

Kernel Contraction and Consolidation of Alignment under Ontology Change

Ahmed ZAHAF

University of Dr Moulay Tahar, Saida, Algeria,
EEDIS Laboratory, University of Djillali Liabès, Sidi BEL-ABBES, Algeria
E-mail: zhmda@gmail.com

Mimoun MALKI

ESI High School of Computer Science, Sidi BEL-ABBES, Algeria
EEDIS Laboratory, University of Djillali Liabès, Sidi BEL-ABBES, Algeria
E-mail: mymalki@gmail.com

Abstract—Alignment overcomes divergence in the specification of the semantics of vocabularies by different but overlapping ontologies. Therefore, it enhances semantic interoperability for many web based applications. However, ontology change following applications new requirements or new perception of domain knowledges can leads to undesirable knowledge such as inconsistent and therefore to a useless alignment. Ontologies and alignments are encoded in knowledge bases allowing applications to store only some explicit knowledge while they derive implicit ones by applying reasoning services on these knowledge bases. This underlying representation of ontologies and alignments leads us to follow base revision theory to deal with alignment revision under ontology change. For that purpose, we adapt kernel contraction framework to design rational operators and to formulate the set of postulates that characterize each class of these operators. We demonstrate the connection between each class of operators and the set of postulates that characterize them. Finally, we present algorithms to compute alignment kernels and incision functions. Kernels are sets of correspondences responsible of undesirable knowledge following alignment semantics. Incision functions determine the sets of correspondences to eliminate in order to restore alignment consistency or to realize a successful contraction.

Index Terms—Ontology Change, Alignment revision, Base Revision, Kernel Contraction, Kernel Consolidation.

I. INTRODUCTION

Ontologies play an important role in semantic web where they provide the semantic vocabulary used to annotate websites in a way meaningful for machine interpretation [1]. The independence in developing ontologies raises divergence in vocabulary specification. Alignment comes to overcome this divergence by specifying semantic relations between entities of ontologies. Thus, the infrastructure made by ontologies and alignments allows many web applications to take

advantage of full semantic interoperability [2]. Euzenat and Shvaiko [3] identify such applications. To name a few, Linked data, peer-to-peer information sharing, web services composition, autonomous communication systems, including agents and mobile devices communication, navigation and query answering on the web.

Usually, ontologies are subject of evolution where the vocabulary and its meaning are changed. Ontological change comes to reflect changes in covered domains or in response to applications needs [4]. In order to ensure a coherent semantic web, alignments should be revised too [5],[6],[7],[8],[9]. This is typically the case if adding new axioms in ontology makes some concepts unsatisfiable in other ontology when reasoning on alignment. Also, adding new ontological entities helps to find new correspondences and discarding other ones may affect some correspondences. The main challenge is how to revise alignment in order to embed the change.

Recently, some approaches [7],[8],[9] have emerged to deal with alignment revision under ontology change. The main challenge of these approaches is how to adapt alignment following ontology change. Influenced by the underlying representation of ontology, many properties about alignment quality are neglected. Considering ontologies as logical theories allows a recent approach [5] to define a formal and general framework for alignment revision mirroring AGM model [10] of belief revision theory. In this framework, ontologies are closed sets under the logical consequence of the underlying semantics of alignment. However, ontologies and hence alignments are encoded in knowledge bases making applications only holds a subset of domain knowledges as explicit and using reasoning services to derive implicit ones. This practical representation of ontologies and alignments leads us to consider a different approach based on base revision [11] to deal with the problem of alignment revision under ontology change. For that purpose, we adapt kernel contraction framework to

- Design and define rational operators for alignment contraction to deal with contracting axioms from

ontologies.

- Define and formulate the set of postulates that should characterize this class of operators.
- Demonstrate the connection between the postulates and these operators.
- Design and define rational operators for alignment consolidation to deal with alignment inconsistency when adding axioms to ontologies.
- Define and formulate the set of postulates that should characterize the class of alignment consolidation operators
- Demonstrate the connection between the postulates and this class of operators.
- And finally, we adapt the known Hitting set algorithm in diagnosis theory [12] to design an algorithm for computing kernels and therefore incision functions. Kernels are the set of correspondences responsible of alignment undesirable logical consequences such as inconsistency. Incision functions determine the sets of correspondences to eliminate in order to restore alignment consistency or to realize a successful contraction.

Before detailing these points, we introduce in section 2 the notion of ontology and alignment as there are adopted by semantic web community. In section 3, we introduce the base revision theory which constitutes the background of our framework. Section 4 details our framework. First, we present the problem of alignment revision under ontology change. Second, we justify our choice to follow base revision approach and finally, we present rational operators dealing with alignment revision. We reserve section 5 for the computational part of our framework. We compare our work with related works in section 6. Finally, we conclude in section 7 and we give some trends for future works.

II. ONTOLOGY ALIGNMENT

2.1 Ontology

An ontology is a logical theory [1], [5], [13]. It is a pair (S, A) , where S is a signature to designate a vocabulary and A is a set of axioms to specify the intended interpretation of this vocabulary in a domain of discourse. OWL is the standard language to represent ontologies on the web [14]. For reasons of expressiveness and reasoning efficiency, three sub-languages have been defined (OWL Lite, OWL DL, OWL FULL). The signature of an OWL DL ontology is the set $S = C \sqcup P \sqcup R \sqcup I$, where, C represents the set of vocabulary to designate concepts. P is the set of vocabulary to designate objects properties. R is the set of vocabulary to designate data properties and I is the set of vocabulary to designate individuals. The interpretation of this vocabulary in the domain of discourse can be illustrated by the following definition.

Definition 2.1 (Ontology Interpretation). Given a

signature $S = C \sqcup P \sqcup R \sqcup I$. An interpretation $I = (\Delta^I, \Delta_D^I, \cdot^I)$ consist of a set Δ^I which is an abstract domain; Δ_D^I a concrete data values and \cdot^I a function that map every concept of C to a subset of Δ^I , every object property of P to a subset of $\Delta^I \times \Delta^I$, every data property of R to a subset of $\Delta^I \times \Delta_D^I$ and every individual of I to an element of Δ^I .

Only interpretations that satisfy axioms in the ontology make sense. These interpretations are called models of ontology.

Definition 2.2 (Ontology Model). An interpretation I is a model of an ontology O if and only if I satisfies every axiom δ in that ontology ($\forall \delta \in O, I \models \delta$).

Logical consequence is a relation between an axiom and a consistent ontology. We use the notion of model to define it as follows.

Definition 2.3 (Ontology Consequence). An axiom δ is a logical consequence of an ontology O (noted $O \models \delta$) if and only if every model I of O , I satisfies δ .

We note by $Cn(O) = \{\delta | O \models \delta\}$ the closure set of logical consequences of an ontology O .

The logical consequence relation characterizes the underlying logic used to represent ontologies. Therefore, logics differ from each other's by the properties that characterize the logical consequence relation. For instance, logical consequence relation in description logics satisfies the following properties [15]:

$$\text{Inclusion } O \subseteq Cn(O) \quad (1)$$

$$\text{Iteration } Cn(O) \subseteq Cn(Cn(O)) \quad (2)$$

$$\text{Monotonicity if } O' \subseteq O, Cn(O') \subseteq Cn(O) \quad (3)$$

$$\text{Compactness if } O \models \delta \text{ then, there is some subset } O' \subseteq O \text{ such that } O' \models \delta \quad (4)$$

Contradicting axioms will allow no possible model [16]. An ontology which has no model is inconsistent [17].

Definition 2.4 (Inconsistent Ontology). An ontology O is inconsistent if and only if O has no model. Otherwise, it is consistent.

Consistency checking is always turns to instance checking problem [18]. Consequently, an ontology is inconsistent if and only if the contradictory axiom holds (i.e., $\perp(a)$, for some individual a).

One potential source of inconsistency is concepts unsatisfiability [17]. A concept is unsatisfiable if it is equivalent to an empty concept. Hence, unsatisfiable concept is always subsumed by the empty concept (i.e., $C \sqsubseteq \perp$).

Definition 2.5 (Concept and property unsatisfiabilities). A concept C (a property P) in an ontology O is unsatisfiable if and only if for each model I of O , we have $C^I = \emptyset$ ($P^I = \emptyset$).

Unsatisfiable concepts or properties are synonyms of logical problems in the ontology. This is categorized as a mistake in modelling or flaw in the ontology [19]. If an ontology contains unsatisfiable concepts or properties is called incoherent ontology. Otherwise, is coherent.

Definition 2.6 (Incoherent ontology). An ontology O is incoherent if and only if there exists an unsatisfiable concept or unsatisfiable property. Otherwise O is coherent.

Incoherence is a form of inconsistency at terminological part in Description Logic based ontologies [17]. However, an incoherent ontology might have models.

2.2 Alignment

An alignment is the output of an ontology matching process [3]. It expresses a set of correspondences between elements of different ontologies. Following [3], we define a correspondence as follows and introduce an alignment as set of correspondences.

Definition 2.7 (Correspondence and Alignment). Given ontologies o_1 and o_2 , let Q be a function that defines sets of matchable elements $Q(o_1)$ and $Q(o_2)$. A correspondence between o_1 and o_2 is a 4-tuple (e, e', r, n) such that $e \in Q(o_1)$, $e' \in Q(o_2)$, r is a semantic relation, and $n \in [0; 1]$ is a confidence value. An alignment M between o_1 and o_2 is a set of correspondences between o_1 and o_2 .

We restrict r to be one of the semantic relations from the set $\{\subseteq, \supseteq, \equiv, \perp\}$.

Example 1. We use Description Logic like syntax to describe ontologies. We use the index number in ontologies notation as name space to designate entities. O_1 and O_2 are two ontologies in education domain and M is an alignment between them.

$$O_1 = \left\{ \begin{array}{l} Student \subseteq Lecturer, \\ PhD\ Student \subseteq Student, \\ Researcher \subseteq Lecturer \end{array} \right\}$$

$$O_2 = \left\{ \begin{array}{l} PhD\ Student \subseteq Researcher, \\ PhD\ Student \subseteq Lecturer, \\ Lecturer \subseteq Employee, \\ PhD\ Student (Ahmed) \end{array} \right\}$$

$$M = \left\{ \begin{array}{l} 1: PhD\ Student =_{1.0} 2: PhD\ Student, \\ 1: Researcher =_{0.83} 2: Researcher, \\ 1: Lecturer =_{0.97} 2: Lecturer, \end{array} \right\}$$

In order to reason about alignment, two classes of approaches have been introduced. The first class is based

on model theory. DDL [20], for distributed description logics is an instance of this class. Based on an axiomatic approach, the second class called reductionist semantics [21]. Alignment semantics of this class interprets correspondences of the alignment as axioms in some merged ontology. The merged ontology is called aligned ontology. Without loss of generality, we use an instance of this semantics called natural semantic. It involves building a merged ontology through the union of the two ontologies to align and axioms obtained by translating relations of the alignment. We introduce this semantics through its aligned ontology.

Definition 2.8 (Natural Semantics). Given an alignment M between two ontologies o_1 and o_2 and $trans: M \rightarrow A$, a function that transforms a correspondence to an axiom. The natural semantics of M is defined by the following aligned ontology:

$$o_1 \cup_M o_2 = o_1 \cup o_2 \cup trans(M) \quad (5)$$

Example 2. The transformation of the alignment M of Example 1 to axioms is as follows.

$$Trans(M) = \left\{ \begin{array}{l} 1: PhD\ Student \equiv 2: PhD\ Student, \\ 1: Researcher \equiv 2: Researcher, \\ 1: Lecturer \equiv 2: Lecturer, \end{array} \right\}$$

We introduce the notion of alignment consequence according to natural semantics as follows.

Definition 2.9 (Alignment consequence): An axiom δ is an alignment consequence of an alignment M between two ontologies o_1 and o_2 if and only if δ is a logical consequence of the aligned ontology $o_1 \cup_M o_2$. We denote this relation by $M \models \delta$. δ either represents an ontological axiom or the image of a correspondence by the transformation function of the alignment.

The closure set of M is $Cn(M) = \{\delta \mid M \models \delta\}$. In the context of natural semantics (5), we can easily derive from (1), (2), (3) and (4) that alignment consequence relation satisfies the following properties:

$$\text{Inclusion } M \subseteq Cn(M) \quad (6)$$

$$\text{Iteration } Cn(M) \subseteq Cn(Cn(M)) \quad (7)$$

$$\text{Monotonicity if } M' \subseteq M, Cn(M') \subseteq Cn(M) \quad (8)$$

$$\text{Compactness if } M \models \delta \text{ then, there is some subset } M' \subseteq M \text{ such that } M' \models \delta \quad (9)$$

Some alignment consequences can affect the consistency of ontologies or the whole aligned ontology. In this case, alignment is called inconsistent.

Definition 2.10 (Alignment Inconsistency): given an aligned ontology $o_1 \cup_M o_2$, M is inconsistent with

respect to o_1 and o_2 if and only if both ontologies o_1 and o_2 are consistent but the aligned ontology $o_1 \cup_M o_2$ is inconsistent. Otherwise, M is consistent.

It follows from definition 2.10, inconsistencies due to ontologies and not to alignment can't be considered as alignment inconsistency.

Example 3. If we set $Researcher \perp Lecturer$ and we remove $PhD Student \sqsubseteq Lecturer$ from O_2 , we get a consistent ontology O_2 but the alignment M becomes inconsistent since no model can satisfy both assertions $lecturer(Ahmed)$ and $Researcher(Ahmed)$ (i.e. $\perp(Ahmed)$).

While alignment consistency preserves ontologies consistency, alignment coherence ensures the satisfiability preservation of ontological entities by the alignment.

Definition 2.11 (Incoherent Alignment): given an aligned ontology $o_1 \cup_M o_2$, M is incoherent with respect to o_1 and o_2 if and only if both ontologies o_1 and o_2 are coherent but the aligned ontology $o_1 \cup_M o_2$ is incoherent. Otherwise, M is coherent.

Example 4. Following example 3, If we remove the assertion $Phd Student (Ahmed)$, $Phd Student$ becomes unsatisfiable and hence the alignment M is incoherent.

III. BELIEF REVISION

Belief revision is a matured research field. It deals with how an agent rationally changes his beliefs. AGM [10] is the most influential model in belief revision research [22]. It is a formal framework in which, three types of belief change are defined: expansion, revision and contraction. Expansion consists in a set-theoretical adding of new knowledge to the set of belief. Expansion assumes no inconsistency occurs when adding knowledge. Revision should incorporate the new knowledge while ensuring consistency of the new set of beliefs. Contraction is the operation to give up knowledge without incorporating any new one. While expansion can be defined in unique way, there exists a class of operators for belief revision, as well as for contraction. Every class is characterized with a set of postulates and a set of constructors that should satisfy these postulates. An intuition guide in formulating these postulates has been the principle of minimal change according to which an agent should change his own beliefs as little as possible [23].

AGM Theory assumes the set of beliefs to be closed under logical consequence. This usually means dealing with infinite beliefs set which cannot be incorporated easily into a computational framework [23]. Another problem within AGM Theory considers beliefs to have an independent standing. However, some beliefs can only be derived from others one [11]. To overcome these problems, belief base was proposed as an alternative to represent the beliefs of an agent. It is a finite set not necessarily closed under logical consequence. Belief base

revision accepts the same types of change as AGM model: expansion, revision and contraction. Unlike AGM model, every constructor is characterized by a set of postulates that is different from the set of postulates of another constructor. Constructors in belief base revision are not equivalents. In what follows, we introduce two operators which serve a foundation for our operators that we will define for alignment revision. As well as we present the set of postulates and the theorem of representation of each operator.

3.1 Kernel Contraction

Given a belief base B and a particular belief α , the objective of contraction is to compute a subset of B that fails to imply α [24]. Kernel contraction is a particular operation of contraction. It consists in finding the set of minimal subset of B that imply α . This set is called the kernel of B by α and denoted by $B \parallel \alpha$. An element of the kernel $B \parallel \alpha$ is called α -kernel. Formally,

Definition 3.1 (Kernel): the kernel of B by α is the set of B' such that:

$$\left\{ \begin{array}{l} B' \sqsubseteq B \text{ (it is a subset of } B) \\ B' \models \alpha \text{ (that imply } \alpha) \\ \forall B'' \sqsubseteq B', B'' \not\models \alpha \text{ (and it is minimal)} \end{array} \right. \quad (10)$$

Then, Kernel contraction uses a function to discard from B at least one element from each α -kernel. This function is called incision function.

Definition 3.2 (Incision function): an incision function σ for B is a function that for all α :

$$\left\{ \begin{array}{l} \sigma(B \parallel \alpha) \sqsubseteq \sqcup(B \parallel \alpha) \\ \text{if } \emptyset \neq X \in B \parallel \alpha, \text{ then } X \cap \sigma(B \parallel \alpha) \neq \emptyset \end{array} \right. \quad (11)$$

Definition 3.3 (Kernel Contraction): let B a belief base, α a belief and σ an incision function, the kernel contraction of B by α is the operator defined as:

$$B -_{\sigma} \alpha = B \setminus \sigma(B \parallel \alpha) \quad (12)$$

The kernel contraction has proved to satisfy the following postulates [24]: success, inclusion, core-retainment and uniformity. The postulate success says that the retracted belief should not be believed after contraction unless it is a tautology. Inclusion ensures no new beliefs should be added to the belief base when realizing contraction. Satisfying Core-retainment means only beliefs that are responsible for implying the contracted belief should be discarded. The uniformity postulates requires that if every subset that implies some belief α implies also another belief β , then the contraction by α and β should be the same. The following representation theorem summarizes these postulates for every kernel contraction operator.

Theorem 3.1 (Kernel Contraction Representation): the operator $-$ is a kernel contraction for a belief base B if

and only if it satisfies the following postulates:

$$[success] \text{ if } \not\models \alpha \text{ then } B - \alpha \not\models \alpha \quad (13)$$

$$[Inclusion] B - \alpha \subseteq B \quad (14)$$

$$[Core - retainment] \text{ if } \beta \in B \text{ and } \beta \notin B - \alpha, \text{ then} \\ \text{there is a subset } B' \text{ of } B \text{ such that, } B' \not\models \alpha \\ \text{but } B' \sqcup \{\beta\} \models \alpha \quad (15)$$

$$[uniformity] \text{ if it holds for all } B' \subseteq B \text{ that } B' \models \alpha \text{ if} \\ \text{and only if } B' \models \beta, \text{ then } B - \alpha = B - \beta \quad (16)$$

3.2 Kernel consolidation

Consolidation is an operation that makes consistent an inconsistent belief base [25]. It can be modelled as contraction by the contradictory belief [26]. Kernel consolidation is kernel contraction by the contradictory belief. For each inconsistency element of the kernel, consolidation removes from the belief base at least one element that is responsible for this inconsistency. Formally,

Definition 3.3 (Kernel Consolidation): let B a belief base and σ an incision function, the kernel consolidation of B is the operator defined as:

$$B!_{C,\sigma} = B \setminus \sigma(B \parallel \perp) \quad (17)$$

The following theorem characterize kernel consolidation operator [26].

Theorem 3.2 (Kernel Consolidation Representation theorem): the operator ! is a kernel consolidation for a belief base B if and only if it satisfies the following postulates:

$$[Consistency] B! \not\models \perp \quad (18)$$

$$[Inclusion] B! \subseteq B \quad (19)$$

$$[Core - retainment] \text{ if } \beta \in B \text{ and } \beta \notin B!, \text{ then there} \\ \text{is a subset } B' \text{ of } B \text{ such that, } B' \not\models \perp \text{ but } B' \sqcup \{\beta\} \models \perp \\ (20)$$

IV. ALIGNMENT REVISION

Usually, ontologies are revised and local bugs are fixed independently to alignments. Therefore, we consider alignment revision under ontology change as the set of changes on correspondences of alignment to fulfil the satisfaction of some semantics constraints. We distinguish three types of changes on alignment: expansion, contraction and revision. Expansion is a set-theoretically adding a correspondence to an alignment. It can happen following adding new ontological entities and we need to align them with others entities. Revision restores alignment consistency following adding new correspondences or new axioms in ontologies.

Contraction is to discard correspondences when concerned entities are deleted from ontologies or some successfully removed axioms from ontologies still logical consequences of alignment. We consider in this framework, only revision when new axioms in ontologies make alignment inconsistent and contraction when successfully removed axioms still a logical consequence of alignment.

Example 5. Assuming the designer of O_2 of the Example 1 decides that PhD Students will no longer be lecturers. To do that, he removes the axiom $2:PhD\ Student \sqsubseteq Lecturer$ from O_2 and he obtains the new version $O'_2 = \left(\begin{array}{l} PhD\ Student \sqsubseteq Researcher, \\ Lecturer \sqsubseteq Employee, \\ PhD\ Student (Ahmed) \end{array} \right)$. But, this axiom stills a logical consequence of M as the following logical consequences demonstrate.

$2:PhD\ Student \sqsubseteq Researcher, 2:Researcher \equiv 1:Researcher, 1:Researcher \sqsubseteq Lecturer, 1:Lecturer \equiv 2:Lecturer, 2:PhD\ Student \sqsubseteq Lecturer.$

Example 6. Furthermore, the designer decides to set the concepts Researcher and Lecturer as disjoint. So, he revises O'_2 of the example 5 to add the axiom $Researcher \perp Lecturer$. The new version is

$$O''_2 = \left(\begin{array}{l} PhD\ Student \sqsubseteq Researcher, \\ Lecturer \sqsubseteq Employee, \\ Lecturer \perp Researcher, \\ PhD\ Student (Ahmed) \end{array} \right)$$

It is easy to verify that O''_2 is consistent but since no model can verify both axioms $2:Lecturer (Ahmed)$ and $2:Researcher (Ahmed)$, the alignment M is inconsistent. These axioms are alignment consequences of M. From $2:Phd\ Student(Ahmed), 2:Phd\ Student \sqsubseteq Researcher, 2:Researcher \equiv 1:Researcher, 1:Researcher \sqsubseteq Lecturer, 1:Lecturer \equiv 2:Lecturer$, we derive $2:Lecturer(Ahmed)$. From $2:Phd\ Student(Ahmed), 2:Phd\ Student \equiv 1:Phd\ Student, 1:Phd\ Student \sqsubseteq Researcher, 1:Researcher \equiv 2:Researcher$, we derive $2:Researcher(Ahmed)$.

In practical, ontologies are encoded in knowledge bases managed by knowledge systems to have access to and to reason about domain knowledge [1]. The set of axioms contained in these bases constitutes the explicit knowledge and implicit knowledges are logical consequences of them. Hence, our approach follows belief base revision approach instead AGM model. More precisely, our objective is to adapt the kernel contraction framework to design rational operators for alignment revision under ontology change. In what follows, we present two operators for alignment revision under ontology change. As well as, we present theorems of representation which summarize the postulates that characterize each operator.

4.1 Alignment Kernel Contraction

Given an alignment M between two ontologies o_1 and

α_2 and α is a successfully removed axiom from one ontology, the objective of contraction is to compute a subset of M that fails to imply α . Alignment kernel contraction consists in finding the set of minimal subsets of M that imply α . We call this set, the kernel of M by α and we denote it by $M \parallel \alpha$. We adapt base kernel definition (10) to define alignment kernel as follows:

Definition 4.1 (Alignment Kernel): the kernel of M by α ($M \parallel \alpha$) is the set of elements M' such that:

$$\left\{ \begin{array}{l} M' \subseteq M \text{ (subset of } M) \\ M' \models \alpha \text{ (} \alpha \text{ is a consequence of } M) \\ \forall M'' \subseteq M', M'' \not\models \alpha \text{ (and it is minimal)} \end{array} \right. \quad (21)$$

We call an element of the kernel ($M \parallel \alpha$) an α -Alignment kernel.

Example 7. The following set K is the kernel of alignment M of the example 5.

$$K = \left\{ \begin{array}{l} \{1: \text{Researcher} =_{0.83} \text{ } 2: \text{Researcher}, \\ \quad 1: \text{Lecturer} =_{0.97} \text{ } 2: \text{Lecturer},\} \\ \{1: \text{Lecturer} =_{0.83} \text{ } 2: \text{Lecturer}, \\ \quad 1: \text{PhD Student} =_{0.97} \text{ } 2: \text{PhD Student}\} \end{array} \right\}$$

Lemma 4.1: the following two conditions are equivalents.

$$\text{For all } M' \subseteq M, M' \models \alpha \text{ iff } M' \models \beta \quad (22)$$

$$M \parallel \alpha = M \parallel \beta \quad (23)$$

Proof: (necessary condition). We should demonstrate that $M \parallel \alpha \subset M \parallel \beta$ and $M \parallel \beta \subset M \parallel \alpha$ follow from the first condition (22). We give a proof for the first inclusion and the same proof holds for the second inclusion.

Let $M' \in M \parallel \alpha$, we should demonstrate that $M' \in M \parallel \beta$. From alignment kernel definition (21), $M' \in M \parallel \alpha$ means $M' \subseteq M$, $M' \models \alpha$ and $\forall M'' \subseteq M', M'' \not\models \alpha$. According to the first condition (22), we have $M' \models \beta$. From alignment kernel definition (21), we conclude $M' \in M \parallel \beta$.

(Sufficient condition). We should demonstrate that the first condition (22) follows from the second (23). Let $M' \models \alpha$ for $M' \subset M$. By alignment compactness (9), there exists a subset $M'' \subset M'$ such that $M'' \models \alpha$. Let M'' be the minimal one. From alignment kernel definition (21), $M'' \in M \parallel \alpha$. According to the second condition (23), we have $M'' \in M \parallel \beta$ and hence, $M'' \models \beta$. By alignment monotony (8), we conclude that $M' \models \beta$. The same proof holds for the inverse.

Alignment kernel contraction uses a function to discard from M at least one element from each α -Alignment kernel. We call such function alignment incision function. We adapt base incision function definition (11) to define alignment incision function as follows:

Definition 4.2 (Alignment Incision function): an incision

function σ for M is a function that for all:

$$\left\{ \begin{array}{l} \sigma(M \parallel \alpha) \subseteq \sqcup(M \parallel \alpha) \\ \text{if } \emptyset \neq X \in M \parallel \alpha, \text{ then } X \cap \sigma(M \parallel \alpha) \neq \emptyset \end{array} \right. \quad (24)$$

Example 8. A possible incision function for the kernel K of the example 7 is.

$$I = \left\{ \begin{array}{l} 1: \text{Researcher} =_{0.83} \text{ } 2: \text{Researcher}, \\ 1: \text{PhD Student} =_{1.0} \text{ } 2: \text{PhD Student} \end{array} \right\}$$

Therefore, we can define alignment kernel contraction by adapting base kernel contraction operator (12) as follows:

Definition 4.3 (Alignment Kernel Contraction): let M an alignment between two ontologies α_1 and α_2 , α a successfully removed axiom from one ontology and σ an alignment incision function, the alignment kernel contraction of M by α is the operator defined as:

$$M -_{\sigma} \alpha = M \setminus \sigma(M \parallel \alpha) \quad (25)$$

Example 9. If we consider the incision function of the example 8, the kernel contraction of M by $\text{PhD Student} \subseteq \text{Lecturer}$ is,

$$M -_{\sigma} \alpha = \{1: \text{Lecturer} =_{0.97} \text{ } 2: \text{Lecturer}\}$$

We adapt the postulates (13), (14), (15) and (16) of belief base kernel contraction to define the postulates that alignment kernel contraction should satisfy them. For alignment kernel contraction, success means that successfully removed axioms from ontologies should not be regenerated again by alignment after contraction. Inclusion ensures no new correspondences should be added to alignment when realizing contraction. Satisfying Core-retainment means only correspondences that are responsible for implying the contracted axiom should be discarded. The uniformity postulates requires that if every subset of the alignment together with ontologies that implies some axiom α implies also another axiom β , then the contraction of alignment by α and β should be the same. The following representation theorem summarizes these postulates for every alignment kernel contraction operator.

Theorem 4.1 (Alignment Kernel Contraction Representation theorem): An operator $-$ is an alignment kernel contraction of an alignment M between two ontologies α_1 and α_2 for a successfully removed axiom α from an ontology if and only if it satisfies the following postulates:

$$[\text{success}] \text{ if } \not\models \alpha \text{ then } M - \alpha \not\models \alpha \quad (26)$$

$$[\text{Inclusion}] M - \alpha \subseteq M \quad (27)$$

$$[\text{Core - retainment}] \text{ if } c \in M \text{ and } c \notin M - \alpha, \text{ then there is a subset } M' \text{ of } M \text{ such that, } M' \not\models \alpha$$

$$\text{but } M' \sqcup \{c\} \vDash \alpha \quad (28)$$

[uniformity] if it holds for all $M' \subseteq M$ that $M' \vDash \alpha$ if and only if $M' \vDash \beta$, then $M - \alpha = M - \beta$ (29)

Proof: (necessary condition). Let $-_{\sigma}$ be an alignment contraction operator such that $M -_{\sigma} \alpha = M \setminus \sigma(M \parallel \alpha)$ for some incision function σ and demonstrates that it satisfies the postulates: success (26), inclusion (27), core-retainment (28) and uniformity (29).

Success (26) and inclusion (27) follow directly from operator definition. Suppose $c \in M$ and $c \notin M -_{\sigma} \alpha$, then $c \in \sigma(M \parallel \alpha)$. From alignment incision function definition (24), we have $\sigma(M \parallel \alpha) \sqsubseteq \sqcup(M \parallel \alpha)$, so there is some set A such that $c \in A \in (M \parallel \alpha)$. Let $M' = A \setminus \{c\}$, then $M' \not\vDash \alpha$ but $M' \sqcup \{c\} \vDash \alpha$. This satisfies core-retainment (28). From lemma 4.1, For all subset M' of M , $M' \vDash \alpha$ if and only if $M' \vDash \beta$ (22) is equivalent to $M \parallel \alpha = M \parallel \beta$ (23). Since σ is a function then $\sigma(M \parallel \alpha) = \sigma(M \parallel \beta)$. It follows $M \setminus \sigma(M \parallel \alpha) = M \setminus \sigma(M \parallel \beta)$. Hence, $M -_{\sigma} \alpha = M -_{\sigma} \beta$. We conclude that $-_{\sigma}$ satisfies success (26), inclusion (27), core-retainment (28) and uniformity (29).

(Sufficient condition). Let $-$ be a contraction operator on an alignment M such that the four postulates (26) (27) (28) (29) are satisfied. We are going to demonstrate that $-$ is a kernel contraction. For that purpose, let σ be such that for α : $\sigma(M \parallel \alpha) = M \setminus M - \alpha$. We need to verify that σ is an incision function for M . To be that, it must: first, be a function and second such that it satisfies i) $\sigma(M \parallel \alpha) \sqsubseteq \sqcup(M \parallel \alpha)$ and ii) if $\emptyset \neq X \in M \parallel \alpha$, then $X \cap \sigma(M \parallel \alpha) \neq \emptyset$. Furthermore, we need to verify that $-$ applied to M coincides with $-_{\sigma}$.

Proof that σ is a function. Let α and β be two correspondences such that $M \parallel \alpha = M \parallel \beta$. It follows from Lemma 4.1 (22) (23) and uniformity (29) that $M - \alpha = M - \beta$. Following our definition $\sigma(M \parallel \alpha) = M \setminus M - \alpha$, we conclude $\sigma(M \parallel \alpha) = \sigma(M \parallel \beta)$.

Proof that the condition i) $\sigma(M \parallel \alpha) \sqsubseteq \sqcup(M \parallel \alpha)$ is satisfied. Let $c \in \sigma(M \parallel \alpha)$. By core-retainment (28), it follows that there is some $A \sqsubseteq M$ such that $A \not\vDash \alpha$ and $A \sqcup \{c\} \vDash \alpha$. By compactness (9), there is some finite subset $A' \sqsubseteq A$ such that $A' \sqcup \{c\} \vDash \alpha$. From $A' \not\vDash \alpha$ and $A' \sqcup \{c\} \vDash \alpha$, it follows that there is some α -alignment kernel A'' that contains c . Then, $c \in A'' \in \sqcup(M \parallel \alpha)$.

Proof that the condition ii) if $\emptyset \neq X \in M \parallel \alpha$, then $X \cap \sigma(M \parallel \alpha) \neq \emptyset$ is satisfied. Suppose that $\emptyset \neq X \in M \parallel \alpha$. Then $X \vDash \alpha$ and by success (26), we have $M - \alpha \not\vDash \alpha$. Since $X \vDash \alpha$, we may conclude that $X \not\subseteq M - \alpha$, i.e., that there is some c such that $c \in X$ and $c \notin M - \alpha$. Since $X \sqsubseteq M$, it follows $c \in M \setminus M - \alpha$; i.e., $c \in \sigma(M \parallel \alpha)$. Thus, $c \in X \cap \sigma(M \parallel \alpha)$ which is sufficient to show condition ii) satisfaction.

Proof that $-$ applied to M coincides with $-_{\sigma}$. By inclusion ($M - \alpha \subseteq M$) and our definition of $\sigma(M \parallel \alpha) = M \setminus M - \alpha$, it follows $M - \alpha = M \setminus \sigma(M \parallel \alpha)$. This finishes the proof.

4.2 Alignment Kernel consolidation

We define alignment consolidation as all operation that makes consistent an alignment. Alignment kernel consolidation is alignment kernel contraction by the contradictory axiom (i.e., \perp (a)). For each inconsistency element of alignment kernel, consolidation removes from alignment at least one element that is responsible for this inconsistency. Formally,

Example 10. The kernel consolidation of alignment M of the example 6 is the same as the kernel contraction of the example 7.

$$K = \left\{ \left\{ \begin{array}{l} 1: \text{Researcher} =_{0.83} 2: \text{Researcher}, \\ 1: \text{Lecturer} =_{0.97} 2: \text{Lecturer}, \end{array} \right\}, \left\{ \begin{array}{l} 1: \text{Lecturer} =_{0.83} 2: \text{Lecturer}, \\ 1: \text{PhD Student} =_{0.97} 2: \text{PhD Student} \end{array} \right\} \right\}$$

We can choose another incision function for this case. $I = \{1: \text{Lecturer} =_{0.97} 2: \text{Lecturer}\}$

We can define alignment kernel consolidation by adapting base kernel consolidation operator (17) as follows:

Definition 3.3 (Alignment Kernel Consolidation): let M be an alignment between two ontologies o_1 and o_2 and σ an alignment incision function, alignment kernel consolidation of M is the operator defined as:

$$M!_{c,\sigma} = M \setminus \sigma(M \perp \perp (a)) \quad (30)$$

Example 11. If we consider the incision function of the example 10, the kernel consolidation of alignment M is

$$M!_{c,\sigma} = \left\{ \begin{array}{l} 1: \text{PhD Student} =_{1.0} 2: \text{PhD Student}, \\ 1: \text{Researcher} =_{0.83} 2: \text{Researcher} \end{array} \right\}$$

We consider also, an operator that makes an alignment coherent as a particular alignment consolidation. In this case, the contraction is done by the following subsumption axiom (i.e., $C \subseteq \perp$).

By adapting the postulates (18), (19) and (20) alignment kernel consolidation satisfies the following postulates: consistency, inclusion and core-retainment. The following theorem presents these postulates. The same holds for alignment consolidation in case of a coherent alignment. However, we rename the consistency postulate by coherency postulate.

Theorem 4.2 (Alignment Kernel Consolidation Representation theorem): An operator $!$ is an alignment kernel consolidation of an alignment M between two ontologies o_1 and o_2 if and only if satisfy the following postulates:

$$[\text{Consistency}] M! \not\subseteq \perp (a) \quad (31)$$

$$[\text{Inclusion}] M! \subseteq M \quad (32)$$

[Core – retainment] if $c \in M$ and $c \notin M!$, then there is a subset M' of M such that, $M' \not\models \perp(a)$
but $M' \sqcup \{c\} \models \perp(a)$ (33)

Proof: (necessary condition). Let $M!_{c,\sigma} = M \setminus \sigma(M \parallel \perp(a))$ be an alignment kernel consolidation operator. Inclusion (32) follows from the definition. We are going to demonstrate that $M!_{c,\sigma}$ satisfies consistency (31). Suppose $\perp(a) \in M!_{c,\sigma}$, by compactness (9) there is some subset $A \subseteq M!_{c,\sigma}$ and $A \models \perp(a)$. Hence, there is some inclusion-minimal subset $C \subseteq A$ such that $C \models \perp(a)$. Thus, $C \in (M \parallel \perp(a))$. Due to $\not\models \perp(a)$, then $C \neq \emptyset$. Following the definition of alignment incision function (24), $C \cap \sigma(M \parallel \perp(a)) \neq \emptyset$. Then, there is some element $c \in C$ and $c \in \sigma(M \parallel \perp(a))$. According to the operator definition, $c \in C \subseteq A \subseteq M!_{c,\sigma}$. But $c \in \sigma(M \parallel \perp(a))$. Hence, C cannot be a subset of $M!_{c,\sigma}$. We conclude that $\perp(a) \notin M!_{c,\sigma}$. Let $c \in M$ and $c \notin M!_{c,\sigma}$. Then, $c \in \sigma(M \parallel \perp(a))$. Following the definition of alignment incision function (24), there is some set C such that $c \in C \in (M \parallel \perp(a))$. Let $X = C \setminus \{c\}$. Then, $X \not\models \perp(a)$ and $X \sqcup \{c\} \models \perp(a)$. This demonstrates the satisfaction of core-retainment postulate (33).

(Sufficient condition). Let $!$ be an alignment consolidation operator on an alignment M such that the three postulates (31) (32) (33) are satisfied. We are going to demonstrate that $!$ is an alignment kernel consolidation based on some alignment incision function σ . For that purpose, let σ be such that for α : $\sigma(M \parallel \alpha) = M \setminus M!$. We need to verify that σ is an incision function for M and to verify $!$ applied to M coincides with $!_{c,\sigma}$. Be that, it must satisfying i) $\sigma(M \parallel \perp(a)) \sqsubseteq \sqcup(M \parallel \perp(a))$ and ii) if $\emptyset \neq A \in M \parallel \perp(a)$, then $A \cap \sigma(M \parallel \perp(a)) \neq \emptyset$.

Clearly σ is a function. To show the first condition i) $\sigma(M \parallel \perp(a)) \sqsubseteq \sqcup(M \parallel \perp(a))$, let $c \in \sigma(M \parallel \perp(a))$. It follows from core-retainment (33) that there is some A such that $A \subseteq M$, $A \not\models \perp(a)$ and $A \sqcup \{c\} \models \perp(a)$. By compactness (9), there is some subset $A' \subseteq A$ such that $A' \sqcup \{c\} \models \perp(a)$. Let A'' an inclusion-minimal subset of A' such that $A'' \sqcup \{c\} \models \perp(a)$. Hence, there is some α -Alignment kernel C such that $c \in C \in (M \parallel \perp(a))$. For the second condition ii), let $\emptyset \neq A \in M \parallel \perp(a)$. By consistency, $M! \not\models \perp(a)$. Since $A \models \perp(a)$, by monotony (8) $A \not\subseteq M!$. That there is a correspondence $c \in A$ and $c \notin M!$. Since, $A \subseteq M$, $c \in M \setminus M!$. This means $c \in \sigma(M \parallel \perp(a))$. Thus, $c \in A \cap \sigma(M \parallel \perp(a))$. This finishes the proof that σ is an alignment incision function.

It follows from inclusion (32) and our definition of σ (i.e., $\sigma(M \parallel \alpha) = M \setminus M!$), $M! = M \setminus \sigma(M \parallel \alpha)$. We conclude that $!$ is an alignment kernel consolidation (i.e., $M! = M!_{c,\sigma}$).

V. COMPUTING ALIGNMENT KERNEL AND INCISION FUNCTIONS

The algorithm to find an α -Alignment kernel is an adaptation of the algorithm presented in [27] to compute a minimal subset of an ontology that is responsible for an

entailment of a given subsumption axiom (see Table1). It consists in removing each element of M and testing if the resulting alignment still implies the axiom α . If this is not the case the element is reintroduced in M . The result of this process is a set $M' \sqsubseteq M$ that do imply α which is minimal. Similar to the algorithm presented in [27], algorithm 1 can compute an α -Alignment kernel in polynomial time in the size of the aligned ontology.

Table 1. α -Alignment kernel algorithm.

Algorithm 1: α -Alignment kernel
α -Alignment kernel (M, o_1, o_2, α) 1. for $c \in M$ 2. do 3. if $M \setminus \{c\} \models \alpha$ 4. then $M \leftarrow M \setminus \{c\}$ 5. return M

Example 12. We take the ontologies and the alignment of the example 5.

$$O_1 = \left\{ \begin{array}{l} Student \sqsubseteq Lecturer, \\ PhD Student \sqsubseteq Student, \\ Researcher \sqsubseteq Lecturer \end{array} \right\}$$

$$O'_2 = \left\{ \begin{array}{l} PhD Student \sqsubseteq Researcher, \\ Lecturer \sqsubseteq Employe, \\ Phd Student (Ahmed) \end{array} \right\}$$

$$M = \left\{ \begin{array}{ll} 1: PhD Student =_{1.0} & 2: PhD Student, \\ 1: Researcher =_{0.83} & 2: Researcher, \\ 1: Lecturer =_{0.97} & 2: Lecturer, \end{array} \right\}$$

Let α be $2: PhD Student \sqsubseteq Lecturer$. We want to compute an α -Alignment kernel by using the algorithm 1.

1. The algorithm iterates over the elements of M (Line 1). Let's assume that it iterates from left to right.
2. Checks $M / \{1: PhD Student =_{1.0} \ 2: PhD Student\} \models \alpha$ (line 3), so it removes $1: PhD Student =_{1.0} \ 2: PhD Student$ from M (line 4).
3. Checks $M / \{1: Researcher =_{0.83} \ 2: Researcher\} \not\models \alpha$, then it does not change M (line 3).
4. Checks $M / \{1: Lecturer =_{0.97} \ 2: Lecturer\} \not\models \alpha$, so it does not change M (line 3).
5. Return $M = \left\{ \begin{array}{ll} 1: Researcher =_{0.83} & 2: Researcher, \\ 1: Lecturer =_{0.97} & 2: Lecturer \end{array} \right\}$ which is an α -Alignment kernel (line 5).

To compute alignment Kernel and incision functions, we adapt the Hitting set algorithm proposed by Reiter [12] to diagnose systems. Given a collection of sets F , a Hitting set is a set that intersects each set of the collection. Hitting set algorithm builds a Tree for a collection of sets F such that, its root is labeled by $\sqrt{}$ if F is empty.

Otherwise, it is labeled by an arbitrary set of F. If n is a node of the tree, define $H(n)$ to be the set of edge labels on the path from the root to the node n . If n is labeled by $\sqrt{\cdot}$, it has no successor nodes in the tree. If n is labeled by a set Σ of F, then for each $\sigma \in \Sigma$, n has a successor node n_σ joined to n by an edge labeled by σ . The label for n_σ is a set $S \in F$ such that $S \cap H(n) = \emptyset$, if such a set S exists. Otherwise, n_σ is labeled by $\sqrt{\cdot}$.

By definition, alignment incision function (24) intersects each α -Alignment kernel. If we consider alignment kernel (21) to be the collection F, it seems naturel to consider incisions functions as its Hitting sets. The nodes of our tree are labeled by α -Alignment kernels and edges are labeled by the elements of these α -Alignment kernels. However, the kernel is not given explicitly and we should compute it. At each node, an α -Alignment kernel of the set $M \setminus H(n)$ is computed if such an α -Alignment kernel exists. Otherwise, $H(n)$ is an alignment incision function. The table 2 outlines this algorithm.

Table 2. Alignment kernel and Incision functions algorithm

Algorithm 2: Alignment Kernel and Incision functions
AlignmentKernelAndIncisionFct (M, o_1, o_2)
1. Incision $\leftarrow \emptyset$
2. Stack \leftarrow Empty
3. $C \leftarrow \alpha$ -Alignment kernel (M, o_1, o_2, α)
4. AKernel $\leftarrow \{C\}$
5. for $c \in C$
6. do insert $\{c\}$ in the top of the stack
7. While Stack not Empty
8. do $Hn \leftarrow$ last element of the stack
9. remove last element of the stack
10. If $M \setminus \{Hn\} \models \alpha$
11. Then $C \leftarrow \alpha$ -Alignment kernel ($M \setminus \{Hn\}, o_1, o_2, \alpha$)
12. AKernel \leftarrow AKernel $\cup \{C\}$
13. for $c \in C$
14. do insert $Hn \cup \{c\}$ in the top of the stack
15. Else Incision \leftarrow Incision $\cup \{Hn\}$
16. End.

Example 13. Following example 12,

1. Algorithm 2 starts by computing one α -Alignment kernel. Let it the same as in example 12:

$$C = \left\{ \begin{array}{l} 1: \text{Researcher} =_{0.83} 2: \text{Researcher}, \\ 1: \text{Lecturer} =_{0.97} 2: \text{Lecturer}, \end{array} \right\} \text{ (line 3-4)}$$

2. Push $\{c\}$ into the stack for every element of C (line 5-6). The stack contains now

$$\text{stack} = \left\{ \begin{array}{l} 1: \text{Researcher} =_{0.83} 2: \text{Researcher}, \\ 1: \text{Lecturer} =_{0.97} 2: \text{Lecturer} \end{array} \right\}$$

3. Get the first element of the stack into Hn .

$$Hn = \{1: \text{Lecturer} =_{0.97} 2: \text{Lecturer}\} \text{ and } \text{stack} = \{1: \text{Researcher} =_{0.83} 2: \text{Researcher}\} \text{ (line 8-9)}$$

4. Checks $M / \{1: \text{Lecturer} =_{0.97} 2: \text{Lecturer}\} \not\models \alpha$, then Incision = $\{\{1: \text{Lecturer} =_{0.97} 2: \text{Lecturer}\}\}$ (lines 10 and 15).
5. Loop line(7).
6. Get the first element of the stack into Hn .

$$Hn = \{1: \text{Researcher} =_{0.83} 2: \text{Researcher}\} \text{ and } \text{stack} = \emptyset \text{ (line 8-9)}$$

7. Checks $M / \{1: \text{Researcher} =_{0.83} 2: \text{Researcher}\} \models \alpha$ line(10), then
8. Run algorithm 1 again, we obtain

$$C = \left\{ \begin{array}{l} 1: \text{PhD Student} =_{1.0} 2: \text{PhD Student}, \\ 1: \text{Lecturer} =_{0.97} 2: \text{Lecturer}, \end{array} \right\} \text{ (line 11)}$$

9. Push $Hn \cup \{c\}$ into the stack for every element of C (line 13-14). The stack contains now,

$$\text{stack} = \left\{ \begin{array}{l} \{1: \text{Researcher} =_{0.83} 2: \text{Researcher}, \\ 1: \text{PhD Student} =_{1.0} 2: \text{PhD Student}\}, \\ \{1: \text{Researcher} =_{0.83} 2: \text{Researcher}, \\ 1: \text{Lecturer} =_{0.97} 2: \text{Lecturer}\} \end{array} \right\}$$

10. Loop line(7).
11. Get the first element of the stack into Hn .

$$Hn = \left\{ \begin{array}{l} 1: \text{Researcher} =_{0.83} 2: \text{Researcher}, \\ 1: \text{Lecturer} =_{0.97} 2: \text{Lecturer} \end{array} \right\} \text{ and } \text{stack} = \left\{ \begin{array}{l} \{1: \text{Researcher} =_{0.83} 2: \text{Researcher}, \\ 1: \text{PhD Student} =_{1.0} 2: \text{PhD Student}\} \end{array} \right\} \text{ (line 8-9)}$$

12. Checks that $M / Hn \not\models \alpha$ line(10), then

$$\text{Incision} = \left\{ \begin{array}{l} \{1: \text{Lecturer} =_{0.97} 2: \text{Lecturer}\}, \\ \{1: \text{Researcher} =_{0.83} 2: \text{Researcher}, \\ 1: \text{Lecturer} =_{0.97} 2: \text{Lecturer}\} \end{array} \right\} \text{ (lines 10 and 15)}$$

13. Loop line(7).
14. Get the first element of the stack into Hn .

$$Hn = \left\{ \begin{array}{l} 1: \text{Researcher} =_{0.83} 2: \text{Researcher}, \\ 1: \text{PhD Student} =_{1.0} 2: \text{PhD Student} \end{array} \right\} \text{ and } \text{stack} = \emptyset \text{ (line 8-9)}$$

15. Checks that $M / Hn \not\models \alpha$ line(10), then Incision =

$$\left\{ \begin{array}{l} \{1: \text{Lecturer} =_{0.97} 2: \text{Lecturer}\}, \\ \left\{ \begin{array}{l} 1: \text{Researcher} =_{0.83} 2: \text{Researcher}, \\ 1: \text{Lecturer} =_{0.97} 2: \text{Lecturer} \end{array} \right\}, \\ \left\{ \begin{array}{l} 1: \text{Researcher} =_{0.83} 2: \text{Researcher}, \\ 1: \text{PhD Student} =_{1.0} 2: \text{PhD Student} \end{array} \right\} \end{array} \right\}$$

16. The stack is empty, line(7).
17. End (line 16)

Our proposed approach is implemented in java. The platform of our prototype is based on OWL API [28] and Align API [29] to manipulate OWL ontologies and alignments between them. The platform integrates pellet [30] as the main reasoning engine on OWL ontologies.

VI. RELATED WORKS

Ontology alignment revision has known the emergence of several approaches in recent years [5], [6]. The main focus of these approaches is maintaining alignment following inconsistency detection. Alignment inconsistency may be the consequence of ontology revision or errors in designing and computing alignment. Therefore, we distinguish two types of approaches. The first type concerns approaches that focus on alignment adaptation following ontology evolution. Usually, they use ontology change handler to guide alignment adaptation process. Groß et al. [7] convert ontology change to an alignment between version of the evolved ontology and compose it with the old alignment to generate the new one. Dinh et al. [8] associates a set of actions for every type of change to deal with alignment adaptation. These actions can add, delete, move, derive a correspondence or change the semantic type of its relation. In another approach [9], alignment is an instance of Semantic Bridge Ontology (SBO). This ontology serves a representation and exchange mechanism of semantic relationships between ontologies. The evolution of alignment in this approach is a process that aims to preserve the semantics of this ontology. This is to detect and correct the invalid entities of SBO. Inspired by ontology evolution strategies [4], the approach proposes a list of strategies to correct invalid entities of SBO. In this approach, only deleted concepts are considered. In all these approaches, the notion of alignment inconsistency is not explicitly defined. Instead, they consider alignment as invalid if some its correspondences are affected by ontology change. Ontological change can affect entities implied in alignment or the relations between them. In previous work [31], we have introduced some constraints that an alignment between versions of the same ontology should satisfy. Changed meaning conservation corresponds to the postulate of success in kernel contraction. Another constraint that has no correspondence in base revision theory is meaning preservation. This constraint is one instance for many that characterize alignment quality [32].

The second supports the problem of inconsistency caused by errors in designing and computing alignment. These approaches can be inserted as an additional component to matching tools to improve the quality of alignment. Ontology matching tools [3] use the knowledge encoded in the ontology to determine correspondences between entities of ontologies. Terminological techniques compare the lexicon used to designate ontological entities, while the semantic one are based on model theory to determine the existence of a correspondence between two entities. Some tools consider the internal structure of the ontology. Others

consider the external structure of the ontology. The ontology extension can also be used. The majority of the existing matching systems combine these techniques to cover different aspects of the ontology. The alignment result of these tools may be subject to invalid correspondences. For some tools, the determination of these correspondences is based on patterns of reasoning. For instance, Lily [33] uses four types of patterns, e.g., redundant mapping, imprecise mapping, Inconsistent mapping and abnormal mapping. ASMOV [34] uses five types of patterns to check semantics, e.g., Multiple-entity correspondences, Crisscross correspondences, disjoint-subsumption contradiction, Subsumption and equivalence incompleteness, Domain and Range incompleteness. The pattern disjoint-subsumption contradiction used by ASMOV corresponds to inconsistent mapping pattern used by Lily. YAM++ [35] relies on ALCOMO¹ system to debug alignment. ALCOMO [36] uses disjoint-subsumption contradiction pattern to check coherency of alignment (see definition 2.11). These patterns are correct but incomplete reasoning methods to verify satisfiability preservation. Based on diagnosis theory, independent approaches [21],[37] use techniques to revise alignment. The revision is triggered following alignment incoherence violation. First, they compute conflict sets which are composed by correspondences responsible for incoherency of alignment. Then, they select from each conflict set one correspondence to form a diagnosis. The selection should be as minimal as possible to compute diagnosis. Finally, the diagnosis is discarded from alignment to restore coherency.

An intuition guide to revise alignment for both types of approaches is the principle of minimal change. However none of these approaches demonstrates the satisfaction of this constraint when revising alignment. An interesting and recent work assembles both types of revision in a single and general framework [5]. The framework defines a set of postulates and operators for revision of network of ontologies that should satisfy them mirroring the framework of AGM model. Basically, our framework relies on base revision theory. Besides, the advantages of this theory over belief revision (see section 3), we can mention the following points that differentiate our framework. On base revision, we cannot define a fixed set of postulates that should be satisfied by any constructor as in AGM model. Instead, every operator is characterized by a different set of postulates which characterize it. We have restricted our framework to the natural semantics alignment but the results are valid in any alignment semantics except those don't satisfy compactness and monotony. Our framework deals only with alignment revision under ontology change, we plan to extend it to include other use cases on the light of base revision theory.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we have presented a formal framework to

¹ <http://web.informatik.uni-mannheim.de/alcomo/>

deal with the problem of alignment revision under ontology change. We have defined alignment kernel contraction operator to deal with the problem of discarding axioms from ontologies, the set of postulates that characterize it and we have demonstrated the connection between the operator and the set of postulates in what we call representation theorem. Furthermore, we have defined another operator that we have called alignment kernel consolidation. The objective of this operator is to restore alignment consistency following adding axioms in ontologies implied in alignment. We have also defined the set of postulates that should characterize this operator and demonstrated the connection between them. The major challenge for these operators is alignment kernel and incision functions computing. For that purpose, we have designed and implemented an algorithm by adapting the known Hitting set algorithm in diagnosis theory.

Our framework can be extended in many ways. We can integrate others operators such as partial meet contraction and consolidation. Our framework is limited to alignment revision under ontology change. Always on the light of base revision, we investigate how to deal with the problem of adding and discarding correspondences from alignment.

REFERENCES

- [1] Grimm,S.,Abecker,A., Völker,J., Studer,R. Ontologies and the Semantic Web. In: Domingue, J., Fensel, D., Hendler, J.A. (eds) Handbook of Semantic Web Technologies; ch 13, pp. 507-579. Springer 2011
- [2] G Yang,G. Feng,J. Database Semantic Interoperability based on Information Flow Theory and Formal Concept Analysis. International Journal of Information Technology and Computer Science, PP.33-42, 2012, DOI: 10.5815/ijitcs.2012.07.05.
- [3] Euzenat, J., Shvaiko, P. Ontology Matching. Second edition. Springer-Verlag, Berlin Heidelberg (DE) 2013.
- [4] Stojanovic, L. Methods and Tools for Ontology Evolution. PhD thesis, University of Karlsruhe. 2004
- [5] Euzenat,J. Revision in networks of ontologies. Artificial Intelligence Volume 228, November 2015, Pages 195–216,(doi:10.1016/j.artint.2015.07.007)
- [6] Dos Reis, J. C., Pruski, C. and Reynaud, C. "State-of-the-art on mapping maintenance and challenges towards a fully automatic approach." Expert Systems with Applications 42.3 (2015): 1465-1478.
- [7] Groß A., Dos Reis, J. C., Hartung, M., Pruski, C., & Rahm, E. Semi-automatic adaptation of mappings between life science ontologies. In C. Baker, G. Butler, & I. Jurisica (Eds.), Data integration in the life sciences. Lecture notes in computer science (Vol. 7970, pp. 90–104). Berlin Heidelberg: Springer.
- [8] Dinh, D., Dos Reis, J.C., Pruski, C., Da Silveira, M., Reynaud-Delafrère, C. Identifying Relevant Concept Attributes to Support Mapping Maintenance under Ontology Evolution, Journal of Web Semantics (JWS): Science, Services and Agents on the World Wide Web, 29(0), 53-66, Life Science and e-Science, (DOI: 10.1016/j.websem.2014.05.002), 2014, Elsevier.
- [9] Martins, H., Silva, N. A User-Driven and a Semantic-Based Ontology Mapping Evolution Approach. In Proc. of ICEIS, 2009.
- [10] Alchourran,C., Gardenfors,P. Makinson,D. On the logic of theory change: partial meet contraction and revision functions, J. Symb. Log. 50(2) (1985) 510–530.
- [11] Hansson, S. O. Logic of belief revision. In E. N. Zalta (Ed.), The Stanford encyclopedia of philosophy. The Metaphysics Research Lab.Center for the Study of Language and Information.Stanford University, 2006. <http://plato.stanford.edu/entries/logic-belief-revision>
- [12] Reiter,R. A theory of diagnosis from first principles. Artificial Intelligence, 1987, 32(1): 57~96.
- [13] Kalfoglou, Y., Schorlemmer, M. Ontology mapping: the state of the art. The Knowledge Engineering Review 18(1), 1–31 2003
- [14] Horrocks, I., Patel-Schneider,P.F. Knowledge Representation and Reasoning on the Semantic Web: OWL. In: Domingue, J., Fensel, D., Hendler, J.A. (eds) Handbook of Semantic Web Technologies; ch 9, pp. 365-398. Springer 2011
- [15] Ribeiro, M., Wassermann, R., Antoniou, G., Flouris, G. and Pan, J. Belief contraction in Web-ontology languages. In: Proceedings of the 3rd International Workshop on Ontology Dynamics (IWOD-09), 2009.
- [16] Haase, P., Stojanovic, L. Consistent evolution of owl ontologies. In In Proc. Of ESWC'05, pages 182–197, 2005.
- [17] Flouris,G., Huang,Z., Pan,J., Plexousakis,D., Wache,H. Inconsistencies, negations and changes in ontologies. In Proc. of AAAI'06, 2006.
- [18] Donini, F. M., Lenzerini, M., Nardi, D., Schaerf, A. Deduction in concept languages: From subsumption to instance checking. Journal of Logic and Computation, 4:1-30, 1994.
- [19] Schlobach,S., Cornet, R. Non-standard reasoning services for the debugging of description logic terminologies. In Georg Gottlob and Toby Walsh, editors, IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, pages 355–362. Morgan Kaufmann, 2003.
- [20] Borgida, A., Serafini, L. Distributed description logics: Assimilating information from peer sources. Journal on Data Semantics, 2003
- [21] Meilicke, C., Stuckenschmidt, H. An Efficient Method for Computing Alignment Diagnoses. Proceedings of the Third International Conference on Web Reasoning and Rule Systems (RR-09), Chantilly, Virginia, USA, 2009
- [22] FermeE., Hansson,S. AGM 25 years, J. Philos. Log. 40 (2011) 295–331.
- [23] Peppas,P. Belief revision. In Handbook of Knowledge Representation, pages 317–359. Elsevier, 2008
- [24] Hansson, S. O. Kernel contraction. Journal of Symbolic Logic, Vol 59, pp.845- 859. 1994
- [25] Hansson, S. O. Semi-revision. Journal of Applied Non-Classical Logic, 7(1-2), pp.151-175. 1997
- [26] Hansson, S. O., Wassermann,R. Local change. Studia Logica, 70(1):49–76, 2002.
- [27] Baader,F., Penaloza,R., Suntisrivaraporn,B. Pinpointing in the description logic EL+. In Proc .KI 2007, Springer LNAI 4667, 2007
- [28] Horridge, M., Bechhofer,S. The OWL API: A Java API for Working with OWL 2 Ontologies. OWLED 2009, 6th OWL Experienced and Directions Workshop, Chantilly, Virginia, October 2009.
- [29] Euzenat, J. An API for ontology alignment. In:McIlraith, S.A., Plexousakis, D., van Harmelen,F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 698–712. Springer, Heidelberg 2004
- [30] Sirin, E., Parsia, B., Grau, B.C, Kalyanpur, A., Katz, Y. Pellet: A practical OWL-DL reasoner. Journal of Web Semantics, 5(2), 2007

- [31] Zahaf, A. Alignment between versions of the same ontology. In: Proc. 4th International Conference on Web and Information Technologies, pp.318-323.
- [32] Yahia Atig, Ahmed Zahaf, Djelloul Bouchiha. "Conservativity Principle Violations for Ontology Alignment: Survey and Trends", International Journal of Information Technology and Computer Science (IJITCS), Vol.8, No.7, pp.61-71, 2016. DOI: 10.5815/ijitcs.2016.07.09
- [33] Wang, P., Xu, B., Lily: Ontology alignment results for OAEI 2008. Proceedings for the Third International Workshop on Ontology Matching, October 26, 2008, Karlsruhe, Germany
- [34] Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R. Ontology matching with semantic verification. Journal of Web Semantics 7(3), 235–251 2009
- [35] Ngo, D., Bellahsene, Z. YAM++: (not) Yet Another Matcher for Ontology Matching Task. In BDA 2012
- [36] Meilicke, C., Stuckenschmidt, H. (2007): Applying logical constraints to ontology matching. In Proc. of the 30th German Conference on Artificial Intelligence, Osnabrück, Germany, 2007.
- [37] Meilicke, C., Tamin, A., Stuckenschmidt, H. (2007): Repairing ontology mappings. In Proc. of the Twenty-Second Conference on Artificial Intelligence, Vancouver, Canada, 2007.

Authors' Profiles



Ahmed ZAHAF received his Engineer degree in computer science from Oran University, Algeria, in 1994, and M. Sc. in computer science from Sidi Bel Abbes University, Algeria, in 2005. Currently, he is a lecturer at Dr Tahar Moulay, University of Saida, Algeria. His research interests

include semantic web, Linked data, ontology engineering, knowledge management and information systems



Mimoun MALKI is graduated with Engineer Degree in Computer Science from National Institute of Computer Science, Algiers, in 1983. He received the MS degree and the PhD degree in Computer Science from the University of Sidi Bel-Abbes, Algeria, in 1992 and 2002, respectively. He was an Associate professor

in the Department of Computer Science at the University of Sidi Bel-Abbes from 2003-2010. Since 2011, he is a full Professor at Djillali Liabes University of Sidi Bel-Abbes. Currently, he is a full professor at the Ecole Supérieure en Informatique of Sidi Bel-Abbes. He is the head of the Evolutionary Engineering and Distributed Information Systems Laboratory. His research interests include Databases, Information Systems Interoperability, Ontology Engineering, Web-based Information Systems, Semantic Web Services, Linked Data Services, Web Reengineering, Enterprise Mashup and Cloud Computing.

How to cite this paper: Ahmed ZAHAF, Mimoun MALKI, "Kernel Contraction and Consolidation of Alignment under Ontology Change", International Journal of Information