An Actual Application of an Analogical Legal Reasoning System Dependent on Legal Purposes

Tokuyasu Kakuta and Makoto Haraguchi

Faculty of Law, Hokkaido University, Sapporo,
Division of Electronics and Information Engineering,
kaku@juris.hokudai.ac.jp, makoto@db-ei.eng.hokudai.ac.jp

Abstract

We have already developed a legal reasoning system by analogy based on a framework, called **Goal-Dependent Abstraction** (**GDA**), to detect similarities dependent on given goals. According to the framework, two legal concepts are regarded as similar ones, provided they share the same explanation of a legal purpose as the goal to be explained. Consequently, we can automatically capture similarities between the premises of legal rules and cases in inquiry, depending on the purposes of legal rules. In this article, we further show that this approach works well in an actual case for which Japanese Supreme Court has analogically applied the Article 93 of the Civil Code of Japan, which prescribes mental reservations, to an act as an agent. Furthermore, to illustrate our reasoning procedure, we also demonstrate the overview and GUI of our legal reasoning system adopting an object-oriented logic programming language to describe legal knowledge.

1 Introduction

Legal rules are usually interpreted dependent on the purposes of the rules. When we interpret legal concepts in a rule according to static and standard terminology, the ranges of possible interpretations are so restricted that we may fail in drawing the right meaning of those concepts in a given case. In our previous studies, Kakuta et al. (1997) and Kakuta et al. (1996), we have developed a basic system to generate interpretations that can reflect the purposes of law. This article demonstrates our newest integrated legal reasoning system in which our previous framework is extended, and shows that our approach can handle an actual case. The case exemplified is a very famous one to deal with the problem of “abuse of the power of an agent” and has a lot of similar cases. Since the problem is unforeseen by the Civil Code of Japan, our Supreme Court has judged the case applying the proviso clause of the Article 93 analogically. This clause prescribes “mental reservations”, and cannot be applied to the case directly. Furthermore, this problem is also illustrated in most of standard textbooks for Japanese civil law. Thus, we can consider the case is placed in an important position to analyze the purpose and analogy in law based on it. By the same reason, the case can be a test example to evaluate reasoning systems concerning analogy in law. In section 4 we report briefly a solution obtained from our system that passes the evaluation.

We further intend to improve the computer operations by introducing a GUI so that lawyers can easily use our system. The GUI presents also the graphical visualizations of conceptual hierarchies and proof trees. On account of the facility of the operations, our system will be available for many lawyers, researchers and students. As a result, we expect that our system can contribute toward making more practical legal knowledge bases. In section 4, we demonstrate the actions of our system with our GUI.

As we have shown in our previous studies, our system is based on **Goal-Dependent Abstraction** (**GDA**, for short) presented by Okubo and Haraguchi (1994). In a GDA framework, two concepts are considered similar when the concepts share a proof. To illustrate the sharing of a proof, we suppose that only the following example is given:

```
p(X:c1)<-q(X).   p(X:c2)<-q(X).   p(X:c3)<-r(X).
q(X:c1).         q(X:c2).         r(X:c3).
---------------- ---------------- ----------------
p(X:c1)          p(X:c2)          p(X:c3)
```

This example shows, for instance, that p(X:c1) is drawn from p(X:c1)<-q(X) and q(X:c1), where the formula p(X:c1) denotes a concept c1 has a property p. Then, we can observe that the proofs of p(X:c1) and p(X:c2) have the same structure. In this case, we can state that c1 and c2 share the proof. According to GDA, c1 and c2 are regarded as similar concepts. However, c3 is not judged similar to them since the proof of p(X:c3) dose not have the same structure. When we consider a hypothetical concept α as a super (i.e. abstract) concept of c1 and c2, we can describe the following *generalized*[1] (i.e. abstract) proof:

```
p(X: α)<-q(X).
q(X: α).
----------------
p(X: α)
```

In these examples, p(X:c1) is regarded as a *goal* used for GDA. Clearly, the *abstraction*[2] depends on a given goal (to put it more precisely, the goal and its proof). Therefore, we call this framework "Goal-Dependent Abstraction". In this article, we show that GDA can be adopted toward analogical applications of legal rules in an actual problem.

In our approach, the purpose of a legal rule and its explanation are regarded as the goal and the proof for GDA respectively. As the result, we can obtain the similarity for using legal analogy dependent on the purpose of the rule and its explanation. Moreover, we can notice that this explanation is available for showing the ground of the analogy, providing that a concept that occurs in the explanation is replacing with a similar concept found by GDA.

Using the following simple example[3], we will illustrate a process of our analogical legal reasoning:

1 For further details of the generalization, see Haraguchi (1995).
2 For further details of abstractions, see Okubo and Haraguchi (1994), Tenenberg (1989) and Plaisted (1981).
3 Although this example is also used in Kedar-cabelli (1985), based on purpose directed analogy, no algorithm to deal with the example is presented in the article.

"vehicles should be prohibited to enter this park" (Hart, 1958).

We can consider several possible purposes of this rule and their explanations[4]. One of the purposes can be:

"a dangerous situation arises if vehicles enter this park, because vehicles are movable and big".

In this case, the explanation is not only valid for vehicles but, for instance, also for horses. According to the GDA framework, besides horses, we can find all of the other concepts (e.g. elephants, helicopters, etc.) that can make the same explanation valid[5]. Grouping all of the concepts, firstly we create a hypothetical super concept α of any concepts (e.g. vehicle, horse, elephant, etc.) that belong to the group. Secondly, we replace the rule by the *generalized* (i.e. *abstract*) rule

"α should be prohibited to enter this park"

hypothetically. The ground is that "a dangerous situation arises if an α enters this park, because α's are movable and big". Finally, since horses are also subsumed by α, we can conclude that horses should be prohibited to enter the park by the rule. In our system, by automatically generating such a hypothetical rule to interpret the original one, we can perform legal reasoning by analogy. In section 4, we show that this approach can work well not only for this kind of toy case but also for an actual case.

2 Knowledge representation

This section presents, as the preliminary to illustrate the actions of our system, a brief introduction of our knowledge representation formalism. To represent knowledge, a logic programming language (Lloyd (1984)) like *F-logic* presented by Kifer et al. (1993) is used in our system. It is also considered a kind of object-oriented language. From the viewpoint of the facility to describe formulae and grasp their meaning, several legal reasoning systems (e.g. Nitta et al. (1993); Yoshino et al. (1993)) adopt this kind of representation.

The language is based on **Order-Sorted Logic** (**OSL**, for short) (e.g. Walter (1988)). Our OSL-formulae are horn-clauses allowed to include *typed* (or *sorted*) constants and variables[6].

Firstly, we illustrate the descriptions based on OSL. For instance, an example of the clauses is shown as follows:

p(X:a)<-q(X,Y:b)[7].

In this clause, 'a' and 'b' are called **sort symbols** (**sorts**, for short). These symbols denote the types of the variables (i.e. 'X' and 'Y'). Furthermore, in our representation, an infix operator '<' is used for denoting a subclass rela-

4 We suppose that the purposes can be deduced from given background knowledge. The rights and the wrongs of this supposition are discussed in Section 5.

5 Needless to say, we suppose that the regarded properties of horse and the other concepts are registered in our background knowledge.

6 Besides, we also suppose no function symbol occurs in our formulae.

7 If this clause is transferred based on first-order logic, "p(X)<-q(X,Y), X∈a, Y∈b" can be obtained.

tionship and another infix operator '@' for a member-of relationship. A set of the declarations of the subclass relationship can be regard as a conceptual hierarchy (called **sort hierarchy**). The member-of relationships are used as type declarations. Such sort hierarchies and type declarations of constants are described in a part separated from descriptions of horn-clauses. The separated part is called **Terminological Knowledge Base**[8] (**TKB**, for short) by Guarino (1991). For instance, supposing the following descriptions occur in our TKB:

```
a<b.
e1@a.
```

Then we can also derive 'e1@b' since 'e1@a' and 'a<b'. Thus, if 'p(X:b)<-' is registered as a fact, we can conclude 'p(X:a)<-' and 'p(e1)<-'. Therefore, according to the representation, we can represent that a concept 'a' inherits the property 'p' of a concept 'b'.

Secondly, we will briefly illustrate our object-oriented macro descriptions. For instance, the following clause represents a sample rule, which denotes "a contract is valid if the declaration of intention to offer the contract is valid and the declaration of intention to accept the contract is valid":

```
valid(C:contract)<-
     valid(D1:declaration_of_intention(
              object->P:offer(principal->Y:person,
                   opposing_party->Z:person,
                   object->C))),
     valid(D2:declaration_of_intention(
              object->A:acceptance(principal->Z:person,
                   opposing_party->Y:person,
                   object->C))).
```

These variables (i.e. A, C, P, D1, D2, Y, Z) can be regarded as object variables and their sorts (i.e. acceptance, contract, offer, declaration_of_intention, person) can be regarded as the classes of the objects. Moreover, 'principal', 'opposing_party' and 'object' are called "**roles**" corresponding to *slots* used in object-oriented or frame-based representations. This formula can be expanded to the following OSL-clause:

```
valid(C:contract)<-
    valid(D1:declaration_of_intention),
    object(D1,P:offer),principal(P,Y:person),
    opposing_party(P,Z:person),object(P,C),
    valid(D2:declaration_of_intention),
    object(D2,A:acceptance),principal(A,Z:person),
    opposing_party(A,Y:person),object(A,C).
```

Lastly, we illustrate a kind of type restriction of slot values called "**role filler restriction**". Most of object-oriented (or frame-based) languages use this kind of type restriction in order to clarify the semantics of the objects or check the types of the slot values beforehand. The restrictions are declared in a part of TKB. For instance, a role filler restriction is given as follows:

8 The other part including horn-clauses is called Relational Knowledge Base (RKB).

```
principal: offer => person.
```

We can transfer this formula into the following one based on first-order logic:

```
Y ∈ person <- X∈ offer, principal(X,Y).
```

The meaning of the restriction is, intuitively speaking, "a principal of an offer should be a person". Furthermore, the information of role filler restrictions is also useful for the efficiency of our algorithm to find our similarities. Adding a condition "similar concepts should have similar structures"[9] to the prerequisites for selecting our appropriate similarities from the candidates, we reduce the number of the candidates. Consequently, we can find efficiently the appropriate similarities as Kakuta et al. (1996) has shown. Then, the information about conceptual structures used in the reduction above is obtained from the role filler restrictions. The reason is that role filler restrictions can be regarded as a kind of representations of conceptual structures.

3 An actual case as our example

In this section, we shall illustrate a case used for our experiment and introduce the relevant legal terminology.

Firstly, we will introduce a concept called **agency** (or **representation**). Agency is an institution whose purpose is the extension and supplement of private (or party) autonomy. The extension has the main purpose to cope with larger and more complex transactions with economy advances. For convenience of our illustration, we deal with only the extension. Figure 3.1 shows the legal relationships regarding agency.

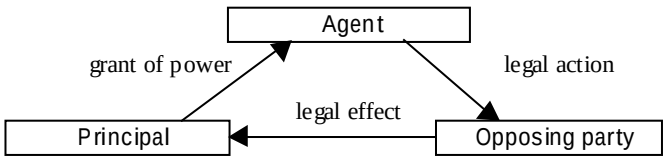

Figure 3.1. Relationships of Agency

Under these relationships, we can define "agency" as "an institution in order to make the principal person accept directly the legal effect by the act as agent". Agency (or representation) is prescribed by the Article 99 of the Civil Code of Japan. The article is shown as follows:

> "A declaration of intention, made by a representative within the scope of his authority and disclosing the fact that he is acting for his principal, shall be effective directly against the principal." (EHS, "The Civil Code of Japan")

This rule means that a representation (i.e. an act as agent) is valid only if the principal confers the authority on the agent that states "the representation is acting for the principal" to the opposing party and the representation is a legal action within the scope of the authority. We should notice here that "for the principal" does not necessarily mean "for final interests of the

9 This property is known in several researches (e.g. Gentner, 1982) for analogy.

principal". That is, so long as the agent discloses he is an agent for the principal, the representation is valid.

Secondly, we illustrate a legal concept called "**abuse of power of an agent**". This concept is used when an agent acts entirely for himself or the third parties although the representation within the scope of the authority is observed objectively. Needless to say, when an agent exceeds the scope, the representation is null in general[10]. However, in cases of "abuse of power of an agent", the representations are considered valid according to any actual legal judgment and theory in Japan. The reason is that, in order to facilitate trading, the opposing party who cannot know the inside affairs between the agents and the principal should be protected. Otherwise, the opposing parties should always ascertain the inside affair without omission. It causes, clearly, an inconsistency with the purpose of the institution. Consequently, a representation with "abuse of power of an agent" can be considered valid if the opposing party was ignorant of the abuse. On the other hand, if the opposing party was aware of the abuse, is the representation then also valid? The answer is "no" according to any actual judgment and theory; the representation is considered null. However, the explanations of the reason differ and no theory has reached a consensus. In this article, we focus only the most famous theory that has been used by Japanese Supreme Court and illustrate the application with an actual legal case.

Lastly, we show an actual case adopted as an example to experiment on our system. According to the judgment by the Japanese Supreme Court at April 20, 1967, the contract between the manager X of a store A and the chief Y of another store B as the agent has been judged null based on an analogical application with the proviso clause of Article 93 of Japanese Civil Code since X was aware of the fact that Y has acted as representative making abuse of the power of agent. The Article 93 including the proviso clause is shown as follows:

> "A declaration of intention shall not be invalidated by the fact that the declarant has made it knowing such declaration not to be his real intention; however, such declaration of intention shall be null and void, if the other party was aware, or should have been aware, of the real intention of the declarant." (EHS, "The Civil Code of Japan")

Simply speaking, if opposing parties that do not know the real intention of principals believe the declaration to be different from the intention, then the declaration should be valid. However, if exceptionally the opposing parties know the intention, then the declaration is null according to the proviso clause. These clauses above prescribe how to deal with "**mental reservation**", an example of whixh is, for instance, a joke. In Japan, any judgments relevant to "abuse of power of an agent" adopt a certain theory based on the analogical application with the proviso clause. Namely, a legal action with such an abuse is judged to be similar to a declaration of intention with mental reservation. This theory is the most predominant one in Japan. We adopt this case as our example used in this article and show that this analogy can be simulated based on our approach.

Here, we will show how the theory explains the reason for the analogical applications of the proviso clause. The purpose of the proviso clause is "it should be prohibited that an opposing party obtains advantage using a

10 Legal representatives are excluded from this case according to the Article 110 of the Civil Code of Japan.

norm (this norm indicates the first clause as far as this case is concerned) in order to protect opposing parties such that an insider affair in principal side is unknown to them, providing that the opposing party knows such an insider affair." The reason is, in a word, this kind of situation is considered unfair according to the theory. In the same way, the opposing party in our case that knows the abuse of power of the agency is also considered to obtain excessive advantage in the theory. Added to these, several popular opinions (e.g. Higashi (1968)) supporting the theory state that we should also take account of a difference that each insider affair includes similarly between the real intention and the legal action (i.e. the declaration or the act as the agent). Thus, the two cases are judged similar based on the theory.

Since we can observe that the explanations of these legal purposes have the same structure, GDA program can find the similarities based on the theory.

4 Demonstrations

In this section, we demonstrate a process of analogical reasoning in our system using an example based on the case and the arguments shown in section 3.

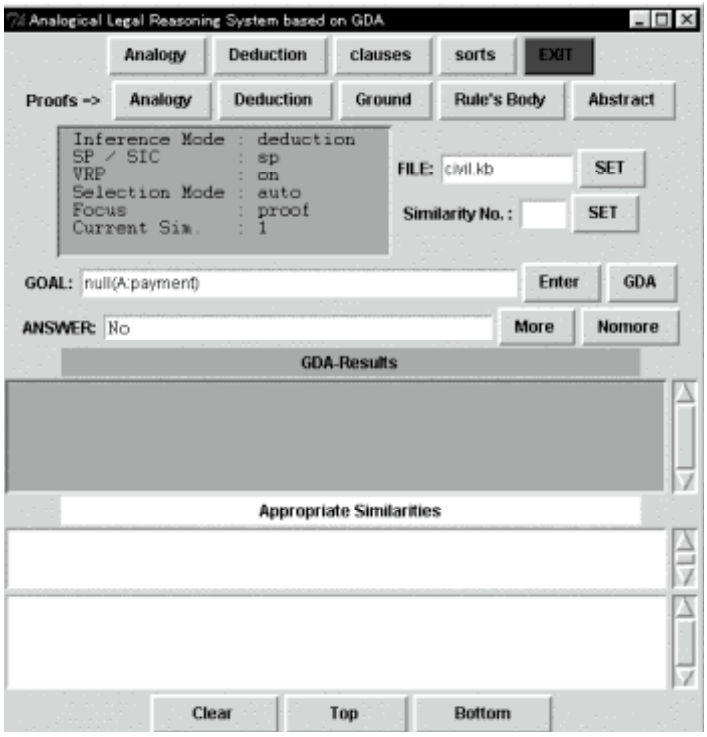

Figure 4.1. Main window

Since our system is described in SICStus-Prolog and Tcl/Tk, it is available independently of hardware environments and operating systems so long as these programming languages are installed into the user environments.

Firstly, after our system is invoked, the main window appears on the display and is shown in Figure 4.1. In this window, to load our case and background knowledge, we input the name "`civil.kb`" of the file in which these are described. Now, we try to inquire whether the payment is null or not in

the case. As shown in the figure, we can find that `null(X:payment)` is inputted in the goal entry.

In the knowledge base, the formula of the proviso clause of the Article 93 is shown as follows:

```
null(E:event):-
    exist(A:declaration_of_intention(
      real_intention->X:event,
        imply->E,
        opposing_party->Y:person(known->D:difference))),
  difference(D,X,E).
```

Moreover, we assume '`payment<event`' to be in our hierarchy. Then, the structure of this rule is illustrated in Figure 4.2.

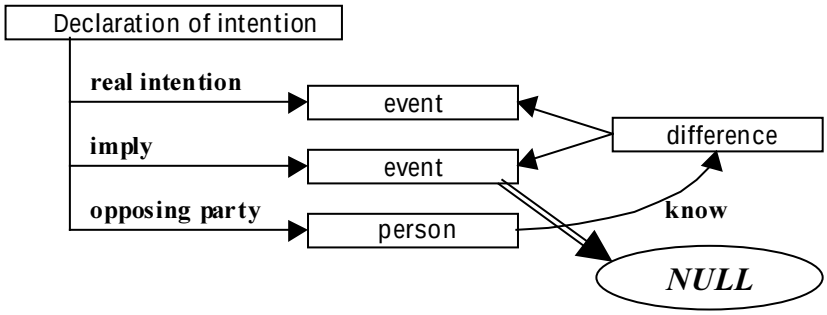

Figure 4.2. The structure of 'mental reservation'

The following formulae represent our case for the abuse of power of the agency:

```
exist(act1:act_as_agent(
    principal->strB:legal_person(
          interest->npm1:nopayment),
          imply->pm1:payment,
          opposing_party->manX:person(
             known->diff1:difference),
          agent->chiY:person)).
difference(diff1,npm1,pm1).
```

This structure is illustrated in Figure 4.3.

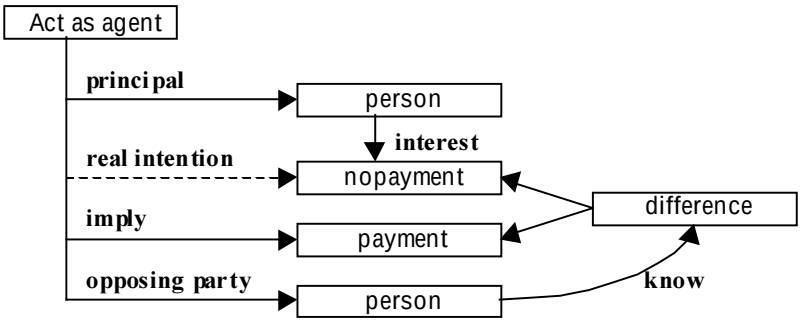

Figure 4.3. The structure of our case

The real intention of the act as the agent cannot be deduced directly from only these formulae. However, by using the following horn-clause described in the knowledge base, we can deduce the real intention as the role link `real_intention` represented by broken line in Figure 4.3:

```
real_intention(A:act_as_agent(
          principal->X:person),E:event):-
                interest(X,E).
```

Nevertheless, we cannot deduce "the payment is null" (i.e. `null(pm1)`) from our formulae. The reason is that the concept `act_as_agent` cannot be applied to the concept `declaration_of_intention` occurring in the formula of the proviso clause since `act_as_agent` is not subsumed to `declaration_of_intention`.

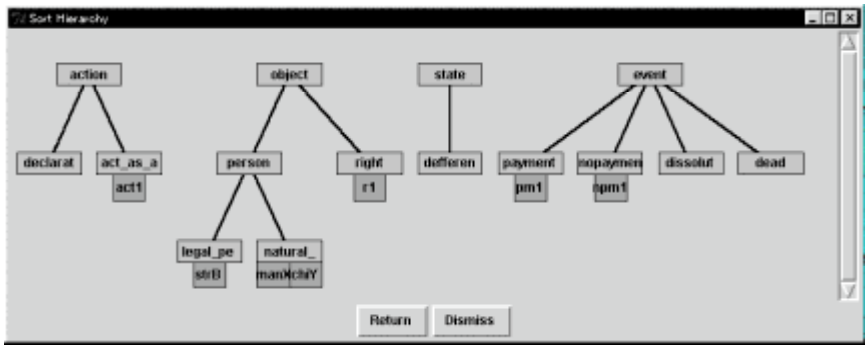

Figure 4.4. Our sort hierarchy

Our subsumptions, namely the subclass relationships, are presented as our sort hierarchy and shown in Figure 4.4. As the result, our system returns "No" as Figure 4.1 shows. To draw "the payment is null", we need to start GDA procedures for analogical reasoning.

Secondly, we will demonstrate the detection of the similarities based on GDA. To compute the similarities, our system executes the following two procedures:

1) The system checks whether undesirable state of affairs arises on the situation provided in our knowledge base. This check is, technical speaking, executed with the system deducing the state from our case and background knowledge using *forward reasoning*[11] (e.g. Inoue *et al.*, (1993)). Thus, this state and its proof (i.e. explanation) are regarded as a goal and a proof in GDA framework respectively.
2) According to *GDA*, the system tries to make the group of sorts such that the proof can have the same structure. These groups are regarded as representations of our similarities. If a proof derived from the legal premise of a rule has the same structure as the proof deduced from the given case, then we can conclude that the rule is analogically applicable.

The structures of the proof are shown in Figure 4.2 and 4.3. Moreover, representing the structures by macro-expanded OSL-formulae, we obtain the formulae in Figures 4.5 and 4.6.

11 A legal reasoning by analogy using forward reasoning is illustrated in Yoshino et al. (1993). Furthermore, the related methodologies are shown in Rouveirol and Puget (1990) and Muggleton (1990).

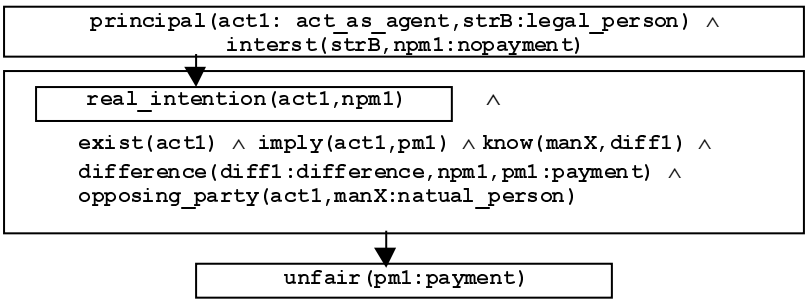

Figure 4.5. The explanation structure from our case

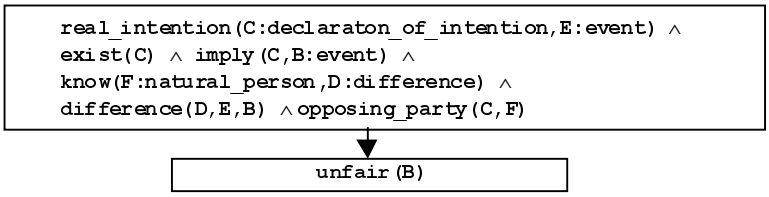

Figure 4.6. The explanation structure from the premises of the rule

Our system displays the formulae as proof trees shown in Figure 4.7 and 4.8. Clearly, we can observe the same partial structures in the trees. Consequently, our system computes the abstract structure shown in Figure 4.9. The results of the detected similarities are shown in Figure 4.10. For instance, as the first detected similarities, `[act_as_agent, declaration_of_intention]` can be observed. This description means that `act_as_agent` and `declaration_of_intention` are similar. The point we wish to emphasize is that a framework to select the significant role links can also be automatically applied to several approaches (e.g. Winston (1980); Gentner (1982)) for structural analogy.

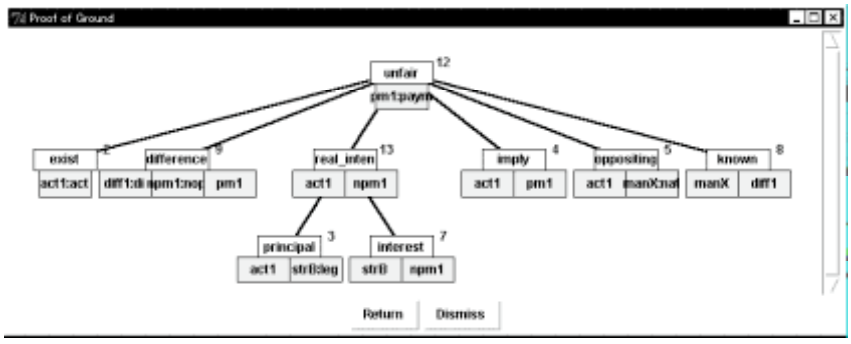

Figure 4.7. The explanation from our case

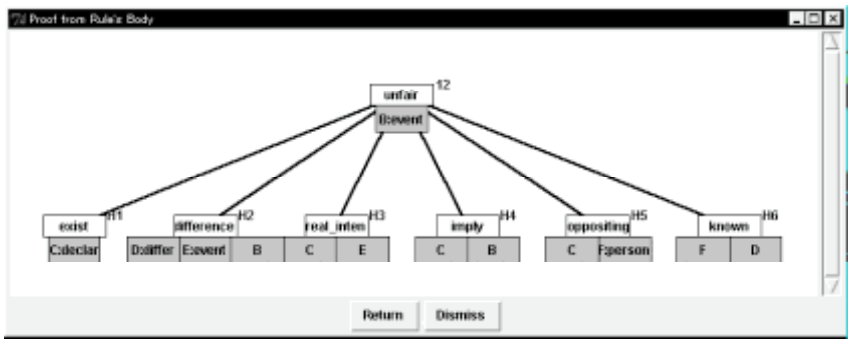

Figure 4.8. The explanation from the premises of the rule

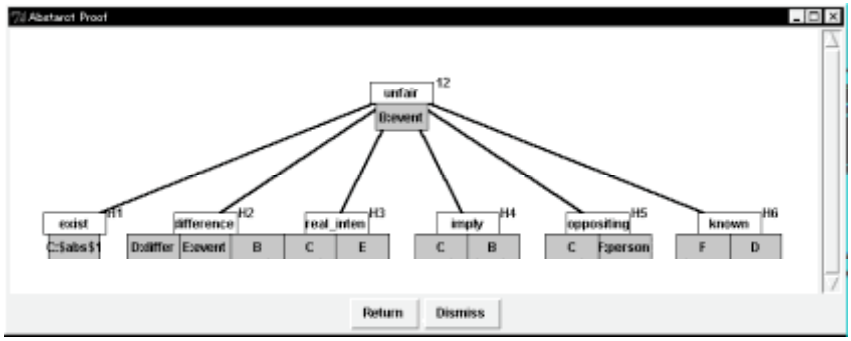

Figure 4.9. The explanation based on the abstraction

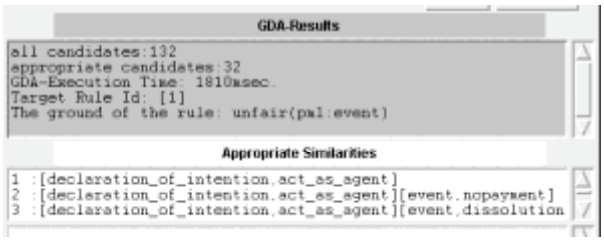

Figure 4.10. Results of GDA

Here, we select the first similarity and switch the reasoning mode from deduction-mode to analogy-mode pushing the button labeled "Analogy". In analogy-mode, our system displays our sort hierarchy shown in Figure 4.11. Observing this hierarchy, we can notice that a hypothetical sort `$abs$1` is added as a super sort of `act_as_agent` and `declaration_of_intention`. Moreover, in Figure 4.12 we can also notice that `declaration_of_intention` in the clause is replaced by the hypothetical sort. This replacement is considered as a generalization of the clause.

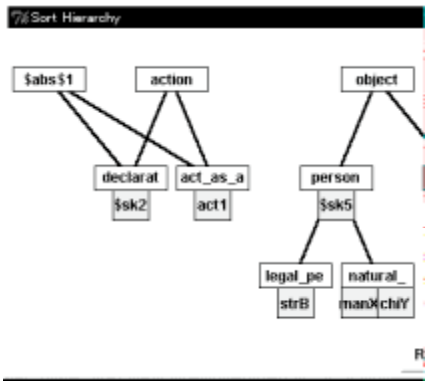

Figure 4.11. Our sort hierarchy in analogy-mode

Figure 4.12. The hypothetical rule

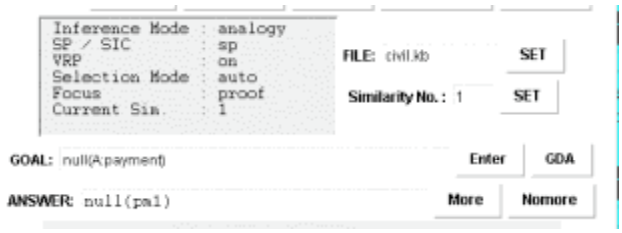

Figure 4.13. The results of inference by analogy

Lastly, we try to inquire whether a payment is null again in analogy-mode. As we have expected, our system returns `null(pm1)` shown in Figure 4.13. The reason is that `act_as_agent` can be applied to the rule since the rule is generalized. Finally, we can observe that our system performs analogical legal reasoning dependent on the legal purpose and the explanation.

5 Discussions

The essential assumption for our system functioning well is that the purposes of legal rules can be drawn from the given background knowledge. In the previous study, where we used only a toy example, a kind of common sense knowledge (e.g. “if a big object moves in a park then the park is dangerous”) is required in the background knowledge. Admittedly, this state may be accepted after we have already heard it. However, it seems difficult to provide it beforehand. On the other hand, trying to check up the example used in this article, we can find no description based on such common sense knowledge. That is, all of the descriptions consist of only the materials described in legal textbooks or treatises. Accordingly, we can expect that the necessity of common sense knowledge decrease proportionately if we deal with more actual legal cases. Our experimental results are considered to support this expectation.

6 Conclusions

In this article, we have shown that our approach, which is based on GDA in order to perform analogical reasoning dependent on legal purposes and the explanations, is applicable to an actual legal case and our system based on the approach works actually. Then, utilizing a kind of biases that the object oriented knowledge representation has, we have more faithfully reflected the semantics and achieved the computational efficiency.

Lastly, our future work is that, as we have mentioned in section 5, we enrich our legal knowledge base for actual legal problems. To build the actual knowledge bases, we forecast that we would need convenient knowledge editors and debug tools. A GDA-based framework to debug knowledge bases has already been presented by Okubo and Haraguchi (1997). Furthermore, our revised system that includes a visual knowledge editor based on the framework is now being designed.

References

Eibun-Horei-Sha (1992)
Eibun-Horei-Sha, The Civil Code of Japan, *EHS Law Bulletin Series* 2, 1992.

Gentner (1982)
Gentner, D., Are Scientific Analogies Metaphors, *Metaphor: Problems and Perspectives* 1982, pp. 106-132.
Guarino (1991)
Guarino, N., A concise presentation of ITL, *Processing Declarative Knowledge, International Workshop PDK '91 Proceedings*, Springer-Verlag 1991, pp. 141-190,.
Haraguchi (1995)
Haraguchi, M., A Reasoning System for Legal Analogy, *Machine Intelligence* Vol.14, Oxford University Press 1995.
Hart (1958)
Hart, H.L.A, Positivism and the Separation of Law and Moral, *Harvard Law Review* 71, 1958, pp. 593-629.
Higashi (1968)
Higashi, T., Applicability of Art.93 of Civil Code to Abuse of Power of an Agent (in Japanese), *Kobe Law Journal* 18(2) 1968, pp. 149-157.
Inoue et al. (1993)
Inoue, K., Ohta, Y., Hasegawa, R. and Nakashima, M., Bottom-up abduction by model generation, *IJCAI-93 Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* 1, 1993, pp.102-108.
Kakuta et al. (1996)
Kakuta, T., Haraguchi, M. and Okubo, Y., Legal reasoning by structural analogy based on goal-dependent abstraction, in Van Kralingen R.W., H.J. van den Herik, J.E.J. Prins, M. Sergot, J. Zeleznikow (eds), *Legal Knowledge Based Systems: Foundations of Legal Knowledge Systems, JURIX '96,* Tilburg: Tilburg University Press 1996, pp.111-121.
Kakuta et al. (1997)
Kakuta, T., Haraguchi, M. and Okubo, Y., A goal-dependent abstraction for legal reasoning by analogy, *Journal of Artificial Intelligence and Law* 5, 1997, pp. 97-118.
Kedar-Cabelli (1985)
Kedar-Cabelli, S., Purpose-Directed Analogy, *Proceedings of the 7th Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Lawrence Erlbaum Associates 1985, pp. 150-159.
Kifer et al. (1993)
Kifer, M., Lausen, G. and Wn, J., *Logical Foundations of Object-Oriented and Frame-Based Languages, Technical Report,* Stony Brook: Department of Computer Science SUNY, 1993.
Lloyd (1984)
Lloyd, J.W., *Foundations of Logic Programming,* Springer-Verlag 1984.
Muggleton (1990)
Muggleton, S.,Inductive logic programming, *Proceedings of 1st Workshop of Algorithmic Learning Theory,* 1990, pp.42-66.
Nitta et al. (1993)
Nitta, K., Wong, S. and Ohtake, Y., A computational model for trial reasoning, *Proceedings of the Fourth International Conference on Artificial Intelligence and Law,* 1993, pp. 20-29.
Okubo and Haraguchi (1994)
Okubo, Y. and Haraguchi, M., Construction of Predicate Mappings for Goal-Dependent Abstraction, *Algorithmic Learning Theory* 1994, pp. 516-531.

Okubo and Haraguchi (1997)
Okubo, Y. and Haraguchi, M., A Method of Knowledge Revision with Goal-Dependent Abstraction (in Japanese), *Computer Software* 14(5), 1997, pp. 67-79.
Plaisted (1981)
Plaisted, D.A., Theorem Proving with Abstraction, *Artificial Intelligence* 16, 1981, pp. 47-108.
Rouveirol and Puget (1990)
Rouveirol, C. and Puget, J.F., Beyond inversion of resolution, *Machine Learning (Proceedings of the Seventh International Conference)*, 1990, pp. 122-30.
Tenenberg (1989)
Tenenberg, J.D., Abstracting First-Order Theories, *Change of Representation and Inductive Bias*, Kluwer Academic Publishers 1989, pp. 67-79.
Walter (1988)
Walter, C., Many-Sorted Unification, *JACM* 35(1), 1988, pp. 1-17.
Winston (1980)
Winston, P.H., Learning and Reasoning by Analogy, *CACM* 2(3), 1980, pp. 689-703.
Yoshino et al. (1993)
Yoshino, H., Haraguchi, M., Sakurai, S. and Kagayama, S., Towards a Legal Analogical Reasoning System, *Proceedings of 4th ICAIL*, 1993, pp. 110-116.