A Demonstration of a Legal Reasoning System Based on Teleological Analogies

Tokuyasu Kakuta
Faculty of Law,
Hokkaido University
Kita-ku, N-9, W-7
Sapporo 060-0809, JAPAN
+81-11-716-2111(ex.3149)
kaku@juris.hokudai.ac.jp

Makoto Haraguchi
Division of Electronics and Information Engineering,
Hokkaido University
Kita-ku, N-13, W-8
Sapporo 060-0813, JAPAN
+81-11-706-7106
makoto@db-ei.eng.hokudai.ac.jp

ABSTRACT

In this article, we demonstrate our analogical legal reasoning system based on a teleological approach to interpret laws, using an actual example. By this demonstration, we show the validity of our approach. The example is based on a real legal problem and consists of an actual case, the actual decision on the case by the Japanese Supreme Court and two major doctrines on the case in Japan. The problem and the doctrines are also analyzed from the viewpoint of **GDA (Goal-Dependent Abstraction)** framework in this article. We further show that our system using GDA can provide helpful information to evaluate and revise interpretations of legal rules.

Keywords

Analogy, Legal reasoning system, Goal-Dependent Abstraction, Order-sorted logic.

1. INTRODUCTION

This article shows the validity of our approach to draw analogies between legal rules and cases, by demonstrating our legal reasoning system with an actual example. Our system adopts a teleological approach in which a kind of practical reasoning [12] is used. The teleological analogy in our system is based on inferring purposes of laws as the "goals". For instance, one of the approaches based on teleological reasoning with goals is analyzed in [2]. Our study is concerned with a relation between teleological reasoning and analogy based on an abstraction strategy.

In this article, we show that our system can present not only simulations to accomplish the purposes of legal rules as things turned out but also helpful information on the evaluations of how reasonably the rules are applied. Actually, the demonstration shown in Section 3 indicates that our system provides crucial information to discover a counterexample to attack an actual doctrine many lawyers have their doubts about.

The teleological analogy in our study is that a legal rule is applied analogically to a case if the case shares a reason why the rule should be applied. In our framework, the reason is regarded as the purpose of the rule. To implement our teleological analogy, our system executes the following steps:

1) Deducing purposes of legal rules by forward reasoning.
2) Discovering similarities that preserve the history (that is, the explanation or proof of the purposes obtained in step 1) based on **GDA** framework.
3) Applying analogically the legal rules by deduction and **Sorted Generalization**.

Step 1) process derives all of the possible purposes that the candidates of legal rules to be applied intend to achieve. We should notice that this process might detect unforeseen purposes by the users. In this step, we suppose that the purposes of legal rules are to prevent infringements of various types of rights or disturbances of legal balance. According to this supposition, we can detect the infringements and the disturbances by checking ones to be deduced if the rules are not provided. Thus, we regard the protection from the detected situations as the purposes of the rules.

Step 2) process finds not only similarities between cases and premises of legal rules but also the ranges of possible premises that can be applied to the rules based on the purposes. These ranges can provide helpful information on the appropriateness of the analogical applications. In Section 3, we show that such information can be used in order to pose counterexamples for attacking the interpretations of the rules or discover new interpretations. To implement this step, we have proposed a framework [7,8,9] based on **Goal-Dependent Abstraction** (**GDA**, for short) [11].

GDA is a framework to detect groups of concepts that can share a proof structure of a given goal. The goal and its proof in our framework are the purpose of a legal rule and the explanation of the purpose respectively. The concepts belonging to a group detected by GDA can be considered as similar concepts in the analogical application of the rule. For instance, we assume the purpose of a legal rule "cars should be prohibited from entering public parks" [5] to be "to avoid dangerous situations" and the explanation of the rule to be "cars being movable and large are dangerous". If cars are allowed to enter a public park by nullifying the rule, we can deduce that a dangerous situation, according to the above explanation, will occur. Then, using a variable α that denotes a hypothetical concept, we can apply the rule analogically and make the following explanation:

"Dangerous situations are caused by αs entering public parks since αs are movable and large".

ICAIL-99 Oslo, Norway Copyright (c) 1999 ACM 1-58113-165-8/99/6 $5.00

GDA can find all concepts that belong to α. Hence, we can find, for instance, that horses and helicopters belong to α. We can observe that the explanation structure is preserved even if concepts belonging to α are replaced each with other[1]. The formulation of GDA is shown in APPENDIX B.

Now, we turn to step 3 for analogical applications of legal rules. Based on groups detected by GDA, first we create a hypothetical conceptual hierarchy by regarding the hypothetical concepts like α, β as the super concepts of the concepts in each group. That is, if a concept belongs to the group of α,α is a super concept of the concepts in the hypothetical hierarchy[2]. For instance, if a set {car, horse, helicopter} is included in α's group, we can create the following hierarchy:

car =< α, horse =< α, helicopter =< α,

where "=<" denotes a subclass relationship between concepts.

Secondly, we replace the concepts in the original legal rule with hypothetical concepts of the groups including each concept. For instance, replacing "car" with α, we can obtain the following hypothetical rule:

"αs should be prohibited from entering public parks".

This operation used to rewrite rules is called "**Sorted Generalization**" [4]. Lastly, by using deductive reasoning, another concept belonging to a hypothetical concept can be applied to the hypothetical rule. For instance, a horse can be applied to the hypothetical rule above since a horse is also α according to the hypothetical hierarchy. These procedures provide a method for analogical applications of laws. This analogy based on Sorted Generalization and deduction was also introduced in [4].

In [9], we have not demonstrated switching analogies dependent on the explanations of the purposes of laws yet. In this article, by showing the behavior of switching, we reconfirm the work of our teleological analogy. Furthermore, we show that our system can provide helpful information to evaluate and revise interpretations of legal rules.

In this article, first we illustrate an actual case in order to demonstrate our system. Secondly, we demonstrate our system with an example based on the actual case. Finally, the formalization of our framework are shown in APPENDIX B.

2. ILLUSTRATION OF AN ACTUAL EXAMPLE

In this article, our example deals with the problem concerned with "abuse of the power of agents". The typical case was decided by the Japanese Supreme Court in 1967 [6]. The outline of this case is shown as follows:

- •An agent 'X' of a company 'C' contracted a sale with a person 'Y'.
- • ' X' had the right to contract sales.
- •The contract was disadvantageous to the principal 'C'.
- • 'X' and 'Y' conspired together to make the contract.

This case involves "abuse of power of agent". A legal concept "abuse of power of agent" is defined as a legal action by the agent obtaining advantages to himself or herself or third parties with disadvantage of the principal and that is objectively an action within the range of the agency. According to legal opinion in Japan, even a representation with such a abuse is considered as valid since the representation by the agent is objectively an action within the range of the agency. Although no article in the Civil Code proscribes abuse, this view is supported by all Japanese doctrines as to agency. The reason is that the purpose of the institution of agency is mainly to allow economic transactions to take place smoothly[3]. Now, if the representation by the agent objectively exceeds the range of agency, the representation is generally null action. This is prescribed by the Civil Code. However, if the opposing party knows about the abuse of agency, the representation is considered null. All doctrines of agency in Japan support this conclusion. However, since the Civil Code has no rule to prohibit the abuse, there are the several explanations as to why this is so. Up to now, no single explanation creates a consensus. In the following subsections, we intend to show the two prevailing doctrines about our case.

2.1 Analogical Applications of Article 93

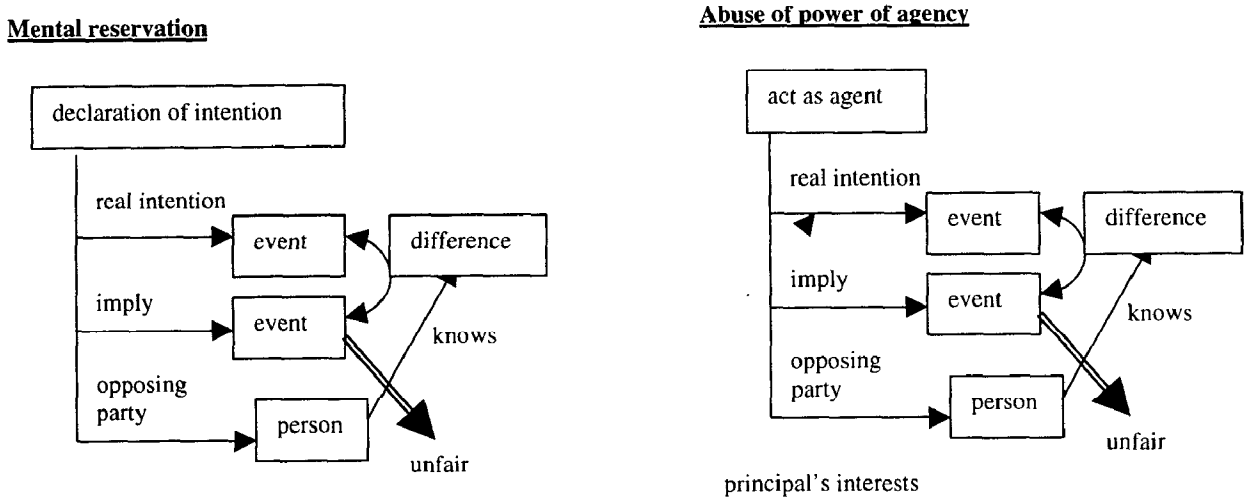
The Supreme Court has adopted a doctrine based on analogical applications of the proviso clause of article 93 of Japanese Civil Code. This article deals with "mental reservations". Mental reservations are declarations of intentions different from the real intention. For instance, a joke is a mental reservation. According to Article 93, a representation meant to be a joke is considered a valid representation if it is believed by the opposing party. However, if a joke is obviously, the representation is null. The later case is dealt with by the proviso clause. The doctrine based on an analogical application of the article considers that the difference between the real intention and the declaration is similar to the difference between the principal's interests and the act as an agent. The reason is, according to the following principle that is a well-known one in juristic fields, that when an outside appearance is different from the internal facts, outside persons who believe the appearance should be protected against disadvantages caused by the difference. Clearly, this principle is applicable to cases of mental reservations and the abuse of agency. Therefore, a person who knows the difference should not be protected. If (s)he is protected, we can forecast that an unfair situation is caused. This explanation is shared between cases of mental reservations and the abuse of agency. The reason the analogical application is accepted is because of the explanation. The structure of the explanation is shown in Figure 1.

---

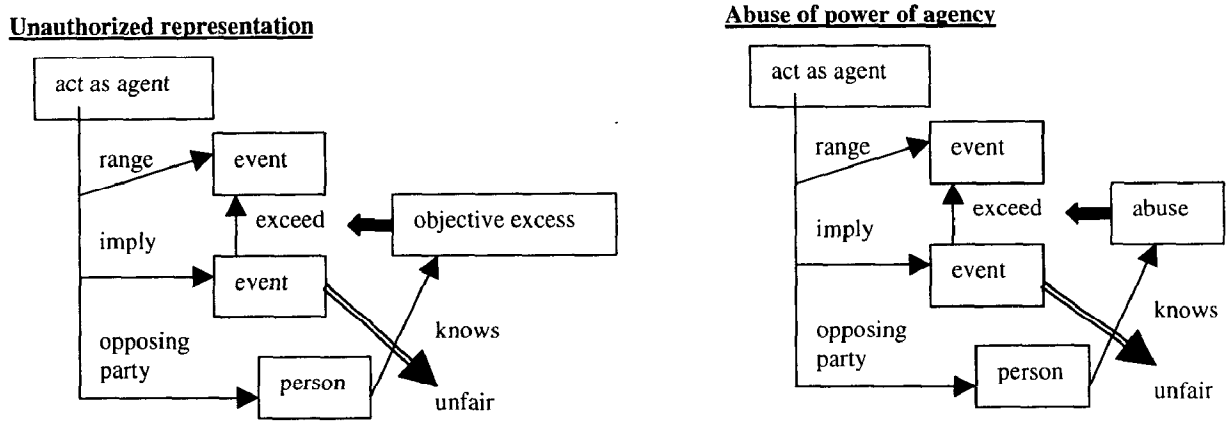
[1] Furthermore, we can replace "public park" with β. Then, GDA can also find all concepts that belong to β simultaneously. Mathematically speaking, GDA detects partitions of a set of concepts.

[2] Semantically, the extension of a hypothetical super concept is the union of extensions of the concepts belonging to the group.

[3] Though there are other purposes, we are not concerned with them in this article.



[Figure 1: Explanation structures (Art. 93)]



[Figure 2: Explanation structures (Art. 113)]

2.2 Interpreting into Unauthorized Representations

Recently, doctrines in which acts as agents based on abuse of power by the agents are regarded as unauthorized representations are strongly advocated in Japan. In this article, we show one of the doctrines, which is proposed in [3].

Generally, representations with abuses of agency are regarded as valid since protecting the opposing parties, who know nothing about the internal facts (i.e., the internal affairs between the agents and the principals), coincides with the purpose of the institution of agency. The purpose is, as we have mentioned, mainly to allow economic transactions to take place smoothly since opposing parties can make contracts without investigating the internal facts. This independence from internal affairs is considered as the essential property of agency. However, recently, some jurists (e.g., [1,3]) advocate that independence should not always be understood dogmatically. According to [3], the condition of independence cannot be satisfied if opposing parties know the internal information. Therefore, we can consider that the opposing parties should not be protected. In other words, if they are protected, we can forecast that an unfair situation is caused. That is, since representations involving abuses of the power of agency exceed the intentions of the principals, unfair situations are caused if the representations are considered valid.

Now, according to Article 113 of the Japanese Civil Code prescribing unauthorized representation, if a representation objectively exceeds the range of agency and the opposing party knows it then the representation is null. The representation is called "unauthorized representation"[4]. When a representation

[4] Although the condition of the opposing party being in bad faith (mala fide) is not described in Article 113, by taking Article 110 into account, which deals with "apparent agency", where the representation of the party in good faith (bona fide) is regarded as valid, we need to add the condition to our logical formulae.

objectively exceeds the range of agency, we can consider that the representation exceeds the intention of the principal. When an opposing party knows that an agent is exceeding the range of agency, there is dependence on the internal affairs of the principal and the agent. Therefore, the purpose of Article 113, to prevent unfair situations from being caused by acts of agency that exceed the intentions of the principals and have the dependence, is valid. This explanation has the same structure as the explanation about abuses of the power of agency. Thus, we can find that these two cases share the same explanation. According to our approach, we can conclude the representation is null by the application of Article 113. Generally, in our legal field, this application isn't said an "analogical" one. However, our framework for analogies can deal with the application as an analogical one. That is, the GDA calculations with the above explanation in [3] can find that abuses of the power of agency is similar to exceeding the objective range of agency. The structure of the explanation is shown in Figure 2.

3. DEMONSTRATIONS

This section demonstrates the work of our system with examples based on the two doctrines that are explained in previous section. The figures in this section are hardcopies of the windows our system displays. Our knowledge descriptions of the example are shown in APPENDIX A. To implement our teleological analogy, our system finds the ranges of applicable concepts to legal rules dependent on the explanation of the purposes of the rules. These ranges are regarded as the similarities used for the analogy and represented by conceptual hierarchies (see, e.g., Figure 7 and 10) in our system. For instance, if the system detects the explanation structure corresponding to the former doctrine shown in previous section, the system can find that "act as agent" is similar to "declaration of intention". On the other hand, in the case of the later doctrine, the system can obtain that "objective excess" is similar to "abuse".

Before we demonstrate the work of the system, let us start with showing the used knowledge base (KB, for short), which consists of the following modules: legal rules, a set of logical formulae to represent facts constructing a case, a conceptual hierarchy and a set of logical formulae to represent legal background knowledge. The set of formulae as our case represents the following statements:

An act as agent implies a payment different from the real intention.

The opposing party knows the difference.

Furthermore, according to the explanations in the previous section, the following statements are registered as the parts of our background knowledge:

a) If the declaration of an intention is different from the real intention and the opposing party who knows the difference is protected, then unfair states are caused in the transaction based on the declaration.

b) If the act as an agent is different from the principal's interest and the opposing party who knows the difference is protected, then unfair states are caused in the transaction based on the agency.

c) If a registration is different from the real intention and the opposing party who knows the difference is protected, then unfair states are caused in the transaction based on the registration.

d) The principal's interest in an agency is regarded as the real intention.

e) If the opposing party of an agency who knows the abuse of the agency is protected, then unfair states are caused in the transaction based on the agency.

f) If the opposing party of an agency who knows the abuse of the agency is protected, then unfair states are caused in the transaction based on the agency.

These statements are often used for interpreting several legal rules in basic legal textbooks and legal treatises. Thus, these are not ad hoc information for dealing with special doctrines. The transformed logical formulae from the statements are shown in APPENDIX A.

Under the KB, our system tries step 1 in Section 1. As the trigger of this step, the users need to confirm the failure to try to derive "the payment by the agency is null" by ordinary deduction. When the failure occurs in the trial, our system lists the candidates of rules that may be applied analogically. In this demonstration, Article 93 and 113 are detected as the candidates. The users can select which rules should be checked by our system. For instance, let us first select Article 93. As we have mentioned in Section 1, if the purpose of the rule (Article 93) and its explanation are applied to our case, we can regard the explanation as a reason for applying the rule analogically.

To calculate the purpose of the rule, our system executes the following procedures:

1) Removing the rule from our KB.

2) Setting the rule's conditions and the negation of the rule's conclusion as the assumptions to use 3).

3) Checking what is derived from the KB by forward reasoning.

4) Outputting the derived conclusion only if the conclusion is denoted by a predicate that occurs in the given CF-predicates list. ("CF-predicates list" is a set of predicate symbols to represent infringements of legal interests, for instance, {unfair, troublesome, irritating, danger, noisy, . . .}. We assume CF-predicates lists to be added to KBs beforehand.)

Prohibiting the derived conclusions is regarded as the purpose of the rule according to our supposition[5] shown in Section 1. In this demonstration, our system derives the goal denoting that the payment based on the agency in our case is "unfair" as such an infringement. The result of this step is shown in Figure 3. This structure can be transformed into Figure 4 based on a semantic network form. We can find the network has the same structure as Figure 1. Furthermore, our system preserves the history of the derivation (i.e., the proof as the explanation of the purpose) and inputs it into GDA process. Our proof[6] is composed using b) in the KB.

Using the proof obtained in previous step, our system starts GDA process. In a word, this process calculates groups of similar concepts that can share the same structure of the proof. Our system represents the groups by hypothetical conceptual hierarchies. For

[5] Let us be remained that this process finds how infringements of legal interests are caused from the KB if the rule is not provided.

[6] This proof is drawn automatically from our KB.

instance, a group of similar concepts {declaration_of_intention, act_as_agent, registration} is found in this demonstration. The group is represented by the hierarchy shown in Figure 7. This group can be also regarded as a range of analogical applications based on the proof. When the range is too small for the user to draw the expected analogies, we can guess that more general knowledge descriptions are shortage or that some concepts being out of the range need some properties occur in proof. On the other hand, if the range is too large, the KB may be short of some descriptions to distinguish the concepts in the group. Generally, if the group includes unexpected concepts, the concepts can be used as crucial elements to discover a new interpretation or a counterexample. Observing Figure 7, actually, we can find that our system includes "registration" in the range based on the explanation corresponding to the first doctrine. As some Japanese jurists point out, this doctrine is too abstract to provide the ground of the analogical application. That is, "registration" can be used as a counterexample to attack the doctrine. Otherwise, we should admit the discovery of the new interpretation to apply the article to the cases that deal with registrations. However, according to treatises of civil law in Japan, it is hard to adopt this interpretation actually. Whether we utilize "registration" as a counterexample or a new interpretation, the important point to note is that our system not only simulates analogical reasoning but also can provide information to check or refine KBs for the analogical applications.

According to the similarity based on the calculated hierarchy, we can conclude that the payment is a null action by our analogy. In this analogical reasoning step, our system rewrites "declaration_of_intention" in Article 93 as the hypothetical concept "abs1" used in Figure 6 and 7, and performs deductive reasoning with the hierarchy and the rewritten article shown in Figure 8. The "act_as_agent" in our case can be applied to the "abs1" in the rewritten article since "abs1" subsumes "act_as_agent". Therefore, we can draw the nullity of the payment.

On the other hand, in the case of selecting Article 113, our system can behave corresponding to the second doctrine shown in previous section. In our demonstration, the system first detects an unfairness of the agency, using f) in the KB. The found explanation structure is shown in the Figure 9. According to this explanation, GDA process outputs a similar group {objective_excess, abuse} as the hierarchy shown in Figure 10, by abstracting the structure shown in Figure 9. This group is considered as a suitable one for the present.

4. CONCLUSIONS

In concluding, by the demonstration, we have confirmed that our system can be utilized in order to solve actual legal problems. Moreover, since our system finds the ranges of the applications of legal rules based on the teleological analogy, we have shown that we can obtain the helpful information to refine KBs or discover new interpretation to apply rules.

Lastly, Our system[7] has already released by AITEC (Research Institute for Advanced Information Technology) in Japan and can be obtained as free software[8]. The environment of our system requires SICStus-Prolog3#5 or #7 and Tcl7.6/Tk4.2 or Tcl8.1/Tk8.1. If both systems are installed into the user's environments, our system can work on Unix and Windows95.

[7] For the basic GDA-algorithm, see [8]. For the GUI and efficiency of our system, see [9].

[8] The URL is "http: //www.icot.or.jp/AITEC/HomePage.html".

REFERENCES

[1] Fukunaga, R. Zur Lehre vom Miβbrauch der Vertretungsmacht (2) (in Japanese). *Sophia Law Review* 22-3 (1979), 177—226.

[2] Hage, J. Teleological reasoning in reason-based logic in *Proceedings of 5th ICAIL* (1995), 11—20.

[3] Hamagami, N. DAIRININ NO KENGENRANYOU (in Japanese). *MINPOU NO HANNREI*, 3rd edition, 29—32, Yuhikaku, 1979.

[4] Haraguchi, M. A Reasoning System for Legal Analogy. *Machine Intelligence* Vol.14 (1995), Oxford University Press.

[5] Hart, H.L.A. Positivism and the Separation of Law and Moral. *Harvard Law Review* 71 (1958), 593— 629.

[6] Higashi, T. Applicability of Art.93 of Civil Code to Abuse of Power of an Agent (in Japanese). *Kobe Law Journal* 18-2 (1968), 149—157.

[7] Kakuta, T., Haraguchi, M. and Okubo, Y. Legal Reasoning by Structural Analogy Based on Goal-Dependent Abstraction. *Legal Knowledge Based Systems (Proceedings of JURIX'96)*, 111—121, 1996.

[8] Kakuta, T., Haraguchi, M. and Okubo, Y. A goal-dependent abstraction for legal reasoning by analogy. *Journal of Artificial Intelligence and Law* 5 (1997), Kluwer Academic Publishers, 97— 118.

[9] Kakuta, T., Haraguchi, M. An Actual Application of an Analogical Legal Reasoning System Dependent on Legal Purposes. *Legal Knowledge Based Systems (Proceedings of JURIX'98)*, 31—44, 1998.

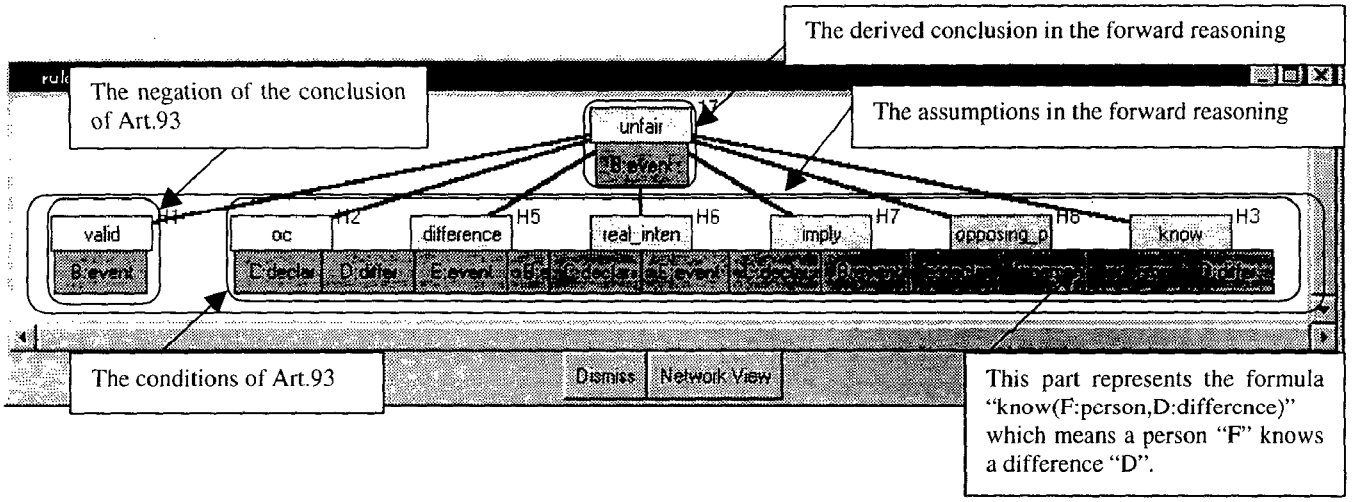
[10] Kifer, M., Lausen, G. and Wn, J. Logical Foundations of Object-Oriented and Frame-Based Languages. *Technical Report, Department of Computer Science SUNY at Stony Book*, 1993.

[11] Okubo, Y. and Haraguchi, M. Construction of Predicate Mappings for Goal-Dependent Abstraction. *Algorithmic Learning Theory '94* (1994), Springer, 516— 531.

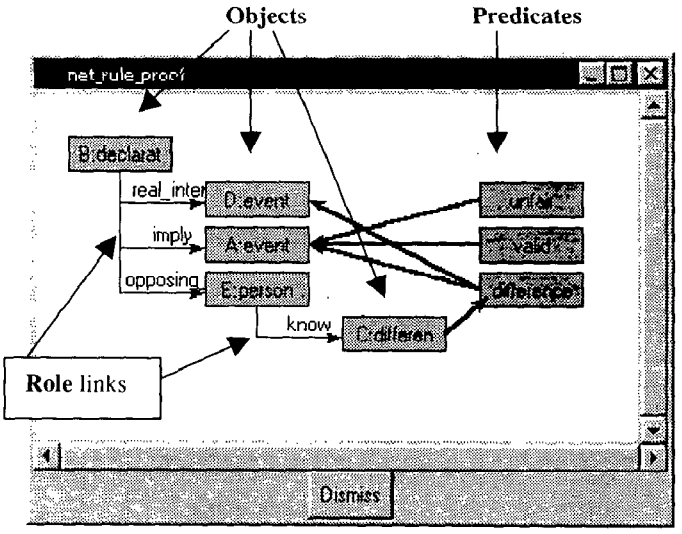
[12] Raz, J. Practical Reasoning, Oxford University Press., 1978.

[13] Tenenberg, J.D. Abstracting First-Order Theories. *Change of Representation and Inductive Bias*, 67— 79, Kluwer Academic Publishers, 1989.

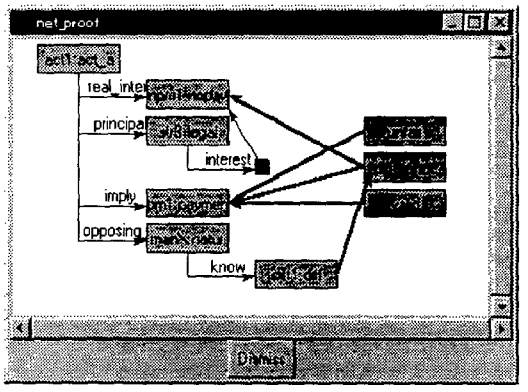
[14] Walter, C. Many-Sorted Unification. *JACM* 35-1 (1988), 1— 17.



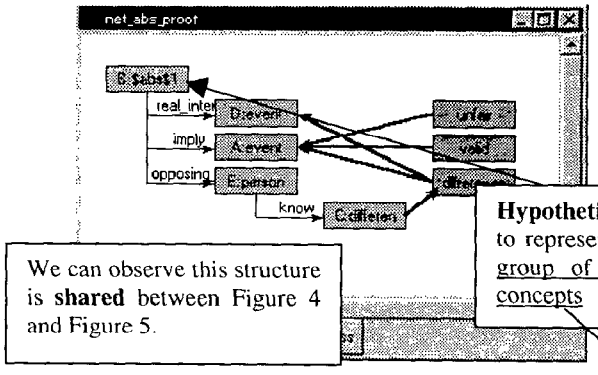
[Figure 3: The structure of the detected proof to derive the purpose of Art. 93]



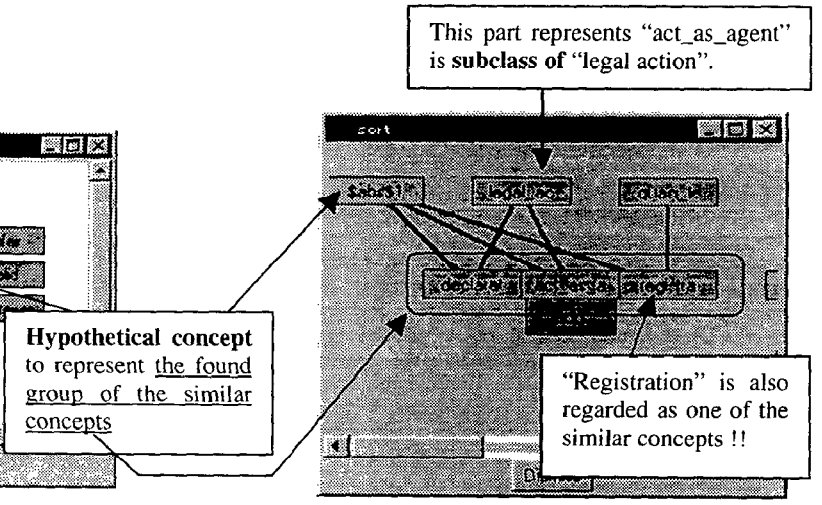
[Figure 4: The network view of Fig.3-1]



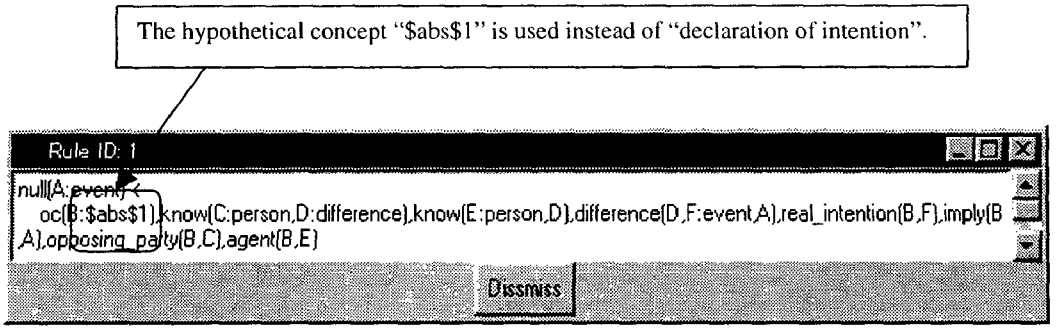
[Figure 5: The structure of our case]



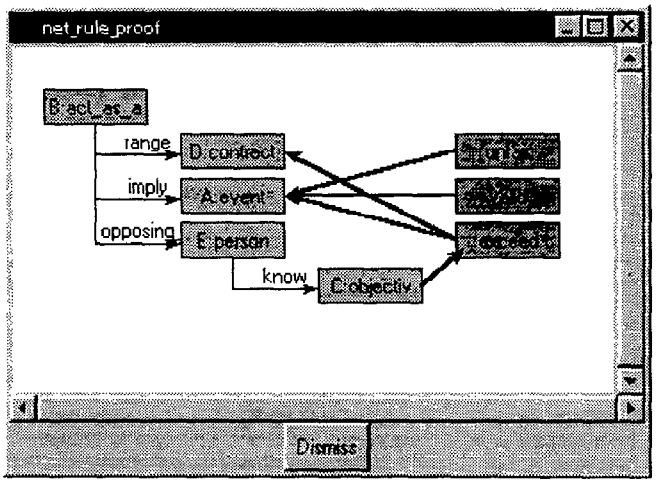
[Figure 6: The abstract structure calculated by GDA process]



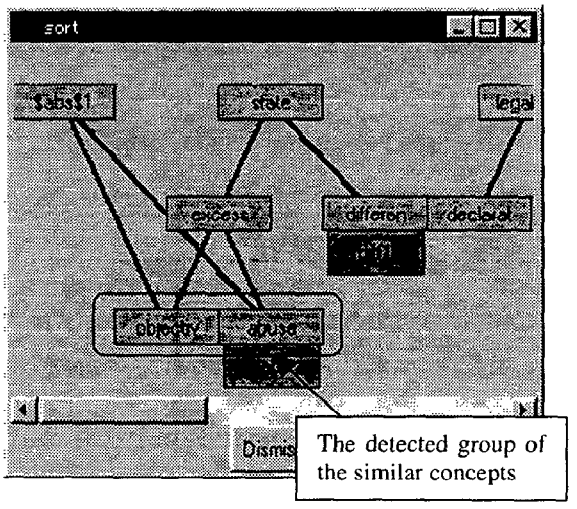
[Figure 7: The composed conceptual hierarchy]



[Figure 8: The generalized Art.93]



[Figure 9: The explanation to derive the purpose of Art. 113]



[Figure 10: The found hierarchy by the 2nd Doctrine]

APPENDIX A: SPECIFICATIONS OF KNOWLEDGE DESCRIPTIONS

Appendix A shows the specifications of our knowledge descriptions. For instance, the following descriptions are used in this article:

%%%%%%%%%% Example KB %%%%%%%%%%

%--- TKB ---

% Hierarchy

payment =< event. nopayment =< event. dissolution =< event.

dead =< event. person =<object. legal_person =< person.

natural_person =< person. contract_set=<action_set.

difference =< state. excess=<state. abuse =< excess.

objective_excess =< excess. act_as_agent =< legal_action.

declaration_of_intention =< legal_action.

registration =< quasi_legal_action.

% Type declarations

act1 @ act_as_agent. pm1 @ payment. npm1 @ nopayment.

diff1 @ difference. strB @ legal_person. cont1 @ contract_set.

manX @ natural_person. chiY @ natural_person. ab @ abuse.

% RFC

agent: declaration_of_intention -> person.

agent: act_as_agent -> person.

opposing_party: declaration_of_intention -> person.

opposing_party: act_as_agent -> person.

principal: act_as_agent -> person.

end: legal_person -> dissolution. end: natural_person -> dead.

%% ---- RKB -----

%% Articles in Japanese Civil Code

% Proviso clause, Article 93 (Mental Reservation)

```
rule::null(E:event):-
  oc(A:declaration_of_intention(
        real_intention->X:event,
        imply->E,
        opposing_party->Y:person,
        agent->P:person)),
  know(Y,D:difference),
  know(P,D),
  difference(D,X,E).
% Article 113 (Unauthorized Representation)
rule::null(E:event):-
  oc(A:act_as_agent(range->R:contract_set,
      imply->E,opposing_party->Y:person)),
  exceed(S:objective_excess,R,E),
  know(Y,S).
% Article 110 (Apparent Agency)
rule::valid(E:event):-
  oc(A:act_as_agent(range->R:action_set,
      imply->E,opposing_party->Y:person)),
  exceed(S:objective_excess,R,E),
  unknown(S,Y).
% Article 99 (Ordinary Agnecy)
rule::valid(E:event):-
  oc(A:act_as_agent(range->R:action_set,imply->E)),
  within(S:objective_excess,R,E).

%% Case : Apl. 20, 1967, Japanese Supreme Court
:-begin_fact.
object::act1:act_as_agent(
   principal->strB:legal_person(
   interest->npm1:nopayment),
   imply->pm1:payment,
   range->cont1:contract_set,
   opposing_party->manX:person,
   agent->chiY:person).
difference(diff1,npm1,pm1).
have(manX,r1).
know(manX,diff1).
know(chiY,diff1).
know(manX,ab).
exceed(ab,cont1,pm1).
:-end_fact.

%% Background Knowledge
%% a) in Section 3
unfair(E:event):-
  valid(E),
  oc(A:declaration_of_intention(real_intention->X:event,
      imply->E,
      opposing_party->Y:person(know->D:difference))),
  difference(D,X,E).
%% b) in Section 3
unfair(E:event):-
  valid(E),
  oc(A:act_as_agent(real_intention->X:event,
      imply->E,
      opposing_party->Y:person(know->D:difference))),
  difference(D,X,E).
%% c) in Section 3
unfair(E:event):-
  valid(E),
  oc(A:registration(real_intention->X:event,
      imply->E,
      opposing_party->Y:person(know->D:difference))),
  difference(D,X,E).
%% d) in Section 3
real_intention(A:act_as_agent(principal->X:person),E:event):-
  interest(X,E).
%% e) in Section 3
unfair(E:event):-
  valid(E),
  oc(A:act_as_agent(range->R:action_set,
        imply->E,
        opposing_party->P:person)),
  exceed(S:objective_excess,R,E),
  know(P,S).
%% f) in Section 3
unfair(E:event):-
  valid(E),
  oc(A:act_as_agent(range->R:action_set,
        imply->E,
        opposing_party->P:person)),
  exceed(S:abuse,R,E),
  know(P,S).

%% --- Additional Info.
cf([unfair/1]).  rev_info(null,valid).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Our descriptions consists of the following 3 parts:

1) Terminological Knowledge Base (**TKB**, for short)

2) Relational Knowledge Base (**RKB**, for short)

3) Additional information

A.1 TKB

Based on BNF notation, TKB syntax is defined as follows:

<TKB-description> ::= <Subclass-Rel> "." | <Type-Dec> "." | <RFC> "."

<Subclass-Rel> ::= <Sort> "=<" <Sort>

<Sort> ::= Prolog atom

<Type-Dec> ::= <Constant> "@" <Sort>

<Constant> ::= Prolog atom

<RFC> ::= <Role> ":" <Sort> "->" <Sort>

<Role> ::= Prolog atom ■

Role symbols are used as predicates in RKB. In this article, <Sort> is used as a concept symbol.

We assume I to be interpretation mapping, D to be a set of individuals (called a domain), C to be a set of constant symbols, R to be a set of role symbols, S to be a set of sort symbol (including special sort symbol "*"), "=<" to be a partial ordered relationship between sort symbols and (S,=<) to be sort hierarchy. Let $s,s1,s2 \in S$, $c,c1,c2 \in C$, $r \in R$. Then, I is defined as follows:

$I(*)=D$

$I(s) \subseteq D$

$I(c) \in D$

$I(s1=<s2) = I(s1) \subseteq I(s2)$

$I(c@s) = I(c) \in I(s)$

$I(r) : D^2 \mapsto \{TRUE,FALSE\}$

$I(r:s1->s2) = \forall x,y\ (x \in I(s1) \wedge I(r)(x,y) \rightarrow y \in I(s2))$ ■

"*" is called top sort, which is usually represented by "T". The intuitive meaning of <RFC> is a type restriction of a role value. For instance, when a father of person "p" is a person "f", the role symbol of father is represented by "father" and "father(p,f)" denotes the relation, RFC is used to restrict the type of "f" as "human". In object-oriented representations and frame based ones, the restrictions are regarded as declarations of slot value types.

A.2 RKB

Our RKB consists of order-sorted horn clauses (OS-clauses, for short) with our macro descriptions. Firstly, we show the definition of OS-clauses. Intuitively speaking, the clauses are ordinary horn clauses with type restrictions. For instance, a OS-clause "p(X:s1):-q(X,Y:s2) " denotes $p(X) \leftarrow q(X,Y) \wedge I(X) \in I(s1) \wedge I(Y) \in I(s2)$. The definition is shown as follows:

<OS-clause> ::= <Goal> ":-" <Goals> "." | <Goal> "."

<Goals> ::= <Goals> "," <Goal> | <Goal >

<Goal> ::= <Predicate> "(" <Terms> ")"

<Predicate> ::= Prolog atom

<Terms> ::= <Terms> "," <Term> | <Term>

<Term> ::= <Sorted-Variable> | <Variable> | <Constant>

<Sorted-Variable> ::= <Variable> ":" <Sort>

<Variable> ::= Prolog variable ■

":-" is used in text files in order to represent "←". Unsorted variables belong to top sort "*". Furthermore, a special predicate "oc" is installed in order to represent existences of individuals that denote relations, events and states.

To distinguish types of clauses, we use the following symbols:

•"rule::" is a tag to denote legal rules.

•":-begin_fact" is described at the beginning of clauses to denote facts.

•":-end_fact" is described at the end of clauses to denote facts.

The remaining clauses are regarded as user's background knowledge.

For user's convenience, we further introduce macro notations like object oriented representations [10]. The macro descriptions are defined as follows:

<Macro-Term> ::= <Variable> ":" <Sort> "(" <Attr-List> ")"

<Attr-List> ::= <Attr-List> "," <Attr> | <Attr>

<Attr> ::= <Role> "->" <Macro-Term>

<Macro-Fact> ::= "object::" <Ground-M-Term> "."

<Ground-M-Term> ::= <Constant> ":" <Sort> "(" <Ground-A-List> ")"

<Ground-A-List> ::= <Ground-A-List> "," <G-Attr> | <G-Attr>

<G-Attr> ::= <Role> "->" <Ground-M-Term> ■

Macro terms and facts are used to write object-centered terms. Macro facts are specially used to represent ground unit clauses as facts. The macro descriptions are expanded according to the following procedure:

1) Let Cl := the initial macro clause.

2) Get the leftmost macro term of Cl as $X:s(s(...,r_i ->M_i,...))$.

3) Rewrite $Cl(X:s(..., r_i ->M,...))$ into $Cl(X:s),..., r_i (X, M_i),....$

4) Go to 2).

A.3 Additional Information

A special unit clause " cf([...,p/N,...]) " denotes a list of predicates to represent legal purposes, where "p" denotes such a predicate and "N" denotes its number of arguments. Another special unit clause "rev_info(p1,p2) " denotes that a predicate "p1" is opposed to a predicate "p2". The later information is not essential for GDA framework. This is used to compensate the lack of notation of negation in our system.

APPENDIX B: FORMALIZATION OF GDA

In this appendix, we show the formalization of GDA under Order-Sorted Logic (For further detail of this, see [14,8]).

The symbols to denote concepts used in our knowledge descriptions are called "sorts". We assume S to be a set of sorts. Then, a similarity is defined as a mapping $\varphi: S \mapsto S'$, where S' is a set of sorts such that $S' \cap S = \phi$. The mapping is called abstraction [8,11,13]. By φ, let $a,b \in S$, then a~b is defined as $\varphi(a)=\varphi(b)$, where "a~b" denotes "a is similar to b". φ has one-to-one correspondence to a partition of S. To select appropriate φs among the possible φs, a criteria based on GDA has been proposed in [8]. Furthermore, φ^{-1} is defined as the following set when $s \in S$:

$\varphi^{-1}(s)=\{x|\varphi(x)=\varphi(s)\}$.

Semantically, we define that $I(\varphi(s))=\bigcup_{i\in\varphi^{-1}(s)} I(i)$. Then, $s'=<\alpha$ if s' is similar to s with respect to φ.

Secondly, assuming C to be a function-free order-sorted clause C, the abstraction of C, which is denoted by $\varphi(C)$, is defined as follows:

$\varphi(C(...,X:s,...))= C(...,X:\varphi(s),...)$,

where the arguments of C represent all sorted-variables occurring in C. That is, $\varphi(C)$ is a operation to rewrite all sorts occurring in C according to φ. Let $\varphi(C)=C'$, then C' is called the abstract clause of C and φ^{-1} is defined as follows:

$\varphi^{-1}(C')=\{x|\varphi(x)=C'\}$.

C is called a concrete clause of C'. The order-sorted deductions with abstract clauses use an order-sorted unification algorithm under the sort hierarchy (S',=<), which can be created based on the subsumption relationships that are defined as the inclusion relationships between extensions of the sorts belong to S', according to our semantics of each $\alpha\in S'$.

Now, to prepare the formalization of our GDA, we show the definition of "subsumption relationships used in a proof".

[Definition 1] (***Subsum$_G$***) Let T be a function-free Order-Sorted Theory, G be a goal deduced from T and the proof $P = ((G_0, C_0, \theta_0), ..., (G_{n-1}, C_{n-1}, \theta_{n-1}))$. Then, assuming $Subsum_G$ to be a set of subsumption relationships used in the proof P, $Subsum_G$ is defined as follows:

$Subsum_G = \{ [x\theta_i] =< [x] \mid x\leftarrow t \in \theta_i \}$,

where $0\leq i<n$, [t] denotes the sort of term t and for each step i of P, G_i is the goal clause, C_i is the input clause and θ_i is the substitution. ■

The purpose of our framework based on GDA is to obtain similarities that can share proofs of goals. Since we deal with similarities as abstractions in our approach, sharing the proofs is equivalent to preserving the proof structures even if the sorts occurring in the proof are rewritten by the abstractions. Therefore, to select abstraction mapping φ that satisfies this condition, the following criterion for appropriate similarities are defined.

[Definition 2] (*Appropriate Similarity*) Let φ be a abstraction mapping, T be a function-free Order-Sorted Theory, $\Sigma=(S,=<)$ be a sort hierarchy, Fact be a set of Order-Sorted unit clauses to represent facts in law, G be a Order-Sorted goal clause to represent unfair situations, $R=(A\leftarrow B)$ be a Order-Sorted clause to represent a candidate legal rule and $SortAbs_\varphi(T) = \{c' \mid \forall F\in\varphi^{-1}(C') T\cup\Sigma \vdash F\}$. Then, φ is an appropriate similarity with respect to G_R when the following conditions are satisfied:

[*Substitutability Condition* (GDA-Condition)]

$\varphi(Proof_G) \subseteq SortAbs_\varphi(T)$,

where assuming I to be a set of input clauses in the proof of G,

$Proof_G$= I-Fact-{c∈T| c is used in the proof in order to derive an atom whose predicate p occurs in R}.

[*Subsumption Preservingness Condition* (**SPC**, for short)]

Let $Subsum_G$ be a set of subsumption relationships used in the proof of G. Then,

$\varphi(Subsum_G) \subseteq SubsumAbs_\varphi(\Sigma)$,

where $SubsumAbs_\varphi(\Sigma) = \{a'=<b' \mid \forall A\in\varphi^{-1}(a')\ \exists B\in\varphi^{-1}(b')\ \Sigma \models A=<B\}$. ■

In [8], we used Similarity Inheritance Condition instead of SPC. In this article, to make order-sorted GDA and original GDA semantically equivalent, we adopt SPC. Since SIC is more strict condition than SPC, it can contribute for reducing the search spaces. However, it is independent of goals and the proofs. Thus, our implementation is designed to allow users to choose whichever they like.

Furthermore, in order to preserves RFC (see APPENDIX A), the following criteria is defined (For further detail of this criteria, see [7]):

[Definition 3] (*VRP*) Let G be a goal, $\Sigma=(S,=<)$ be a sort hierarchy, VR be a set of value restrictions and R(G) be a set of roles used in the proof of G.

A similarity φ is said to satisfy Value Restriction Preservingness (**VRP**, for short) iff for any sorts $\mathbf{s_1},\mathbf{s_2}\in S$,

if $\varphi(\mathbf{s_1}) = \varphi(\mathbf{s_2})$, then for each $\mathbf{r}\in R(G)$,

1. $\exists \mathbf{s_1}'\in mlb_\Sigma(filler(\mathbf{r},\mathbf{s_1})), \exists \mathbf{s_2}'\in mlb_\Sigma(filler(\mathbf{r},\mathbf{s_2}))$ such that $\varphi(\mathbf{s_1}') = \varphi(\mathbf{s_2}')$

or

2. $filler(\mathbf{r},\mathbf{s_1}) = filler(\mathbf{r},\mathbf{s_2}) = \phi$,

where $filler(\mathbf{r},\mathbf{s_i}) = \{\mathbf{s}' \mid \mathbf{s_i} =< \mathbf{s}^*$ and $\mathbf{r}:\mathbf{s}^*\text{->}\mathbf{s}'\in VR\}$

and $mlb_\Sigma(E)$=

{s'∈S | s' is a maximal lower bound of E under =<}. ■

In our system, this criterion is available as an optional condition to reduce the search space.