Reasoning Theory for D3L with Compositional Bridge Rules

Zhao Xiaofei[1], Tian Dongping[1], Chen Limin[2], and Shi Zhongzhi[1]

[1] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
zhaoxf@ics.ict.ac.cn
[2] Research Institute of China Unicom, Beijing, China

Abstract. The semantic mapping in Distributed Dynamic Description Logics(D3L) allows knowledge to propagate from one ontology to another. The current research for knowledge propagation in D3L is only for a simplified case when only two ontologies are involved. In this paper we study knowledge propagation in more complex cases. We find in the case when more than two ontologies are involved and bridge rules form chains, knowledge does not always propagate along chains of bridge rules even if we would expect it. Inspired by Package-based description Logics, we extend the original semantics of D3L by imposing so called compositional consistency condition on domain relations in D3L interpretations. Under this semantics knowledge propagates along chains of bridge rules correctly. Furthermore we provide a distributed Tableaux reasoning algorithm for deciding satisfiability of concepts which is decidable in D3L under compositional consistency. Compared with original one, the extended D3L provides more reasonable logic foundation for distributed, dynamic system such as the information integration system and the Semantic Web.

Keywords. Distributed Dynamic Description Logics(D3L), Knowledge Propagation, Compositional Consistency, Bridge Rule Chain

1 Introduction

As the distributed extension of traditional dynamic description logics(DDL)[1][2], distributed dynamic description logics(D3L)[3][4] enables reasoning with multiple heterogeneous DDL ontologies interconnected by directional semantic mapping. D3L captures the idea of importing and reusing knowledge between several ontologies. This idea combines well with the basic assumption of the distributed and dynamic system such as the Semantic Web and the heterogeneous information integration system.

An important innovation of D3L is to describe the semantic mapping between heterogeneous ontologies through bridge rules[3]. Bridge rules can assert that the concept/relation/action, local to ontology T_1, is mapped to an independent ontology T_2 as a subconcept/relation/action or superconcept/relation/action. Bridge rules are directed, and hence if there is a bridge rule with direction from T_1 to T_2, then T_2 reuses knowledge from T_1 but not necessarily the other way around. Bridge rules lead to the important characteristic--knowledge dissemination which makes D3L different from DDL. Knowledge propagation theory of D3L has been studied in our early work, but only for a simplied case when only two DDL ontologies are involved and bridge rules doesn't construct chain. Because D3L is presented primarily as the logical foundation for complex distributed dynamic ontology, the research for knowledge propagation in more complex cases has important theoretical value and practical value.

In this paper we study subsumption propagation theory in more complex cases, when bridge rules in multiple DDL ontologies construct chains. We discover that original semantics of D3L can't assure knowledge propagate correctly in that situations even if we would expect it. Inspired by Package-based description Logics, we extend the original semantics of D3L by imposing so called compositional consistency condition on domain relations in D3L interpretations. Under this semantics knowledge propagates along chains of bridge rules correctly. Then we study the general laws of knowledge propagation under the new semantics. At last, we study the decidability for reasoning satisfiability of concepts in D3L under compositional consistency and provide a distributed Tableaux reasoning algorithm for deciding it. The algorithm can reason with acyclic D3L knowledge bases build on top of D-ALC as the local language.

2 Demonstrations of the Problem

By two examples, this section will illustrate the problem which occurs in D3L knowledge propagation in the case of bridge rule chains exist.

Example 1. In DDL ontology T_1 there are action *BuyVehicle* and its subaction *BuyCar*. There is also another DDL ontology T_2 with action *Buy* and *BuyFord*. An into-action bridge rule maps from *BuyVehicle* to *Buy* and states that *BuyVehicle* is a subaction of *Buy*. An onto-action bridge rule maps from *BuyCar* to *BuyFord* and states that *BuyCar* is a superaction of *BuyFord*. Thanks to action subsumption propagation theorem[4], the knowledge that *BuyCar* is a subaction of *BuyVehicle* propagates to T_2 and we infer that *BuyFord* is a subaction of *Buy* and other inferences such as action realizability.

Consider a change in the situation, T_1 is divided into T_1 and T_1', as depicted in Figure 1, T_1' contains *BuyCar*, and there is an into-action bridge rule from T_1' to T_1 stating that *BuyCar* is a subaction of *BuyVehicle*. Actions in T_2 are unchanged, but the two bridge rules now map, first: from T_1 to T_2 asserting *BuyVehicle* a subaction of *Buy*, and second from T_1' to T_2: asserting *BuyCar* a superaction of *BuyFord*. In D3L we no longer infer that *BuyFord* is a subaction of *Buy*.

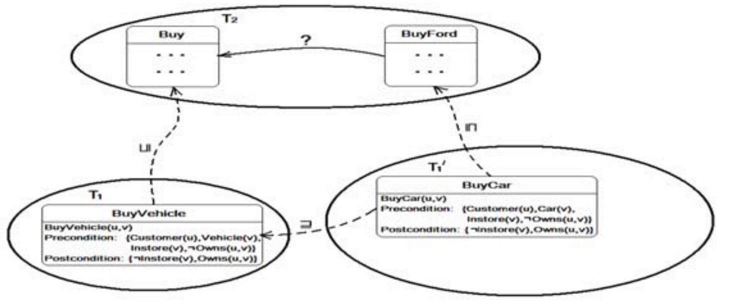
T₂
Buy
...
...
?
BuyFord
...
...
⊑
⊒
T₁′
BuyCar
BuyCar(u,v)
Precondition: {Customer(u),Car(v), Instore(v),¬Owns(u,v)}
Postcondition: {¬Instore(v),Owns(u,v)}
T₁
BuyVehicle
BuyVehicle(u,v)
Precondition: {Customer(u),Vehicle(v), Instore(v),¬Owns(u,v)}
Postcondition: {¬Instore(v),Owns(u,v)}
⊒

Fig. 1. Depiction of distributed ontologies in example 1

Example 2. There are three DDL ontologies T_1, T_2, T_3 in D3L knowledge base, as depicted in Figure 2(a), by applying concept subsumption propagation theorem repeatedly we can infer that *MyMastiff* is a kind of *DangerousAnimal*. This situation can be extended to another one where there are many similar intermediate ontologies as T_2 between T_1 and T_3. In such cases knowledge can be correctly propagated along bridge rule chains. However, the restriction condition in above cases is too strict, if conditions are violated, *Carnivore* and *HouseDog* are located in different ontology, as depicted in Figure 2(b), we can't infer that *MyMastiff* is a subconcept of *DangerousAnimal* although the superconcept-subconcept relation between *Canidae* and *Canis* should be propegated to *Carnivore* and *HouseDog*, and then to *DangerousAnimal* and *MyMastiff* again.

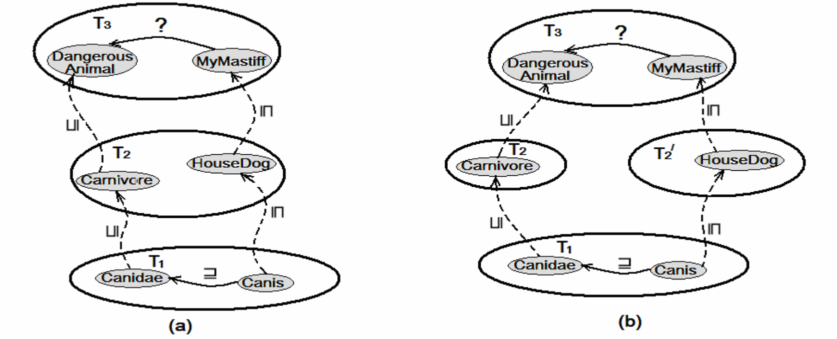

Fig. 2. Depiction of distributed ontologies in example 2

From above two examples we can see, the traditional semantics of D3L does not guarantee the knowledge that we hope to import into the target ontology spreads along the bridge rule chains correctly. In section 4, we will give a detailed analysis of the reasons, and extend D3L to solve this problem.

3 Distributed Dynamic Description Logics

A basic form of D3L knowledge base dubbed distributed *Tbox*, includes a set of local *Tboxes* each over its own DDL language and a set of bridge rules that provides mappings between local *Tboxes*. In this work, we use the formal definition as follows.

Definition 1. Assume a DDL language DDL$_X$, a non-empty index set I, a set of concept names $N_C=\cup_{i\in I}N_{ci}$, a set of relation names $N_R=\cup_{i\in I}N_{Ri}$ and a set of action names $N_A=\cup_{i\in I}N_{Ai}$, A Distributed Tbox over DDL$_X$ is a pair $<\{T_i\}_{i\in I}, BR>$ such that: each local *Tbox* T_i is a collection of general inclusions over N_{Ci}, N_{Ri} and N_{Ai} in the local language of T_i, a sub-language of DDL$_X$; the set of bridge rules *BR* divides into sets of bridge rules $BR=\cup_{i,\ j\in I,\ i\neq j}\ BR_{ij}$. Each BR_{ij} is a collection of bridge rules in direction from T_i to T_j which are of four forms:

$i: C\xrightarrow{\sqsubseteq}j: E$ （into-concept（relation） bridge rule）;

$i: C\xrightarrow{\sqsupseteq}j: E$ （onto- concept（relation） bridge rule）;

$i: \alpha\xrightarrow{\sqsubseteq}j: \beta$ （into-action bridge rule）;

$i: \alpha\xrightarrow{\sqsupseteq}j: \beta$ （onto-action bridge rule）.

As we have already mentioned in the introduction, the direction of bridge rules matters and hence B_{ij} and B_{ji} are possibly and expectedly distinct. The bridge graph $G_{DTB}=<V, E>$ of a distributed *Tbox DTB* is defined as follows: $V=I$ and $<i, j>\in E$ if $BR_{ij}\neq\emptyset$. We say that *DTB* is acyclic if G_{DTB} is acyclic.

Given a *Tbox T* , a hole is an interpretation $I^{\epsilon}=<\emptyset, \bullet^{\epsilon}>$ with empty domain. Holes are used for fighting propagation of inconsistency. A distributed interpretation $DI=<\{I_i\}_{i\in I}, \{r_{ij}\}_{i,\ j\in I,\ i\neq j}>$ of a distributed *Tbox DTB* consists of a set of local interpretations $\{I_i\}_{i\in I}$ and a set of domain relations $\{r_{ij}\}_{i,\ j\in I,\ i\neq j}$. For each $i\in I$, either $I_i=(\Delta^{Ii}, \bullet^{Ii})$ is an interpretation of local Tbox or $I_i=I^{\epsilon}$ is a hole. Each domain relation r_{ij} is a subset of $\Delta^{Ii}\times\Delta^{Ij}$. We denote by $r_{ij}(d)$ the set $\{d'|<d, d'>\in r_{ij}\}$ and by $r_{ij}(D)$ the set $\cup_{d\in D}r_{ij}(d)$.

Definition 2. For every $i, j\in I$, a distributed interpretation *DI* satisfies the elements of a distributed *Tbox DTB* （denoted by $DI\vDash_d\bullet$） according to the following clauses:

1. $DI\vDash_d i: C\xrightarrow{\sqsubseteq}j: E$ if $r_{ij}(C^{Ii})\subseteq E^{Ij}$;
2. $DI\vDash_d i: C\xrightarrow{\sqsupseteq}j: E$ if $r_{ij}(C^{Ii})\supseteq E^{Ij}$;
3. $DI\vDash_d i: \alpha\xrightarrow{\sqsubseteq}j: \beta$ if $r_{ij}(\alpha^{Ii})\subseteq\beta^{Ij}$;
4. $DI\vDash_d i: \alpha\xrightarrow{\sqsupseteq}j: \beta$ if $r_{ij}(\alpha^{Ii})\supseteq\beta^{Ij}$;
5. $DI\vDash_d i: C\sqsubseteq E$ if $I_i\vDash C\sqsubseteq E$;
6. $DI\vDash_d i: \alpha\sqsubseteq\beta$ if $I_i\vDash\alpha\sqsubseteq\beta$;
7. $DI\vDash_d T_i$ if $I_i\vDash T_i$;
8. $DI\vDash_d BR$ if $DI\vDash_d br$ for every $br\in BR$;
9. $DI\vDash_d DTB$ if $DI\vDash_d T_i$ and $DI\vDash_d BR$ for every $i\in I$;
10. $DTB\vDash_d i: C\sqsubseteq E$ if $DI\vDash_d DTB$ implies $DI\vDash_d i: C\sqsubseteq E$ for every DI;
11. $DTB\vDash_d i: \alpha\sqsubseteq\beta$ if $DI\vDash_d DTB$ implies $DI\vDash_d i: \alpha\sqsubseteq\beta$ for every DI.

If $DI \vDash_d DTB$ then we say that *DI* is a distributed model of *DTB*.

Among the properties of D3L we find the characterization of subsumption propagation, which formally describes the mechanism of knowledge reuse of D3L. Theorem 1 constitutes the most basic version of this property: thanks to a pair of one into-bridge rule and one onto-bridge rule local subsumption relationship is propagated from the source ontology of these bridge rules to the target ontology.

Theorem 1. In each distributed *Tbox DTB*, if $i: C \xrightarrow{\sqsupseteq} j: G \in BR$ and $i: D \xrightarrow{\sqsubseteq} j: H \in BR$, then the following holds: $DTB \vDash_d i: C \sqsubseteq D \Rightarrow DTB \vDash_d j: G \sqsubseteq H$; if $i: \alpha \xrightarrow{\sqsupseteq} j: \beta \in BR$ and $i: \pi \xrightarrow{\sqsubseteq} j: \rho \in BR$, then the following holds: $DTB \vDash_d i: \alpha \sqsubseteq \pi \Rightarrow DTB \vDash_d j: \beta \sqsubseteq \rho$. （Subsumption propagation theorem）

Theorem 1 states subsumption propagation property, one of the D3L properties. So far subsumption propagation laws in more complex cases have been studied. However all these cases do not extend the case captured by Theorem 1 in that aspect that only two local DDL ontologies that are directly connected with bridge rules are studied. The reader is kindly redirected to reference [4] for all details and discussion.

4 Knowledge Propagation in D3L under Compositional Consistency

Why does not the knowledge propagate along bridge rule chains as we would expect in the examples shown in section 2? In the following we will give the detailed analysis from semantic perspective. There are three bridge rules in Example 1: $1': BuyCar \xrightarrow{\sqsupseteq} 2: BuyFord \in BR$, $1: BuyVehicle \xrightarrow{\sqsubseteq} 2: Buy \in Br$ and $1': BuyCar \xrightarrow{\sqsubseteq} 1: BuyVehicle \in BR$. Given a model of *DTB*, we get some semantic constraints within Δ^{I2}, particularly $BuyFord^{I2} \subseteq r_{1'2}(BuyCar^{I1'})$ and $r_{12}(BuyVehicle^{I1}) \subseteq Buy^{I2}$ thanks to two of the bridge rules. However, the third bridge rule does not help anyhow to establish relation between $r_{1'2}(BuyCar^{I1'})$ and $r_{12}(BuyVehicle^{I1})$. In Example 2, in each distributed model *DI* of *DTB* the interpretations of the two concepts "on the way"-- $HouseDog^{I2}$ and $Carnivore^{I2}$ --are totally unrelated. By composition of the bridge rules that are available here we derive the inclusions: $MyMastill^{I3} \subseteq r_{2'3}(HouseDog^{I2'}) \subseteq r_{2'3}(r_{12'}(Canis^{I1}))$ and $r_{23}(r_{12}(Canidae^{I1})) \subseteq r_{23}(Carnivore^{I2}) \subseteq DangerousAnimal^{I3}$. However $r_{2'3}(r_{12'}(Canis^{I1}))$ and $r_{23}(r_{12}(Canidae^{I1}))$ are not related in Δ^{I3}, it does not help that $Canis^{I1} \subseteq Canidae^{I1}$ in Δ^{I1}.

From above analysis we can see the main reason why the subsumption does not always propagate to remote ontologies in more complex cases as we would intuitively expect is all the semantic constraints generated by remote bridge rules do not propagate to remote parts of the system. In Package-based Description Logics(P-DL)[5][8], so called compositional consistency semantics is imposed on the importing relation in a distributed ontology environment so that knowledge can be propagated along importing relation chains uninterruptedly. The problem in our work is similar to that in P-DL. Inspired by this, we apply compositional consistency semantics on the D3L framework to try to solve the problem.

Definition 3. Given a distributed interpretation *DI*, we say that domain relation *r*(and also *DI*) satisfies compositional consistency if for each $i，j，k\in I$ and for each $x\in\Delta^{Ii}$ with r_{ij} （x） $=d$ we have r_{jk} （d） $= r_{ik}$ （x） .

We say that D3L is under compositional consistency if only domain relations that satisfy the compositional consistency condition are allowed in distributed interpretations. We denote by D3L$_{（ccy）}$ D3L under compositional consistency. Accordingly, D3L under original semantics is denoted by D3L$_{（orl）}$.

Let us now reconsider what happens in the examples shown in section 2 if the compositional consistency semantics is enforced. In Example 2, by composition of the bridge rules that are available in the example we derive the inclusions: $MyMastill^{I3}\subseteq r_{2'3}$ （$HouseDog^{I2'}$） $\subseteq r_{2'3}$ （$r_{12'}$ （$Canis^{I1}$ ）） and r_{23} （r_{12} （$Canidae^{I1}$ ）） $\subseteq r_{23}$ （$Carnivore^{I2}$ ） $\subseteq DangerousAnimal^{I3}$. Compositional consistency implies $r_{2'3}$ （$r_{12'}$ （$Canis^{I1}$）） $= r_{13}$ （$Canis^{I1}$） and r_{13} （$Canidae^{I1}$） $=$ r_{23} （r_{12} （$Canidae^{I1}$）） . Finally, since $DTB\models_d 1：Canis \sqsubseteq Canidae$ we are now able to derive r_{13} （$Canis^{I1}$） $\subseteq r_{13}$ （$Canidae^{I1}$） and $MyMastill^{I3}\subseteq DangerousAnimal^{I3}$, and hence we can derive that *MyMastill* is a kind of *DangerousAnimal* according to Definition 2. Similar case also occurrs in Example 1, due to space limitations we omit the discussion for it. From above analysis we can see the problem of subsumption propagation between ontologies which are connected indirectly can be solved by our extension. In the following we will give a in-depth theoretical analysis and provement for our extension.

Theorem 2. Given a distributed *Tbox DTB* and a subsumption formula ϕ, if $DTB\models_d \phi$ in D3L$_{（orl）}$, then $DTB\models_d \phi$ also holds in D3L$_{（ccy）}$, i.e. the subsumption propagation property is satisfied in D3L$_{（ccy）}$.

Proof. This follows from the fact that each model that satisfies compositional consistency is also a model in the original D3L semantics. If a formula ϕ is satisfied in each model from the set of models of DTB according to the original semantics, it is also satisfied by each model from its subset--the set of models of DTB that we obtain under compositional consistency. Hence we can derive that if $DTB\models_d \phi$ in D3L$_{（orl）}$, then $DTB\models_d \phi$ also holds in D3L$_{（ccy）}$. □

Subsumption relations that can be propagated in D3L$_{（orl）}$ can also be propagated in D3L$_{（ccy）}$. The only difference is that in the adjusted semantics possibly some more subsumption relations which can not be propagated along bridge rule chains in the former can be propagated correctly in addition, thus Theorem 1 must be extended for D3L$_{（ccy）}$. Theorem 3 gives the general form of subsumption propagation in D3L$_{（ccy）}$ （take action subsumption for example）:

Theorem 3. Given a distributed *Tbox* $DTB=<\{T_0, T_1, ..., T_n\}, BR>$, action $\pi, \rho\in T_0$, $\alpha_i, \beta_i\in T_i$, $1\leq i\leq n$, $1\leq k\leq n$ such that

1. $DTB \models_d i: \alpha_i \sqsubseteq \beta_i, 1 \leq i \leq n,$

2. $i+1: \alpha_{i+1} \xrightarrow{\sqsupseteq} i: \beta_i \in BR, 1 \leq i < k,$

3. $i: \beta_i \xrightarrow{\sqsubseteq} i+1: \alpha_{i+1} \in BR, k \leq i < n,$

4. $1: \alpha_1 \xrightarrow{\sqsupseteq} 0: \pi \in BR$ *and* $n: \beta_n \xrightarrow{\sqsubseteq} 0: \rho \in BR.$

In D3L（ccy） it follows that $DTB \models_d 0: \pi \sqsubseteq \rho$.

Proof. The chain of bridge rules $i+1: \alpha_{i+1} \xrightarrow{\sqsupseteq} i: \beta_i \in BR, 1 \leq i < k$ together with the local subsumptions in $T_1, T_2, T_3, \ldots, T_k$ in the assumptions allows us to relate $r_{10}(\alpha_1^{I1}) \subseteq r_{10}(r_{21}(\ldots r_{kk-1}(\alpha_k^{Ik}) \ldots))$. From compositional consistency we derive $r_{10}(r_{21}(\ldots r_{kk-1}(\alpha_k^{Ik}) \ldots)) = r_{k0}(\alpha_k^{Ik})$. For the other chain of bridge rules alike. Now it is easy to see that $\pi^{I0} \subseteq r_{10}(\alpha_1^{I1}) \subseteq r_{k0}(\alpha_k^{Ik}) \subseteq r_{k0}(\beta_k^{Ik}) \subseteq r_{10}(\beta_1^{I1}) \subseteq \rho^{I0}$, hence we can derive $DTB \models_d 0: \pi \sqsubseteq \rho$. □

Extending Theorem 3 to the case of concept（relation） bridge rule chains, similarly we can draw concept（relation） subsumption propagation theorem, due to space limitations we omit the discussion for it.

5 Distributed Tableaux Algorithm for D3L（ccy）

In this section, we introduce a distributed Tableaux algorithm for deciding satisfiability of concepts with respect to an acyclic distributed *Tbox* for D3L（ccy） over D-ALC. We use the formal definition as follows:

Definition 4. Assume a distributed *Tbox* $DTB=<\{T_i\}_{i \in I}, BR>$ over D-ALC with index set I, concept names $N_C = \cup_{i \in I} N_{ci}$, relation names $N_R = \cup_{i \in I} N_{Ri}$ and action names $N_A = \cup_{i \in I} N_{Ai}$. Let CC_i be the set of all (atomic and complex) concepts over N_{ci}, N_{Ri} and N_{Ai} in negation normal form. A distributed completion tree $T=\{T_i\}_{i \in I}$ is a set of labeled trees $Ti=<V_i, E_i, L_i, r_i>$, such that for each $i \in I$:

1. the members of $\{V_i\}_{i \in I}$ are mutually disjoint;

2. the members of $\{E_i\}_{i \in I}$ are mutually disjoint;

3. the labeling function L_i labels each node $x \in V_i$ with $L(x) \subseteq 2^{CCi}$ and each edge $<x, y> \in E_i$ with $L(<x, y>) \in N_{Ri}$;

4. the labeling function r_i labels each node $x \in V_i$ with a set of references to its r-images $r_i(x) \subseteq \{j: y | j \in I \wedge y \in V_j\}$.

During the run of the tableaux algorithm, tableaux expansion rules are applied on the completion tree and the tree is expanded by each rule application. If no more rules are applicable any more, we say that the tree is complete. There is a clash in the completion tree T if for some $x \in V_i$ and for some $C \in N_{ci}$ we have $\{C, \neg C\} \subseteq L_i(x)$. Given a distributed completion tree $T=\{T_i\}_{i \in I}$, a node $x \in V_i$ is blocked, if it has an ancestor $y \in V_i$ such that $L_i(x) \subseteq L_i(y)$. In such a case we also say that x is blocked by y. Node $y\in V_i$ is said to be an R-successor of $x\in V_i$, if $<x, y>\in E_i$ and L_i（$<x, y>$）$=R$.

Taking a distributed *Tbox DTB* and a concept C in NNF as its inputs, the distributed tableaux algorithm continues in three steps:

（1）Create new distributed tree $T=\{T_j\}_{j\in I}$ such that $T_j=<\{s_0\}, \varnothing, \{s_0\mapsto\{C\}\}, \varnothing>$ for $j=i$ and $T_j=<\varnothing, \varnothing, \varnothing, \varnothing>$ for $j\neq i$.

（2）Apply the following tableaux expansion rules exhaustingly:

（a）⊓-rule: If $C_1\sqcap C_2\in L_i$（x）for some $x\in V_i$ and $\{C_1, C_2\}\not\subseteq L_i$（$x$）, and x is not blocked, then set L_i（x）$= L_i$（x）$\cup\{C_1, C_2\}$.

（b）⊔-rule: If $C_1\sqcup C_2\in L_i$（x）for some $x\in V_i$ and $\{C_1, C_2\}\cap L_i$（x）$=\varnothing$, and x is not blocked, then either set L_i（x）$= L_i$（x）$\cup\{C_1\}$ or set L_i（x）$= L_i$（x）$\cup\{C_2\}$.

（c）∃-rule: If $\exists R.C\in L_i$（x）for some $x\in V_i$ with no *R-successor* y s.t. $C\in L_i$（y）, and x is not blocked, then add new node z to V_i, add the edge $<x, z>$ to E_i, and set L_i（z）$=\{C\}$ and L_i（$<x, z>$）$=\{R\}$.

（d）∀-rule: If $\forall R.C\in L_i$（x）for some $x\in V_i$, and there is *R-successor* y of x s.t. $C\notin L_i$（y）, and x is not blocked, then set L_i（y）$= L_i$（y）$\cup\{C\}$.

（e）Action α-rule: If $[\alpha]C\in L_i$（x）for some $x\in V_i$, $\alpha=$（P_α, E_α）, then set L_i（x）$= L_i$（x）$\backslash\{$ *all conditions in* $P_\alpha\}\cup\{$*all conclusions in* $E_\alpha\}\cup\{C\}$

（f）$\xrightarrow{\sqsupseteq}$-rule: If G（or β）$\in L_j$（x）for some $x\in V_j$, i：C（or α）$\xrightarrow{\sqsupseteq}j$：$G$（or β）$\in BR$, and there is no $y\in V_i$ s.t. C（or α）$\in L_i$（y）and j：$x\in r_i$（y）, and x is not blocked, then add new node y to V_i and set L_i（y）$=\{C$（or α）$\}$, and set r_i（y）$=\{j$：$x\}\cup r_j$（x）.

（g）$\xrightarrow{\sqsubseteq}$-rule: If D（or π）$\in L_i$（x）for some $x\in V_i$, i：D（or π）$\xrightarrow{\sqsubseteq}j$：$H$（or ρ）$\in BR$ and there is $y\in V_j$ s.t. j：$y\in r_i$（x）and H（or ρ）$\notin L_j$（y）, then set L_j（y）$= L_j$（y）$\cup\{H$（or ρ）$\}$.

（3）If none of the tableaux expansion rules is applicable any more (i.e., the distributed tree is now complete), answer "C is satisfiable" if a clash-free completion tree has been constructed. Answer "C is unsatisfiable" otherwise.

Theorem 4. The satisfiability problem of concept in D3L$_{(ccy)}$ is decidable.

Proof. To determine whether the concept is satisfiable is to check whether the completion tree includes clash. In D3L$_{(ccy)}$, a distributed tree can be expanded by applying tableaux rules, as described in above algorithm. Now we ought to prove that the algorithm, once starts with input, eventually terminates. We will consider each step of the algorithm separately:

It is obvious that the establishment of a new tree in the first step eventually terminates for both $j=i$ and $j\neq i$.

In the second step, tree is expanded by applying seven tableaux rules. Only ⊔-rule is uncertain, but at worst complete binary tree is constructed which can be finished in exponential time. ⊓-rule, ∃-rule and ∀-rule which can be finished in polynomial time are all terminable. Action α-rule has no effect on the decidability of D3L$_{(ccy)}$ too, for detailed discussion please refer to [1]. The terminability of $\xrightarrow{\sqsupseteq}$-rule and $\xrightarrow{\sqsubseteq}$-rule is discussed as follows: If there are onto-bridge rules ingoing into T_i then possibly in some $x \in V_i$ such that C (*or* α) $\in L_i$ (x) and C (*or* α) appears on a right hand side of an onto-bridge rule, say k: D (*or* π) $\xrightarrow{\sqsupseteq}$ i: C (*or* α), then the $\xrightarrow{\sqsupseteq}$-rule is applied and computation is triggered in T_k. By structural subsumption we assume that this computation eventually terminates (the length of the longest incoming path of into-bridge rules decreased for T_k compared to T_i, and all such paths are finite because of acyclicity). During this process we possibly get some new concepts in L_i (x) that are introduced thanks to incoming into-bridge rules that trigger the $\xrightarrow{\sqsubseteq}$-rule--but only finitely many. The computation now continues in x and its descendants and possibly the $\xrightarrow{\sqsupseteq}$-rule is triggered again in some $y \in V_i$, a descendant of x. But thanks to subset blocking, this happens only finitely many times. Hence $\xrightarrow{\sqsupseteq}$-rule and $\xrightarrow{\sqsubseteq}$-rule eventually terminate.

Clash checking process of distributed completion tree in the third step is terminable. Either concept is unsatisfiable because there are clashes in the tree, or concept is satisfiable because there are not clashes. Hence the satisfiability problem of concept in D3L$_{(ccy)}$ is decidable. □

6 Conclusion and Future Work

Although knowledge propagation theory has been studied, only simple cases with two DDL ontologies were involved. In this paper we focused on the general cases with arbitrary number of ontologies and bridge rules chains. We found original semantics of D3L could not ensure correct knowledge propagation because domain relations were not transitive. Inspired by Package-based description Logics, we extended the original semantics of D3L by imposing so called compositional consistency condition on domain relations. Under this semantics knowledge propagates along chains of bridge rules correctly. At last, we studied the decidability of satisfiability problem of concepts in D3L under compositional consistency and provided a distributed Tableaux algorithm for deciding it.

Extending presented algorithm for expressive D3L, analysing computational complexity of the algorithm and developing corresponding reasoners are left for future work.

Acknowledgment

This work is supported by the Chinese National Natural Science Foundation(No. 61035003, No. 61072085, No. 60970088).

References

1. Shi Zhongzhi, Dong Mingkai, Jiang Yuncheng, et al.: A Logic Foundation for the Semantic Web. Science in China, Series F Information Sciences, 48(2):161-178 (2005)
2. Shi Zhongzhi, Jiang Yuncheng, Zhang Haijun, et al.: Agent service matchmaking based on description logic. Chinese Journal of Computers, 27(5): 625-635 (2004) (in Chinese)
3. Jiang Yuncheng, Shi Zhongzhi, Tang Yong, et al.: A Distributed Dynamic Description Logic. Journal of Computer Research and Development, 43(9): 1603-1608 (2006) (in Chinese)
4. Wang Zhuxiao.: Distributed Information Retrieval Oriented Automatic Reasoning. Doctor's thesis, Institute of Computing Technology, Chinese Academy of Sciences, Beijing (2010)
5. Bao, J., Caragea, D., Honavar, V.: On the semantics of linking and importing in modular ontologies. LNCS, vol. 4273, pp. 72-86. Springer, Heidelberg (2006)
6. Franz, B., Diego, C., Deborah, M., et al.: The description logic handbook: Theory, implementation, and applications. 2nd edition. Cambridge University Press, Cambridge (2007)
7. Aquin, M. D., Schlicht, A., Stuckenschmidt, H., et al.: Criteria and Evaluation for Ontology Modularization Techniques. LNCS, vol. 5445, pp.67-89. Springer, Heidelberg (2009)
8. Bao, J., Caragea, D., Honavar, V. G.: A distributed tableau algorithm for package-based description logics. In: 2nd International Workshop on Context Representation and Reasoning. IOS Press, Italy (2008)
9. Grau B. C., Parsia B., Sirin E.: Combining OWL ontologies using Ɛ-connections. Journal of Web Semantics, 4(1): 40-59 (2006)
10. Grau B. C., Parsia B., Sirin E.: Ontology Integration Using Ɛ –Connections. LNCS, vol. 5445, pp. 293-320. Springer, Heidelberg (2009)
11. Kutz O., Lutz C., Wolter F., et al.: Ɛ -connections of abstract description systems. Articial Intelligence, 156(1): 1-73 (2004)
12. Bao J., Voutsadakis G., Slutzki G., et al.: Package-Based Description Logics. LNCS, vol. 5445, pp. 349-371. Springer, Heidelberg (2009)
13. Lutz C., Walther D., Wolter F.: Conservative extensions in expressive description logics. In: Twentieth International Joint Conference on Artificial Intelligence(IJCAI'07), pp. 453-458. AAAI Press, California (2007)
14. Serafini L., Borgida A., Tamilin A.: Aspects of Distributed and Modular Ontology Reasoning. In: Tenth International Joint Conference on Artificial Intelligence(IJCAI'05), pp. 570-575. AAAI Press, California (2005)