Decentralized Reasoning on a Network of Aligned Ontologies with Link Keys

Jérémy Lhez[1], Chan Le Duc[1], Thinh Dong[2], and Myriam Lamolle[1(✉)]

[1] LIASD, Université Paris 8 - IUT de Montreuil, Montreuil, France
{lhez,leduc,lamolle}@iut.univ-paris8.fr
[2] University of Danang, Da Nang, Vietnam
dnnthinh@kontum.udn.vn

Abstract. Link keys are recently introduced to formalize data interlinking between data sources. They are considered as a new kind of correspondences included in ontology alignments. We propose a procedure for reasoning in a decentralized manner on a network of ontologies with alignments containing link keys. In this paper, the ontologies involved in such a network are expressed in the logic $\mathcal{ALC}$ while the alignments can contain concept, individual and link key correspondences equipped with a loose semantics. The decentralized aspect of our procedure is based on a process of knowledge propagation through the network via correspondences. This process allows to reduce polynomially global reasoning to local reasoning.

1 Introduction

Reasoning on a network of aligned ontologies has been investigated in different contexts where the semantics given to correspondences differs from one to another. To be able to develop a procedure for reasoning on a network of aligned ontologies, it is needed to equip the correspondences of the alignment with a semantics compatible with those defined in the ontologies. A simple approach to this issue consists in considering the correspondences as logical axioms expressed in the ontology language and merging all involved ontologies and the alignments into a unique ontology. In this case, the reasoning problem on such a network of aligned ontologies can be expressed as the following usual entailment:

$$\bigcup_{1\leq i\leq n} O_i \cup \bigcup_{1\leq i<j\leq n} A_{ij} \models \alpha \tag{1}$$

where O_i is an $\mathcal{ALC}$ ontology, A_{ij} is an alignment between O_i and O_j, and α[1] is a link key or a concept assertion/axiom.

This approach is characterized by the following two main aspects: (i) the correspondences of the alignments are semantically handled as ontology assertion/axioms, and (ii) reasoning is performed on the unique ontology in a centralized manner, *i.e.* all reasoning tasks are carried out on a single location with

[1] Consistency of the network can be reduced to the entailment (1) with $\alpha = \bot(x)$.

C. Ghidini et al. (Eds.): ISWC 2019, LNCS 11778, pp. 418–434, 2019.
https://doi.org/10.1007/978-3-030-30793-6_24

a reasoner. Such an approach is quite unexploitable in the context of the Web where numerous ontologies and alignments are located in different sites. There have been researches [1–6], which aimed to distribute reasoning over several locations. However, these approaches usually lead to an exponential blow-up of message passing between local reasoners associated with different locations [5,10]. The main reason for this exponential blow-up is due to the strong semantics of the correspondences involved in the alignments.

In this paper, we introduce a new semantics of correspondences which are weaker than the usual one and propose a procedure for reasoning on a network of aligned ontologies in a decentralized manner—that means—reasoning can be independently performed on different sites following a process of knowledge propagation through the network of the ontologies via the alignments with link keys. Usefulness of link keys in Semantic Web applications and the problem of reasoning with them in the centralized context have been investigated by Atencia and Gmati [7,8].

To illustrate our settings, we consider the following example in which knowledge is modelled in description logics. This formalism is used to encode the semantics of web languages such as OWL2.

Example 1. Consider two ontologies, denoted O_1 and O_2, where O_1 describes a terminology used by conference organizers, and O_2 stores information about researchers and conferences they have attended. In O_1, there are classes $\mathsf{Participant}$, $\mathsf{Presenter}$, $\mathsf{DemoPaperPresenter}$; and a property $\mathsf{present}$. In O_2, we can find classes $\mathsf{Researcher}$, $\mathsf{PhDStudent}$, $\mathsf{Developer}$; and a property $\mathsf{registerTo}$ (*i.e.* someone registers to present a paper).

An alignment A_{12} tells us that $\mathsf{DemoPaperPresenter}$ is simultaneously aligned with $\mathsf{Researcher}$ and $\mathsf{Developer}$.

$$\mathsf{DemoPaperPresenter} \rightarrow \mathsf{Researcher} \tag{2}$$

$$\mathsf{DemoPaperPresenter} \rightarrow \mathsf{Developer} \tag{3}$$

In addition, A_{12} contains a link key which says that if a participant presents in the conference the same paper as that to which a researcher registers the conference then the participant and the researcher would be the same person.

$$\{\langle \mathit{present}, \mathit{registerTo}\rangle\}\ \mathit{linkkey}\ \langle \mathit{Participant}, \mathit{Researcher}\rangle \tag{4}$$

If we now add to O_1 and O_2 the following axioms/assertion

$$O_1 : \mathsf{DemoPaperPresenter}(\mathsf{Anna}) \tag{5}$$

$$O_1 : \mathsf{DemoPaperPresenter} \sqsubseteq \mathsf{Participant} \tag{6}$$

$$O_2 : \mathsf{PhDStudent} \sqsubseteq \mathsf{Researcher} \tag{7}$$

$$O_2 : \mathsf{Researcher} \sqsubseteq \neg\mathsf{Developer} \tag{8}$$

then a reasoner can find the entailment:

$$\begin{aligned} &O_1 \cup O_2 \cup A_{12} \models \\ &\{\langle \mathsf{present}, \mathsf{registerTo}\rangle\}\ \text{linkkey}\ \langle \mathsf{DemoPaperPresenter}, \mathsf{PhDStudent}\rangle \end{aligned} \tag{9}$$

This entailment holds because of the axioms (6), (7) and the link key (4). If we now interpret the correspondences (2) and (3) as subsumption in the standard semantics then the network $O_1 \cup O_2 \cup A_{12}$ is inconsistent because of the assertion/axiom (5) and (8). However, if we interpret these correspondences as a means for propagating concept unsatisfiability, *i.e.* unsatisfiability of the "subsumer" implies unsatisfiability of the "subsumee", then the network is consistent. In the following sections, we show that the weakened semantics corresponding to the latter interpretation of concept correspondences leads to a substantial change of the computational complexity of algorithms for reasoning.

In addition, the weakened semantics would not be really interesting for the correspondences (2) and (3). However, it would be more relevant for correspondences between ontologies of different nature. Given two ontologies about `equipment` and `staff` and a correspondence $\mathsf{Computer} \rightarrow \mathsf{Developer}$ between them. With this correspondence, the weakened semantics tells us that if there is no developer then there is no computer. The standard semantics is irrelevant in this case. □

Based on the weakened semantics of alignments, we introduce in this paper the notion of consistency for a network of ontologies with alignments containing link keys (or an *ontology network* for short). Then, we propose an algorithm for checking consistency of an ontology network by reducing this task to checking consistency of each ontology which is polynomially extended. This consists in (i) propagating individual equalities of the form $a \approx b$ through all ontologies of the network via individual correspondences of the same form $a \approx b$, (ii) applying link keys in the alignments, which may lead to add new individual correspondences, (iii) propagating concept unsatisfiabilities through all ontologies of the network via concept correspondences of the form $C \rightarrow D$. We show that the complexity of the process of knowledge propagation is polynomial in the size of the network. In addition, we also prove that consistency of the ontologies and alignments extended by this process of knowledge propagation is equivalent to consistency of the network.

The remainder of the paper is organised as follows. Section 2 positions our work with respect to works on distributed reasoning in description logics. Section 3 describes the logic $\mathcal{ALC}$ with individuals, alignments, a new semantics of alignments and inference services. Section 4 provides the algorithms for propagating individual equalities, applying link keys and propagating concept satisfiabilities. We also prove that reasoning on the ontology network is reducible to reasoning on each ontology extended by the algorithms, and this reduction is polynomial in the size of the ontology network. Section 5 presents examples of the use of the algorithms. Section 6 describes the architecture of Draon in which the algorithms are implemented in a decentralized manner. We also report some experimental results. Section 7 concludes the paper and presents future work.

2 Related Work

In the literature, there have been several reasoning approaches which either (i) merge all ontologies and alignments into a unique ontology and perform reasoning over that unique ontology, or (ii) use a distributed semantics such as DDL (Distributed Description Logics) [1], E-connection [2], IDDL (Integrated Distributed Description Logics) [4], Package-based Description Logics [3] and design a distributed algorithm for reasoning. The second option consists in defining new formalisms which allow reasoning with multiple domains in a distributed way. The new semantics of these formalisms reconcile conflicts between ontologies, but they do not adequately formalize the quite common case of ontologies related with ontology alignments produced by third party ontology matchers. Indeed, these formalisms assert cross-ontology correspondences (bridge rules, links or imports) from one ontology's point of view, while often, such correspondences are expressed from a point of view that encompasses both aligned ontologies. Another issue of these non-standard semantics is that reasoners such as Drago [9], Pellet [10], an early version of Draon [11] using the distributed algorithms resulting from the corresponding semantics require an exponential number of message exchanges over network. This exponential blow-up results from exchanging model portions (the so-called distributed tableau) between modules of the reasoner located on different sites.

Recenty, Atencia and Gmati [7,8] have proposed a tableau algorithm for reasoning in the centralized context on an $\mathcal{ALC}$ ontology with link keys. They have showed that adding link keys to $\mathcal{ALC}$ does not augment the complexity of the tableau algorithm.

3 Preliminaries

The syntax and semantics of the logic $\mathcal{ALC}$ are defined below.

Definition 1 (Syntax of $\mathcal{ALC}$). *Let* $\mathbf{C}$*,* $\mathbf{R}$ *and* $\mathbf{I}$ *be non-empty sets of* concept names, role names *and* individuals*, respectively. The set of* $\mathcal{ALC}$-concepts *(or simply concepts) is the smallest set such that*

- *every concept name in* $\mathbf{C}$*,* $\top$ *and* $\bot$ *are concepts, and*
- *if* C, D *are concepts and* R *is a role name in* $\mathbf{R}$ *then* $C \sqcap D$*,* $C \sqcup D$*,* $\neg C$*,* $\forall R.C$ *and* $\exists R.C$ *are concepts.*

A general concept inclusion *(GCI) is an expression of the form* $C \sqsubseteq D$ *where* C, D *are concepts. A* terminology *or* TBox *is a finite set of GCIs.*

An ABox assertion *is an expression of the form* $C(a)$*,* $R(a, b)$*,* $a \approx b$ *or* $a \not\approx b$ *where* C *is a concept,* R *is a role name in* $\mathbf{R}$ *and* a, b *are individuals in* $\mathbf{I}$*. An* ABox *is a finite set of ABox assertions. A pair* $O = (\mathcal{A}, \mathcal{T})$*, where* $\mathcal{T}$ *is a TBox and* $\mathcal{A}$ *is an ABox, is called an* $\mathcal{ALC}$ ontology*. We use* $\mathsf{Voc}_I(O)$*,* $\mathsf{Voc}_C(O)$ *and* $\mathsf{Voc}_R(O)$ *to denote the sets of individuals, concept names and role names occurring in* O*.*

Definition 2 (Semantics of $\mathcal{ALC}$). *An* interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *is composed of a non-empty set* $\Delta^{\mathcal{I}}$*, called the* domain *of* $\mathcal{I}$*, and a* valuation $\cdot^{\mathcal{I}}$ *which maps every concept name to a subset of* $\Delta^{\mathcal{I}}$*, every role name to a subset of* $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ *and each individual to an element of* $\Delta^{\mathcal{I}}$*. The valuation is extended to constructed concepts such that, for all concepts* C, D *and role name* R*, the following is satisfied:*

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}, \bot^{\mathcal{I}} = \emptyset$$
$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}, (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$
$$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$$
$$(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$$
$$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$$

An interpretation $\mathcal{I}$ satisfies *a* GCI $C \sqsubseteq D$*, denoted by* $\mathcal{I} \models C \sqsubseteq D$*, if* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$*.* $\mathcal{I}$ *is a* model *of a* TBox $\mathcal{T}$ *if* $\mathcal{I}$ *satisfies every GCI in* $\mathcal{T}$*.*

An interpretation $\mathcal{I}$ *satisfies the ABox assertions*

$$C(a) \text{ if } a^{\mathcal{I}} \in C^{\mathcal{I}}$$
$$R(a, b) \text{ if } \langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$$
$$a \approx b \text{ if } a^{\mathcal{I}} = b^{\mathcal{I}}$$
$$a \not\approx b \text{ if } a^{\mathcal{I}} \neq b^{\mathcal{I}}$$

Given an ABox assertion α*,* $\mathcal{I} \models \alpha$ *denotes that* $\mathcal{I}$ *satisfies* α*.* $\mathcal{I}$ *is a model of an ABox* $\mathcal{A}$ *if it satisfies every ABox assertion in* $\mathcal{A}$*.*

An interpretation $\mathcal{I}$ *is a model of an* $\mathcal{ALC}$ *ontology* $O = (\mathcal{A}, \mathcal{T})$ *if* $\mathcal{I}$ *is a model of* $\mathcal{T}$ *and* $\mathcal{A}$*. An ontology* O *is* consistent *if there exists a model of* O*. An ontology* O entails *a GCI, an ABox assertion, written* $O \models \alpha$*, if every model of* O *satisfies* α*.*

We need notations and definitions that will be used in the paper. We use $|S|$ to denote the cardinality of a set S. Given an $\mathcal{ALC}$ ontology $\mathcal{O} = \langle \mathcal{A}, \mathcal{T}\rangle$, we denote by $\mathsf{sub}(\mathcal{O}) = \mathsf{sub}(\mathcal{A}, \mathcal{T})$ the set of all sub-concepts occurring in $\mathcal{A}, \mathcal{T}$. The size of an ontology $\mathcal{O}$ is denoted by $|\mathcal{O}| = |\mathcal{A}| + |\mathcal{T}|$ where $|\mathcal{A}|$ is the size of all assertions, $|\mathcal{T}|$ the size of all GCIs. It holds that $|\mathsf{sub}(\mathcal{O})|$ is polynomially bounded by $|\mathcal{O}|$ since if a concept is represented as a string then a sub-concept is a substring.

To be able to define a network of aligned ontologies, we need alignments which represent semantic links between ontology entities such as individuals, concepts or roles.

Definition 3 (network of aligned ontologies). *An* $\mathcal{ALC}$ *network of aligned ontologies is a tuple* $\langle \{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n \rangle$ *where* O_i *is an* $\mathcal{ALC}$ *ontology with* $1 \leq i \leq n$*, and* A_{ij} *with* $1 \leq i < j \leq n$ *is an alignment containing correspondences of the following forms:*

- $C \rightarrow D$ *or* $C \leftarrow D$ *where* $C \in \mathsf{sub}(O_i)$ *and* $D \in \mathsf{sub}(O_j)$*. Such a correspondence is called concept correspondence.*

– *$a \approx b$ ($a \not\approx b$) where $a \in \mathsf{Voc}_I(O_i)$ and $b \in \mathsf{Voc}_I(O_j)$. Such a correspondence is called individual correspondence.*
– *a link key $\{\langle P_k, Q_k\rangle\}_{k=1}^n \mathsf{linkkey}\langle C, D\rangle$ where $P_k \in \mathsf{Voc}_R(O_i)$, $Q_k \in \mathsf{Voc}_R(O_j)$ for $1 \leq k \leq n$ and $C \in \mathsf{sub}(O_i)$ and $D \in \mathsf{sub}(O_j)$. Such a correspondence is called link key correspondence.*

Note that when we write $C \rightarrow D \in A_{ij}$ this means $C \in \mathsf{Voc}_C(O_i)$ and $D \in \mathsf{Voc}_C(O_j)$. Analogously, $C \leftarrow D \in A_{ij}$ implies $C \in \mathsf{Voc}_C(O_i)$ and $D \in \mathsf{Voc}_C(O_j)$. Semantically, $C \rightarrow D$ is different from $C \leftarrow D$. The following definition formalizes the semantics of correspondences in an alignment so that it is compatible with that of ontologies. We retain the standard semantics for individual and link key correspondences while the semantics of concept correspondences is weakened.

Definition 4 (semantics of alignments). *An $\mathcal{ALC}$ network of aligned ontologies is a tuple $\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ where O_i is an $\mathcal{ALC}$ ontologies with $1 \leq i \leq n$, and A_{ij} is an alignment with $1 \leq i < j \leq n$. Let $\mathcal{I}$ and $\mathcal{J}$ be models of O_i and O_j respectively.*

– *If $C \rightarrow D$ is in A_{ij} then $D^{\mathcal{J}} = \emptyset$ implies $C^{\mathcal{I}} = \emptyset$.*
– *If $a \approx b$ is in A_{ij} then $a^{\mathcal{I}} = a^{\mathcal{J}}$.*
– *If $a \not\approx b$ is in A_{ij} then $a^{\mathcal{I}} \neq a^{\mathcal{J}}$.*
– *If $\{\langle P_k, Q_k\rangle\}_{k=1}^n \mathsf{linkkey}\langle C, D\rangle$ is in A_{ij} then $(a_k^i)^{\mathcal{I}} = (a_k^j)^{\mathcal{J}}$, $\langle a^{\mathcal{I}}, (a_k^i)^{\mathcal{I}}\rangle \in P_k^{\mathcal{I}}$, $\langle b^{\mathcal{J}}, (a_k^j)^{\mathcal{J}}\rangle \in Q_k^{\mathcal{J}}$ for all $1 \leq k \leq n$, $a^{\mathcal{I}} \in C^{\mathcal{I}}$, $b^{\mathcal{J}} \in D^{\mathcal{J}}$ imply $a^{\mathcal{I}} = b^{\mathcal{J}}$.*

The notion of consistency for a network of aligned ontologies can be naturally introduced thanks to the semantics of ontologies and alignments involved in the network.

Definition 5 (network consistency). *Let $\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ be a network of aligned ontologies in $\mathcal{ALC}$. The network is consistent if there is a model $\mathcal{I} = \{\mathcal{I}_i\}_{i=1}^n$ where $\mathcal{I}_i = \langle \Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i}\rangle$ is a model of O_i for all $1 \leq i \leq n$ such that*

1. *For each correspondence $a \approx b$ in A_{ij} with $1 \leq i < j \leq n$, $a^{\mathcal{I}_i} = b^{\mathcal{I}_j}$. For each correspondence $a \not\approx b$ in A_{ij} with $1 \leq i < j \leq n$, $a^{\mathcal{I}_i} \neq b^{\mathcal{I}_j}$.*
2. *For each correspondence $C \rightarrow D$ in A_{ij} with $1 \leq i < j \leq n$, if $D^{\mathcal{I}_j} = \emptyset$ then $C^{\mathcal{I}_i} = \emptyset$.*
3. *For each correspondence $\{\langle P_k, Q_k\rangle\}_{k=1}^n \mathsf{linkkey}\langle C, D\rangle$ in A_{ij} with $1 \leq i < j \leq n$, if $(a_k^i)^{\mathcal{I}_i} = (a_k^j)^{\mathcal{I}_j}$, $\langle a^{\mathcal{I}_i}, (a_k^i)^{\mathcal{I}_i}\rangle \in P_k^{\mathcal{I}_i}$, $\langle b^{\mathcal{I}_j}, (a_k^j)^{\mathcal{I}_j}\rangle \in Q_k^{\mathcal{I}_j}$ for all $1 \leq k \leq n$, $a^{\mathcal{I}_i} \in C^{\mathcal{I}_i}$, $b^{\mathcal{I}_j} \in D^{\mathcal{I}_j}$ then $a^{\mathcal{I}_i} = b^{\mathcal{I}_j}$.*

A network $N = \langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ entails *a link key α, written $N \models \alpha$, if every model of N satisfies α. In particular, an alignment A_{ij} is called clash-free if $\{a \approx b, a \not\approx b\} \not\subseteq A_{ij}$.*

We finish this section by proving the following lemma which allows to reduce link key entailment to consistency of the network of aligned ontologies.

Lemma 1 (Reduction of link key entailment to consistency). *Let $\langle\{O_1, O_2\}, A_{12}\rangle$ be a network of aligned ontologies in $\mathcal{ALC}$. It holds that*

$$\langle\{O_1, O_2\}, A_{12}\rangle \models (\{\langle P_i, Q_i\rangle\}_{i=1}^{m} \ \mathsf{linkkey}\ \langle C, D\rangle) \text{ iff } \\ \langle\{O'_1, O'_2\}, A'_{12}\rangle \text{ is inconsistent}$$

with $O'_1 = O_1 \cup \{C(x)\} \cup \{P_i(x, z_i)\}_{i=1}^{n}$, $O'_2 = O_2 \cup \{D(y)\} \cup \{Q_i(y, z'_i)\}_{i=1}^{n}$, $A'_{12} = A_{12} \cup \{z_i \approx z'_i\}_{i=1}^{n} \cup \{x \not\approx y\}$, $x, z_1, \cdots, z_n$ are new individuals in O_1 and $y, z'_1, \cdots, z'_n$ are new individuals in O_2.

Proof. Let $\lambda = \{\langle P_i, Q_i\rangle\}_{i=1}^{n}\ \mathsf{linkkey}\ \langle C, D\rangle$. Assume that $\langle\{O_1, O_2\}, A_{12}\rangle \models \lambda$. We show that $\langle\{O'_1, O'_2\}, A'_{12}\rangle$ is inconsistent. By contradiction, assume that $\langle\{O'_1, O'_2\}, A'_{12}\rangle$ has a model $\mathcal{I} = \langle\mathcal{I}_1, \mathcal{I}_2\rangle$, *i.e.* O'_1 and O'_2 have models $\mathcal{I}_1$ and $\mathcal{I}_2$ satisfying Definition 5. This implies that $\mathcal{I}_1$ and $\mathcal{I}_2$ are models of O_1 and O_2. Hence, $x^{\mathcal{I}_1} \in C^{\mathcal{I}_1}$, $y^{\mathcal{I}_2} \in D^{\mathcal{I}_2}$, $\langle x^{\mathcal{I}_1}, z_i^{\mathcal{I}_1}\rangle \in P_i^{\mathcal{I}_1}$, $\langle y^{\mathcal{I}_2}, {z'}_i^{\mathcal{I}_2}\rangle \in Q_i^{\mathcal{I}_2}$,

---

Algorithm 1. Propagating individual equalities

---

function PROPAGATEEQUAL(O_i, O_j, A_{ij})
  while A_{ij} or O_i or O_j is unstationary **do**
    if O_i or O_j is inconsistent or A_{ij} is not clash-free **then**
      return false
    end if
    for $a_i^1 \approx a_j^1 \in A_{ij}$, $a_i^2 \approx a_j^2 \in A_{ij}$ **do**
      for $O_k \models a_k^m \approx a_k^h$, $k \in \{i, j\}$, $m, h \in \{1, 2\}$, $m \neq h$ **do**
        $A_{ij} \leftarrow A_{ij} \cup \{a_i^h \approx a_j^m, a_i^m \approx a_j^h\}$
        $O_k \leftarrow O_k \cup \{a_k^1 \approx a_k^2\}$
      end for
    end for
    for each $\{\langle P_k, Q_k\rangle\}_{k=1}^{n}\mathsf{linkkey}\langle C, D\rangle \in A_{ij}$ **do**
      for $a_k^i \approx a_k^j \in A_{ij}$, $a \in \mathsf{Voc}_I(O_i)$, $b \in \mathsf{Voc}_I(O_j)$, $P_k(a', {a'}_k^i) \in O_i$,
       $Q_k(b', {a'}_k^j) \in O_j$, $O_i \models a \approx a'$, $O_i \models a_k^i \approx {a'}_k^i$, $O_j \models b \approx b'$
       $O_j \models a_k^j \approx {a'}_k^j$ for all $1 \leq k \leq n$ **do**
        if $O_i \cap \{C(a), \neg C(a)\} = \emptyset$ **then**
          $O_i \leftarrow O_i \cup \{(C \sqcup \neg C)(a)\}$
        end if
        if $O_j \cap \{D(b), \neg D(b)\} = \emptyset$ **then**
          $O_j \leftarrow O_j \cup \{(D \sqcup \neg D)(b)\}$
        end if
      end for
      for $a_k^i \approx a_k^j \in A_{ij}$, $O_i \models C(a)$, $O_j \models D(b)$, $P_k(a', {a'}_k^i) \in O_i$,
       $Q_k(b', {a'}_k^j) \in O_j$, $O_i \models a \approx a'$, $O_i \models a_k^i \approx {a'}_k^i$, $O_j \models b \approx b'$
       $O_j \models a_k^j \approx {a'}_k^j$ for all $1 \leq k \leq n$ **do**
        $A_{ij} \leftarrow A_{ij} \cup \{a \approx b\}$
      end for
    end for
  end while
  return true
end function

Algorithm 2. Propagating concept unsatisfiability

function PROPAGATEUNSAT(O_i, O_j, A_{ij})
  while A_{ij} or O_i or O_j is unstationary **do**
    if O_i or O_j is inconsistent or A_{ij} is not clash-free **then**
      return false
    end if
    for each $D \rightarrow C \in A_{ij}$ **do**
      if $O_j \models C \sqsubseteq \bot$ **then**
        $O_i \leftarrow O_i \cup \{D \sqsubseteq \bot\}$
      end if
    end for
    for each $D \leftarrow C \in A_{ij}$ **do**
      if $O_i \models D \sqsubseteq \bot$ **then**
        $O_j \leftarrow O_j \cup \{C \sqsubseteq \bot\}$
      end if
    end for
    for $C_i^1 \rightarrow C_j^1 \in A_{ij}, C_i^2 \leftarrow C_j^2 \in A_{ij}$ **do**
      for $O_j \models C_j^1 \sqsubseteq C_j^2$ **do**
        $A_{ij} \leftarrow A_{ij} \cup \{C_i^1 \rightarrow C_j^2, C_i^2 \leftarrow C_j^1\}$
      end for
    end for
    for $C_i^1 \leftarrow C_j^1 \in A_{ij}, C_i^2 \rightarrow C_j^2 \in A_{ij}$ **do**
      for $O_i \models C_i^1 \sqsubseteq C_i^2$ **do**
        $A_{ij} \leftarrow A_{ij} \cup \{C_i^1 \rightarrow C_j^2, C_i^2 \leftarrow C_j^1\}$
      end for
    end for
  end while
  return true
end function

$z_i^{\mathcal{I}_1} = z'^{\mathcal{I}_2}_i$ and $x^{\mathcal{I}_1} \neq y^{\mathcal{I}_2}$. This implies that $\mathcal{I} \not\models \lambda$. Thus, we have a model $\mathcal{I}$ of $\langle\{O'_1, O'_2\}, A'_{12}\rangle$ such that $\mathcal{I} \not\models \lambda$. Therefore, $\langle\{O_1, O_2\}, A_{12}\rangle \not\models \lambda$, which contradicts the assumption.

Assume now that $\langle\{O_1, O_2\}, A_{12}\rangle \not\models \lambda$. Let us show that $\langle\{O'_1, O'_2\}, A'_{12}\rangle$ is consistent. Since $\langle\{O'_1, O'_2\}, A'_{12}\rangle \not\models \lambda$, then there exists an interpretation $\mathcal{I} = \langle\mathcal{I}_1, \mathcal{I}_2\rangle$ such that $\mathcal{I} \models \langle\{O'_1, O'_2\}, A'_{12}\rangle$ and $\mathcal{I} \not\models \lambda$.

Since $\mathcal{I} \not\models \lambda$, by the semantics of link keys, there exist $\delta, \delta_1, \ldots, \delta_n \in \Delta_1^{\mathcal{I}_1}$ and $\delta', \delta'_1, \ldots, \delta'_n \in \Delta_2^{\mathcal{I}_2}$ such that $\delta \in C^{\mathcal{I}_1}$, $\delta' \in D^{\mathcal{I}_2}$, $(\delta, \delta_1) \in P_1^{\mathcal{I}_1}, (\delta', \delta'_1) \in Q_1^{\mathcal{I}_2}, \ldots, (\delta, \delta_n) \in P_n^{\mathcal{I}_1}, (\delta', \delta_n) \in Q_n^{\mathcal{I}_2}$, $\delta_1 = \delta'_1, \ldots, \delta_n = \delta'_n$ and $\delta \neq \delta'$. Let us extend $\mathcal{I}$ by defining $x^{\mathcal{I}_1} = \delta$, $y^{\mathcal{I}_2} = \delta'$, $z_1^{\mathcal{I}_1} = \delta_1, \ldots, z_n^{\mathcal{I}_1} = \delta_n$, $z'^{\mathcal{I}_2}_1 = \delta'_1, \ldots, z'^{\mathcal{I}_2}_n = \delta'_n$. Then, $\mathcal{I}$ is a model of $\langle\{O'_1, O'_2\}, A'_{12}\rangle$. Therefore, $\langle\{O'_1, O'_2\}, A'_{12}\rangle$ is consistent. $\square$

This lemma can be extended to a general network of aligned ontologies containing more than two ontologies.

4 An Algorithm for a Network of Aligned Ontologies

The algorithm for deciding consistency of a network of aligned ontologies deals with pair by pair of ontologies in the network. For each pair of ontologies and an alignment between them, the algorithm repeats the following three tasks: propagating individual equalities from one ontology to the other via individual correspondences; applying link key correspondences which may lead to adding new individual correspondences; and propagating concept unsatisfiabilities from one ontology to the other via concept correspondences. The execution of a task may trigger the execution of another task. The execution of these tasks may lead to a change of ontologies and alignments in the network. The algorithm terminates on the pair of ontologies when the ontologies and the alignment reach stationarity. The first and second tasks are described in Algorithm 1 while the third one is outlined in Algorithm 2.

The following lemma establishes that the propagation performed by Algorithms 1 and 2 and consistency of the pair of the extended ontologies suffice to decide consistency of the network composed of the initial ontologies and the alignment.

Lemma 2 (reduction for a pair). *Let O_1, O_2 be two consistent ontologies and A_{12} be an alignment. We use $\widehat{O_1}$, $\widehat{O_2}$ and $\widehat{A_{12}}$ to denote the resulting ontologies and alignment obtained by calling* $\mathsf{propagatePair}(O_1, O_2, A_{12})$*. It holds that $\widehat{O_1}, \widehat{O_2}$ are consistent and $\widehat{A_{12}}$ is clash-free iff the network $\langle\{O_1, O_2\}, \{A_{12}\}\rangle$ is consistent.*

Before providing a complete proof of the lemma, we summarize the main arguments. The soundness of the if-direction of Lemma 2 is straightforward since Algorithms 1 and 2 add only logical consequences of the network to the ontologies and alignments. The soundness of the only-if-direction of the lemma is based on the following elements: (i) consistency of the extended ontologies and clash-freeness of the extended alignments imply consistency of the initial ontologies and clash-freeness of the initial alignments; (ii) Algorithms 1 and 2 make explicit all individual equalities, and thus eventual clashes of the kind $a \approx b, a \not\approx b$ must be discovered. This ensures that two models of the extended ontologies satisfy individual correspondences; (iii) Algorithms 1 and 2 apply link keys until they are not applicable over the initial individuals in the ontologies. Since models of an $\mathcal{ALC}$ ontology are tree-shaped and $\mathcal{ALC}$ does not allow for inverse roles, satisfaction of the link keys over the initial individuals is sufficient; and (iv) Algorithms 1 and 2 propagate concept unsatisfiabilities. If the "subsumer" of a concept correspondence is satisfiable then a model of the ontology can be extended such that the interpretation of the subsumer in this model is not empty. This implies that the concept correspondence is satisfied.

Proof. "If-direction". Assume that the network $\langle\{O_1, O_2\}, \{A_{12}\}\rangle$ is consistent. By definition, O_i has a model $\mathcal{I}_i$ with $1 \leq i \leq 2$ such that they satisfy all correspondences $\alpha \in A_{12}$. We show that $\mathcal{I}_1$ is a model of $\widehat{O_1}$. For this, we have to prove that:

– $a_0^{\mathcal{I}_1} = a_n^{\mathcal{I}_1}$ if $a_0 \approx a_n$ is added to O_1 by Line 9 in Algorithm 1. We have $a_0 \approx a_n$ is added to O_1 if Algorithm 1 discovers a sequence of equalities $a_0 \approx a_1, \cdots, a_{n-1} \approx a_n$ such that $a_i \approx a_{i+1} \in \widehat{O_1} \cup \widehat{O_2} \cup \widehat{A_{12}}$ for $0 \leq i \leq n-1$. This sequence of equalities implies $a_0^{\mathcal{I}_1} = a_n^{\mathcal{I}_1}$. Note that if there is some $a \approx b \in A_{ij}$ then $\Delta^{\mathcal{I}_1} \cap \Delta^{\mathcal{I}_2} \neq \emptyset$ according to Definition 4. By using the same argument, we can show $a_0^{\mathcal{I}_2} = a_n^{\mathcal{I}_2}$ if $a_0 \approx a_n$ is added to O_2 by Line 9 in Algorithm 1.
– $C_0^{\mathcal{I}_1} = \emptyset$ if $C_0 \sqsubseteq \bot$ is added to O_1 by Line 8 in Algorithm 2. We have $C_0 \sqsubseteq \bot$ is added to O_1 if Algorithm 1 discovers a sequence $C_0 \Rightarrow C_1, \cdots, C_{n-1} \Rightarrow C_n$ such that $\widehat{O_1} \models C_i \Rightarrow C_{i+1}$ or $\widehat{O_2} \models C_i \Rightarrow C_{i+1}$ or $C_i \Rightarrow C_{i+1} \in \widehat{A_{12}}$ for $0 \leq i \leq n-1$, and $\widehat{O_i} \models C_n^{\mathcal{I}_i} \sqsubseteq \bot$ $(i \in \{1,2\})$ where "$\Rightarrow$" represents "$\rightarrow$" or "$\sqsubseteq$" and $C \leftarrow D = D \Rightarrow C$, $C \sqsupseteq D = D \Rightarrow C$. This implies $C_i^{\mathcal{I}_i} = \emptyset$ for $1 \leq i \leq n$. By using the same argument, we can show $C_0^{\mathcal{I}_2} = \emptyset$ if $C_0 \sqsubseteq \bot$ is added to O_2 by Line 13 in Algorithm 2.
– The concepts $(C \sqcup {\sim}C)(a)$ and $(D \sqcup {\sim}D)(b)$ added by Lines 17 and 20 in Algorithm 2 do not change consistency since they are tautologies.

"Only-If-direction". Since $\widehat{O_i}$ is consistent, according to [12], $\widehat{O_i}$ has a tree-shaped model $\mathcal{I}_i$ where each interpretation domain Δ_i of $\mathcal{I}_i$ is composed of a set of initial individuals I^i_{old} and a set of new individuals I^i_{new} for $1 \leq i \leq 2$. Since $O_i \subseteq \widehat{O_i}$, $\mathcal{I}_i$ is a model of O_i with $1 \leq i \leq 2$. We will extend $\mathcal{I}_1$ and $\mathcal{I}_2$ so that they satisfy the correspondences in A_{12}.

– If $a \approx b \in \widehat{A_{12}}$ then $a^{\mathcal{I}} = a^{\mathcal{J}}$ for all models $\mathcal{I}$ and $\mathcal{J}$ of O_1 and O_2 respectively due to Definition 4. We define $a^{\mathcal{I}_1} = a^{\mathcal{I}_2}$. Thus, $a^{\mathcal{I}_1} = a^{\mathcal{I}_2}$ for each $a \approx b \in A_{12}$ since $A_{12} \subseteq \widehat{A_{12}}$. By construction, $\mathcal{I}_1$ and $\mathcal{I}_2$ satisfy all of the individual correspondences in A_{12} according to Definition 4.
– If $a \not\approx b \in \widehat{A_{12}}$ then $a \approx b \notin \widehat{A_{12}}$ since $\widehat{A_{12}}$ is clash-free.
– Let $C_h \rightarrow D_h \in A_{12}$. If $\widehat{O_2} \models D_h \sqsubseteq \bot$ then $C_h \sqsubseteq \bot$ is added to $\widehat{O_1}$ by Algorithm 2. Hence, $D_h^{\mathcal{I}_2} = \emptyset$ implies $C_h^{\mathcal{I}_1} = \emptyset$. Note that if $\widehat{O_2} \models D_h \sqsubseteq \bot$ then $\widehat{O'_2} \models D_h \sqsubseteq \bot$ for all $\widehat{O_2} \subseteq \widehat{O'_2}$.
Assume that $\widehat{O_2} \not\models D_h \sqsubseteq \bot$. Thus, $\widehat{O_2} \cup \{D_h(x_h)\}$ is consistent where x_h is a new individual. According to [12], $\widehat{O_2} \cup \{D_h(x_h)\}$ has a tree-shaped model $\mathcal{I}'_2$ of $\widehat{O_2} \cup \{D_h(x_h)\}$. We show that if $\widehat{O_2} \cup \{D_1(x_1)\}$ and $\widehat{O_2} \cup \{D_2(x_2)\}$ are consistent with new individual x_1, x_2 then $\widehat{O_2} \cup \{D_1(x_1), D_2(x_2)\}$ is consistent. Indeed, running the standard tableau algorithm in [12] on $\widehat{O_2} \cup \{D_1(x_1)\}$ can build a set $\mathbf{T}$ of completion trees rooted at the initial individuals in $\widehat{O_2}$ and a completion tree T_{x_1} rooted at x_1. Analogously, if the standard tableau algorithm runs on $\widehat{O_2} \cup \{D_2(x_2)\}$, it can build a set $\mathbf{T}$' of completion trees rooted at the initial individuals in $\widehat{O_2}$ and a completion tree T_{x_2} rooted at x_2. All trees are clash-free and complete. Hence, $\mathbf{T} \cup \{T_{x_1}, T_{x_2}\}$ can represent a model of $\widehat{O_2} \cup \{D_1(x_1), D_2(x_2)\}$.
Therefore, we can run the standard tableau algorithm in [12] on $\widehat{O_2} \cup \{D_i(x_i)\}_{i=1}^m$ to obtain a tree-shaped model $\mathcal{J}_2$ of $\widehat{O_2} \cup \{D_i(x_i)\}_{i=1}^m$ where

x_h is a new individual and $\widehat{O_2} \not\models D_h \sqsubseteq \bot$ for $1 \leq h \leq m$.
By using the same argument, we can obtain a tree-shaped model $\mathcal{J}_1$ of $\widehat{O_1} \cup \{D'_1(x'_1), \cdots, D'_{m'}(x'_{m'})\}$. By construction, $\mathcal{J}_1$ and $\mathcal{J}_2$ satisfy all of the concept correspondences in A_{12} according to Definition 4. In addition, they remain to satisfy all of the individual correspondences in A_{12}. For the sake of the simplicity, we rename $\mathcal{J}_1$ and $\mathcal{J}_2$ to $\mathcal{I}_1$ and $\mathcal{I}_2$.

– Assume that $\{\langle P_k, Q_k\rangle\}_{k=1}^{n}\mathsf{linkkey}\langle C, D\rangle$ is a link key in A_{12} and $(a_k^1)^{\mathcal{I}_1} = (a_k^2)^{\mathcal{I}_2}$, $\langle a^{\mathcal{I}_1}, (a_k^1)^{\mathcal{I}_1}\rangle \in P_k^{\mathcal{I}_1}$, $\langle b^{\mathcal{I}_2}, (a_k^2)^{\mathcal{I}_2}\rangle \in Q_k^{\mathcal{I}_2}$ for all $1 \leq k \leq n$, $a^{\mathcal{I}_1} \in C^{\mathcal{I}_1}$, $b^{\mathcal{I}_2} \in D^{\mathcal{I}_2}$.
  1. If $(a_k^1)^{\mathcal{I}_1} = (a_k^2)^{\mathcal{I}_2}$ then there is a sequence $a_0 \approx a_1, \cdots, a_{m-1} \approx a_m$ (discovered by Algorithm 1) such that $a_i \approx a_{i+1} \in \widehat{O_1} \cup \widehat{O_2} \cup \widehat{A_{12}}$ for $0 \leq i \leq m-1$ with $a_k^1 = a_0, a_k^2 = a_m$. This implies that $a_k^1 \approx a_k^2 \in \widehat{A_{12}}$ for $1 \leq k \leq n$.
  2. Since $\mathcal{I}_1$ and $\mathcal{I}_2$ are tree-shaped whose roots are the old individuals, the condition of the link key holds only if all individuals a_k^1, a_k^2 for $1 \leq k \leq n$, and a, b are contained $I_{old}^1 \cup I_{old}^2$. Hence, $\langle a^{\mathcal{I}_1}, (a_k^1)^{\mathcal{I}_1}\rangle \in P_k^{\mathcal{I}_1}$ iff $P_k(a', {a'}_k^1) \in O_i$ with $O_i \models a \approx a'$, $O_i \models a_k^1 \approx {a'}_k^1$ for $1 \leq k \leq n$ where a, a' and $a_k^1, {a'}_k^1$ are old individuals. Analogously, $\langle b^{\mathcal{I}_2}, (a_k^2)^{\mathcal{I}_2}\rangle \in Q_k^{\mathcal{I}_2}$ iff $Q_k(b', {a'}_k^2) \in O_j$ with $O_j \models b \approx b'$, $O_j \models a_k^2 \approx {a'}_k^2$ for $1 \leq k \leq n$ where b, b' and $a_k^2, {a'}_k^2$ are old individuals.
  3. Since $a^{\mathcal{I}_1} \in C^{\mathcal{I}_1}$ and $(C \sqcup \neg C)(a) \in O_1$ (Line 17, Algorithm 1), we have $\widehat{O_1} \models C(a)$. Analogously, from $b^{\mathcal{I}_2} \in D^{\mathcal{I}_2}$ and $(D \sqcup \neg D)(b) \in O_2$ (Line 20, Algorithm 1), we obtain $\widehat{O_2} \models D(b)$.

  Therefore, the 3 items above trigger Line 26 in Algorithm 1 which adds to $\widehat{A_{12}}$ the assertion $a \approx b$. Thus, we obtain $a^{\mathcal{I}_1} \approx b^{\mathcal{I}_2}$.

Algorithm 3. Complete propagation over the whole network

```
function PROPAGATEOVERNETWORK(⟨{O_i}_{i=1}^n, {A_ij}_{i,j=1,i≠j}^n⟩)
    while O_i, O_j, A_ij are unstationary for all 1 ≤ i < j ≤ n do
        for 1 ≤ i < j ≤ n do
            while O_i, O_j, A_ij are unstationary do
                if propagateEqual(O_i, O_j, A_ij) returns false then
                    return false
                end if
                if propagateUnsat(O_i, O_j, A_ij) returns false then
                    return false
                end if
            end while
        end for
    end while
    return true
end function
```

We have proven that $\mathcal{I}_1$ and $\mathcal{I}_2$ are models of O_1 and O_2 which satisfy all of the correspondences in A_{12}. □

We can observe that Algorithms 1 and 2 can be implemented in a decentralized manner since each call for checking ontology entailment or consistency can be sent to a local reasoner associated with the ontology located on a different site.

To check consistency of a network of aligned ontologies, it is needed to run Algorithms 1 and 2 on each pair of ontologies with the alignment between them until all ontologies and alignments are stationary. Note that saturating a pair of ontologies with the alignment can make a saturated pair of ontologies unsaturated. This is due to the fact that an ontology can be shared by several pairs of ontologies.

The following theorem is a consequence of Lemma 2.

Theorem 1 (reduction for network). *Let $\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ be a network of aligned ontologies. We use $\widehat{O_i}$ and $\widehat{A_{ij}}$ to denote the resulting ontologies and alignments obtained by calling* $\mathsf{propagateOverNetwork}(\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle)$*. It holds that $\widehat{O_i}$ is consistent for all $1 \leq i \leq n$ and $\widehat{A_{ij}}$ is clash-free for all $1 \leq i < j \leq n$ iff the network $\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ is consistent.*

We now investigate the complexity of the algorithms. Under the hypothesis in which a call to reasoners associated with ontologies is considered as an oracle, *i.e.* an elementary operation, our algorithms are tractable.

Theorem 2. *Let $\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ be a network of aligned ontologies. The algorithm* $\mathsf{propagateOverNetwork}(\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle)$ *runs in polynomial time in the size of the network if each check of entailment or consistency occurring in the algorithms is considered as an oracle.*

Proof. The complexity of the algorithm propagateOverNetwork depends on the complexity of propagateEqual, propagateUnsat. When running these algorithms, each ontology is monotonically extended. It is straightforward to obtain that the number of axioms of the form $C \sqsubseteq \bot$ added to ontologies O_i and O_j is bounded by a polynomial function in the size of initial alignments since C must occur in initial correspondences. Analogously, the number of individuals correspondences $a \approx b$ added to alignments A_{ij} is bounded by a polynomial function in the size of initial alignments since a, b must occur in initial correspondences. This implies that the number of iterations of the *while* loops in Algorithms 1, 2 and 3 is bounded by a polynomial function in the size of initial alignments.

In addition, the number of iterations of the *for* loops in Algorithms 1, 2 and 3 is bounded by a polynomial function in the size of initial alignments, the size of ontologies and the number of ontologies and alignments included in the network. This observation completes the proof. □

5 Examples

This section provides some examples for showing the difference of the standard semantics from the new one in terms of reasoning and how to use the algorithms presented in Sect. 4.

Example 2. The ontologies and alignment in Example 1 can be rewritten as follows :

$$O_1 = \{DP \sqsubseteq P, DP(a)\}, O_2 = \{PS \sqsubseteq R, R \sqsubseteq \neg D\},$$
$$A_{12} = \{DP \rightarrow R, DP \rightarrow D, \langle pr, re\rangle \mathsf{linkkey}\langle P, R\rangle\}$$

If the correspondences are considered as standard subsumptions then the ontology $O_1 \cup O_2 \cup A_{12}$ is inconsistent. Indeed, assume that there is a model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$ of the ontology. This implies that $a^{\mathcal{I}} \in DP^{\mathcal{I}}$, $DP^{\mathcal{I}} \subseteq R^{\mathcal{I}}$ and $DP^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Thus, $a^{\mathcal{I}} \in R^{\mathcal{I}} \cap D^{\mathcal{I}}$. However, we have $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \setminus D^{\mathcal{I}}$, which is a contradiction.

If we now interpret the correspondences under the semantics given in Definition 4 then there is no propagation needed according to Algorithms 1 and 2. It is obvious that O_1 and O_2 are consistent, and the network $\langle\{O_1, O_2\}, A_{12}\rangle$ is consistent under the semantics given in Definition 4.

Example 3. In this example, we reduce the two correspondences in Example 2 to one as follows.

$$O_1 = \{DP \sqsubseteq P, DP(a)\}, O_2 = \{PS \sqsubseteq R, R \sqsubseteq \neg D\},$$
$$A_{12} = \{DP \rightarrow R \sqcap D, \langle pr, re\rangle \mathsf{linkkey}\langle P, R\rangle\}$$

We now interpret the correspondence under the semantics given in Definition 4. Since $O_2 \models R \sqcap D \sqsubseteq \bot$, Algorithm 2 propagates unsatisfiability of $R \sqcap D$ to O_1 via the correspondence $DP \rightarrow R \sqcap D$. Hence, it adds $DP \sqsubseteq \bot$ to O_1. This leads to inconsistency of $\widehat{O_1}$. Therefore, the network $\langle\{O_1, O_2\}, A_{12}\rangle$ is not consistent.

Example 4. The ontologies and alignment in Example 1 can be rewritten as follows :

$$O_1 = \{DP \sqsubseteq P, DP(a)\}, O_2 = \{PS \sqsubseteq R, R \sqsubseteq \neg D\},$$
$$A_{12} = \{DP \rightarrow R, DP \rightarrow D, \langle pr, re\rangle \mathsf{linkkey}\langle P, R\rangle\}$$

We consider whether $\langle\{O_1, O_2\}, A_{12}\rangle \models \lambda$ where $\lambda = \langle pr, re\rangle \mathsf{linkkey}\langle P, R\rangle$. Due to Lemma 1, we extend O_1, O_2 and A_{12} by adding to O_1 assertions $DP(x), pr(x, x_1)$, to O_2 assertions $PS(y), re(y, y_1)$, and to A_{12} assertions $x_1 \approx y_1, x \not\approx y$. Let $\widehat{O_1}, \widehat{O_2}$ and $\widehat{A_{12}}$ be the extended ontologies and alignment.

If there are models $\mathcal{I}_1$ and $\mathcal{I}_2$ of $\widehat{O_1}, \widehat{O_2}$ then, we have $x \in DP^{\mathcal{I}_1}$ and $y \in PS^{\mathcal{I}_2}$, and $DP^{\mathcal{I}_1} \subseteq P^{\mathcal{I}_1}$ and $PS^{\mathcal{I}_2} \subseteq R^{\mathcal{I}_2}$.

Thus, the link key $\langle pr, re\rangle \mathit{linkkey}\langle P, R\rangle$ is applicable, and Algorithm 1 adds $x \approx y$ to $\widehat{A_{12}}$. This leads to a clash in $\widehat{A_{12}}$ and thus the network $\langle\{\widehat{O_1}, \widehat{O_2}\}, \widehat{A_{12}}\rangle$ is not consistent. Therefore, $\langle\{O_1, O_2\}, A_{12}\rangle \models \lambda$ holds.

6 Implementation and Experimental Results

An implementation of the proposed algorithms has been integrated within a reasoner written in Java, called Draon [11], which already allowed to reason in a decentralized manner on a network of aligned ontologies under the IDDL semantics [5]. Algorithms 1, 2 and 3 can be naturally implemented such that reasoning tasks on ontologies can be independently performed by different reasoners located on different sites.

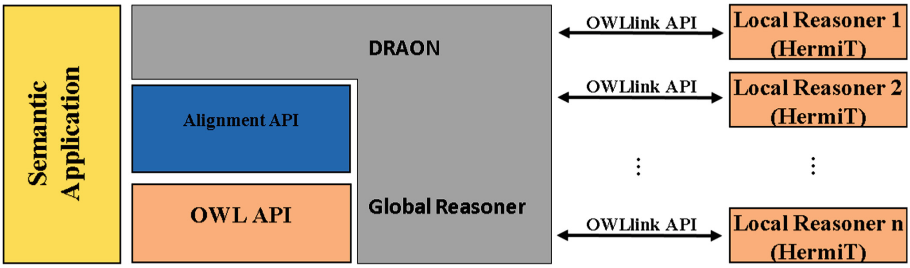

Fig. 1. Architecture of Draon

The architecture of Draon is despicted in Fig. 1. A global reasoner implements Algorithm 3. This global reasoner loads alignments and executes Algorithm 3. It propagates assertion/axioms to local reasoners located on different sites. Then it asks local reasoners to check entailment and consistency of the ontology associated with each local reasoner. The global reasoner and each local reasoner use HermiT [13] as OWL reasoner. The communication between the global reasoner and all local reasoners is based on OWLLink [14]. When connecting to a local reasoner, the global reasoner creates a Java thread which deals with the communication between them. Data shared by the threads are synchronized and protected by using semaphores. Note that we can replace HermiT with any OWL reasoner since OWLLink supports a generic OWL reasoner.

Table 1 provides information on the ontologies and alignments used for the experiments. These datasets are taken from OAEI2012[2] and OAEI2018[3] Campaigns. We have chosen small ontologies and alignments such as `iasted.owl`, `sigkdd.owl`, `iasted-sigkdd.rdf` to test our algorithm on alignments with link keys since they are well understood and manually checkable. This allows us to create manually relevant link keys (to our best knowledge, there is no system which can generate link keys expressed in the alignment syntax). In addition, we have selected large ontologies and alignment such as SNOMED, FMA, FMA-SNOMED in order that the difference between the reasoning complexities of the two semantics IDDL (implemented in Draon) and APPROX (the new semantics introduced in the paper) is more noticeable.

[2] cs.ox.ac.uk/isg/projects/SEALS/oaei/2012/.
[3] oaei.ontologymatching.org/2018/conference.

Table 1. Ontologies and alignments without link keys and their characteristics

	Concepts	Roles	Individuals	Axioms/Correspondences
Iasted	141	38	6	551
Sigkdd	50	18	5	210
iast-sigkdd (without link keys)				15
Conference	60	46	2	414
Ekaw	74	33	4	351
conference-ekaw (without link keys)				27
Cmt	30	49	3	327
Edas	104	30	117	1025
cmt-edas (without link keys)				14
FMA	10157	0	0	47467
SNOMED	13412	18	0	47104
FMA-SNOMED (without link keys)				9139
NCI	25591	87	0	135556
FMA-NCI (without link keys)				3038

We use two remote DELL servers with Intel 3.4 GHz Processor 8 cores and 32 Gb RAM on which two HermiT-based local reasoners are running. The global reasoner is also launched on a third computer with the same configuration.

We run Draon to check consistency of several networks of ontologies each of which is composed of ontologies and alignment described in Table 1. The results are put in Table 2 which shows execution times of Draon under the two different semantics IDDL and APPROX. The difference of the performances in time results from the fact that reasoning under IDDL may require in the worst case an exponential number of message exchanges between the global reasoner and the local reasoners while reasoning under APPROX needs at most a polynomial number of message exchanges.

Table 3 provides first experimental results when running Draon to check consistency of networks containing small ontologies and alignment with link keys. The alignments in this table are obtained by adding to the corresponding alignments in Table 2 some link keys manually created.

Table 2. Execution time for checking consistency of ontology networks according to different semantics

Ontology 1	Ontology 2	Alignment	IDDL	APPROX
Iasted	Sigkdd	iasted-sigkdd (without link keys)	3.5 s	9 ms
Conference	Ekaw	conference-ekaw (without link keys)	7.5 s	11 ms
Cmt	Edas	cmt-edas (without link keys)	7.5 s	16 ms
FMA	SNOMED	FMA-SNOMED (without link keys)	>15 min	81 s
FMA	NCI	FMA-NCI (without link keys)	>15 min	10 s

Table 3. Execution time (in milliseconds) for checking consistency of ontology networks with link keys

Ontology 1	Ontology 2	Alignment	Consistency in APPROX
Iasted	Sigkdd	iast-sigkdd (with link keys)	9 ms
Conference	Ekaw	conference-ekaw (with link keys)	11 ms
Cmt	Edas	cmt-edas (with link keys)	17 ms

7 Conclusion and Future Work

We have presented a new semantics of alignments which is weaker than the standard semantics. This weakened semantics of alignments allows us to express correspondences between ontologies of different nature on the one hand and to propose an efficient algorithm for reasoning on a network of ontologies with alignments containing link keys on the other hand. This new kind of correspondences is useful for establishing data links between heterogeneous datasets. The complexity of the proposed algorithm is polynomial in the size of the network if each call for checking ontology entailment or consistency is considered as an oracle. We have integrated an implementation of our algorithm within a distributed reasoner, called Draon, and reported some experimental results.

Our algorithm can be extended to deal with ontologies expressed in a more expressive Description Logic than $\mathcal{ALC}$ in condition that the new logic does not allow for inverse roles. This restriction on expressiveness prevents the current algorithm from merging individuals which are initially not in the ontology. Another extension of the current work aims to add role correspondences to alignments. This may require the algorithm to support ontologies allowing for hierarchy of roles and the negation of roles. We plan to carry out experiments of Draon on ontologies and alignments located on a large number of nodes equipped with a local reasoner. New evaluations of Draon on alignments with a large number of link keys are also expected.

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

References

1. Borgida, A., Serafini, L.: Distributed description logics: assimilating information from peer sources. J. Data Semant. (1), 153–184 (2003)
2. Grau, B.C., Parsia, B., Sirin, E.: Combining OWL ontologies using $\mathcal{E}$-connections. J. Web Semant. **4**(1), 40–59 (2006)
3. Bao, J., Caragea, D., Honavar, V.G.: A distributed tableau algorithm for package-based description logics. In: Proceedings of the ECAI Workshop on on Context Representation and Reasoning (2006)

4. Zimmermann, A., Euzenat, J.: Three semantics for distributed systems and their relations with alignment composition. In: Cruz, I., et al. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 16–29. Springer, Heidelberg (2006). https://doi.org/10.1007/11926078_2
5. Zimmermann, A., Le Duc, C.: Reasoning with a network of aligned ontologies. In: Calvanese, D., Lausen, G. (eds.) RR 2008. LNCS, vol. 5341, pp. 43–57. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88737-9_5
6. Adjiman, P., Chatalic, P., Goasdoué, F., Rousset, M., Simon, L.: Distributed reasoning in a peer-to-peer setting: application to the semantic web. J. Artif. Intell. Res. **25**, 269–314 (2006)
7. Atencia, M., David, J., Euzenat, J.: Data interlinking through robust linkkey extraction. In: Schaub, T., Friedrich, G., O'Sullivan, B., (eds.) Proceedings 21st European Conference on Artificial Intelligence (ECAI), Praha (CZ), Amsterdam (NL), pp. 15–20. IOS Press (2014)
8. Gmati, M., Atencia, M., Euzenat, J.: Tableau extensions for reasoning with link keys. In: Proceedings of the 11th International Workshop on Ontology Matching, Kobe, Japan, pp. 37–48 (2016)
9. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: a pratical OWL-DL reasoner. J. Web Semant. **5**(2), 51–53 (2007)
10. Serafini, L., Tamilin, A.: DRAGO: distributed reasoning architecture for the semantic web. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 361–376. Springer, Heidelberg (2005). https://doi.org/10.1007/11431053_25
11. Le Duc, C., Lamolle, M., Zimmermann, A., Curé, O.: DRAOn: a distributed reasoner for aligned ontologies. In: Informal Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE-2013), Ulm, Germany, 22 July 2013, pp. 81–86 (2013)
12. Horrocks, I., Sattler, U., Tobies, S.: Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In: McAllester, D. (ed.) CADE 2000. LNCS, vol. 1831, pp. 482–496. Springer, Heidelberg (2000). https://doi.org/10.1007/10721959_39
13. Shearer, R., Motik, B., Horrocks, I.: HermiT: a highly-efficient OWL reasoner. In: Ruttenberg, A., Sattler, U., Dolbear, C., (eds.) Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2008 EU), Karlsruhe, Germany, 26–27 October 2008
14. Liebig, T., Luther, M., Noppens, O., Wessel, M.: Owllink. Semant. Web **2**(1), 23–32 (2011)