Local tableaux for reasoning in distributed description logics *

Luciano Serafini
ITC-IRST, 38050 Povo, Trento, Italy
luciano.serafini@itc.it

Andrei Tamilin
DIT - University of Trento, 38050 Povo, Trento, Italy
andrei.tamilin@dit.unitn.it

Abstract

The last decade of basic research in the area of Description Logics (DL) has created a stable theory, efficient inference procedures, and has demonstrated a wide applicability of DL to knowledge representation and reasoning. The success of DL in the semantic web and the distributed nature of the last one inspired recently a proposal of Distributed DL framework (DDL). DDL is composed of a set of stand alone DLs pairwise interrelated with each other via collection of bridge rules. In this paper, we investigate the reasoning mechanisms in DDL and introduce a tableau-based reasoning algorithm for DDL, built on the top of the state of the art tableaux reasoners for DL. We also describe a first prototype implementation of the proposed algorithm.

1 Introduction

Ontologies have been advocated as the basic tools to support interoperability between distributed applications and web services. The basic idea is that different autonomously developed applications can meaningfully communicate by using a common repository of meaning, i.e. a shared ontology. The optimal solution obviously lies in having a unique worldwide shared ontology describing all possible domains. Unfortunately, this is unachievable in practice. The actual situation in the web is characterized by a proliferation of different ontologies. Each ontology describes a specific domain from different perspectives and at different level of granularity. The initial interoperability problem, therefore, passes from the application level to the ontology level. Though the semantic standardization is far to be reached, the syntactic standardization is almost there, as it is widely accepted that ontologies should be expressed in a language, which is a variation of a descriptive language [6, 8].

Given this situation, one of the challenges in the semantic web is of being able to deal with a large number of overlapping and heterogeneous *local ontologies*. We use the term "local" to stress the fact that each ontology describes a domain of interest from a local and subjective perspective. In this paper, we focus on the problem of

*We thank Chiara Ghidini and Floris Roelofsen for their feedback on this paper.

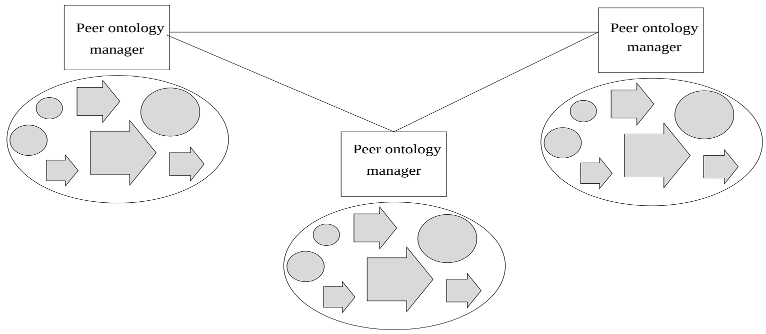

Figure 1: P2P architecture for managing multiple ontologies. In each peer, circles stand for ontologies, and arrows for semantic relations between ontologies.

reasoning within such web of local ontologies. We start form a long tradition of logics for distributes systems, based on propositional Multi-Context Systems [5, 4] and its Local Models Semantics [2], the extension of First Order Logics which leads to Distributed First Order Logics [3], and the extension of Description Logics which leads to Distributed Description Logics (DDL) [1]. Starting from these logical studies, our goal is to propose a *theoretically grounded* and *scalable* solution to the problem of reasoning with a set of distributed, heterogeneous, and overlapping local ontologies. Most of state of the art formalizations of that problem are based on a *global ontology* that allows to uniformly represent a set of local ontologies and semantic relations between them. In these approaches, reasoning in a set of local ontologies is rephrased into a problem of reasoning in the global ontology.

The approaches based on the global ontology, however, present two main drawbacks. First, from a computational complexity point of view it is more convenient to keep the reasoning as much local as possible, exploiting the structure provided by semantic relations for the propagation of reasoning through the local ontologies. Some intuition in this direction can be found in the computational complexity results for satisfiability in Multi-Context Systems described in [11]. Second, the reasoning procedure that has to be implemented in the global ontology should be capable of dealing with the most general local language, whereas having a more distributed approach would allow to apply to every local ontology the specific reasoner, optimized for the local language.

From the architectural point of view, our idea is inspired by peer-to-peer (P2P) distributed knowledge management architectures, proposed in the SWAP [13] and Edamok [12] projects, and by the C-OWL language [14]. We have implemented a P2P architecture, shown in Figure 1, consisting of peer ontology managers, providing reasoning services on a set of local ontologies, and capable of requesting reasoning services to other peers. The ontology manager of a pear p is capable of providing *local* and *global* reasoning services. Local services involve only ontologies local to p, while global services involve both ontologies in p and in other semantically related peers. Among the provided reasoning services, the fundamental ones are checking a local and a global subsumptions.

The paper contributes to the realization of the architecture described above with the following four points: (i) we describe a logical framework (DDL) capable of capturing the behavior of the overall system, i.e. how subsumptions propagate through peers; (ii) give a general reference (and naïve) global algorithm for computing global subsumption, which is sound and complete w.r.t. any topology of the P2P ontology network; (iii) propose a distributed tableau algorithm for computing global subsumption, built as a composition of standard tableaux algorithms for computing local subsumption, which is sound and complete w.r.t. acyclic topology; and finally, (iv) describe a java-based prototype implementing the distributed tableau algorithm.

2 Distributed Description Logics

Distributed description logics (DDL), defined by Borgida and Serafini in [1], is a knowledge representation and reasoning formalism for describing distributed environments composed of a set of distinct description logics interrelated between each other through a set of pairwise inference connectives. In this section we briefly recall the definition of DDL as given in [1].

Before giving the formal definitions of DDL framework let us make several preliminary remarks. Given a non empty set I of indexes, let $\{\mathcal{D}L_i\}_{i \in I}$ be a collection of description logics. Each $\mathcal{D}L_i$ can be one of the logics which is weaker or equivalent to $\mathcal{SHIQ}$ [9] (e.g. $\mathcal{ALC}$, $\mathcal{ALCN}$, $\mathcal{SH}$)[1]. For each $i \in I$ let us denote a T-box of $\mathcal{D}L_i$ as $\mathcal{T}_i$. To distinguish descriptions in each $\mathcal{D}L_i$, we will prefix them with the index of corresponding description logics. E.g. to reflect that any concept C is stated locally in a terminology of $\mathcal{D}L_i$ we will write $i : C$; similarly, to reflect the fact that particular axiom, say $C \sqsubseteq D$, holds locally in a terminology of $\mathcal{D}L_i$ we will write $i : C \sqsubseteq D$.

Bridge rules are used to express semantic relations between different T-boxes.

Definition 2.1 (Bridge rules). A *bridge rule*, from i to j is an expression of the following two forms:

1. $i : x \xrightarrow{\sqsubseteq} j : y$, an *into-bridge rule*;
2. $i : x \xrightarrow{\sqsupseteq} j : y$, an *onto-bridge rule*;

where x and y are either two concepts, or two roles, or two individuals of $\mathcal{D}L_i$ and $\mathcal{D}L_j$ respectively.

In spite of this general definition, in this paper we concentrate on bridge rules between concepts. Intuitively, the into-bridge rule $i : C \xrightarrow{\sqsubseteq} j : D$ states that, from the j-th point of view the concept C in $\mathcal{D}L_i$ is less general than its local concept D. Similarly, the onto-bridge rule $i : C \xrightarrow{\sqsupseteq} j : D$ expresses the fact that, according to j, C in $\mathcal{D}L_i$ is more general than D in $\mathcal{D}L_j$. Therefore, bridge rules from i to j represent the possibility of $\mathcal{D}L_j$ to translate (under some approximation) the foreign concepts of $\mathcal{D}L_i$ into its internal model. Note, that bridge rules are directional and reflect the subjective point of view of particular DL on other DLs surrounding it. Therefore, rules from j to i are not necessarily the inverse of rules from i to j.

Example 2.1. The International Standard Classification of Occupations (ISCO-88)[2] is an ontology that organizes occupations in a hierarchical framework. At the lowest

[1] We assume familiarity with DLs and related tableaux-based systems described in [9].
[2] `http://www.ilo.org/public/english/bureau/stat/class/isco.htm`

```
ISCO-88
2 Professionals
 21 Physical, mathematical and engineering science professionals
  211 Physicists, chemists and related professionals
   2111 Physicists and astronomers
   2114 Geologists and geophysicists
  212 Mathematicians, statisticians and related professionals
   2121 Mathematicians and related professionals
   2122 Statisticians
  213 Computing professionals
   2131 Computer systems designers, analysts and programmers
   2139 Computing professionals not elsewhere classified
  214 Architects, engineers and related professionals
   2141 Architects, town and traffic planners
   2146 Chemical engineers
3 Technicians and associate professionals
 31 Physical and engineering science associate professionals
  311 Physical and engineering science technicians
```

```
WordNet
adEntity
 Causal_agency
  Cause
   Causal_agent
    Entity
     Physical_object
      Object
       Animate_thing
        Living_thing
         Being
          Organism
           Person
            Self
             Grownup
             Nurser
             Engineer
             Worker
```

Figure 2: An extract from ISCO-88 and WordNet.

level is the unit of classification - a job - which is defined as a set of tasks or duties designed to be executed by one person. An extract of ISCO-88 is shown on the left side of Figure 2. A similar, though less detailed, ontology can be extracted from the People sub-hierarchy of WordNet[3]. Notice, that in WordNet there is no hierarchical classification of jobs, as the term "worker" is at the same level than "engineer". If, for whatever reason, one wants to import the ISCO-88 classification into WordNet, an example of bridge rules would be the following:

$$\text{ISCO} : \mathsf{Professionals} \xrightarrow{\sqsubseteq} \text{WNP} : \mathsf{Worker} \tag{1}$$

$$\text{ISCO} : \mathsf{Technicians\_And\_Associate\_Professionals} \xrightarrow{\sqsubseteq} \text{WNP} : \mathsf{Worker} \tag{2}$$

$$\text{ISCO} : \begin{array}{r}\mathsf{Architects\_Engineers\_And\_Related\_Professionals} \sqcup \\ \mathsf{Physical\_And\_Engineering\_Science\_Associate\_Professionals}\end{array} \xrightarrow{\sqsupseteq} \text{WNP} : \mathsf{Engineer} \tag{3}$$

$$\text{ISCO} : \top \xrightarrow{\sqsubseteq} \text{WNP} : \neg\mathsf{Child} \tag{4}$$

$$\text{ISCO} : \mathsf{Doorkeepers\_watchpersons\_and}\ldots \xrightarrow{\sqsupseteq} \text{WNP} : \mathsf{Gatekeeper} \tag{5}$$

Definition 2.2 (Distributed T-box). A *distributed T-box (DTB)* $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I}, \mathfrak{B} \rangle$ consists of a collection of T-boxes $\{\mathcal{T}_i\}_{i \in I}$, and a collection of bridge rules $\mathfrak{B} = \{\mathfrak{B}_{ij}\}_{i \neq j \in I}$ between them.

In order to deal with ontologies which are locally unsatisfiable (this can happen when a set of local axioms are not satisfiable or when bridge rules with other ontologies are not satisfiable) we will introduce two special types of local interpretations, called *holes*.

Definition 2.3 (Holes). A *full hole* in a T-box $\mathcal{T}$ is an interpretation $\mathcal{I}^{\Delta} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}^{\Delta}} \rangle$, where $\Delta^{\mathcal{I}}$ is the original nonempty domain in $\mathcal{T}$, and $\cdot^{\mathcal{I}^{\Delta}}$ is a function that maps every concept expression in $\mathcal{T}$ in the whole $\Delta^{\mathcal{I}}$. An *empty hole* in $\mathcal{T}$ as an interpretation $\mathcal{I}^{\emptyset} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}^{\emptyset}} \rangle$, where $\Delta^{\mathcal{I}}$ is the original nonempty domain $\mathcal{T}$, and $\cdot^{\mathcal{I}^{\emptyset}}$ is a function that maps every concept expression in $\mathcal{T}$ in the empty set.

[3] `http://xmlns.com/wordnet/1.6/Person`

According to the above definition, holes interpret every concept, both atomic and complex ones, either in the empty set or in the universe. The recursive definition of the interpretation of a concept does not apply for holes. One should notice that the interpretation of the concepts $(\neg C)$, denoted as $(\neg C)^{\mathcal{I}_\emptyset}$, is not $\Delta^{\mathcal{I}_\emptyset} \setminus C^{\mathcal{I}_\emptyset} = \Delta^{\mathcal{I}_\emptyset}$, but is $\emptyset$. The consequence of this fact is that $\mathcal{I}_\emptyset \models C \sqsubseteq D$ and $\mathcal{I}_\Delta \models C \sqsubseteq D$ for any pair of concepts C and D. Obviously, since both $\mathcal{I}^\Delta$ and $\mathcal{I}^\emptyset$ satisfy all (even contradictory) concepts in $\mathcal{T}$, they are models of $\mathcal{T}$, i.e. $\mathcal{I}^\Delta \vDash \mathcal{T}$ and $\mathcal{I}^\emptyset \vDash \mathcal{T}$. Holes represent interpretations of locally inconsistent T-boxes.

Definition 2.4 (Domain relation). A *domain relation* r_{ij} from $\Delta^{\mathcal{I}_i}$ to $\Delta^{\mathcal{I}_j}$ is a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$. We use $r_{ij}(d)$ to denote $\{d' \in \Delta^{\mathcal{I}_j} \mid \langle d, d' \rangle \in r_{ij}\}$; for any subset D of $\Delta^{\mathcal{I}_i}$, we use $r_{ij}(D)$ to denote $\bigcup_{d \in D} r_{ij}(d)$; for any $R \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}$ we use $r_{ij}(R)$ to denote $\bigcup_{\langle d, d' \rangle \in R} r_{ij}(d) \times r_{ij}(d')$.

A domain relation r_{ij} represents the capability of $\mathcal{T}_j$ to map the elements of $\Delta^{\mathcal{I}_i}$ into its domain $\Delta^{\mathcal{I}_j}$. For instance if $\mathsf{John} \in \Delta^{\mathcal{I}_1}$ is a person and $\mathsf{J12} \in \Delta^{\mathcal{I}_2}$ is an individual that represents the student John in a specific school, the pair $\langle \mathsf{John}, \mathsf{J12} \rangle$ will be contained in r_{12}. Notice that r_{12} is not necessarily a function. Indeed, John could attend two schools, and therefore, correspond to two individuals in $\Delta^{\mathcal{I}_2}$.

Definition 2.5 (Distributed interpretation). A *distributed interpretation* $\mathfrak{I} = \langle \{\mathcal{I}_i\}_{i \in I}, \{r_{ij}\}_{i \neq j \in I} \rangle$ of distributed T-box $\mathfrak{T}$ consists of local interpretations $\mathcal{I}_i$ on local domains $\Delta^{\mathcal{I}_i}$ for all $\mathcal{T}_i$, and a family of domain relations r_{ij} between these local domains.

Definition 2.6. A distributed interpretation $\mathfrak{I}$ satisfies (written $\mathfrak{I} \vDash_d$) the elements of a DTB $\mathfrak{T}$ according to the following clauses: for every $i, j \in I$

1. $\mathfrak{I} \vDash_d i : A \sqsubseteq B$, if $\mathcal{I}_i \vDash A \sqsubseteq B$;
2. $\mathfrak{I} \vDash_d \mathcal{T}_i$, if $\mathfrak{I} \vDash_d i : A \sqsubseteq B$ for all $A \sqsubseteq B$ in $\mathcal{T}_i$;
3. $\mathfrak{I} \vDash_d i : x \xrightarrow{\sqsubseteq} j : y$, if $r_{ij}(x^{\mathcal{I}_i}) \subseteq y^{\mathcal{I}_j}$;
4. $\mathfrak{I} \vDash_d i : x \xrightarrow{\sqsupseteq} j : y$, if $r_{ij}(x^{\mathcal{I}_i}) \supseteq y^{\mathcal{I}_j}$;
5. $\mathfrak{I} \vDash_d \mathfrak{B}_{ij}$, if $\mathfrak{I}$ satisfies all bridge rules in $\mathfrak{B}_{ij}$;
6. $\mathfrak{I} \vDash_d \mathfrak{T}$, if for every $i, j \in I$, $\mathfrak{I} \vDash_d \mathcal{T}_i$ and $\mathfrak{I} \vDash_d \mathfrak{B}_{ij}$;
7. $\mathfrak{T} \vDash_d i : C \sqsubseteq D$ if for every $\mathfrak{I}$, $\mathfrak{I} \vDash_d \mathfrak{T}$ implies $\mathfrak{I} \vDash_d i : C \sqsubseteq D$.

Let us see now how bridge rules affect concept subsumption. Hereafter, $\mathfrak{B}_{ij}^{\text{into}}$ and $\mathfrak{B}_{ij}^{\text{onto}}$ will denote the set of into- and onto-bridge rules of $\mathfrak{B}_{ij}$ respectively.

Monotonicity Bridge rules do not delete local subsumptions. Formally:

$$\mathcal{T}_i \vDash A \sqsubseteq B \implies \mathfrak{T} \models_d i : A \sqsubseteq B \tag{6}$$

Directionality T-box without incoming bridge rules is not affected by other T-boxes. Formally, if $\mathfrak{B}_{ki} = \emptyset$ for any $k \neq i \in I$, then:

$$\mathfrak{T} \models_d i : A \sqsubseteq B \implies \mathcal{T}_i \vDash A \sqsubseteq B \tag{7}$$

Strong directionality Sole into- or onto-bridge rules incoming to local terminology do not affect it. Formally, if for all $k \neq i$ either $\mathfrak{B}_{ki}^{\text{into}} = \emptyset$ or $\mathfrak{B}_{ki}^{\text{onto}} = \emptyset$, then:

$$\mathfrak{T} \models_d i : A \sqsubseteq B \implies \mathcal{T}_i \models A \sqsubseteq B \tag{8}$$

Local inconsistency The fact that $\mathfrak{B}_{ij}$ contains into- and onto-bridge rules does not imply that inconsistency propagates. Formally:

$$\mathfrak{T} \models_d i : \top \sqsubseteq \bot \;\;\not\Longrightarrow\;\; \mathfrak{T} \models_d j : \top \sqsubseteq \bot \tag{9}$$

Simple subsumption propagation Combination of onto- and into-bridge rules allows to propagate subsumptions across ontologies. Formally, if $\mathfrak{B}_{ij}$ contains $i : A \xrightarrow{\sqsupseteq} j : G$ and $i : B \xrightarrow{\sqsubseteq} j : H$, then:

$$\mathfrak{T} \models_d i : A \sqsubseteq B \implies \mathfrak{T} \models_d j : G \sqsubseteq H \tag{10}$$

Generalized subsumption propagation If $\mathfrak{B}_{ij}$ contains $i : A \xrightarrow{\sqsupseteq} j : G$ and $i : B_k \xrightarrow{\sqsubseteq} j : H_k$ for $1 \leq k \leq n$, then:

$$\mathfrak{T} \models_d i : A \sqsubseteq \bigsqcup_{k=1}^{n} B_k \implies \mathfrak{T} \models_d j : G \sqsubseteq \bigsqcup_{k=1}^{n} H_k \tag{11}$$

Among the given properties, property (9) and property (11) play special roles. The first one is important as it allows us to explain how full and empty holes constitute "locally inconsistent interpretations". The second one is important as it constitutes the main reasoning step of the tableau algorithm proposed in the next section. The proofs of the above properties can be found in [1, 10].

Example 2.2. In the hierarchy WNP of the previous example there is no subsumption relation between $\mathsf{Engineer}$ and Worker. From bridge rules (1–3) and from the fact that in the ISCO-88 ontology the concept Architects_Engineers_And_Related_Professionals is a subclass of $\mathsf{Professionals}$, it is impossible to infer that $\mathsf{Engineers}$ is a subclass of Worker, i.e. that in WNP $\mathsf{Engineers} \sqsubseteq \mathsf{Worker}$. Similarly, the bridge rules (4) and (5) allow to infer that WNP classes $\mathsf{Gatekeeper}$ and Child are disjoint, i.e. that WNP : $\mathsf{Gatekeeper} \sqcap \mathsf{Child} \sqsubseteq \bot$.

3 Distributed reasoning in DDL

The reasoning services one would like to have in the web of ontologies are the following:

Local reasoning services are all kind of reasoning services one wants to have for a local ontology. The adjective "local" indicates that these reasoning services consider a local ontology as a stand alone object (no bridge rules are taken into account). The fundamental local reasoning service is *local subsumption*, i.e. the fact that $\mathcal{T}_i \models C \sqsubseteq D$.

Global reasoning services are services which take into account local ontologies in the context of the whole ontology space. These services should allow to infer subsumption between concepts on the basis of bridge rules, as well as new bridge rules on the basis of the existing ones. In this paper, we will focus on the basic global reasoning service that computes *global subsumption*, i.e. the fact that $\mathfrak{T} \models_d i : C \sqsubseteq D$.

A first proposal for implementing global reasoning services in DDL is based on reduction of a DTB $\mathfrak{T}$ to an equivalent global T-box $\mathcal{T}_G$, such that subsumption in $\mathfrak{T}$ can be computed via subsumption in $\mathcal{T}_G$ (see [1] for the transformation details). In this approach, however, a DTB can not be trivially reduced to a single global T-box simply by indexing the concepts and roles with the T-box they occur in. Furthermore, the reformulation done, works in the limited case when all local T-boxes are consistent. We therefore, would like to investigate a more general decision procedure.

Our proposal consist in building a distributed tableau for DDL on top of state of the art DL tableaux, implemented in FaCT and DLP[7], RACER[15], Pellet, and other DL systems. Given a concept C, they generate a tableau of C, $\mathbf{Tab}(C)$. Subsumption between concepts C and D is performed by checking the presence of clashes in all the branches of $\mathbf{Tab}(C \sqcap \neg D)$.

To understand how local tableaux are combined in order to check global subsumption we first consider a limited case of DDL that is composed of only two T-boxes $\mathcal{T}_1$ and $\mathcal{T}_2$, and bridge rules of only one direction from 1 to 2. Though this is unrealistic limitation, it constitutes a mandatory step, from which one can generalize and build a procedure capable of dealing with complex DDL topologies. For the sake of simplicity, we assume the second premise that requires the atomicity of concepts involved into bridge rules. This restriction can be later relaxed, since any bridge rule involving a complex concept $i : C$, can be replaced with a bridge with a new atomic concept $i : A$ and by the addition of the definition $A \equiv C$ to $\mathcal{T}_i$.

Example 3.1. For a distributed T-box $\mathfrak{T}_{12} = \langle \mathcal{T}_1, \mathcal{T}_2, \mathfrak{B}_{12} \rangle$, suppose that $\mathcal{T}_1$ contains axioms $A_1 \sqsubseteq B_1$ and $A_2 \sqsubseteq B_2$, $\mathcal{T}_2$ does not contain any axiom, and that $\mathfrak{B}_{12}$ contains the following bridge rules:

$$1 : B_1 \xrightarrow{\sqsubseteq} 2 : H_1 \qquad 1 : B_2 \xrightarrow{\sqsubseteq} 2 : H_2 \tag{12}$$

$$1 : A_1 \xrightarrow{\sqsupseteq} 2 : G_1 \qquad 1 : A_2 \xrightarrow{\sqsupseteq} 2 : G_2 \tag{13}$$

Let us show that $\mathfrak{T}_{12} \models_d 2 : G_1 \sqcap G_2 \sqsubseteq H_1 \sqcap H_2$, i.e. that for any distributed interpretation $\mathfrak{I} = \langle \mathcal{I}_1, \mathcal{I}_2, r_{12} \rangle$, $(G_1 \sqcap G_2)^{\mathcal{I}_2} \subseteq (H_1 \sqcap H_2)^{\mathcal{I}_2}$.

1. Suppose that by contradiction there is an $x \in \Delta_2$ such that $x \in (G_1 \sqcap G_2)^{\mathcal{I}_2}$ and $x \notin (H_1 \sqcap H_2)^{\mathcal{I}_2}$.

2. Then $x \in G_1^{\mathcal{I}_2}$, $x \in G_2^{\mathcal{I}_2}$, and either $x \notin H_1^{\mathcal{I}_2}$ or $x \notin H_2^{\mathcal{I}_2}$.

3. Let us consider the case where $x \notin H_1^{\mathcal{I}_2}$. From the fact that $x \in G_1^{\mathcal{I}_2}$, by the bridge rule (13), there is $y \in \Delta_1$ with $\langle y, x \rangle \in r_{12}$, such that $y \in A_1^{\mathcal{I}_1}$.

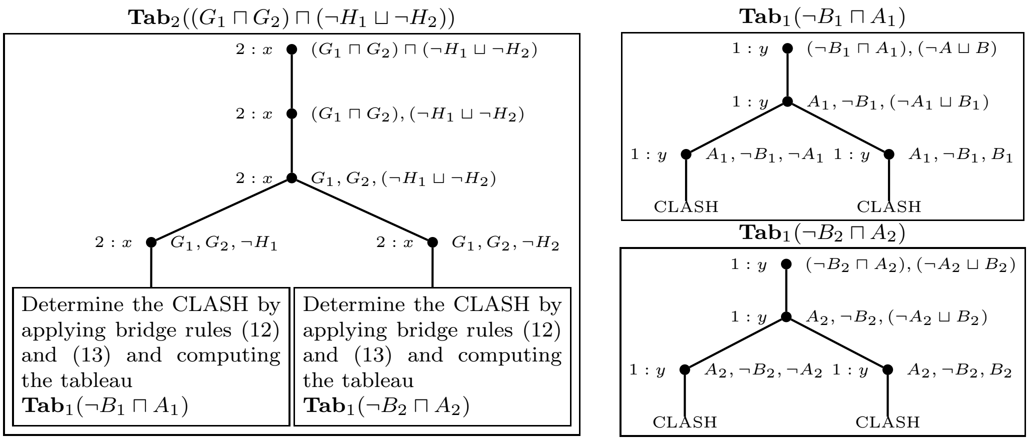

Figure 3: An example of distributed tableau.

4. From the fact that $x \notin H_1^{\mathcal{I}_1}$, by bridge rule (12), we can infer that for all $y \in \Delta_1$ if $\langle y, x\rangle \in r_{12}$ then $y \notin B_1^{\mathcal{I}_1}$.

5. But, since $A \sqsubseteq B \in \mathcal{T}_1$, then $y \in B_1^{\mathcal{I}_1}$, and this is a contradiction.

6. The case where $x \notin H_2^{\mathcal{I}_2}$ is similar.

The above combination of a tableau in $\mathcal{T}_2$ with a tableau in $\mathcal{T}_1$ gives a distributed tableau in $\mathfrak{T}$, depicted in Figure 3.

The intuitions given in Example 3.1 can be generalized for the case of multiple T-boxes, when there are no cyclical references between them. Formally, distributed T-box $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i\in I}, \{\mathfrak{B}_{ij}\}_{i\neq j\in I}\rangle$ is acyclical if $\mathfrak{B}_{ij} \neq \emptyset$ requires $i < j$ for all $i, j \in I$.

Algorithm 1 implements a distributed reasoning procedure intuitively introduced above. Here we define a distributed procedure **dTab**, which takes as an input a complex concept Φ to be verified and returns the result of its (un)satisfiability test. The algorithm first builds a local completion tree T by running local tableau algorithm **Tab**, and further attempts to close open branches of T by checking the bridge rules, which are capable of producing the clash in nodes of T. According to the local tableau algorithm, each node x introduced during creation of the completion tree, is labeled with a function $L(x)$ containing concepts that x must satisfy.

4 Prototype implementation

To evaluate the proposed distributed reasoning procedure we built a prototype modeling the P2P architecture given in Figure 1. Each peer ontology manager maintains ontologies in OWL and mappings in C-OWL, and provides local/global reasoning services, such as performing classification and checking entailment.

The key role in the ontology manager is played by a distributed reasoning engine, implementing developed distributed tableau algorithm. The kernel of the engine is formed by Pellet OWL DL reasoner[4]. Opennes of the source code and implementation

[4] `http://www.mindswap.org/2003/pellet`.

Algorithm 1 Distributed reasoning procedure

$\mathbf{dTab}_j(\Phi)$

BEGIN
T=$\mathbf{Tab}_j(\Phi)$; {perform local reasoning and create completion tree}
if (T is not clashed) **then**
  for each open branch β in T **do**
    repeat
      select node $x \in \beta$ and an $i \neq j$;
      $\mathbb{C}_i^{onto}(x) = \{C \mid i : C \xrightarrow{\sqsupseteq} j : D,\ D \in L(x)\}$;
      $\mathbb{C}_i^{into}(x) = \{C \mid i : C \xrightarrow{\sqsubseteq} j : D, \neg D \in L(x)\}$;
      if $((\mathbb{C}_i^{onto}(x) \neq \emptyset)$ and $(\mathbb{C}_i^{into}(x) \neq \emptyset))$ **then**
        for each $C \in \mathbb{C}^{onto}$ **do**
          if ($\mathbf{dTab}_i(C \sqcap \neg \bigsqcup \mathbb{C}_i^{into})$ is not satisfiable) **then**
            close β; {clash in x}
            break; {verify next branch}
          end if
        end for
      end if
    until ((β is open) and (there exist not verified nodes in β))
  end for{all branches are verified}
end if
if (T is clashed) **then**
  return unsatisfiable;
else
  return satisfiable;
end if
END

in java made Pellet a good candidate for our prototype. Extension of the core reasoning functionality of Pellet transforms it to the distributed successor called *D-Pellet*.

To depicture the life cycle of D-Pellet, consider the case where a peer ontology manager is asked to perform one of the supported reasoning services in a local ontology it maintains. The ontology manager submits this query to D-Pellet, which in turn invokes the relative core Pellet functionality and checks for available mappings. Mapping processing can generate subqueries which are dispatched by the ontology manager to the corresponding foreign ontology manager. In turn, this starts another reasoning cycle. The reasoning stops when the initial D-Pellet receives the answers to the subproblems it sent out. Analysis of the subproblem answers defines the final reasoning result.

5 Conclusions

In this paper we have presented a tableau-based distributed reasoning procedure for DDL. We made several assumptions to study the reasoning in DDL, such as acyclicity of bridge rules and atomicity of concepts involved into bridge rules. The future work is to relax these assumptions in order to receive a practically usable framework.

References

[1] A.Borgida and L.Serafini. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics*, pages 153–184, 2003.

[2] C.Ghidini and F.Giunchiglia. Local model semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.

[3] C.Ghidini and L.Serafini. Distributed first order logics. In *Proc. of the Frontiers of Combining Systems*, pages 121–139, 2000.

[4] F.Giunchiglia. Contextual reasoning. *Epistemologia, special issue on I Linguaggi e le Macchine*, XVI:345–364, 1993.

[5] F.Giunchiglia and L.Serafini. Multilanguage hierarchical logics (or: How we can do without modal logics). *Artificial Intelligence*, 65(1):29–70, 1994.

[6] G.Antoniou and F. van Harmelen. Web ontology language: Owl. In *Handbook on Ontologies in Information Systems*, pages 67–92, 2003.

[7] I.Horrocks and P.F.Patel-Schneider. FaCT and DLP. In *Proc. of the Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX'98)*, pages 27–30, 1998.

[8] I.Horrocks, P.F.Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[9] I.Horrocks, U.Sattler, and S.Tobies. Practical reasoning for very expressive description logics. *Logic Journal of IGPL*, 8(3):239–263, 2000.

[10] L.Serafini and A.Tamilin. Distributed reasoning services for multiple ontologies. Technical Report DIT-04-029, University of Trento, 2004.

[11] L.Serafini and F.Roelofsen. Satisfiability for propositional contexts. In *Proc. of the Principles of Knowledge Representation and Reasoning (KR2004)*, 2004. Accepted for publication.

[12] M.Bonifacio, P.Bouquet, and P.Traverso. Enabling distributed knowledge management. Managerial and technological implications. *Novatica and Informatik/Informatique*, III(1), 2002.

[13] M.Ehrig, Ch.Tempich, J.Broekstra, F. van Harmelen, M.Sabou, R.Siebes, S.Staab, and H.Stuckenschmidt. A metadata model for semantics-based p2p systems. In *Proc. of the 2nd Konferenz Professionelles Wissensmanagement*, 2003.

[14] P.Bouquet, F.Giunchiglia, F. van Harmelen, L.Serafini, and H.Stuckenschmidt. C-owl: Contextualizing ontologies. In *Proc. of the 2d International Semantic Web Conference (ISWC2003)*, pages 164–179, 2003.

[15] V.Haarslev and R.Moller. Racer system description. In *Proc. of the International Joint Conference on Automated Reasoning (IJCAR2001)*, pages 701–706, 2001.