Reasoning with Instances of Heterogeneous Ontologies*

Luciano Serafini and Andrei Tamilin

Data & Knowledge Management Group
Foundation Bruno Kessler - IRST
Via Sommarive 18, 38100 Povo di Trento, Italy

Abstract. We address the problem of reasoning with instances of heterogeneously formalized ontologies. Given a set of semantic mappings, reconciling conceptual and instance level heterogeneity between the input ontologies, we build our approach upon the capability of mappings to enforce a propagation of concept membership assertions between ontologies. The approach is formally grounded on a distributed description logic framework, which formally encodes ontologies as description logic knowledge bases and mappings as bridge rules and individual correspondences. We first give a logical characterization to the propagation of concept membership assertions along bridge rules and individual correspondences between the input $\mathcal{SHIQ}$-ontologies, and further define a sound and complete tableau algorithm for capturing such a propagation.

1 Motivation and Approach

Ontology *heterogeneity* is one of the crucial problems to be solved on the semantic web. To sustain this claim it is enough to give a glance on the actual situation on the web – different ontologies although representing the same or largely overlapping domains do it in different, heterogeneous ways.

The state of the art approaches to the problem of reconciling heterogeneity between ontologies are built upon the utilization of *semantic mappings*. Roughly, a mapping comprises relations between semantically related elements of different ontologies. For example, a mapping can express the fact that the concept Automobile in one ontology is semantically equivalent to the concept Car in another ontology, or that the instance ferrary_enzo in one ontology semantically corresponds to the instance f60 in the other ontology. To discover mappings a number of (semi-)automated techniques and tools can be applied; we refer the reader to the comprehensive overview of the state of the art by Euzenat and Shvaiko in [5].

Once mappings are stated, it is necessary to provide a method for *reasoning* with them. Formally, this amounts to evaluating logical consequences of mappings on the mapped ontologies. Mappings form a source to a target ontology can be used to transfer knowledge between the two ontologies. Due to the formal correspondences of ontologies to DL knowledge bases, there are two types of knowledge that can be transferred: terminological knowledge (i.e., mappings can force new concept subsumption axioms

* This is a revised version of the paper "Instance Migration over Heterogeneous Ontology Environments" accepted for presentation at ISWC2007

in the target ontology) and assertional knowledge (i.e., mappings can force new instance assertions to concepts in the target ontology).

The main objective of the paper is to provide a logical characterization of the assertional information enforced by a set of mappings and on the base of this characterization enable reasoning with individuals of mapped ontologies. Our approach relies on the logical framework of distributed description logics (DDL) introduced by Borgida and Serafini in [3]. In such a framework a *distributed knowledge base* consists of a family of standard DL knowledge bases, corresponding to each given ontology, a set of *bridge rules*, corresponding to mapping between pairs of terminologies (T-boxes), and *individual correspondences*, corresponding to mapping between pairs of elements in instance storages (A-boxes).

This work presents: (1) the logical characterization of the capability of bridge rules and individual correspondences to propagate concept membership assertions across mapped ontologies and its affection on reasoning with instances of the ontologies; (2) the overview of a sound and complete tableau algorithm for reasoning with instances of $\mathcal{SHIQ}$-ontologies, built as an extension to the classical $\mathcal{SHIQ}$-A-box tableau [10] with the backward chaining strategy for computation of propagated concept membership assertions; (3) the outline of the practical implementation of the A-box reasoning algorithm in a distributed DDL Reasoner DRAGO.

The paper is organized as follows. In Section 2 we recall a definition of DDL's distributed knowledge base with bridge rules and individual mappings. In Section 3 we investigate reasoning with instances in distributed knowledge bases; we start with analysis of knowledge propagation along bridge rules and individual correspondences and further introduce in Section 4 a tableau reasoning algorithm capturing it. We end up with an overview of related work and concluding remarks.

2 Distributed Knowledge Bases

Given a setting of multiple ontologies interconnected by directed semantic mappings, the distributed description logics allows to formally encode it in terms of a distributed knowledge base. Following the original definitions of Borgida and Serafini in [3], in this section we recall the basics of distributed knowledge bases and reasoning tasks available for them.

2.1 Syntax and Semantics

The first component of a distributed knowledge base is a family of knowledge bases $\mathcal{K} = \{\mathcal{K}_i\}_{i \in I}$. According to a standard DL definitions, each $\mathcal{K}_i$ consists of a terminological component $\mathcal{T}_i$ (T-box) and an assertional component $\mathcal{A}_i$ (A-box). Since the very same symbol can be used in two knowledge bases with different meaning, to unambiguously refer to elements of $\mathcal{K}_i$, they are prefixed with the index i of the knowledge base. The notations $i{:}\, a$ $i{:}\, C$, $i{:}\, C \sqsubseteq D$, $i{:}\, C(a)$ and $i{:}\, R(a, b)$, stand for an individual a, concept C, subsumption $C \sqsubseteq D$, assertions $C(a)$ and $R(a, b)$, respectively in the knowledge base $\mathcal{K}_i$.

Mappings from $\mathcal{K}_i$ to $\mathcal{K}_j$ ($i \neq j$) are encoded as sets of bridge rules

– $i{:}C \xrightarrow{\sqsubseteq} j{:}D$ (into-bridge rule)

– $i{:}C \xrightarrow{\sqsupseteq} j{:}D$ (onto-bridge rule)

and individual correspondences

– $i{:}a \longmapsto j{:}b$ (individual correspondence)

where C and D are concept names of $\mathcal{T}_i$ and $\mathcal{T}_j$, and a and b are individuals of $\mathcal{A}_i$ and $\mathcal{A}_j$ respectively[1].

Both bridge rules and individual correspondences from $\mathcal{K}_i$ to $\mathcal{K}_j$ express a subjective possibility of $\mathcal{K}_j$ to translate some of the concepts and individuals of $\mathcal{K}_i$ into its local concepts and individuals. For example, the following mapping between two ontologies describing the domain of cars

$$i{:}\mathsf{Transmission} \xrightarrow{\sqsubseteq} j{:}\mathsf{Gearbox} \tag{1}$$

$$i{:}\mathsf{Motor} \xrightarrow{\sqsupseteq} j{:}\mathsf{V\_Engine} \tag{2}$$

$$i{:}\mathsf{sequential\_manual\_transmission} \longmapsto j{:}\mathsf{f1\_gearbox} \tag{3}$$

can be given with the following intuitive reading: from $\mathcal{K}_j$'s point of view, i's concept Transmission is more specific than its local concept Gearbox, i's concept Motor is conversely more general than its local concept V_Engine, and finally that i's individual sequential_manual_transmission can be translated into its local individual f1_gearbox. Note that in the general case, DDL admits that an individual can have more than one translation.

A *distributed T-box* $\mathfrak{T}$ consists of T-boxes $\mathcal{T}_i$ and a collection $\mathfrak{B}$ of bridge rules between them. A *distributed A-box* $\mathfrak{A}$ consists of A-boxes $\mathcal{A}_i$ and a collection of individual correspondences $\mathfrak{C}$. A *distributed knowledge base* $\mathfrak{K}$ is then a tuple $\langle \mathfrak{T}, \mathfrak{A} \rangle$.

The semantics of DDL is defined with a fundamental assumption that each knowledge base $\mathcal{K}_i$ in the family is *locally interpreted* on its *local interpretation domain*. To support directionality, (i.e., mappings from i to j only propagate in the i-to-j-direction), we admit the hole interpretation $\mathcal{I}_\epsilon$ with empty domain (see more details in [12])[2]. By definition, we impose that $\mathcal{I}_\epsilon$ satisfies any knowledge base.

A *distributed interpretation* $\mathfrak{I}$ of a distributed knowledge base $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{A} \rangle$ consists of a family of local interpretations $\mathcal{I}_i$ on local interpretation domains $\Delta^{\mathcal{I}_i}$ and a family of domain relations $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ between pairs of local domains. Domain relation r_{ij} is defined to denote $\{d' \in \Delta^{\mathcal{I}_j} \mid \langle d, d' \rangle \in r_{ij}\}$.

A distributed interpretation $\mathfrak{I}$ *satisfies* a distributed knowledge base $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{A} \rangle$, is called a model of $\mathfrak{K}$, if all its' components are satisfied according to the following rules:

– $\mathcal{I}_i$ satisfies $\mathcal{K}_i$

[1] In this work we concentrate only on individual correspondences, and don't consider complete correspondences as introduced in [3].

[2] Classically, DL interpretation maps every individual into an *element* of the domain, while the hole maps everything into the empty *set*. To allow homogeneous treatment of standard DL interpretations and holes, we require that any individual x is standardly interpreted into a singleton set, rather than into an element of the domain. Hence, $\mathcal{I}_i \models C(a) \Longleftrightarrow a^{\mathcal{I}_i} \subseteq C^{\mathcal{I}_i}$, rather than $a^{\mathcal{I}_i} \in C^{\mathcal{I}_i}$.

– $r_{ij}(C^{\mathcal{I}_i}) \supseteq D^{\mathcal{I}_j}$ for all $i{:}C \xrightarrow{\sqsupseteq} j{:}D$
– $r_{ij}(C^{\mathcal{I}_i}) \subseteq D^{\mathcal{I}_j}$ for all $i{:}C \xrightarrow{\sqsubseteq} j{:}D$
– $b^{\mathcal{I}_j} \subseteq r_{ij}(a^{\mathcal{I}_i})$ for all $i{:}a \longmapsto j{:}b$

2.2 Distributed inference services

Although both in DL and DDL the fundamental reasoning services include checking concept subsumption and instance checking within a certain ontology, in DDL, besides the ontology itself, the other related by mappings ontologies should be taken into account. Given a distributed knowledge base $\mathfrak{K} = \langle \mathfrak{T}, \mathfrak{A} \rangle$, DDL defines the following *distributed inference services*:

Subsumption: A concept C is *subsumed* by a concept D in i with respect to $\mathfrak{K}$ if for every distributed interpretation $\mathfrak{I}$ of $\mathfrak{K}$ we have that $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$. In this case we will write $\mathfrak{K} \models i{:}C \sqsubseteq D$.

Instantiation: An individual a is an *instance of* a concept C in i with respect to $\mathfrak{K}$ if for every distributed interpretation $\mathfrak{I}$ of $\mathfrak{K}$ we have that $a^{\mathcal{I}_i} \subseteq C^{\mathcal{I}_i}$. In this case we will write $\mathfrak{K} \models i{:}C(a)$.

Subsumption service is typically called a terminological reasoning service, while instantiation service is called assertional reasoning service. So far, the task of terminological reasoning has been undisclosed in [12]. It has been shown that certain combinations of into- and onto-bridge rules can lead to the propagation of knowledge in form of subsumption axioms across ontologies participating in DDL. Moreover, in case of DDL with $\mathcal{SHIQ}$ components without instances adding these additional propagation rules to existing DL tableaux algorithms leads to a correct and complete reasoning in DDL. In the following sections we close the gap by addressing the question of assertional reasoning in DDL.

3 Characterization of Reasoning with Instances

For the sake of clarity, we start considering the case of DDL with two component knowledge bases and unidirectional sets of bridge rules and individual correspondences. The general results and proofs can be found in the technical report [13].

3.1 Inference patterns

In the following we characterize the knowledge propagated from a knowledge base i (the source) to j (the target) by a set of *propagation rules* of the form:

$$\frac{(1)\text{ facts in } i, \quad (2)\text{ bridge rules from } i \text{ to } j, \quad (3)\text{ individual mappings from } i \text{ to } j}{(4)\text{ fact in } j}$$

which must be read as: if the facts in (1) are true in $\mathcal{K}_i$, the bridge rules in (2) are contained in $\mathfrak{B}_{ij}$, the individual correspondences in (3) are contained in $\mathfrak{C}_{ij}$, then the fact in (4) must be true in $\mathcal{K}_j$.

Following the semantics of mappings in DDL outlined in the previous section, it can be observed that the individual correspondences can interact with into-bridge rules with the effect of propagating concept membership assertions:

$$\frac{i{:}\,C(a), \quad i{:}\,C \xrightarrow{\sqsubseteq} j{:}\,D, \quad i{:}\,a \longmapsto j{:}\,b}{j{:}\,D(b)} \tag{4}$$

Because $b^{\mathcal{I}_j} \subseteq r_{ij}(a^{\mathcal{I}_i}) \subseteq r_{ij}(C^{\mathcal{I}_i}) \subseteq D^{\mathcal{I}_j}$, we indeed have that $\mathfrak{I} \models j{:}\,D(b)$.

In languages that support disjunction, the above propagation can be generalized to the propagation of concept membership assertions over a disjunction of $n \geqslant 0$ concepts:

$$\frac{i{:}\,(C_1 \sqcup \ldots \sqcup C_n)(a), \quad i{:}\,C_k \xrightarrow{\sqsubseteq} j{:}\,D_k \ (1 \leqslant k \leqslant n), \quad i{:}\,a \longmapsto j{:}\,b}{j{:}\,(D_1 \sqcup \ldots \sqcup D_n)(b)} \tag{5}$$

Rule (5) appears to be the *most general* form of assertion propagation in DDL when individual correspondences are restricted to be *functional*. A set of individual correspondences $\mathfrak{C}_{ij}$ is functional if for every individual a of $\mathcal{A}_i$ the set $\mathfrak{C}_{ij}$ contains at most one individual correspondence $i{:}\,a \longmapsto j{:}\,b$. For the sake of presentation, in this paper we restrict ourself to functional individual correspondences, leaving the most general case to the technical report [13])[3].

It is also important to note, that when $n = 0$, the inference pattern in (5) becomes the following inference rule:

$$\frac{i{:}\,\bot(a), \quad i{:}\,a \longmapsto j{:}\,b}{j{:}\,\bot(b)} \tag{6}$$

which states that to propagate the inconsistency of $\mathcal{K}_i$ to $\mathcal{K}_j$ it's enough to have one single individual correspondence. From the representational point of view this inference rule is very fragile. We currently do not see an easy solution to fix this sensitivity to inconsistency propagation. This topic will be subject for further studies.

3.2 Soundness and completeness

To demonstrate the correctness and completeness of the inference pattern presented in Section 3.1, we follow the approach similar to the one taken in [12]. The main idea consists in construction of an operator which essentially applies the generalized inference pattern (5) to extend knowledge bases with new assertions induced by mappings.

Given a set of bridge rules $\mathfrak{B}_{12}$ and set of individual correspondences $\mathfrak{C}_{12}$ from $\mathcal{K}_1$ to $\mathcal{K}_2$, the *individual correspondence operator* $\mathfrak{C}_{12}(\cdot)$, taking as input a knowledge base

[3] To give an intuition of the effect of non functional individual mappings, consider the case in which there are two into-bridge rules $i : C_1 \xrightarrow{\sqsubseteq} j : D_1$ and $i : C_2 \xrightarrow{\sqsubseteq} j : D_2$ and, the non functional set of individual mappings $\{i : a \longmapsto j : b, \ \ i : a \longmapsto j : c\}$. Then the fact that $\mathcal{K}_i \models C_1 \sqcup C_2(a)$ entails the disjunctive assertion $(D_1(b) \wedge D_1(c)) \vee (D_2(b) \wedge D_2(c))$. This implies that the general case requires technicalities for disjunctive A-boxes.

$\mathcal{K}_1$ and producing an A-box of $\mathcal{K}_2$, is defined as follows:

$$\mathfrak{C}_{12}(\mathcal{K}_1) = \left\{ (D_1 \sqcup \ldots \sqcup D_n)(b) \;\middle|\; \begin{array}{l} \mathcal{K}_1 \models (C_1 \sqcup \ldots \sqcup C_n)(a) \\ 1{:}\, C_k \xrightarrow{\sqsubseteq} 2{:}\, D_k \in \mathfrak{B}_{12}\ (1 \leqslant k \leqslant n) \\ 1{:}\, a \;\longmapsto 2{:}\, b \;\; \in \mathfrak{C}_{12} \end{array} \right\}$$

It is remarkable that *onto*-bridge rules do not affect instance propagation. The reason is that onto-bridge rules impose only existence of preimages of objects that already exists in the target ontology. Into-bridge rules, instead, constraint the individual mappings to be defined whithin a certain range. The individual correspondence operator formalizes the assertional knowledge that is propagated across ontologies.

The characterization of the propagation of the terminological knowledge is characterized by an analogous operator, called *bridge operator*, introduced in [12] and defined as follows: $\mathfrak{B}_{12}(\cdot)$, taking as input a knowledge base $\mathcal{K}_1$ and producing a T-box of $\mathcal{K}_2$:

$$\mathfrak{B}_{12}(\mathcal{K}_1) = \left\{ B \sqsubseteq D_1 \sqcup \ldots \sqcup D_n \;\middle|\; \begin{array}{l} \mathcal{T}_1 \models A \sqsubseteq C_1 \sqcup \ldots \sqcup C_n \\ 1{:}\, C_k \xrightarrow{\sqsubseteq} 2{:}\, D_k \in \mathfrak{B}_{12}\ (1 \leqslant k \leqslant n) \\ 1{:}\, A \;\xrightarrow{\sqsupseteq} 2{:}\, B \;\; \in \mathfrak{B}_{12} \end{array} \right\}$$

With the remarkable exception of inconsistency propagation—by rule (6)—the individual correspondences do not affect the propagation of terminological knowledge. The inferences formalized by the two operators described above *completely* describe the possible propagations that are forced by a set of bridge rules and individual correspondences. This is formally stated in the following theorem.

Theorem 1 (Soundness and completeness). *Let $\mathfrak{K}_{12}$ be a distributed knowledge base consisting of $\mathcal{K}_1$, $\mathcal{K}_2$ $\mathcal{SHIQ}$ knowledge bases, and $\mathfrak{B}_{12}$, $\mathfrak{C}_{12}$ mappings between them. For any statement ϕ (of the form $C \sqsubseteq D$ or $C(a)$) in the language of $\mathcal{K}_2$*

$$\mathfrak{K}_{12} \models 2 : \phi \quad \Longleftrightarrow \quad \langle \mathcal{T}_2 \cup \mathfrak{B}_{12}(\mathcal{K}_1),\ \mathcal{A}_2 \cup \mathfrak{C}_{12}(\mathcal{K}_1) \rangle \models \phi$$

The proof of the generalization of the Theorem 1 is fully described in the technical report. Some remarks are necessary.

Independence between terminological and assertional propagation From the characterization above one can see that propagation of terminological and assertional knowledge are orthogonal. The two effects can be computed independently in parallel. What is more important, however, is that the change of the A-box does not affect the propagation of the terminological knowledge. This means that if the source T-box does not change the terminological propagation is computed once for all.

Local propagation of assertional knowledge Assertional propagation operator ensures, if a change of the source A-box involves only the set of individuals $\{a_1, \ldots, a_n\}$, then assertional propagation must be computed only for the portion of the target A-box $\mathcal{A}_2$ concerning the set of individuals $\{b \mid 1 : a_i \longmapsto 2 : b \in \mathfrak{C}_{12}\}$.

Upper bound and complexity If the mapping from 1 to 2 is finite and contains m into-bridge rules, n onto-bridge rules, and o individual correspondences, then the

bridge operator $\mathfrak{B}_{12}$ generates at most $n * 2^m$ subsumption statements, and the individual operator $\mathfrak{C}_{12}$ generates at most $o * 2^m$ instance membership statements. Since the propagation of statements needs checking subsumption and instantiation in the source knowledge base, which is EXPTIME complete, we have that computing subsumption and instantiation in a distributed setting is EXPTIME complete in the dimension of the source knowledge base plus mappings.

Vanilla implementation The above theorem supports a vanilla implementation of *forward chaining* inference engine for DDL. The implementation consists of three steps: computation of propagation operators $\mathfrak{B}_{12}(\mathcal{K}_1)$ and $\mathfrak{C}_{12}(\mathcal{K}_1)$, construction of extended version of knowledge base $\mathcal{K}_2$ as $\langle \mathcal{T}_2 \cup \mathfrak{B}_{12}(\mathcal{K}_1),\ \mathcal{A}_2 \cup \mathfrak{C}_{12}(\mathcal{K}_1)\rangle$, and finally applying to this knowledge base one of existing DL reasoners, such as FaCT++ [15], Racer [7], or Pellet [14].

The vanilla approach to reasoning has a strong advantage of reuse of existing highly optimized DL reasoners, however it can be very costly for situations when semantic mappings are changing dynamically or when the number of reasoning questions to be verified is relatively small. In the next section, we propose an alternative, *backward chaining* approach to reasoning, which does "lazy" computation of propagated axioms and hence better fits to instable and short-living distributed environments.

4 Distributed $\mathcal{SHIQ}$-A-box Tableaux Algorithm

In this section we present a distributed tableaux algorithm for reasoning with instances in DDL. Our design idea consists in constructing a network of standard DL tableaux, one for each ontology, which communicate via mappings in a backward fashion.

Since we restricted the expressivity of ontologies participating in DDL to $\mathcal{SHIQ}$ DL, we will consider in the following that ontologies $\mathcal{K}_1$ and $\mathcal{K}_2$ from a distributed knowledge base $\mathfrak{K}_{12} = \langle \mathfrak{T}_{12}, \mathfrak{A}_{12}\rangle$ are attached with $\mathcal{SHIQ}$-tableau reasoning procedures $\mathbf{Tab}_1$ and $\mathbf{Tab}_2$ [10]. Due to the reduction of reasoning with concepts to reasoning with instances [2], we suppose that each procedure $\mathbf{Tab}_i(\alpha)$ can check the satisfiability of any statement α of form $i{:}\, C \sqsubseteq D$, $i{:}\, C(a)$.

As described in [10], the $\mathcal{SHIQ}$-tableau works on a so called "completion forest", a collection of trees whose root nodes correspond to instances in A-box. Given a knowledge base, the algorithm initializes a completion forest $\mathcal{F}$ with a set of root nodes $\mathbf{x}_0 = \{x_0^k\}$ corresponding to a set of instances b_k in A-box, labels each x_0^k with a set $\mathcal{L}(x_0^k)$ of concepts C for each concept assertion $C(b_k)$ in A-box, and finally draws an edge between x_0^k and x_0^m for each role assertion $R(h_k, h_m)$ in A-box. After that, the set of $\mathcal{SHIQ}$ completion rules expanding the forest $\mathcal{F}$ is applied. The fully expanded forest then represents a model of the knowledge base. To test entailment of arbitrary assertion $X(a)$, $\neg X(a)$ is added to A-box and further the tableau is expanded to see whether a model of such knowledge base can be constructed or not.

To accommodate the knowledge propagation from $\mathcal{K}_1$ to $\mathcal{K}_2$ in $\mathfrak{K}_{12}$, we intervene in the completion process of $\mathbf{Tab}_2$ in order to capture new facts induced by bridge rules and individual correspondences. Hence, we get a *distributed tableaux procedure* $\mathbf{DTab}_2$ which extends $\mathbf{Tab}_2$ with two additional expansion rules:

$\mathfrak{C}_{12}$-rule:

if 1. $x \in \mathbf{x}_0$, such that $x = b^{\mathcal{T}_2}$ and $1{:}\, a \longmapsto 2{:}\, b$,
$\mathbf{H} \subseteq \{H_k \mid 1{:}\, B_k \xrightarrow{\sqsubseteq} 2{:}\, H_k \in \mathfrak{B}_{12}\}$,
$\mathbf{B} = \{B_k \mid H_k \in \mathbf{H}, 1{:}\, B_k \xrightarrow{\sqsubseteq} 2{:}\, H_k \in \mathfrak{B}_{12}\}$,
2. $\mathbf{Tab}_1((\bigsqcup \mathbf{B})(a)) = \mathit{true}$ for $\bigsqcup \mathbf{H} \notin \mathcal{L}(x)$,
then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{\bigsqcup \mathbf{H}\}$

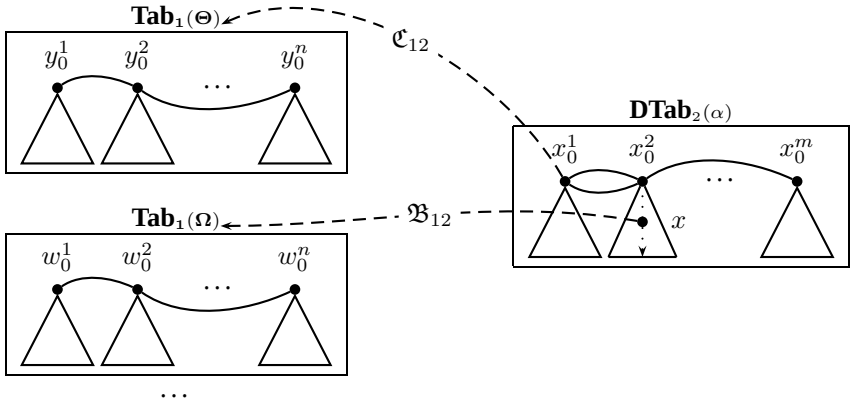
$\mathfrak{B}_{12}$-rule:

if 1. $G \in \mathcal{L}(x)$, such that $1{:}\, A \xrightarrow{\sqsupseteq} 2{:}\, G \in \mathfrak{B}_{12}$,
$\mathbf{H} \subseteq \{H_k \mid 1{:}\, B_k \xrightarrow{\sqsubseteq} 2{:}\, H_k \in \mathfrak{B}_{12}\}$,
$\mathbf{B} = \{B_k \mid H_k \in \mathbf{H}, 1{:}\, B_k \xrightarrow{\sqsubseteq} 2{:}\, H_k \in \mathfrak{B}_{12}\}$,
2. $\mathbf{Tab}_1(A \sqsubseteq \bigsqcup \mathbf{B}) = \mathit{true}$ for $\bigsqcup \mathbf{H} \notin \mathcal{L}(x)$,
then $\mathcal{L}(x) \longrightarrow \mathcal{L}(x) \cup \{\bigsqcup \mathbf{H}\}$

The principle idea of these additional expansion rules consists in implementing backward versions of bridge and individual correspondences operators introduced in Section 3.2. According to rule $\mathfrak{C}_{12}$, if **DTab**$_2$ encounters a root node x connected by an individual correspondence, then a disjunction of concepts $\bigsqcup \mathbf{H}$ should be added to the label $\mathcal{L}(x)$ if $\bigsqcup \mathbf{H}(x)$ is entailed by interaction of individual correspondence with onto-rules. To determine this entailment, **DTab**$_2$ remotely requests foreign **Tab**$_1$ to check if it is the case that $\bigsqcup \mathbf{B}(b)$ in $\mathcal{K}_1$.

The role of $\mathfrak{B}_{12}$-rule is to analyse the nodes of completion forest and import consequences of subsumption propagations. If **DTab**$_2$ encounters a node x which contains a label G connected by an onto-bridge rule, then if $G \sqsubseteq \bigsqcup \mathbf{H}$ is entailed by the bridge rules, the label $\bigsqcup \mathbf{H}$ is added to x. While in order to determine the entailment, **DTab**$_2$ invokes the procedure **Tab**$_1$ with a question whether a subsumption $A \sqsubseteq \bigsqcup \mathbf{B}$ holds in $\mathcal{K}_1$.

The distributed execution of **DTab**$_2$ can be intuitively depicted as follows:



Theorem 2 (Termination, Soundness, Completeness). *Given $\mathcal{SHIQ}$ DL knowledge bases $\mathcal{K}_1$ and $\mathcal{K}_2$, let $\mathfrak{K}_{12} = \langle\langle\{\mathcal{T}_1, \mathcal{T}_2\}, \mathfrak{B}_{12}\rangle, \langle\{\mathcal{A}_1, \mathcal{A}_2\}, \mathfrak{C}_{12}\rangle\rangle$ be a distributed knowledge base. Then, given a $\mathcal{SHIQ}$ statement α*

1. *a distributed procedure* **DTab**$_2(\alpha)$ *terminates, and*

2. *α is satisfiable in $\mathcal{K}_2$ with respect to $\mathfrak{R}_{12}$ if and only if* $\mathbf{DTab}_2(\alpha)$ *yields a complete and clash-free completion forest.*

It can be shown that the proposed algorithm enjoys generalization to arbitrary number of $\mathcal{SHIQ}$ knowledge bases participating in DDL, and moreover can be extended to distributed knowledge bases containing cyclical pathes of bridge rules and individual correspondences. For the sake of clarity, we omit the discussion of these generalizations and refer the reader to the technical report [13] for details.

Note that due to the remark to Theorem 1 on independence of terminological and assertional propagation, the implementation of the tableaux introduced in this section can be constructed on top of existing implementation of DRAGO DDL Reasoner by reusing the implementation of bridge completion rule and adding additionally the individual completion rule as described in the present algorithm.

5 Related Work

The importance of resolving heterogeneity problem on the web pushes the big research efforts to devising frameworks capable of representing and reasoning with multiple ontologies interrelated by semantic mappings. While DL is already the standard for working with web ontologies, the question of formal representations and reasoning with mappings is still a subject to the standardization.

In *SomeWhere* [6], the authors target a question of decentralized approach to querying heterogeneous ontologies. Mappings in *SomeWhere* has a form of a subsumption statements and the reasoning is based on rewriting techniques for combining reasoning over heterogeneous ontologies. The big advantage of the presented approach is its scalability, while the disadvantage is its limitation to a "propositional" ontologies, containing only disjunction, conjunction and negation.

Another recent example of decentralized infrastructure for querying distributed ontologies is *KAONp2p* [8, 9]. The authors adopt the approach of [4] to express mappings as correspondences between conjunctive queries over ontologies. The querying further requires the terminologies and mapping to be merged into a single global ontology, while instance data is then retrieved from distributed instance storages.

The recent study of query answering in *distributed description logics* has been proposed in [1]. The main idea consist in constructing a closure ontology by forward propagating, via DDL mappings, relevant axioms contained in other mapped ontologies (in a vein of vanilla implementation of DDL reasoner discussed in the current study). Doing so, further enables reformulation of distributed query answering problem into local query answering. Although the approach of [1] is sound, the authors point out the incompleteness of their study.

Another important framework is *$\mathcal{E}$-connections* [11]. Original purpose of $\mathcal{E}$-connections is to aggregate ontologies that model different (non-overlapping) aspects of the world, rather then integrate those overlapping as in DDL. Nonetheless, it has been shown in [11] that mathematically DDL constructs can be simulated in $\mathcal{E}$-connections, however sacrificing the directionality of knowledge propagation. Another difference concerns with reasoning approach. In contrast to distributed coordinating tableaux in DDL, in $\mathcal{E}$-connections a global tableau, both theoretically and practically, needs to be constructed.

6 Conclusion

In the present study, we investigated a task of correct and complete reasoning with instances over heterogeneous ontologies. We formally grounded our approach on DDL framework. Theoretically, we formalized inferences with instances and defined the distributed tableaux algorithm for reasoning with multiple $\mathcal{SHIQ}$ DL ontologies. Practically, we extended terminological reasoning services available in the DRAGO DDL Reasoner with the support of assertional reasoning tasks.

References

1. F. Alkhateeb and A. Zimmermann. Query Answering in Distributed Description Logics. In *Proc. of the 1st Conference on New Technologies, Mobility and Security (NTMS)*, 2007.
2. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. 2003.
3. A. Borgida and L. Serafini. Distributed Description Logics: Assimilating Information from Peer Sources. *Journal of Data Semantics*, 1:153–184, 2003.
4. D. Calvanese, G. De Giacomo, and M. Lenzerini. A Framework for Ontology Integration. In *Proc. of the Semantic Web Working Symposium (SWWS-2001)*, pages 303–316, 2001.
5. J. Euzenat and P. Shvaiko, editors. *Ontology Matching*. Springer Verlag, 2007.
6. F. Goasdoué and M-C. Rousset. Querying Distributed Data through Distributed Ontologies: a Simple but Scalable Approach. *IEEE Intelligent Systems*, 18(5):60–65, 2003.
7. V. Haarslev and R. Moller. RACER System Description. In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR-2001)*, pages 701–706, 2001.
8. P. Haase and B. Motik. A Mapping System for the Integration of OWL-DL Ontologies. In *Proceedings of the First International Workshop on Interoperability of Heterogeneous Information Systems (IHIS 05)*, pages 9–16. ACM Press, 2005.
9. P. Haase and Y. Wang. A Decentralized Infrastructure for Query Answering over Distributed Ontologies. In *Proceedings of the 22nd Annual ACM Symposium on Applied Computing (SAC-2007)*, 2007.
10. I. Horrocks, U. Sattler, and S. Tobies. Reasoning with Individuals for the Description Logic SHIQ. In *Proceedings of the 17th International Conference on Automated Deduction (CADE-2000)*, pages 482–496, 2000.
11. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-Connections of Abstract Description Systems. *Artificial Intelligence*, 156(1):1–73, 2004.
12. L. Serafini, A. Borgida, and A. Tamilin. Aspects of Distributed and Modular Ontology Reasoning. In *Proc. of the 19th Joint Conference on Artificial Intelligence (IJCAI)*, 2005.
13. L. Serafini and A. Tamilin. Reasoning with Instances in Distributed Description Logics. Technical report, Fondazione Bruno Kessler - IRST, 2007. `http://sra.itc.it/people/tamilin/publications/2007/swap/tr.pdf`.
14. E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz. Pellet: A Practical OWL-DL Reasoner. *Journal of Web Semantics*, 2006.
15. D. Tsarkov and I. Horrocks. FaCT++ Description Logic Reasoner: System Description. In *Proc. of the International Joint Conference on Automated Reasoning (IJCAR)*, volume 4130, pages 292–297, 2006.