Legal Knowledge Based Systems

JURIX'96

Foundations of legal knowledge systems

The Foundation for Legal Knowledge Systems

Editors:
R.W. van Kralingen
H.J. van den Herik
J.E.J. Prins
M. Sergot
J. Zeleznikow

The publisher is aware that, in spite of all care taken by editors, authors and publisher, shortcomings may occur in this publication. For these, however, editors, authors and publisher cannot accept any liability.

Paper version: Tilburg University Press, ISBN 90-361-9657-4, NUGI 699

CONTENTS

LEGAL REASONING BY STRUCTURAL ANALOGY BASED ON GOAL DEPENDENT ABSTRACTION

Tokuyasu Kakuta
Department of Systems Science
Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku
Yokohama 226, JAPAN
email: kaku@cs.dis.titech.ac.jp

Makoto Haraguchi and Yoshiaki Okubo
Division of Electronics and Information Engineering
Hokkaido University
N-13 W-8, Sapporo 060, JAPAN
{makoto, yoshiaki}@db.huee.hokudai.ac.jp

Abstract

In this paper we present a notion of *structural similarities* under a terminological structure consisting of *is_a_subclass_of relations* and *roles*. Furthermore we show a method for finding such similarities using the terminological structure as a point of departure.

Since the similarities should accord with a legal goal we like to achieve, we first identify which parts of the terminological structure and the background knowledge are relevant to the goal. A notion of *Goal-Dependent Abstraction* is used to this purpose. The similarities are required to be consistent with the *is_a_subclass_of* relations and they are required to preserve the *value restrictions* of roles, focusing on the relevant terminological substructure and the relevant part of domain knowledge.

We use a simple example of a legal sign to show that the structural constraints on similarities greatly reduce the number of possible similarities and that they assist in making the similarities conceptually persuasive.

1 Introduction

Legal reasoning is concerned with several types of legal knowledge including *conceptual hierarchies* (*taxonomic hierarchies*). It is well known that such a hierarchy plays an important role in realizing our legal reasoning, since it roughly prescribes the conceptual relationships between concepts. For instance, Helic-II (Ohtake *et al*., 1994) utilizes a legal taxonomy to represent legal knowledge more compactly and to perform deductive reasoning more efficiently. In addition, we can also carry out legal analogical reasoning (Haraguchi,1996) and case-based reasoning (Ohtake *et al.*, 1994) by extracting information about similarities from the taxonomy.

However, besides the similarities drawn from the taxonomy, we have a number of similarities between concepts. Our legal conclusion will be affected by the similarities we consider the most important and relevant, just as the studies on legal case-based reasoning systems have already pointed out (*e.g.*, Ashley, 1990). From this viewpoint, we have proposed a computational method, called a *Goal-Dependent Abstraction* (GDA) (Okubo *et al*., 1994; Kakuta *et al*., 1996), to find similarities not derived from the taxonomic hierarchy.

When a legal rule is not applicable to a present case, GDA is invoked to compute a family of similar concepts including c_{rule}, for which the rule is bound to apply, and c_{case} appearing in the present case. Then we replace c_{rule} with c_{case} to form a *hypothetical rule*. The problem is to justify the similarity based on which the hypothetical rule is obtained. From this viewpoint, we have introduced two postulates in (Kakuta *et al*., 1996):

Substitutability Condition and *Similarity Inheritance Condition*.

The former requires similar things to share the same explanation structure of purposes of rule, which we call *grounds* of rule. For instance, suppose we have a legal sign “No car in the park”. If one of the reasons of legislation is that “it is dangerous for a car to enter the park”, then every dangerous thing can be a candidate to be replaced with **car**. That is, we consider that the purpose (grounds) of the sign is to avoid dangerous circumstances. Thus, **tank**, **horse** and also **hit_man** are the candidates. Furthermore, in order to strengthen the persuasive power in the legal arguments, the similar things are supposed to share the same explanation of grounds. For instance, if we argue that **car** is dangerous because it is large and movable, then **horse** and **tank** would be similar to **car**, but **hit_man** not so.

For we consider in this paper only legal rules that prohibit some acts, we can find the grounds of rule by a standard forward reasoning, given our legal domain theory also a hypothetical situation in which the rule is violated. For instance, we reason that something dangerous happens in a park from the domain theory of park, assuming further that a car enters it.

The latter postulate, *Similarity Inheritance Condition* (SIC for short) (Kakuta *et al*., 1996), is related to our terminological representation of legal domain. SIC asserts that similarities should be consistent with our taxonomy. We have introduced SIC to find similarities more efficiently and to make our similarities conceptually clearer. So if we introduce some other condition originated from the terminological knowledge, then we would obtain more powerful effect in searching similarities and in making the detected similarities more persuasive. From this viewpoint, this paper presents a new postulate on ***roles***, the standard primitive used in terminological knowledge representations (Nebel, 1990; Brachman *et al*., 1991):

> **Value Restriction Preservingness (VRP):**
> Our similarity should preserve the value restrictions on roles in defining the concepts, provided information about the roles are used in the explanation of grounds.

This condition corresponds to the *structure mapping theory* of analogy (Gentner, 1982; Winston, 1980). So we call our similarities *structural analogy*. Combining this new requirement with SIC, we can say that

> Focusing on a part of our knowledge base from which our grounds of rules are derived, the similarity we have to find is supposed to preserve both the *is_a_subclass_of* relationships and the value restrictions on roles. Consequently, those similar concepts meeting our requirements will share the same terminological structure relevant to the ground of rule.

The paper is organized as follows. Section 2 presents preliminary definitions and surveys our previous result. In Section 3 we discuss the problem of introducing value restrictions on roles. In Section 4, we illustrate how we find our similarity with a simple example. It is shown that the number of possible similarities is drastically reduced with VRP. In Section 5, we summarize our argument and make some comments on our future works.

2 Analogical legal reasoning based on GDA

In this section, we present preliminary definitions and briefly survey the previous work (Kakuta *et al*., 1996) based on which we introduce value restrictions on roles to make structural similarities.
A terminological language we use in this paper is basically a function-free *order-sorted language* with a *sort hierarchy* $(S,\leq)$, where S is a set of sorts that are conceptual classes

of individual objects, and $\leq$ denotes *is_a_subclass_of* relations. For instance, **car**$\leq$ **vehicle** means that the class **car** is a subclass of **vehicle**. Our domain theory consists of order-sorted clauses of the form $A \leftarrow B_1,\dots,B_n$, where A,B_j are order sorted atoms of the form $p(t_1,\dots,t_k)$. Each term t_j is a variable or a constant both of which are typed by some sort. For instance X:**car** denotes a variable whose possible values are instances of **car**. Similarly every individual is assumed to have its primary sort (*e.g.*, the primary sort of an individual **john** is **man**.).

To represent similarities not derived by the hierarchy $(S,\leq)$, we introduce a notion of *sort mapping* $\varphi : S \rightarrow S'$, where S' is a set of new sort symbols not appearing in S. Two sorts $\mathbf{s}_1$ and $\mathbf{s}_2$ is said to be *similar* with respect to φ if $\varphi(\mathbf{s}_1)=\varphi(\mathbf{s}_2)$. Thus the equation φ (**tank**)=φ (**gun**) asserts that **tank** and **gun** are similar with respect to φ . The mapping is denoted by φ {**tank**, **gun**} → α, for instance. In what follows, the term "similarity", "hypothetical hierarchy" and "sort mapping" are used alternatively. The mapping φ is easily extended to a mapping over a set of clauses as follows:

$$\varphi(p(x{:}s)) = p(x{:}\varphi(s))$$
$$\varphi(A \leftarrow B_1,\dots,B_n) = \varphi(A) \leftarrow \varphi(B_1),\dots,\varphi(B_n)$$

2.1 Ground of rule

Since the similarity is generally altered according to a legal goal, it is very important to prescribe what is regarded as the legal goal.

The legal rules we consider in this paper are supposed to prohibit some actions. For instance:

> "A vehicle is prohibited from entering a public park" (Hart,1958).
> Encoded rule C: **prohibit_in(X:vehicle,Y:public_park)** ← .

In a word, we investigate possible reasons showing why a vehicle is prohibited entering the park. It is natural to conceive that the rule is established because a vehicle would violate some state to be kept when it is allowed to enter on the contrary. From this viewpoint, we introduce a set of *CF-predicates* denoting the violation of state we expect and its related knowledge in our domain theory. Then, assuming tentatively the negated conclusion ¬**prohibit_in(X:vehicle,Y:public_park)** of legal rule, our task is to check if some CF-predicate can be drawn from the tentative assumption in addition to our domain theory. If a CF-predicate is derived, then we register it as a possible ground. When we have more than two possible grounds g_j for the rule C in inquiry, we take their conjunction $\;_j\, g_j$ denoted by *ground*(C). We take the ground as our legal goal, and send it to the following GDA process as its input data.

2.2 Appropriate similarity

After receiving the ground G, we apply our GDA-algorithm to find a sort mapping $\varphi{:}Sort_{concrete} \rightarrow Sort_{abstract}$ such that every clause used in the proof of G is abstracted and preserved at the abstract level. To make the notion clearer and to save the space, we first illustrate it with a simple example and summarize our argument on GDA by our general definition.

Suppose we have a fact set:

{**bigger(_:car,_:human)**, **bigger(_:tank,_:human)**,
bigger(_:mountain,_:human)}

and a sort mapping: φ_1: {**car**,**tank**,**mountain**}→ α. If we conceive that the φ is a similarity we desire, the dissimilarities of **car**, **tank** and **mountain** are disregarded, and the fact set will be transformed into a single fact **bigger(_:α,_:human)**. This might be true when we do not postulate some particular ground of rule.

Suppose further that we are talking about a legal rule with which some relations between human and artificial things are concerned. Then we would say that **mountain** is not relevant to the rule and that **mountain** and **car** are never similar w.r.t. the ground of rule. Our GDA simulates this kind of argument as follows:

1. Since **car** and **mountain** is not similar w.r.t. the ground, our domain theory would contain a clause, **designed_by_human(_:car)**, for instance, showing the dissimilarity w.r.t the ground.
2. Then a clause **designed_by_human(_:α)** is never an abstract clause, for it depends on our way to instantiate it.
   In fact, we have **designed_by_human(_:car)** as an instance for a case of **car**, while we do not for a case of **mountain**.
3. Consequently we conclude that those two notions are never similar w.r.t. the ground.

As explained in the second point, we say that a clause is *abstract* if any instance clause obtained by taking an instance sort s $\varphi^{-1}(\alpha)$ arbitrarily holds in the concrete level knowledge, where $\varphi^{-1}(\alpha)=\{s \mid \varphi(s)=\alpha\}$.

It should be noted here that it is indeed necessary to consider only clauses that appear in the proof of ground. If otherwise, even **car** and **tank** in this case will be judged as dissimilar ones, for they have distinct properties in another aspect. It is now clear that GDA computes a similarity represented by a sort mapping, depending on a ground of legal rule under our domain theory. We summarize our argument by the following definitions:

[**Definition**] **1** (**SortAbs**) *Let* φ:*S* a *S' be a sort mapping and T be a set of clauses. SortAbs*$_\varphi$(*T*) *is the abstract clause set of T under* φ *defined as follows:*

$$SortAbs_\varphi(T) = \{C' \mid \ F \quad \varphi^{-1}(C') \ \ T \mid\!- F\} \quad \text{n}$$

[**Definition**] **2** (φ $_C$) We say that φ is **an appropriate similarity w.r.t. the grounds of a rule** *C*, if the following two conditions are satisfied:

- **Substitutability Condition**

  $\varphi(Proof_{ground(C)}) \quad SortAbs_\varphi(T)$, where $Proof_{ground(C)}$ denotes the set of clauses that are used to derive *ground*(*C*).
- **Similarity Inheritance Condition**

  $\varphi(s_1) = \varphi(s)$ whenever $s_1 \leq s_2$ and $\varphi(s_2) = \varphi(s)$. n

2.3 Recovering unification failure

As mentioned previously, we suppose a situation in which some legal rule is not applicable to a case represented by a set of facts. Since we describe every rule and fact in terms of order-sorted signature, such an application failure would be a type error caused by matching a top goal of our reasoning and a conclusion part of legal rule. For instance, suppose we have the following rule and a goal:

prohibit_in(X:vehicle, Y:park) ←.
← **prohibit_in(Z:horse, Y:park)**. n

In this case, two notions **vehicle** and **horse** are mismatched. It is therefore necessary to obtain an appropriate sort mapping φ such that **vehicle** and **horse** are mapped to a single abstract sort. Our GDA algorithm is designed to search only mappings satisfying the condition. For more details, see (Haraguchi,1996; Kakuta *et al*.,1996).

3 Appropriateness of similarity with Value Restriction Preservingness

We present in this section a new definition of appropriateness of similarity. The appropriateness is defined by adding a new criterion, called *Value Restriction Preservingness* (VRP), to Definition 2 originally presented in (Kakuta *et al*., 1996). By introducing VRP, we can consider a similarity from the viewpoint of conceptual structure. In order to state the new criterion, we firstly introduce notions of *role* and *value restriction*.

3.1 Value restriction on role

A *role* is a binary relation between individuals in our domain. Let **r** be a binary relation that is intended to represent some role. We call **r** a *role predicate symbol* (or simply *role*). The second argument of **r** is called the *filler* of **r**. It is often referred as **r**-filler.

A *value restriction* on role is used to restrict the range of role filler to individuals of some sort. We assume in this paper that such a restriction is of the form

$$\mathbf{r} : \mathbf{s} \to \mathbf{s}',$$

where **r** is a role and **s** and **s**′ are sort symbols. The meaning of **r**:**s** → **s**′ is that **r**-filler of **s** should be filled with an individual of **s**′. In other words, **r**-filler of the sort concept denoted by **s** is always the concept denoted by **s**′. In this sense, we can consider that a value restriction defines a *structure of concept*.
From the semantics of sorts[1], we can observe inheritance of value restrictions, that is, inheritance of structures of concepts.

Proposition 1 (**Inheritance of Value Restriction**)
Let **r**:**s** → **s**′ *be a value restriction of* **s** *and* ” *be a sort symbol such that* ” ≤ **s**. *Then the sort* ” *inherits a value restriction* **r**: ” → **s**′ *from* **s**. n

Thus, each sort inherits a value restriction from its super sort. It should be noted that a sort may inherit several restrictions, since it has more than one super sort in general.

[1] Using an interpretation function *I*, a sort **s** is interpreted as a set of individuals *I*(**s**)   *D*, where *D* is a domain (the set of all individuals). If **s** ≤ **s**', then *I*(**s**)   *I*(**s**').

3.2 Value Restriction Preservingness

We present here a new criterion, called *Value Restriction Preservingness* (VRP, for short), for our appropriateness of similarity. Before giving it formally, it would be useful for the reader to provide a basic idea underlying it.

The following intuition is the basis of VRP:

> It would be natural to consider that for similar sort concepts, their structures are similar also.

In studies on analogical reasoning (Gentner, 1982; Winston, 1980), analogy is carried out based on such an intuition. We consider this intuition as a constraint on similarity. That is, for similar sort concepts, their structures should be similar. We state this constraint in terms of value restrictions, since a value restriction can be viewed as a definition of structure of concept. As an example, let us assume that we have two value restrictions

$$\mathbf{agent} : \mathbf{conversation} \rightarrow \mathbf{human} \quad \text{and}$$
$$\mathbf{agent} : \mathbf{network\_communication} \rightarrow \mathbf{computer} \,.$$

The constraint on similarity claims that if **conversation** and **network_communication** are considered similar, then **human** and **computer** have to be similar as well.

Let us consider a similarity φ such that

$$\varphi\,(\mathbf{conversation}) = \varphi\,(\mathbf{network\_communication}) = \alpha \quad \text{and}$$
$$\varphi\,(\mathbf{human}) = \varphi\,(\mathbf{computer}) = \beta.$$

It is obvious that the similarity satisfies the constraint. Under φ , the value restrictions are preserved as the same restriction $\mathbf{agent}{:}\alpha \rightarrow \beta$. Therefore we call the constraint on similarity Value Restriction Preservingness (VRP). We would like to consider that an appropriate similarity should satisfy VRP.

In the remainder of this subsection, we discuss VRP more precisely and present its formal definition.

In order to examine whether a similarity satisfies VRP, it is necessary to collect all value restrictions of each sort. More concretely, for each sort $\mathbf{s}$, we have to collect the value restrictions that are of the form $\mathbf{r{:}s} \rightarrow \mathbf{s}'$, where $\mathbf{r}$ and $\mathbf{s}'$ are arbitrary role and sort, respectively. However, it should be noted that some of such restrictions may not appear in our knowledge base *explicitly*, since as previously mentioned, some restrictions of **s** may be inherited from a super sort of **s**. Therefore, such a collection of all value restrictions is completed by collecting each value restriction of the form **r: ”** → **”**′ for any super sort **”** of **s** and transforming it into **r:s** → **”**′.

Let us assume that for sorts $\mathbf{s}_1$ and $\mathbf{s}_2$, sets of value restrictions VR($\mathbf{s}_1$) and VR($\mathbf{s}_2$) are collected, respectively. Moreover, suppose we try to examine whether a similarity φ under which $\varphi\,(\mathbf{s}_1)$=$\varphi\,(\mathbf{s}_2)$ holds satisfies VRP. In a simple case, for each restriction $\mathbf{r{:}s}_1 \rightarrow \mathbf{s}'_1$ in VR($\mathbf{s}_1$), if there exists a value restriction $\mathbf{r{:}s}_2 \rightarrow \mathbf{s}'_2$ in VR($\mathbf{s}_2$) such that $\varphi\,(\mathbf{s}'_1) = \varphi\,(\mathbf{s}'_2)$, then the similarity φ is considered to satisfy VRP. However, we often encounter a more complex case.

Let VR(**s**) be a set of collected value restrictions of **s** according to the above procedure. In general, VR(**s**) contains more than one restriction on a role **r**. That is, VR(**s**) might contain several restrictions $\mathbf{r:s} \rightarrow \mathbf{s}'_1, \ldots, \mathbf{r:s} \rightarrow \mathbf{s}'_\mathbf{n}$. We have to discuss how to examine whether a similarity satisfies VRP in such a case.

By the value restrictions $\mathbf{r:s} \rightarrow \mathbf{s}'_1, \ldots$ and $\mathbf{r:s} \rightarrow \mathbf{s}'_\mathbf{n}$, it is required that for each individual of **s**, the **r**-filler should be filled with an individual belonging to each of $\mathbf{s}'_1, \ldots$ and $\mathbf{s}'_\mathbf{n}$. From the semantical viewpoint, we can consider a sort $\mathbf{s}'$ whose extension is exactly $I(\mathbf{s}'_1)$ I ... I $I(\mathbf{s}'_\mathbf{n})$. Assuming such a sort $\mathbf{s}'$, we can consider a value restriction R = $\mathbf{r:s} \rightarrow \mathbf{s}'$ that is equivalent to the set of restrictions $\mathbf{r:s} \rightarrow \mathbf{s}'_1, \ldots, \mathbf{r:s} \rightarrow \mathbf{s}'_\mathbf{n}$. Based on R, we would be able to examine a similarity according to the same procedure in the above simple case.

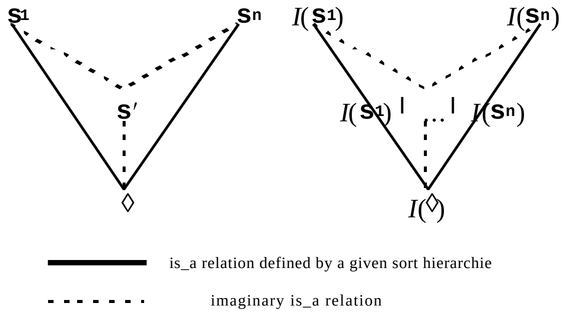

Figure 1:Relationship between ◊ and $\mathbf{s}'$

However, it should be noted that such a sort $\mathbf{s}'$ might be an *imaginary* one. That is, we might have *no* such a sort $\mathbf{s}'$ in our sort hierarchy.[2] If we have no $\mathbf{s}'$, $\mathbf{r:s} \rightarrow \mathbf{s}'$ cannot be used as an equivalent restriction, since we can explicitly handle no sort other than those appearing in our hierarchy. In such a case, we have to substitute some sort in our hierarchy for $\mathbf{s}'$. We propose therefore to substitute a *maximal lower bound* ◊ of $\mathbf{s}_1, \ldots, \mathbf{s}_\mathbf{n}$ for $\mathbf{s}'$ in our sort hierarchy.[3] Figure 1 would be useful for the reader in understanding the relationship between ◊ and $\mathbf{s}'$. Then, we use a value restriction R′ = $\mathbf{r:s} \rightarrow$ ◊ to examine a similarity instead of R. From the semantics of sorts, since I(◊) is a subset of $I(\mathbf{s}'_1)$ I ... I $I(\mathbf{s}'_\mathbf{n})$ (= $I(\mathbf{s}')$), such a substitution is *sound*. That is, satisfying R′ is a sufficient condition for satisfying R. We should note that it is not a necessary condition. In this sense, this substitution would be considered as one of *heuristics*.

For a formal definition of VRP, we have another important point that remains to be discussed. We discuss here relevance of value restrictions to a given goal (ground of rule). Since we consider in this paper an appropriateness of similarity depending on a given goal, conceptual structures (that is, value restrictions) taken into account should be *relevant to the given goal*.

In this paper, we take only roles that appear in a proof of a given goal into account. They are called *relevant roles to* ***G***. Then, the value restrictions on the roles are considered to be relevant to the goal as well. When we try to examine whether a similarity satisfies VRP, only relevant value restrictions are collected for each sort. More

[2] As a reason why we have no such a $\mathbf{s}'$, it would be considered that any sort whose extension is $I(\mathbf{s}'_1)$ I ... I $I(\mathbf{s}'_\mathbf{n})$ is out of our interest or not so important for us.

[3] Note that we might have several maximal lower bounds in general.

precisely, for a goal G to be proved, let $R(G)$ be the set of roles that appear in a proof of G. For each sort $\mathbf{s}$, all value restrictions of the form $\mathbf{r:s} \rightarrow \mathbf{s}'$ are collected according to the procedure explained above, where $\mathbf{r}$ $R(G)$and $\mathbf{s}'$ is an arbitrary sort.

Based on the above discussions, we can formally present VRP as follows:

[**Definition**] **3** (**Value Restriction Preservingness**)
Let G be a goal, (*S*,≤) *be a sort hierarchy,* VR *be a set of value restrictions and* $R(G)$ *be a set of relevant roles to G.*

A similarity φ *is said to satisfy Value Restriction Preservingness (VRP) iff for any sorts* $\mathbf{s}_1,\mathbf{s}_2$ *S*,

if φ ($\mathbf{s}_1$) = φ ($\mathbf{s}_2$), then for each **r** $R(G)$,

1. $\mathbf{s}'_1$ $mlb_{(S,\leq)}(filler(\mathbf{r},\mathbf{s}_1))$, $\mathbf{s}'_2$ $mlb_{(S,\leq)}(filler(\mathbf{r},\mathbf{s}_2))$
   such that φ ($\mathbf{s}'_1$) = φ ($\mathbf{s}'_2$) *or*
2. $filler(\mathbf{r},\mathbf{s}_1) = filler(\mathbf{r},\mathbf{s}_2) = \phi$,

where $filler(\mathbf{r},\mathbf{s}_i) = \{\mathbf{s}' |$ $\mathbf{s}_i \leq$ ” and **r: ”** → **s**′ VR } and
$mlb_{(S,\leq)}(E)=\{\mathbf{s}'$ S | **s**′ *is a maximal lower bound of E under* ≤}. n

Now we can define a new appropriateness of similarity by adapting VRP as an additional postulate.

[**Definition**] **4** (**Appropriateness of Similarity with VRP**)
If a similarity φ *satisfies the following conditions, then* φ *is said to be appropriate.*

ü Substitutability Condition
ü Similarity Inheritance Condition
ü Value Restriction Preservingness n

Since VRP is newly introduced, the number of appropriate similarities is reduced as compared with previous one. An example of such reduction is shown in the next section. Although a system for finding appropriate similarities based on the new definition has currently been under implementation, we expect that based on VRP, we would be able to obtain a drastic reduction of search space of similarities.

4 Finding appropriate similarity for ground of rule

This section illustrates a process of finding appropriate similarities for ground of rules. A knowledge base shown in Figure 2 is used throughout the illustration.

Let us consider the following ground (goal) *G* and the set of clauses *Proof*(*G*) that are used to derive *G* from the knowledge base:

```
G=danger(car0),
Proof(G)={
          allow_in(car0,this_park).,
          locomotive_parts(car0,tire0).,
          good_repair(tire0).,
          bigger(_:car,_:human).,
          movable(X:tire)←good_repair(X).,
          danger(X:object)←
             allow_in(X,_:park),locomotive_parts(X,Y),
             movable(Y),bigger(X,_:human). },
```

where **car0** and **tire0** are constants such that **car0** **car** and **tire0** **tire**.

For each of the following similarities[4], let us examine its appropriateness:

$\varphi_1 = \{\{\mathbf{car},\mathbf{tank},\mathbf{truck}\},\{\mathbf{tire},\mathbf{caterpillar\_track}\}\}$
$\varphi_2 = \{\{\mathbf{car},\mathbf{tank},\mathbf{tire}\}\}$
$\varphi_3 = \{\{\mathbf{car},\mathbf{tank}\}\}$
$\varphi_4 = \{\{\mathbf{car},\mathbf{tank},\mathbf{truck},\mathbf{horse}\},\{\mathbf{tire},\mathbf{caterpillar\_track}\}\}$

Under φ_1, a clause *C*=**bigger(_:car,_:human)** in *Proof*(*G*) is mapped into *C'*= **bigger(_:α,_:human)**, where α is the image (hypothetical abstract sort) of **car**, **tank** and **truck** under φ_1. The instances of *C'* under φ_1 are **bigger(_:car,_:human)**, **bigger(_:tank,_:human)** and **bigger(_:truck,_:human)**. Since each of them is provable from the domain theory, the clause *C'* can be contained in $SortAbs_{\varphi_1}(T)$. Similarly, we can verify that each clause of $\varphi_1(Proof(G))$ can be contained in $SortAbs_{\varphi_1}(T)$. That is, φ_1 satisfies Substitutability Condition.

On the other hand, φ_2 does not satisfy the condition, that is, the similarity φ_2 is not appropriate for *G*. Assuming β is the image of **car**, **tank** and **tire** under φ_2, although *C* is mapped into **bigger(_:β,_:human)**, one of its instantiations **bigger(_:tire, _:human)** is not provable from the theory. The reader would easily be able to verify that the similarities φ_3 and φ_4 satisfy Substitutability Condition.

Sort Hierarchy:

```
tank≤object. car≤object. truck≤car. animal≤object.
tire≤object. leg≤object. park≤place. human≤animal.
horse≤animal.  caterpillar_track≤object.
tank1∈tank.  my_car∈car.  tire1∈tire.
ct1∈caterpillar_track.  this_park∈park.
```

Value Restriction:

```
locomotive_parts:car→tire. locomotive_parts:animal→leg.
locomotive_parts:tank→caterpillar_track.
locomotive_parts:object→object.
```

Domain Knowledge:

```
danger(X:object)←
        allow_in(X,_:park),locomotive_parts(X,Y),
           movable(Y),bigger(X,_:human).
locomotive_parts(my_car,tires1).
locomotive_parts(tank1,ct1).
good_repair(tires1). good_repair(ct1).
bigger(_:car,_:human). bigger(_:horse,_:human).
bigger(_:tank,_:human).
movable(X:tire)←good_repair(X).
movable(X:leg)←not_injured(X).
movable(X:caterpillar_track)←good_repair(X).
legal_rule:: prohibit_in(X:car,this_park)←
        locomotive_parts(X,Y:object),good_repair(Y).
```

[4] Each similarity (sort mapping) is represented by a partition of the set of sorts. Each element of partition is called a *cell* and denotes a similar class of sorts.

Negation Pairs: [(prohibit_in,allow_in)]
CF-Prediates: [danger/1]
Original GOAL: prohibit_in(X:tank,this_park).

Figure 2: Example of knowledge base

Then, let us examine whether φ_3 satisfies Similarity Inheritance Condition (SIC). Although **truck** is a subsort of **car** and **car** is similar to **tank**, **truck** is not similar to **tank** under the similarity. Therefore the similarity φ_3 does not satisfy SIC.

On the other hand, we can verify that φ_1 and φ_4 satisfy SIC. According to our previous definition of similarity, therefore, both φ_1 and φ_4 are considered appropriate.

Then, let us try to examine each of the similarities satisfies VRP or not. In order to do so, we firstly have to obtain the relevant roles to G. In this case, since only **locomotive_parts** appears in $Proof(G)$, it is the relevant role.

Let us examine the similarity φ_4. We have the following value restrictions on **locomotive_parts** for**car**, **tank**, **truck** and **horse**, respectively:

locomotive_parts: car → tire.
locomotive_parts: tank → caterpillar_track.
locomotive_parts: truck → tire.
locomotive_parts: horse → leg.

It should be noted that each of the restrictions of **truck** and **horse** is inherited from **car** and **animal** respectively. By VRP, it is required that **tire**, **caterpillar_track** and **leg** should be similar. However, such a similarity is not implied by φ_4. Therefore, φ_4 does not satisfy VRP, that is, φ_4 is considered inappropriate according to our new definition. On the other hand, we can verify that φ_1 satisfies VRP, that is, φ_1 is appropriate for G.

Table 1 shows a comparison between the numbers of appropriate similarities obtained according to the new definition and the previous one, respectively. The example used in this examination is an extended example in Figure 2. It consists of 22 clauses and 17 sorts. From the table, it is shown that the number of appropriate similarities is drastically reduced by VRP.

	New Appropriateness	Previous Appropriateness
Number of possible partitions	82,864,869,804 (about 8.0×10^{10})	
Number of appropriate partitions	1	364
Combinations for original analogy	{car,truck,tank}	{car,truck,tank} {car,truck,tank,horse}

Table 1:Comparison between appropriatenesses

5 Concluding remarks

We have discussed the problem of introducing value restrictions on roles into our GDA framework for finding similarities depending on a legal goal (ground of rule). Although the number of possible similarities (sort mappings) is drastically reduced by the structural constraint VRP, we have not yet tested for more complex legal knowledge. We are now examining two real legal cases in which some statutes have been applied analogically. One is from Japanese Civil Code, and the other is a case for Convention on Contract for the International Sales of Goods. The latter one is a legal field which a Japanese project chaired by Prof. H.Yoshino at Meiji-Gakuin University is now analyzing and implementing.

The second point we have to cope with is the problem of Incompleteness of Domain Knowledge. Our GDA approach assumes that we can extract some clauses showing similarities or dissimilarities, when we judge if some concepts are similar or not. The problem is generally very hard, since our legal knowledge is contextual in our open world. Instead of attacking this hard problem directly, we are now developing a knowledge revision procedure invoked when some unintended similar is observed by our method (Okubo *et al*., 1996). The procedure tries to revise our knowledge base so that GDA never produces the unintended similarity after the revision.

References

Ashley, K.D. (1990). *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*, MIT Press.

Brachman, R.J., A. Borgida and D.L. McGuinness (1991). Living with CLASSIC. in: J. Sowa (ed.) *Principles of Semantic Networks*, pp. 401-456. Morgan Kaufmann.

Gentner, D. (1982). Are Scientific Analogies Metaphors. *Metaphor: Problems and Perspectives*, pp. 106-132, The Harvester Press.

Hart, H.L.A. (1958). Positivism and the Separation of Law and Morals. *Harvard Law Review 71*, pp. 593-629.

Kakuta, T., M. Haraguchi,M. and Y. Okubo (1996). A Goal-Dependent Abstraction for Legal Reasoning by Analogy. submitted to Int'l J., Feb. 1996. (The extended abstract in Japanese is found in *Proc. 10th Annual Conf. of JSAI*, pp. 91-94, June. 1996).

Haraguchi, M. (1995). A Reasoning System for Legal Analogy. *Machine Intelligence,* vol. 14, pp. 323-346.

Nebel, B. (1990). *Reasoning and Revision in Hybrid Representation Systems*. Springer LNAI 422.

Ohtake, Y., K. Nitta, S. Maeda, M. Ono, H. Osaki. and K. Sakane (1994). Legal reasoning system HELIC-II. *Trans. Inf. Process. Soc. Japan.* (in Japanese), vol. 35, no.6, pp. 986-96.

Okubo, Y. and M. Haraguchi (1994). Constructing Predicate Mappings for Goal-Dependent Abstraction. In: *Algorithmic Learning Theory '94*, Springer LNAI 872, pp. 516-531.

Okubo, Y. and M. Haraguchi (1996). Towards Knowledge Acquisition with Goal-Dependent Abstraction. *Proc. 10th Annual Conf. of JSAI* (in Japanese), pp. 289-292.

Tenenberg, J.D. (1989). Abstracting First-Order Theories. *Change of Representation and Inductive Bias*, pp. 67-79, Kluwer Academic Publishers.

Winston, P.H. (1980). Learning and Reasoning by Analogy. *CACM*, vol.2, no.3, pp. 689-703.

Yamaguti, K. (1994). Constructing a Legal Ontology from a General Ontology. *Proc. Int'l Workshop on Application of Logic Programming to Legal Reasoning*, pp. 73-80.