A Federated Tableau Algorithm for F-$\mathcal{ALCI}$

George Voutsadakis[1], Giora Slutzki[1], Jie Bao[2], and Vasant Honavar[1]

[1] Iowa State University, Ames, IA 50011, USA,
[2] Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Abstract. Many semantic web applications require support for knowledge representation and inference over a federation of multiple autonomous ontology modules, without having to combine them in one location. Federated $\mathcal{ALCI}$ or F-$\mathcal{ALCI}$ is a modular description logic, each of whose modules is roughly an $\mathcal{ALCI}$ ontology ($\mathcal{ALC}$ with inverse roles). F-$\mathcal{ALCI}$ supports importing of both concepts and roles across modules as well as contextualized interpretation of logical connectives. We present a federated tableau algorithm for reasoning with a collection of interlinked F-$\mathcal{ALCI}$ ontology modules without the need to combine the modules into a single ontology. Local reasoners apply tableau expansion rules as in the ordinary $\mathcal{ALCI}$ tableau algorithm. Coordination is achieved by message exchanges between local tableaux maintained by the individual reasoners. We prove soundness and completeness of the federated tableau algorithm and show that its worst-case running time is nondeterministic doubly exponential in the size of the largest ontology module.

1 Introduction

In its traditional form the world-wide web consists of data that is geared towards human understanding and processing. However, the rapid increase in the amount and complexity of information available on the web calls for methods for its automated analysis and interpretation. Thus, there is an increasing emphasis on machine interpretable representations of information with the goal of transforming the current web into a semantic web. The most common way to-date to represent and reason about information on the semantic web is by organizing it into various ontologies, or clusters of domain-specific data. Each ontology addresses a particular domain of knowledge. Ontologies are usually developed and maintained independently by autonomous groups that borrow terminology, facts, instances etc, from each other. They are usually built using ontology languages, such as OIL and OWL [12]. OWL is now recommended by the W3C consortium [5]. Ontology languages are based on Description Logics (DLs), which constitute a family of logic-based knowledge representation languages. They are ordinarily decidable fragments of first-order logic or decidable extensions of those fragments. They are often equipped with tableau-based decision procedures for problems such as computing subsumption hierarchies of concepts, satisfiability of concepts, or instances of a concept expression. In [13] a decision procedure for the very expressive DL $\mathcal{SHOIQ}$ is presented. It extends a tableau algorithm for $\mathcal{SHIQ}$ [14], which gave rise to several implemented reasoners [11, 17, 22].

The need to balance the requirement of autonomy against those of collaboration in developing ontologies has recently led to increasing interest in modular ontologies. These are ontologies that are physically and/or conceptually distributed. Each module of such an ontology is independently developed to address specific aspects or subdomain of expertise of a large domain of knowledge. The modules are interdependent in the sense that the various subdomains are most conveniently described by borrowing concepts and data from each other. This supports the autonomy of different groups engaged in developing ontology modules in their respective areas of expertise, eliminates duplication and redundancy and encourages modularity in the construction of ontologies. Multiple modular ontology languages have been proposed to facilitate such an autonomous collaborative development of ontologies. Among them, the best known are Distributed Description Logics (DDL) [7], $\mathcal{E}$-Connections [8, 15] and Package-Based Description Logics (P-DLs) [4].

Since its inception, the semantic web has been envisioned as a non-centralized, highly distributed collection of ontologies with a degree of redundant and overlapping knowledge [6]. The decentralized nature of the web necessarily implies that distant and diverse users will create and use their own local ontologies to organize and reason about their data depending on their special needs. The independence and autonomy clearly makes it easier for these local communities to revise, update, or modify their ontologies according to their own requirements or to accommodate new data. This scenario calls for a distributed reasoning approach in which ontologies have each its own reasoning capabilities and the reasoners communicate with each other whenever a need arises. Computationally, it is advantageous to do as much reasoning as possible at the local level, taking advantage of the specific properties of the local environment. On many occasions it is not even feasible to reason with a "centralized" ontology resulting from integrating the various modules because its size is too large for such an integration to be efficient. In other cases, such an integration may not be possible because the autonomous ontology modules can only selectively share information with each other due to issues of security, privacy, copyright etc. These factors accentuate the need for a federated approach to reasoning that does not require the physical integration of the ontology modules. Numerous researchers, including Serafini et al. [19, 21, 20] have argued eloquently in favor of the modular approach and even designed specific systems/architectures (DRAGO) to facilitate decentralized reasoning over distributed ontologies. Some authors advocated partitioning of large ontologies into smaller but computationally "leaner & meaner" and possibly more coherent ontologies [23, 18]. Against this background, several algorithms have been presented for the three paradigms mentioned above. Serafini et al. [19, 21, 20] introduced a tableau algorithm for reasoning with DDL. Grau et al. [10, 9] present a tableau procedure for $\mathcal{E}$-Connections. Finally, Bao et al. [2] present distributed reasoning algorithms for P-DLs.

The main goal of this paper is to present a federated tableau-based algorithm for the fully contextualized federated description logic F-$\mathcal{ALCI}$, introduced in [24]. Each of the F-$\mathcal{ALCI}$ modules is an $\mathcal{ALCI}$ ontology. One of the distinctive features of this modular language is that it contextualizes all logical connectives (contrast with P-DLs, where only negation is contextual). Moreover, it allows greater semantic flexibility than P-DLs at the expense of some properties that may be desirable in some contexts, but not in others, such as transitive reusability of knowledge and preservation of concept unsatisfiability. We present a nondeterministic doubly exponential federated tableau-based algorithm, that allows us to test concept satisfiability in F-$\mathcal{ALCI}$ from a specific module's point of view. Among its novel features, specifically designed to handle contextualized connectives, are: (a) a new normal form for concept expressions, called negation local form, replacing negation normal form, (b) new "contextual" tableaux expansion rules and (c) a specially tailored synchronization mechanism based on message exchanges. Although it is well-known (see [24]) that a F-$\mathcal{ALCI}$ ontology can be integrated in a context sensitive way into a single $\mathcal{ALCI}$ ontology, the algorithm presented here does not require such an integration.

2 F-$\mathcal{ALCI}$ Syntax and Semantics

Let $G = \langle V, E\rangle$, with $V = \{1, 2, \dots, n\}$, be a directed acyclic graph augmented with loops. The nodes of this graph represent modules of the federated ontology and the edges represent, roughly speaking, direct importing relations, i.e., allowable importing links through which a target module may import either concepts or roles from the source module. For every node $i \in V$, the i-language always includes a set $\mathcal{C}_i$ of *i-concept names* and a set $\mathcal{R}_i$ of *i-role names*. The set $\widehat{\mathcal{R}}_i$ of **i-role expressions** consists of expressions of the form R, R^-, with $R \in \mathcal{R}_j, (j, i) \in E$ (R^- stands for the inverse of R). The set $\widehat{\mathcal{C}}_i$ of **i-concept expressions** consists of recursively defined expressions of the form:

$$C \in \mathcal{C}_j, \top_j, \bot_j, \neg_j C, C \sqcap_j D, C \sqcup_j D, \exists_j R.C, \forall_j R.C,\ \ (j, i) \in E, \tag{1}$$

where $C, D \in \widehat{\mathcal{C}}_i \cap \widehat{\mathcal{C}}_j$ and $R \in \widehat{\mathcal{R}}_i \cap \widehat{\mathcal{R}}_j$. The **$i$-formulas** are of the form $C \sqsubseteq D$, with $C, D \in \widehat{\mathcal{C}}_i$. A **local TBox** T_i is a finite set of i-formulas and a **knowledge base** (KB) or **TBox** is a collection $T = \{T_i\}_{i\in V}$.

We turn now to the semantics of F-$\mathcal{ALCI}$. An **interpretation** $\mathcal{I} = \langle\{\mathcal{I}_i\}_{i\in V}, \{r_{ij}\}_{(i,j)\in E}\rangle$ consists of a family $\mathcal{I}_i = \langle \Delta^i, \cdot^i\rangle, i \in V$, of **local interpretations**, together with a family of **image domain relations** $r_{ij} \subseteq \Delta^i \times \Delta^j, (i, j) \in E$, such that $r_{ii} = \mathrm{id}_{\Delta^i}$, for all $i \in V$. We require that at least one of the local domains Δ^i be nonempty. For a binary relation $r \subseteq \Delta^i \times \Delta^j$, $X \subseteq \Delta^i$ and $S \subseteq \Delta^i \times \Delta^i$, we set

$$r(X) := \{y \in \Delta^j : (\exists x \in X)((x, y) \in r)\},$$

$$r(S) := \{(z, w) \in \Delta^j \times \Delta^j : (\exists (x, y) \in S)((x, z), (y, w) \in r)\}$$

to denote the images of X and S under the binary relation r. The basic features of the local interpretation function $\cdot^i$ are as follows (see [24]):

- $C^i \subseteq \Delta^i$, for all $C \in \mathcal{C}_i$,

- $C^i = r_{ji}(C^j)$, for all $(j,i) \in E$ and $C \in \mathcal{C}_j \cap \widehat{\mathcal{C}}_i$,
- $R^i \subseteq \Delta^i \times \Delta^i$, for all $R \in \mathcal{R}_i$,
- $R^i = r_{ji}(R^j)$, for all $R \in \mathcal{R}_j \cap \widehat{\mathcal{R}}_i$,
- $\top_j^i = r_{ji}(\Delta^j)$, $\bot_j^i = \emptyset$.

The recursive features of the local interpretation function $\cdot^i$ are as follows, for all $R \in \widehat{\mathcal{R}}_i$ and $C, D \in \widehat{\mathcal{C}}_i$:

- ${R^-}^i = {R^i}^-$
- $(\neg_j C)^i = r_{ji}(\Delta^j - C^j)$
- $(C \sqcap_j D)^i = r_{ji}(C^j \cap D^j)$
- $(C \sqcup_j D)^i = r_{ji}(C^j \cup D^j)$
- $(\exists_j R.C)^i = r_{ji}(\{x \in \Delta^j : (\exists y)((x,y) \in R^j \text{ and } y \in C^j)\})$
- $(\forall_j R.C)^i = r_{ji}(\{x \in \Delta^j : (\forall y)((x,y) \in R^j \text{ implies } y \in C^j)\})$

For all $i \in V$, i-**satisfiability**, denoted by $\models_i$, is defined by $\mathcal{I} \models_i C \sqsubseteq D$ iff $C^i \subseteq D^i$. Given a TBox $T = \{T_i\}_{i \in V}$, $\mathcal{I} \models_i T_i$ iff $\mathcal{I} \models_i \tau$, for every $\tau \in T_i$. $\mathcal{I} \models T$ iff $\mathcal{I} \models_i T_i$, for every $i \in V$. An interpretation $\mathcal{I} = \langle \{\mathcal{I}_i\}_{i \in V}, \{r_{ij}\}_{(i,j) \in E} \rangle$ is a **model** of a F-$\mathcal{ALCI}$ KB $T = \{T_i\}_{i \in V}$ if $\mathcal{I} \models T$.

Given a node $w \in V$, let $G_w = \langle V_w, E_w \rangle$ be the subgraph of G induced by the subset of vertices of G from which the vertex w is reachable. Given a KB $T = \{T_i\}_{i \in V}$, let $T_w^* = \{T_i\}_{i \in V_w}$ be the **importing closure** of w. T is **consistent as witnessed by a module** T_w if T_w^* has a model $\mathcal{I} = \langle \{\mathcal{I}_i\}_{i \in V_w}, \{r_{ij}\}_{(i,j) \in E_w} \rangle$, such that $\Delta^w \neq \emptyset$. A concept C is **satisfiable as witnessed by** T_w if there is a model of T_w^*, such that $C^w \neq \emptyset$. A concept subsumption $C \sqsubseteq D$ **is valid as witnessed by** T_w, denoted by $C \sqsubseteq_w D$, if for every model of T_w^*, $C^w \subseteq D^w$. We use $C \equiv_w D$ as the abbreviation of $C \sqsubseteq_w D$ and $D \sqsubseteq_w D$. It becomes clear from these definitions that in F-$\mathcal{ALCI}$ the consistency, satisfiability and subsumption problems are always answered from the local point of view of a *witness module*. Furthermore, it is possible for different modules to draw different conclusions from their own points of view.

3 Negation Local Form of Concept Expressions

Before introducing the notion of tableau for F-$\mathcal{ALCI}$, we will discuss a special normal form that we need in place of the negation normal form, which does not seem to exist for F-$\mathcal{ALCI}$-concept expressions. The need arises from the fact that, in most tableaux algorithms for description logics, the input is first transformed into negation normal form, i.e., a form in which negation occurs only before concept names. To illustrate why the transformation to negation normal form is problematic in the case of contextualized connectives, consider the following example:

Example 1: Let T_1 and T_2 be two modules and assume that A and B are concept names in $\mathcal{C}_1$ and that T_2 is allowed to import names and connectives from T_1. Consider the extension

$$(\neg_2(A \sqcap_1 B))^2 = \Delta^2 \backslash r_{12}(A^1 \cap B^1).$$

Note that the expression $\neg_2 A \sqcup_1 \neg_2 B$ does not even make sense because T_1 is not allowed to import concepts and connectives from T_2. So the only hope for a negation normal form for the concept expression $\neg_2(A \sqcap_1 B)$ in T_2 would be $\neg_1 A \sqcup_1 \neg_1 B$. But its extension is

$$(\neg_1 A \sqcup_1 \neg_1 B)^2 = r_{12}(\Delta^1 \backslash (A^1 \cap B^1)),$$

which, since r_{12} is an arbitrary relation, is not guaranteed to equal $\Delta^2 \backslash r_{12}(A^1 \cap B^1)$. This example unveils some of the difficulties encountered when one attempts to discover a possible normal form for concept expressions that deals with negation and preserves the relevant semantics. $\square$

In the present context, it will be assumed that all concepts are in a variant of the negation normal form, which will be called **negation local form** (NLF). The transformation to NLF affects only concept expressions containing i-negations appearing in module T_i before other i-connectives. In this case, the i-negation is pushed "inward" using a number of simple syntactical rules, similar to the ones used to transform an ordinary $\mathcal{ALC}$-formula into negation normal form. The NLF of an i-concept $C \in \widehat{\mathcal{C}_i}$ is denoted by $\mathtt{nlf}_i(C)$. It is defined recursively on the structure of concepts in $\widehat{\mathcal{C}_i}$ by applying the following rules:

- $\mathtt{nlf}_i(\top_j) = \top_j, \mathtt{nlf}_i(\bot_j) = \bot_j$
- $\mathtt{nlf}_i(\neg_j \top_k) = \begin{cases} \bot_i, & \text{if } j = k = i, \\ \neg_j \top_k, & \text{otherwise} \end{cases}$, $\mathtt{nlf}_i(\neg_j \bot_k) = \begin{cases} \top_i, & \text{if } j = k = i, \\ \neg_j \bot_k, & \text{otherwise} \end{cases}$, for all $(j, i) \in E$
- $\mathtt{nlf}_i(C) = C$, for all $(j, i) \in E$ and all $C \in \mathcal{C}_j$;
-
  - $\mathtt{nlf}_i(\neg_j C) = \neg_j C$, for all $(j, i) \in E$ and all $C \in \mathcal{C}_k \cap \widehat{\mathcal{C}_i} \cap \widehat{\mathcal{C}_j}$;
  - $\mathtt{nlf}_i(\neg_j \neg_k C) = \begin{cases} \mathtt{nlf}_i(C), & \text{if } j = k = i \\ \neg_j \neg_k \mathtt{nlf}_i(C), & \text{otherwise} \end{cases}$, $(j, i) \in E, \neg_k C \in \widehat{\mathcal{C}_i} \cap \widehat{\mathcal{C}_j}$;
  - $\mathtt{nlf}_i(\neg_j (C \sqcap_k D)) = \begin{cases} \neg_i \mathtt{nlf}_i(C) \sqcup_i \neg_i \mathtt{nlf}_i(D), & \text{if } i = j = k \\ \neg_j (\mathtt{nlf}_i(C) \sqcap_k \mathtt{nlf}_i(D)), & \text{otherwise} \end{cases}$, for all $(j, i) \in E$, $C, D \in \widehat{\mathcal{C}_i} \cap \widehat{\mathcal{C}_j}$;
  - $\mathtt{nlf}_i(\neg_j (C \sqcup_k D))$ is similar;
  - $\mathtt{nlf}_i(\neg_j \forall_k R.C) = \begin{cases} \exists_i R.\neg_i \mathtt{nlf}_i(C), & \text{if } i = j = k \\ \neg_j \forall_k R.\mathtt{nlf}_i(C), & \text{otherwise} \end{cases}$, for all $(j, i) \in E$ and $\forall_k R.C \in \widehat{\mathcal{C}_i} \cap \widehat{\mathcal{C}_j}$;
  - $\mathtt{nlf}_i(\neg_j \exists_k R.C)$ is similar;
- $\mathtt{nlf}_i(C \sqcap_j D) = \mathtt{nlf}_j(C) \sqcap_j \mathtt{nlf}_j(D)$, for all $(j, i) \in E$;
- $\mathtt{nlf}_i(C \sqcup_j D) = \mathtt{nlf}_j(C) \sqcup_j \mathtt{nlf}_j(D)$, for all $(j, i) \in E$;
- $\mathtt{nlf}_i(\forall_j R.C) = \forall_j R.\mathtt{nlf}_j(C)$, for all $(j, i) \in E$;
- $\mathtt{nlf}_i(\exists_j R.C) = \exists_j R.\mathtt{nlf}_j(C)$, for all $(j, i) \in E$.

Example 2: Consider the k-concept expression

$$C = \neg_k((D \sqcap_i \neg_j (E \sqcup_j F)) \sqcup_k \neg_k (E \sqcap_l G)),$$

where D, E, F and G are concept names, i, j and k are distinct and the concept expression is valid, i.e., the importing relations that allow building this concept expression in $\widehat{\mathcal{C}}_k$ hold. If we apply $\mathtt{nlf}_k$ to it, we get

$$\begin{aligned}\mathtt{nlf}_k(C) &= \neg_k \mathtt{nlf}_k(D \sqcap_i \neg_j(E \sqcup_j F)) \sqcap_k \neg_k \mathtt{nlf}_k(\neg_k(E \sqcap_l G))\\ &= \neg_k(\mathtt{nlf}_k(D) \sqcap_i \mathtt{nlf}_k(\neg_j(E \sqcup_j F)) \sqcap_k \neg_k \neg_k(\mathtt{nlf}_k(E) \sqcap_l \mathtt{nlf}_k(G))\\ &= \neg_k(D \sqcap_i (\neg_j(\mathtt{nlf}_k(E) \sqcup_j \mathtt{nlf}_k(F)))) \sqcap_k (E \sqcap_l G)\\ &= \neg_k(D \sqcap_i (\neg_j(E \sqcup_j F))) \sqcap_k (E \sqcap_l G)\end{aligned}$$

□

The next lemma asserts that the transformation from a concept $C \in \widehat{\mathcal{C}}_i$ into its negation local form $\mathtt{nlf}_i(C)$ does not change its meaning from the point of view of module T_i.

Lemma 1 *Let $\Sigma = \{T_i\}_{i \in V}$ be a F-$\mathcal{ALCI}$ KB, $\mathcal{I} = \langle \{\mathcal{I}_i\}_{i \in V},\ \{r_{ij}\}_{i \in P_j^*} \rangle$ an interpretation for Σ, $i \in V$ and $C \in \widehat{\mathcal{C}}_i$. Then $(\mathtt{nlf}_i(C))^i = C^i$.*

Proof:

We employ structural induction on C.

Suppose, first, that $(j, i) \in E$ and $C \in \mathcal{C}_j$. Then $(\mathtt{nnf}_i(C))^i = C^i$ by the definition of $\mathtt{nnf}_i(C)$. Similarly $(\mathtt{nnf}_i(\top_j))^i = \top_j^i$ and $(\mathtt{nnf}_i(\bot_j))^i = \bot_j^i$.

For $C \sqcap_j D$, we have

$$\begin{aligned}(\mathtt{nnf}_i(C \sqcap_j D))^i &= (\mathtt{nnf}_j(C) \sqcap_j \mathtt{nnf}_j(D))^i \quad \text{(by the definition of } \mathtt{nnf}_i)\\ &= r_{j,i}((\mathtt{nnf}_j(C))^j \cap (\mathtt{nnf}_j(D))^j) \quad \text{(by the definition of } \cdot^i)\\ &= r_{j,i}(C^j \cap D^j) \quad \text{(by the induction hypothesis)}\\ &= (C \sqcap_j D)^i. \quad \text{(by the definition of } \cdot^i)\end{aligned}$$

The case of $C \sqcup_j D$ may be handled similarly.

For $\exists_j R.C$, we get

$$\begin{aligned}(\mathtt{nnf}_i(\exists_j R.C))^i &= (\exists_j R.\mathtt{nnf}_j(C))^i \quad \text{(by the definition of } \mathtt{nnf}_i)\\ &= r_{j,i}(\{x \in \Delta^j : (\exists y \in (\mathtt{nnf}_j(C))^j)((x,y) \in R^j)\})\\ &\quad \text{(by the definition of } \cdot^i)\\ &= r_{j,i}(\{x \in \Delta^j : (\exists y \in C^j)((x,y) \in R^j)\})\\ &\quad \text{(by the induction hypothesis)}\\ &= (\exists_j R.C)^i. \quad \text{(by the definition of } \cdot^i)\end{aligned}$$

The case of $\forall_j R.C$ may be handled similarly.

Next, we turn to negation and concentrate on the forms that do change. We have

- $(\mathtt{nlf}_i(\neg_i \top_i))^i = \bot_i^i = \emptyset = \Delta^i \backslash \Delta^i = \Delta^i \backslash \top_i^i = (\neg_i \top_i)^i$.
- $(\mathtt{nlf}_i(\neg_i \bot_i))^i = \top_i^i = \Delta^i = \Delta^i \backslash \emptyset = \Delta^i \backslash \bot_i^i = (\neg_i \bot_i)^i$.
- $(\mathtt{nlf}_i(\neg_i \neg_i C))^i = C^i = \Delta^i \backslash (\Delta^i \backslash C^i) = (\neg_i \neg_i C)^i$.

– For $\neg_i(C \sqcap_i D)$ we have

$$\begin{aligned}(\mathtt{nlf}_i(\neg_i(C \sqcap_i D)))^i &= (\neg_i \mathtt{nlf}_i(C) \sqcup_i \neg_i \mathtt{nlf}_i(D))^i \\ &= (\Delta^i \backslash \mathtt{nlf}_i(C)^i) \cup (\Delta^i \backslash \mathtt{nlf}_i(D)^i) \\ &= (\Delta^i \backslash C^i) \cup (\Delta^i \backslash D^i) \\ &= \Delta^i \backslash (C^i \cap D^i) \\ &= (\neg_i(C \sqcap_i D))^i.\end{aligned}$$

– The case of $\neg_j(C \sqcup_k D)$ is handled similarly.
– For $\neg_j \exists_k R.C$ we have

$$\begin{aligned}(\mathtt{nlf}_i(\neg_i \exists_i R.C))^i &= (\forall_i R.\neg_i C)^i \\ &= \{x \in \Delta^i : (\forall y)((x,y) \in R^i \to y \notin C^i\} \\ &= \Delta^i \backslash \{x \in \Delta^i : (\exists y)((x,y) \in R^i \text{ and } y \in C^i)\} \\ &= \Delta^i \backslash (\exists_i R.C)^i \\ &= (\neg_i \exists_i R.C)^i.\end{aligned}$$

– Finally, the case of $\neg_j \forall_k R.C$ is handled similarly.

■

As far as the NLF is concerned, the reader should notice that, in module T_i, an i-negation is either followed by a concept name $C \in \widehat{\mathcal{C}_i}$ or by a concept C, whose outermost connective is a j-connective, for some $j \neq i$. The first kind will be called **of type 1** and the second **of type 2 with trace** j. This observation will be important for the formulation and analysis of the distributive algorithm and will be called upon many times in the sequel.

4 Federated Tableaux for F-$\mathcal{ALCI}$

Tableau-based algorithms are used to test satisfiability of concepts in description logics. The main idea behind the F-$\mathcal{ALCI}$ tableau algorithm is to construct multiple, federated local tableaux, one for each module, using, to the furthest extent possible, only knowledge locally available to that module. The coordination between local tableaux is achieved via inter-module messages which relate pairs of elements across different local tableaux. In effect, this will build a representation of possible image-domain relations r_{ij}, for $(i,j) \in E$. An i-subconcept of an i-concept expression C is a substring of C, which forms also an i-concept expression. We make this notion precise in Definition 2. It will be used in the definition of a federated tableau for a F-$\mathcal{ALCI}$-concept in NLF with respect to a module T_w, that follows.

Definition 2 *The i-subconcepts* $\mathrm{sub}_i(C)$ *of an F-$\mathcal{ALCI}$ concept $C \in \widehat{\mathcal{C}_i}$ in NLF is inductively defined as:*

$\mathrm{sub}_i(A) = \{A\},\ A \in \bigcup_{(j,i)\in E}(\mathcal{C}_j \cup \{\top_j, \bot_j\})$
$\mathrm{sub}_i(C \boxplus_j D) = \{C \boxplus_j D\} \cup \mathrm{sub}_j(C) \cup \mathrm{sub}_j(D),\ \boxplus \in \{\sqcap, \sqcup\}, (j,i) \in E,$
$\mathrm{sub}_i(\rtimes_j R.C) = \{\rtimes_j R.C\} \cup \mathrm{sub}_j(C),\ \rtimes \in \{\exists, \forall\}, (j,i) \in E,$
$\mathrm{sub}_i(\neg_j C) = \{\neg_j C\} \cup \mathrm{sub}_j(C),\ (j,i) \in E, j \neq i,$
$\mathrm{sub}_i(\neg_i C) = \begin{cases} \{\neg_i C\} \cup \mathrm{sub}_i(C), & \textit{if } \neg_i C \textit{ is of type } 1 \\ \{\neg_i C\} \cup \mathrm{sub}_j(C), & \textit{if } \neg_i C \textit{ is of type } 2 \textit{ with trace } j \end{cases}$

Moreover, define, for every concept expression $C \in \widehat{\mathcal{C}}_i$, $\mathrm{Rol}(C) \subseteq \widehat{\mathcal{R}}_i$ to be the (finite) set of role expressions appearing in C.

For every module T_i, we define

$$C_{T_i} = \top_i \sqcap_i \bigsqcap_{C \sqsubseteq D \in T_i} (\mathtt{nlf}_i(\neg_i C) \sqcup_i \mathtt{nlf}_i(D)),$$

where the $\bigsqcap$ also refers to the i-th conjunction symbol.

Let T_w be a module and $D \in \widehat{\mathcal{C}}_w$ an F-$\mathcal{ALCI}$ concept in NLF. A **federated tableau for** D **with respect to** T_w is a tuple $M = \langle \{M_i\}_{i \in V_w}, \{m_{ij}\}_{(i,j) \in E_w} \rangle$, where each M_i is a local tableau, for $i \in V_w$, and m_{ij} is a **tableau relation** from a local tableau M_i to a local tableau M_j, for $(i, j) \in E_w$.

Each local tableau is a tuple $M_i = \langle U_i, F_i, \mathcal{L}_i \rangle$, where

- U_i is a set of individuals,
- $F_i \subseteq U_i \times U_i$ is a binary relation on U_i,
- $\mathcal{L}_w$ is a label function that assigns elements of $2^{\mathrm{sub}_w(D) \cup \mathrm{sub}_w(C_{T_w})}$ to individuals in U_w and elements of $2^{\mathrm{Rol}(D) \cup \mathrm{Rol}(C_{T_w})}$ to pairs in F_w whereas $\mathcal{L}_i$ is a label function that assigns elements of $2^{\mathrm{sub}_i(C_{T_i})}$ to individuals in U_i and elements of $2^{\mathrm{Rol}(C_{T_i})}$ to pairs in F_i, for $i \neq w$.

Each tableau relation m_{ij} is a subset of $U_i \times U_j$, $(i, j) \in E_w$.

The federated tableau M should satisfy the following conditions:

(D1) there exists $x \in U_w$, such that $D \in \mathcal{L}_w(x)$;
(D2) for every $x \in U_i$, $C_{T_i} \in \mathcal{L}_i(x)$;
(B1) $C \in \mathcal{L}_i(x)$ iff there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $C \in \mathcal{L}_j(x')$, for all $C \in \widehat{\mathcal{C}}_i \cap (\mathcal{C}_j \cup \{\top_j\}), (j, i) \in E_w$;
(B2) $R \in \mathcal{L}_i(\langle x, y \rangle)$ iff there exist $x', y' \in U_j$, with $(x', x), (y', y) \in m_{ji}$, such that $R \in \mathcal{L}_j(\langle x', y' \rangle)$, for all $R \in \widehat{\mathcal{R}}_i \cap \mathcal{R}_j, (j, i) \in E_w$;
(N1) if $C \in \mathcal{L}_i(x)$, then $\neg_i C \notin \mathcal{L}_i(x)$, for every $C \in \widehat{\mathcal{C}}_i$, such that $\neg_i C$ is of type 1;
(N2) if $\neg_i C \in \mathcal{L}_i(x)$ is of type 2 with trace j, then, if $x' \in U_j$, with $(x', x) \in m_{ji}$, then $\neg_j C \in \mathcal{L}_j(x')$, for all $C \in \widehat{\mathcal{C}}_i \cap \widehat{\mathcal{C}}_j, (j, i) \in E_w, j \neq i$;
(N3) if $\neg_j C \in \mathcal{L}_i(x)$, then, there exists $x' \in U_j$, with $(x', x) \in m_{ji}$ and $\neg_j C \in \mathcal{L}_j(x')$, for all $C \in \widehat{\mathcal{C}}_i \cap \widehat{\mathcal{C}}_j, (j, i) \in E_w, j \neq i$;
(A1) if $C_1 \sqcap_j C_2 \in \mathcal{L}_i(x)$, then, if $i = j$, $C_1, C_2 \in \mathcal{L}_i(x)$ and, if $i \neq j$, then, there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $C_1 \sqcap_j C_2 \in \mathcal{L}_j(x')$, for every $C_1, C_2 \in \widehat{\mathcal{C}}_i \cap \widehat{\mathcal{C}}_j$ and $(j, i) \in E_w$;
(A2) if $C_1 \sqcup_j C_2 \in \mathcal{L}_i(x)$, then, if $i = j$, then $C_1 \in \mathcal{L}_i(x)$ or $C_2 \in \mathcal{L}_i(x)$ and, if $i \neq j$, then, there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $C_1 \sqcup_j C_2 \in \mathcal{L}_j(x')$, for every $C_1, C_2 \in \widehat{\mathcal{C}}_i \cap \widehat{\mathcal{C}}_j$ and $(j, i) \in E_w$;
(A3) if $\forall_j R.C \in \mathcal{L}_i(x)$, then, if $i = j$, then, for all $y \in U_i$, such that $R \in \mathcal{L}_i(\langle x, y \rangle)$, we have $C \in \mathcal{L}_i(y)$, and, if $i \neq j$, then, there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $\forall_j R.C \in \mathcal{L}_j(x')$, for all $R \in \widehat{\mathcal{R}}_i \cap \widehat{\mathcal{R}}_j, C \in \widehat{\mathcal{C}}_i \cap \widehat{\mathcal{C}}_j, (j, i) \in E_w$;

(A4) if $\exists_j R.C \in \mathcal{L}_i(x)$, then, if $i = j$, then, there exists $y \in U_i$, such that $R \in \mathcal{L}_i(\langle x, y\rangle)$ and $C \in \mathcal{L}_i(y)$, and, if $i \neq j$, then, there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $\exists_j R.C \in \mathcal{L}_j(x')$, for all $R \in \widehat{\mathcal{R}}_i \cap \widehat{\mathcal{R}}_j, C \in \widehat{\mathcal{C}}_i \cap \widehat{\mathcal{C}}_j, (j, i) \in E_w$.

Condition (D1) ensures that the interpretation of D in the model described by the federated tableau is nonempty. Condition (D2) ensures the satisfiability of all federated TBox axioms in the model. Conditions (B1) and (B2) stipulate that the interpretations of imported concept names and imported role names are inherited from their corresponding interpretations in their original module. Conditions (N1)-(N3) guarantee that all relevant properties of the contextualized negations will be satisfied in the resulting model. In particular, Conditions (N1) and (N2) safeguard the consistency of the model. Conditions (A1)-(A4) ensure the correctness of the interpretation of the remaining localized connectives.

The following two lemmas establish the correspondence between concept satisfiability, and, thus, also between TBox consistency and concept subsumption, and the existence of a federated tableau for that concept in F-$\mathcal{ALCI}$.

Lemma 3 *Let $T = \{T_i\}_{i \in V}$ be a F-$\mathcal{ALCI}$ KB and D be concept in T_w. If D has a federated tableau w.r.t. T_w, then D is satisfiable as witnessed by T_w.*

Proof:
For the "if" direction, let $\langle \{M_i\}_{i \in V_w}, \{m_{ij}\}_{(i,j) \in E_w}\rangle$, with $M_i = \langle U_i, F_i, \mathcal{L}_i\rangle$, be a tableau for D w.r.t. T^*_w. Then, a federated model $\mathcal{I} = \langle \{\mathcal{I}_i\}_{i \in V_w}, \{r_{ij}\}_{(i,j) \in E_w}\rangle$ of T^*_w may be defined as follows:

$$\begin{aligned}
&\Delta^i = U_i;\\
&A^i = \{x \in U_i : A \in \mathcal{L}_i(x)\}, \text{ for every } i\text{-concept name } A;\\
&R^i = \{\langle x, y\rangle \in F_i : R \in \mathcal{L}_i(\langle x, y\rangle)\}, \text{ for every } i\text{-role name } R;\\
&r_{ij} = m_{ij}.
\end{aligned}$$

By using induction on the structure of an i-concept, we show that

$$C \in \mathcal{L}_i(x) \quad \text{implies} \quad x \in C^i. \tag{2}$$

- If C is an i-concept name, then $C \in \mathcal{L}_i(x)$ if and only if, by the definition of C^i, $x \in C^i$.
- If C is a j-concept name or $\top_j$, $j \neq i$, and $C \in \mathcal{L}_i(x)$, then, by Property (B1), there exists $x' \in U_j$, with $(x', x) \in m_{ji} = r_{ji}$, such that $C \in \mathcal{L}_j(x')$. Therefore $x \in r_{ji}(x') \subseteq r_{ji}(C^j) = C^i$.
- Suppose that $\neg_i C \in \mathcal{L}_i(x)$ is of type 1. Then, by Property (N1), $C \notin \mathcal{L}_i(x)$, whence
  - If $C \in \mathcal{C}_i$, then, by the definition of C^i, $x \notin C^i$ and
  - if $C \in \mathcal{C}_j, j \neq i$, then $x \notin C^i = r_{ji}(C^j) = m_{ji}(C^j)$, since otherwise, by Property (B1) and the definition of C^j, there would exist $x' \in U_j$, with $(x', x) \in m_{ji} = r_{ji}$, such that $x' \in C^j$, which would imply $x \in C^i$, contradicting $x \notin C^i$.

- Suppose that $\neg_i C \in \mathcal{L}_i(x)$ is of type 2 with trace j. We must show that $x \in (\neg_i C)^i = \Delta^i \backslash C^i = \Delta^i \backslash r_{ji}(C^j)$. Suppose, to the contrary, that $x \in r_{ji}(C^j)$. Then, there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $x' \in C^j$. But, in that case, by Property (N2), $\neg_j C \in \mathcal{L}_j(x')$, implying, by the induction hypothesis, that $x' \in (\neg_j C)^j = \Delta^j \backslash C^j$, a contradiction.
- For the last case involving negation, assume that $\neg_j C \in \mathcal{L}_i(x)$. Then by Property (N3), there exists $x' \in U_j$, with $(x', x) \in m_{ji}$ and $\neg_j C \in \mathcal{L}_j(x')$. Therefore, using the previous case, we get $x \in m_{ji}(x') \subseteq m_{ji}((\neg_j C)^j) = m_{ji}(\Delta^j \backslash C^j) = (\neg_j C)^i$.
- If $C_1 \sqcap_j C_2 \in \mathcal{L}_i(x)$, then, by Property (A1), there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $C_1 \in \mathcal{L}_j(x')$ and $C_2 \in \mathcal{L}_j(x')$. Therefore, using the induction hypothesis, $x \in m_{ji}(x') \subseteq m_{ji}(C_1^j \cap C_2^j) = m_{ji}((C_1 \sqcap_j C_2)^j) = (C_1 \sqcap_j C_2)^i$.
- The case $C = C_1 \sqcup_j C_2$ may be handled similarly, using Property (A2).
- If $\forall_j R.C \in \mathcal{L}_i(x)$, then, by Property (A3), there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that, for all $y' \in U_j$, with $R \in \mathcal{L}_j(\langle x', y' \rangle)$, $C \in \mathcal{L}_j(y')$. Thus, by the definition of R^j, we get, using the induction hypothesis, that, there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $x' \in (\forall_j R.C)^j$. Hence $x \in m_{ji}((\forall_j R.C)^j) = (\forall_j R.C)^i$.
- Finally, suppose that $\exists_j R.C \in \mathcal{L}_i(x)$. Then, by Property (A4), there exist $x', y' \in U_j$, such that $(x', x) \in m_{ji}$, $R \in \mathcal{L}_j(\langle x', y' \rangle)$ and $C \in \mathcal{L}_j(y')$. Thus, again using the definition of R^j and the induction hypothesis, we get that, there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $x' \in (\exists_j R.C)^j$. This shows that $x \in m_{ji}(x') \subseteq m_{ji}((\exists_j R.C)^j) = (\exists_j R.C)^i$.

Notice, now, that $D^w \neq \emptyset$. In fact, by Property (D1), there exists $x \in U_w$, such that $D \in \mathcal{L}_w(x)$. Therefore, using Implication (2), $x \in D^w \neq \emptyset$. Finally, again using Implication (2), it is shown that, if $C \sqsubseteq D$ is an i-formula, then $C^i \subseteq D^i$. In fact, using Properties (D2) and (A1), we get that $\neg_i C \sqcup D \in \mathcal{L}_i(x)$. Thus, by Property (A2), either $\neg_i C \in \mathcal{L}_i(x)$ or $D \in \mathcal{L}_i(x)$. Therefore, by Implication (2), $x \notin C^i$ or $x \in D^i$, whence $C^i \subseteq D^i$, and, hence, $\mathcal{I}_i \models T_i$. ■

In Lemma 4, the converse is established, i.e., that, if an F-$\mathcal{ALCI}$ concept D in a module T_w is satisfiable as witnessed by T_w, then it has a federated tableau with respect to T_w.

Lemma 4 *Let D be a concept in a module T_w of an F-$\mathcal{ALCI}$ KB $T = \{T_i\}_{i \in V}$. If D is satisfiable as witnessed by T_w, then D has a federated tableau w.r.t. T_w.*

Proof:

Suppose that $\mathcal{I} = \langle \{\mathcal{I}_i\}_{i \in V_w}, \{r_{ij}\}_{(i,j) \in E_w} \rangle$ is a model of T_w^*, with $D^w \neq \emptyset$. A federated tableau $M = \langle \{M_i\}_{i \in V_w}, \{m_{ij}\}_{(i,j) \in E_w} \rangle$ for T_w^*, with $M_i = \langle U_i, F_i, \mathcal{L}_i \rangle$, may be defined as follows:

$$U_i = \Delta^i;$$
$$F_i = \bigcup \{R^i : R \in \widehat{\mathcal{R}}_i\};$$

$$\begin{aligned}
\mathcal{L}_w(x) &= \{C \in \mathrm{sub}_w(D) \cup \mathrm{sub}_w(C_{T_w}) : x \in C^w\},\ x \in \Delta^w;\\
\mathcal{L}_w(\langle x, y\rangle) &= \{R \in \mathrm{Rol}(D) \cup \mathrm{Rol}(C_{T_w}) : \langle x, y\rangle \in R^w\},\ x, y \in \Delta^w;\\
\mathcal{L}_i(x) &= \{C \in \mathrm{sub}_i(C_{T_i}) : x \in C^i\},\ x \in \Delta^i, i \neq w;\\
\mathcal{L}_i(\langle x, y\rangle) &= \{R \in \mathrm{Rol}(C_{T_i}) : \langle x, y\rangle \in R^i\},\ x, y \in \Delta^i;\\
m_{ij} &= r_{ij}.
\end{aligned}$$

We now verify that M is indeed a tableau for D w.r.t. T_w, i.e., that it satisfies all conditions in the definition of a federated tableau (Conditions (D1)-(A4)).

(D1): Since $D^w \neq \emptyset$, there exists $x \in U_w$, such that $D \in \mathcal{L}_w(x)$.

(D2): Since $\mathcal{I}_i$ is a model of T_i, we have, for every $x \in U_i$, $x \in C^i_{T_i}$, whence $C_{T_i} \in \mathcal{L}_i(x)$.

(B1): Suppose $C \in \widehat{\mathcal{C}}_i \cap (\mathcal{C}_j \cup \{\top_j\}), (j, i) \in E$. Then we have $C \in \mathcal{L}_i(x)$ iff, by the definition of $\mathcal{L}_i(x)$, $x \in C^i = r_{ji}(C^j) = m_{ji}(C^j)$ iff, there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $x' \in C^j$, iff, there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $C \in \mathcal{L}_j(x')$.

(B2): Suppose that $R \in \widehat{\mathcal{R}}_i \cap \mathcal{R}_j$. Then $R \in \mathcal{L}_i(\langle x, y\rangle)$ iff $(x, y) \in R^i = r_{ji}(R^j) = m_{ji}(R^j)$ if and only if, there exist $x', y' \in U_j$, with $(x', x), (y', y) \in m_{ji}$, such that $(x', y') \in R^j$ iff $R \in \mathcal{L}_j(\langle x', y'\rangle)$.

(N1): If $C \in \mathcal{L}_i(x)$, such that $\neg_i C$ is of type 1, then $x \in C^i$, whence $x \notin \Delta^i \backslash C^i = (\neg_i C)^i$. Thus $\neg_i C \notin \mathcal{L}_i(x)$.

(N2): Suppose $\neg_i C \in \mathcal{L}_i(x)$ is of type 2 with trace j and $x' \in U_j$, with $(x', x) \in m_{ji} = r_{ji}$. For the sake of obtaining a contradiction, suppose that $\neg_j C \notin \mathcal{L}_j(x')$. Then $x' \notin (\neg_j C)^j = \Delta^j \backslash C^j$, i.e., $x' \in C^j$. Therefore, $x \in r_{ji}(C^j)$, whence $x \notin \Delta^i \backslash r_{ji}(C^j) = (\neg_i C)^i$. This yields $\neg_i C \notin \mathcal{L}_i(x)$, which contradicts our hypothesis.

(N3): Finally, suppose that $C \in \widehat{\mathcal{C}}_i \cap \widehat{\mathcal{C}}_j, (j, i) \in E, j \neq i$, with $\neg_j C \in \mathcal{L}_i(x)$. Thus $x \in (\neg_j C)^i = r_{ji}(\Delta^j \backslash C^j)$. Thus, there exists $x' \in U_j$, with $(x', x) \in r_{ji} = m_{ji}$, such that $x' \notin C^j$. But, then, $x' \in \Delta^j \backslash C^j = (\neg_j C)^j$, whence $\neg_j C \in \mathcal{L}_j(x')$.

(A1): If $C_1 \sqcap_j C_2 \in \mathcal{L}_i(x)$, then $x \in (C_1 \sqcap_j C_2)^i = r_{ji}(C_1^j \cap C_2^j)$. Thus, there exists $x' \in U_j$, with $(x', x) \in r_{ji} = m_{ji}$, such that $x' \in C_1^j$ and $x' \in C_2^j$, i.e., such that $C_1 \in \mathcal{L}_j(x')$ and $C_2 \in \mathcal{L}_j(x')$.

(A2): This case is handled very similarly to the previous one.

(A3): Suppose that $\forall_j R.C \in \mathcal{L}_i(x)$. Then

$$x \in (\forall_j R.C)^i = r_{ji}(\{x' \in \Delta^j : (\forall y' \in \Delta^j)((x', y') \in R^j \to y' \in C^j)\}).$$

This means that there exists $x' \in \Delta^j = U_j$, with $(x', x) \in r_{ji} = m_{ji}$, such that, for all $y' \in \Delta^j = U_j$, with $(x', y') \in R^j$, i.e., $R \in \mathcal{L}_j(\langle x', y'\rangle)$, $y' \in C^j$, i.e., $C \in \mathcal{L}_j(y')$.

(A4): Finally, suppose that $\exists_j R.C \in \mathcal{L}_i(x)$. Then

$$x \in (\exists_j R.C)^i = r_{ji}(\{x' \in \Delta^j : (\exists y' \in \Delta^j)((x', y') \in R^j \text{ and } y' \in C^j)\}).$$

Thus, there exists $x' \in \Delta^j = U_j$, with $(x', x) \in r_{ji} = m_{ji}$, such that, there exists a $y' \in \Delta^j = U_j$, with $(x', y') \in R^j$, i.e., $R \in \mathcal{L}_j(\langle x', y'\rangle)$, and $y' \in C^j$, i.e., $C \in \mathcal{L}_j(y')$.

■

By combining Lemmas 3 and 4, we obtain the first main result of the paper establishing the equivalence between satisfiability and the existence of a tableau.

Theorem 5 *Let $T = \{T_i\}_{i \in V}$ be a F-$\mathcal{ALCI}$ KB and D be a concept in module T_w. Then D is satisfiable as witnessed by T_w iff D has a federated tableau with respect to T_w.*

5 Tableau Algorithm for F-$\mathcal{ALCI}$

We now proceed to describe a sound and complete algorithm to determine the existence of a tableau for an F-$\mathcal{ALCI}$ concept D with respect to a witness package T_w. The algorithm allows a local tableau to be created and maintained by a local reasoner. Thus, reasoning is carried out by a federation of reasoners that communicate with each other via messages instead of a single reasoner over an integrated ontology. Some implementation details, especially those concerning synchronization issues of the federated reasoners, are omitted.

5.1 Federated Completion Graph

The algorithm works on a dynamically evolving federated completion graph, which is a partial finite description of a tableau. A **federated completion graph** is a set $G = \{G_i\}_{i \in V_w}$, of local completion graphs. A **local completion graph** $G_i = \langle V_i, E_i, \mathcal{L}_i \rangle$, $i \in V_w$, consists of a finite set of finite trees, i.e., a forest, where V_i and E_i are the corresponding sets of nodes and edges respectively, and of a function $\mathcal{L}_i$, that assigns labels to nodes and edges in G_i, exactly as was the case with local tableaux. Each node x in V_i represents an individual in the corresponding tableau, denoted $i : x$, and is labeled with $\mathcal{L}_i(x)$, a set of concepts of which x is a member. Each edge $\langle x, y \rangle \in E_i$ represents an edge in the tableau, and is labeled with $\mathcal{L}_i(\langle x, y \rangle)$, the set of roles of which it is an instance.

If $R \in \mathcal{L}_i(\langle x, y \rangle)$ or $R^- \in \mathcal{L}_i(\langle y, x \rangle)$, y is said to be a **local R-successor** of x and x is said to be a **local R-predecessor** of y. **Local ancestors** and **local descendants** of a node are defined in the usual manner.

Every node x has associated with it a set of nodes $\mathrm{org}(x)$, which, informally speaking, are the nodes from which x is related via image domain relations. If $(i : x) \in \mathrm{org}(j : y)$ and $(i, j) \in E_w$, we say that node $y \in V_j$ is an **image** of node $x \in V_i$, that node x is a **pre-image** of node y, and that there is a **graph relation** $\langle x, y \rangle$.

A typical federated completion graph consists of local successor relations in local forests together with graph relations across forests in different local reasoners. To construct a model for the ontology resulting by integrating all modules, as was done in [3], a different technique is used here. One keeps all forests disjoint, but uses the graph relations to map nodes in one forest to nodes in other forests.

5.2 Federated Tableau Expansion

A federated F-$\mathcal{ALCI}$ completion graph is constructed by applying a set of tableau expansion rules and by exchanging messages between local reasoners. The F-$\mathcal{ALCI}$ expansion rules are adapted from the $\mathcal{ALCI}$ expansion rules. The label of each node in each local completion graph M_i will contain C_{T_i}, the internalization of T_i. A local completion graph can create images or pre-images of its local nodes in another local completion graph, as needed, during an expansion.

As in the tableau algorithm for $\mathcal{ALCI}$, some nodes in the graph may be *blocked*. The exact definition, whose main motivation is the detection of cycles in tableau expansions, is as follows:

Definition 6 (Equality Blocking) *For a federated completion graph of an F-$\mathcal{ALCI}$ ontology, a node x is* **directly blocked** *by a node y, if both x and y are in the same local completion graph G_i, for some i, y is a local ancestor of x, and $\mathcal{L}_i(x) = \mathcal{L}_i(y)$. Node x is* **indirectly blocked** *by a node y if one of x's local ancestors is directly blocked by y. Node x is* **blocked** *by y if it is directly or indirectly blocked by y.*

Equality blocking in F-$\mathcal{ALCI}$ only depends on the *local* information in completion graphs, i.e., a node is blocked only by its *local* ancestors.

A **concept reporting message** creates image or pre-image nodes and/or propagates concept labels of a node to the corresponding image node or pre-image node. We use $S\mathrel{+}= X$ to denote the operation of adding the elements of the set X to a set S, i.e., the operation $S = S \cup X$. We have five kinds of concept reporting messages and each of these messages may be transmitted *only once*.

- A **forward concept reporting message** $r^{i\rightarrow j}(x, C)$ executes the following action: **if** there exists $x' \in V_j$, such that $x \in \mathrm{org}(x')$ and $C \notin \mathcal{L}_j(x')$, **then** $\mathcal{L}_j(x')\mathrel{+}= \{C\}$.
- A **soft backward concept reporting message** $r^{j\dashleftarrow i}(x, C)$ executes the following actions: **if** $x' \in V_j$, with $x' \in \mathrm{org}(x)$, **then** $\mathcal{L}_j(x')\mathrel{+}= \{C\}$, if $C \notin \mathcal{L}_j(x')$.
- A **backward concept reporting message** $r^{j\leftarrow i}(x, C)$ executes the following action: create an $x' \in V_j$, with $x' \in \mathrm{org}(x)$ and $\mathcal{L}_j(x') = \{C\}$.
- A **forward role reporting message** $r^{i\rightarrow j}(\langle x, y\rangle, R)$ executes the following action: **if** there exist $x', y' \in V_j$, such that $x \in \mathrm{org}(x'), y \in \mathrm{org}(y')$ and $R \notin \mathcal{L}_j(\langle x', y'\rangle)$, **then** $\mathcal{L}_j(\langle x', y'\rangle)\mathrel{+}= \{R\}$.
- A **backward role reporting message** $r^{j\leftarrow i}(\langle x, y\rangle, R)$ executes the following action: create $x', y' \in V_j$, with $x' \in \mathrm{org}(x), y' \in \mathrm{org}(y)$, and set $\mathcal{L}_j(\langle x', y'\rangle) = \{R\}$.

The expansion rules are:

- A rule ensuring that every element in G_i satisfies C_{T_i}.
  - **D-rule**: **if** $C_{T_i} \notin \mathcal{L}_i(x)$, **then** $\mathcal{L}_i(x)\mathrel{+}= \{C_{T_i}\}$.
- Four rules imposing forward and backward concept and role compatibilities:

  - FCN-rule: **if** $C \in \mathcal{L}_j(x)$, $C \in \widehat{\mathcal{C}}_i \cap (\mathcal{C}_j \cup \{\top_j\}), (j,i) \in E$, and x is not blocked, **then** transmit $r^{j \to i}(x, C)$.
  - BCN-rule: **if** $C \in \mathcal{L}_i(x)$, $C \in \widehat{\mathcal{C}}_i \cap (\mathcal{C}_j \cup \{\top_j\}), (j,i) \in E$, and x is not blocked, **then** transmit $r^{j \leftarrow i}(x, C)$.
  - FRN-rule: **if** $R \in \mathcal{L}_j(\langle x, y\rangle)$, $R \in \widehat{\mathcal{R}}_i \cap \mathcal{R}_j, (j,i) \in E$, and x or y are not blocked, **then** transmit $r^{j \to i}(\langle x, y\rangle, R)$.
  - BRN-rule: **if** $R \in \mathcal{L}_i(\langle x, y\rangle)$, $R \in \widehat{\mathcal{R}}_i \cap \mathcal{R}_j, (j,i) \in E$, and x or y are not blocked, **then** transmit $r^{j \leftarrow i}(\langle x, y\rangle, R)$.
- Two negation rules (a local and a foreign one):
  - L¬-rule: **if** $\neg_i C \in \mathcal{L}_i(x)$ is of type 2 with trace j, $C \in \widehat{\mathcal{C}}_i \cap \widehat{\mathcal{C}}_j, (j,i) \in E, i \neq j$, and x is not blocked, **then** transmit $r^{j \dashleftarrow i}(x, \texttt{nlf}_j(\neg_j C))$.
  - F¬-rule: **if** $\neg_j C \in \mathcal{L}_i(x)$, $C \in \widehat{\mathcal{C}}_i \cap \widehat{\mathcal{C}}_j, (j,i) \in E, i \neq j$, and x is not blocked, **then** transmit $r^{j \leftarrow i}(x, \texttt{nlf}_j(\neg_j C))$.
- Two conjunction rules (a local and a foreign one):
  - L⊓-rule: **if** $C_1 \sqcap_i C_2 \in \mathcal{L}_i(x)$, x is not blocked, and $\{C_1, C_2\} \not\subseteq \mathcal{L}_i(x)$, **then** $\mathcal{L}_i(x) \mathrel{+}= \{C_1, C_2\}$.
  - F⊓-rule: **if** $C_1 \sqcap_j C_2 \in \mathcal{L}_i(x)$, $(j,i) \in E, j \neq i$, and x is not blocked, **then** transmit $r^{j \leftarrow i}(x, C_1 \sqcap_j C_2)$.
- Two disjunction rules (a local and a foreign one):
  - L⊔-rule: **if** $C_1 \sqcup_i C_2 \in \mathcal{L}_i(x)$, x is not blocked, and $\{C_1, C_2\} \cap \mathcal{L}_i(x) = \emptyset$, **then** $\mathcal{L}_i(x) \mathrel{+}= \{C_1\}$ or $\mathcal{L}_i(x) \mathrel{+}= \{C_2\}$.
  - F⊔-rule: **if** $C_1 \sqcup_j C_2 \in \mathcal{L}_i(x)$, $(j,i) \in E, j \neq i$, and x is not blocked, **then** transmit $r^{j \leftarrow i}(x, C_1 \sqcup_j C_2)$.
- Two universal quantification rules (a local and a foreign one):
  - L∀-rule: **if** $\forall_i R.C \in \mathcal{L}_i(x)$, x is not blocked and, there exists $y \in V_i$, with $R \in \mathcal{L}_i(\langle x, y\rangle)$, **then** $\mathcal{L}_i(y) \mathrel{+}= \{C\}$, if $C \notin \mathcal{L}_i(y)$.
  - F∀-rule: **if** $\forall_j R.C \in \mathcal{L}_i(x)$, $(j,i) \in E, j \neq i$, and x is not blocked, **then** transmit $r^{j \leftarrow i}(x, \forall_j R.C)$.
- Two existential quantification rules (a local and a foreign one):
  - L∃-rule: **if** $\exists_i R.C \in \mathcal{L}_i(x)$ and x is not blocked and x has no local R-successor y, with $C \in \mathcal{L}_i(y)$, **then** create a new node $y \in V_i$ and set $\mathcal{L}_i(\langle x, y\rangle) = \{R\}$ and $\mathcal{L}_i(y) = \{C\}$.
  - F∃-rule: **if** $\exists_j R.C \in \mathcal{L}_i(x)$, $(j,i) \in E, j \neq i$, and x is not blocked, **then** transmit $r^{j \leftarrow i}(x, \exists_j R.C)$.

All the rules presented above correspond to properties that the federated tableau must satisfy. The D-rule makes sure that Property (D2) is satisfied. The FCN (Forward Concept Name) and BCN (Backward Concept Name) rules ensure that Property (B1) of a tableau is satisfied by the completion graphs. Similarly, the FRN (Forward Role Name) and BRN (Backward Role Name) rules take care of Property (B2). The negation, conjunction, disjunction, universal and existential quantification rules have both an L (Local) and an F (Foreign) version. These ten rules collectively make sure that all properties pertaining to negation, conjunction, disjunction and the quantifiers, i.e., Properties (N1)-(A4), of a tableau are satisfied by the completion graphs.

A federated completion graph is **complete** if no F-$\mathcal{ALCI}$ expansion rule can be applied to it, and it is **clash-free** if there is no x in any local completion graph G_i, such that both C and $\neg_i C$ are in $\mathcal{L}_i(x)$, for some concept C.

For a satisfiability query of a concept D as witnessed by a module T_w, a local completion graph G_w, with an initial node x_0, with $\mathcal{L}_w(x_0) = \{D\}$, will be created first. The F-$\mathcal{ALCI}$ tableau expansion rules will be applied until a complete and clash-free federated completion graph is found or until all search efforts for such a federated completion graph fail.

Example 3: We present a very simple example to illustrate how some of the expansion rules and some of the concept reporting messages work in the algorithm. Suppose that the underlying graph G is the complete graph on two nodes, 1 and 2. Module T_1, corresponding to node 1, has two concept names A, B and module T_2, corresponding to node 2, does not have any concept names. Suppose that T_1 consists of the single subsumption $A \sqsubseteq B$ and that we want to check the satisfiability of $\neg_2 A \sqcup_1 B$ from the point of view of the second module. We define $C_{T_1} = \top_1 \sqcap_1 (\neg_1 A \sqcup_1 B)$ and the algorithm is initialized by creating a node x_0 in G_2, such that $\mathcal{L}_2(x_0) = \{\neg_2 A \sqcup_1 B\}$. (See Figure 1.)

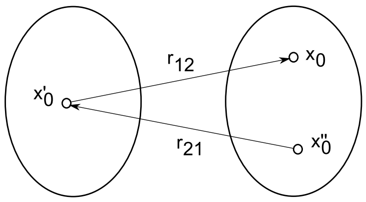

Fig. 1. The distributed model of Example 2.

The F$\sqcup$-rule applies, whence a concept reporting message $r^{1\leftarrow 2}(x_0, \neg_2 A \sqcup_1 B)$ is sent from module 2 to module 1. This creates a node x'_0 in G_1, with $\mathcal{L}_1(x'_0) = \{\neg_2 A \sqcup_1 B\}$. We also have that $\mathrm{org}(x_0) = \{x'_0\}$.

Next, the D-rule is applied to x'_0, whence $\mathcal{L}_1(x'_0) = \{\top_1 \sqcap_1 (\neg_1 A \sqcup_1 B), \neg_2 A \sqcup_1 B\}$, and, then, the L$\sqcap$-rule and twice the L$\sqcup$-rule apply to obtain

$$\mathcal{L}_1(x'_0) = \{\top_1 \sqcap_1 (\neg_1 A \sqcup_1 B), \neg_2 A \sqcup_1 B, \top_1, \neg_1 A \sqcup_1 B, \neg_1 A, \neg_2 A\}.$$

Finally, the occurrence of $\neg_2 A$ triggers an application of the F$\neg$-rule, which causes the transmission of a $r^{2\leftarrow 1}(x'_0, \neg_2 A)$ concept reporting message. A new node x''_0 is created in G_2, with label $\mathcal{L}_2(x''_0) = \{\neg_2 A\}$ and $\mathrm{org}(x'_0) = \{x''_0\}$. As a consequence of this the L$\neg$-rule is applied in G_2 and an $r^{1\dashleftarrow 2}(x_0, \neg_1 A)$ concept reporting message is delivered. This message, however, does not cause any changes because there is no x in G_1, such that $x \in \mathrm{org}(x''_0)$.

The final federated model $\mathcal{I} = \langle \{\mathcal{I}_i\}_{i=1,2}, \{r_{ij}\}_{i,j=1,2} \rangle$, therefore, consists of $\Delta^1 = \{x'_0\}, \Delta^2 = \{x_0, x''_0\}$, with the interpretation $A^1 = B^1 = \emptyset$ and, for the image domain relations we get

$$r_{12} = \{(x_0', x_0)\}, \quad r_{21} = \{(x_0'', x_0')\}.$$

Since no clash occurred, the algorithm returns that $\neg_2 A \sqcup_1 B$ is satisfiable from the point of view of T_2. Indeed we have:

$$\begin{aligned}
(\neg_2 A \sqcup_1 B)^2 &= r_{12}((\neg_2 A \sqcup_1 B)^1) \\
&= r_{12}((\neg_2 A)^1 \cup B^1) \\
&= r_{12}(r_{21}(\Delta^2 \backslash A^2) \cup B^1) \\
&= r_{12}(r_{21}(\Delta^2 \backslash r_{12}(A^1)) \cup B^1) \\
&= r_{12}(r_{21}(\Delta^2) \cup B^1) \\
&= r_{12}(\{x_0'\}) \\
&= \{x_0\}.
\end{aligned}$$

5.3 Synchronization

The federated tableau algorithm depends crucially on being able to synchronize the processes occurring in each of the local reasoners. It must be ensured that, at all times during the execution of the parallel threads of the algorithm, all local processes, i.e., all applications of local expansion rules in the various local completion graphs, refer to the same sequence of non-deterministic choices. This may be achieved in a variety of different ways. One method was presented in detail in [3] in the context of P-DLs. It uses clocks, timestamps and a token to achieve synchronization and to implement efficient backtracking when a clash occurs and another non-deterministic option has to be tried.

In the present paper, we prefer to give an abstract view of the synchronization process without providing a detailed description of either the synchronizing entities or the specific steps. There are many possible choices and which one is adopted is a decision that can be relegated to the implementation of the algorithm.

The guiding principle in building the local completion graphs in our case will also be the sequence of applications of the L$\sqcup$-rules. Only one module is allowed to apply the L$\sqcup$-rule at any particular time during the execution of the algorithm. Otherwise, various other rules may be applied by the participating modules simultaneously. A newly generated concept label or role label is accompanied by a tag indicating the latest application of the L$\sqcup$-rule before its creation. New labels can be created not only by the local rules, but also by the foreign rules that involve concept and role reporting message transmissions. If a clash is detected in a local completion graph, then all labels that have been created after the last non-deterministic choice will be deleted and all nodes or edges without labels will be purged. This returns the algorithm to the same state that preceded the latest application of an L$\sqcup$-rule. Another non-deterministic choice that has not been applied before will be made and the process will start again. This "pruning operation" is necessary to restore all local completion graphs to their status just before the choice which led to the clash, or to the initial status of the local tableau, if no choice at all was ever made.

Note that local completion graphs may perform expansions on different reasoning subtasks concurrently. This improves the overall efficiency and scalability of the reasoning process. Further, note that with the introduction of messages, equality blocking in F-$\mathcal{ALCI}$ is *dynamic*: it can be established, broken and re-established. Moreover, the completeness of a local completion graph is also dynamic. A complete local completion graph may become incomplete, i.e., some expansion rules may become applicable, when a new reporting message arrives.

6 Correctness and Complexity

In order to show that the algorithm is a decision procedure for concept satisfiability in F-$\mathcal{ALCI}$, it is necessary to prove that the algorithm terminates, that the models that can be constructed from clash-free and complete federated completion graphs, generated from the algorithm, are valid with respect to the semantics of the logic (soundness) and that the algorithm always finds a model if one exists (completeness).

Termination and complexity of the algorithm is obtained by proving that there is an upper bound for the total size of all local completion graphs. We use the following notation throughout the analysis of the algorithm:

$$\begin{array}{ll} n_i = |C_{T_i}|,\ i \neq w; & n_w = |C_{T_w}| + |D|; \\ s_i = \sum_{(i,j)\in E} n_j,\ i \neq w; & s_w = n_w, \end{array}$$

where n_i is the combined length of all i-formulas in T_i in negation local form and s_i is the sum of the n_j's for all j's, such that P_j imports P_i. The reason why s_i is important in the analysis of the federated algorithm is that all j-axioms that contain an i-connective will cause a foreign rule to send a backward concept reporting or a backward role reporting message to P_i to be processed. More specifically, we have the following lemma:

Lemma 7 *Let $\Sigma = \{T_i\}_{i\in V}$ be an F-$\mathcal{ALCI}$ KB, $D \in \widehat{\mathcal{C}}_w$ and $m = \sum_{i\in V_w} |T_i|$. The F-$\mathcal{ALCI}$ tableau algorithm runs in worst case non-deterministic $O(2^m \cdot \prod_{j\in V_w} 2^{2^{s_j \log s_j}})$ time.*

Proof:
We define $f(x) = 2^{2^{x \log x}}$. We start with a set of observations:

- *For every node that has no local predecessor* (*called* **local top node** *henceforth*)*, its local descendants have a tree shape.* This observation follows from the form of the expansion rules.
- *For every local top node $j : x$, $j \neq w$, x must be a preimage of a node in another local completion graph G_i, such that $(j, i) \in E_w$.* This holds because such an x must be created by either a backward concept reporting message triggered by an application of the BCN-rule or of a foreign rule or by a backward role reporting message triggered by an application of the BRN-rule.

– *For all j, x, all local descendants of $j : x$ in G_j, that are not successors, are not preimages of nodes in any other local completion graph.* This holds because a local descendant of $j : x$, that is not a successor, is generated only by an application of the L∃-rule, while a preimage node is created only by an application of the the BCN, the BRN or a foreign rule.

Hence, 1) each local completion graph is a forest; 2) the root of every tree, i.e., a local top node, in a local completion graph, except for the root of G_w, is "copied" from, i.e., it is a preimage of, a node in another local completion graph.

Next we prove that the size of each local completion graph, hence also the total size of the "global completion graph", is limited.

First, due to subset blocking, for any local top node in G_j, the depth of its local descendant tree is bounded by $O(2^{s_j})$ and its breadth is bounded by the number of "$\exists_j$" in all $\bigcup_{(j,i)\in E} C_{T_i}$, for $j \neq w$, or in $C_{T_w} \sqcup D$, for $j = w$, which is smaller than s_j. Thus, the size of the tree is bounded by $O({s_j}^{2^{s_j}}) = O(f(s_j))$.

Since there is only acyclic importing, we can put all modules $\{T_i\}_{i\in V_w}$ in an ordered list $\mathfrak{L}$, such that $\mathfrak{L}_1 = T_w$ and each module T_i comes in $\mathfrak{L}$ before all modules $\{T_j\}_{j\in V_i}$, in a way similar to topological sorting in DAG. Let $\#(\mathfrak{L}_j)$ be the subscript of the module at $\mathfrak{L}_j$. Then, we have that the size of $G_{\#(\mathfrak{L}_j)}$ is bounded by:

$$|G_{\#(\mathfrak{L}_1)}| : O(f(s_w))$$
$$|G_{\#(\mathfrak{L}_j)}| : O\Big(\sum_{k<j} |G_{\#(\mathfrak{L}_k)}| \times f(s_{\#(\mathfrak{L}_j)})\Big), \text{ for } j > 1$$

This holds because there is only one local top node in $G_{\#(\mathfrak{L}_1)} = G_w$ (the original node), and, for every $j > 1$ and $p = \#(\mathfrak{L}_j)$, the number of local top nodes in G_p is limited by $\sum_{(p,q)\in E_w} |G_q|$, i.e., by the total size of the local completion graphs of modules that directly import T_p, since all nodes in T_p must be preimage nodes of nodes in those local completion graphs. In the worst case, $\{T_q : (p,q) \in E_w\}$ contains all modules that are before j in $\mathfrak{L}$. On the other hand, the size of a tree under a local top node in G_k is limited by $f(s_k)$.

Setting $|G_{\#(\mathfrak{L}_j)}| = t_j$ and $e_j = f(s_{\#(\mathfrak{L}_j)})$, we obtain that t_j is bounded by

$$O\big((t_1 + t_2 + ... + t_{j-1}) \times e_j\big). \tag{3}$$

Using induction, it will now be shown that t_j is bounded by

$$O\big(2^{j-2} \times e_1 \times ... \times e_j\big), \text{ for } j > 1. \tag{4}$$

By Equation (3), when $j = 2$, t_2 is bounded by $O(t_1 \times e_2) = O(e_1 \times e_2)$, whence Equation (4) holds. Let $j > 2$. Assuming, as the induction hypothesis, that, for every $1 < k < j$, Equation (4) holds, we have, by Equation (3), that t_j is bounded by

$$\begin{aligned} O\big((t_1 + t_2 + \cdots + t_{j-1}) \times e_j\big) &< O\big((e_1 + 2^0 e_1 e_2 + \cdots + 2^{j-3} e_1 e_2 \cdots e_{j-1}) e_j\big) \\ &< O\big((1 + 2^0 + \cdots + 2^{j-3}) \times e_1 e_2 \cdots e_j\big) \\ &= O\big(2^{j-2} e_1 e_2 \cdots e_j\big) \end{aligned}$$

This finishes the induction step and concludes the proof of Equation (4). Hence, the size of all local completion graphs is bounded by:

$$O\Big(e_1 + \sum_{2\leq j\leq m}\Big(2^{j-2}\prod_{k\leq j} e_j\Big)\Big) \leq O\Big(2^{m-1}\times\prod_{j\in V_w} f(s_j)\Big)$$
$$< O\Big(2^{m}\times\prod_{j\in V_w} 2^{2^{s_j\times\log s_j}}\Big)$$

■

Lemma 7 leads to the following theorem on the complexity of the federated algorithm for deciding F-$\mathcal{ALCI}$ concept satisfiability.

Theorem 8 (Termination and Complexity) *Let Σ be an F-$\mathcal{ALCI}$ ontology and $D \in \widehat{\mathcal{C}}_w$. The F-$\mathcal{ALCI}$ tableau algorithm runs in worst case 2*NExpTime *w.r.t. the size of D and the sum of the sizes of the modules in $\{T_i\}_{i\in V_w}$.*

Proof:
Let $s = \max\{s_i : i \in V_w\}$. In general, $m \ll 2^{s\log s}$. By Lemma 7, it follows that the total size of all local completion graphs is bounded by

$$O\Big(2^m\cdot 2^{m2^{(s+|D|)\log(s+|D|)}}\Big) < O\Big(2^{2^{(s+|D|)^2}}\Big).$$

■

In the following two lemmas, soundness and completeness of the F-$\mathcal{ALCI}$ algorithm are stated.

Theorem 9 (Soundness) *If the F-$\mathcal{ALCI}$ algorithm yields a complete and clash-free federated completion graph for a concept D w.r.t. a witness module T_w, then D has a federated tableau w.r.t. T_w.*

Proof:
Let $G = \{G_i\}$, with $G_i = (V_i, E_i, \mathcal{L}_i^g)$, be a complete and clash-free federated completion graph generated by the F-$\mathcal{ALCI}$ algorithm. We will obtain a tableau by "unraveling" blocked nodes and tableau relations. For a directly blocked node x, we denote by $\mathrm{bk}(x)$ the node that directly blocks x. Thus, we have $\mathcal{L}_i^g(x) \subseteq \mathcal{L}_i^g(\mathrm{bk}(x))$. We define a tableau $M = \langle\{M_i\}_{i\in V_w}, \{m_{ij}\}_{(i,j)\in E_w}\rangle$, with $M_i = \langle U_i, F_i, \mathcal{L}_i^m\rangle$, for D w.r.t. T_w in the following way:

$$\begin{aligned}
U_i &= \{x \in V_i : x \text{ is not blocked}\};\\
F_i &= E_i \restriction_{U_i^2};\\
\mathcal{L}_i^m(x) &= \mathcal{L}_i^g(x);\\
\mathcal{L}_i^m(\langle x,y\rangle) &= \mathcal{L}_i^g(\langle x,y\rangle) \cup \bigcup_{z:y=\mathrm{bk}(z)} \mathcal{L}_i^g(\langle x,z\rangle);\\
m_{ij} &= \{\langle x,y\rangle \in U_i\times U_j |\ x \in \mathrm{org}(y)\},\ \text{for } (i,j)\in E_w.
\end{aligned}$$

We show that M satisfies all tableau properties.

(D1): Since $x_0 \in V_w$, x_0 is not blocked (it does not have any ancestors), and $D \in \mathcal{L}^g_w(x_0)$, we get that $x_0 \in U_w$ and $D \in \mathcal{L}^m_w(x_0)$.

(D2): Property (D2) holds because of the D-rule.

(B1): Suppose, first, that there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $C \in \mathcal{L}^m_j(x')$. Then $x' \in U_j$ is not blocked and $x' \in \mathrm{org}(x)$, whence, since $C \in \mathcal{L}^g_j(x')$, we get, by the FCN-rule, $C \in \mathcal{L}^g_i(x)$, i.e., $C \in \mathcal{L}^m_i(x)$ and the "if" direction of Property (B1) holds.

Suppose, conversely, that $C \in \mathcal{L}^m_i(x)$. Then $C \in \mathcal{L}^g_i(x)$ and x is not blocked, whence, by the BCN-rule, there exists $x' \in V_j$, which is not blocked because it is a local top node, such that $x' \in \mathrm{org}(x)$ and $C \in \mathcal{L}^m_j(x')$. Therefore, $(x', x) \in m_{ji}$ and $C \in \mathcal{L}^g_j(x')$. Therefore, Property (B1) holds.

(B2): Suppose, first, that there exists $x', y' \in U_j$, with $(x', x), (y', y) \in m_{ji}$, such that $R \in \mathcal{L}^m_j(\langle x', y'\rangle)$. Then $x', y' \in U_j$ are not blocked and $x' \in \mathrm{org}(x), y' \in \mathrm{org}(y)$ and $R \in \mathcal{L}^g_j(\langle x', y'\rangle)$. Thus, by the FRN-rule, $R \in \mathcal{L}^g_i(\langle x, y\rangle)$, i.e., $R \in \mathcal{L}^m_i(\langle x, y\rangle)$ and the "if" direction of Property (B2) holds.

Suppose, conversely, that $R \in \mathcal{L}^m_i(\langle x, y\rangle)$. Then x is not blocked and $R \in \mathcal{L}^g_i(\langle x, y\rangle)$. Thus, by the BRN-rule, there exists $x', y' \in V_j$, with $x' \in \mathrm{org}(x)$ and $y' \in \mathrm{org}(y)$, such that $R \in \mathcal{L}^g_j(\langle x', y'\rangle)$ and x', y' cannot be blocked. Hence $R \in \mathcal{L}^m_j(\langle x', y'\rangle)$ and Property (B2) holds.

(N1): This property follows directly by the hypothesis that G is a clash-free federated completion graph.

(N2): Suppose $\neg_i C \in \mathcal{L}^m_i(x)$ is of type 2 with trace j and that $x' \in U_j$, with $(x', x) \in m_{ji}$. Then $\neg_i C \in \mathcal{L}^g_i(x)$, $x' \in V_j$, with $x' \in \mathrm{org}(x)$ and neither x nor x' are blocked. Hence, by the L$\neg$-rule, $\neg_j C \in \mathcal{L}^g_j(x')$. Therefore, $\neg_j C \in \mathcal{L}^m_j(x')$ and Property (N2) holds.

(N3): Suppose that $\neg_j C \in \mathcal{L}^m_i(x)$. Then $\neg_j C \in \mathcal{L}^g_i(x)$. Thus, by the F$\neg$-rule, there exists $x' \in V_j$, with $x' \in \mathrm{org}(x)$, such that $\neg_j C \in \mathcal{L}^g_j(x')$. This shows that, there exists $x' \in U_j$, such that $(x', x) \in m_{ji}$ and $\neg_j C \in \mathcal{L}^m_j(x')$. So Property (N3) holds.

(A1): Suppose that $C_1 \sqcap_j C_2 \in \mathcal{L}^m_i(x)$. Then $C_1 \sqcap_j C_2 \in \mathcal{L}^g_i(x)$. Therefore, if $j = i$, by the L$\sqcap$-rule, we get that $C_1, C_2 \in \mathcal{L}^g_i(x)$, whence $C_1, C_2 \in \mathcal{L}^m_i(x)$. On the other hand, if $j \neq i$, then, by the F$\sqcap$-rule, there exists $x' \in V_j$, with $x' \in \mathrm{org}(x)$, such that $C_1 \sqcap_j C_2 \in \mathcal{L}^g_j(x')$. Hence, by the previous case, we get that $C_1, C_2 \in \mathcal{L}^g_j(x')$, showing that, there exists $x' \in U_j$, with $(x', x) \in m_{ji}$, such that $C_1, C_2 \in \mathcal{L}^m_j(x')$. Thus, Property (A1) holds.

(A2): The proof of this case is very similar to that of Property (A1).

(A3): Suppose that $\forall_j R.C \in \mathcal{L}^m_i(x)$. Then $\forall_j R.C \in \mathcal{L}^g_i(x)$. If $j = i$ and $R \in \mathcal{L}^m_i(\langle x, y\rangle)$, then we have that $R \in \mathcal{L}^g_i(\langle x, y\rangle)$, whence, by the L$\forall$-rule, $C \in \mathcal{L}^g_i(y)$, showing that $C \in \mathcal{L}^m_i(y)$. If, on the other hand, $j \neq i$, we get, by the F$\forall$-rule, that there exists $x' \in V_j$, with $x' \in \mathrm{org}(x)$, with $\forall_j R.C \in \mathcal{L}^g_j(x')$. Thus, by the L$\forall$-rule, as applied in the previous case, for all $y' \in U_j$, with $R \in \mathcal{L}^m_j(\langle x', y'\rangle)$, we get that $C \in \mathcal{L}^m_j(y')$.

(A4): Suppose that $\exists_j R.C \in \mathcal{L}^m_i(x)$. Then $\exists_j R.C \in \mathcal{L}^g_i(x)$. If $j = i$, then, by the L$\exists$-rule, there exists $y \in V_i$, such that $R \in \mathcal{L}^g_i(\langle x, y\rangle)$, with $C \in \mathcal{L}^g_i(y)$. Thus, in this case, $R \in \mathcal{L}^m_i(\langle x, y\rangle)$ and $C \in \mathcal{L}^m_i(y)$. If, on the other hand, $j \neq i$, we get, by the F$\forall$-rule, that there exists $x' \in V_j$, with $x' \in \mathrm{org}(x)$, with

$\exists_j R.C \in \mathcal{L}_j^g(x')$. Thus, by the L$\exists$-rule, as applied in the previous case, there exists $y' \in U_j$, with $R \in \mathcal{L}_j^g(\langle x', y'\rangle)$ and $C \in \mathcal{L}_j^g(y')$. Hence, we get that $R \in \mathcal{L}_j^m(\langle x', y'\rangle)$ and $C \in \mathcal{L}_j^m(y')$.

■

The following lemma shows that the federated algorithm is complete, i.e., that it always finds a complete and clash-free federated completion graph whenever there exists a federated tableau.

Theorem 10 (Completeness) *If a concept D has a federated tableau w.r.t. to a witness module T_w of an F-$\mathcal{ALCI}$ KB $T = \{T_i\}_{i\in V}$, then the F-$\mathcal{ALCI}$ algorithm produces a complete and clash-free federated completion graph for D w.r.t. T_w.*

Proof:

Let $M = \langle \{M_i\}_{i\in V_w}, \{m_{ij}\}_{(i,j)\in E_w}\rangle$, with $M_i = \langle U_i,\ F_i, \mathcal{L}_i^m\rangle$, be a tableau for D w.r.t. T_w. We will use M to guide the application of the non-deterministic L$\sqcup$-rule in a way that yields a complete and clash-free federated completion graph $G = \{G_i\}$, with $G_i = (V_i, E_i, \mathcal{L}_i^g)$.

To construct G, we start with a single node x_0 in the local tableau M_w, with $D \in \mathcal{L}_w^m(x_0)$. Such an x_0 exists, since M is a tableau for D w.r.t. T_w. Let $\pi \subseteq \bigcup_{i\in V_w}(V_i \times U_i)$ be a function that maps all individuals in local completion graphs to individuals in corresponding local tableaux. Initially, we have $V_w = \{x_0\}$, $\mathcal{L}_w^g(x_0) = \{D\}$, $\pi(x_0) = x_0$ and all G_i, $i \neq w$, being empty. Next, we apply F-$\mathcal{ALCI}$ expansion rules to extend G and π, in such a way that the following conditions always (inductively) hold:

$$\begin{cases} \mathcal{L}_i^g(x) \subseteq \mathcal{L}_i^m(\pi(x)) \\ \text{if } R \in \mathcal{L}_i^g(\langle x, y\rangle), \text{ then } R \in \mathcal{L}_i^m(\langle \pi(x), \pi(y)\rangle) \\ \text{if } x \in \mathrm{org}(y) \text{ in } G, \text{ then } \langle \pi(x), \pi(y)\rangle \in m_{ij} \text{ in } T, \text{ for } (i,j) \in E_w \end{cases} \tag{5}$$

- D-rule: **if** $C_{T_i} \notin \mathcal{L}_i^g(x)$, **then** $\mathcal{L}_i^g(x) {+}{=} \{C_{T_i}\}$. Since, by Property (D2), $C_{T_i} \in \mathcal{L}_i^m(\pi(x))$, this rule can be applied without violating Conditions (5).
- FCN-rule: **if** $C \in \mathcal{L}_j^g(x)$, x is not blocked, **then** transmit $r^{j\to i}(x, C)$, i.e., if there exists $x' \in V_i$, such that $x \in \mathrm{org}(x')$, then $C \in \mathcal{L}_i^g(x')$. In that case, by the induction hypothesis, $C \in \mathcal{L}_j^m(\pi(x))$ and $(\pi(x), \pi(x')) \in m_{ji}$, whence by Property (B1), we obtain that $C \in \mathcal{L}_i^m(\pi(x'))$. Thus, Conditions (5) are not violated.
- BCN-rule: **if** $C \in \mathcal{L}_i^g(x)$, **then** transmit $r^{j\leftarrow i}(x, C)$. This will create and $x' \in V_j$, with $x' \in \mathrm{org}(x)$ and $C \in \mathcal{L}_j^g(x')$. Since $C \in \mathcal{L}_i^g(x)$, we get that $C \in \mathcal{L}_i^m(\pi(x))$, whence, by Property (B1) of a federated tableau, there exists $z \in U_j$, with $(\pi(x), z) \in m_{ji}$, with $C \in \mathcal{L}_j^m(z)$. Set $\pi(x') = z$. Then we have that $\mathcal{L}_j^g(x') = \{C\} \subseteq \mathcal{L}_j^m(z) = \mathcal{L}_j^m(\pi(z))$. Moreover, we get $(\pi(x), \pi(x')) = (\pi(x), z) \in m_{ji}$ and, therefore, Conditions (5) are not violated.
- FRN-rule: **if** $R \in \mathcal{L}_j^g(\langle x, y\rangle)$, x or y not blocked, **then** transmit $r^{j\to i}(\langle x, y\rangle, R)$, i.e., if there exist x', y', such that $x \in \mathrm{org}(x'), y \in \mathrm{org}(y')$, then $R \in \mathcal{L}_j^g(\langle x', y'\rangle)$. If $R \in \mathcal{L}_j^g(\langle x, y\rangle)$, then $R \in \mathcal{L}_i^m(\langle \pi(x), \pi(y)\rangle)$ and, if $x \in$

org(x'), $y \in$ org(y'), then $(\pi(x), \pi(x')), (\pi(y), \pi(y')) \in m_{ij}$, whence, by the tableau Property (B2), we must have $R \in \mathcal{L}_j^m(\langle \pi(x'), \pi(y') \rangle)$, whence Property (5) is not violated.

- BRN-rule: **if** $R \in \mathcal{L}_i^g(\langle x, y \rangle)$ and x or y are not blocked, **then** transmit $r^{j \leftarrow i}(\langle x, y \rangle, R)$, i.e., create $x', y' \in V_j$, with $x' \in$ org(x), $y' \in$ org(y), such that $R \in \mathcal{L}_j^g(\langle x', y' \rangle)$. Since $R \in \mathcal{L}_i^g(\langle x, y \rangle)$, we get that $R \in \mathcal{L}_i^m(\langle \pi(x), \pi(y) \rangle)$. Therefore, by Property (B2), there exists $z, w \in U_j$, with $(z, \pi(x)) \in m_{ji}$ and $(w, \pi(y)) \in m_{ji}$, such that $R \in \mathcal{L}_j^m(\langle z, w \rangle)$. Set $\pi(x') = z$ and $\pi(y') = w$. Then, we get that $(\pi(x'), \pi(x)), (\pi(y'), \pi(y)) \in m_{ji}$ and $R \in \mathcal{L}_j^m(\langle \pi(x'), \pi(y') \rangle)$. Thus, Conditions (5) are not violated.
- L$\neg$-rule: **if** $\neg_i C \in \mathcal{L}_i^g(x)$ is of type 2 with trace j and x is not blocked, **then** transmit $r^{j \dashleftarrow i}(x, \neg_j C)$, i.e., if there exists $x' \in V_j$, with $x' \in$ org(x), then $\neg_j C \in \mathcal{L}_j^g(x')$. Under these circumstances, we have, by the induction hypothesis, that $\neg_i C \in \mathcal{L}_i^m(\pi(x))$ and $(\pi(x'), \pi(x)) \in m_{ji}$. Thus, by Property (N2), we get that $\neg_j C \in \mathcal{L}_j^m(\pi(x'))$, showing that Conditions (5) are not violated.
- F$\neg$-rule: **if** $\neg_j C \in \mathcal{L}_i^g(x)$ and x is not blocked, **then** transmit $r^{j \leftarrow i}(x, \neg_j C)$, i.e., create $x' \in V_j$, with $x' \in$ org(x), such that $\neg_j C \in \mathcal{L}_j^g(x')$. By the induction hypothesis, we have that $\neg_j C \in \mathcal{L}_i^m(\pi(x))$. Thus, by Property (N3), we get that, there exists $z \in U_j$, with $(z, \pi(x)) \in m_{ji}$, such that $\neg_j C \in \mathcal{L}_j^m(z)$. If we set $\pi(x') = z$, we get that $(\pi(x'), \pi(x)) = (z, \pi(x)) \in m_{ji}$ and $\neg_j C \in \mathcal{L}_j^m(\pi(x'))$. Hence, Conditions (5) are not violated.
- L$\sqcap$-rule: **if** $C_1 \sqcap_i C_2 \in \mathcal{L}_i^g(x)$ and x is not blocked, **then** $C_1, C_2 \in \mathcal{L}_i^g(x)$. In this case, by the induction hypothesis, $C_1 \sqcap_i C_2 \in \mathcal{L}_i^m(\pi(x))$. Thus, by Property (A1), we get that $C_1, C_2 \in \mathcal{L}_i^m(\pi(x))$, which shows that Conditions (5) are not violated.
- F$\sqcap$-rule: **if** $C_1 \sqcap_j C_2 \in \mathcal{L}_i^g(x)$ and x is not blocked, **then** transmit $r^{j \leftarrow i}(x, C_1 \sqcap_j C_2)$, i.e., create $x' \in V_j$, with $x' \in$ org(x), such that $C_1 \sqcap_j C_2 \in \mathcal{L}_j^g(x')$. In this case, by the induction hypothesis, $C_1 \sqcap_j C_2 \in \mathcal{L}_i^m(\pi(x))$. Thus, by Property (A1), there exists $z \in U_j$, with $(z, \pi(x)) \in m_{ji}$, such that $C_1 \sqcap_j C_2 \in \mathcal{L}_j^m(z)$. Hence, if we set $\pi(x') = z$, we get that $(\pi(x'), \pi(x)) = (z, \pi(x)) \in m_{ji}$ and $C_1 \sqcap_j C_2 \in \mathcal{L}_j^m(\pi(x'))$. Therefore, Conditions (5) are not violated.
- L$\sqcup$-rule: **if** $C_1 \sqcup_i C_2 \in \mathcal{L}_i^g(x)$ and x is not blocked, **then** $C_1 \in \mathcal{L}_i^g(x)$ or $C_2 \in \mathcal{L}_i^g(x)$. In this case, by the induction hypothesis, $C_1 \sqcup_i C_2 \in \mathcal{L}_i^m(\pi(x))$. Thus, by Property (A2), we get that $C_1 \in \mathcal{L}_i^m(\pi(x))$ or $C_1 \in \mathcal{L}_i^m(\pi(x))$, which shows that Conditions (5) are not violated.
- F$\sqcup$-rule: **if** $C_1 \sqcup_j C_2 \in \mathcal{L}_i^g(x)$ and x is not blocked, **then** transmit $r^{j \leftarrow i}(x, C_1 \sqcup_j C_2)$, i.e., create $x' \in V_j$, with $x' \in$ org(x), such that $C_1 \sqcup_j C_2 \in \mathcal{L}_j^g(x')$. In this case, by the induction hypothesis, $C_1 \sqcup_j C_2 \in \mathcal{L}_i^m(\pi(x))$. Thus, by Property (A2), there exists $z \in U_j$, with $(z, \pi(x)) \in m_{ji}$, such that $C_1 \sqcup_j C_2 \in \mathcal{L}_j^m(z)$. Hence, if we set $\pi(x') = z$, we get that $(\pi(x'), \pi(x)) = (z, \pi(x)) \in m_{ji}$ and $C_1 \sqcup_j C_2 \in \mathcal{L}_j^m(\pi(x'))$. Therefore, Conditions (5) are not violated.
- L$\forall$-rule: **if** $\forall_i R.C \in \mathcal{L}_i^g(x)$, x is not blocked, and there exists $y \in V_i$, with $R \in \mathcal{L}_i^g(\langle x, y \rangle)$, **then** $C \in \mathcal{L}_i^g(y)$. In this case, by the induction hypothesis, we get that $\forall_i R.C \in \mathcal{L}_i^m(\pi(x))$ and $R \in \mathcal{L}_i^m(\langle \pi(x), \pi(y) \rangle)$, whence, by Property (A3), $C \in \mathcal{L}_i^m(\pi(y))$. Hence, Conditions (5) are not violated.

- F$\forall$-rule: **if** $\forall_j R.C \in \mathcal{L}_i^g(x)$ and x is not blocked, **then** transmit $r^{j\leftarrow i}(x, \forall_j R.C)$, i.e., create $x' \in V_j$, with $x' \in \mathrm{org}(x)$, such that $\forall_j R.C \in \mathcal{L}_j^g(x')$. If $\forall_j R.C \in \mathcal{L}_i^g(x)$, then, by the induction hypothesis, $\forall_j R.C \in \mathcal{L}_m^g(x)$, whence, by Property (A3), there exists $z \in U_j$, with $(z, \pi(x)) \in m_{ji}$ such that, for all $w \in U_j$, with $R \in \mathcal{L}_j^m(\langle z, w\rangle)$, $C \in \mathcal{L}_j^m(w)$. Set $\pi(x') = z$. Then we have that $(\pi(x'), \pi(x)) = (z, \pi(x)) \in m_{ji}$ and, by the previous case, $\forall_j R.C \in \mathcal{L}_j^m(\pi(x'))$.
- L$\exists$-rule: **if** $\exists_i R.C \in \mathcal{L}_i^g(x)$, x is not blocked, and there does not exist $y \in V_i$, with $R \in \mathcal{L}_i^g(\langle x, y\rangle)$ and $C \in \mathcal{L}_i^g(y)$, **then** create such a y. In this case, by the induction hypothesis, we get that $\exists_i R.C \in \mathcal{L}_i^m(\pi(x))$, whence, by Property (A4), there exists $z \in U_i$, such that $R \in \mathcal{L}_i^m(\langle \pi(x), z\rangle)$ and $C \in \mathcal{L}_i^m(z)$. Set $\pi(y) = z$. Then $R \in \mathcal{L}_i^m(\langle \pi(x), \pi(y)\rangle)$ and $C \in \mathcal{L}_i^m(\pi(y))$. Hence, Conditions (5) are not violated.
- F$\exists$-rule: **if** $\exists_j R.C \in \mathcal{L}_i^g(x)$ and x is not blocked, **then** transmit $r^{j\leftarrow i}(x, \exists_j R.C)$, i.e., create $x' \in V_j$, with $x' \in \mathrm{org}(x)$, such that $\exists_j R.C \in \mathcal{L}_j^g(x')$. If $\exists_j R.C \in \mathcal{L}_i^g(x)$, then, by the induction hypothesis, $\exists_j R.C \in \mathcal{L}_m^g(x)$, whence, by Property (A4), there exist $z, w \in U_j$, with $(z, \pi(x)) \in m_{ji}$, $R \in \mathcal{L}_j^m(\langle z, w\rangle)$ and $C \in \mathcal{L}_j^m(w)$. Set $\pi(x') = z$. Then we have that $(\pi(x'), \pi(x)) = (z, \pi(x)) \in m_{ji}$ and, by the previous case, $\exists_j R.C \in \mathcal{L}_j^m(\pi(x'))$. Thus, Conditions (5) are not violated in this case either.

G must be clash-free, since, if there existed i, x, C, such that $\{C, \neg_i C\} \subseteq \mathcal{L}_i^g(x)$, then, by Conditions (5), $\{C, \neg_i C\} \subseteq \mathcal{L}_i^m(\pi(x))$, which would contradict tableau Property (N1) for M. Hence, whenever an expansion rule is applicable to G, it can be applied in such a way that maintains Conditions (5). By Lemma 7, any sequence of rule applications must terminate. Hence, we will obtain a complete and clash-free completion graph G for D from M. ∎

By combining Theorems 8, 9 and 10, we obtain the following theorem, which is the main result of the paper.

Theorem 11 *Let Σ be an F-$\mathcal{ALCI}$ ontology and $D \in \widehat{\mathcal{C}}_w$. The F-$\mathcal{ALCI}$ tableau algorithm is a sound, complete, and terminating decision procedure for satisfiability of D as witnessed by T_w. This decision procedure is in* 2NExpTime *w.r.t. the size of D and the sum of the sizes of the modules in $\{T_i\}_{i \in V_w}$.*

7 Summary and Discussion

Many semantic web applications require support for knowledge representation and inference over a federation of multiple autonomous ontology modules, without having to combine them in one location. Federated $\mathcal{ALCI}$ or F-$\mathcal{ALCI}$ is a modular description logic, each of whose modules is roughly an $\mathcal{ALCI}$ ontology. F-$\mathcal{ALCI}$ supports importing of both concepts and roles across modules as well as contextualized interpretation of logical connectives. We have presented a federated tableau algorithm for deciding satisfiability of a concept expression from a specific module's point of view in F-$\mathcal{ALCI}$. We have shown that the algorithm is sound and complete and that its worst-case running time is non-deterministic doubly exponential with respect to the size of the input concept and the sum of the sizes of all modules in the federated ontology. From the complexity-theoretic point of view, this is equivalent to being non-deterministic doubly exponential with respect to the size of the input concept and the size of the largest module in the federated ontology, since the number of modules is assumed to be fixed. In the non-federated case, several tableau-based algorithms with high complexity upper bounds have been optimized to perform well in practice [1]. We are currently in the process of implementing the federated algorithm. Experimentation and further optimizations may lead to a practically useful federated F-$\mathcal{ALCI}$ reasoner.

Acknowledgement This research was supported in part by the grant IIS 0639230 from the National Science Foundation.

References

1. Franz Baader and Ulrike Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69(1):5–40, 2001.
2. Jie Bao, Doina Caragea, and Vasant Honavar. A tableau-based federated reasoning algorithm for modular ontologies. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 404–410. IEEE Press, 2006.
3. Jie Bao, George Voutsadakis, Giora Slutzki, and Vasant Honavar. Distributed reasoning with modular $\mathcal{ALCP}_C$ ontologies. 2008.
4. Jie Bao, George Voutsadakis, Giora Slutzki, and Vasant Honavar. *Ontology Modularization*, chapter Package based Description Logics. Berlin: Springer (In Press), 2008.
5. Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. Owl web ontology language reference. Technical report, W3C, 2004.
6. T Berners-Lee, J Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
7. Alexander Borgida and Luciano Serafini. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics*, 1:153–184, 2003.
8. Bernardo Cuenca Grau and Bijan Parsia. From shoq(d) toward e-connections. In *Description Logics*, 2004.
9. Bernardo Cuenca Grau, Bijan Parsia, and Evren Sirin. Tableau algorithms for e-connections of description logics. Technical report, University of Maryland Institute for Advanced Computer Studies (UMIACS), TR 2004-72, 2004.
10. Bernardo Cuenca Grau, Bijan Parsia, and Evren Sirin. Working with multiple ontologies on the semantic web. In McIlraith et al. [16], pages 620–634.
11. Volker Haarslev and Ralf Möller. Racer system description. In *IJCAR*, pages 701–706, 2001.
12. Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL: the making of a web ontology language. *J. Web Sem.*, 1(1):7–26, 2003.
13. Ian Horrocks and Ulrike Sattler. A tableau decision procedure for $\mathcal{SHOIQ}$. *J. Autom. Reasoning*, 39(3):249–276, 2007.

14. Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Practical reasoning for expressive description logics. In *LPAR*, pages 161–180, 1999.
15. Oliver Kutz, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. E-connections of abstract description systems. *Artif. Intell.*, 156(1):1–73, 2004.
16. Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors. *The Semantic Web - ISWC 2004: Third International Semantic Web Conference,Hiroshima, Japan, November 7-11, 2004. Proceedings*, volume 3298 of *Lecture Notes in Computer Science*. Springer, 2004.
17. Peter F. Patel-Schneider and Ian Horrocks. Dlp and fact. In TABLEAUX, pages 19–23, 1999.
18. Marie-Christine Rousset. Small can be beautiful in the semantic web. In McIlraith et al. [16], pages 6–16.
19. Luciano Serafini, Alexander Borgida, and Andrei Tamilin. Aspects of distributed and modular ontology reasoning. In *IJCAI*, pages 570–575, 2005.
20. Luciano Serafini and Andrei Tamilin. Local tableaux for reasoning in distributed description logics. In *Description Logics Workshop 2004, CEUR-WS Vol 104*, 2004.
21. Luciano Serafini and Andrei Tamilin. Drago: Distributed reasoning architecture for the semantic web. In *ESWC*, pages 361–376, 2005.
22. Evren Sirin and Bijan Parsia. Pellet: An OWL DL Reasoner. In *Description Logics Workshop*, 2004.
23. Heiner Stuckenschmidt and Michel C. A. Klein. Integrity and change in modular ontologies. In *IJCAI*, pages 900–908, 2003.
24. George Voutsadakis, Jie Bao, Giora Slutzki, and Vasant Honavar. Federated $\mathcal{ALCI}$: Preliminary report. *WI* 2008*, To appear*, 2008.