Journal Title 0 (0) 1
IOS Press

On the relation between keys and link keys for data interlinking

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, may be used to perform data interlinking, i.e. finding identical resources in different RDF datasets. However, the precise relationship between keys and link keys has not been fully determined yet. A common formal framework encompassing both keys and link keys is necessary to ensure the correctness of data interlinking tools based on them, and for determining their scope and possible overlapping. In this paper, we provide a semantics for keys and link keys within description logics. We determine under which conditions they are legitimate to generate links. We provide conditions under which link keys are logically equivalent to keys. In particular, we show that data interlinking with keys and ontology 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

The linked data initiative has made possible the development of a continuously growing web of data accessible to machines. Data is published using RDF, which enables describing web resources identified by Internationalized Resource Identifiers (IRIs) in terms of property values [1].

Interoperability in the web of data largely relies on links between data from different RDF datasets and especially links asserting the identity of resources bearing different IRIs, specified using the owl:sameAs property. Since RDF datasets tend to be large, automatically discovering owl:sameAs links between RDF datasets is an important and challenging task. This task is referred to as data interlinking and different algorithms and tools for data interlinking have been proposed so far [2, 3].

Among the state-of-the-art approaches to data interlinking, some are based on finding keys [4–7] or link keys [8, 9] across RDF datasets. Both keys and link keys are devices characterising what makes two resources to be identical. Hence, it is natural to exploit them for discovering links across datasets. Even though both techniques have been proven to be effective in data interlinking scenarios, their relationship has not been formally established yet.

The objective of this paper is to clarify the relationship between keys and link keys. In order to do so, we first provide the semantics of (RDF) keys and link keys. More specifically, we formalise how a key, in its different versions, can be combined with an alignment between ontologies for data interlinking. Then, we extend the current definition of a link key by defining the semantics of six kinds of link keys — weak, plain and strong link keys, and their in- and eq-variants — and we logically ground the usage of link keys for data interlinking. Finally, we establish the conditions under which link keys are equivalent to keys

*Corresponding author.

0000-0000/0-1900/$00.00 © 0 – IOS Press and the authors. All rights reserved

and show 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 the context and related work of the paper. Section 3 introduces the notations used throughout the paper. Section 4 recalls two different semantics of keys and Section 5 logically justifies their use for data interlinking. Section 6 defines link keys. Section 7 logically grounds the use of link keys for data interlinking. The relations between keys and link keys are established in Section 8, both with respect to their logical entailment and the links they produce. Section 9 concludes the paper and discusses future work.

All definitions are illustrated with concrete examples taken from real-world datasets.

2. Context and Related Work

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 the owl:sameAs property. To decide whether two IRIs represent the same entity or not is mainly based on comparing their values for a selected number of properties. Data interlinking is reminiscent of the task of record linkage in databases [10] but it is applied to RDF data described with RDFS/OWL ontologies.

Link discovery frameworks such as SILK [11, 12] and LIMES [13] enable users to process link specifications to generate links. Link specifications 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].

A key is a set of (datatype or object) properties that uniquely identify the instances of a class within a dataset. For example,

$$\{\mathsf{creator}, \mathsf{title}\} \text{ key } \mathsf{Book}$$

states that, if two instances of the class Book coincide on values for the properties hasCreator and hasTitle, then the two instances are the same. Key-based approaches to data interlinking first extract key candidates from RDF datasets and then select the most accurate candidates according to different quality measures [4–7]. When the data of two RDF datasets are described using the same ontologies, then keys, if available, can be directly used for interlinking the datasets, but if the data are described using different ontologies then they need to be combined with ontology alignments [16] relating the properties and classes of the data. For example, the previous key could be combined with the alignment correspondences creator $\equiv$ auteur, title $\equiv$ titre and Book $\equiv$ Livre to interlink the books of English and French libraries.

Keys can be used to build link specifications or can be translated into logical rules to perform data interlinking. The latter allows to take advantage of logical reasoning [17–19]. Key extraction algorithms discover either S-keys [5–7] or F-keys [4, 20]. There are two kinds of keys since RDF properties are multivalued contrary to relational attributes which are functional. 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 the AssistantProfessor class). But if the properties form an F-key then the instances must share *all* values (e.g. hasPoem property for the PoemAnthology class because 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.

When datasets are described with different ontologies, alignments must be used, either during the key extraction process or later when performing data interlinking. For example, the approach proposed in [5] searches in a source dataset for S-keys over classes which are equivalent to classes in a target dataset and then selects among the discovered S-keys those composed of properties which are equivalent to properties of the target dataset.

Link keys generalise the combination of keys and ontology alignments for data interlinking [8, 16]. A link key is a set of pairs of properties that uniquely identify the instances of two classes of two RDF datasets. For example,

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

states that, if an instance of Book has the same values for auteur and titre as an instance of Book has for creator and title, the two instances are the same. Unlike the previous key, this link key could be used directly to interlink the books of English and French libraries, without the need to involve any ontology alignment.

The key-based approaches to data interlinking proposed in [6, 7] are different from [5] and closer to link keys [8]. Indeed, their goal is to discover S-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.

The formal semantics of S-keys and F-keys have been given in [21] using rules, but the combination of S-keys and F-keys with ontology alignments for data interlinking is not formally addressed. In this paper, we address it using description logics.

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 main goal of these approaches is to study decidability of reasoning with keys or functional dependencies in description logics. Instead, we use description logics to provide the semantics of keys and link keys in order to fully understand the relation between them in the context of data interlinking.

In this paper, we will use description logics to express the semantics of various types of keys and link keys. This will allow to ground their legitimacy in generating links across RDF datasets. This will also be used to compare the respective key and link key expressions on the basis of their entailments and the links they generate.

3. Preliminaries

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

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

Definition 1 (Ontology)**.** *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 $\mathcal{O}$.*

A schema is, thus, 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 $p_1 \mathcal{R} q_2$ with $\mathcal{R} \in \{\sqsubseteq, \sqsupseteq, \equiv, \bot\}$. A dataset is a set of assertional axioms between individuals: $C(a)$ and $p(a_1, a_2)$. Classes, properties and individuals $(C_1, p_1, a_1, \ldots)$ define the vocabulary of an ontology. 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 [16]. Alignment relations between classes and properties are usually subsumption, equivalence and disjointness. In the case of individuals, they are typically related by the owl:sameAs property, which expresses equality of individuals. Alignment statements between classes and properties are referred to as correspondences, whereas equality statements between individuals will be 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 (Alignment). *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 $p \mathcal{R} q$ with $\mathcal{R} \in \{\sqsubseteq, \sqsupseteq, \equiv, \bot\}$, 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 $\mathcal{L}_{\mathcal{O},\mathcal{O}'}$ as* 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. Without loss of generality, we can assume that the vocabularies of $\mathcal{O}$ and $\mathcal{O}'$ are disjoint.

In what follows, given an ontology $\mathcal{O}$, 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 $\mathcal{O}'$ ($\mathcal{R} \in \{\sqsubseteq, \sqsupseteq, \equiv, \bot\}$).

4. Two Kinds of Keys in Description Logics

In order to compare keys and link keys, we start by reformulating the semantics of keys [21] as description logic axioms. We distinguish between several types of keys which apply in this context. 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 are related to the conditions (1) and (2) in Definitions 3 and 4 given below.

4.1. Semantics of keys

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 (in-key)**.** *An* in-key assertion*, or simply an in-key, has the form*

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

such that $p_1, \ldots, p_k$ 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 $\delta, \delta' \in C^{\mathcal{I}}$,

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

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

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

such that $p_1, \ldots, p_k$ 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 $\delta, \delta' \in C^{\mathcal{I}}$,

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

According to Definition 3, if two instances of a class share at least *one* value for each of the properties of an in-key for the class, then we can infer that they are the same instance. This is formalised in Proposition 1.

Proposition 1. *The following holds:*

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

Proof. This is a direct consequence of Definition 3: for any interpretation $\mathcal{I}$ satisfying all the antecedent axioms of the entailment, $a^{\mathcal{I}}$ and $b^{\mathcal{I}}$ will belong to $C^{\mathcal{I}}$ and will share one value for each of the properties of the in-key, hence $a^{\mathcal{I}}$ will be equal to $b^{\mathcal{I}}$. □

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 (and at least one) for each of the properties of the key. However, we need to be sure that *all known values* indeed are *all values* that the instances may have. This is proved in Proposition 2.

Proposition 2. *The following holds:*

$$\begin{aligned}
C(a), \{p_i(a, c_i^1), \ldots, p_i(a, c_i^{r_i})\}_{i=1}^{k} \\
\{\{a\} \sqsubseteq \forall p_i.\{c_i^1, \ldots, c_i^{r_i}\}\}_{i=1}^{k} \\
C(b), \{p_i(b, d_i^1), \ldots, p_i(b, d_i^{r_i})\}_{i=1}^{k}
\end{aligned}$$

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

Proof. Let $\mathcal{I}$ be an interpretation that satisfies all the antecedent axioms of the above entailment. Let us prove that $\mathcal{I}$ satisfies $a \approx b$ too. Since $\mathcal{I} \models p_i(a, c_i^l)$ then $(c_i^l)^{\mathcal{I}} \in p_i^{\mathcal{I}}(a^{\mathcal{I}})$ for $i = 1, \ldots, k$ and $l = 1, \ldots, r_i$. Also, since $\mathcal{I} \models \{a\} \sqsubseteq \forall p_i.\{c_i^1, \ldots, c_i^{r_i}\}$ then $p_i^{\mathcal{I}}(a^{\mathcal{I}}) \subseteq \{(c_i^l)^{\mathcal{I}}\}_{l=1}^{r_i}$. Therefore, $p_i^{\mathcal{I}}(a^{\mathcal{I}}) = \{(c_i^l)^{\mathcal{I}}\}_{l=1}^{r_i}$. Similarly, $q_i^{\mathcal{I}}(b^{\mathcal{I}}) = \{(d_i^l)^{\mathcal{I}}\}_{l=1}^{r_i}$. Now, since $\mathcal{I} \models c_i^l \approx d_i^l$ then $(c_i^l)^{\mathcal{I}} = (d_i^l)^{\mathcal{I}}$, which implies that $p_i^{\mathcal{I}}(a^{\mathcal{I}}) = q_i^{\mathcal{I}}(b^{\mathcal{I}})$. Furthermore, $p_i^{\mathcal{I}}(a^{\mathcal{I}}) = q_i^{\mathcal{I}}(b^{\mathcal{I}}) \neq \emptyset$ since $r_i \geqslant 1$. Additionally, since $\mathcal{I} \models C(a)$ and $\mathcal{I} \models C(b)$ then $a^{\mathcal{I}}, b^{\mathcal{I}} \in C^{\mathcal{I}}$. Finally, since $\mathcal{I} \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{eq}}\ C)$, and we have $a^{\mathcal{I}}, b^{\mathcal{I}} \in C^{\mathcal{I}}$ and $p_i^{\mathcal{I}}(a^{\mathcal{I}}) = q_i^{\mathcal{I}}(b^{\mathcal{I}}) \neq \emptyset$ for $i = 1 \ldots, k$, then we can infer that $a^{\mathcal{I}} = b^{\mathcal{I}}$, i.e. $\mathcal{I} \models a \approx b$. □

Thus, 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 and linked open data, it is also expected to be made in certain controlled scenarios. The semantics of owl:hasKey in OWL2 corresponds to the semantics of in-keys 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 (Generalised key)**.** *A* key assertion*, or simply a key, has the form*

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

such that $p_1, \ldots, p_k$ 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).

Example 1 below provides examples of in-keys and eq-keys in real RDF datasets.

Example 1. Insee is the French institution in 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.[1] Insee ontologies are available too. In this example and Example 2, we only consider the Insee data related to administrative districts (the 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

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

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.

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, the codeINSEE property is an in-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.[2]

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 three different communes: Bully in the department of Loire, Bully in Rhône and Bully in Seine-Maritime. However, there is 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 unified keys):

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

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

4.2. Relations between the different types of keys

Compared to the semantics of S-keys and F-keys defined in [21], the semantics of in-keys corresponds directly to the semantics of S-keys. This is not the case for eq-keys and F-keys. Every eq-key is an F-key

[2]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.

but not the other way around. The equivalence would be established if condition (2) in Definition 4 were replaced by

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

The prerequisite that the sets of property values must be non-empty enables to consider in-keys as a subset of eq-keys (which does not hold between S-keys and F-keys). This result is stated in Proposition 3.

Proposition 3. $(\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C) \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{eq}}\ C)$

Proof. Let $\mathcal{I}$ be an interpretation such that $\mathcal{I} \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C)$. We have to prove that $\mathcal{I} \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{eq}}\ C)$. Let $\delta, \delta' \in C^{\mathcal{I}}$ such that $p_i^{\mathcal{I}}(\delta) = p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. We have to prove that $\delta = \delta'$. Since $p_i^{\mathcal{I}}(\delta)$ and $p_i^{\mathcal{I}}(\delta')$ are equal and non-empty, then $p_i^{\mathcal{I}}(\delta) \cap p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. So we have $\delta, \delta' \in C^{\mathcal{I}}$ and $p_i^{\mathcal{I}}(\delta) \cap p_i^{\mathcal{I}}(\delta') \neq \emptyset$ $(i = 1, \ldots, k)$. Since $\mathcal{I} \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C)$, then $\delta = \delta'$. □

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 4. *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 $\delta, \delta' \in C^{\mathcal{I}}$ 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 $|p_i^{\mathcal{I}}(\delta')| \leqslant 1$, but since their intersection is not empty then $|p_i^{\mathcal{I}}(\delta)| = 1$ and $|p_i^{\mathcal{I}}(\delta')| = 1$, thus 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 $\mathcal{I} \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C)$. □

Proposition 5 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.

Proposition 5. *The following holds:*

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

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

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

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

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

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

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

Proof. Properties (5) and (6) follow directly from Definitions 3 and 4, and 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 establish when it is legitimate to combine in-keys and eq-keys with alignments for data interlinking.

5. Data Interlinking with Keys and Alignments

So far, we have considered keys independently from their use for data interlinking. Keys are able to identify duplicate resources within the same dataset and links between resources from different datasets described using the same ontologies. But as soon as the datasets do not share the schemas, keys alone are not enough for performing data interlinking, and alignments are required.

In this section, we uncover the implicit or explicit role of ontology alignments in the process of data interlinking with keys. We show that data interlinking can be expressed as a direct logical consequence of the semantics of keys and alignments. We also highlight the need for completion when interlinking data with eq-keys.

Data interlinking can be formulated 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 (possibly discovered with the help of key extraction tools) and an alignment $\mathcal{A} = \langle \mathcal{C}, \mathcal{L} \rangle$ 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:

$$\mathcal{O}, \mathcal{O}', \mathcal{A} \models a \approx b \tag{11}$$

Of course, there is a specific case in which an alignment is not needed: this is when the two datasets use the same schema, i.e. when $\mathcal{S} = \mathcal{S}'$. Such a case can be reduced to data deduplication in the ontology $\langle \mathcal{S}, \mathcal{D} \cup \mathcal{D}', \mathcal{K} \cup \mathcal{K}' \rangle$. It can be seen as a particular instance of (11) when $\mathcal{A}$ is the identity alignment.

In the following, 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 $\kappa \in \mathcal{K}$, are sufficient for inferring a (new) link $a \approx b$. These conditions change depending on whether κ is an in-key or an eq-key, as specified in Theorem 1 and Theorem 2 below. These two theorems provide the logical grounds 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* $\mathcal{A} = \langle \mathcal{C}, \mathcal{L} \rangle$ *and alignment between* $\mathcal{O}$ *and* $\mathcal{O}'$ *such that*

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

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

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

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

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 Proposition 1. □

Theorem 1 provides the logical ground of data interlinking with in-keys and alignments: if we know that the properties $p_1,\ldots,p_k$ constitute an in-key for a class C in $\mathcal{O}$, and that, according to an alignment $\mathcal{A}$, 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 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 $\mathcal{S}'$.

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*

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

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

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

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

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 Proposition 2. □

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 equality and not on an initial set of known same-as links $\mathcal{L}$).

Another interesting remark on Theorems 1 and 2 is that only one key of $\mathcal{O}$ and no key from $\mathcal{O}'$ is needed to infer links. Actually, under the assumptions of the theorem, by Proposition 5, $\{q\}_{i=1}^{k}$ is guaranteed to be an in-key (in Theorem 1) or an eq-key (in Theorem 2) for the class D.

Even though Theorems 1 and 2 are not difficult to prove, they highlight some peculiarities of data interlinking with keys and alignments that have not received attention in the literature: the fact that equivalence of properties is not required for interlinking with in-keys, and that local completeness is necessary for interlinking with eq-keys.

Finally, it is possible to provide semantic versions of Theorems 1 and 2 in which the antecedent axioms are not asserted in the ontologies and alignments but are inferred from them (e.g. $\mathcal{O}, \mathcal{O}', \mathcal{A} \models (\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C)$ instead of $(\{p_1, \ldots, p_k\} \text{ key}_{\text{in}}\ C) \in \mathcal{K}$). We have decided to present the asserted versions to stress the nature of each axiom (mapping, data, schema knowledge or links).

We finish this section with the definition of the link set generated by a key.

Definition 6 (Link set generated by a key)**.** *Let $\mathcal{O}$ and $\mathcal{O}'$ be two ontologies. Let $\mathcal{A}$ be an alignment between $\mathcal{O}$ and $\mathcal{O}'$. Let κ be a key. The set of links between $\mathcal{O}$ and $\mathcal{O}'$ generated by κ under $\mathcal{A}$ is defined as*

$$\mathcal{L}_{\kappa}^{\mathcal{O},\mathcal{O}',\mathcal{A}} = \{a \approx b : \mathcal{O}, \mathcal{O}', \mathcal{A}, \kappa \models a \approx b \text{ and } \mathcal{O}, \mathcal{O}', \mathcal{A} \not\models a \approx b\}$$

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

6. Link Keys

To be compared with keys, link keys require a precise semantics. Here we formalise and extend the semantics introduced in [9] to other link keys that are useful in practice. This semantics generalises the semantics of keys to the case of different RDF datasets.

6.1. Semantics of link keys

The semantics of link keys considered in [9] is reproduced in Definition 7. It is natural to extend this semantics to eq-keys too, and we do so in Definition 8. These kinds of link keys will be referred to as *weak link keys*.

Definition 7 (Weak in-link key)**.** *A* weak in-link key assertion*, or simply a weak in-link key, has the form*

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

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

An interpretation $\mathcal{I}$ satisfies $(\{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\} \text{ linkkey}_{\text{in}}^{\text{w}}\ \langle C, D\rangle)$ 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 8 (Weak eq-link key)**.** *A* weak eq-link key assertion*, or simply a weak eq-link key, has the form*

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

such that $p_1, \ldots, p_k$ 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 $\eta \in D^{\mathcal{I}}$,

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

Interestingly, every key $(\{p_1, \ldots, p_k\} \text{ key}_x\ C)$ can be expressed as an equivalent weak link key $(\{\langle p_1, p_1\rangle, \ldots, \langle p_k, p_k\rangle\} \text{ linkkey}^{\text{w}}_x \langle C, C\rangle)$, with $x \in \{\text{in}, \text{eq}\}$.

Weak link keys are called *weak* because they are not necessarily composed of keys. We introduce *strong link keys*, which embed two keys. Because of this, they will facilitate the comparison with keys in Section 8. We only give the definition of strong in-link keys, as strong eq-link keys can be defined analogously.

Definition 9 (Strong in-link key)**.** *A* strong in-link key assertion*, or simply a strong in-link key, has the form*

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

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

An interpretation $\mathcal{I}$ satisfies $(\{\langle p_1, q_1\rangle, \ldots, \langle p_k, q_k\rangle\} \text{ linkkey}^{\text{s}}_{\text{in}} \langle C, D\rangle)$ iff

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

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 $\{p_i\}_{i=1}^k$ and $\{q_i\}_{i=1}^k$ 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 7 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 enable finding links between two different datasets, but strong link keys do more. Indeed, since the properties of a strong link key are keys for the classes separately then 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 intermediate between weak and 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 be keys 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 are defined similarly. Figure 1 illustrates the differences between weak, plain and strong in-link keys.

Definition 10 (Plain in-link key)**.** *A* plain in-link key assertion*, or simply a plain in-link key, has the form*

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

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

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

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

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

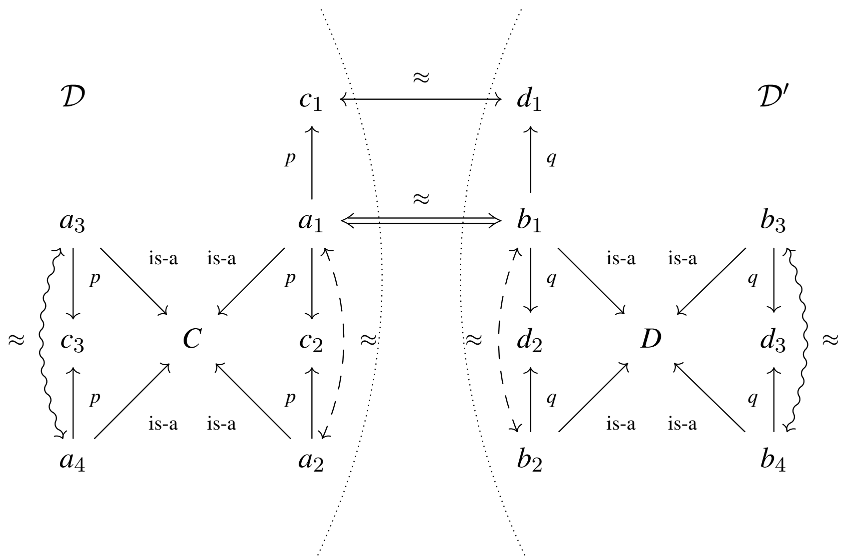

Fig. 1. Two datasets and links generated depending on the type of link keys (double=weak, dashed=plain, waved=strong).

Figure 1 illustrates the differences between the types of link keys on two data sets $\mathcal{D}$ and $\mathcal{D}'$:

$$\mathcal{D} = \{p(a_1, c_1), p(a_1, c_2), C(a_1), p(a_2, c_2), C(a_2), C(a_3), p(a_3, c_3), C(a_4), p(a_4, c_3)\}$$
$$\mathcal{D}' = \{q(b_1, d_1), q(b_1, d_2), D(b_1), q(b_2, d_2), D(b_2), D(b_3), q(b_3, d_3), D(b_4), q(b_4, d_3)\}$$

with the initial set of links:

$$\mathcal{L} = \{c_1 \approx d_1\}$$

Considering an in-link key: $(\{\langle p, q\rangle\} \text{ linkkey}^{y}_{\text{in}} \langle C, D\rangle)$, depending on the value of y, it will generate:

weak: $a_1 \approx b_1$ (double-line arrow),
plain: additionally $a_1 \approx a_2$ and $b_1 \approx b_2$ (dashed arrows),
strong: additionally $a_3 \approx a_4$ and $b_3 \approx b_4$ (wave arrows).

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

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

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

Alignments may be naturally extended to include a set of link keys. From here on, given two ontologies $\mathcal{O}$ and $\mathcal{O}'$ equipped with keys, an alignment $\mathcal{A}$ between $\mathcal{O}$ and $\mathcal{O}'$ will be a triple $\mathcal{A} = \langle \mathcal{C}, \mathcal{L}, \mathcal{LK}\rangle$ which, in addition to 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).[3] 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.[4]

We have extracted the different link key conditions for the property pair ⟨ins:codeINSEE, ign:numInsee⟩ on the union of Insee and IGN datasets taking into account the existing owl:sameAs links. There happen to be strong in-link keys for the class pairs ⟨ins:Com, ign:Com⟩, ⟨ins:Arr, ign:Arr⟩, ⟨ins:Dép, ign:Dép⟩ and ⟨ins:Rég, ign:Rég⟩. In symbols:

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

where $\mathcal{I}^*$ is a canonical interpretation of the RDF graph resulting from the union of the Insee and IGN datasets whose 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 extracted 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⟩. This has been performed 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

[3]http://data.ign.fr
[4]The ign prefix denotes the namespace http://data.ign.fr/def/geofla#.

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 $\langle \text{ins:nom}, \text{rdfs:label} \rangle$ is a strong in-link key for $\langle \text{ins:Dép}, \text{ign:Dép} \rangle$ and $\langle \text{ins:Rég}, \text{ign:Rég} \rangle$. The property pairs $\langle \text{ins:subdivisionDe}, \text{ign:arr} \rangle$ and $\langle \text{ins:subdivisionDe}, \text{ign:dpt} \rangle$ together with $\langle \text{ins:nom}, \text{rdfs:label} \rangle$ constitute weak (and plain) in-link keys for the class pairs $\langle \text{ins:Com}, \text{ign:Com} \rangle$ and $\langle \text{ins:Arr}, \text{ign:Arr} \rangle$, 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 (as well as ign:dpt) refers to a single administrative unit, though subdivisionDe refers to several administrative units due to transitivity. In symbols,

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

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

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

6.2. Relations between different link keys

In what follows, we provide theoretical results stating the relations between the different kinds of link keys. Propositions 6 and 7 are the counterparts of Propositions 3 and 4 for link keys and can be proven similarly.

Proposition 6. *The following holds:*

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

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

Proposition 7. *If* $p_1, \ldots, p_k$ *and* $q_1, \ldots, q_k$ *are functional then*

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

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

Proposition 8 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 the transitivity of equality.

Proposition 8. *The following holds:*

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

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

Proof. The first two propositions follow direclty from the definitions of link keys. We prove the validity of the third one Let $\mathcal{I}$ be a DL interpretation such that $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}^{\text{w}}_{\text{eq}} \langle C, D\rangle)$, and let us prove that $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}^{\text{p}}_{\text{eq}} \langle C, D\rangle)$. Let $\delta \in C^{\mathcal{I}}$ and $\eta \in D^{\mathcal{I}}$ be such that $p_i^{\mathcal{I}}(\delta) = q_i^{\mathcal{I}}(\eta) \neq \emptyset$ $(i = 1, \ldots, k)$. Since $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}^{\text{w}}_{\text{eq}} \langle C, D\rangle)$, 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 $\mathcal{I} \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}^{\text{w}}_{\text{eq}} \langle C, D\rangle)$ implies $\delta' = \eta$, and, since $\delta = \eta$, then $\delta = \delta'$. The last condition of plain eq-link keys can be proven analogously. $\square$

In the following section, we establish when it is legitimate to use link keys for data interlinking.

7. Data interlinking with link keys

Theorems 3 and 4 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 of the proofs of Theorems 1 and 2 and are omitted.

Theorem 3. *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. Let* $\mathcal{A} = \langle \mathcal{C}, \mathcal{L}, \mathcal{LK}\rangle$ *be an alignment between* $\mathcal{O}$ *and* $\mathcal{O}'$ *such that*

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

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

The counterpart of Theorem 3 for weak eq-link keys is Theorem 4:

Theorem 4. *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. Let* $\mathcal{A} = \langle \mathcal{C}, \mathcal{L}, \mathcal{LK}\rangle$ *be an alignment between* $\mathcal{O}$ *and* $\mathcal{O}'$ *such that*

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

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

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

- $\{\{b\} \sqsubseteq \forall q_i.\{d_i^1, \ldots, d_i^{r_i}\}\}_{i=1}^k \subseteq \mathcal{S}'$ *and*
- $\bigcup_{i=1}^{k} \{c_i^j \approx d_i^j\}_{j=1}^{r_i} \subseteq \mathcal{L}$

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

Theorems 3 and 4 prove that weak link keys are enough and do not need mappings between classes and properties to perform data interlinking.

Notice that, by Proposition 8, any plain or strong link key is a weak link key, so Theorems 3 and 4 also hold for them. Plain and strong link keys can be used in the same way to infer equality statements between individuals of the same dataset.

We finish this section with the definition of the link set generated by a link key.

Definition 11 (Link set generated by a link key)**.** *Let $\mathcal{O}$ and $\mathcal{O}'$ be two ontologies. Let $\mathcal{A}$ be an alignment between $\mathcal{O}$ and $\mathcal{O}'$. Let λ be a link key. The set of links between $\mathcal{O}$ and $\mathcal{O}'$ generated by λ under $\mathcal{A}$ is defined as*

$$\mathcal{L}_{\lambda}^{\mathcal{O},\mathcal{O}',\mathcal{A}} = \{a \approx b : \mathcal{O}, \mathcal{O}', \mathcal{A}, \lambda \models a \approx b \text{ and } \mathcal{O}, \mathcal{O}', \mathcal{A} \not\models a \approx b\}$$

In the following section, we compare data interlinking with link keys with data interlinking with keys and alignments as described in Section 5.

8. Relation between Keys and Link Keys

Keys and link keys are data interlinking devices that we have developed so far in a parallel manner. One then may expect that their application always results in the generation of the same links. We are now able to formally establish the relation between keys and link keys, and to show that, although there may be data interlinking scenarios in which they will return the same links, this will not always be the case.

This section starts by studying the relation between keys and link keys as description logic axioms (§8.1). Theorem 5 states the correspondence between strong link keys and keys and alignments. This correspondence no longer holds for weak link keys (Theorem 6). We also study the impact of these results on the generation of links (§8.2): Theorems 8 and 9 show that the links generated by a strong link key are the same as the links generated by its corresponding keys and proper alignments. There are cases, though, in which it is possible to generate links with weak link keys while it is not possible with keys and alignments.

8.1. Logical relations between keys and link keys

The theorems presented here are consequences of stronger results included in Appendix A. We have decided to not include the latter in this section because the former are more directly related to data interlinking with keys and link keys.

Theorem 5 states the correspondence between strong link keys and keys and alignments: (12) says that strong link keys entail keys; (13) and (14) express conditions at which the converse of (12) holds.

Theorem 5. *The following holds:*

$$(\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_x^{\text{s}} \langle C, D\rangle) \models (\{p_i\}_{i=1}^k \text{ key}_x\ C) \tag{12}$$

$$(\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C), C \sqsupseteq D, \{p_i \sqsupseteq q_i\}_{i=1}^k \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{s}} \langle C, D\rangle) \tag{13}$$

$$(\{p_i\}_{i=1}^k \text{ key}_{\text{eq}}\ C), C \sqsupseteq D, \{p_i \equiv q_i\}_{i=1}^k \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{eq}}^{\text{s}} \langle C, D\rangle) \tag{14}$$

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

Proof. (12) is a direct consequence of the definition of strong link keys (Definition 9). (13) and (14) are consequences of Proposition 11 in Appendix A. □

Given the symmetry of the link key definitions, (12), (13) and (14) hold for the right-hand side of the link key too (with reversed subsumption relations).

Theorem 5 states that it is possible to infer keys from strong link keys. This is not surprising because strong link keys are composed of keys by definition. We call these keys the side keys associated with a strong link key. More interestingly, Theorem 5 also states that strong link keys can be inferred from keys and proper alignments. Note that one key is enough to entail the strong link key as long as the alignment holds (these alignments are different depending on whether in-link keys or eq-link keys are considered).

The converses of (13) and (14) are only partly true: strong link keys entail keys, but strong link keys (nor plain or weak link keys) do not necessarily entail an alignment between their properties and classes. This refutes the idea that link keys embed alignments. Link keys do not assert alignments, but express conditions for identifying individuals. A link key between two classes C and D does not assert that C and D are equivalent, nor that one of the classes subsumes the other, it just specifies how to link individuals that are described as instances of C and D, but there may be individuals in both classes that do not belong to the other class. For example, there may exist a link key between the classes AdministrativeCentre and Town, although no equivalence, nor subsumption holds between them (some administrative centres are towns, others are cities; some towns are administrative centres, others not).

Is Theorem 5 still valid for weak and plain link keys? (13) and (14) hold, but (12) does not. In other words: keys and proper alignments entail weak and plain link keys (Corollary 5.1); however, the side components of neither weak nor plain link keys are necessarily keys (Theorem 6).

Corollary 5.1. *The following holds:*

$$(\{p_i\}_{i=1}^k \text{ key}_{\text{in}}\ C), C \sqsupseteq D, \{p_i \sqsupseteq q_i\}_{i=1}^k \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{y} \langle C, D\rangle) \tag{15}$$

$$(\{p_i\}_{i=1}^k \text{ key}_{\text{eq}}\ C), C \sqsupseteq D, \{p_i \equiv q_i\}_{i=1}^k \models (\{\langle p_i, q_i\rangle\}_{i=1}^k \text{ linkkey}_{\text{eq}}^{y} \langle C, D\rangle) \tag{16}$$

with $y \in \{\text{w}, \text{p}\}$.

Proof. This is a direct consequence of Theorem 5 since, by Proposition 8, any strong link key is also a plain and a weak link key. □

Unlike strong link keys, the side components of neither weak nor plain link keys are necessarily keys. The proof of Theorem 6 provides two ontologies that are consistent with a weak link key but are inconsistent with any of its side components.

Theorem 6. *There exist ontologies that are consistent with a weak link key but inconsistent with each of its side components.*

Proof. Consider the following ontologies:

$\mathcal{O}:$	$\mathcal{O}':$
$C(u_1), C(u_2), C(u_3), C(u_4),$	$C'(u'_1), C'(u'_2), C'(u'_3), C'(u'_4),$
$p(u_1, v_1), p(u_2, v_2), p(u_3, v_1), p(u_4, v_2),$	$p'(u'_1, v_1), p'(u'_2, v'_2), p'(u'_3, v_1), p'(u'_3, v_1),$
$q(u_1, w_1), q(u_2, w_1), q(u_3, w_2), q(u_4, w_1),$	$q'(u'_1, w_1), q'(u'_2, w_1), q'(u'_3, w'_2), q'(u'_4, w'_2),$
$u_1 \not\approx u_2 \not\approx u_3 \not\approx u_4$	$u'_1 \not\approx u'_2 \not\approx u'_3 \not\approx u'_4$

It can be checked that

$$\lambda = (\{\langle p, p'\rangle, \langle q, q'\rangle\} \text{ linkkey}^{\text{w}}_{\text{in}} \langle C, C'\rangle)$$

is consistent with $\mathcal{O} \cup \mathcal{O}'$. Notice that λ together with $\mathcal{O}$ and $\mathcal{O}'$ entails the link $u_1 \approx u'_1$.

However, the side components of λ, i.e.

$$\kappa = (\{p, q\} \text{ key}_{\text{in}}\ C) \qquad \kappa' = (\{p', q'\} \text{ key}_{\text{in}}\ C')$$

are inconsistent with $\mathcal{O}$ and $\mathcal{O}'$, respectively. Indeed, $\mathcal{O} \cup \{\kappa\} \models u_2 \approx u_4$ because u_2 and u_4 share one value for p (namely, v_2) and q (w_1). Also, $\mathcal{O} \cup \{\kappa\} \models u_2 \not\approx u_4$ because $u_2 \not\approx u_4$ belongs to $\mathcal{O}$. This means that $\mathcal{O} \cup \{\kappa\}$ is inconsistent. In the same way, it can shown that κ' is inconsistent with $\mathcal{O}'$. □

It is noteworthy that not a single useful key (i.e. a key that can be used to generate links) can be found in the ontologies of the proof of Theorem 6: $(\{p\} \text{ key}_{\text{in}}\ C)$ and $(\{q\} \text{ key}_{\text{in}}\ C)$ are both inconsistent with $\mathcal{O}$, and $(\{p'\} \text{ key}_{\text{in}}\ C)$ and $(\{q'\} \text{ key}_{\text{in}}\ C)$ with $\mathcal{O}'$. As a consequence, in this example, data interlinking is possible with link keys (λ allows to find $u_1 \approx u'_1$) but not with keys.

Example 3 makes clear in the context of a realistic data interlinking scenario that the converse of (12) of Theorem 5 does not hold for weak link keys.

Example 3. The following statement of Example 2:

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

expresses a weak in-link key satisfied by $\mathcal{I}^*$, the canonical interpretation of the RDF graphs of Example 2 whose linked individuals are merged.

Let us consider the side components of the above weak link key:

$$(\{\text{ins:nom}, \text{ins:subdivisionDe}\} \text{ key}_{\text{in}}\ \text{ins:Com}) \qquad (\{\text{rdfs:label}, \text{ign:arr}\} \text{ key}_{\text{in}}\ \text{ign:Com})$$

However, as explained in Example 2, $(\{\text{ins:nom}, \text{ins:subdivisionDe}\} \text{ key}_{\text{in}}\ \text{ins:Com})$ is not satisfied by $\mathcal{I}^*$ due to the transitivity of the property ins:subdivisionDe.

One may think that data interlinking is still possible with $(\{\text{rdfs:label}, \text{ign:arr}\} \text{ key}_{\text{in}}\ \text{ign:Com})$, which is indeed satisfied by $\mathcal{I}^*$. This would require the following alignment correspondences to hold

$$\text{ins:nom} \sqsupseteq \text{rdfs:label} \qquad \text{ins:subdivisionDe} \sqsupseteq \text{ign:arr}$$

However, $\mathcal{I}^*$ does not satisfy ins:nom $\sqsupseteq$ rdfs:label but the reversed subsumption ins:nom $\sqsubseteq$ rdfs:label.

Even though the side components of a weak link key are not necessarily keys for the ontologies separately, every weak link key entails one key in the vocabulary of the ontologies together, as stated by Theorem 7 below. Unfortunately, this link key is of very limited use in practice because the inferred key holds for the intersection of the classes that we actually want to interlink (it is not known in advance which individuals belong to both classes).

Theorem 7. *The following holds:*

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

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

Proof. This is a consequence of Proposition 9 in Appendix A. $\square$

8.2. Relations between generated link sets

The difference between using link keys for data interlinking instead of keys and ontology alignments becomes evident when comparing Theorem 1 with Theorem 3 and Theorem 2 with Theorem 4. In both cases, knowledge about keys and alignments is replaced by knowledge about link keys. Theorem 8 shows that the generated link sets are exactly the same.

Theorem 8. *Let* $\mathcal{O}$ *and* $\mathcal{O}'$ *be two ontologies. Let* $\mathcal{A} = \langle \mathcal{C}, \mathcal{L}, \mathcal{LK} \rangle$ *be an alignment between* $\mathcal{O}$ *and* $\mathcal{O}'$ *such that* $\{C \sqsupseteq D\} \cup \{p_i \sqsubseteq q_i\}_{i=1}^k \subseteq \mathcal{C}$. *Let* $\kappa = (\{p_i\}_{i=1}^k \text{ key}_{\text{in}} C)$ *and* $\lambda = (\{\langle p_i, q_i \rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{s}} \langle C, D \rangle)$. *Then* $\mathcal{L}_\kappa^{\mathcal{O},\mathcal{O}',\mathcal{A}} = \mathcal{L}_\lambda^{\mathcal{O},\mathcal{O}',\mathcal{A}}$.

Proof. The result follows from Definitions 6 and 11 and the fact that, since $\{C \sqsupseteq D\} \cup \{p_i \sqsubseteq q_i\}_{i=1}^k \subseteq \mathcal{C}$ then, by clause (13) of Theorem 5, we have $\mathcal{O}, \mathcal{O}', \mathcal{A}, \kappa \models \lambda$, and also $\mathcal{O}, \mathcal{O}', \mathcal{A}, \lambda \models \kappa$. $\square$

The same holds for eq-keys and eq-link keys.

Theorem 9. *Let* $\mathcal{O}$ *and* $\mathcal{O}'$ *be two ontologies. Let* $\mathcal{A} = \langle \mathcal{C}, \mathcal{L}, \mathcal{LK} \rangle$ *be an alignment between* $\mathcal{O}$ *and* $\mathcal{O}'$ *such that* $\{C \sqsupseteq D\} \cup \{p_i \equiv q_i\}_{i=1}^k \subseteq \mathcal{C}$. *Let* $\kappa = (\{p_i\}_{i=1}^k \text{ key}_{\text{in}} C)$ *and* $\lambda = (\{\langle p_i, q_i \rangle\}_{i=1}^k \text{ linkkey}_{\text{in}}^{\text{s}} \langle C, D \rangle)$. *Then* $\mathcal{L}_\kappa^{\mathcal{O},\mathcal{O}',\mathcal{A}} = \mathcal{L}_\lambda^{\mathcal{O},\mathcal{O}',\mathcal{A}}$.

Proof. The result follows from Definitions 6 and 11 and the fact that, since $\{C \sqsupseteq D\} \cup \{p_i \equiv q_i\}_{i=1}^k \subseteq \mathcal{C}$ then, by clause (14) of Theorem 5, we have $\mathcal{O}, \mathcal{O}', \mathcal{A}, \kappa \models \lambda$, and also $\mathcal{O}, \mathcal{O}', \mathcal{A}, \lambda \models \kappa$. $\square$

The lesson from Theorems 8 and 9 is that, for interlinking two datasets, if there is a key for one dataset and a proper alignment from the key to the vocabulary of the other dataset, then using the key or the strong link key entailed by the key and the alignment is strictly equivalent.

However, as explained in the previous section, weak link keys may exist even when keys and proper alignments do not exist. As a conclusion, in general, link keys are more suitable than keys for data interlinking. Thus, data interlinking algorithms are justified in discovering link keys rather than keys and alignments.

9. Conclusions and Further Work

The relation between keys and link keys is much more subtle than one may think at first sight, and one may not be replaced by the other without care. In particular, we have shown that data interlinking with keys requires (a) a proper alignment (Theorems 1 and 2), and (b) completion in the case of eq-keys (Theorem 2). Data interlinking with link keys, in turn, does not need alignments (Theorems 3 and 4) but needs anyway completion in the case of eq-link keys (Theorem 4).

Strong link keys entail keys by definition, and we have proven that keys with proper alignments entail strong link keys (Theorem 5). In this case, the links generated by a strong link key are the same as the links generated by their associated side keys and alignments (Theorems 8 and 9).

Neither strong nor weak link keys require ontology alignments for data interlinking but weak link keys are not associated with keys in the ontologies separately (Theorem 6; if they are, then they are strong link keys), and yet they may be useful for interlinking datasets.

These results provide a clear picture of the key-inspired devices available for data interlinking. They can be easily integrated within the generalised notions of a key and a link key.

The work presented in this paper contributes grounding data interlinking methods based on keys and link keys. In particular, it justifies the work for directly extracting weak link keys [8] instead of searching for keys with matching alignments. Link key extraction directly focuses on what may be used for data interlinking instead of generating keys and alignments that may not be possible to exploit. Also, when no strong link key exists, it may find a suitable weak link key, though key extraction would not return useful keys.

The clarification of the semantics of link keys tackled in this paper should also lead to more powerful data interlinking methods. For instance, using the formal semantics of link keys, reasoners may derive link keys. Such entailed link keys could be exploited by extended versions of reasoning-based data interlinking tools. This should also enable breaking the extraction+interlinking process by reasoning on link keys before interlinking in order to provide more accurate links, eventually more efficiently.

Acknowledgements

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

Manuel Atencia, Jérôme David and Jérôme Euzenat acknowledge the numerous discussions they had with Michel Chein, Madalina Croitoru, Chan Le Duc, 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] 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.

[6] 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.

[7] 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.

[8] 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.

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

[17] 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.

[18] M. Al-Bakri, M. Atencia, S. Lalande and M. Rousset, Inferring Same-As Facts from Linked Data: An Iterative Import-by-Query Approach, in: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, AAAI Press, 2015, pp. 9–15.

[19] 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.

[20] 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.

[21] 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.

[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.

Appendix A. Proofs of Section 8.1

This appendix describes the relations between keys and link keys in a more precise way than it was done in Section 8.1. Some of the results of Section 8.1 are synthetic consequences of the ones presented here.

Proposition 9. *The following holds:*

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

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

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

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 $\delta = \delta'$. □

Proposition 10 is the counterpart of Proposition 9 for strong link keys. Notice that this time the consequent is a key in the union of classes, and not only in the intersection.

Proposition 10. *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 that $\delta \in C^{\mathcal{I}}$, $\delta' \in D^{\mathcal{I}}$. 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 9 to conclude that $\delta = \delta'$.

The other two statements can be proven similarly. □

Proposition 11 is the converse of Proposition 10. 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.

Proposition 11. *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 5, 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 5, 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 5, 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 5. □