Towards a Legal Analogical Reasoning System: Knowledge Representation and Reasoning Methods

Hajime Yoshino
Faculty of Law, The Meiji Gakuin University
Shirakanedai, Minato-ku, Tokyo 108, JAPAN
Makoto Haraguchi, Seiichiro Sakurai
Department of Systems Science, Tokyo Institute of Technology
4259 Nagatsuta-cho, Midoriku, Yokohama 227
Sigeru Kagayama
Faculty of Law, Osaka University
Kaneyama-cho, Toyonaka 560, JAPAN

Abstract

Analogy has many important functions in the domain of law. Since the number of legal rules is restricted and their content is often incomplete, it is necessary at times for a lawyer to opt for an analogical application of a legal rule to a given case in order to decide the case properly. He may apply the rule, though it may not have originally been deemed related to such an event, on the basis of some similarity between the event of the case and the requirement of the relevant legal rule. This type of reasoning is called legal analogy. This paper analyzes an actual case of legal analogy in the field of Japanese civil law in order to clarify the reasoning methods used in analogy, as well as knowledge to justify the analogy. Finally it will be shown how the knowledge is utilized in a symbolic reasoning system both in terms of inverse and standard resolution.

1 Logical Structure of Legal Analogy

Analogy has many important functions in the domain of law. Since the number of legal rules is restricted and their content is often incomplete, it is necessary at times for a lawyer to opt for an analogical application of a legal rule to a given case in order to decide the case properly. He may apply the rule, though it may not have originally been deemed related to such an event, on the basis of some similarity between the event of the case and the requirement of the relevant legal rule. This type of reasoning is called legal analogy.

$$\frac{A \rightarrow X \qquad A \sim_{\alpha} B}{B \rightarrow X}$$

Figure 1: analogy schema

Legal Analogy is conceived as a process of generation of a hypothetical rule, which supplies the lack of law for a certain particular case. In other words legal Analogy is said to be an act of replacing a requirement A of statue rule $A \rightarrow X$ by other requirement B to generate a hypothetical rule for a give case, provided these two requirements are similar with respect to some important legal aspects. This logical structure can be shown following Fig. 1 .

Let us examine Fig. 1 for a while. Given a current case subsumed by B, suppose that a lawyer has in his mind a target goal, which is an intended legal conclusion never derived from the case under the present domain theory. In other words, the domain theory is too weak to get the desired conclusion. Then the problem is to make the hypothesis $B \rightarrow X$ in Fig. 1 in order to obtain X for the present case. The logic used to justify such an analogy can be stated as follows [9]: The legal rule $B \rightarrow X$ is considered valid because $\alpha \rightarrow X$ is valid and because $A \rightarrow \alpha$. Moreover, since $B \rightarrow \alpha$, we can conclude that the same effect X is derived from B under the existence of common α. It should be noted here that the intermediate hypothesis $\alpha \rightarrow X$ represents the heart of the law in a sense. The situation is more precisely illustrated by the diagram shown in Fig. 2 .

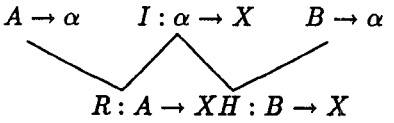


Figure 2: analogy diagram

Fig. 2 consists of two sub-diagrams with the same shape V. The left V is to obtain an intermediate hypothesis $I : \alpha \rightarrow X$ by a generalization. The other V means a deduction to get the final hypothesis H, which is the result of analogical interpretation.

As we have just observed, the primitive operators to realize analogical interpretation of legal rules are generalization and deduction. Needless to say, deductive engines are necessary for most legal reasoning systems. For instance, LES [7] performs deductive reasoning under a set of legal rules encoded as Horn clauses. The system is therefore implemented in resolution-based deductive engine, especially in Prolog. On the other hand, generalization has been studied in the field of machine learning, and various computational operators for generalization have been proposed by many researchers. Since the reasoning system we expect should be basically deductive to execute various types of knowledge, we choose an absorption operator [4] to realize our generalization. This is because the absorption is a kind of inverse resolution and is therefore easily implemented in a resolution-based deductive engine. A more detailed description on absorption, deduction, and analogy can be found in [2] which defines the analogical reasoning we use here in terms of absorption and deduction.

Now we are ready to discuss a method of representing legal rules and other related knowledge. In Section 2 and 3, we first present the knowledge representation language, and try to represent Article 94 of Japanese civil code for which analogy was applied.

2 Knowledge representation

LES is a logical system to perform deductive reasoning under a set of Horn clauses. This means that both legal rules and other rules to interpret requirements of rules are encoded to Horn clauses. Each predicate, called a compound predicate, has the following form:

$$< predicate\_name > (\ < \ relation\_identifier >, \\ < \ list\ of\ slots >),$$

The *relation_identifier* is a name of the relation and is used as a reference to the relation. Hence it has a function similar to a pointer in a frame based system. *slot* is a pair of slot name and its value, and is written as *slot_name* : *value*.

Article 94(Fictitious declaration of intention):

Clause 1 A fictitious declaration of intention made in collusion with the other party is null and void.

Clause 2 The nullity of declaration of intention as mentioned in the preceding clause cannot be set up against a good faith (bona fide) third person.

Figure 3: Article 94 of Japanese Civil Code

In this paper, each compound predicate is denoted by a word *p_id*, predicate name *p* followed by a constant word "id". Moreover, for each predicate *p_id*, we assume that *p* has the unique definition clause:

$$p(Slot\_list) \leftarrow p\_id(Identifier, Slot\_list).$$

Such predicate *p* is useful for readability of program clauses. Each rule is described as:

$$X \leftarrow A_1, ..., A_n$$

where A_j is an atomic formula of our language and X should be an instance of compound predicate.

For example, suppose we have the following rules as well as facts:

$p\_id(id1, [agt : a, obj : c])$.
$q\_id(id2, [[agt : b, obj : id1])$.
$p\_id(P, [agt : Agt1, obj : \_]), q([agt : Agt2, obj : P])$
$\quad \rightarrow r^*(([agt : Agt2, obj : Agt1]]))$,

where
"_" denotes an anonymous variable and, $pred^*(Slots)$ is an abbreviation of $pred\_id(r(Slots), Slots)$ for each non-compound predicate *pred*. The meaning of each clause is:

1. A relation instance $p([agt : a, obj : c])$ holds.
2. There is a relation instance of q whose agent and object cases are b and $p([agt : a, obj : c])$, respectively.
3. If we have a relation instances of q whose agent and object are $Agt2$ and a relation instance $(p([agt : Agt, ...])$ of p, respectively, then we also have a relation instance of r whose agent and object are $Agt2$ and $Agt1$, respectively.

As a result, we have $r([agt : b, obj : a])$ whose identifier is $r([agt : b, obj : a])$ itself.

Now we are ready to encode legal norm sentences in the form of compound predicates. Clause 2, Article 94 of Japanese Civil Code, which is one of the most frequently applied rules in legal analogy, is shown in Figure 3.

The Clause 1 means that the declaration is invalid if the parties negotiated that the declared intention was different from the real intention. The Clause 2 means that the invalidity cannot be set up against a good faith, who do not know the fact that the declaration is different from the real intention. The purpose of the Clause 2 is to protect the right of a good faith person who believed the declaration of intention.

The Clause2, for instance, can be encoded in the form of compound predicates as follows:

Rule 1 (Article94, Clause2)

contract_id(Ctrct1,[parties:[X,Y], obj:Obj, time:_]),
falsity_id(Falsity,[obj:Ctrct1]),
contract_id(Ctrct2,[parties:[Y,Z]], obj:Obj, time:_),
good_faith([agt:Z, obj:Falsity])]
→ cannot_set_up([agt: X, goa:Z, obj:Ctrct2])*

The rule was not directly encoded from legal sentence but compiled with legal knowledge. In other words, the definite clauses corresponds to a rule interpreted by a lawyer.

In Rule 1, symbols, which are preceded by a capital letter, denotes logical variables. The predicate *contract* denotes some contract. The predicate *falsity* means that the declaration of intention is different from the real intention in its object's value, $Cntrct1$. The predicate *cannot_set_up* means that X cannot claim the invalidity of the contract, $Cntrct2$, because of the invalidity of the contract, $Cntrct1$. Note that this rule is applied in spite of the legal invalidity of its requirement part.

The predicate *good_faith* is defined as follows:

$$good\_faith^*([agt : Agt,\ obj : Content) \leftarrow not(know([agt : Agt,\ obj : Content])$$

where *not* is the standard Prolog's *not* predicate, and the predicate *know* is defined so that $know(Agt, Content)$ succeeds iff the relation instance $Content$ is proved under the set of facts Agt knows. We use an auxiliary predicate $know\_fact$, where $know\_fact(Agt, Id)$ means that Agt knows a relation indexed by an identifier Id. All the possible instances of $know\_fact$ are initially presented in our fact database, as shown in Fact 1. According to such a representation of facts, the predicate *know* is easily realized as a kind of meta-interpreter:

$know(Agt, (Goal, Goals)) : -!,$
$know(Agt, Goal), know(Agt, Goals).$
$know(Agt, Goal) : -clause(Goal, Goals), !,$
$(Goals = true- > know\_fact\_rel(Agt, Goal)$
$; know(Agt, Goals)).$
$know\_fact\_rel(Agt, < comp\_pred > (Id, X)) : -$
$know\_fact(Agt, Id), < comp\_pred > (Id, X),$

Case of a petition against registration of passage of a house's title
((O)No.107-1951,judgment of the second pretty bench, Aug. 20th 1951))

Case: After "A" bought a house, which "O"owned, from "O", he approved the registration of passage of title from "O" to "B" without his real intention of the passage. Registered the passage of title, "B" sold the house to a good faith, "C", who did not know the real intention of "A" and only know that "B" registered. "C" registered the passage of ownership title. "A" claimed that "C" must do cancellation procedure of passage of title and others because the ownership of the house in the case should belong to "A", and "B" and "C" did not't have the ownership of the house.

Judgment: By analogical application of Clause2, Art. 94 of Japanese Civil Code, the nullity (falsity) of the registration of 'B" cannot be set up against the good faith "C", who did not know the real intention of "A", so that A cannot claim that "C" must do cancellation procedure of passage of title and others of the registration

Figure 4: Example of analogical application

where $< comp\_pred >$ is a meta variable ranging over every compound predicate.

In legal reasoning, it is important to represent facts which constitute the case. Before describing the representation of facts, a leading case of analogical application of Clause 2, Article 94 is shown in Figure 4. In the case, since "B" sold the house to "C" without its real ownership, "A", who was the real owner, claimed his ownership right. However, the judge approved that "C" had the real ownership right by legal analogy.

Legal cases are formalized by a set of approved facts, and an approved fact is also represented by a compound predicate. An example set is shown in Facts 1 . In Fact 1, the p_a, p_b, p_c and p_o represent the agents of the case. The first argument of each compound predicate is an identifier for reference and the second argument is a list of pairs of case and value. The name, "agt:", represents an agent case and "obj:" represents an object case. The predicate reg_of_ptitle means "registration of passage of title" and "reg" means "registration".

Other knowledge is needed to realize legal reasoning, and it is also represented by rules.

Facts 1 (Facts of the case in Figure 4)

sale_of_immovables_id(id1,[parties:[p_o,p_a],
obj:imm_X, time:t0]).
ownership_id(id8, [agt:p_a, obj:imm_X, time:t0]).

ownership_id(id2, [agt:p_a, obj:imm_X, time:t1]).
reg_of_ptitle_id(id3,[parties:[p_o,p_b],
obj:imm_X, time:t1]).
reg_id(id4,[agt:p_b, obj:imm_X, time:t1]).
approval_id(id5,[agt:p_a, obj:id3, time:t1]).
sale_of_immovables_id(id6,[parties:[p_b,p_c],
obj:imm_X, time:t2]).
reg_of_ptitle_id(id7,[parties:[p_b,p_c],
obj:imm_X, time:t2]).
know_fact([agt:p_c,obj:id3]).
know_fact([agt:p_c,obj:id4]).
know_fact([agt:p_c,obj:id6]).
know_fact([agt:p_c,obj:id7]).

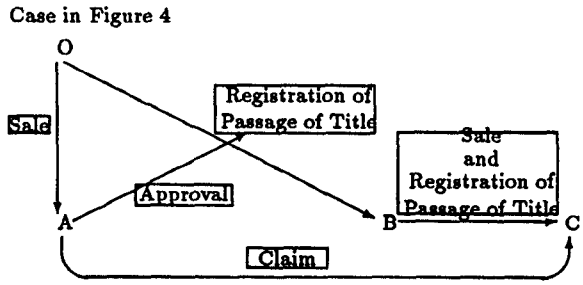
3 Analysis of analogical application and knowledge needed to realize legal analogy

Let us consider the case in Figure 4, again. Obviously, if the good faith, "C", did not appear, the ownership of "A" would be approved. Then, what inference is done in this case? What knowledge is needed to realize such legal analogy?

To clarify the reason why the rule can be applied to the case in Figure 4, let us consider the difference between Clause 2, Article 94 and the case. Their relation is shown in Figure 5. If O in Figure 5 is omitted, both of them can be seen as a trinary relation. If so, one-to-one partial correspondence, $\{A\text{-}X, B\text{-}Y, C\text{-}Z, ...\}$, can be found. However, although the rule has a relation between X and Y as a legal requirement, the case has no relation between A and B. What relation correspond to the contract between X and Y or the falsity of the contract? In such a case, a judge seems to transform the case with his common-sense or legal common-sense knowledge. Therefore, legal analogy requires some knowledge in order to transform the case description so that a correspondence between a case and a rule can be found easily.

The transformed case is shown in Figure 6. In Figure 6, two virtual relations of passage of title are introduced. In other words, we can assume that the passage of title from O to A and the passage of title from A to B, although they were not registered formally.

If the case is transformed as shown in Figure 6, can we apply Clause 2, Article 94? Obviously, a relation, which corresponds to the falsity in Clause 2, Article 94, is missing. In legal reasoning, a missing relation disables the application of a rule, such a relation must be fulfilled by some knowledge before the application. We can assume theories of interpretation as such a knowledge. A rule of a theory of interpretation is also written as a compound predicate rule. Figure 7 shows an example of such a rule. The rule is paraphrased as:



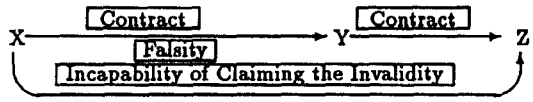


Figure 5: Relation between the case in Figure 4 and Clause 2, Article 94

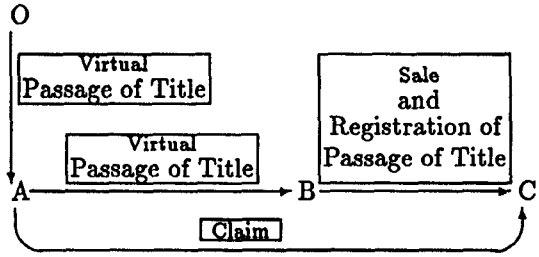


Figure 6: Transformed Case in Figure 4

> If a content of the representation of a contract is different from that of the state of affair of the contract, then it is concluded that the contract has falsity.

In Figure 7, to represent the variable content, a slot variable, *Atr*, is used. The predicate, *repr_of_ctrct*, represents the "representation of contract" and the predicate *soa_of_ctrct*, represents "state of affair of contract".

If the rule in Figure 7 is assumed, a relation may be introduced. However, to introduce the relation, falsity, more knowledge must be assumed since the rule is not about the registration but about the contract. In order to fill the gap between the registration and the con-

$$contract\_id(Ctrct, \_)$$
$$repr\_of\_ctrct([obj : Ctrct, Atr : ReprCts]),$$
$$soa\_of\_ctrct([obj : Ctrct, Atr : RealCts])),$$
$$ReprCts \neq RealCts$$
$$\rightarrow falsity^*([obj : Ctrct])$$

Figure 7: A Rule of A Theory of Interpretation(False Declaration)

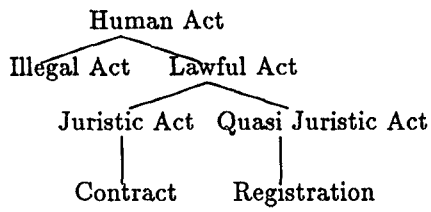


Figure 8: Example of Classification Knowledge

Textbookish knowledge

$contract\_id(Id, Cts) \rightarrow juristic\_act\_id(Id, Cts)$.
$juristic\_act\_id(Id, Cts) \rightarrow lawful\_act\_id(Id, Cts)$.
$reg\_of\_ptitle\_id(Id, Cts) \rightarrow registration\_id(Id, Cts)$.
$registration\_id(Id, Cts) \rightarrow quasi\_juristic\_act\_id(Id, Cts)$.
$quasi\_juristic\_act\_id(Id, Cts) \rightarrow lawful\_act\_id(Id, Cts)$.

Commonsense Knowledge

$reg\_of\_ptitle\_id(Regp, [parties : [\_, Agt], obj : Obj, time : \_])$
$\rightarrow repr\_of\_reg^*([obj : Regp, agt : Agt])$.
$reg\_of\_ptitle\_id(Regp, [parties : \_, obj : Obj, time : Time])$,
$ownership([agt : Agt, obj : Obj, time : Time])$,
$\rightarrow soa\_of\_reg^*([obj : Regp, agt : Agt])$.

$repr\_of\_ctrct\_id(Id, Cnts) \rightarrow repr\_id(Id, Cnts)$.
$soa\_of\_ctrct\_id(Id, Cnts) \rightarrow soa\_id(Id, Cnts)$.
$repr\_of\_reg|inst(Id, Cnts) \rightarrow repr\_id(Id, Cnts)$.
$soa\_of\_reg\_id(Id, Cnts) \rightarrow soa\_isnt(Id, Cnts)$.

Figure 9: Description of Classification Knowledge

tract, classification knowledge may be needed as such a knowledge. Figure 8 shows an example of classification knowledge.

The knowledge in Figure 8 is a textbookish knowledge. In order to fill the gap, another type of classification knowledge is needed. Figure 9 shows an example of such a knowledge.

The classification knowledge plays another important role in legal analogy. Even if virtual relations are correctly introduced with theories of interpretation, the number of possible paring between the case and the rule is very huge. To reduce the number and obtain an adequate pairing, the classification knowledge can be used. For example, we can obtain the pairing between the contract and the passage of title, since the upper class of the both concepts is the lawful act.

4 Utilization of legal knowledge by generalization and deduction

Now we are ready to show how to utilize legal knowledge and how to carry out legal reasoning in a symbolic reasoning system.

First suppose that we try to give a legal explanation to protect the right of good faith person p_c appeared in the case presented in Section 3, 4. Since the plaintiff p_a claimed that p_c should do cancellation procedure of passage of title, it suffices to show that he cannot set up p_c with respect to the passage of title. Hence we first make the following goal to be explained:

$$cannot\_set\_up([agt : p_a, goa : p_c, obj : id7]) \quad (1)$$

where $id7$ is the identifier of the registration of passage of title of C.

Since our system is basically an extension of Prolog interpreter with some additional reasoning functions, it tries to resolve the goal with some rules as well as facts. Resolving the goal [1]. with Rule ?? of Clause2, Article 94, we have the following goal list as the next goals:

$$contract\_id(Ctrct1, [parties : [p_a, Y], obj : Obj, time : T1]), \quad (2)$$
$$falsity\_id(Falsity, [obj : Ctrct1]), \quad (3)$$
$$contract\_id(id7, [parties : [Y, p_c], obj : Obj, time : T2]), \quad (4)$$
$$good\_faith([agt : p_c, obj : Falsity]). \quad (5)$$

The first goal (2) in the goal list cannot be resolved with any rules. The standard Prolog interpreter goes back to the previous goal (1) and tries to resolve it with some another rule. On the contrary, our system tries to generalize such a failure goal, provided there is no other goal in a goal list that can be resolved with some rules.

Control Strategy 1 ([3]) *Let A be a selected goal [2] in a goal list $A_1, ..., A_n$, and suppose we fails in resolving it with some rules. If there is no other goal A_j which can be resolved, then try to generalize A. Otherwise the generalization is delayed in order to suppress over generalization, and the standard resolution is tried for another goal in the goal list. In other words, our system tries generalization for some goals in a goal list only when every goal in the list fails.*

The goal (3) can be resolved with the legal theory rule in Figure 7. Thus our system prefers the resolution for

[1] Precisely speaking, this goal is firstly resolved with $cannot\_set\_up(Slots) \leftarrow cannot\_set\_up\_id(Id, Slots)$, and is converted to a goal $cannot\_set\_up\_id(Id, [agt : p_a, goa : p_c, obj : id7])$. Since, for each predicate *pred*, *pred* and *pred_id* denote the same relation at conceptual level, and since the process of converting the former predicate instance to the latter one is a trivial one step resolution, we identify them unless there is confusion.

[2] Normally the left most goal is selected.

(3) above generalization. Replacing (3) with the subgoal list:

$$contract\_id(Ctrct1, \_), \quad (6)$$
$$repr\_of\_ctrct([obj : Ctrct1, atr : ReprCts1]) \quad (7)$$
$$soa\_of\_ctrct([obj : Ctrct1, atr : RealCts1])) \quad (8)$$
$$RepreCts1 \neq RealCts1 \quad (9)$$

the goal list consequently becomes [(2), (6), (7), (8), (9), (4), (10), where

$$good\_faith([agt : p_c, obj : falsity([obj : Ctrct1]). \quad (10)$$

is an instantiated goal of (5).

Now any rule in the rule base is not applicable to any goal in the goal list [3]. We therefore try to generalize some goal. The generalization is carried out by "abductive goal reduction", which is a kind of inverse resolution. According to the standard resolution, a head of rule is unified with a goal. On the contrary, the inverse resolution tries to subsume body of clause to some goals in the goal list. Then, remove the subsumed goals from the goal list, and add the head of clause to the goal list. A formal description is given as follows:

Generalization by Abductive Goal Reduction[2]
let $G : A_1, A_2, ..., A_n$ be a current goal list. If there exists a rule $A \leftarrow B_1, ...B_k$ and a substitution θ such that $(B_1, ..., B_k)\theta \subseteq A_1, ..., A_n$, then remove $B_j\theta$ from $A_1, ..., A_n$ for each j, and add $A\theta$ to the goal.

Generally speaking, given a goal list, there exists several subsets of goals for which the generalization by abductive goal reduction can be applied.

Control Strategy 2 (Generalization Order)
Given a goal list such that every goal is a failure goal, choose an arbitrary subset of goal list, where the subset should be an instance of a body part of some rule in the rule base.

Suppose we choose a goal (2) to be generalized. Then by the rule

$$contract\_id(Id, Ct) \rightarrow juristic\_act\_id(Id, Ct)$$

the goal (2) is replaced with

$$juristic\_act\_id(Ctrct1, [parties : [p_a, Y], obj : Obj, time : T1]) \quad (11)$$

[3]We assume that the predicate *good_faith* is freezed. This means that it is evaluated only when all the arguments are instantiated to some ground terms.

Now our system tries to resolve the new goal list (11), (6), (7), (8), (9), (4), (10). The goal (11) can be resolved with the classification rule

$$juristic\_act\_id(Id, Ct) \leftarrow contract\_id(Id, Ct) \quad (12)$$

However the resolution with this rule results in the original goal (2). This causes a cyclic reasoning. In order to avoid such a generalization - specialization cycle, we need a test called a subsumption check: [4]

Control Strategy 3 (Subsumption Check [3])
Suppose that we have a current goal list $G : A_1, ..., A_n$ and that a next goal list G' is obtained by a generalization or a resolution. Whenever G' is subsumed by a previous goal list, cancel the reduction from G to G', and try another resolution or generalization according to Control Strategy 1.

Thus, the goal (11) and the rule (12) is never resolved by the subsumption check. Since there exists no juristic relation from p_a in our fact database, the goal (11) fails.

We have now goal list in which each goal is a failure goal. Repeating similar generalizations several times, we have a goal list:

$lawful\_act\_id(Ctrct1, [parties : [p_a, Y], obj : Obj, time : T1]),$
$lawful\_act\_id(Ctrct1, \_)),$
$repr([obj : Ctrct1, atr : ReprCts]),$
$soa\_of\_ctrct([obj : Ctrct1, atr : RealCts]),$
$ReprCts \neq RealCts,$
$contract\_id(id7, [parties : [Y, p_c], obj : Obj, time : T2]),$
$good\_faith([agt : p_c, obj : falsity([obj : Ctrct1])$

Now the goal $repr([obj : Ctrct1, atr : ReprCts])$, succeeds since $repr([obj : id3, atr : p\_b])$ can be provable from our facts and rules, where $id3$ referring the registration of passage of title from p_o to p_b. The next goal list is:

$lawful\_act\_id(id3, [parties : [p_a, Y], obj : Obj, time : T1]),$
$lawful\_act\_id(id3, \_)),$
$soa\_of\_ctrct([obj : id3, atr : RealCts]),$
$p\_b \neq RealCts,$
$contract\_id(id7, [parties : [Y, p_c]]),, obj : Obj, time : T2]),$
$good\_faith([agt : p_c, obj : falsity([obj : id3])$

Now $good\_faith$ can be evaluated and in fact succeeds. The goal $soa\_of\_ctrct([obj : id3, atr : RealCts])$ is once generalized to $soa([obj : id3, atr : RealCts])$ that results in the solution $RealCts = p\_a$ as well as $p\_b \neq p\_a$. Similarly $contract\_id(id7, ...)$ is generalized to $lawful\_act\_id(id7, ...)$ which is true under our database. Thus we have the final goal list:

$lawful\_act(id3, [parties : [p\_a, p\_b]]),$
$lawful\_act(id3, \_)),$

[4]Similarly "specialization - generalization" cycle can occur. Our subsumption check is applied to both types of cycles.

No generalization is possible for this goal list. In such a case, our system presents us the goal list as a final output.

This result is due to the fact that there is no direct legal relationship between p_a and p_b. As we have explained in the previous sections, judges seems to have assumed a virtual relation $lawful\_act(id*, [parties : [p\_a, p\_b]])$. In fact the followings seems to support the virtual relation:

Default: Normally a sale of immovable implies registration. $reg\_of\_ptitle([parties : [p\_o, p\_a], ...])$ is therefore assumable.

Assumption supporting facts: The recorded ownership p_b registered is also realized by assuming $reg\_of\_ptitle([parties : [p\_a, p\_b], ...])$.

Substantial reasoning: From the default and the assumption supporting facts in the real case, $reg\_of\_ptitle([parties : [p\_a, p\_b], ...])$ is substantially assumable.

However it still remains as a future work to develop knowledge as well as reasoning method to reason such virtual relations that support the final output of our present system.

5 Some comments on conflicts in generalizing rules

We have described various types of knowledge used for interpreting legal rules analogically and a fundamental mechanism to perform the interpretation. Both generalization and analogy give us possible ways of interpreting legal rules. As long as our generalization and analogy are based on a legal conceptual hierarchy with which most lawyers agree, a hypothesis produced by our system will be approved. However our knowledge base might contain several rules to generalize legal rules in different ways. Some lawyers agree with some generalizations with which another lawyers disagree. Thus there exist conflicts in generalizing rules. From a computational point of view, this problem of conflicts can be considered as the problem of nondeterministic choice of generalizations. Roughly speaking we have in our mind two ways to cope with this problem.

The first one is to introduce semantic constraints by which some inappropriate generalizations are rejected due to the inconsistency with the constraints. This is also a well known approach to check the appropriateness of generalizations, and can be implemented in an ATMS-like truth maintenance system in principle. However it is true that we cannot decide what constrains are needed before reasoning. So the truth maintenance system is used only for checking the appropriateness of hypotheses.

The second way to cope with the problem of nondeterministic choice of generalizations is now investigated by one of authors. The key idea is to find past cases for which the same or similar generalizations are used. In a word, this is a case-based justification of generalization and analogy. According to this approach, no new generalization is found by our system. However, the system can generate a case-based explanation showing the reason why it chooses a particular generalization to interpret legal rules. A forthcoming paper will make the idea more clear.

References

[1] H. Tanaka. *Introduction to the study of positive law (in Japanese)*, University of Tokyo Press, 1974.

[2] M. Haraguchi *A form of analogy as an abductive inference*, In *Proc. 2nd Workshop on Algorithmic Learning Theory*, pages 266-274, Japanese Society for Artificial Intelligence, 1991.

[3] M. Haraguchi *What kinds of knowledge and inferences are needed to realize legal reasoning?* (in Japanese), *Proc. 6th symposium on knowledge representation and legal reasoning system*, Legal Expert System Association in Japan, 1992.

[4] S. Muggleton. and W. Buntine. *Machine invention of first-order predicates by inverting resolution*, In *Proc. Workshop on Machine Learning*, pages 339–352, 1988.

[5] S. Muggleton. *Inductive logic programming*, in *Proc. 1st Workshop of Algorithmic Learning Theory*, pages 42–66, 1990.

[6] H. Yoshino, M. Haraguchi, S. Kagayama, Y. Matsumura. *Foundation of systematization of analogy in law* (in Japanese) , In *Proc. National conference of Japanese Association for Artificial Intelligence*, pages 219–222, 1991.

[7] H. Yoshino *Legal expert system LES-2*, in *Springer Lecture Notes in Computer Science, Logic Programming '86* pages 34-45, 1987.

[8] Z. Aoumi. *Introduction to Philosophy of Law* (in Japanese), Koubunn-dou, 1989.

[9] H. Gasyuu *Analogy in law* (in Japanese), unpublished lecture note, Legal Expert Systems Association, Meiji Gakuinn Univ., Tokyo, 1986.

[10] Rouveirol C.: *"ITOU: Induction of First Order Theories"*, in *Proceedings of the first Inductive Learning Programming Workshop*, Viana de Castelo, march 1991a