Approximate Query Reformulation for Ontology Integration

Jun-ichi Akahani, Kaoru Hiramatsu, and Tetsuji Satoh
NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan
{akahani, hiramatu, satoh}@cslab.kecl.ntt.co.jp

Abstract

This paper proposes an approximate query reformulation framework for integrating heterogeneous ontologies. In order to achieve semantic interoperability in the Semantic Web, multiple ontologies have to be integrated. Ontology integration requires approximation mechanisms, since often no perfectly corresponding ontologies exist. However, most previous research efforts on ontology integration have not provided clear semantics for approximation. We have therefore proposed a framework for approximate query reformulation. In this framework, a query represented in one ontology is reformulated approximately into a query represented in another ontology based on an ontology mapping specification. In this paper, we focus on a fragment of OWL DL and provide a reformulation method for value restrictions and negation.

1 Introduction

Ontologies play a central role in the Semantic Web. However, the decentralized nature of the Web makes it difficult to construct or standardize a single ontology. Regional information is one reason for this, because ontologies vary from region to region due to the cultural differences. Ontologies also vary over time. Different people may update or customize an ontology independently. In such cases, one may query based on an updated ontology for the original ontology, or vice versa. We thus have to integrate heterogeneous ontologies both in temporal and spatial dimensions.

When integrating ontologies, it is rare to find those that correspond exactly; for example, there may be no corresponding class for Cajun restaurants in a Japanese ontology for restaurants. In such a case, one may use an approximation mechanism to replace "Cajun" with the "American" restaurant class in the Japanese ontology. However, most previous research efforts on ontology integration have not provided clear semantics for approximation.

We have therefore proposed a basic framework for approximate query reformulation [2]. In this framework, a query represented in one ontology is reformulated approximately into a query represented in another ontology based on an ontology mapping specification which is also described as an ontology. In order to characterize *closer* reformulation, the framework introduces two types of reformulation: *minimally-containing* reformulation and *maximally-contained* reformulation. However, this paper focuses on a simple ontology language and one-to-one subsumption mapping between ontologies.

Ontology description languages such as OWL [6] have much expressive power by providing constructs for value restrictions (`allValuesFrom` and `someValuesFrom`) and negation (`disjointWith` and `complementOf`). We therefore extend the approximate query reformulation framework for a fragment of OWL DL [6]. We focus on minimally-containing reformulation because they require non-standard inferences in Description Logics. Based on the formal framework, we provide a reformulation method for value restrictions and negation.

In the following sections, we first present an approximate query reformulation framework. We then provide a reformulation method for value restrictions and negation. Finally, we relate our framework to previous efforts in the field and present our conclusions.

2 Approximate Query Reformulation

In the approximate query reformulation framework [2], a query represented in one ontology is reformulated approximately into a query represented in another ontology based on an ontology mapping specification, as shown in Figure 1. This section and the next present a formal framework for approximate query reformulation. Throughout these sections, we use the simple example in the figure to illustrate our framework. More complex examples will be shown in the latter sections.

2.1 Queries in Ontologies

In this paper, we focus on a fragment of OWL DL [6] to describe ontologies. We use a Description Logic syntax for the sake of simplicity. Our ontology description language provides class description constructs for value restrictions ($\forall P.C$ for `allValuesFrom` and $\exists P.C$ for `someValuesFrom`) and negation ($\neg C$ for `complementOf`). We distinguish classes and properties in each ontology as follows.

Definition 1 (Class Description) *Given a set $\mathcal{C}_0^i$ of atomic classes and a set $\mathcal{P}^i$ of properties in an ontology O^i, a set $\mathcal{C}^i$ of classes in the ontology O^i consists of the following class descriptions:*

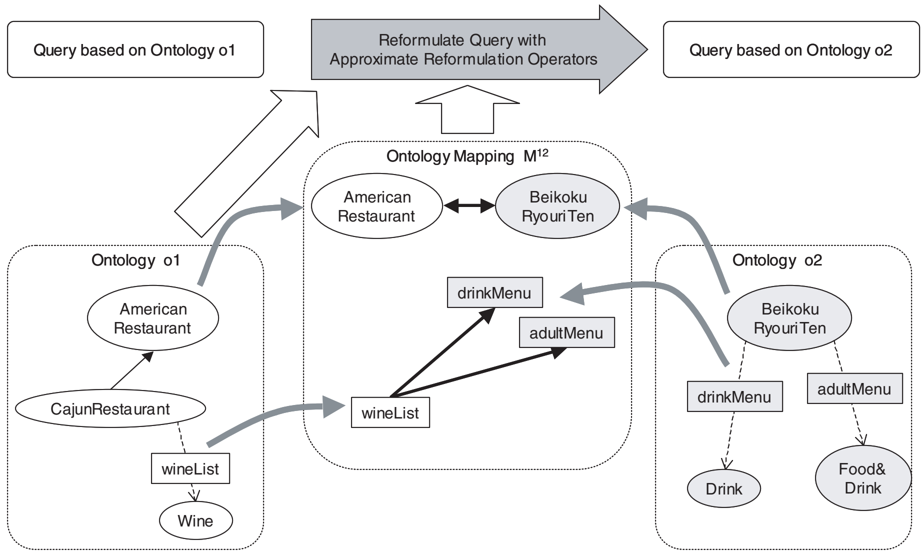

Figure 1: Approximate Query Reformulation Framework

- C^i *where* $C^i \in \mathcal{C}^i_0$.
- $\forall P^i.C^i$, $\exists P^i.C^i$, $\neg C^i$ *where* $C^i \in \mathcal{C}^i$, *and* $P^i \in \mathcal{P}^i$.

Ontologies are described by subsumption relations for classes and properties, and disjointness of classes. For the sake of simplicity, we do not distinguish individuals and data literals.

Definition 2 (Ontology Description) *Given a set $\mathcal{C}^i$ of classes and a set $\mathcal{P}^i$ of properties in an ontology O^i, and a set $\mathcal{L}$ of individuals and data literals, an ontology description of ontology O^i is a set $\mathcal{O}^i$ of axioms of the following forms:*

- $C^i_1 \sqsubseteq C^i_2$, $P^i_1 \sqsubseteq P^i_2$, $C^i_1 \sqsubseteq \neg C^i_2$,
- $a : C^i$, $\langle a, b \rangle : P^i$,

where $C^i_1, C^i_2, C^i \in \mathcal{C}^i$, $P^i_1, P^i_2, P^i \in \mathcal{P}^i$, *and* $a, b \in \mathcal{L}$.

For example, the ontology description $\mathcal{O}^1$ in ontology `o1` shown in Figure 1 contains the following axiom:

- $\texttt{CajunRestaurant}^1 \sqsubseteq \texttt{AmericanRestaurant}^1$.

(We denote the subsumption relation (i.e., `subClassOf` and `subPropertyOf`) as a solid arrow for easy visualization.)

The semantics of our ontology description language is defined using an interpretation function $\mathcal{I}$ in the usual way. Table 1 summarizes the syntax and semantics of our ontology description language. We denote $S \models A$ iff an axiom A logically follows from a set of axioms S.

We next define queries in ontologies. A query is a conjunction of a query about classes and a query about properties.

Definition 3 (Query) *Let $\mathcal{V}$ be a set of variables disjoint from $\mathcal{L}$. A query Q^i in ontology O^i is of the form*

$$Q^i_C \wedge Q^i_P,$$

where

- Q^i_C *is a conjunction of $C^i(x)$ where $C^i \in \mathcal{C}^i$ and $x \in \mathcal{L} \cup \mathcal{V}$,*
- Q^i_P *is a conjunction of $P^i(x, y)$ where $P^i \in \mathcal{P}^i$ and $x, y \in \mathcal{L} \cup \mathcal{V}$.*

For example, the following denotes a query about a Cajun restaurant that has Merlot on its wine list in ontology `o1`.

- $\texttt{CajunRestaurant}^1(x) \wedge \texttt{wineList}^1(x, `\texttt{Merlot}')$.

The answer to a query Q^i can be obtained by substituting variables $v_1, \cdots, v_n$ contained in Q^i by a tuple $\langle a_1, \cdots, a_n \rangle$ of objects. We denote this substitution σ. The answer set $A(Q^i)$ is a set of tuples such that $\mathcal{O}^i \models Q^i\sigma$. The semantic relation between different queries are defined as follows.

Definition 4 (Query Containment) *For queries Q_1 and Q_2, Q_1 is said to be contained in Q_2 (denoted by $Q_1 \sqsubseteq Q_2$) if $A(Q_1) \subseteq A(Q_2)$.*

2.2 Mapping among Multiple Ontologies

In our framework, a query represented in one ontology is reformulated approximately into a query represented in another ontology by using an ontology mapping specification. In this paper, we describe ontology mapping specifications using the ontology description language.

Definition 5 (Ontology Mapping Specification) *Ontology mapping M^{ij} between ontology O^i and O^j is a set of axioms of the following forms:*

- $C^i \sqsubseteq C^j$, $C^j \sqsubseteq C^i$, $C^i \sqsubseteq \neg C^j$, $C^j \sqsubseteq \neg C^i$,
- $P^i \sqsubseteq P^j$, $P^j \sqsubseteq P^i$,

where $C^i \in \mathcal{C}^i$, $C^j \in \mathcal{C}^j$, $P^i \in \mathcal{P}^i$ *and* $P^j \in \mathcal{P}^j$.

The mapping range $R(M^{ij})$ of ontology mapping M^{ij} is defined to be a set of classes and properties in ontology O^j that appear in M^{ij}.

For example, the ontology mapping M^{12} between ontologies `o1` and `o2` in Figure 1 contains the following axioms:

- $\texttt{AmericanRestaurant}^1 \sqsubseteq \texttt{BeikokuRyouriTen}^2$.

OWL Construct	Syntax	Semantics
`allValuesFrom`	$\forall P^i.C^i$	$\{x \mid \forall y.\langle x,y\rangle \in \mathcal{I}(P^i) \supset y \in \mathcal{I}(C^i)\}$
`someValuesFrom`	$\exists P^i.C^i$	$\{x \mid \exists y.\langle x,y\rangle \in \mathcal{I}(P^i) \wedge y \in \mathcal{I}(C^i)\}$
`complementOf`	$\neg C^i$	$\mathcal{I}(\Delta) \setminus \mathcal{I}(C^i)$
`subClassOf`	$C_1^i \sqsubseteq C_2^i$	$\mathcal{I}(C_1^i) \subseteq \mathcal{I}(C_2^i)$
`subPropertyOf`	$P_1^i \sqsubseteq P_2^i$	$\mathcal{I}(P_1^i) \subseteq \mathcal{I}(P_2^i)$
`disjointWith`	$C_1^i \sqsubseteq \neg C_2^i$	$\mathcal{I}(C_1^i) \cap \mathcal{I}(C_2^i) = \phi$
(individual axioms)	$a : C^i$	$\mathcal{I}(a) \in \mathcal{I}(C^i)$
(individual axioms)	$\langle a,b\rangle : P^i$	$\mathcal{I}(\langle a,b\rangle) \in \mathcal{I}(P^i)$

Table 1: Syntax and Semantics of Ontology Description Language

- $\texttt{BeikokuRyouriTen}^2 \sqsubseteq \texttt{AmericanRestaurant}^1$.
- $\texttt{wineList}^1 \sqsubseteq \texttt{drinkMenu}^2$.
- $\texttt{wineList}^1 \sqsubseteq \texttt{adultMenu}^2$.

There may be many possible reformulated queries, but we prefer closer reformulation. We therefore adapt and extend the notion of *maximally-contained* reformulation [4] in the database literature. Specifically, we characterize two kinds of reformulation: minimally-containing reformulation and maximally-contained reformulation. In minimally-containing reformulation, the reformulated query minimally covers the original query. On the other hand, in maximally-contained reformulation, the reformulated query is maximally covered by the original query.

Assuming that ontology mapping M^{ij} is consistent with ontologies $\mathcal{O}^i$ and $\mathcal{O}^j$, we characterize approximate query reformulation using query containment in the merged ontology $\mathcal{O}^i \cup M^{ij} \cup \mathcal{O}^j$. We extend the definition of the answer set $A(Q)$ to be a set of tuples such that $\mathcal{O}^i \cup M^{ij} \cup \mathcal{O}^j \models Q\sigma$. Approximate query reformulation is defined as follows.

Definition 6 (Approximate Query Reformulation) *Let Q^i be a query in ontology O^i and Q^j be a query in ontology O^j described by classes and properties in the mapping range $R(M^{ij})$ of ontology mapping M^{ij}.*

- *Q^j is an equivalent reformulation of Q^i if $Q^j \sqsubseteq Q^i$ and $Q^i \sqsubseteq Q^j$.*
- *Q^j is a* **minimally-containing** *reformulation of Q^i if $Q^i \sqsubseteq Q^j$ and there is no other query Q_1^j such that $Q^i \sqsubseteq Q_1^j$ and $Q_1^j \sqsubseteq Q^j$.*
- *Q^j is a* **maximally-contained** *reformulation of Q^i if $Q^j \sqsubseteq Q^i$ and there is no other query Q_1^j such that $Q^j \sqsubseteq Q_1^j$ and $Q_1^j \sqsubseteq Q^i$.*

Recall the example query above. A reformulated query

- $\texttt{BeikokuRyouriTen}^2(x) \wedge \texttt{drinkMenu}^2(x, \text{`}\texttt{Merlot}\text{'})$

is not a minimally-containing reformulation, as there is a minimally-containing reformulated query as follows;

- $\texttt{BeikokuRyouriTen}^2(x) \wedge \texttt{drinkMenu}^2(x, \text{`}\texttt{Merlot}\text{'}) \wedge \texttt{adultMenu}^2(x, \text{`}\texttt{Merlot}\text{'})$

3 Approximate Reformulation Operators

In this section, we provide approximate reformulation operators, which we call the most special generalizers, for minimally-containing reformulation.

A reformulated query consists of classes and properties appeared in the *range* of ontology mapping. Intuitively, classes and properties in a minimally-containing reformulation should minimally subsume those in the original query. Therefore, it is necessary to calculate the least upper bounds and the greatest lower bounds for classes and properties. We first define the least upper bounds and greatest lower bounds for a class and a property. This definition is an extended version of [7].

Definition 7 (Least Upper Bounds and Greatest Lower Bounds) *Let C be a class, O be a ontology description, and TC be a set of classes, then the least upper bounds $LUB(C, O, TC)$ and greatest lower bounds $GLB(C, O, TC)$ are defined as follows:*

- *$LUB(C, O, TC) = \{C' \mid C' \in TC$, $O \models C \sqsubseteq C'$ and there is no other $C'_1 \in TC$ such that $O \models C \sqsubseteq C'_1$ and $O \models C'_1 \sqsubseteq C'\}$.*
- *$GLB(C, O, TC) = \{C' \mid C' \in TC$, $O \models C' \sqsubseteq C$ and there is no other $C'_1 \in TC$ such that $O \models C' \sqsubseteq C'_1$ and $O \models C'_1 \sqsubseteq C\}$.*

The least upper bounds and greatest lower bounds for a property are defined similarly.

Minimally-containing reformulation requires calculation of the least upper bounds of classes and properties in the original query with respect to the merged ontology. For example, the least upper bounds of a class $\texttt{CajunRestaurant}^1$ with respect to ontology $\mathcal{O}^1 \cup M^{12} \cup \mathcal{O}^2$ in the mapping range $R(M^{ij})$ is the following set.

- $LUB(\texttt{CajunRestaurant}^1, \mathcal{O}^1 \cup M^{12} \cup \mathcal{O}^2, R(M^{12})) = \{\texttt{BeikokuRyouriTen}^2\}$.

Similarly, the least upper bounds of a property $\texttt{wineList}^1$ with respect to ontology $\mathcal{O}^1 \cup M^{12} \cup \mathcal{O}^2$ in the mapping range $R(M^{12})$ is the following set.

- $LUB(\texttt{wineList}^1, \mathcal{O}^1 \cup M^{12} \cup \mathcal{O}^2, R(M^{12})) = \{\texttt{drinkMenu}^2, \texttt{adultMenu}^2\}$.

Using the least upper bounds, we can define the most special generalizers for class queries and property queries.

Definition 8 (Most Special Generalizers) *Let $\mathcal{O}^i$ and $\mathcal{O}^j$ be ontology descriptions in ontology O^i and O^j and M^{ij} be an ontology mapping. A most special generalizer for a class query $C^i(x)$ is defined as follows:*

$$MSG(C^i(x), \mathcal{O}^i, \mathcal{O}^j, M^{ij}) = C_1^j(x) \wedge \cdots \wedge C_n^j(x),$$

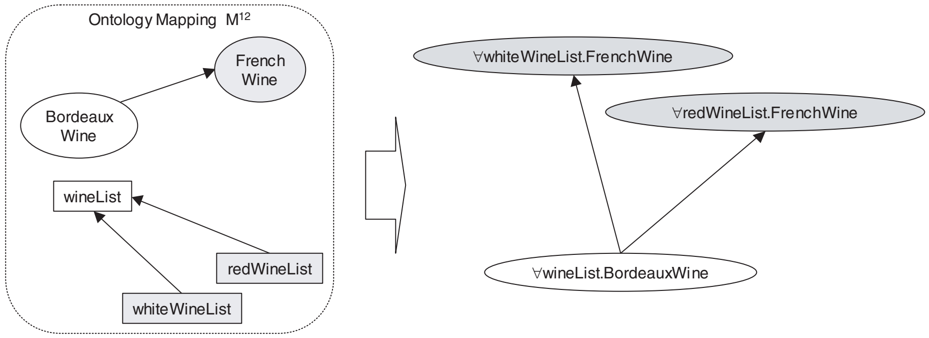

Figure 2: Least Upper Bounds for Value Restrictions

where $LUB(C^i, \mathcal{O}^i \cup M^{ij} \cup \mathcal{O}^j, R(M^{ij})) = \{C_1^j, \cdots, C_n^j\}$.

A most special generalizer for a property query $P^i(x,y)$ is defined as follows:

$$MSG(P^i(x,y), \mathcal{O}^i, \mathcal{O}^j, M^{ij}) = P_1^j(x,y) \wedge \cdots \wedge P_n^j(x,y),$$

where $LUB(P^i, \mathcal{O}^i \cup M^{ij} \cup \mathcal{O}^j, R(M^{ij})) = \{P_1^j, \cdots, P_n^j\}$.

Based on the above examples of least upper bounds, we have the following most special generalizers.

$MSG(\texttt{CajunRestaurant}^1(x), \mathcal{O}^1, \mathcal{O}^2, M^{12}) =$
$\texttt{BeikokuRyouriTen}^2(x)$.

$MSG(\texttt{wineList}^1(x, \text{"}\texttt{Merlot}\text{"}), \mathcal{O}^1, \mathcal{O}^2, M^{12}) =$
$\texttt{drinkMenu}^2(x, \text{"}\texttt{Merlot}\text{"})$
$\wedge\, \texttt{adultMenu}^2(x, \text{"}\texttt{Merlot}\text{"})$.

Applying these most special generalizers, the query in ontology `o1`

- $\texttt{CajunRestaurant}^1(x) \wedge \texttt{wineList}^1(x, \text{`}\texttt{Merlot}\text{'})$

is reformulated approximately into the query in ontology `o2`

- $\texttt{BeikokuRyouriTen}^2(x) \wedge \texttt{drinkMenu}^2(x, \text{`}\texttt{Merlot}\text{'}) \wedge \texttt{adultMenu}^2(x, \text{`}\texttt{Merlot}\text{'})$.

The following theorem assures the correctness of our framework.

Theorem 1 *Let Q^i be a query in ontology O^i, then*

- *if Q^i is reformulated into Q_g^j in ontology O^j by the most special generalizers, then Q_g^j is a minimally-containing reformulation of Q^i.*

4 Computing Least Upper Bounds

As we have seen in the previous section, our approximate query reformulation framework requires computation of least upper bounds. This computation of least upper bounds for classes and properties varies according to the expressive power of ontology description languages. Our ontology description language only allows subsumption relationship for properties that are used in normal ontology description languages, such as OWL. It is therefore easy to compute the least upper bounds for properties using subsumption relationship.

However, our ontology description language allows value restrictions and negation for class description. In this section, we provide least upper bounds for value restrictions and negation. In the following, we write $LUB(C)$ instead of $LUB(C, \mathcal{O}^i \cup M^{ij} \cup \mathcal{O}^j, R(M^{ij}))$ for simplicity.

4.1 Value Restrictions

Least upper bounds for a value restriction such as $\forall P^i.C^i$ are a set of value restrictions in the target ontology. Because a value restriction consists of a class and a property, we have to take into consideration both class and property hierarchies.

We start with the observation that the following holds.

- If $P_1 \sqsubseteq P_2$ and $C_1 \sqsubseteq C_2$, then $\forall P_2.C_1 \sqsubseteq \forall P_1.C_2$ and $\exists P_1.C_1 \sqsubseteq \exists P_2.C_2$.

For example, if a white wine list property is a sub-property of a wine list property:

- $\texttt{whiteWineList}^2 \sqsubseteq \texttt{wineList}^1$,

and a Bordeaux wine class is a sub-class of a French wines class:

- $\texttt{BordeauxWine}^1 \sqsubseteq \texttt{FrenchWine}^2$,

then we have the following.

- $\forall \texttt{wineList}^1.\texttt{BordeauxWine}^1$
$\sqsubseteq \forall \texttt{whiteWineList}^2.\texttt{FrenchWine}^2$

Intuitively, a class that has only Bordeaux wines on its wine list is subsumed by a class that has only French wines on its white wine list, because the latter may also have Italian rose wine.

Thus, a value restriction that subsumes $\forall P^i.C^i$ consists from a sub-property (e.g., $\texttt{whiteWineList}^2$) of property P^i (e.g., $\texttt{wineList}^1$). Therefore, computation of the least upper bounds for value restriction $\forall P^i.C^i$ requires that of the greatest lower bounds of property P^i. A greatest lower bound is semantically a disjunction of all the element of the greatest lower bound. As properties occur negatively in $\forall P.C$, negation of the greatest lower bound is a conjunction of negation of each element.

Strictly speaking, the following proposition holds from the semantics of value restrictions.

Proposition 1 (Least Upper Bounds for Value Restrictions) *The least upper bounds for value restrictions are defined as follows:*

- $LUB(\forall P^i.C^i) = \{\forall P_k^j.C_l^j \mid P_k^j \in GLB(P^i)$ *and* $C_l^j \in LUB(C^i)\}$.
- $LUB(\exists P^i.C^i) = \{\exists P_k^j.C_l^j \mid P_k^j \in LUB(P^i)$ *and* $C_l^j \in LUB(C^i)\}$.

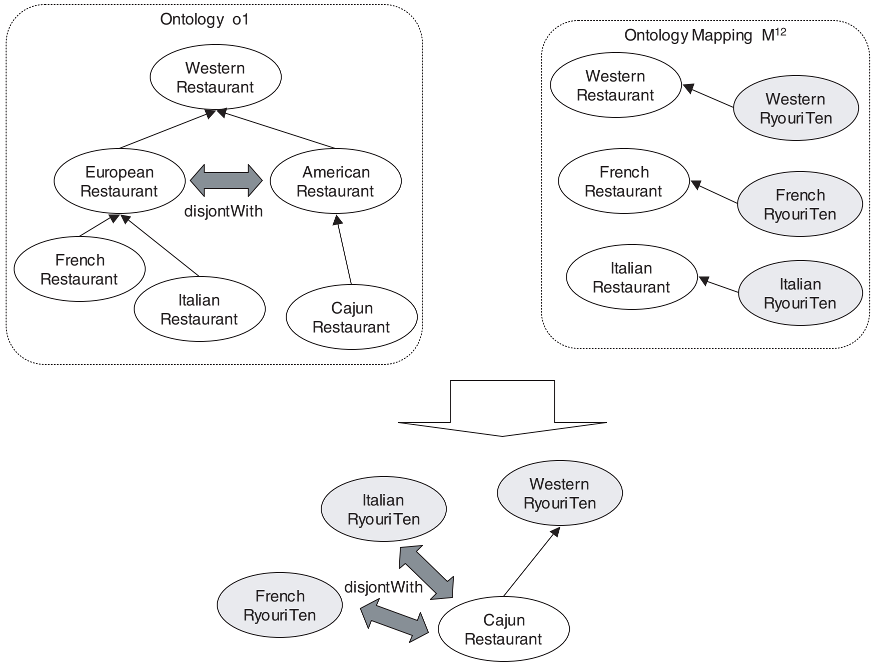

Figure 3: Least Upper Bounds for Negation

For example, the following can be calculated from the ontology mapping in Figure 2.

- $GLB(\mathtt{wineList}^1) = \{\mathtt{whiteWineList}^2, \mathtt{redWineList}^2\}$.
- $LUB(\mathtt{BordeauxWine}^1) = \{\mathtt{FrenchWine}^2\}$.

We thus have the following as shown in Figure 2.

- $LUB(\forall\mathtt{wineList}^1.\mathtt{BordeauxWine}^1 = \{\forall\mathtt{whiteWineList}^2.\mathtt{FrenchWine}^2, \forall\mathtt{redWineList}^2.\mathtt{FrenchWine}^2\}$.

Note that least upper bounds for a value restriction are computed recursively. A naive algorithm may cause an infinite loop. However, a simple blocking mechanism is sufficient to avoid infinite recursion.

4.2 Negation

Negation is useful in practical applications. Consider, for example, the ontology description and the ontology mapping in Figure 3. The ontology description contains the following axioms. Suppose that one would like to make a query about a "Cajun" restaurant in ontology o2. There is no class corresponding to "Cajun" nor "American" restaurants. However, the "American" restaurant class is defined to be disjoint with "European" restaurant class, and there are corresponding classes for subclasses (e.g., "French" and "Italian" restaurants) of the "European" restaurant class. It is possible to construct a reformulated query for the query about a "Cajun" restaurant using these classes.

The ontology o1 contains the following.

- $\mathtt{AmericanRestaurant}^1 \sqsubseteq \neg\mathtt{EuropeanRestaurant}^1$.

Thus, computation of $LUB(\mathtt{CajunRestaurant}^1)$ requires that of $LUB(\neg\mathtt{EuropeanRestaurant}^1)$. It is easy to show that

- if $C_1 \sqsubseteq C_2$, then $\neg C_2 \sqsubseteq \neg C_1$.

Therefore, the class $\neg\mathtt{EuropeanRestaurant}^1$ is subsumed by negation of each subclass as follows:

- $\neg\mathtt{EuropeanRestaurant}^1 \sqsubseteq \neg\mathtt{FrenchRestaurant}^1$.
- $\neg\mathtt{EuropeanRestaurant}^1 \sqsubseteq \neg\mathtt{ItalianRestaurant}^1$.

Strictly speaking, the following proposition holds from the semantics of negation.

Proposition 2 (Least Upper Bounds for Negation) *The least upper bounds for negation are defined as follows:*

- $LUB(\neg C^i) = \{\neg C_k^j \mid C_k^j \in GLB(C^i)\}$.

Applying the proposition to above example, we have

- $LUB(\neg\mathtt{EuropeanRestaurant}^1) = \{\neg\mathtt{FrenchRyouriTen}^2, \neg\mathtt{ItalianRyouriTen}^2\}$.

Thus, a class for "Cajun" restaurant is approximated to a generalized class "WesternRyouriTen" except "French" and "Italian" as shown in Figure 3.

- $LUB(\mathtt{CajunRestaurant}^1) = \{\mathtt{WesternRyouriTen}^2, \neg\mathtt{FrenchRyouriTen}^2, \neg\mathtt{ItalianRyouriTen}^2\}$.

Computation of greatest lower bounds may require the greatest lower bounds for value restrictions. The following is a dual of Proposition 1.

Proposition 3 (Greatest Lower Bounds for Value Restrictions) *The greatest lower bounds for value restrictions are defined as follows:*

- $GLB(\forall P^i.C^i) = \{\forall P^j_k.C^j_l \mid P^j_k \in LUB(P^i)$ *and* $C^j_l \in GLB(C^i)\}$.
- $GLB(\exists P^i.C^i) = \{\exists P^j_k.C^j_l \mid P^j_k \in GLB(P^i)$ *and* $C^j_l \in GLB(C^i)\}$.

5 Related Work

Approximate terminological query framework [8] provides a formal framework for query approximation. In this framework, query approximation is used to improve the efficiency in a single ontology. Thus, the authors did not provide ontology mapping. They also introduce query containment, but it is used as a measure for the degree of query approximation. On the other hand, we address approximate query reformulation between ontologies and introduce minimally-containing reformulation and maximally-contained reformulation.

The approximate information filtering framework [7] has also been proposed. However, the author only dealt with simple class hierarchies and the maximally-contained reformulation in our framework. On the other hand, we deal with complex class description such as value restrictions and negation and minimally-containing reformulation, which requires non-standard inference in Description Logics.

Most previous research efforts on ontology integration have used ad-hoc mapping rules between ontologies (as surveyed in [9]). This approach allows flexibility in ontology integration, but most works do not provide semantics for the mapping rules. One exception is the Ontology Integration Framework [3] which provides clear semantics for ontology integration by defining *sound* and *complete* semantic conditions for each mapping rule. However, each mapping rule and its semantic conditions have to be specified by users. It is therefore difficult to ensure consistency in the mapping rules. In contrast, our framework can generate *sound* and *complete* mapping rules by specifying ontology mapping. It is relatively easy to check the consistency, since ontology mapping specifications are described as an ontology.

6 Conclusions

In this paper, we presented an approximate query reformulation framework for a fragment of OWL DL. In our framework, a query in one ontology is reformulated approximately into a query in another ontology based on an ontology mapping specification. To characterize closer reformulation, we introduced two types of reformulation: minimally-containing reformulation and maximally-contained reformulation. For the former, we provided the most special generalizers to reformulate a class (or property) expression in an original query into conjunction of the least upper bounds of the class (or property). We also provided a reformulation method for value restrictions and negation.

This paper focused on a fragment of OWL DL. Specifically, our ontology description language lacks unnamed conjunctions of classes (`intersectionOf`), disjunctions of classes (`unionOf`), and number restrictions (`cardinality`, etc.). However, number restrictions can be incorporated into our framework. The main reason for this restriction is that our framework reformulates each conjunct of queries. Further investigation is necessary for this direction.

As ontology mapping specifications are described in an ontology language, our framework is useful for dealing with updated or customized ontologies. If one makes queries based on an updated ontology for the original ontology, our framework can reformulate the queries based on ontology mapping between the original and updated ontologies.

The approximate query reformulation framework has been incorporated into the GeoLinkAgent system [1]. In the prototype system, agents coordinate regional information services provided by the GeoLink system, which is used in the Digital City Kyoto prototype [5]. Approximate query reformulation is required for such domains that have cross-cultural aspects, because ontologies vary from region to region due to cultural differences.

References

[1] J. Akahani, K. Hiramatsu, Y. Furukawa, and K. Kogure. Agent-based coordination of regional information services. In *Digital Cities II: Computational and Sociological Approaches*, LNCS 2362, pages 283–291. Springer-Verlag, 2002.

[2] J. Akahani, K. Hiramatsu, and T. Satoh. Approximate query reformulation based on hierarchical ontology mapping. In *Proc. of International Workshop on Semantic Web Foundations and Application Technologies (SWFAT)*, pages 43–46, 2003.

[3] D. Calvanese, G. De Giacomo, and M. Lenzerini. A framework for ontology integration. In I. Cruz, S. Decker, J. Euzenat, and D. McGuinness, editors, *The Emerging Semantic Web*, pages 201–214. IOS Press, 2002.

[4] A. Y. Halevy. Theory of answering queries using views. *SIGMOD Record*, 29(4):40–47, 2000.

[5] T. Ishida, J. Akahani, K. Hiramatsu, K. Isbister, S. Lisowski, H. Nakanishi, M. Okamoto, Y. Miyazaki, and K. Tsutsuguchi. Digital City Kyoto: Towards a social information infrastructure. In *Proc. of International Workshop on Cooperative Information Agents (CIA-99)*, pages 23–35, 1999.

[6] P. F. Patel-Schneider, P. Hayes, and I. Horrocks. Owl web ontology language semantics and abstract syntax. http://www.w3.org/TR/2003/WD-owl-semantics-20030331/, 2003.

[7] H. Stuckenschmidt. Approximate information filtering with multiple classification hierarchies. *International Journal of Computational Intelligence and Applications*, 2(3):295–302, September 2002.

[8] H. Stuckenschmidt and F. van Harmelen. Approximating terminological queries. In *Proc. of the 4th International Conference on Flexible Query Answering Systems (FQAS'02)*. Springer-Verlag, 2002.

[9] H. Wache, T. Voegele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Huebner. Ontology-based integration of information – a survey of existing approaches. In *Proc. of IJCAI 2001 Workshop on Ontologies and Information Sharing*, 2001.