Tableau extensions for reasoning with link keys

Maroua Gmati[2,1], Manuel Atencia[1,2], and Jérôme Euzenat[2,1]

[1]Univ. Grenoble Alpes and [2]INRIA, France
gm.gmati.maroua@gmail.com,{Manuel.Atencia,Jerome.Euzenat}@inria.fr

Abstract. Link keys allow for generating links across data sets expressed in different ontologies. But they can also be thought of as axioms in a description logic. As such, they can contribute to infer ABox axioms, such as links, or terminological axioms and other link keys. Yet, no reasoning support exists for link keys. Here we extend the tableau method designed for $\mathcal{ALC}$ to take link keys into account. We show how this extension enables combining link keys with classical terminological reasoning with and without ABox and TBox and generate non trivial link keys.

1 Motivation

Part of the added value of linked data lies in the links between entities denoting the same individual in data sets issued by different sources as it allows for making inferences across data sets. For instance, links may identify the same books and articles in different bibliographical data sources. So finding the manifestation of the same entity across several data sets is an important task of linked data.

One way of identifying entities is to use link keys which generalise keys usually found in data bases to the case of different data sets. A link key [3] is a statement of the form:

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

stating that whenever an instance of the class Livre has the same values for properties auteur and titre as an instance of class Book has for properties creator and title, then they denote the same entity. Such keys are slightly more complex than those of databases because, in RDF, properties are not necessarily functional (they may have several values) and their values may be other objects.

One further difference is that RDF data, together with ontologies expressed in the OWL or RDFS languages, are logic theories. In such a context, a link key is a statement as any other logical statement. As such, it may contribute deducing other statements. Indeed, the above link key entails:

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

or

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

whenever Novel is subsumed by Book.

Hence, it is possible to reason on link keys in different ways:

– deducing link keys from OWL statements,
– deducing link keys from link keys,
– deducing OWL statements from link keys.

Our goal is to study reasoning procedures for link keys. For that purpose, we define a preliminary extension of the tableau method for $\mathcal{ALC}$ dealing with link keys and we provide examples for each of the inference types above.

In the following, we first discuss related work (§2) and define more precisely the problem (§3). Then we present a tableau extensions allowing for ABox reasoning with link keys (§4) and for reducing link key inference to that ABox reasoning (§5).

2 Related work

Data interlinking is a very active area [9]. Two main approaches are used for coping with this problem: numerical methods and logical methods. The numerical methods usually compute a similarity between resources based on their property values to establish links between those which are highly similar [11; 13]. Logical methods for data interlinking use an axiomatic characterisation of what makes two resources the same to find the links between different data sets [12; 1; 3].

This work belongs to the logic-based approach. It uses a generalisation of keys in relational databases, called link keys, for expressing the condition for identifying resources across different ontologies. Keys in databases indicate that a set of properties uniquely identifies individuals. Relational properties are functional (have only one value) and concrete (the value comes from a data type).

RDF data differ from relational data in their properties, which are not functional, and their values, which may be resources. Hence, keys have been generalised to cope with this problem [2]. RDF property values are considered the same if they are the same concrete value or are interpreted as the same individual. Coping with non functionality lead to define two different types of keys: in-keys and eq-keys. Eq-keys require that the properties of two objects have exactly the same values for them to be equal, while in-keys only require that each property shares at least one common value. In this work, we focus on in-keys.

Keys may be introduced in description logics either as global constraints in a specific KBox [7; 10], or as a new concept constructor [6]. [7] discusses the introduction of keys in the $\mathcal{DLR}$ logic but does not provide any reasoning method. Keys based on features (functional roles whose value is from a concrete domain) have been introduced within the $\mathcal{ALCOK}(\mathcal{D})$ and $\mathcal{SHROIC}(\mathcal{D})$ logics [10] and an extension of the tableau method has been provided to deal with these logics.

Keys identify objects within a single data source with a single schema. Link keys have been designed for coping with heterogeneous data sources [8]. They can be seen either as a generalisation of keys across two data sets or as a merge between keys and alignments. They express conditions by which two individuals, from two different classes, must be considered the same by comparing values of properties.

Link keys raise two distinct problems: the first one is to extract link keys from data sets [3]; the second one is to take advantage of link keys to generate links. These two problems may be thought of as two steps of a link generation procedure: first extract link keys, then generate links from them.

Here we tackle a third problem (not unrelated to the second one): reasoning with link keys, i.e., infering links, ontological and assertional statements as well as other link keys. We define this problem more precisely below.

3 Preliminaries

Data interlinking is the process of generating links across data sets that can help finding equivalent resources representing the same entity on the web for linked data. These links are usually owl:sameAs statements between two resources across different RDF data sets. We will consider that these data sets are description logic knowledge bases ($KB = \langle T, A \rangle$) made of a TBox T and an ABox A. Description logics [4] are at the basis of OWL, so this is quite natural.

We decided to extend the tableau method used for checking entailment in the $\mathcal{ALC}$ family of description logics for several reasons:

- $\mathcal{ALC}$ is a subset of OWL;
- The tableau method is extensible, so it is possible to add rules for dealing with more expressive logics. We could have started with procedure specific to less expressive logics ($\mathcal{EL}$, DL-Lite, OWL-RL), but we could barely extend them.

An $\mathcal{ALC}$ TBox is a set of general concept inclusion axioms of the form $C \sqsubseteq C'$. Concepts are defined by:

$$C = A|\bot|\top|C \sqcap C'|C \sqcup C'|\neg C|\forall R.C|\exists R.C$$

and roles are simply atomic roles ($R = r$).

The ABox is made of assertions of the form $C(a)$ and $r(a, b)$. We will use two specific statements $a = b$ and $\neq b$ which are interpreted as usual. These two predicates are the transcription of `owl:sameAs` and `owl:differentFrom`.

The semantics of such logics is defined by interpretations $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ such that $\Delta^\mathcal{I}$ is a non empty set and $\cdot^\mathcal{I}$ is a function such that: $a^\mathcal{I} \in \Delta^\mathcal{I}$, $C^\mathcal{I} \subseteq \Delta^\mathcal{I}$, and $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ with:

$$(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I} \qquad \bot^\mathcal{I} = \varnothing \qquad \top^\mathcal{I} = \Delta^\mathcal{I}$$
$$(C \sqcap C')^\mathcal{I} = C^\mathcal{I} \cap C'^\mathcal{I} \qquad (\forall r.C)^\mathcal{I} = \{\delta \in \Delta^\mathcal{I} | \forall \delta'; \langle \delta, \delta' \rangle \in r^\mathcal{I} \Rightarrow \delta' \in C^\mathcal{I}\}$$
$$(C \sqcup C')^\mathcal{I} = C^\mathcal{I} \cup C'^\mathcal{I} \qquad (\exists r.C)^\mathcal{I} = \{\delta \in \Delta^\mathcal{I} | \exists \delta' \in C^\mathcal{I}; \langle \delta, \delta' \rangle \in r^\mathcal{I}\}$$

An interpretation satisfies an axiom (denoted by $\mathcal{I} \models \alpha$) in the following conditions:

$$\mathcal{I} \models C(a) \text{ iff } a^\mathcal{I} \in C^\mathcal{I} \qquad \mathcal{I} \models r(a, b) \text{ iff } \langle a^\mathcal{I}, b^\mathcal{I} \rangle \in r^\mathcal{I}$$
$$\mathcal{I} \models a = b \text{ iff } a^\mathcal{I} = b^\mathcal{I} \qquad \mathcal{I} \models a \neq b \text{ iff } a^\mathcal{I} \neq b^\mathcal{I}$$
$$\mathcal{I} \models C \sqsubseteq C' \text{ iff } C^\mathcal{I} \subseteq C'^\mathcal{I}$$

A model of a knowledge base KB is an interpretation satisfying all its axioms and an assertion α is entailed by a knowledge base (denoted by $KB \models \alpha$) if it is satisfied by all the models of KB.

We extend description logics with a KBox K which contains link keys instead of simple keys. The KBox is a set of link keys: $\{\langle p_i, q_i\rangle\}_{i\in I}\ linkkey^w_{in}\ \langle C, D\rangle$ with C and D two classes coming from different data sets and p_i and q_i roles, from the data sets of C and D respectively, indexed by a finite set of indices I. Since we concentrate specifically on weak in-link keys, we use the keyword $linkkey^w_{in}$.

The semantics of description logics is extended to cover link keys: An interpretation $\mathcal{I}$ satisfies $(\{\langle p_i, q_i\rangle\}_{i\in I}\ linkkey^w_{in}\ \langle C, D\rangle)$ iff, for any $\delta \in C^{\mathcal{I}}$ and $\eta \in D^{\mathcal{I}}$,

$$\bigwedge_{i\in I}(\exists z_i \in \Delta^{\mathcal{I}}; \langle \delta, z_i\rangle \in p_i^{\mathcal{I}} \wedge \langle \eta, z_i\rangle \in q_i^{\mathcal{I}}) \Rightarrow \delta = \eta$$

Any key $\{p_i\}_{i\in I}\ keyFor\ C$ is equivalent to the link key $\{\langle p_i, p_i\rangle\}_{i\in I}\ linkkey\ \langle C, C\rangle$. In this paper, we only consider hierarchical KBoxes, i.e., KBoxes in which there cannot be circular dependencies between link keys.

It is possible, to establish entailment rules for link keys considered as assertions:

$$\{\langle p_i, q_i\rangle\}_{i\in I}\ linkkey^w_{in}\ \langle C, D\rangle \models \{\langle p_i, q_i\rangle\}_{i\in I\cup J}\ linkkey^w_{in}\ \langle C, D\rangle$$
$$\{\langle p_i, q_i\rangle\}_{i\in I}\ linkkey^w_{in}\ \langle C, D\rangle, C' \sqsubseteq C \models \{\langle p_i, q_i\rangle\}_{i\in I}\ linkkey^w_{in}\ \langle C', D\rangle$$
$$\{\langle p_i, q_i\rangle\}_{i\in I}\ linkkey^w_{in}\ \langle C \sqcup C', D\rangle \models \{\langle p_i, q_i\rangle\}_{i\in I}\ linkkey^w_{in}\ \langle C, D \sqcap D'\rangle$$

Proving all such rules one by one is tedious, so an inference procedure for doing this would be useful.

4 Links and Abox entailments with link keys

The basic way of applying link keys is to start with two datasets A and A' described by two ontologies T and T' and a set K of link keys across these ontologies and to generate links, i.e., statements of the form $a = b$ with a and b from each data set.

We consider this problem more widely as that of reasoning in a knowledge base[1] $KB = \langle T \cup T', K, A \cup A'\rangle$. We will consider more precisely the decision problem of checking the entailment of any ABox axiom α from such a knowledge base.

Problem: ABOX AXIOM ENTAILMENT
INSTANCE:
– A knowledge base $KB = \langle T, K, A\rangle$
– An ABox assertion α.
QUESTION: Does $KB \models \alpha$?

4.1 Tableau rule for applying link keys

The tableau method is the classical technique to reason with $\mathcal{ALC}$. Explaining the method is out of the scope of this paper (see [4; 5]). To summarise, this method attempts to find a model of a knowledge base $KB = \langle T, A\rangle$ in negation normal form.

[1] We assume no unwanted name conflicts, i.e., the same name or URI in both data sets must have the same interpretation.

For that purpose, it starts with a representation of the ABox A and applies rules (see Appendix) guided by T until no rule is applicable [5]. In such a case, there exists a model of KB. However, there are special constraints, called clashes, which express the impossibility to build a model: if such a clash is satisfied, then the current representation cannot be turned into a model and the algorithm must explore eventual alternative representations. Finally, for guaranteeing the termination of the process due to infinitely expanding rules, provisions are taken for detecting this and blocking some parts of the representation to be expanded. We rely here on the classical tableau method for $\mathcal{ALC}$ and use a graphical representation of partial models in which nodes (x) represent individuals labeled ($L(x)$) by sets of class descriptions and edges ($\langle x, y\rangle$) represent relations labeled ($L(\langle x, y\rangle)$ by role descriptions. The tableau method may be used for finding a model or for proving that there exist no model of a knowledge base.

In order to tackle the ABox Axiom entailment problem within the tableau method we introduce the Linkkey-rule:

Linkkey-rule

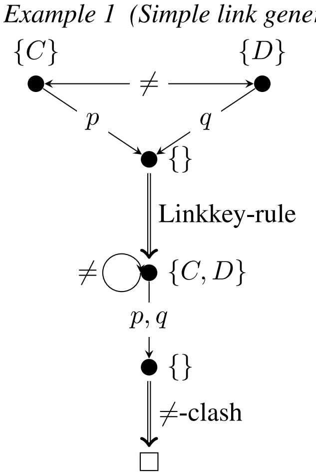
Condition: $\{\langle p_i, q_i\rangle\}_{i\in I}\ linkkey^w_{in}\ \langle C, D\rangle \in K$,
$\exists x, y$, not blocked, such that $C \in L(x)$, $D \in L(y)$, and
$\forall i \in I$, $\exists z_i$, such that $p_i \in L(\langle x, z_i\rangle)$ and $q_i \in L(\langle y, z_i\rangle)$

Action: $L(x) := L(x) \cup L(y)$
Replace y by x in all edges starting from or ending at y
Suppress node y

This rule is sound, i.e., any model has to satisfy it, as it strictly follows the semantics of link keys. It generalises rule T14 in [10] to link keys.

The use of this rule for checking a link $a = b$ can be illustrated on the straightforward Example 1: For proving the entailment of $a = b$, we proceed by refutation, i.e., we prove that it is not possible to create a model satisfying the antecedents and the negation of the consequence ($a \neq b$). A representation of such a model is created and the rules are applied on it. The Linkkey-rule merges the two nodes satisfying the link key condition which makes them fall under the $\neq$-clash.

Example 1 (Simple link generation).



Problem:
$\langle p, q\rangle\ linkkey^w_{in}\ \langle C, D\rangle$,
$C(a), D(b), p(a, v), q(b, v)$
$\models a = b$?

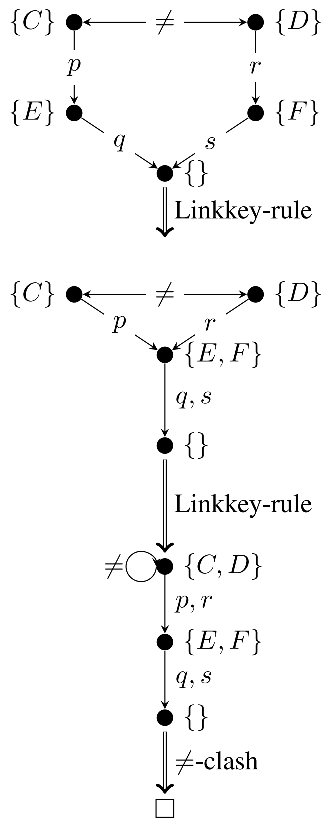
Knowledge base:

$$T = \{\}$$
$$K = \{\langle p, q\rangle\ linkkey^w_{in}\ \langle C, D\rangle\}$$
$$A = \{C(a), D(b), p(a, v), q(b, v), a \neq b\}$$

4.2 Combining link key reasoning and ABox reasoning

Example 2 shows the use of these rules for chaining the use of two link keys. However, it may be used in any ABox reasoning development.

Example 2 (Chaining link generation).



Problem:
$\langle p, r\rangle\ linkkey^{w}_{in}\ \langle C, D\rangle$,
$\langle q, s\rangle\ linkkey^{w}_{in}\ \langle E, F\rangle$,
$C(a), p(a, c), E(c), q(c, v), D(b), r(b, d), F(d), s(d, v)$
$\models a = b$?

Knowledge base:

$$T = \{\}$$
$$K = \{\langle p, r\rangle\ linkkey^{w}_{in}\ \langle C, D\rangle, \langle q, s\rangle\ linkkey^{w}_{in}\ \langle E, F\rangle\}$$
$$A = \{C(a), p(a, c), E(c), q(c, v), D(b), r(b, d), F(d), s(d, v), a \neq b\}$$

Solving the ABox entailment problem may not be the most efficient way to generate links from RDF especially if the size of the considered ABox is very large. A more interesting use of such reasoning is for checking link key entailment.

5 Link key entailment

The link key entailment problem aims at checking if a link key is entailed by a knowledge base. Because this resorts to the terminological level, i.e., without regard to a particular ABox, it is defined only on a knowledge base made of a TBox and a KBox. Indeed, some link keys may be entailed from terminological axioms, some others from other link keys of a mix of this.

Problem: LINK KEY ENTAILMENT
INSTANCE:
– A knowledge base $KB = \langle T, K\rangle$
– A link key λ.
QUESTION: Does $KB \models \lambda$?

5.1 Reducing link key entailment to knowledge base satisfiability

The tableau method cannot be directly used for refuting a link key axiom because there is no negation for link keys: a link key is an axiom of our logic, the negation of a link key is not.

Other authors have considered expressing keys as simple concept constructors [6]:

$$C \sqsubseteq \mathsf{key}(\{p_i\}_{i\in I})$$

This could be transposed for link keys as:

$$\langle C, D\rangle \sqsubseteq \ linkkey^w_{in}\ (\{\langle p_i, q_i\rangle\}_{i\in I})$$

such statements would solve half of the problem as it is possible to negate the subsumption statements, but this would lead to strange statements as they concern pairs of classes. They would also be stronger than, and not equivalent to, our actual link key statements.

Adding the negation of a link key to the logic is another solution to this problem. However, since its only use would be for the decision procedure, we preferred to avoid this solution.

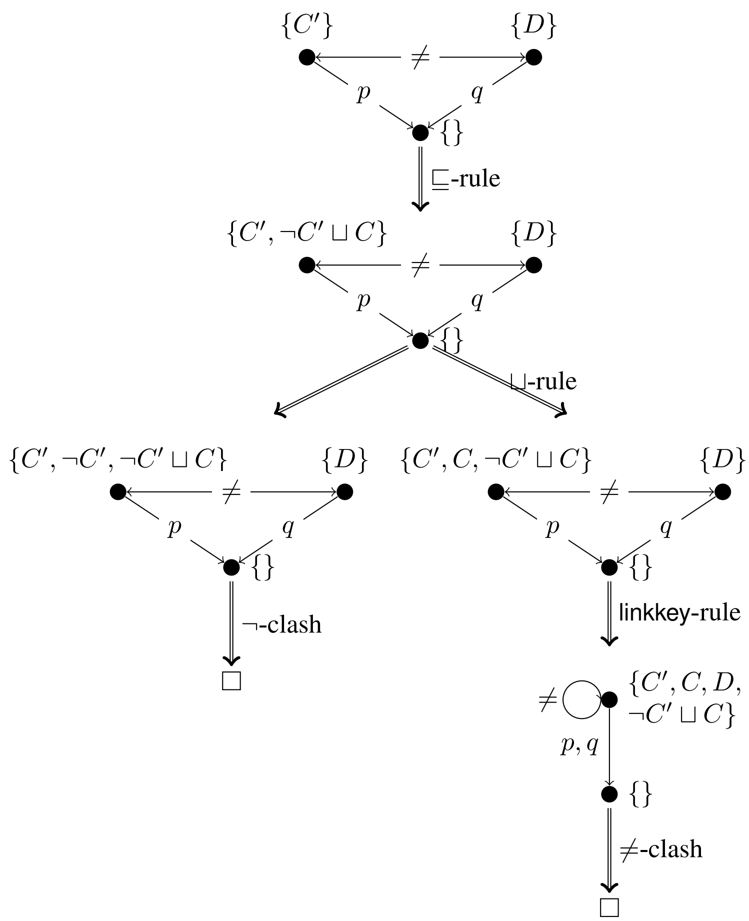
We choose a simpler method given that our goal is simply to have negated link keys as the statement to refute: we use a set of ABox statements as witness of the unsatisfiability of a link key. This set is given by the function ρ:

$$\rho(\{\langle p_i, q_i\rangle\}_{i\in I}\ linkkey^w_{in}\ \langle C, D\rangle) = \{C(x), D(y), x \neq y\} \cup \{p_i(x, v_i), q_i(y, v_i)\}_{i\in I}$$

Checking the entailment of a link key λ by a knowledge base $\langle \varnothing, T, K\rangle$ can be reduced to checking the satisfiability of the knowledge base $KB = \langle T, K, \rho(\lambda)\rangle$. Any model in which the link key λ is not valid satisfies $\rho(\lambda)$. Hence, if KB is satisfiable, then λ is not entailed.

Example 3 (Link key inference from other link keys and TBox).
Problem: $\langle p,q\rangle\ linkkey^{w}_{in}\ \langle C,D\rangle, C' \sqsubseteq C \models \langle p,q\rangle\ linkkey^{w}_{in}\ \langle C',D\rangle$?
Knowledge base:

$$T = \{C' \sqsubseteq C\}$$
$$K = \{\langle p,q\rangle\ linkkey^{w}_{in}\ \langle C,D\rangle\}$$
$$A = \{C'(a), D(b), p(a,v), q(b,v), a \neq b\}$$



For instance, one of the example given in Section 3 is a link key entailed from another link key and terminological axioms. Example 3 shows how this is performed without introducing any new rule or clash in the tableau procedure.

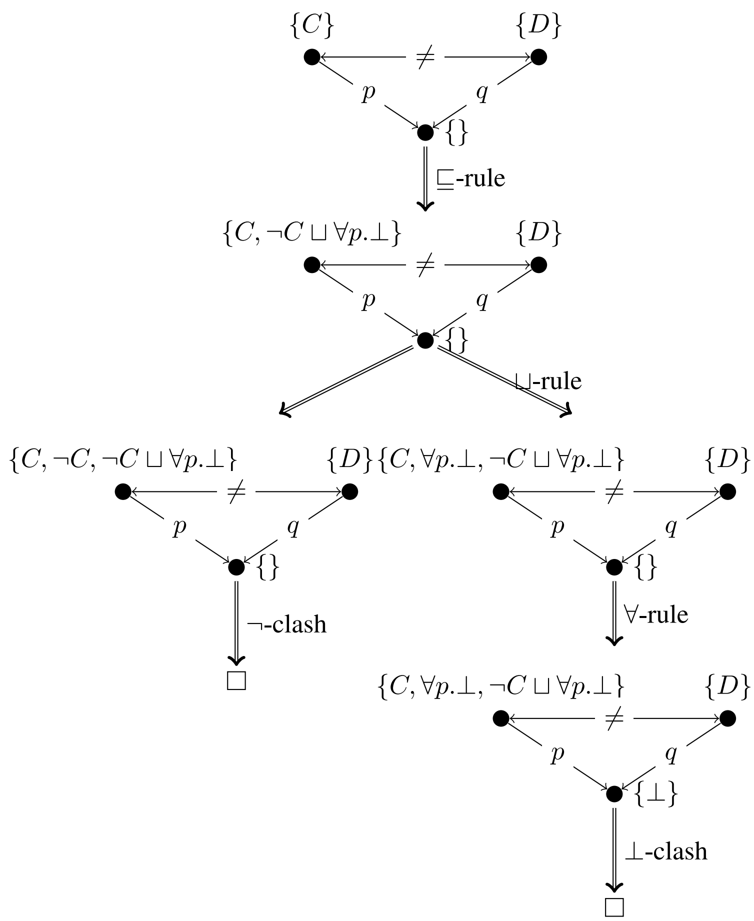
This shows the importance of being able to reason with the ABox, since the refutation of the KB is mostly carried out by reasoning in the ABox even if the problem does not have an ABox. It also shows that link key rules can be adequately interleaved with $\mathcal{ALC}$ rules. This suggests that extensions can properly work in the same way.

5.2 Link key entailed from terminological axioms

Some other link keys may only be entailed by terminological axioms. We illustrate this by the counter-intuitive Example 4. This inference is of little use, but it shows that the method indeed proves this valid link key.

Example 4 (Link key inference from TBox alone).
Problem:
$C \sqsubseteq \forall p.\bot \models \langle p,q\rangle\ linkkey^{w}_{in}\ \langle C,D\rangle$?
Knowledge base:

$$T = \{C \sqsubseteq \forall p.\bot\}$$
$$K = \{\}$$
$$A = \{C(a), p(a,v), D(b), q(b,v), a \neq b\}$$



It is noteworthy that Example 4 does not use the Linkkey-rule; it only relies on the encoding of the problem and classical $\mathcal{ALC}$ reasoning.

The use of the tableau method allows both to check inference rules and to determine minimal logics in which they hold. Example 4 shows that the given entailment holds in any description logic, with $\mathcal{ALC}$-style models, which accepts subsumption axioms ($\sqsubseteq$), universal quantification ($\forall$) and the empty concept ($\bot$).

6 Conclusion and future work

Link keys are very useful for generating links from data sources, but they can be studied independently from data sources as axioms. In order to prove when a particular knowledge base, eventually with link keys, entails a particular link key, we proposed extensions of the tableau method for $\mathcal{ALC}$ enabling the interpretation of link keys. We showed that these extensions also allow for checking link key entailment.

We considered the tableau method because it is well-adapted to $\mathcal{ALC}$ and thus to OWL as a whole. Weaker fragments of OWL ($\mathcal{EL}$, DL-Lite, OWL-RL) are supported efficiently by other reasoning methods. It would be interesting to investigate the opportunity to reason with and about link keys in this context.

This work is preliminary and many developments may be undertaken from here. We discuss a few of them.

First, we need to determine the properties of the proposed extension. We have yet no formal proof to offer, but basic arguments for these. Although correctness of rules and clash independently seems to be straightforward, proving the completeness of the designed procedure with various logics must be considered. Termination can be guaranteed with a blocking mechanisms and because no rule erases any other rule condition (the Linkkey-rule merges nodes, but preserves the constraints on these nodes). Finally, the current link key rule should not increase the complexity of existing tableau methods since the rule does not introduce branches. The Linkkey-rule may offer new development opportunities by merging nodes but (*i*) this process is bounded, and (*ii*) new tableau developments should not go beyond current complexity.

Then, we want to implement these extensions. This would allow us to check automatically the link key inference rules that we designed. It would also be interesting, in a further step, to develop techniques to generate (specific) entailed assertions in a forward deduction style.

Finally, it would be worth considering the other type of link key conditions (eq-link keys). However, this may not be easy to integrate with the open world aspect of description logic semantics.

A $\mathcal{ALC}$+Linkkey rules

We provide the full set of rules for helping the reader to read the examples.

A.1 Completion rules

$\sqcap$-rule
Condition: $C \sqcap D \in L(x)$, x is not blocked; $\{C, D\} \not\subseteq L(x)$
Action: $L(x) := L(x) \cup \{C, D\}$
$\sqcup$-rule
Condition: $C \sqcup D \in L(x)$, x is not blocked; $C \notin L(x)$, $D \notin L(x)$
Action: $L(x) := L(x) \cup \{C\}$, or $L(x) := L(x) \cup \{D\}$
$\exists$-rule
Condition: $\exists r.C \in L(x)$, x is not blocked; $\nexists y; r(x, y) \wedge C \in L(y)$
Action: create a new node y with $L(\langle x, y\rangle) = \{r\}$ and $L(y) = \{C\}$
$\forall$-rule
Condition: $\forall r.C \in L(x)$, x is not blocked; $\exists y; r(x, y) \wedge C \notin L(y)$
Action: $L(y) := L(y) \cup \{C\}$
$\sqsubseteq$-rule
Condition: $C \sqsubseteq D \in T$, x is not blocked, $\neg C \sqcup D \notin L(x)$
Action: $L(x) := L(x) \cup \{\neg C \sqcup D\}$
Linkkey-rule
Condition: $\{\langle p_i, q_i\rangle\}_{i\in I}\ linkkey^{w}_{in}\ \langle C, D\rangle \in K$,
$\exists x, y$, not blocked, such that $C \in L(x)$, $D \in L(y)$, and
$\forall i \in I$, $\exists z_i$, such that $p_i \in L(\langle x, z_i\rangle)$ and $q_i \in L(\langle y, z_i\rangle)$
Action: $L(x) := L(x) \cup L(y)$
Replace y by x in all edges starting from or ending at y
Suppress node y

A.2 Clash conditions

$$\begin{aligned}
\neg\textbf{-clash}: &\ \exists x; \{C, \neg C\} \subseteq L(x)\\
\bot\textbf{-clash}: &\ \exists x; \bot \in L(x)\\
\neq\textbf{-clash}: &\ \exists \langle x, x\rangle; \neq \in L(\langle x, x\rangle)
\end{aligned}$$

References

1. Mustafa Al-Bakri, Manuel Atencia, Steffen Lalande, and Marie-Christine Rousset. Inferring same-as facts from linked data: an iterative import-by-query approach. In Blai Bonet and Sven Koenig, editors, *Proc. 29th Conference on Artificial Intelligence (AAAI), Austin (TX US)*, pages 9–15, 2015.
2. Manuel Atencia, Michel Chein, Madalina Croitoru, Jérôme David, Michel Leclère, Nathalie Pernelle, Fatiha Saïs, François Scharffe, and Danai Symeonidou. Defining key semantics for the RDF datasets: experiments and evaluations. In *Proc. 21st International Conference on Conceptual Structures (ICCS), Iasi (RO)*, pages 65–78, 2014.
3. Manuel Atencia, Jérôme David, and Jérôme Euzenat. Data interlinking through robust linkkey extraction. In *Proc. 21st european conference on artificial intelligence (ECAI), Praha (CZ)*, pages 15–20, 2014.
4. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The description logic handbook: theory, implementations and applications*. Cambridge University Press, 2003.
5. Franz Baader, Ian Horrocks, and Ulrike Sattler. Description logics. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*, chapter 3, pages 135–179. Elsevier, Amsterdam (NL), 2008.
6. Alexander Borgida and Grant Weddell. Adding uniqueness constraints to description logics (preliminary report). In *Proc. 5th Deductive and Object-Oriented Databases conference (DOOD)*, volume 1341 of *LNCS*, pages 85–102, Montreux (CH), 1997.
7. Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Vardi. View-based query processing for regular path queries with inverse. In *Proceedings of the 19th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 58–66, 2000.
8. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer, Heidelberg (DE), 2 edition, 2013.
9. Alfio Ferrara, Andriy Nikolov, and François Scharffe. Data linking for the semantic web. *International Journal of Semantic Web and Information Systems*, 7(3):46–76, 2011.
10. Carsten Lutz, Carlos Areces, Ian Horrocks, and Ulrike Sattler. Keys, nominals, and concrete domains. *Journal of Artificial Intelligence Research*, 23:667–726, 2005.
11. Axel-Cyrille Ngonga Ngomo and Sören Auer. LIMES: A time-efficient approach for large-scale link discovery on the web of data. In *Proc. 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2312–2317, Barcelona (ES), 2011.
12. Fatiha Saïs, Nathalie Pernelle, and Marie-Christine Rousset. Combining a logical and a numerical method for data reconciliation. *Journal on Data Semantics*, 12:66–94, 2009.
13. Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and maintaining links on the web of data. In *Proc. 8th International Semantic Web Conference (ISWC)*, volume 5823 of *Lecture notes in computer science*, pages 650–665, Chantilly (VA US), 2009.