

Semantic Cooperation and Knowledge Reuse by Using Autonomous Ontologies

Yuting Zhao¹, Kewen Wang¹, Rodney Topor¹, Jeff Z. Pan², and Fausto Giunchiglia³

¹ Griffith University, Australia

² University of Aberdeen, UK

³ University of Trento, Italy

{yuting.zhao, k.wang, r.topor}@griffith.edu.au
jpan@csd.abdn.ac.uk, fausto@dit.unitn.it

Abstract. Several proposals have been put forward to support distributed agent cooperation in the Semantic Web, by allowing concepts and roles in one ontology be reused in another ontology. In general, these proposals reduce the autonomy of each ontology by defining the semantics of the ontology to depend on the semantics of the other ontologies.

We propose a new framework for managing autonomy in a set of cooperating ontologies (or ontology space). In this framework, each language entity (concept/role/individual) in an ontology may have its meaning assigned either locally with respect to the semantics of its own ontology, to preserve the autonomy of the ontology, or globally with respect to the semantics of any neighbouring ontology in which it is defined, thus enabling semantic cooperation between multiple ontologies.

In this way, each ontology has a “subjective semantics” based on local interpretation and a “foreign semantics” based on semantic binding to neighbouring ontologies. We study the properties of these two semantics and describe the conditions under which entailment and satisfiability are preserved. We also introduce two reasoning mechanisms under this framework: “cautious reasoning” and “brave reasoning”. Cautious reasoning is done with respect to a local ontology and its neighbours (those ontologies in which its entities are defined); brave reasoning is done with respect to the transitive closure of this relationship. This framework is independent of ontology languages. As a case study, for Description Logic \mathcal{ALCN} we present two tableau-based algorithms for performing each form of reasonings and prove their correctness.

1 Introduction

The Semantic Web vision is to develop a distributed environment in which software agents can automatically, conveniently and effectively interpret and apply the data that is available on the Web. To this end, a system of knowledge representation which supports semantic cooperation between distributed agents is required. Such a system must be based on ontologies which define the terms and relationships used in a particular application domain. Each such ontology reflects the objective and shared views of a community of users working in that domain. However, the original use of the word

“ontology” in philosophy was to describe a complete, self-contained domain of discourse. This usage does not scale to the open and distributed Web, where there are ontologies for each different application domain and even different ontologies for the same domain. Thus, to support semantic cooperation between agents, it is necessary to manage and reason about multiple ontologies, which we call an ontology space. How to do this effectively is a major research problem for the Semantic Web.

AI researchers have also studied management and reasoning in multiple representations of application domains using contextual reasoning [6,7,5]. For this reason, it is a natural and interesting issue to combine ontology-based and context-based approaches so that the advantages of both ontology and contextual reasoning can be employed in the same system. This idea led a series of interesting work.

C-OWL [4] sets up relations outside the ontologies by a set of “bridge rules” between the concepts (individuals) from different ontologies. Its semantics relies on the *domain relation* in DDL [3], which is a directional mapping from the elements of one domain to the other domain. \mathcal{E} -Connection [10] puts the relations inside the ontology, by extending OWL with a new kind of “link property”. It connects two sets of strictly disjoint concepts from different domains. P-DL [2] treats every foreign term as an imported relation, and semantically interprets it by an *image domain relation* which is a one-to-one and compositional consistent mapping between two domains. Semantic Importing [12] focuses on the overlaps of domains; it allows a subconcept which falls into the conjunction of two domains to be semantically imported and used in the other ontology. Conservative Extension [9] restricts multiple ontology modules in the same global interpretation domain and allows them be interpreted using standard semantics.

It is easy to see that the above approaches somehow weaken the autonomy of an ontology. In order to bridge the gaps between the semantics of different ontologies, a class of approaches based on cross-domain relations (i.e., domain relations in C-OWL, directional binary relation in \mathcal{E} -connection, and image domain relation in P-DL) needs the information of the domain element in the other ontology to interpret a bridge rule in C-OWL, a link property in \mathcal{E} -connection, or an importing relation in P-DL. For example, suppose that there is well-accepted ontology on the Web called “Vehicle”, in which a concept “Car” is defined as: “A car consists of two parts: engine and body”. In C-OWL, this concept is represented as $Vehicle : Car$ which indicates that the concept “Car” is defined in the ontology “Vehicle”. Both BMW and Toyota may wish to borrow the concept “Car” when they design their own ontologies. These two companies have different interpretations on “engine” and “body”. In C-OWL, $Vehicle : Car$ has different local interpretations in BMW and Toyota. However, a user cannot distinguish this difference and confusion may be caused when other people use $Vehicle : Car$. Semantic Importing does not rely on domain relations, but one needs to know exactly the domain elements and how the concepts/roles are interpreted in the other ontology. Conservative extensions require a single global domain, and prevent each module from local interpretation its axioms under its own context.

In this paper, we propose a new framework for managing autonomy in a distributed ontology space. On the one hand a language entity (concept/role/individual) is interpreted totally under local domain semantics in order to preserve the autonomy of an ontology; on the other hand a (shared) language entity is restricted by a *semantic*

binding if necessary in order to enable semantic cooperation among several ontologies. In this way, one ontology is able to express its “subjective” opinion by local interpretation, and to receive its *foreign semantics* by semantic binding. We use the term “foreign semantics” of one ontology to express the “semantic meaning” of an foreign entity from another ontology connected by semantic binding. Accordingly, we also introduce two reasoning mechanisms: *cautious reasoning* and *brave reasoning*. The former relies on an ontology and its foreign semantics from its neighbors’ ontologies connected by semantic binding, but it does not trust the foreign semantics of its neighbour from their neighbors. The latter believes an ontology and its foreign semantics, and also its neighbour ontologies and their foreign semantics.

The main contribution of this paper are the followings:

- We introduce a novel approach to define semantic cooperation between different ontologies. By semantic binding, the semantics of language entities from one ontology is able to be accessed in other ontologies. This is different from the existing domain-relation based approaches.
- We formalize two forms of reasoning mechanisms: cautious reasoning and brave reasoning. The former only does reasoning in one ontology and its neighbors’ ontologies associated by its semantic binding; the latter does reasoning in one ontology and its neighbours’ ontologies and their neighbours’ ontologies and so on.

In the rest, we briefly review ontologies and ontology spaces in Section-2, and introduce the notion of an autonomous ontology in Section-3. In Section-4 we introduce the two reasoning mechanisms and the algorithms. The related work review and conclusion are given in Section-5.

2 Preliminaries: Ontology Space and Foreign Entity

In general understanding [14], an ontology is a set of annotated *terminological axioms* and *facts*. Current discussion is based on normal Description Logic (DL) [1]. The proposed framework can be restricted or generalized to some DL languages such as OWL, SHOIN(D+), etc.

2.1 Ontology and Ontology Space

Let \mathbb{C} be a finite set of concept names, \mathbb{R} a finite set of role names, and \mathbb{E} a finite sets of individual names. A *language* \mathbb{L} has a vocabulary of the disjoint union of \mathbb{C} , \mathbb{R} and \mathbb{E} .

Definition 1 (Ontology). Let \mathbb{L} be the language. An Ontology \mathcal{O} is a tuple $\langle T, A \rangle$, where T and A are TBox and ABox respectively in Description Logic on \mathbb{L} .

Definition 2 (Ontology interpretation). An (abstract) ontology interpretation \mathcal{I} is a pair $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, in where $\Delta^{\mathcal{I}}$ is a nonempty domain, and $\cdot^{\mathcal{I}}$ is a mapping that assigns

1. to each concept name $c \in \mathbb{C}$ a subset of $\Delta^{\mathcal{I}}$,
2. to each role name $R \in \mathbb{R}$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$,
3. to each individual name $e \in \mathbb{E}$ an element of $\Delta^{\mathcal{I}}$.

Definition 3 (Ontology Space). Let I be a set of indexes, standing for a set of URIs for ontologies. Let $\mathbb{L}_I = \{\mathbb{L}_i\}_{i \in I}$ be a set of languages. An Ontology Space \mathbb{O}_I on \mathbb{L}_I is a family $\{O_i\}_{i \in I}$, s.t. every O_i is an ontology on language \mathbb{L}_i , where $i \in I$.

In ontology space $\mathbb{O}_I = \{O_i\}_{i \in I}$, we denote, by \mathbb{C}_i the set of concept names in ontology O_i . Analogous to \mathbb{R}_i and \mathbb{E}_i . Actually language \mathbb{L}_i has a vocabulary of the disjoint union of \mathbb{C}_i , \mathbb{R}_i and \mathbb{E}_i . In the rest of the paper, we use *language entity* to denote concept, role, or individual in one ontology.

2.2 Foreign Entity

In ontology space, sometimes a language entity is defined in one ontology, but could be used in another ontology. So we partition the language \mathbb{L}_i in two parts: the *local entity* and the *foreign entity* (originated from local language and foreign language in [4]). Intuitively, local entities are the roles, concepts, and individuals that one invites in her own ontology; foreign entities are the roles, concepts, and individuals that she borrows from the other ontologies in order to define something in her ontology.

In this paper, when we are talking about semantics and reasoning, we always tell a language entity in the ontology space by a way showing (1) where it is used, and (2) where it is originally defined. Suppose that $C \in \mathbb{L}_i$ and $i, j \in I$, then formally in ontology space we have a language entity like, $(i : j : C)$, which means a language entity C appears in ontology O_i , but is originally defined in ontology O_j . This kind of denotation is applied to concepts, roles, and individuals in ontology space.

One of the advantages of this denotation is, in syntax two concepts/roles with the same name but used in different ontologies are distinguishable. For example, suppose we use $(BMW : auto : engine)$ for the engines in BMW car ontology, and $(Toyota : auto : engine)$ for those in Toyota car ontology; obvious these two concepts should be different, and it is easy to see from the syntax: $(BMW : auto : engine) \neq (Toyota : auto : engine)$. This denotation is important in this paper because, the approach proposed in this paper assigns local semantics to this kind of concepts/roles, and treats them as totally different entities.

3 Autonomous Ontology

In one ontology space, each ontology reflects the subjective opinion on a partial structure of the universe. In Semantic Web, in general one party presents her personal knowledge (understanding) by her ontology. Thereafter we argue each ontology should be semantical independent and keeping autonomy.

Formally an *autonomous ontology* is composed by two parts: one is an ontology which is to be interpreted locally in order to keep the autonomy of one party; the other is a set of foreign entities, which is called *semantic binding* in this paper, in order to accept foreign information or knowledge from the other parties.

Definition 4 (Autonomous Ontology). Let $\mathbb{O}_I = \{O_i\}_{i \in I}$ be an ontologiespace, where I is a set of indexes. An autonomous ontology is a tuple $AO_i = \langle \mathbb{B}_i, O_i \rangle$, in which $O_i \in \mathbb{O}_I$, and \mathbb{B}_i is the semantic binding of AO_i , which is a set of foreign entities.

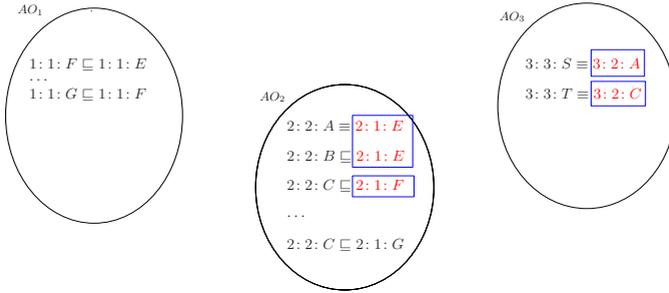


Fig. 1. Autonomous ontology space

An *autonomous ontology space* $\mathbb{A}\mathbb{O}_I$ is a set of autonomous ontologies. In an autonomous ontology space $\mathbb{A}\mathbb{O}_I = \{AO_i\}_{i \in I}$, if an entity $(j:j:x)$, which is original defined in AO_j , is semantic bounded in AO_i ; i.e., $(i:j:x) \in \mathbb{B}_i$, then we say AO_j is the *binding neighbour* of AO_i .

In fact it is easy to treat an autonomous ontology as a normal ontology; for example in OWL we could just introduce a binding annotation like:
 Annotation (binding <http://www.auto.org/engine#>)
 to express the semantic binding. So in the rest of the paper sometime we also mean an autonomous ontology by ontology.

Example 1 (Autonomous Ontology). Consider the autonomous ontology space in Figure-1, suppose we have following semantic bindings: $\mathbb{B}_1 = \emptyset$, $\mathbb{B}_2 = \{(2:1:E), (2:1:F)\}$, and $\mathbb{B}_3 = \{(3:2:A), (3:2:C)\}$. So AO_1 is the binding neighbour of AO_2 , and AO_2 is the binding neighbour of AO_3 . We note that in autonomous ontology it allows some foreign entity outside of the semantic binding, e.g., $(2:1:G)$ in AO_2 . \square

3.1 Local Interpretation

Definition 5 (Local Interpretation). For autonomous ontology $AO_i = \langle \mathbb{B}_i, O_i \rangle$, a local interpretation \mathcal{I}_i is a pair $\langle \Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i} \rangle$, in where $\Delta^{\mathcal{I}_i}$ is a nonempty domain, and $\cdot^{\mathcal{I}_i}$ is a mapping, s.t. $\cdot^{\mathcal{I}_i}$ assigns

1. (for local entities)
 - (a) a subset of $\Delta^{\mathcal{I}_i}$ to each local concept name $(i:i:c)$;
 - (b) a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}$ to each local role name $(i:i:r)$;
 - (c) an element of $\Delta^{\mathcal{I}_i}$ to each local individual name $(i:i:e)$,
2. and for $i \neq j$ (for foreign entities)
 - (a) a subset of $\Delta^{\mathcal{I}_i}$ to each foreign concept name $(i:j:c)$;
 - (b) a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}$ to each foreign role name $(i:j:r)$;
 - (c) an element of $\Delta^{\mathcal{I}_i}$ to each foreign individual name $(i:j:e)$.

As we see in the above definition, in local interpretation not only local entities but also foreign entities including those in the semantic binding are interpreted under local domain. Actually, the local interpretation of an autonomous ontology is the interpretation of the ontology. From this aspect the abstract interpretation of ontology in Section-2 is

also a local interpretation. The difference is, the definition in the last section is for a single ontology, but here we focus on one ontology among an ontology space.

Following common understanding, if an axiom $a \in A$ is true under an interpretation \mathcal{I} , we say that interpretation \mathcal{I} *satisfies* the axiom a , and denote this by $\mathcal{I} \models a$.

Definition 6 (Satisfiability of autonomous ontology). Let AO_i be an autonomous ontology and an \mathcal{I} its local interpretation, we say that \mathcal{I} satisfies AO_i , if for any axiom $a \in AO_i$, we have $\mathcal{I} \models a$. We call \mathcal{I} a local model of AO_i , and denote this fact by $\mathcal{I} \models_{\mathcal{L}} AO_i$.

Definition 7 (Local Entailment). Let AO_i be an autonomous ontology, λ a concept description or an assertion. We say that λ is a local entailment of AO_i , iff for any local model \mathcal{I}_i of AO_i , $\mathcal{I}_i \models \lambda$. This fact is denoted by $AO_i \models_{\mathcal{L}} \lambda$.

Example 2 (Local semantics). Consider the autonomous ontology space in Figure-1. We have $AO_2 \models_{\mathcal{L}} (2:2:B) \sqsubseteq (2:2:A)$, because for any local model \mathcal{I} ,

$$\begin{array}{c} \mathcal{I} \models (2:2:B) \sqsubseteq (2:1:E), \quad \text{and} \\ \mathcal{I} \models (2:2:A) \equiv (2:1:E) \\ \hline \mathcal{I} \models (2:2:B) \sqsubseteq (2:2:A) \end{array}.$$

3.2 C-Binding Consistency

For autonomous ontology $AO_i \in \mathbb{A}_{O_I}$, let $\mathbb{B}_i = \{\bigcup_{j \in I} \mathbb{B}_{ij}\}$ be the semantic binding¹, in which $\mathbb{B}_{ij} = \{(i:j:C)\}_{i,j \in I}$ contains all of the semantic-bounded foreign entities which are original defined in O_j .

In autonomous ontology $AO_i \in \mathbb{A}_{O_I}$, a j -concept (j -role) is an class (property) description which is composed by the entities in \mathbb{B}_{ij} . For example, a j -concept of AO_i could be $\exists(i:j:hasChild).(i:j:Male)$.

Let λ_j be a j -concept, obviously in λ_j all of the entities are prefixed by “ $i:j:$ ”. If we change the prefix of every entity in λ_j from “ $i:j:$ ” to “ $j:j:$ ”, and then we get λ'_j . We call λ'_j the *original image* of λ_j in O_j . For example, let λ_j be $(i:j:Person) \sqcap \forall(i:j:hasChild).(i:j:Female)$, then its original image λ'_j is $(j:j:Person) \sqcap \forall(j:j:hasChild).(j:j:Female)$.

We note that the original image λ'_j of a j -concept λ_j may not has a concept name in O_j ; it may not be explicitly defined there.

Definition 8 (C(j)-binding Consistent Model). Let $AO_i = \langle \mathbb{B}_i, O_i \rangle$ be an autonomous ontology, and $j \in I$. Let \mathcal{I} be a local model of AO_i , if for any j -concept λ_j , we have

$$\mathcal{I} \models \lambda_j \quad \text{iff} \quad AO_j \models_{\mathcal{L}} \lambda'_j \quad (1)$$

then we say \mathcal{I} is the C(j)-binding consistent model of AO_i . This fact is denoted by $\mathcal{I} \models_{c(j)} AO_i$

Example 3 (C(j)-binding consistent model). Let $\Delta_2 = \{\alpha, \beta\}$ be the domain of AO_2 in Figure-1. Considering following two local interpretations of AO_2 :

¹ It is possible that some foreign entities are not included in \mathbb{B}_i . Actually these foreign entities are not semantic-bounded; they are “free-access” entities according to [17].

Table 1. Example of interpretations for the autonomous ontology AO_2

symbol	\mathcal{I}_{21}	\mathcal{I}_{22}
\top	$\{\alpha, \beta\}$	$\{\alpha, \beta\}$
(2:2:A)	$\{\alpha\}$	$\{\alpha, \beta\}$
(2:2:B)	$\{\alpha\}$	$\{\alpha\}$
(2:2:C)	$\{\alpha, \beta\}$	$\{\beta\}$
(2:1:E)	$\{\alpha\}$	$\{\alpha, \beta\}$
(2:1:F)	$\{\alpha, \beta\}$	$\{\beta\}$
(2:1:G)	$\{\alpha, \beta\}$	$\{\alpha, \beta\}$
...
(2:1:F) \sqsubseteq (2:1:E)	<i>Not satisfiable</i>	<i>Satisfiable</i>

Obviously both \mathcal{I}_{21} and \mathcal{I}_{22} are local models of AO_2 , i.e., $\mathcal{I}_{21} \models_{\mathcal{L}} AO_2$ and $\mathcal{I}_{22} \models_{\mathcal{L}} AO_2$. Let $\lambda = (2:1:F) \sqsubseteq (2:1:E)$, from Table-1 we find $\mathcal{I}_{21} \not\models \lambda$. Since $\mathbb{B}_{21} = \{(2:1:E), (2:1:F)\}$, λ is the I -concept, and $AO_1 \models_{\mathcal{L}} \lambda$, we have \mathcal{I}_{21} is not a $C(I)$ -binding consistent model. Actually \mathcal{I}_{22} is a $C(I)$ -binding consistent model of AO_2 . In this example we also note that foreign entity outside of the semantic binding does not carry any semantic information from its original ontology; e.g., since $(2:1:G) \notin \mathbb{B}_{21}$, although $AO_1 \models_{\mathcal{L}} (1:1:G) \sqsubseteq (1:1:F)$, in AO_2 $C(I)$ -binding consistent model \mathcal{I}_{22} does not need to satisfy $(2:1:G) \sqsubseteq (2:1:F)$. \square

Definition 9 (C(j)-binding Entailment). Let AO_i be an autonomous ontology, λ a concept description² or an assertion. We say that λ is the $C(j)$ -binding entailment of AO_i , iff for any $C(j)$ -binding consistent model \mathcal{I} of AO_i , $\mathcal{I} \models \lambda$. This fact is denoted by $AO_i \models_{C(j)} \lambda$.

In Example-3 we have $AO_2 \models_{C(I)} (2:1:F) \sqsubseteq (2:1:E)$, and $AO_2 \models_{C(I)} (2:2:C) \sqsubseteq (2:2:A)$.

Definition 10 (C-binding Consistency). Let \mathcal{I} be a local model of AO_i . If for any $j \neq i \in I$, \mathcal{I} is the $C(j)$ -binding consistent model of AO_i , then \mathcal{I} is the C -binding consistent model of AO_i . We say AO_i is C -binding consistent in $\mathbb{A}\mathbb{O}_I$ if there exists a C -binding consistent model.

We say an autonomous ontology $AO_i = \langle \mathbb{B}_i, O_i \rangle$ is C -satisfiable, if it is C -binding consistent and O_i is satisfiable.

Let AO_i be an autonomous ontology and $i \neq j \in I$, we note that not every local model is a $C(j)$ -binding consistent model; it is not necessary for a local model to satisfy a j -concept, but it is for a $C(j)$ -binding consistent model. We also note that not every $C(j)$ -binding consistent model is a C -binding consistent model; a $C(j)$ -binding consistent model may not satisfies a k -concept for $k \neq j$. So an autonomous ontology which is satisfiable under the local semantics could be unsatisfiable under the autonomous semantic.

² Here for convenient we treat a subsumption $x \sqsubseteq y$ as a concept description $\neg x \sqcup y$. Same for the rest of the paper.

Definition 11 (C-entailment). Let AO_i be an autonomous ontology, λ a concept description or an assertion. We say that λ is the C-entailment of AO_i , iff for any C-binding consistent model \mathcal{I}_i of AO_i , $\mathcal{I}_i \models \lambda$. This fact is denoted by $AO_i \models_C \lambda$.

Lemma 1. Let AO_i be an autonomous ontology, λ a concept description or an assertion. We have $AO_i \models_{\mathcal{L}} \lambda \implies AO_i \models_C \lambda$.

In this paper, we use $\Pi_C(AO_i) = \{\lambda \mid AO_i \models_C \lambda\}$ to denote the set of C-entailments of AO_i , and call it *Cautious theory* of AO_i . Comparably, we also use $\Pi_{\mathcal{L}}(AO_i) = \{\lambda \mid AO_i \models_{\mathcal{L}} \lambda\}$ to denote the set of local entailment of AO_i under local semantics, and call it *local theory* of AO_i .

Theorem 1. For a autonomous ontology AO_i , we have $\Pi_{\mathcal{L}}(AO_i) \subseteq \Pi_C(AO_i)$.

Proposition 1. Let $AO_i = \langle \mathbb{B}_i, O_i \rangle$ be C-binding consistent, for any $j \neq i \in I$ and any j-concept λ of AO_i , we have $AO_i \models_C \lambda$ if $AO_j \models_{\mathcal{L}} \lambda'$.

Example 4. In the autonomous ontology space in Figure-1, we have $AO_2 \models_C (2:2:C) \sqsubseteq (2:2:A)$. However for AO_3 , since $AO_2 \not\models_{\mathcal{L}} (2:2:C) \sqsubseteq (2:2:A)$, $AO_3 \not\models_C (3:2:C) \sqsubseteq (3:2:A)$, and then we do not have AO_3 entails $(3:3:T) \sqsubseteq (3:3:S)$ under C-entailment. \square

In this paper when we say *cautious semantics* of an autonomous ontology we mean the C-binding model(s). We say an autonomous ontology space $\mathbb{A}\mathbb{O}_I$ is *C-binding consistent* if every autonomous ontology is C-binding consistent.

Definition 12 (C-entailment of Ontology Space). Suppose autonomous ontology space $\mathbb{A}\mathbb{O}_I = \{AO_i\}_{i \in I}$ is C-binding consistent. Let $AO_i \in \mathbb{A}\mathbb{O}_I$, and λ a concept description or an assertion. We say that λ is the C-entailment of autonomous ontology space $\mathbb{A}\mathbb{O}_I$, denoted by $\mathbb{A}\mathbb{O}_I \models_C \lambda$, iff there exists $i \in I$, s.t. $AO_i \models_C \lambda$. We also say $i \in I$ is the provenance of the entailment λ .

3.3 B-Binding Consistency

B-binding stands for “brave binding”. One autonomous ontology not only relies on its binding neighbors, it also trusts the neighbors of its binding neighbors. In this way B-binding could build more stronger semantic cooperation among multiple ontologies than C-binding, in the sense that some information in one ontology is transitively reused by, not only its neighbour but also the neighbour’s neighbour.

For example in the autonomous ontology space in Figure-1, AO_3 does not entail $(3:3:T) \sqsubseteq (3:3:S)$ under C-binding semantics because $AO_2 \not\models_{\mathcal{L}} (2:2:C) \sqsubseteq (2:2:A)$; but it could entail this subsumption under B-binding semantics because $AO_2 \models_C (2:2:C) \sqsubseteq (2:2:A)$. Details will be given later.

Definition 13 (B-binding Entailment). Let $AO_i = \langle \mathbb{B}_i, O_i \rangle$ be an autonomous ontology, and \mathcal{I} a local interpretation of it. We say \mathcal{I} is a B-binding consistent model of AO_i , which is denoted by $\mathcal{I} \models_B AO_i$, if

- 1 \mathcal{I} is a local model of AO_i , and
- 2 for any $j \in I$ and any j -concept λ_j we have $\mathcal{I} \models \lambda_j$ iff
 - (a) $AO_j \models_{\mathcal{L}} \lambda'_j$, or
 - (b) $AO_j \models_{\mathcal{C}} \lambda'_j$, or
 - (c) $AO_j \models_{\mathcal{B}} \lambda'_j$

We say AO_i is **B-binding consistent** in $\mathbb{A}\mathbb{O}_I$ if there exists a **B-binding consistent model**. Let ψ be a formula. For any **B-binding model** \mathcal{I} of AO_i , if $\mathcal{I} \models \psi$, then we say ψ is the **B-entailment** of AO_i . This fact is denoted by $AO_i \models_{\mathcal{B}} \psi$

Above is a recursive definition on the **B-binding semantics** for an autonomous ontology. Local knowledge in one ontology is used to support a logical result in its *binding neighbour-reached* ontology. We say AO_j is **binding neighbour-reached** from AO_i , if there exists a sequence $(AO_{x,1}, AO_{x,2}, \dots, AO_{x,k})$ such that for $1 \leq y < k$ $AO_{x,(y+1)}$ is a **binding neighbour** of $AO_{x,y}$ and $AO_{x,1} = AO_i$ and $AO_{x,k} = AO_j$.

From above definition it is easy to get the following lemma.

Lemma 2. For a autonomous ontology $AO_i \in \mathbb{A}\mathbb{O}_I$, every **B-binding model** is a **C-binding model**.

Lemma 3. Let AO_i be an autonomous ontology, λ a concept description or an assertion. We have $AO_i \models_{\mathcal{C}} \lambda \implies AO_i \models_{\mathcal{B}} \lambda$.

In this paper, we use $\Pi_{\mathcal{B}}(AO_i) = \{\lambda \mid AO_i \models_{\mathcal{B}} \lambda\}$ to denote the set of **B-entailments** of AO_i , and call it *brave theory* of AO_i . From Lemma-2 and Lemma-3, the following theorem is obvious.

Theorem 2. For a autonomous ontology AO_i , we have $\Pi_{\mathcal{C}}(AO_i) \subseteq \Pi_{\mathcal{B}}(AO_i)$.

According to Definition-13 the following proposition is obvious.

Proposition 2. Let $AO_i = \langle \mathbb{B}_i, O_i \rangle$ be **B-binding consistent**, for any $j \neq i \in I$ and any j -concept λ of AO_i , we have $AO_i \models_{\mathcal{B}} \lambda$ if $AO_j \models_{\mathcal{C}} \lambda'$.

Example 5. In the autonomous ontology space in Figure-1, we have $AO_3 \models_{\mathcal{B}} (3:3:T) \sqsubseteq (3:3:S)$ under the brave semantics.

In this paper when we say *brave semantics* of an autonomous ontology we mean the **B-binding model(s)**. We say an autonomous ontology space $\mathbb{A}\mathbb{O}_I$ is **B-binding consistent** if every autonomous ontology is **B-binding consistent**.

Definition 14 (B-entailment of Ontology Space). Suppose autonomous ontology space $\mathbb{A}\mathbb{O}_I = \{AO_i\}_{i \in I}$ is **B-binding consistent**. Let $AO_i \in \mathbb{A}\mathbb{O}_I$, and λ a concept description or an assertion. We say that λ is the **B-entailment** of autonomous ontology space $\mathbb{A}\mathbb{O}_I$, denoted by $\mathbb{A}\mathbb{O}_I \models_{\mathcal{B}} \lambda$, iff there exists $i \in I$, s.t. $AO_i \models_{\mathcal{B}} \lambda$. We also say $i \in I$ is the *provenance of the entailment* λ .

4 Tableaux Algorithms of Reasoning on \mathcal{ALCN}

In this section, we present two distributed tableaux algorithms to realize *cautious reasoning* (under cautious semantics) and *brave reasoning* (under brave semantics) in autonomous ontology space respectively.

Here we consider ontologies represented as \mathcal{ALCN} TBoxes (which consists of only class axioms). These algorithms are designed for verifying class satisfiability in an autonomous ontology space $\mathbb{A}\mathbb{O}_I$, and can also be used to verify class subsumption³.

4.1 Preliminary of Tableaux Algorithm

Tableaux algorithms are very useful to solve class satisfiability problem. They test the satisfiability of a class λ ⁴ by trying to construct an interpretation for λ , which is represented by a *completion tree* \mathcal{T} which is formally defined as following: A *completion tree* is a tuple $\mathcal{T} = \langle x_0, N, E, \mathcal{L} \rangle$, where x_0 is the root of \mathcal{T} , N and E are the sets of nodes and edges, respectively, of \mathcal{T} , and \mathcal{L} is a function that maps a node x in \mathcal{T} to its label $\mathcal{L}(x)$, and an edge $\langle x, y \rangle$ in \mathcal{T} to its $\mathcal{L}(\langle x, y \rangle)$, respectively.

A tableaux algorithm starts from an labeled initial tree (usually simply a root node), and is expanded by repeatedly applying the completion rules. The algorithm terminates either when \mathcal{T} is *complete* (no further completion rules can be applied) or when an obvious contradiction, or *clash*, has been revealed.

Intuitively, our tableaux algorithm expands a completion tree *w.r.t.* the local axiom box, and then project some part of the tree (which is related to other autonomous ontology) for further expansion *w.r.t.* the axiom boxes of the neighbour autonomous ontology, (e.g., sending the original image of a j -concept to AO_j and start a new tableaux algorithm to check the satisfiability) and then back-project some semantics results to the local completion tree. We say that a completion tree \mathcal{T} is $S(j)$ -bound if there exist some j -concept or j -role descriptions in the labels of all nodes and edges of \mathcal{T} . In this section, we use procedure $\text{Tab}(AO_i, \mathcal{T})$ as a well known (local) \mathcal{ALCN} tableaux algorithm to expand \mathcal{T} *w.r.t.* a local ontology AO_i . $\text{Tab}(AO_i, \mathcal{T})$ has two distinguished features that we need: (1) it takes not only a single node but an arbitrary initial completion tree, (2) the algorithm can cache reasoning states, *i.e.*, backtracking points. In the algorithms below we also use $\text{backtrack}(\text{Tab}(AO_i, \mathcal{T}))$ to denote the operation to expand tree \mathcal{T} from the backtracking point and return a completion tree.

Projection is to bring information from one autonomous ontology to the binding neighbour.

Definition 15 (Projection of Completion Tree). Let $AO_i = \langle \mathbb{B}_i, O_i \rangle$ be an autonomous ontology, and $\mathbb{B}_{ij} \in \mathbb{B}_i$ the set of semantic binding of foreign entities from another autonomous ontology AO_j . Let \mathcal{T} be a clash-free completion tree with root x_0 in AO_i . The projection of \mathcal{T} *w.r.t.* \mathbb{B}_{ij} , denoted as $P(\mathcal{T}, \mathbb{B}_{ij})$, is a completion tree $\mathcal{T}' = \langle x'_0, N', E', \mathcal{L} \rangle$ generated in the following way:

³ Since subsumption relation $C \sqsubseteq D$ in AO_i *w.r.t.* $\mathbb{A}\mathbb{O}_I$ iff $C \sqcap \neg D$ is unsatisfiable in AO_i *w.r.t.* $\mathbb{A}\mathbb{O}_I$.

⁴ Here we assume λ is in negation normal form; *i.e.*, negation is only applied to class names.

1. $N' = \{x' \mid \text{if } x \in N\}$,
2. $E' = \{\langle x', y' \rangle \mid \text{if } \langle x, y \rangle \in E\}$,
3. $\mathcal{L}(x') = \{(j:j:C) \mid \text{if } (i:j:C) \in \mathcal{L}(x) \text{ and } (i:j:C) \in \mathbb{B}_{ij}\}$,
4. $\mathcal{L}(\langle x', y' \rangle) = \{(j:j:R) \mid \text{if } (i:j:R) \in \mathcal{L}(\langle x, y \rangle) \text{ and } (i:j:R) \in \mathbb{B}_{ij}\}$,

4.2 Cautious Reasoning

Cautious reasoning relies on the knowledge of an ontology as well as the knowledge of its binding neighbour ontologies.

Given an autonomous ontology space $\mathbb{A}\mathbb{O}_I = \{AO_i\}_{i \in I}$, the procedure $\mathcal{C}\text{-Tab}(\mathbb{A}\mathbb{O}_I, k, \lambda)$ ⁵ verifies the satisfiability of an \mathcal{ALC} class description λ in ontology AO_k under the cautious semantics *w.r.t.* $\mathbb{A}\mathbb{O}_I$.

Algorithm A-1. $\mathcal{C}\text{-Tab}(\mathbb{A}\mathbb{O}_I, k, \lambda)$

- 1: Let $\mathcal{T} := \text{Tab}(AO_k, \langle x_0, \{x_0\}, \emptyset, \{\mathcal{L}(x_0) = \{\lambda\}\})$ // local expansion *w.r.t.* AO_k
- 2: **repeat**
- 3: **if** \mathcal{T} has a clash **then**
- 4: **return** *unsatisfiable*
- 5: **end if**
- 6: **for** every binding neighbour autonomous ontology AO_i ($i \in I$) of AO_k **do**
- 7: **if** there exist $S(i)$ -bound maximal sub-trees $\mathcal{T}_1, \dots, \mathcal{T}_n$ of \mathcal{T} with roots x_1, \dots, x_n , respectively **then**
- 8: $\mathcal{T}'_1 := P(\mathcal{T}_1, \mathbb{B}_{ki}), \dots, \mathcal{T}'_n := P(\mathcal{T}_n, \mathbb{B}_{ki})$ // sub-trees projection to AO_i
- 9: $\mathcal{T}'_1 := \text{Tab}(AO_i, \mathcal{T}'_1), \dots, \mathcal{T}'_n := \text{Tab}(AO_i, \mathcal{T}'_n)$ // local expansion of $\mathcal{T}'_1, \dots, \mathcal{T}'_n$ *w.r.t.* AO_i
- 10: **if** any of $\mathcal{T}'_1, \dots, \mathcal{T}'_n$ has a clash **then**
- 11: **if** \mathcal{T} is backtrackable **then**
- 12: $\mathcal{T} := \text{backtrack}(\text{Tab}(AO_i, \mathcal{T}))$ // backtrack and expand
- 13: **else**
- 14: **return** *unsatisfiable*
- 15: **end if**
- 16: **end if**
- 17: **end if**
- 18: **end for**
- 19: **return** *satisfiable*
- 20: **until** false

In this algorithm, \mathcal{T} is initialized with a root x_0 with $\mathcal{L}(x_0) = \{\lambda\}$, and is expanded by local completion rules *w.r.t.* AO_k (line 1 of **A-1**). As \mathcal{T} can have multiple binding neighbour ontologies, each of them should be taken care (line 6 of **A-1**). Note that \mathcal{T} might not be k -bound, the algorithm just project the maximal k -bound sub-trees, and then expand them by local completion rules *w.r.t.* AO_i , and expanded *w.r.t.* $\mathbb{A}\mathbb{O}_I$ (lines 7-9 of **A-1**). If any of the projected sub-tree has a clash, \mathcal{T} needs to be backtracked (line 12 of **A-1**), expanded and start the checking all over again.

Theorem 3. $\mathcal{C}\text{-Tab}(\mathbb{A}\mathbb{O}_I, k, \lambda)$ is a decision procedure to verify the cautious semantics satisfiability of an \mathcal{ALCN} -class description λ in ontology AO_k *w.r.t.* $\mathbb{A}\mathbb{O}_I$.

⁵ This Algorithm is originated from [12], in which only positive concepts/roles can be projected to the original ontology, but the approach in this paper does not have this restriction.

To prove the theorem, we need to show that: (1) The algorithm always terminates. (2) The algorithm returns unsatisfiable if $\mathbb{A}\mathbb{O}_I \not\models_C \lambda$. (3) The algorithm returns satisfiable if $\mathbb{A}\mathbb{O}_I \models_C \lambda$. Due to limited space, we skip the detail.

4.3 Brave Reasoning

Brave reasoning not only relies on its binding neighbors, but also trusts the neighbors of its binding neighbors.

Given an autonomous ontology space $\mathbb{A}\mathbb{O}_I = \{AO_i\}_{i \in I}$, the procedure $\mathcal{B}\text{-Tab}(\mathbb{A}\mathbb{O}_I, k, \lambda)$ verifies the satisfiability of an \mathcal{ALC} class description λ in ontology AO_k w.r.t. $\mathbb{A}\mathbb{O}_I$ under brave semantics. It calls a recursive procedure $\mathcal{DB}\text{-Tab}(\mathbb{A}\mathbb{O}_I, k, \mathcal{T})$ to expand a completion tree \mathcal{T} of AO_k w.r.t. $\mathbb{A}\mathbb{O}_I$ under brave semantics.

Algorithm A-2. $\mathcal{B}\text{-Tab}(\mathbb{A}\mathbb{O}_I, k, \lambda)$

- 1: Let $\mathcal{T} := \langle x_0, \{x_0\}, \emptyset, \{\mathcal{L}(x_0) = \{\lambda\}\} \rangle$
- 2: $\mathcal{T} := \mathcal{DB}\text{-Tab}(k, \mathcal{T})$
- 3: **if** \mathcal{T} has a clash **then**
- 4: **return** *unsatisfiable*
- 5: **else**
- 6: **return** *satisfiable*
- 7: **end if**

Algorithm A-3: $\mathcal{DB}\text{-Tab}(\mathbb{A}\mathbb{O}_I, k, \mathcal{T})$

- 1: let $\mathcal{T} := \text{Tab}(AO_k, \mathcal{T})$ // local expansion w.r.t. AO_k
- 2: **repeat**
- 3: **if** \mathcal{T} has a clash **then**
- 4: **return** \mathcal{T} // *unsatisfiable*
- 5: **end if**
- 6: **for** every binding neighbour autonomous ontology AO_i of AO_k **do**
- 7: **if** there exist $S(i)$ -bound maximal sub-trees $\mathcal{T}_1, \dots, \mathcal{T}_n$ of \mathcal{T} with roots x_1, \dots, x_n , respectively **then**
- 8: $\mathcal{T}'_1 := P(\mathcal{T}_1, \mathbb{B}_{ki}), \dots, \mathcal{T}'_n := P(\mathcal{T}_n, \mathbb{B}_{ki})$ // sub-trees projection from AO_k to AO_i
- 9: $\mathcal{T}'_1 := \mathcal{DB}\text{-Tab}(\mathbb{A}\mathbb{O}_I, i, \mathcal{T}'_1), \dots, \mathcal{T}'_n := \mathcal{DB}\text{-Tab}(\mathbb{A}\mathbb{O}_I, i, \mathcal{T}'_n)$ // recursive calling $\mathcal{DB}\text{-Tab}$ for brave reasoning on $\mathcal{T}'_1, \dots, \mathcal{T}'_n$ w.r.t. AO_i
- 10: **if** any of $\mathcal{T}'_1, \dots, \mathcal{T}'_n$ has a clash **then**
- 11: **if** \mathcal{T} is backtrackable **then**
- 12: $\mathcal{T} := \text{backtrack}(\text{Tab}(\mathcal{O}_i, \mathcal{T}))$ // backtrack and expand
- 13: **else**
- 14: **return** \mathcal{T} // *unsatisfiable*
- 15: **end if**
- 16: **else**
- 17: $\mathcal{T}_1 := \beta(\mathbb{B}_{ki}, \mathcal{T}_1, \mathcal{T}'_1), \dots, \mathcal{T}_n := \beta(\mathbb{B}_{ki}, \mathcal{T}_n, \mathcal{T}'_n)$ //back-project
- 18: **end if**
- 19: **end if**
- 20: **end for**
- 21: **if** \mathcal{T} is not changed, **then**
- 22: **return** \mathcal{T} // *satisfiable*
- 23: **end if**
- 24: **until** false

In this algorithm, initially \mathcal{T} has a root x_0 with $\mathcal{L}(x_0) = \{\lambda\}$ (line 1 of **A-2**), and then it is expanded by local completion rules *w.r.t.* AO_k (line 1 of **A-3**). As \mathcal{T} can have multiple binding neighbour ontologies, each of them should be taken care (line 6 of **A-3**). Note that \mathcal{T} might not be $S(i)$ -bound, maximal $S(i)$ -bound sub-trees then should be projected, and expanded by local completion rules *w.r.t.* AO_i , if possible it also need to project to the binding neighbour of AO_i . (lines 7-9 of **A-3**). If any of the projected sub-tree has a clash, \mathcal{T} needs to be backtracked (line 12 of **A-3**), expanded and start the checking all over again; otherwise, we need to back-project the new $S(i)$ -bound labels back to \mathcal{T} (line 17 of **A-3**). The algorithm **A-3** would not stop until \mathcal{T} is not changed.

Theorem 4. \mathcal{B} -Tab($\mathbb{A}\mathbb{O}_I, k, \lambda$) is a decision procedure to verify the brave semantics satisfiability of an \mathcal{ALCN} -class description λ in ontology AO_k *w.r.t.* $\mathbb{A}\mathbb{O}_I$.

To prove the theorems, we need to show that: (1) The algorithms always terminates. (2) The algorithm returns unsatisfiable if $\mathbb{A}\mathbb{O}_I \not\models_{\mathcal{B}} \lambda$. (3) The algorithm returns satisfiable if $\mathbb{A}\mathbb{O}_I \models_{\mathcal{B}} \lambda$. Due to limited space, we skip the proof.

5 Conclusions

In general understanding, ontologies are used for describing the structure of domain knowledge. Techniques for (partial) ontology reuse are important for ontology building, ontology discovery, and practical application of ontologies.

How to realise semantic cooperation among multiple ontologies is an important problem in the field of (partial) ontology reuse. In this paper we have proposed and analyzed a new framework for managing multiple ontologies that both preserves the autonomy of individual ontologies and also enables the semantic cooperation of different ontologies. We have also proposed two different reasoning mechanisms, called cautious reasoning and brave reasoning, for this framework and studied their properties. As we discussed in Section-1, this work is related to DDL [3], C-OWL [4], \mathcal{E} -Connection based approach [10], P-DL [2], Semantic Importing [12], and conservative extension [9].

Ontology modularization [15,16,8,13,11] is another interesting problem in this field. It attempts to partition one ontology and isolate functional modules. How to present formal functional modules in a single ontology by semantic binding (as proposed in this paper) could be an interesting extension of this approach.

Another interesting task is to explore whether our approach to ontology spaces and semantic binding can be applied to problems of forgetting, importing and extending in ontology spaces.

A natural task for the immediate future is to implement our two reasoning mechanisms and empirically evaluate their performance.

Acknowledgment. Thanks to Luciano Serafini at ITC-IRST in Trento Italy for discussions on primary ideas of this paper. This work was partially supported by the Australia Research Council (ARC) Discovery Project 0666107.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge (2003)
2. Bao, J., Caragea, D., Honavar, V.: Towards collaborative environments for ontology construction and sharing. In: *Proc. of the International Symposium on Collaborative Technologies and Systems (CTS 2006)*, pp. 90–108. IEEE Computer Society Press, Los Alamitos (2006)
3. Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantic* 1, 153–184 (2003)
4. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing ontologies. In: *Proc. of the 2nd International Semantic Web Conference (ISWC 2003)*. USA, pp. 164–179 (2003)
5. Ghidini, C., Giunchiglia, F.: Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence* 127(2), 221–259 (2001)
6. Giunchiglia, F.: Contextual reasoning. *Epistemologia*, special issue on *I Linguaggi e le Macchine XVI*, 345–364 (1993)
7. Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics, or: how we can do without modal logics. *Artificial Intelligence* 65(1), 29–70 (1994)
8. Grau, B.C., Parsia, B., Sirin, E., Kalyanpur, A.: Modularizing OWL Ontologies. In: *Proc. of the KCAP 2005 Workshop on Ontology Management* (2005)
9. Grau, B.C., Kutz, O.: Modular ontology languages revisited. In: *Proc. of the IJCAI 2007 Workshop on Semantic Web for Collaborative Knowledge Acquisition* (2007)
10. Grau, B.C., Parsia, B., Sirin, E.: Working with multiple ontologies on the semantic web. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) *ISWC 2004*. LNCS, vol. 3298, Springer, Heidelberg (2004)
11. Grau, B.C., Parsia, B., Sirin, E., Kalyanpur, A.: Modularity and web ontologies. In: *Proc. of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)*. UK (June 2006)
12. Pan, J.Z., Serafini, L., Zhao, Y.: Semantic import: An approach for partial ontology reuse. In: *Proc. of the ISWC2006 Workshop on Modular Ontologies (WoMO)* (2006)
13. Paslaru, E.B., Mochol, M., Tolksdorf, R.: Case Studies on Ontology Reuse. In: *I-Know 2005*. *Proc. of the 5th International conference on Knowledge Management*, Austria (2005)
14. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: OWL web ontology language semantics and abstract syntax (February 10, 2004)
15. Rector, A.L.: Modularisation of domain ontologies implemented in description logics and related formalisms including owls. In: *Proc. of the 2nd International Conference on Knowledge Capture (K-CAP)*. USA (2003)
16. Stuckenschmidt, H., Klein, M.: Structure-based partitioning of large concept hierarchies. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) *ISWC 2004*. LNCS, vol. 3298, Springer, Heidelberg (2004)
17. Zhao, Y., Serafini, L., Giunchiglia, F.: Autonomous ontology: Operations and semantics **or** local semantics with semantic binding on foreign entity. In: Mizoguchi, R., Shi, Z., Giunchiglia, F. (eds.) *ASWC 2006*. LNCS, vol. 4185, Springer, Heidelberg (2006)