 1$.

Proposition 2. *If $p_1, \ldots, p_k$ are functional, then*

$$(\{p_1, \ldots, p_k\} \text{ key}_{\text{eq}}\ C) \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C)$$

Proof. Let $\mathcal{I}$ be an interpretation such that $\mathcal{I} \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{eq}}\ C)$. Let δ and $\delta' \in C^{\mathcal{I}}$ such that $p_i^{\mathcal{I}}(\delta) \cap p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. Since p_i is functional then $|p_i^{\mathcal{I}}(\delta)| \leqslant 1$ and $|p_i^{\mathcal{I}}(\delta')| \leqslant 1$ $(i = 1, \ldots, k)$. The sets $p_i^{\mathcal{I}}(\delta)$ and $p_i^{\mathcal{I}}(\delta')$ thus contain one or no element, but since their intersection is not empty then they are equal and not empty, i.e. $p_i^{\mathcal{I}}(\delta) = p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. Since $\mathcal{I} \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{eq}}\ C)$ then we can infer that $\delta = \delta'$. This proves that also $\mathcal{I} \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C)$. □

The semantics of owl:HasKey in OWL 2 corresponds to the semantics of an in-key but restricted to being applied to named instances only (thus excluding blank nodes).

Although in-keys and eq-keys have been introduced separately, it is also possible to consider a unified notion of key.

Definition 5. *A* key assertion*, or simply a key, has the form*

$$(\{p_1, \ldots, p_k\}\{q_1, \ldots, q_l\} \text{ key } C)$$

where $p_1, \ldots, p_k$ and $q_1, \ldots, q_l$ are properties, and C is a class.

An interpretation $\mathcal{I}$ satisfies the key $(\{p_1, \ldots, p_k\}\{q_1, \ldots, q_l\} \text{ key } C)$ if, for any $\delta, \delta' \in C^{\mathcal{I}}$,

$$p_1^{\mathcal{I}}(\delta) \cap p_1^{\mathcal{I}}(\delta') \neq \emptyset, \ldots, p_k^{\mathcal{I}}(\delta) \cap p_k^{\mathcal{I}}(\delta') \neq \emptyset \textit{ and}$$
$$q_1^{\mathcal{I}}(\delta) = q_1^{\mathcal{I}}(\delta') \neq \emptyset, \ldots, q_l^{\mathcal{I}}(\delta) = q_l^{\mathcal{I}}(\delta') \neq \emptyset \textit{ implies } \delta = \delta'.$$

From here on, an ontology $\mathcal{O}$ will be a triple $\mathcal{O} = \langle \mathcal{S}, \mathcal{D}, \mathcal{K} \rangle$ which, besides the schema $\mathcal{S}$ (TBox) and dataset $\mathcal{D}$ (ABox), has as a third component a set of keys $\mathcal{K}$ (KBox).

Below we provide examples of in-keys and eq-keys.

Example 1. Insee is a French institution which takes charge of collecting and publishing information about French economy and society. Part of the Insee data is available in the form of RDF triples and can be downloaded as an RDF dump or queried through a SPARQL endpoint.[2] Insee ontologies are available too. In this example and Example 2, we only consider the Insee data related to administrative districts (COG dataset).

The Insee vocabulary comprises four class names for describing the main administrative divisions in France: Commune, Arrondissement, Département and Région. Among the properties of these classes, we find the datatype property nom (used to specify the name of an administrative division), the object property subdivisionDe (to specify that an administrative division is a subdivision of another one, for example, that the commune of Grenoble is a subdivision of Isère department) and the datatype property codeINSEE (which is an identifier for territories, including administrative divisions, and can be thought of the key in the Insee database). The property subdivisionDe is declared to be transitive in the Insee ontology.

[2] http://rdf.insee.fr.

No owl:HasKey axiom is declared in the Insee ontology. Nevertheless, we have checked the in-key and eq-key conditions for the properties and classes mentioned before. We have done so in the RDF graph of Insee extended with the transitivity of subdivisionDe. This generalises to the fully inferred graph as no other axiom of the Insee ontology may have an impact on the satisfiability of the examined key axioms.

As one would expect, codeINSEE is an in-key (and, thus, an eq-key) for Commune, Arrondissement, Région and Département. In symbols,

$$\mathcal{I}^*_{\text{Insee}} \models (\{\mathsf{codeINSEE}\} \text{ key}_{\text{in}} \ \mathsf{Commune})$$
$$\mathcal{I}^*_{\text{Insee}} \models (\{\mathsf{codeINSEE}\} \text{ key}_{\text{in}} \ \mathsf{Arrondissement})$$
$$\mathcal{I}^*_{\text{Insee}} \models (\{\mathsf{codeINSEE}\} \text{ key}_{\text{in}} \ \mathsf{Département})$$
$$\mathcal{I}^*_{\text{Insee}} \models (\{\mathsf{codeINSEE}\} \text{ key}_{\text{in}} \ \mathsf{Région})$$

where $\mathcal{I}^*_{\text{Insee}}$ is the natural DL interpretation of the inferred Insee graph.[3]

Concerning the property nom, it turns out to be an in-key for Région and Département, but neither for Arrondissement nor Commune. Indeed, there exist different communes (and arrondissements) sharing the same name. For instance, Bully may refer to 3 communes: Bully in the department of La Loire, Bully in Rhône and Bully in Seine-Maritime. But there exists no pair of communes of the same department sharing the same name. In fact, nom and subdivisionDe, when put together, form a key for the class Commune. The property subdivisionDe, though, must be treated in the sense of eq-keys. This is because, since subdivisionDe is a transitive property, all French communes share (at least) a value for subdivisionDe, namely, the Insee entity representing the country France. The same holds for the class Arrondissement. In symbols (note that we use the unified notion of a key),

$$\mathcal{I}^*_{\text{Insee}} \models (\{\mathsf{nom}\} \text{ key}_{\text{in}} \ \mathsf{Département})$$
$$\mathcal{I}^*_{\text{Insee}} \models (\{\mathsf{nom}\} \text{ key}_{\text{in}} \ \mathsf{Région})$$
$$\mathcal{I}^*_{\text{Insee}} \models (\{\mathsf{nom}\}\{\mathsf{subdivisionDe}\} \text{ key } \mathsf{Arrondissement})$$
$$\mathcal{I}^*_{\text{Insee}} \models (\{\mathsf{nom}\}\{\mathsf{subdivisionDe}\} \text{ key } \mathsf{Commune})$$

From here on, we will use the shortcuts Reg, Dep, Arr and Com for the corresponding Insee classes.

Proposition 3 shows basic properties of in-keys and eq-keys that will be later used in the proofs of other theorems. In certain occasions, we will write $(\{p_i\}_{i=1}^k \text{ key}_x \ C)$ instead of $(\{p_1, \ldots, p_k\} \text{ key}_x \ C)$ $(x \in \{\text{in}, \text{eq}\})$ to shorten too long expressions. Property (5) is a version of Armstrong's augmentation axiom for functional dependencies on relational databases. Properties (6), (7) and (8) specify how keys behave with subsumption, intersection and union of classes, respectively. Properties (9) and (10) specify how keys behave with subsumption and equivalence of properties. Interestingly, (9) does not hold for eq-keys.

[3] More specifically, this is the interpretation whose domain is made up of all IRIs and literals of the Insee graph (there are no blank nodes), it interprets domain individuals as themselves, and classes and properties as their extensions in the graph.

Proposition 3. *The following holds:*

$$(\{p_1 \ldots, p_k\} \text{ key}_x\ C) \models (\{p_1 \ldots, p_k, p_{k+1}\} \text{ key}_x\ C) \tag{5}$$

$$(\{p_1 \ldots, p_k\} \text{ key}_x\ C), C \sqsupseteq D \models (\{p_1 \ldots, p_k\} \text{ key}_x\ D) \tag{6}$$

$$(\{p_1 \ldots, p_k\} \text{ key}_x\ C) \models (\{p_1 \ldots, p_k\} \text{ key}_x\ C \sqcap D) \tag{7}$$

$$(\{p_1 \ldots, p_k\} \text{ key}_x\ C \sqcup D) \models (\{p_1 \ldots, p_k\} \text{ key}_x\ C) \tag{8}$$

$$(\{p_1 \ldots, p_k\} \text{ key}_{\text{in}}\ C), \{p_i \sqsupseteq q_i\}_{i=1}^k \models (\{q_1 \ldots, q_k\} \text{ key}_{\text{in}}\ C) \tag{9}$$

$$(\{p_1 \ldots, p_k\} \text{ key}_x\ C), \{p_i \equiv q_i\}_{i=1}^k \models (\{q_1 \ldots, q_k\} \text{ key}_x\ C) \tag{10}$$

where $x \in \{\text{in}, \text{eq}\}$.

Proof.

- Properties (5) and (6) follow directly from Definitions 3 and 4.
- Properties (7) and (8) are direct consequences of property (6).
- Let us prove (9). Let $\mathcal{I}$ be an arbitrary DL interpretation such that $\mathcal{I} \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C)$ and $\mathcal{I} \models p_i \sqsupseteq q_i$ $(i = 1, \ldots, k)$. We have to prove that $\mathcal{I} \models (\{q_1, \ldots, q_k\} \text{ key}_{\text{in}}\ C)$. Let $\delta, \delta' \in C^{\mathcal{I}}$ such that $q_i^{\mathcal{I}}(\delta) \cap q_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. Since $\mathcal{I} \models p_i \sqsupseteq q_i$ then $q_i^{\mathcal{I}}(\delta) \subseteq p_i^{\mathcal{I}}(\delta)$ and $q_i^{\mathcal{I}}(\delta') \subseteq p_i^{\mathcal{I}}(\delta')$, and, since $q_i^{\mathcal{I}}(\delta) \cap q_i^{\mathcal{I}}(\delta') \neq \emptyset$, then $p_i^{\mathcal{I}}(\delta) \cap p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. This together with $\mathcal{I} \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C)$ implies $\delta = \delta'$. Therefore, $\mathcal{I} \models (\{q_1, \ldots, q_k\} \text{ key}_{\text{in}}\ C)$.
- Property (10) can be proven analogously.

□

In the following section we formalise how in-keys and eq-keys can be combined with ontology alignments for data interlinking.

5. Data Interlinking with Keys and Alignments

We formulate data interlinking as an inference problem: for two given ontologies $\mathcal{O} = \langle \mathcal{S}, \mathcal{D}, \mathcal{K} \rangle$ and $\mathcal{O}' = \langle \mathcal{S}', \mathcal{D}', \mathcal{K}' \rangle$ equipped with keys (possible discovered with the help of key extraction tools), and an alignment between $\mathcal{O}$ and $\mathcal{O}'$, the problem is to check, for any pair of individual names a and b of $\mathcal{O}$ and $\mathcal{O}'$, respectively, if the following inference is valid:

$$\mathcal{O}, \mathcal{O}', \mathcal{A} \models a \approx b$$

In this section, we provide conditions on the schemas $\mathcal{S}$ and $\mathcal{S}'$, datasets $\mathcal{D}$ and $\mathcal{D}'$, set of class and property correspondences $\mathcal{C}$, and set of (known) links $\mathcal{L}$, that, in the presence of a key in $\lambda \in \mathcal{K}$, are sufficient for inferring a (new) link $a \approx b$. These conditions change depending on whether λ is an in-key or an eq-key, as specified in Theorem 1 and Theorem 2. These two theorems give the logical ground of data interlinking with keys and alignments.

Theorem 1. *Let* $\mathcal{O} = \langle \mathcal{S}, \mathcal{D}, \mathcal{K} \rangle$ *and* $\mathcal{O}' = \langle \mathcal{S}', \mathcal{D}', \mathcal{K}' \rangle$ *be two ontologies and* $\mathcal{A} = \langle \mathcal{C}, \mathcal{L} \rangle$ *an alignment between* $\mathcal{O}$ *and* $\mathcal{O}'$ *such that*

(1) $(\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C) \in \mathcal{K}$*, and*

(2) $\{C \sqsupseteq D\} \cup \{p_i \sqsupseteq q_i\}_{i=1}^{k} \subseteq \mathcal{C}$

Then, for any pair of individual names a and b of $\mathcal{O}$ *and* $\mathcal{O}'$*, respectively, if*

(1) $\{C(a)\} \cup \{p_i(a, c_i)\}_{i=1}^{k} \subseteq \mathcal{D}$,
(2) $\{D(b)\} \cup \{q_i(b, d_i)\}_{i=1}^{k} \subseteq \mathcal{D}'$ *and*
(3) $\{c_i \approx d_i\}_{i=1}^{k} \subseteq \mathcal{L}$

then $\mathcal{O}, \mathcal{O}', \mathcal{A} \models a \approx b$.

Proof. Notice that $C \sqsupseteq D$ and $D(b)$ entail $C(b)$, and that $p_i \sqsupseteq q_i$ and $q_i(b, d_i)$ entail $p_i(b, d_i)$. Then, the statement follows from (3) of Section 4. □

Theorem 1 provides the logical ground of data interlinking with in-keys and alignments: if we know that the properties $p_1, \ldots, p_k$ form an in-key for a class C in $\mathcal{O}$, and that, according to some alignment, C subsumes a class D of $\mathcal{O}'$ and $p_1, \ldots, p_k$ pairwise subsume properties $q_1, \ldots, q_k$ of $\mathcal{O}'$, then, we can infer that an instance a of C is equal to an instance b of D if a has for p_i a value c_i which is equal to a value d_i that b has for q_i ($i = 1, \ldots, k$).

Theorem 2 below provides the logical ground of data interlinking with eq-keys and alignments. Notice that, unlike Theorem 1, $p_1, \ldots, p_k$ have to be pairwise equivalent to $q_1, \ldots, q_k$. Moreover, to infer $a \approx b$, we need to know all the values that a and b may have for p_i and q_i, respectively, and that these values are the same. This local completeness is expressed as axioms in the ontology schemas $\mathcal{S}$ and $\mathcal{S}'$ (items 2 and 4).

Theorem 2. *Let* $\mathcal{O} = \langle \mathcal{S}, \mathcal{D}, \mathcal{K} \rangle$ *and* $\mathcal{O}' = \langle \mathcal{S}', \mathcal{D}', \mathcal{K}' \rangle$ *be two ontologies and* $\mathcal{A} = \langle \mathcal{C}, \mathcal{L} \rangle$ *an alignment between* $\mathcal{O}$ *and* $\mathcal{O}'$ *such that*

(1) $(\{p_1, \ldots, p_k\} \text{ key}_{\text{eq}}\ C) \in \mathcal{K}$ *and*
(2) $\{C \sqsupseteq D\} \cup \{p_i \equiv q_i\}_{i=1}^{k} \subseteq \mathcal{C}$

Then, for any pair of individual names a and b of $\mathcal{O}$ *and* $\mathcal{O}'$*, respectively, if*

(1) $\{C(a)\} \cup \bigcup_{i=1}^{k} \{p_i(a, c_i^j)\}_{j=1}^{r_i} \subseteq \mathcal{D}$,
(2) $\{\{a\} \sqsubseteq \forall p_i.\{c_i^1, \ldots, c_i^{r_i}\}\}_{i=1}^{k} \subseteq \mathcal{S}$,
(3) $\{D(b)\} \cup \bigcup_{i=1}^{k} \{q_i(b, d_i^j)\}_{j=1}^{r_i} \subseteq \mathcal{D}'$,
(4) $\{\{b\} \sqsubseteq \forall q_i.\{d_i^1, \ldots, d_i^{r_i}\}\}_{i=1}^{k} \subseteq \mathcal{S}'$ *and*
(5) $\bigcup_{i=1}^{k} \{c_i^j \approx d_i^j\}_{j=1}^{r_i} \subseteq \mathcal{L}$

then $\mathcal{O}, \mathcal{O}', \mathcal{A} \models a \approx b$.

Proof. Notice that $C \sqsupseteq D$ and $D(b)$ entail $C(b)$, and that $p_i \equiv q_i$ entails $p_i \sqsupseteq q_i$, which along with $q_i(b, d_i^j)$, entails $p_i(b, d_i^j)$. Also, $p_i \equiv q_i$ entails $p_i \sqsubseteq q_i$, which along with $\{b\} \sqsubseteq \forall q_i.\{d_i^1, \ldots, d_i^{r_i}\}$, entails $\{b\} \sqsubseteq \forall p_i.\{d_i^1, \ldots, d_i^{r_i}\}$. Then, the statement follows from (4) of Section 4. □

Notice that in both theorems we only address the case when property values are individuals, i.e. when keys are composed of object properties only. The case when property values are literals, i.e. keys with datatype properties, does not make a difference for our purpose (although, in this case, the comparison of property values is based on similarity functions and not on a initial set of known same-as links $\mathcal{L}$).

Another interesting remark on the theorems is that only one key of $\mathcal{O}$, and no key from $\mathcal{O}'$, is needed for inferring links. Actually, under the assumptions of the theorem, by Proposition 3, $\{q\}_{i=1}^{k}$ is guaranteed to be an in-key (in Theorem 1) or an eq-key (in Theorem 2) for the class D.

In the following sections, we will introduce link keys, relate link keys with keys, and formalise data interlinking with link keys. We will show that data interlinking with link keys is more general than data interlinking with keys and alignments.

6. Link Keys

Link keys have been proposed as a generalisation of keys to different RDF datasets, and, as such, can be used for data interlinking. In [6], an algorithm for extracting link keys from two RDF datasets is described. This algorithm allows discovering a particular kind of link keys whose formal semantics is defined in [9]. In this section, we first recall this semantics, and then extend it to capture other kinds of link keys that can be useful in practice too.

The semantics of link keys considered in [6, 9] generalises the semantics of in-keys, and it is reproduced in Definition 6 below. It is natural to extend this semantics to generalise eq-keys too, and we do so in Definition 7. These link keys will be referred to as *weak link keys*.

Definition 6. *A* weak in-link key assertion*, or simply a weak in-link key, has the form*

$$(\{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{w}}_{\text{in}} \langle C, D\rangle)$$

where $p_1, \ldots, p_k$ and $q_1, \ldots, q_k$ are properties and C and D are classes.

An interpretation $\mathcal{I}$ satisfies $(\{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{w}}_{\text{in}} \langle C, D\rangle)$ iff, for any $\delta \in C^{\mathcal{I}}$ and $\eta \in D^{\mathcal{I}}$,

$$p_1^{\mathcal{I}}(\delta) \cap q_1^{\mathcal{I}}(\eta) \neq \emptyset, \ldots, p_k^{\mathcal{I}}(\delta) \cap q_k^{\mathcal{I}}(\eta) \neq \emptyset \textit{ implies } \delta = \eta.$$

Weak eq-link keys are defined below.

Definition 7. *A* weak eq-link key assertion*, or simply a weak eq-link key, has the form*

$$(\{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{w}}_{\text{eq}} \langle C, D\rangle)$$

where $p_1, \ldots, p_k$ and $q_1, \ldots, q_k$ are properties and C and D are classes.

An interpretation $\mathcal{I}$ satisfies $(\{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{w}}_{\text{in}} \langle C, D\rangle)$ iff, for any $\delta \in C^{\mathcal{I}}$ and $\eta \in D^{\mathcal{I}}$,

$$p_1^{\mathcal{I}}(\delta) = q_1^{\mathcal{I}}(\eta) \neq \emptyset, \ldots, p_k^{\mathcal{I}}(\delta) = q_k^{\mathcal{I}}(\eta) \neq \emptyset \textit{ implies } \delta = \eta.$$

It is noteworthy that any key $(\{p_1, \ldots, p_k\} \text{ key}_x\ C)$ can be expressed as an equivalent weak link key $(\{\langle p_1, p_1\rangle, \ldots, \langle p_k, p_k\rangle\} \text{ linkkey}^{\text{w}}_{x} \langle C, C\rangle)$, where $x \in \{\text{in}, \text{eq}\}$.

Weak link keys are called *weak* because they are not necessarily composed of keys. If this is the case, link keys will be called *strong link keys*. We only give the definition of strong in-link keys, as strong eq-link keys can be defined analogously.

Definition 8. *A* strong in-link key assertion*, or simply a strong in-link key, has the form*

$$(\{\langle p_1, q_1\rangle, \dots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{s}}_{\text{in}} \langle C, D\rangle)$$

where $p_1, \dots, p_k$ and $q_1, \dots, q_k$ are properties and C and D are classes.
An interpretation $\mathcal{I}$ satisfies $(\{\langle p_1, q_1\rangle, \dots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{s}}_{\text{in}} \langle C, D\rangle)$ iff

(1) $\mathcal{I} \models (\{\langle p_1, q_1\rangle, \dots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{w}}_{\text{in}} \langle C, D\rangle)$
(2) $\mathcal{I} \models (\{p_1, \dots, p_k\} \text{ key}_{\text{in}}\ C)$
(3) $\mathcal{I} \models (\{q_1, \dots, q_k\} \text{ key}_{\text{in}}\ D)$

In the definitions above it is not specified to which ontology vocabulary the classes and properties of a link key belong. In practice, the classes C and D, and properties $\{p_i\}_i^k$ and $\{q_i\}_i^k$ of a link key will belong to different ontology schemas, and the instances of C and D to different datasets. This will become explicit in Section 8 when we formalise data interlinking with link keys. Link keys, thus, are the natural generalisation of keys to different datasets, possibly described using different ontologies.

Both strong and weak link keys will allow to find same-as links between two different datasets, but strong link keys will do more. Indeed, since the properties of a strong link key are keys for the classes separately, they can be used to find same-as links within the datasets, i.e. to identify duplicates. Finally, we introduce *plain link keys*, which are less restrictive than strong link keys. A set of property pairs is a plain link key for a pair of classes if it is a weak link key, and, although the properties may not form a key for the classes separately, the key conditions must hold for the instances that will be linked. As before, we only give the definition of a plain in-link key, since plain eq-link keys can be defined similarly. Figure 1 illustrates the differences between weak, plain and strong in-link keys.

Definition 9. *A* plain in-link key assertion*, or simply a plain in-link key, has the form*

$$(\{\langle p_1, q_1\rangle, \dots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{p}}_{\text{in}} \langle C, D\rangle)$$

where $p_1, \dots, p_k$ and $q_1, \dots, q_k$ are properties and C and D are classes.
An interpretation $\mathcal{I}$ satisfies $(\{\langle p_1, q_1\rangle, \dots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{p}}_{\text{in}} \langle C, D\rangle)$ iff, for any $\delta \in C^{\mathcal{I}}$ and $\eta \in D^{\mathcal{I}}$,

$$p_1^{\mathcal{I}}(\delta) \cap q_1^{\mathcal{I}}(\eta) \neq \emptyset, \dots, p_k^{\mathcal{I}}(\delta) \cap q_k^{\mathcal{I}}(\eta) \neq \emptyset \textit{ implies}$$

(1) $\delta = \eta$
(2) for any $\delta' \in C^{I}$, $p_1^{\mathcal{I}}(\delta) \cap p_1^{\mathcal{I}}(\delta') \neq \emptyset, \dots, p_k^{\mathcal{I}}(\delta) \cap p_k^{\mathcal{I}}(\delta') \neq \emptyset$ implies $\delta = \delta'$
(3) for any $\eta' \in D^{I}$,$q_1^{\mathcal{I}}(\eta) \cap q_1^{\mathcal{I}}(\eta') \neq \emptyset, \dots, q_k^{\mathcal{I}}(\eta) \cap q_k^{\mathcal{I}}(\eta') \neq \emptyset$ implies $\eta = \eta'$

As we have done for keys in Definition 5, it is possible to define unified notions of a weak, plain and strong link keys, bringing together the in- and eq-conditions:

$$(\{\langle p_i, q_i\rangle\}_{i=1}^{k}\{\langle r_j, s_j\rangle\}_{j=1}^{l} \text{ linkkey}^{y} \langle C, D\rangle)$$

where $y \in \{\text{w}, \text{p}, \text{s}\}$.

Alignments may be naturally extended to include a set of link keys. From here on, given two ontologies $\mathcal{O} = \langle \mathcal{S}, \mathcal{D}, \mathcal{K}\rangle$ and $\mathcal{O}' = \langle \mathcal{S}', \mathcal{D}', \mathcal{K}'\rangle$, equipped with keys, an alignment $\mathcal{A}$ between $\mathcal{O}$ and $\mathcal{O}'$ will be a

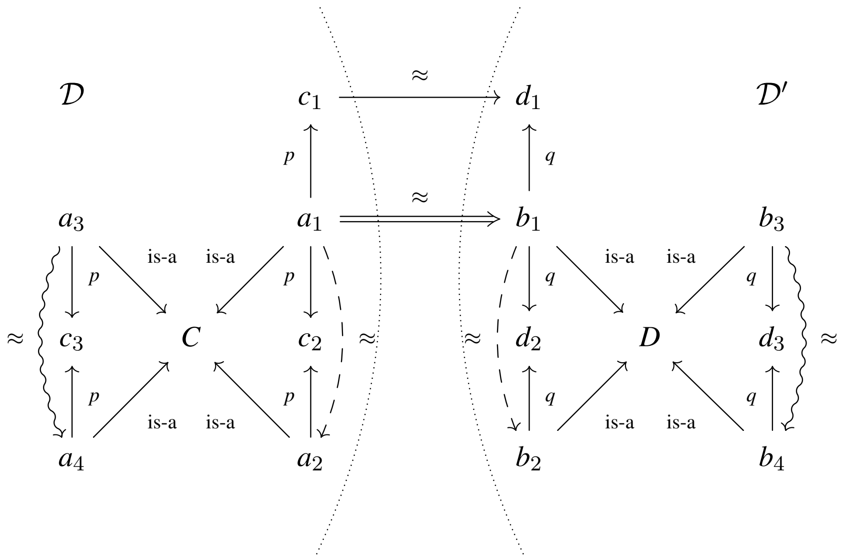

Fig. 1. depicts the following two datasets and initial set of links between them: $\mathcal{D} = \{p(a_1,c_1), p(a_1,c_2), C(a_1), p(a_2,c_2), C(a_2), C(a_3), p(a_3,c_3), C(a_4), p(a_4,c_3)\}$, $\mathcal{D}' = \{q(b_1,d_1), q(b_1,d_2), D(b_1), q(b_2,d_2), D(b_2), D(b_3), q(b_3,d_3), D(b_4), q(b_4,d_3)\}$ and $\mathcal{L} = \{c_1 \approx d_1\}$. If $\{\langle p,q\rangle\}$ is a weak in-link key for $\langle C,D\rangle$ then the link $a_1 \approx b_1$ will be inferred (double-line arrow). If $\{\langle p,q\rangle\}$ is a plain in-link key then $a_1 \approx a_2$ and $b_1 \approx b_2$ will be inferred too (dashed arrows). If $\{\langle p,q\rangle\}$ is a strong in-link key then $a_3 \approx a_4$ and $b_3 \approx b_4$ will be inferred too (wave arrows).

triple $\mathcal{A} = \langle \mathcal{C}, \mathcal{L}, \mathcal{LK}\rangle$ which, apart from a set of class and property correspondences $\mathcal{C}$ and a link set $\mathcal{L}$, has a set $\mathcal{LK}$ of link keys between the vocabularies of $\mathcal{O}$ and $\mathcal{O}'$ as a third component.

Below we give examples of link keys in real datasets.

Example 2. The Insee dataset includes links to the IGN dataset (French National Geographic Institute).[4] There exist owl:sameAs links between the resources representing the French communes, arrondissements, departments and regions, gathered together in the two datasets using the same class names. These links can be found by comparing the Insee codes, which are declared in both datasets — using the ins:codeINSEE property in the Insee dataset and ign:numInsee in the IGN dataset.[5]

We have checked the different link key conditions for the property pair $\langle$ins:codeINSEE, ign:numInsee$\rangle$ on the union of Insee and IGN datasets taking into account the existing owl:sameAs links. It happens to be a strong in-link key for the class pairs $\langle$ins:Com, ign:Com$\rangle$, $\langle$ins:Arr, ign:Arr$\rangle$, $\langle$ins:Dép, ign:Dép$\rangle$ and $\langle$ins:Rég, ign:Rég$\rangle$. In symbols:

$$\mathcal{I}^* \models (\{\langle \text{ins:codeINSEE}, \text{ign:numInsee}\rangle\} \text{ linkkey}^{\text{s}}_{\text{in}} \langle \text{ins:Com}, \text{ign:Com}\rangle)$$

$$\mathcal{I}^* \models (\{\langle \text{ins:codeINSEE}, \text{ign:numInsee}\rangle\} \text{ linkkey}^{\text{s}}_{\text{in}} \langle \text{ins:Arr}, \text{ign:Arr}\rangle)$$

[4] http://data.ign.fr

[5] The ign prefix replaces the namespace http://data.ign.fr/def/geofla#.

$$\mathcal{I}^* \models (\{\langle \textsf{ins:codeINSEE}, \textsf{ign:numInsee}\rangle\} \text{ linkkey}^{\text{s}}_{\text{in}} \langle \textsf{ins:Dép}, \textsf{ign:Dép}\rangle)$$

$$\mathcal{I}^* \models (\{\langle \textsf{ins:codeINSEE}, \textsf{ign:numInsee}\rangle\} \text{ linkkey}^{\text{s}}_{\text{in}} \langle \textsf{ins:Rég}, \textsf{ign:Rég}\rangle)$$

where $\mathcal{I}^*$ is a canonical interpretation of the RDF graph resulting from the union of the Insee and IGN datasets where the linked individuals are merged.

Let us consider the other properties of Example 1. The property rdfs:label is used in the IGN dataset in the same way as ins:nom is used in the Insee dataset. Instead of ins:subdivisionDe, however, IGN uses the three properties ign:arr, ign:dpt and ign:region to declare the arrondissement, department and region an administrative unit belongs to. We have checked the different link key conditions for the combinations of these properties in the scope of the class pairs ⟨ins:Com, ign:Com⟩, ⟨ins:Arr, ign:Arr⟩, ⟨ins:Dép, ign:Dép⟩ and ⟨ins:Rég, ign:Rég⟩. We have done so in the graph resulting from the union of the Insee graph, extended by transitivity of subdivisionDe, and the IGN graph, and again considering the owl:sameAs links. This generalises to the fully inferred RDF graph, as no other axiom of neither the Insee ontology nor the IGN ontology may have an impact on the satisfiability of the examined link key axioms. As one would expect, the property pair ⟨ins:nom, rdfs:label⟩ is a strong in-link key for ⟨ins:Dép, ign:Dép⟩ and ⟨ins:Rég, ign:Rég⟩. The property pairs ⟨ins:subdivisionDe, ign:arr⟩ and ⟨ins:subdivisionDe, ign:dpt⟩ together with ⟨ins:nom, rdfs:label⟩ constitute weak (and plain) in-link keys for the class pairs ⟨ins:Com, ign:Com⟩ and ⟨ins:Arr, ign:Arr⟩, respectively. They are not strong link keys because, as explained in Example 1, subdivisionDe must be used as an eq-key. And they are not eq-link keys because ign:arr and ign:dpt refer each of them to one administrative unit only. In symbols,

$$\mathcal{I}^* \models (\{\langle \textsf{ins:nom}, \textsf{rdfs:label}\rangle\}, \langle \textsf{ins:subdivisionDe}, \textsf{ign:arr}\rangle\} \text{ linkkey}^{\text{w}}_{\text{in}} \langle \textsf{ins:Com}, \textsf{ign:Com}\rangle)$$

$$\mathcal{I}^* \models (\{\langle \textsf{ins:nom}, \textsf{rdfs:label}\rangle\}, \langle \textsf{ins:subdivisionDe}, \textsf{ign:dpt}\rangle\} \text{ linkkey}^{\text{w}}_{\text{in}} \langle \textsf{ins:Arr}, \textsf{ign:Arr}\rangle)$$

$$\mathcal{I}^* \models (\{\langle \textsf{ins:nom}, \textsf{rdfs:label}\rangle\} \text{ linkkey}^{\text{s}}_{\text{in}} \langle \textsf{ins:Dép}, \textsf{ign:Dép}\rangle)$$

$$\mathcal{I}^* \models (\{\langle \textsf{ins:nom}, \textsf{rdfs:label}\rangle\} \text{ linkkey}^{\text{s}}_{\text{in}} \langle \textsf{ins:Rég}, \textsf{ign:Rég}\rangle)$$

where $\mathcal{I}^*$ is a canonical interpretation of the before-mentioned RDF graph where the linked individuals are merged.

Obviously, the above link keys could be used for rediscovering the links.

In what follows, we provide theoretical results stating the relationships between the different kinds of link keys.

Propositions 4 and 5 are the counterparts of Propositions 1 and 2 for link keys and can be proven similarly.

Proposition 4. *The following holds:*

$$(\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}^{y}_{\text{in}} \langle C, D\rangle) \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}^{y}_{\text{eq}} \langle C, D\rangle)$$

where $y \in \{\text{w}, \text{p}, \text{s}\}$.

Proposition 5. *If* $p_1, \ldots, p_k$ *and* $q_1, \ldots, q_k$ *are functional then*

$$(\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}^{y}_{\text{eq}} \langle C, D\rangle) \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}^{y}_{\text{in}} \langle C, D\rangle)$$

where $y \in \{\mathrm{w}, \mathrm{p}, \mathrm{s}\}$.

Proposition 6 shows the relations between weak link keys, plain link keys and strong link keys: a strong link key is always a plain link key, which is always a weak link key. Interestingly, there is no distinction between weak eq-link keys and plain eq-link keys. This is due to transitivity of equality.

Proposition 6. *The following holds:*

$$(\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_x^{\mathrm{s}} \langle C, D\rangle) \models (\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_x^{\mathrm{p}} \langle C, D\rangle)$$
$$(\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_x^{\mathrm{p}} \langle C, D\rangle) \models (\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_x^{\mathrm{w}} \langle C, D\rangle)$$
$$(\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_{\mathrm{eq}}^{\mathrm{w}} \langle C, D\rangle) \models (\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_{\mathrm{eq}}^{\mathrm{p}} \langle C, D\rangle)$$

where $x \in \{\mathrm{in}, \mathrm{eq}\}$.

Proof. We only prove the validity of the third inference as the others follow direclty from the definitions of link keys. Let $\mathcal{I}$ be a DL interpretation such that $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_{\mathrm{eq}}^{\mathrm{w}} \langle C, D\rangle)$, and let us prove that $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_{\mathrm{eq}}^{\mathrm{p}} \langle C, D\rangle)$. Let $\delta \in C^{\mathcal{I}}$ and $\eta \in D^{\mathcal{I}}$ such that $p_i^{\mathcal{I}}(\delta) = q_i^{\mathcal{I}}(\eta) \neq \emptyset$ $(i = 1, \ldots, k)$. Since $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_{\mathrm{eq}}^{\mathrm{w}} \langle C, D\rangle)$, then $\delta = \eta$. Now, let $\delta' \in C^{\mathcal{I}}$ with $p_i^{\mathcal{I}}(\delta) = p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. From $p_i^{\mathcal{I}}(\delta) = q_i^{\mathcal{I}}(\eta) \neq \emptyset$ and $p_i^{\mathcal{I}}(\delta) = p_i^{\mathcal{I}}(\delta') \neq \emptyset$, we can infer that $p_i^{\mathcal{I}}(\delta') = q_i^{\mathcal{I}}(\eta) \neq \emptyset$ $(i = 1, \ldots, k)$. This together with $\delta' \in C^{\mathcal{I}}$, $\eta \in D^{\mathcal{I}}$ and $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_{\mathrm{eq}}^{\mathrm{w}} \langle C, D\rangle)$ implies $\delta' = \eta$, and, since $\delta = \eta$, then $\delta = \delta'$. The last condition of plain eq-link keys can be proven analogously. □

In the following section, we provide theoretical results stating the relation between link keys and keys.

7. Relation between Keys and Link Keys

This section studies the relation between keys and link keys. Theorem 3 states the relation between weak link keys and keys, while Theorem 4 and Theorem 5 state the relation between strong link keys and keys.

The relation between weak link keys and keys is given by Theorem 3. It states that if a weak in-link key for a pair of classes is composed of pairs of properties related by a subsumption then the subsumed properties form an in-key for the intersection of the classes. The same holds for weak eq-link keys if the properties of the link key are equivalent.

Theorem 3. *The following holds:*

$$(\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_{\mathrm{in}}^{\mathrm{w}} \langle C, D\rangle), \{p_i \sqsubseteq q_i\}_{i=1}^{k} \models (\{p_i\}_{i=1}^{k} \text{ key}_{\mathrm{in}}\, C \sqcap D)$$
$$(\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_{\mathrm{in}}^{\mathrm{w}} \langle C, D\rangle), \{p_i \sqsupseteq q_i\}_{i=1}^{k} \models (\{q_i\}_{i=1}^{k} \text{ key}_{\mathrm{in}}\, C \sqcap D)$$
$$(\{\langle p_i, q_i\rangle\}_{i=1}^{k} \text{ linkkey}_{\mathrm{eq}}^{\mathrm{w}} \langle C, D\rangle), \{p_i \equiv q_i\}_{i=1}^{k} \models (\{p_i\}_{i=1}^{k} \text{ key}_{\mathrm{eq}}\, C \sqcap D)$$

Proof. Let us prove the first entailment. Let $\mathcal{I}$ such that $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{w}} \langle C, D\rangle)$ and $\mathcal{I} \models p_i \sqsubseteq q_i$ $(i = 1, \ldots, k)$, and let us prove that $\mathcal{I} \models (\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C \sqcap D)$. Let $\delta, \delta' \in (C \sqcap D)^{\mathcal{I}}$ such that $p_i^{\mathcal{I}}(\delta) \cap p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. Since $\delta, \delta' \in (C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$ then $\delta, \delta' \in C^{\mathcal{I}}$ and $\delta, \delta' \in D^{\mathcal{I}}$. In particular, $\delta \in C^{\mathcal{I}}$ and $\delta' \in D^{\mathcal{I}}$. Now, since $\mathcal{I} \models p_i \sqsubseteq q_i$, then, $p_i^{\mathcal{I}}(\delta') \subseteq q_i^{\mathcal{I}}(\delta')$ $(i = 1, \ldots, k)$. From this and the fact that $p_i^{\mathcal{I}}(\delta) \cap p_i^{\mathcal{I}}(\delta') \neq \emptyset$, we can infer that $p_i^{\mathcal{I}}(\delta) \cap q_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. Since $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{w}} \langle C, D\rangle)$ and $\delta \in C^{\mathcal{I}}$ and $\delta' \in D^{\mathcal{I}}$, then $\delta = \delta'$. The second entailment can be proven analogously.

Let us prove the third entailment. Let $\mathcal{I}$ such that $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{eq}}^{\text{w}} \langle C, D\rangle)$ and $\mathcal{I} \models p_i \equiv q_i$ $(i = 1, \ldots, k)$, and let us prove that $\mathcal{I} \models (\{p_i\}_{i=1}^k \text{ key}_{\text{eq}}\ C \sqcap D)$. Let $\delta, \delta' \in (C \sqcap D)^{\mathcal{I}}$ such that $p_i^{\mathcal{I}}(\delta) = p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. Since $\delta, \delta' \in (C \sqcap D)^{\mathcal{I}}$ then $\delta \in C^{\mathcal{I}}$ and $\delta' \in D^{\mathcal{I}}$. Now, since $\mathcal{I} \models p_i \equiv q_i$, then, we have $p_i^{\mathcal{I}}(\delta') = q_i^{\mathcal{I}}(\delta')$ $(i = 1, \ldots, k)$. From this and the fact that $p_i^{\mathcal{I}}(\delta) = p_i^{\mathcal{I}}(\delta') \neq \emptyset$, we can infer that $p_i^{\mathcal{I}}(\delta) = q_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. Finally, since $\delta \in C^{\mathcal{I}}$ and $\delta' \in D^{\mathcal{I}}$ and $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{eq}}^{\text{w}} \langle C, D\rangle)$ then it must be $\delta = \delta'$. □

Theorem 4 is the counterpart of Theorem 3 for strong link keys. Notice that this time the consequent is a key in the union of classes, and not only in the intersection.

Theorem 4. *The following holds:*

$$(\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{s}} \langle C, D\rangle), \{p_i \sqsubseteq q_i\}_{i=1}^k \models (\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C \sqcup D)$$

$$(\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{s}} \langle C, D\rangle), \{p_i \sqsupseteq q_i\}_{i=1}^k \models (\{q_i\}_{i=1}^k \text{ key}_{\text{in}}\ C \sqcup D)$$

$$(\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{eq}}^{\text{s}} \langle C, D\rangle), \{p_i \equiv q_i\}_{i=1}^k \models (\{p_i\}_{i=1}^k \text{ key}_{\text{eq}}\ C \sqcup D)$$

Proof. We only prove the first entailment. Let $\mathcal{I}$ such that $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{s}} \langle C, D\rangle)$ and $\mathcal{I} \models p_i \sqsubseteq q_i$ $(i = 1, \ldots, k)$, and let us prove that $\mathcal{I} \models (\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C \sqcup D)$. Let $\delta, \delta' \in (C \sqcup D)^{\mathcal{I}}$ such that $p_i^{\mathcal{I}}(\delta) \cap p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. We have $\delta, \delta' \in (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$. Let us consider three cases: (1) $\delta, \delta' \in C^{\mathcal{I}}$, (2) $\delta, \delta' \in D^{\mathcal{I}}$ and (3) $\delta \in C^{\mathcal{I}}$ and $\delta' \in D^{\mathcal{I}}$ (the case $\delta' \in C^{\mathcal{I}}$ and $\delta \in D^{\mathcal{I}}$ is equivalent to this last one).

(1) Assume that $\delta, \delta' \in C^{\mathcal{I}}$. Since $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{s}} \langle C, D\rangle)$ then $\mathcal{I} \models (\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C)$. From this and the fact that $\delta, \delta' \in C^{\mathcal{I}}$ and $p_i^{\mathcal{I}}(\delta) \cap p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$, we can conclude that $\delta = \delta'$.

(2) Assume that $\delta, \delta' \in D^{\mathcal{I}}$. Since $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{s}} \langle C, D\rangle)$ then $\mathcal{I} \models (\{q_i\}_{i=1}^k \text{ key}_{\text{in}}\ D)$. Now, we also have that $\mathcal{I} \models p_i \sqsubseteq q_i$. Thus, $p_i^{\mathcal{I}}(\delta) \subseteq q_i^{\mathcal{I}}(\delta)$ and $p_i^{\mathcal{I}}(\delta') \subseteq q_i^{\mathcal{I}}(\delta')$ $(i = 1, \ldots, k)$. From this, and $p_i^{\mathcal{I}}(\delta) \cap p_i^{\mathcal{I}}(\delta') \neq \emptyset$, we can infer that $q_i^{\mathcal{I}}(\delta) \cap q_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. This along with the fact that $\delta, \delta' \in D^{\mathcal{I}}$ and $\mathcal{I} \models (\{q_i\}_{i=1}^k \text{ key}_{\text{in}}\ D)$ implies $\delta = \delta'$.

(3) Assume $\delta \in C^{\mathcal{I}}$, $\delta' \in D^{\mathcal{I}}$. Since $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{s}} \langle C, D\rangle)$ then $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{w}} \langle C, D\rangle)$. It is possible to proceed like in the proof of the first statement of Theorem 3 to conclude that $\delta = \delta'$.

The other two statements can be proven similarly. □

Theorem 5 is the converse of Theorem 4. Notice, however, that, in the case of in-link keys, the subsumptions are inverted, i.e. they are the subsuming and not the subsumed properties the ones that must form an in-key in the union of classes.

Theorem 5. *The following holds:*

$$(\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C \sqcup D), \{p_i \sqsupseteq q_i\}_{i=1}^k \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{s}}\ \langle C, D\rangle)$$

$$(\{q_i\}_{i=1}^k \text{ key}_{\text{in}}\ C \sqcup D), \{p_i \sqsubseteq q_i\}_{i=1}^k \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{s}}\ \langle C, D\rangle)$$

$$(\{p_i\}_{i=1}^k \text{ key}_{\text{eq}}\ C \sqcup D), \{p_i \equiv q_i\}_{i=1}^k \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{eq}}^{\text{s}}\ \langle C, D\rangle)$$

Proof. We only prove the first inference. Let $\mathcal{I}$ be an interpretation such that $\mathcal{I} \models (\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C \sqcup D)$ and $\mathcal{I} \models p_i \sqsupseteq q_i$ $(i = 1, \ldots k)$.

Since $\mathcal{I} \models (\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C \sqcup D)$, by (8) of Proposition 3, we have that $\mathcal{I} \models (\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C)$.

Let us prove $\mathcal{I} \models (\{q_i\}_{i=1}^k \text{ key}_{\text{in}}\ D)$. Since $\mathcal{I} \models (\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C \sqcup D)$, by (8) of Proposition 3, we have $\mathcal{I} \models (\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ D)$, and, since $\mathcal{I} \models p_i \sqsupseteq q_i$, by (9) of Proposition 3, we also have that $\mathcal{I} \models (\{q_i\}_{i=1}^k \text{ key}_{\text{in}}\ D)$.

Finally, let us prove that $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{w}}\ \langle C, D\rangle)$. Let $\delta \in C^{\mathcal{I}}$ and $\delta' \in D^{\mathcal{I}}$ with $p_i^{\mathcal{I}}(\delta) \cap q_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. From $\delta \in C^{\mathcal{I}}$ and $\delta' \in D^{\mathcal{I}}$ we have $\delta, \delta' \in C^{\mathcal{I}} \cup D^{\mathcal{I}} = (C \sqcup D)^{\mathcal{I}}$. Since $\mathcal{I} \models p_i \sqsupseteq q_i$, we have $q_i^{\mathcal{I}}(\delta') \subseteq p_i^{\mathcal{I}}(\delta')$ $(i = 1, \ldots, k)$. From this and $p_i^{\mathcal{I}}(\delta) \cap q_i^{\mathcal{I}}(\delta') \neq \emptyset$ we infer $p_i^{\mathcal{I}}(\delta) \cap p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. This together with $\delta, \delta' \in (C \sqcup D)^{\mathcal{I}}$ and $\mathcal{I} \models (\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C \sqcup D)$ implies $\delta = \delta'$.

The second entailment can be proven analogously. The third entailment can be proven analogously too, but will use (10) of Proposition 3. □

In the following section, we logically ground the use of link keys for data interlinking and compare it with the use of keys described in Section 5.

8. Data interlinking with link keys

Theorems 6 and 7 give the logical grounds of data interlinking with weak in-link keys and eq-link keys, respectively. Their proofs follow the same ideas and techniques that we have used so far in the paper, and are omitted.

Theorem 6. *Let $\mathcal{O} = \langle \mathcal{S}, \mathcal{D}, \mathcal{K}\rangle$ and $\mathcal{O}' = \langle \mathcal{S}', \mathcal{D}', \mathcal{K}'\rangle$ be two ontologies, and $\mathcal{A} = \langle \mathcal{C}, \mathcal{L}, \mathcal{LK}\rangle$ an alignment between $\mathcal{O}$ and $\mathcal{O}'$ such that*

- $(\{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\} \text{ linkkey}_{\text{in}}^{\text{w}}\ \langle C, D\rangle) \in \mathcal{LK}$

Then, for any pair of individual names a and b of $\mathcal{O}$ and $\mathcal{O}'$, respectively, if

- $\{C(a)\} \cup \{p_i(a, c_i)\}_{i=1}^k \subseteq \mathcal{D}$,
- $\{D(b)\} \cup \{q_i(b, d_i)\}_{i=1}^k \subseteq \mathcal{D}'$ *and*
- $\{c_i \approx d_i\}_{i=1}^k \subseteq \mathcal{L}$

then $\mathcal{O}, \mathcal{O}', \mathcal{A} \models a \approx b$.

Theorem 7. *Let $\mathcal{O} = \langle \mathcal{S}, \mathcal{D}, \mathcal{K}\rangle$ and $\mathcal{O}' = \langle \mathcal{S}', \mathcal{D}', \mathcal{K}'\rangle$ be two ontologies, and $\mathcal{A} = \langle \mathcal{C}, \mathcal{L}, \mathcal{LK}\rangle$ an alignment between $\mathcal{O}$ and $\mathcal{O}'$ such that*

- $(\{\langle p_1, q_1\rangle, \dots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{w}}_{\text{eq}} \langle C, D\rangle) \in \mathcal{LK}$

Then, for any pair of individual names a and b of $\mathcal{O}$ and $\mathcal{O}'$, respectively, if

- $\{C(a)\} \cup \bigcup_{i=1}^{k} \{p_i(a, c_i^j)\}_{j=1}^{r_i} \subseteq \mathcal{D}$,
- $\{\{a\} \sqsubseteq \forall p_i.\{c_i^1, \dots, c_i^{r_i}\}\}_{i=1}^{k} \subseteq \mathcal{S}$,
- $\{D(b)\} \cup \bigcup_{i=1}^{k} \{q_i(b, d_i^j)\}_{j=1}^{r_i} \subseteq \mathcal{D}'$,
- $\{\{b\} \sqsubseteq \forall q_i.\{d_i^1, \dots, d_i^{r_i}\}\}_{i=1}^{k} \subseteq \mathcal{S}'$ *and*
- $\bigcup_{i=1}^{k} \{c_i^j \approx d_i^j\}_{j=1}^{r_i} \subseteq \mathcal{L}$

then $\mathcal{O}, \mathcal{O}', \mathcal{A} \models a \approx b$.

Notice first that, in order to perform data interlinking, weak link keys are sufficient. Plain and strong link keys can be used, in a similar way, to also infer equality statements between individuals of the same dataset.

The difference between using link keys for data interlinking instead of keys and alignments becomes clear when comparing Theorem 1 with Theorem 6 and Theorem 2 with Theorem 7. In both cases, knowledge about keys and alignment correspondences ($\mathcal{K}$ and $\mathcal{C}$) is replaced by knowledge about link keys ($\mathcal{LK}$). More specifically, a link key replaces a key ($\{p_1, \dots, p_k\} \text{ key}_{\text{in}}\, C$ or $\{p_1, \dots, p_k\} \text{ key}_{\text{eq}}\, C$) together with class and property correspondences ($C \sqsupseteq D$ and $\{p_i \sqsupseteq q_i\}_{i=1}^{k}$ or $\{p_i \equiv q_i\}_{i=1}^{k}$).

But we can say more: data interlinking with link keys is *more general* than data interlinking with keys and alignments. By this, we mean that data interlinking with keys and alignments can be reduced to data interlinking with link keys, but not the other way around. Indeed, under the conditions of Theorems 1 and 2, the property pairs $\{\langle p_i, q_i\rangle\}_{i=1}^{k}$ form a strong link key for the class pair $\langle C, D\rangle$. This is now easy to prove: by Theorem 5 and the fact that $C \sqsupseteq D$ implies that C is equivalent to $C \sqcup D$, we have

$$(\{p_1, \dots, p_k\} \text{ key}_{\text{in}}\, C), \{C \sqsupseteq D\}, \{p_i \sqsupseteq q_i\}_{i=1}^{k} \models (\{\langle p_1, q_1\rangle, \dots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{s}}_{\text{in}} \langle C, D\rangle)$$
$$(\{p_1, \dots, p_k\} \text{ key}_{\text{eq}}\, C), \{C \sqsupseteq D\}, \{p_i \equiv q_i\}_{i=1}^{k} \models (\{\langle p_1, q_1\rangle, \dots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{s}}_{\text{eq}} \langle C, D\rangle)$$

However, as stated in Theorems 6 and 7, to perform data interlinking, weak link keys are sufficient, and weak link keys are not necessarily made up of properties that form keys separately, nor related by alignments.

The fact that data interlinking with weak link keys cannot always be reduced to data interlinking with keys and alignments can also be exemplified using the following weak in-link key of Example 2:

$$(\{\langle \mathsf{ins{:}nom}, \mathsf{rdfs{:}label}\rangle\}, \langle \mathsf{ins{:}subdivisionDe}, \mathsf{ign{:}arr}\rangle\} \text{ linkkey}^{\text{w}}_{\text{in}} \langle \mathsf{ins{:}Com}, \mathsf{ign{:}Com}\rangle)$$

Although the properties of this weak link key do form keys for the classes separately, these properties are related by opposite subsumptions:

$$\mathsf{ins{:}nom} \sqsubseteq \mathsf{rdfs{:}label}$$
$$\mathsf{ins{:}subdivisionDe} \sqsupseteq \mathsf{ign{:}arr}$$

and, therefore, the application of Theorem 1 is not possible.

9. Conclusions and Further Work

This paper provides the logical foundations for data interlinking with keys and link keys. For this purpose, it has considered the semantics of in-keys and eq-keys and has generalised it to six kinds of link keys. The relations between these kinds of link keys have been established, and their practical relevance have been demonstrated with real-world data interlinking scenarios.

The provided formalisation states the conditions under which each kind of key and link key can be used for data interlinking. Moreover, it relates link keys to keys, and specifies when data interlinking with link keys is equivalent to data interlinking with keys and ontology alignments. It turns out that link keys may not be composed of separate keys, and that data interlinking with keys and alignments can be reduced to data interlinking with link keys but not the other way around.

This formalisation contributes grounding data interlinking methods based on keys and link keys, and it should lead to more powerful data interlinking methods. For example, algorithms can be proposed to discover and exploit the proposed new kinds of link keys, which can allow generating more links between datasets, and finding duplicates inside the same datasets. Also, by having defined their formal semantics, new link keys could be exploited by extended versions of existing reasoning-based data interlinking tools.

In practice, link keys may have exceptions and yet be useful for data interlinking. As future work, we plan to define the formal semantics of *pseudo* link keys, and formalise their use for data interlinking. For this goal, we will extend the semantics of weighted ontology alignments proposed in [31].

Acknowledgements

This work has been partially supported by the ANR French project Elker (ANR-17-CE23-0007-01).

Manuel Atencia and Jérôme David acknowledge the numerous discussions they had with Michel Chein, Madalina Croitoru, Michel Leclère, Nathalie Pernelle, Fatiha Saïs, François Scharffe and Danai Symeonidou on the various definitions of keys that may apply in logical formalisms.

References

[1] T. Heath and C. Bizer, *Linked Data : Evolving the Web into a Global Data Space*, Morgan and Claypool, 2011.

[2] A. Ferrara, A. Nikolov and F. Scharffe, Data Linking for the Semantic Web, *International Journal of Semantic Web and Information Systems* **7**(3) (2011), 46–76.

[3] M. Nentwig, M. Hartung, A.-C. Ngonga Ngomo and E. Rahm, A survey of current Link Discovery frameworks, *Semantic Web* **8**(3) (2017), 419–436. doi:10.3233/SW-150210.

[4] M. Atencia, J. David and F. Scharffe, Keys and pseudo-keys detection for web datasets cleansing and interlinking, in: *Knowledge Engineering and Knowledge Management - 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings*, Lecture Notes in Computer Science, Vol. 7603, Springer, 2012, pp. 144–153.

[5] N. Pernelle, F. Saïs and D. Symeounidou, An Automatic Key Discovery Approach for Data Linking, *Journal of Web Semantics* **23** (2013), 16–30.

[6] M. Atencia, J. David and J. Euzenat, Data interlinking through robust linkkey extraction, in: *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, Frontiers in Artificial Intelligence and Applications, Vol. 263, IOS Press, 2014, pp. 15–20.

[7] M. Atencia, M. Chein, M. Croitoru, J. David, M. Leclère, N. Pernelle, F. Saïs, F. Scharffe and D. Symeonidou, Defining Key Semantics for the RDF Datasets: Experiments and Evaluations, in: *Graph-Based Representation and Reasoning - 21st International Conference on Conceptual Structures, ICCS 2014, Iaşi, Romania, July 27-30, 2014, Proceedings*, Lecture Notes in Computer Science, Vol. 8577, Springer, 2014, pp. 65–78.

[8] J. Euzenat and P. Shvaiko, *Ontology matching*, 2nd edn, Springer, Heidelberg (DE), 2013.

[9] M. Gmati, M. Atencia and J. Euzenat, Tableau extensions for reasoning with link keys, in: *Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016.*, CEUR Workshop Proceedings, CEUR-WS.org, 2016, pp. 37–48.

[10] P. Christen, *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Data-Centric Systems and Applications, Springer, 2012.

[11] J. Volz, C. Bizer, M. Gaedke and G. Kobilarov, Discovering and Maintaining Links on the Web of Data, in: *The Semantic Web - ISWC 2009, 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings*, Lecture Notes in Computer Science, Vol. 5823, Springer, 2009, pp. 650–665.

[12] R. Isele, A. Jentzsch and C. Bizer, Efficient Multidimensional Blocking for Link Discovery without losing Recall, in: *Proceedings of the 14th International Workshop on the Web and Databases 2011, WebDB 2011, Athens, Greece, June 12, 2011*, 2011.

[13] A.N. Ngomo and S. Auer, LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data, in: *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, IJCAI/AAAI, 2011, pp. 2312–2317.

[14] A.N. Ngomo and K. Lyko, EAGLE: Efficient Active Learning of Link Specifications Using Genetic Programming, in: *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*, Lecture Notes in Computer Science, Vol. 7295, Springer, 2012, pp. 149–163.

[15] M.A. Sherif, A.N. Ngomo and J. Lehmann, Wombat - A Generalization Approach for Automatic Link Discovery, in: *The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 10249, Springer, 2017, pp. 103–119.

[16] D. Symeonidou, V. Armant, N. Pernelle and F. Saïs, SAKey: Scalable Almost Key Discovery in RDF Data, in: *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference,*, Lecture Notes in Computer Science, Vol. 8796, Springer, 2014, pp. 33–49.

[17] M. Achichi, M.B. Ellefi, D. Symeonidou and K. Todorov, Automatic Key Selection for Data Linking, in: *Knowledge Engineering and Knowledge Management - 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings*, Vol. 10024, Lecture Notes in Computer Science, 2016, pp. 3–18.

[18] H. Farah, D. Symeonidou and K. Todorov, KeyRanker: Automatic RDF Key Ranking for Data Linking, in: *Proceedings of the Knowledge Capture Conference, K-CAP 2017, Austin, TX, USA, December 4-6, 2017*, ACM, 2017, pp. 7–178.

[19] A. Hogan, A. Zimmermann, J. Umbrich, A. Polleres and S. Decker, Scalable and Distributed Methods for Entity Matching, Consolidation and Disambiguation over Linked Data Corpora, *Web Semantics: Science, Services and Agents on the World Wide Web* **10**(0) (2012), 76–110.

[20] M. Al-Bakri, M. Atencia, J. David, S. Lalande and M. Rousset, Uncertainty-sensitive reasoning for inferring sameAs facts in linked data, in: *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, Frontiers in Artificial Intelligence and Applications, Vol. 285, IOS Press, 2016, pp. 698–706.

[21] T. Soru, E. Marx and A.-C. Ngonga Ngomo, ROCKER – A Refinement Operator for Key Discovery, in: *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, ACM, 2015.

[22] A. Borgida and G. Weddell, Adding Uniqueness Constraints to Description Logics (Preliminary Report), in: *Deductive and Object-Oriented Databases, 5th International Conference, DOOD'97, Montreux, Switzerland, December 8-12, 1997, Proceedings*, Lecture Notes in Computer Science, Vol. 1341, Springer, 1997, pp. 85–102.

[23] D. Toman and G. Weddell, On Keys and Functional Dependencies as First-Class Citizens in Description Logics, *Journal of Automated Reasoning* **40**(2–3) (2008), 117–132.

[24] D. Calvanese, G. De Giacomo and M. Lenzerini, Keys for Free in Description Logics, in: *Proceedings of the 2000 International Workshop on Description Logics (DL2000), Aachen, Germany, August 17-19, 2000*, CEUR Workshop Proceedings, CEUR-WS.org, 2000, pp. 79–88.

[25] D. Calvanese, G. De Giacomo and M. Lenzerini, Identification Constraints and Functional Dependencies in Description Logics, in: *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001*, Morgan Kaufmann, 2001, pp. 155–160.

[26] C. Lutz, C. Areces, I. Horrocks and U. Sattler, Keys, Nominals, and Concrete Domains, *Journal of Artificial Intelligence Research* **23** (2005), 667–726.

[27] C. Lutz and M. Milicic, Description Logics with Concrete Domains and Functional Dependencies, in: *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*, IOS Press, 2004, pp. 378–382.

[28] S. Rudolph, Foundations of Description Logics, in: *Reasoning Web. Semantic Technologies for the Web of Data - 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures*, LNCS, Vol. 6848, Springer, 2011, pp. 76–136.

[29] A. Borgida and L. Serafini, Distributed Description Logics: Assimilating Information from Peer Sources, *Journal on Data Semantics* **1** (2003), 153–184.

[30] A. Zimmermann and J. Euzenat, Three Semantics for Distributed Systems and Their Relations with Alignment Composition, in: *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, Lecture Notes in Computer Science, Vol. 4273, Springer, 2006, pp. 16–29.

[31] M. Atencia, A. Borgida, J. Euzenat, C. Ghidini and L. Serafini, A Formal Semantics for Weighted Ontology Mappings, in: *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 7649, Springer, 2012, pp. 17–33.