Translating Case-Based Reasoning into Abductive Logic Programming

Ken Satoh[1]

Abstract. This paper presents a translation method of case-based reasoning which changes similarity according to context into abductive logic programming based on a generalized stable model semantics [5]. This kind of dynamic similarity can be found in CBR systems for legal reasoning such as HYPO [1]. Abductive logic programming is suitable to implement this dynamic similarity by regarding abducible predicates as similarity and changing abducible predicates by context. In this paper, we define a relevance criteria of cases for defendant and plaintiff and show how to change similarity between cases according to its context (defendant vs plaintiff, the current case and the cited case). We show correspondence between properties in a case used for similarity and abducibles used in a translated abductive logic program and show that we can construct an argument by using abducibles which explains why the current case is similar to the cited case and the current case is not similar to every counter case.

1 INTRODUCTION

Ashley[1, Page 2] pointed out the following characteristics in legal reasoning which are obstacles in constructing a legal system by computer.

1. Legal rules tend to be incomplete, logically and semantically ambiguous, sometimes inconsistent, and frequently hard to find or even to identify.
2. The law is adversarial; legal problems frequently have no one right answer. Usually there are at least two opposing viewpoints, and the arguments on either side may be quite two opposing.
3. Legal reasoning involves reasoning by analogy. Lawyers argue by drawing analogies to past cases and posing hypotheticals.

Then, Ashley proposed an approach of *adversarial case-based reasoning* which involves dynamic analogy to similar past cases dependent on the viewpoint. Although legal reasoning is not deductive, there is some underlying reasoning mechanism. Thus, Ashley proposed a logical representation for relevance criteria [2] in adversarial case-based reasoning which is a logical specification of the reasoning mechanism in HYPO [1].

In this paper, we propose another logical formalism of adversarial case-based reasoning by using abductive logic programming. Firstly, we propose a partial matching mechanism of the current case with a previous case. This matching mechanism can change similarity according to context (for defendant or for plaintiff, the current case and the cited case). Secondly, in order to implement dynamic nature of similarity, we use abducible predicates in abductive logic programming. Abducible predicates are abduced dynamically according to a goal which should be achieved and therefore, they are suitable to express dynamic aspect of similarity.

Our idea of the dynamic similarity is as follows. A case is regarded as a set of properties satisfied in the case. We assume that there are two kinds of cases, OK cases and NG cases which correspond to cases favorable with a plaintiff and a defendant respectively. Suppose that we would like to show that the current case is favorable with a plaintiff. Then, we try to divide properties into two categories one of which can be ignored to ascribe the current case to a OK case and the other of which cannot be ignored to distinguish the current case with every NG case.

To realize the above idea, we translate case-based reasoning into abductive logic programming as follows. We translate a case into a clause in an abductive logic program. Our translation is *local* in the sense that a case is directly changed into a clause without any reference to the other cases. Every properties is translated into an abducible so that it can be either important or unimportant for similarity. To express a position(defendant or plaintiff), we use a goal to prove NG/OK. Then, abducibles will be changed according to the goal. The semantics of abductive logic programming is based on generalized stable model [5] which can express a dynamic aspect of abducibles. We use our previous procedure [8] based on the semantics which is an enhanced procedure of [3] with bottom-up checking of integrity constraints. The procedure automatically computes the above division of properties in order to retrieve a favorable case and distinguish all counter cases and provides an explanation why the current case is similar to the cited case and the current case is not similar to every counter case.

An application of generalized stable models of abductive logic programming to analogical reasoning is not new [6, 4]. However, in these works, analogical predicates should be specified before analogical reasoning is performed. So analogical predicates are static. On the other hand, we propose a dynamic selection of analogical predicates according to the context.

2 FORMALIZATION

We begin with definitions of a case and relevance criteria between cases. The following representation of a case by a set

[1] Division of Electronics and Information Engineering, Hokkaido University, N13W8 Kita-ku Sapporo 060 Japan, email: ksatoh@db.huee.hokudai.ac.jp

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

of properties[2] and the representation of relevant similarities by the shared properties are similar to the representations in HYPO.

Definition 1 *Let $\mathcal{P}$ be a finite set called* a set of properties. *We call any subset of $\mathcal{P}$ is called a* case. *Let $\mathcal{OK}$ be a set of cases called* OK cases *and $\mathcal{NG}$ be a set of cases called* NG cases *such that $\mathcal{OK} \cap \mathcal{NG} = \emptyset$. Let CB be $\mathcal{OK} \cup \mathcal{NG}$ called* a casebase.
Let C be a case. Then, we say that C is OK *if there exists a case $C_{ok} \in \mathcal{OK}$ and a subset of $\mathcal{P}$, I_{ok} such that*
- $C \cap I_{ok} = C_{ok} \cap I_{ok}$ *and*
- *for every case $C_{ng} \in \mathcal{NG}$, $C \cap I_{ok} \neq C_{ng} \cap I_{ok}$.*

We call C_{ok} a supporting case to prove OK for C *and I_{ok}* a set of important properties to prove OK for C.

We also say that C is NG *if there exists a case $C_{ng} \in \mathcal{NG}$ and a subset of $\mathcal{P}$, I_{ng} such that*
- $C \cap I_{ng} = C_{ng} \cap I_{ng}$ *and*
- *for every case $C_{ok} \in \mathcal{OK}$, $C \cap I_{ng} \neq C_{ok} \cap I_{ng}$.*

We call C_{ng} a supporting case to prove NG for C *and I_{ng}* a set of important properties to prove NG for C.

Note that conditions for OK and conditions for NG are symmetrical.

Example 1 *Let $\mathcal{P} = \{p1, p2, p3, p4\}$, $\mathcal{OK} = \{c0, c1\}$, $\mathcal{NG} = \{c2, c3\}$, $c0 = \{p1, p2\}$, $c1 = \{p1, p3\}$, $c2 = \{p2, p3\}$, and $c3 = \{p1, p4\}$.*
Suppose that the current case c is $\{p1, p2, p4\}$. Let $I_1 = \{p2, p3\}$. Then, $c0 \cap I_1 = \{p2\} = c \cap I_1$ and for every case in $\mathcal{NG}$, $c2 \cap I_1 = \{p2, p3\} \neq c \cap I_1$ and $c3 \cap I_1 = \emptyset \neq c \cap I_1$. Thus, there exists a supporting case $c0$ and a set of important properties I_1 to prove OK for c. Therefore, c is OK. Note that there are the other two sets of important properties corresponding with $c0$, that is, $\{p1, p2, p3\}$ and $\{p1, p2\}$, and $c1$ cannot be a supporting case because there is no set of important properties corresponding with $c1$.

If the status of important properties of the current case coincides with the status of the properties of an OK(NG) case, then the current case is supposed to be similar to the OK(NG, respectively) case by ignoring unimportant other properties, and vice versa. So, in order to prove that the current case is OK(NG, respectively), we try to find out a supporting case which is similar to the current case and distinguish NG(OK, respectively) cases. Therefore, even if a casebase is fixed, important properties are relative to the context; the current case, a supporting case which we would like to cite, and the purpose(proving OK or NG). We believe that the above definition reflects some characteristics of CBR for a legal reasoning system like HYPO.

The following example shows that if the current case or the supporting case is changed, a set of important properties is varied.

Example 2 *Let $\mathcal{P}$, $\mathcal{OK}$ and $\mathcal{NG}$ be the same as Example 1. Suppose that the current case c is $\{p1, p2, p3\}$. I_1 in Example 1 is no longer a set of important properties because $c0 \cap I_1 = \{p2\} \neq c \cap I_1$. Let $I_2 = \{p1, p4\}$. Then, $c0 \cap I_2 = \{p1\} = c \cap I_2$ and for every case in $\mathcal{NG}$, $c2 \cap I_2 = \emptyset \neq c \cap I_2$ and $c3 \cap I_2 = \{p1, p4\} \neq c \cap I_2$. Thus, there exists a supporting case $c0$ and a set of important properties I_2 to prove OK for c. Therefore, c is OK. There are the other two sets of important properties corresponding with $c0$, that is, $\{p1, p2, p4\}$ and $\{p1, p2\}$, and there are three sets of important properties corresponding with $c1$, that is, $\{p1, p3, p4\}$, $\{p1, p3\}$ and $\{p1, p4\}$.*

The following example shows that if the purpose is changed, a set of important properties is changed. We believe that this reflects the behavior according to a position (defendant or plaintiff).

Example 3 *Let $\mathcal{P}$, $\mathcal{OK}$ and $\mathcal{NG}$ and the current case c be the same as Example 1. Let $I_3 = \{p3, p4\}$. Then, $c3 \cap I_3 = \{p4\} = c \cap I_3$ and for every case in $\mathcal{OK}$, $c0 \cap I_3 = \emptyset \neq c \cap I_3$ and $c1 \cap I_3 = \{p3\} \neq c \cap I_3$. Thus, there exists a supporting case $c3$ and a set of important properties I_3 to prove NG for c. Therefore, c is NG. There are the other two sets of important properties corresponding with $c3$, that is, $\{p1, p3, p4\}$ and $\{p1, p4\}$ and there is no set of important properties corresponding with $c2$.*

3 TRANSLATION

Now, we give a translation of CBR into abductive logic programming. The translation consists of three parts; translation of a casebase, translation of the current case, and translation of the purpose.

Definition 2 Basic translation *of a case C in $\mathcal{CB}$ is as follows. Let $p_1, ..., p_k$ be properties in C and $n_1, ..., n_m$ be properties not in C.*
If $C \in \mathcal{OK}$,
$ok(id(C)) \leftarrow f(p_1), ..., f(p_k), nf(n_1), ..., nf(n_m)$,
and if $C \in \mathcal{NG}$,
$ng(id(C)) \leftarrow f(p_1), ..., f(p_k), nf(n_1), ..., nf(n_m)$,
where $id(C)$ denotes an identification symbol for a case C. We also add the following rules.
$f(P) \leftarrow cf(P)$
$f(P) \leftarrow \sim imp^*(P)$
$nf(P) \leftarrow ncf(P)$
$nf(P) \leftarrow \sim imp^*(P)$
where cf and ncf will be defined below and P is a variable and $\sim$ expresses negation as failure *and imp^* is a abducible predicate. We denote the above program T_{basic}.*

Example 4 *Let $\mathcal{P}$, $\mathcal{OK}$ and $\mathcal{NG}$ be the same as Example 1. Then, the basic translation for the casebase is as follows:*
$ok(c0) \leftarrow f(p1), f(p2), nf(p3), nf(p4)$
$ok(c1) \leftarrow f(p1), nf(p2), f(p3), nf(p4)$
$ng(c2) \leftarrow nf(p1), f(p2), f(p3), nf(p4)$
$ng(c3) \leftarrow f(p1), nf(p2), nf(p3), f(p4)$
$f(P) \leftarrow cf(P)$
$f(P) \leftarrow \sim imp^*(P)$
$nf(P) \leftarrow ncf(P)$
$nf(P) \leftarrow \sim imp^*(P)$

The idea of the above translation is simple. Firstly, every case is translated into a rule whose conclusion is a consequence of the case and whose body consists of the status of the properties in the case. If the current case coincides with a case in the important properties, the current case is supposed to have the same consequence as the matched case. This matching is done by using $imp^*(p)$ which expresses that p is an important property. Suppose that a property is not important. This means that $\sim imp^*(p)$ is true. Then, $f(p)$ and $nf(p)$ become automatically true. This corresponds with the

[2] Properties in this paper corresponds with dimensions in HYPO.

behavior that the property p is ignored to match the current case and a case in a casebase. On the other hand, suppose that a property is important. This means that $\sim imp^*(p)$ is false. Then, $f(p) \leftarrow cf(p)$ and $nf(p) \leftarrow ncf(p)$ are the only rules for $f(p)$ and $nf(p)$. Since $cf(p)$ and $ncf(p)$ represent the status of properties in the current case (see below), this corresponds with partial matching such that the status of the important properties in the current case and the supporting case must coincide.

Definition 3 Translation for the current case C *is as follows. Let* $p_1, ..., p_k$ *be properties in* C *and* $n_1, ..., n_m$ *be properties not in* C.

$$cf(p_1) \leftarrow$$
$$\vdots$$
$$cf(p_k) \leftarrow$$
$$ncf(n_1) \leftarrow$$
$$\vdots$$
$$ncf(n_m) \leftarrow$$

We denote the above program as $T_{current}$

Example 5 *Let the current case c be the same as Example 1. The translation for c is the following program:*

$cf(p1) \leftarrow$
$cf(p2) \leftarrow$
$ncf(p3) \leftarrow$
$cf(p4) \leftarrow$

The following definition provide a translation of the purpose of CBR.

Definition 4 *To prove that c is OK, we add the following integrity constraint:*

$\perp \leftarrow ng(X)$

to $T_{basic} \cup T_{current}$ *(denoted as* T_{ok}*), and ask* $? - ok(C)$, *where* $\perp$ *means contradiction and* X *is a variable.*

When we would like to prove that the current case is OK, we add an integrity constraint expressing that $ng(X)$ must not be derived, that is, that every derivation of $ng(X)$ must fail. Symmetrical treatment is made in order to prove that the current case is NG.

Definition 5 *To prove that c is NG, we add the following integrity constraint:*

$\perp \leftarrow ok(X)$

to $T_{basic} \cup T_{current}$ *(denoted as* T_{ng}*), and ask* $? - ng(C)$.

Semantics of the above translated programs is based on generalized stable model [5] as follows.

Definition 6 *Let* T *be a logic program and* Π_T *be a set of ground rules obtained by replacing all variables in each rule in* T *by every element of its Herbrand universe. Let* M *be a set of ground atoms from* Π_T *and* Π_T^M *be the following program.*

$$\Pi_T^M = \{H \leftarrow B_1, ..., B_k | \; H \leftarrow B_1, ..., B_k, \sim A_1, ..., \sim A_m \in \Pi_T \text{ and } A_i \notin M \text{ for each } i = 1, ..., m.\}$$

Let $min(\Pi_T^M)$ *be the least model of* Π_T^M. *A* stable model *for a logic program* T *is* M *iff* $M = min(\Pi_T^M)$ *and* $\perp \notin M$.

Definition 7 *Let* T *be a logic program with abducible predicates and* Θ *be a set of ground instances of abducible predicates (called* abducibles*). A* generalized stable model $M(\Theta)$ *for* T *is a stable model of* $T(\Theta)$ *where* $T(\Theta) = T \cup \{H \leftarrow | H \in \Theta\}$.

The following theorem states that if there is a model for a translated program, then the abducibles in the model correspond with important properties to prove OK for the current case.

Theorem 1 *Suppose that there exists a generalized stable model* $M(\Theta)$ *for* T_{ok} *and a symbol* c_{ok} *such that* $M(\Theta) \models ok(c_{ok})$. *Then,* $C_{ok} = id^{-1}(c_{ok})$ *is a supporting case and* $\{p | imp^*(p) \in \Theta\}$ *is a set of important properties to prove OK for the current case.*

The following theorem states that if there are important properties to prove OK for the current case, then there is a model which contains abducibles corresponding with the important properties.

Theorem 2 *Suppose that* C_{ok} *is a supporting case and* I *is a set of important properties to prove OK for the current case. Then, there exists a generalized stable model* $M(\Theta)$ *for* T_{ok} *such that* $M(\Theta) \models ok(id(C_{ok}))$ *and* $\Theta = \{imp^*(p) | p \in I\}$.

Proofs are found in Appendix. Note that the similar theorems hold for proving NG.

Example 6 *Let* T_{ok} *be a union of programs in Example 4 and Example 5. Let* $\Theta = \{imp^*(p2), imp^*(p3)\}$ *and*

$M(\Theta) = \Theta \cup$
$\{ok(c0), cf(p1), cf(p2), ncf(p3), cf(p4),$
$f(p1), nf(p1), f(p2), nf(p3), f(p4), nf(p4)\}$.

Then, $min(\Pi_{T_{ok}(\Theta)}^{M(\Theta)}) =$
$\{f(p1), f(p4), nf(p1), nf(p4),$
$cf(p1), cf(p2), ncf(p3), cf(p4), imp^*(p2), imp^*(p3),$
$f(p2), nf(p3), ok(c0)\} = M(\Theta)$

Therefore, $M(\Theta)$ *is a generalized stable model for* T_{ok}. *This model corresponds with* I_1 *in Example 1.*

We can also show that the following are two models for T_{ok}:

$\{f(p4), nf(p4), cf(p1), cf(p2), ncf(p3), cf(p4),$
$imp^*(p1), imp^*(p2), imp^*(p3), f(p1), f(p2), nf(p3), ok(c0)\}$

$\{f(p3), nf(p3), f(p4), nf(p4), cf(p1), cf(p2), ncf(p3), cf(p4),$
$imp^*(p1), imp^*(p2), f(p1), f(p2), ok(c0)\}$

which correspond with other sets of important properties.

4 EXPLANATION

We use a proof procedure for the above translated programs which is an extension of goal-directed proof procedure of [8] with an interface to produce an explanation. We explain briefly how the procedure works for $? - ok(X)$.

Firstly, the procedure searches a rule for the initial query $? - ok(X)$. Then, the procedure tries to satisfy the body of the rule. Recursively, the procedure tries to find a rule for $f(p_i)$ and $nf(n_i)$ in the body. $f(p_i)$ is satisfied by finding $cf(p_i)$ or assuming $\sim imp^*(p_i)$, and $nf(n_i)$ is satisfied by finding $ncf(n_i)$ or assuming $\sim imp^*(p_i)$.

Then, the procedure checks that relevant NG cases cannot be proved to be similar to the current case. A rule for relevant NG case has some $f(p_i)$ and $nf(n_i)$ in its body which also exist in the body of the rule of $ok(X)$. To show that relevant NG cases cannot be proved, the procedure tries to show that there exists some $f(p_i)$ or $nf(n_i)$ which are not satisfied. This is done by showing that there is no $cf(p_i)$ and assuming

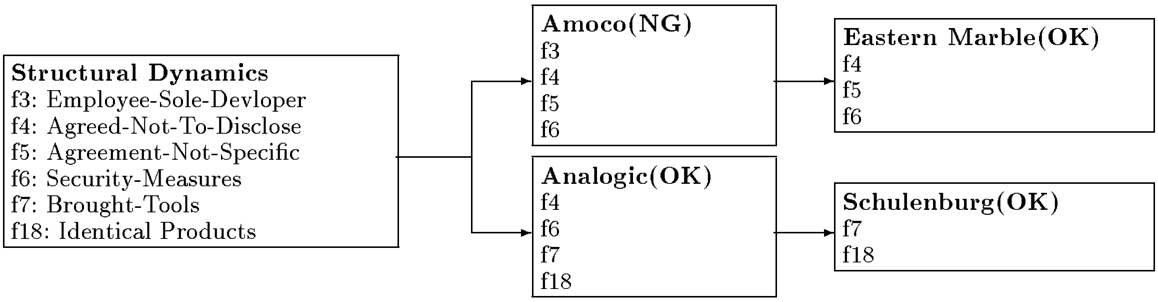

Figure 1. Claim Lattice in HYPO

$imp^*(p_i)$ or by showing that there is no $ncf(n_i)$ and assuming $imp^*(n_i)$. During the execution, there are selections of rules and abducibles. So, if there exists a NG case which the procedure cannot show to be distinguished from the current case in a sequence of selections, the execution fails and the procedure exploits alternative selection by backtracking.

The procedure accumulates abducibles and other derived atoms along the execution and then, produces an explanation by accumulated abducibles and atoms. Since there is no unique "right answer" in legal domain, our intention is that we show every possible explanation to the user and the user will decide which one is the most preferable.

In this paper, we take examples used in [2] from the domain of trade secrets law. The cases are shown in Figure 1. Refer to [1] for the meaning of properties in a case. We assume that cases except the *Structural Dynamics* case are in a case base and we show that the *Structural Dynamics* case can be either OK or NG.

The following is an explanation generated by the proof procedure for the *Structural Dynamics* case to prove OK.

```
The current case is OK since
  the current case is similar to
    OK case analogic since
      we believe that
        properties f4,f7 are important
      and the above properties are shared
then
  the current case is related with
    NG case amoco,
  but can be distinguished since
    the current case is different
       in properties f7
    and the difference is important
in this interpretation,
  the current case is distinguished from
    OK case eastern_marble since
      the current case is different
         in properties f7
and,
  the current case is distinguished from
    OK case schulenburg since
      the current case is different
         in properties f4
in this interpretation, properties f6,f18
  can be either important or unimportant
```

In the explanation, although status of properties $f3$ and $f5$ in the *Analogic* case is different from the status in the current case, the difference is ignored since we assume that the only properties $f4$ and $f7$ are important. There is one counter case, the *Amoco* case , but it is supposed to be different because the status of the important property $f7$ in the *Amoco*case is different from the status in the current case. Unfortunately, however, by this interpretation, the *Eastern Marble* case cannot be a supporting case because the status of important property $f7$ in the *Easter Marble* case is different from the one in the current case. Similarly, the *Schulenburg* case cannot be a supporting case either. The last two lines show that the properties $f6$ and $f18$ can be added to the important properties.

The following is an explanation generated by the proof procedure for the *Structural Dynamics* case to prove NG.

```
The current case is NG since
  the current case is similar to
    NG case amoco since
      we believe that
        properties f3 are important
      and the above properties are shared
then
  the current case is related with
    OK case schulenburg,
  but can be distinguished since
    the current case is different
       in properties f3
    and the difference is important
and,
  the current case is related with
    OK case eastern_marble,
  but can be distinguished since
    the current case is different
       in properties f3
    and the difference is important
and,
  the current case is related with
    OK case analogic,
  but can be distinguished since
    the current case is different
       in properties f3
    and the difference is important
in this interpretation, properties f4,f5,f6
  can be either important or unimportant
```

In the explanation, although the status of properties $f7$ and $f18$ in the *Amoco* case is different from the status in the current case, the difference is ignored since we assume that the only property $f3$ is important. There are three counter cases, the *Eastern Marble* case, the *Analogic* case and the *Schulenburg* case, but they are supposed to be different because the status of the important property $f3$ in these cases is different from the status in the current case.

5 CONCLUSION

We believe that the following are contributions of this paper.

- We provide a translation method of case-based reasoning into abductive logic programming and show one to one correspondence of important properties which make the current case OK/NG and abducibles in abductive logic programming language.
- We show that we can produce explanation for analogical interpretation of cases by using abducibles.

We need the following future work.

- We would like to combine rule-based reasoning and case-based reasoning as CABARET [7] does. It should be easy to combine these reasonings in abductive logic programming because a clause can be regarded as a rule. A straightforward extension is to introduce integrity constraints which exclude some combinations for important properties. For example, if we add $\bot \leftarrow f(p1), f(p2)$ to the program T_{ok} in Example 6, it prohibits models which make $f(p1)$ and $f(p2)$ simultaneously true. In Example 6, we can choose the unique generalized stable model whose abducibles correspond with important properties $\{p2, p3\}$.
- We would like to extend properties with some value ranges so that quantitative properties can be expressed. Although we only consider atomic property in this paper, we believe that our approach can be extended to first-order property by introducing a quantitative predicate such as inequality and representing definition of the predicate in clauses.
- We would like to make a theoretical relationship between HYPO and our abductive reasoning. In HYPO, they use a more-on-point comparison which prefers a case having more shared properties with the current case in terms of set inclusion. To formalize this, we might need a representation of preferences between a set of abducibles.

APPENDIX: PROOFS

Proof of Theorem 1:
Let the rule for C_{ok} be

$ok(c_{ok}) \leftarrow f(p_1), ..., f(p_k), nf(n_1), ..., nf(n_m)$

Since there is no other rule for C_{ok}, this rule should be satisfied. Therefore, for each $f(p_i)$, $M(\Theta) \models f(p_i)$ and for each $nf(n_i)$, $M(\Theta) \models nf(n_i)$. $M(\Theta) \models f(p_i)$ means that either $M(\Theta) \models cf(p_i)$ or $M(\Theta) \not\models imp^*(p_i)$. Then, the following hold:

- $f(p_i)$ in the body of the rule of $ok(c_{ok})$ means $p_i \in C_{ok}$.
- $M(\Theta) \models cf(p_i)$ means $p_i \in C$ where C is the current case.
- $M(\Theta) \not\models imp^*(p_i)$ means $p_i \notin I$ where $I = \{p | imp^*(p) \in \Theta\}$.

Thus, $p_i \in C_{ok}$ and $p_i \in I$ implies $p_i \in C$. This means that $C_{ok} \cap I \subseteq C$, thus $C_{ok} \cap I \subseteq C \cap I$. In a symmetrical way, $M(\Theta) \models nf(p_i)$ leads to $C \cap I \subseteq C_{ok} \cap I$. Therefore, $C \cap I = C_{ok} \cap I$.

Since $M(\Theta)$ satisfies every integrity constraint, for every $C_{ng} \in NG$, $M(\Theta) \not\models ng(c_{ng})$ where $c_{ng} = id(C_{ng})$. Let the rule for C_{ng} be

$ng(c_{ng}) \leftarrow f(p_1), ..., f(p_k), nf(n_1), ..., nf(n_m)$

Then, there exists $f(p_i)$ such that $M(\Theta) \not\models f(p_i)$ or there exists $nf(n_i)$ such that $M(\Theta) \not\models nf(n_i)$. Suppose that there exists $f(p_i)$ such that $M(\Theta) \not\models f(p_i)$. This means that $M(\Theta) \not\models cf(p_i)$ and $M(\Theta) \models imp^*(p_i)$. Therefore, $p_i \in C_{ng}$ and $p_i \in I$ but $p_i \notin C$. This means $C \cap I \neq C_{ng} \cap I$. On the other hand, suppose that there exists $nf(n_i)$ such that $M(\Theta) \not\models nf(n_i)$. In a similar way, we can show that $C \cap I \neq C_{ng} \cap I$. In either case, $C \cap I \neq C_{ng} \cap I$.

Therefore, C_{ok} is a supporting case and I is a set of important properties to prove OK for the current case C. □

Proof of Theorem 2:
Let $T' = T_{ok}$

$- \{f(P) \leftarrow \sim imp^*(P) \text{ and } nf(P) \leftarrow \sim imp^*(P)\}$
$\cup \{f(p) \leftarrow | p \notin I\} \cup \{nf(p) \leftarrow | p \notin I\}$

and $M(\Theta) = min(\Pi_{T'(\Theta)})$.

We first show that $M(\Theta) \models ok(id(C_{ok}))$. Suppose that $M(\Theta) \not\models ok(id(C_{ok}))$. There is only one rule for $ok(id(C_{ok}))$ denoted as:

$ok(c_{ok}) \leftarrow f(p_1), ..., f(p_k), nf(n_1), ..., nf(n_m)$

where $c_{ok} = id(C_{ok})$. Then, since $M(\Theta) \not\models ok(c_{ok})$, there exists $f(p_i)$ such that $M(\Theta) \not\models f(p_i)$ or there exists $nf(n_i)$ such that $M(\Theta) \not\models nf(n_i)$.

In either case, $C \cap I \neq C_{ok} \cap I$ and this contradicts with the condition of I. Therefore, $M(\Theta) \models ok(c_{ok})$. Similarly, we can show that $M(\Theta) \not\models \bot$ and since $\Pi^{M(\Theta)}_{T_{ok}(\Theta)} = \Pi_{T'(\Theta)}$,

$min(\Pi^{M(\Theta)}_{T_{ok}(\Theta)}) = min(\Pi_{T'(\Theta)}) = M(\Theta)$

Therefore, $M(\Theta)$ is a generalized stable model of T_{ok}. □

Acknowledgements: We thank the referees for comments to improve this paper.

REFERENCES

[1] Ashley, K. D., *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*, MIT press (1990).

[2] Ashley, K. D., and Aleven, V., "A Logical Representation for Relevance Criteria", S. Wess, K-D. Althoff and M. Richter (eds.), *Topics in Case-Based Reasoning, LNAI 837*, 338 – 352 (1994).

[3] Eshghi, K. and Kowalski, R., "Abduction Compared with Negation by Failure", *Proceedings of ICLP'89*, 234 – 254 (1989).

[4] Iwayama, N., Satoh, K. and Arima, J., "A Formalization of Generalization-based Analogy in General Logic Programs", *Proceedings of ECAI-92*, 129 – 133 (1992).

[5] Kakas, A. C., Mancarella, P., "Generalized Stable Models: A Semantics for Abduction", *Proc. of ECAI-90*, 385 – 391 (1990).

[6] Orihara, R., "Analogical Reasoning as a Form of Hypothetical Reasoning and Justification-based Knowledge Acquisition", *Proc. of the Second Workshop on Algorithmic Learning Theory*, 243 – 254 (1991).

[7] Rissland, E. L,, and Skalak, D. B., "CABARET: Statutory Interpretation in a Hybrid Architecture", *International Journal of Man-Machine Studies*, **34**(6), 839 – 887 (1991).

[8] Satoh, K. and Iwayama, N., "A Query Evaluation Method for Abductive Logic Programming", *Proceedings of JICSLP'92*, 671 – 685 (1992).