Using Two Level Abduction to Decide Similarity of Cases

Ken Satoh[1]

Abstract. In [7], we formalized *adversarial case-based reasoning* implemented in HYPO [1, 2] where similarity of the current case and the precedent cases is changed according to one's position such as defendant and plaintiff, and we also proposed its implementation by translating the formalism into abductive logic programming.

In this paper, we enhance representation power of cases; In [7] we only consider propositional factors which take the truth value whereas in this paper we consider factors which can take other kind of value such as numerical or symbolic one.

To do so, we need a mechanism to match two different values for a factor. We introduce two kinds of abducibles in abductive logic programming; one kind expresses importance of the factors and the other expresses similarity of the values of factors. This reasoning can be regarded as two-level abduction since in the first level, we determine a set of important factors which contributes to comparison of cases and in the second level, we try to abduce the similarity of different values for soft matching.

According to the correspondence between the formalism and semantics of a translated program, we can guarantee that every possible important factors and every soft matching can be exactly computed.

1 INTRODUCTION

We often encounter the situation in which we need to argue against an opponent whose purpose conflicts with our desire. For example, suppose that we have a report at school and because of some reason, we are late for the deadline of the report. We try to persuade the teacher to accept the report even if the deadline was over.

If we know some precedent cases where the same or similar purpose was achieved, then we cite these cases to persuade an opponent. This way of conflict resolution is called *adversarial case-based reasoning* [1, 2] and originated from American and English legal reasoning where a law decision is made by citing precedent cases. The characteristics of this reasoning is as follows.

- Similarity between cases dynamically changes according to one's position and different previous favorable cases to the position are cited according to this dynamic similarity.
- We need to find an unrebuttable favorable case so that the unfavorable cases are not cited by the same reason for citing case.

For example, in the above delayed-report situation, if we know some previous cases in which the teacher accepted the report even if the deadline was over, then we try to persuade the teacher for the previous case to be similar to the current case by focusing some similar factors of both cases. But if the similarity used for citing the previous case is also applicable to another previous case in which the teacher did not accept the report, then the teacher might cite the unfavorable case to reject the report.

In the previous paper [7], we give a definition of dynamic similarity over propositional factors which take the truth value. However in the real situation, there are many factors which take the other kind of value such as numerical value or symbolic value. If we exactly match these values, we could not cite many precedent cases, so we need to soften the matching so that some range of value or some set of values can be similar.

In this paper, we propose a formalism for this enhanced reasoning which handles matching of the different values in a factor and we give a translation of the formalism into abductive logic programming. The novel technique of the translation in this paper is the following.

1. We introduce two kind of abducibles; one kind expresses importance of the factors and the other expresses similarity between the values of the factors.
2. We introduce a set of integrity constraints over similarity abducibles which only allows similarity abduction between the different values which is consistent with these integrity constraints.

The semantics of abductive logic programming is based on generalized stable model [4] reflecting a dynamic aspect of abducibles according to goals. We translate a casebase into an abductive logic program and the position into its goal and show a correspondence between two kinds of abducibles in a generalized stable model for a translated program and important factors and similarity over the values.

2 DEFINITION

We begin with definitions of cases and relevance criteria between cases.

Definition 1 *Let $D_1, ..., D_n$ be domains of the values. Let $v_i (1 \leq i \leq n)$ be a value of the domain D_i. We call a sequence*

[1] Division of Electronics and Information Engineering, Hokkaido University, N13W8 Kita-ku Sapporo 060-8628 Japan, email: ksatoh@db-ei.eng.hokudai.ac.jp

ECAI 98. *13th European Conference on Artificial Intelligence*
Edited by Henri Prade
Published in 1998 by John Wiley & Sons, Ltd.

$\langle v_1, ..., v_n \rangle$ *a* case. *We define a function* $f_i (1 \leq i \leq n)$ *from a case to the* i*-th value,* v_i*, of the case. We call* f_i *a factor and write a set of all factors as* $\mathcal{F}$*. We use id as a function from a case to an identifier for the case.*

In the above definition, boolean-valued factors are included since these factors have a domain $\{true, false\}$. Therefore, this definition properly subsumes the definition of the previous paper [7].

We denote an equivalence relation over a domain D for a factor f as $\sim_f$. This equivalence relation performs soft matching; if $a \sim_f b$ then a and b are assumed to be similar. The equivalence relation is dependent on a factor f because soft matching will be made for each factor and soft matching may be different for two factors even if the domain of the two factors are identical.

We assume that there is a set of constraints EC_f (including reflexive, symmetrical and transitive property) for $\sim_f$. For example, if we would like to exactly match the values of D such as boolean value matching, then we add the following constraints EC_f:

for every $a, b \in D$, $a \sim_f b$ iff $a =_D b$,

where $=_D$ is the identity relation over D.

Definition 2 *We call a finite set of cases a* casebase. *Let* $\mathcal{CB}$ *be a casebase. We divide* $\mathcal{CB}$ *into two sets* $\mathcal{OK}$ *and* $\mathcal{NG}$ *which are called* OK cases *and* NG cases *respectively.*

Example 1 *We consider the following example throughout the paper* [2]

Cases in Case base

$c1$**:** The report is 1 day late. The student gave an excuse saying that his printer was down. The teacher accepted the report.

$c2$**:** The report is 2 days late. The student did not give any excuse. The teacher did not accept the report.

$c3$**:** The report is 6 days late. The student gave an excuse that his printer was down. The teacher did not accepted the report.

$c4$**:** The report is 7 days late. The student gave an excuse that the head of the school ordered him to do some social volunteer work and he had a letter from the head. The teacher accepted the report.

$c5$**:** The report is 8 days late. The student gave an excuse that he was ill and he had a letter from his doctor. The teacher did not accept the report.

Current Case c**:** The report is 5 days late. The student gave an excuse that he was ill and he had a letter from his doctor.

Constraints We have the following constraints for equivalence relation for factors besides the reflexive, symmetrical and transitive properties.

EC1: If the date, X, of giving a report to the teacher in the current case is 4 days later than the date, Y, in a case in $\mathcal{CB}$, then X cannot be similar to Y.

EC2: If the date, X, of giving a report to the teacher in the current case is assumed to be similar to the giving date, Y, in a case in $\mathcal{CB}$ and $X < Y$ then, X is also similar to the giving date, Z, in another case in $\mathcal{CB}$ between X and Y.

[2] This example is inspired by the invited lecture by Prof. Rissland at the legal reasoning workshop in Japan.

EC3: We cannot analogize the cases in one of which there exists an excuse and in the other of which there exists no excuse.

To perform case-based reasoning, we consider the following factors for the above cases.

dl**:** delaying factor taking the value $\{1, 2, 5, 6, 7, 8\}$ which expresses the days of lateness.

ex**:** excuse factor taking the value $\{y, n\}$ which expresses that there is an excuse or no excuse respectively.

rn**:** reason factor taking the value $\{n, p, h, d\}$ which expresses that the reason is none, by the printer, by the head of the school and by the doctor respectively.

Then, the above cases are expressed as follows.

$c1$**:** $c1 \in \mathcal{OK}$, $dl(c1) = 1$, $ex(c1) = y$, $rn(c1) = p$.
$c2$**:** $c2 \in \mathcal{NG}$, $dl(c2) = 2$, $ex(c2) = n$, $rn(c2) = n$.
$c3$**:** $c3 \in \mathcal{NG}$, $dl(c3) = 6$, $ex(c3) = y$, $rn(c3) = p$.
$c4$**:** $c4 \in \mathcal{OK}$, $dl(c4) = 7$, $ex(c4) = y$, $rn(c4) = h$.
$c5$**:** $c5 \in \mathcal{NG}$, $dl(c5) = 8$, $ex(c5) = y$, $rn(c5) = d$.
c**:** $dl(c) = 5$, $ex(c) = y$, $rn(c) = d$.

And the above constraints for equivalence relation for factors can be expressed as follows.

EC1: This can be expressed as

$$((X - Y) \geq 4) \supset X \not\sim_{dl} Y.$$

In this example, this is equivalent to $5 \not\sim_{dl} 1$ since we are only interested in similarity between the value of current case (5 days late) and other values.

EC2: This can be expressed as

$$(X \sim_{dl} Y) \wedge (X < Z) \wedge (Z < Y) \supset X \sim_{dl} Z.$$

In this example, this is equivalent to $(5 \sim_{dl} 7) \supset (5 \sim_{dl} 6)$, $(5 \sim_{dl} 8) \supset (5 \sim_{dl} 7)$ and $(5 \sim_{dl} 8) \supset (5 \sim_{dl} 6)$.

EC3: This can be expressed as

$$(X \sim_{ex} Y) \equiv X =_{\{y,n\}} Y.$$

In this example, this is equivalent to $y \not\sim_{ex} n$. □

We also introduce the following (meta-)constraint for important factors IF since if rn is an important factor, existence of reason should be also relevant factor:

If $rn \in IF$ then $ex \in IF$.

Definition 3 *Let* $IF_{OK} \subseteq \mathcal{F}$*. We say that* C *is OK w.r.t. important factors* IF_{OK} *if there exists* $C_{ok} \in \mathcal{OK}$ *and an unmatched factor* UMF *which is a function from* $\mathcal{NG}$ *to* IF_{OK} *s.t. for all* $f \in IF_{OK}$*, adding the following constraints for equivalence relation* $EC_{OK(f)}$ *into* EC_f *does not lead to contradiction:*

$\{f(C) \sim_f f(C_{ok})\}$
$\cup \{f(C) \not\sim_f f(C_{ng}) \mid C_{ng} \in \mathcal{NG}$ *and* $f = UMF(C_{ng})\}$

We call C_{ok} *a* supporting case to prove OK for C *and* IF_{OK} *a* set of important factors to prove OK for C *and* $EC_{OK(f)}$ *a* set of additional constraints for the equivalence relation to prove OK for C.

We can have a symmetrical definition of C to be NG by exchanging OK and NG in the above definition.

The above definition intends the following:

- Firstly, we identify the important factors between cases whose values are subject to soft matching.
- If the values of all the important factors between cases are consistently assumed to be similar then the cases are similar.
- We try to distinguish every NG case, C_{ng}, from the current case by assuming difference between values of some important factor (denoted as $UMF(C_{ng})$) between the current case and C_{ng}.
- The above assumption of similarity and dissimilarity of values must be consistent with the given constraints for equivalence.

Note that the current case can sometimes be NG and OK by focusing a different set of important factors.

Example 2 *The following shows that the current case can be NG and OK.*

We consider the same setting as Example 1.

The current case, c, is OK w.r.t. $\{dl, ex, rn\}$ since there is a supporting case $c4$ and an unmatched factor UMF from $\mathcal{NG}$ to $\mathcal{F}$ where $UMF(c2) = ex$, $UMF(c3) = rn$ and $UMF(c5) = dl$ s.t.

1. for dl, adding constraints, $dl(c) \sim_{dl} dl(c4)(= 5 \sim_{dl} 7)$ and $dl(c) \not\sim_{dl} dl(c5)(= 5 \not\sim_{dl} 8)$, to the original constraints EC_{dl} does not lead to contradiction.
2. for ex, adding constraints, $ex(c) \sim_{ex} ex(c4)(= y \sim_{ex} y)$ and $ex(c) \not\sim_{ex} ex(c2)(= y \not\sim_{ex} n)$, to EC_{ex} does not lead to contradiction.
3. for rn, adding constraints, $rn(c) \sim_{rn} rn(c4)(= d \sim_{rn} h)$ and $rn(c) \sim_{rn} rn(c3)(= d \not\sim_{rn} p)$, to EC_{rn} does not lead to contradiction.

$c4$ is similar to the current case because we can consistently assume that 5 days delay and 7 days delay are similar and the reason of illness is similar to the reason by volunteer work. And the above UMF expresses how to distinguish c from NG cases as follows;

1. We can distinguish $c2$ since there is no excuse in $c2$ while there is an excuse in c.
2. We can distinguish $c3$ since we argue that the reason of delay by printer trouble is different from the reason by illness. Note that we cannot use the difference of delay to distinguish $c3$ since we already assume $5 \sim_{dl} 7$ and therefore $5 \sim_{dl} 6$ must be assumed as well.
3. We can distinguish $c5$ since we argue that the delay is quite longer than the current case; that is, we assume $5 \not\sim_{dl} 8$

Note that $c1$ cannot be a supporting case by the following reason:

1. It is always $5 \not\sim_{dl} 1$ from the constraint EC1.
2. Therefore, dl cannot be an important factor. Then, if we use ex or rn, we cannot distinguish $c1$ and NG case $c3$.

On the other hand, the current case is NG w.r.t. $\{ex, rn\}$ since there is a supporting case $c5$ and an unmatched factor UMF from $\mathcal{OK}$ to $\mathcal{F}$ where $UMF(c1) = rn$, $UMF(c4) = rn$ and adding constraints, $d \not\sim_{rn} p$ and $d \not\sim_{rn} h$, does not lead to contradiction.

Note that to show that the current case is OK, we argue that reason of illness is similar to the reason of volunteer work whereas to show that the current case is NG, we argue that the above two reasons are not similar. This expresses that dynamic aspects of similarity exists not only in deciding important factors but also in analogizing different values of factors. □

3 TRANSLATION

3.1 Translation for cases in a casebase

For every case C, we translate it into the following rule:

- If $C \in \mathcal{OK}$ then $ok(id(C))$ is a head of the rule and if $C \in \mathcal{NG}$ then $ng(id(C))$ is a head of the rule.
- For every factor f, $f(f(C))$ is added to the body of the rule.

Example 3 *The following shows the translation of the case base in Example 1.*

$c1$: $ok(c1) \leftarrow dl(1), ex(y), rn(p).$
$c2$: $ng(c2) \leftarrow dl(2), ex(n), rn(n).$
$c3$: $ng(c3) \leftarrow dl(6), ex(y), rn(p).$
$c4$: $ok(c4) \leftarrow dl(7), ex(y), rn(h).$
$c5$: $ng(c5) \leftarrow dl(8), ex(y), rn(d).$

3.2 Translation for the current case

Let C be the current case. For every factor $f \in \mathcal{F}$, we make the following rule:

- $f(X) \leftarrow sim^*(f(C), X, f).$
  where sim^* is an abducible predicate.
- $f(X) \leftarrow$ not $imp^*(f).$
  where imp^* is also an abducible predicate and " not " is negation as failure.

Example 4 *The following shows the translation of the current case in Example 1.*

$dl(X) \leftarrow sim^*(5, X, dl).$
$dl(X) \leftarrow$ *not* $imp^*(dl).$
$ex(X) \leftarrow sim^*(y, X, ex).$
$ex(X) \leftarrow$ *not* $imp^*(ex).$
$rn(X) \leftarrow sim^*(d, X, rn).$
$rn(X) \leftarrow$ *not* $imp^*(rn).$

The idea of the above translation is as follows. Firstly, every case is translated into a rule whose conclusion is a consequence of the case and whose body consists of the values of the factors in the case. If the current case coincides with a case which have the similar value to the current case in the important factors, the current case is supposed to have the same consequence as the matched case. This matching is done by using $imp^*(f)$ which expresses that f is assumed to be an important factor and $sim^*(v, X, f)$ which expresses that the value v of the factor f in the current case is assumed to be similar to X.

Suppose that a factor is not important. This means that not $imp^*(f)$ is true. Then, $f(X)$ becomes automatically true whatever value X is. This corresponds with the behavior that the factor f is ignored to match the current case and a case in a casebase. On the other hand, suppose that a factor is important. This means that $imp^*(f)$ is true. Then, $f(X) \leftarrow sim^*(v, X, f)$ is the only rule for $f(X)$. In this case, soft matching will take place by assuming $sim^*(v, X, f)$.

3.3 Translation for Constraints

Constraints are firstly translated into a clausal form:
$(p_1 \wedge ... \wedge p_m) \supset (p_{m+1} \vee ... \vee p_n)$
and the clausal form is translated into the following rule with the contradiction symbol $\bot$ as a head:
$\bot \leftarrow p_1, ..., p_m,$ not $p_{m+1}, ...,$ not p_n

We always have the following constraints.

Constraints for Equivalence Relation: For every factor f and every $d, d_1, d_2, d_3 \in D$ where D is a range of f,
$\bot \leftarrow sim^*(d, d, f).$
$\bot \leftarrow sim^*(d_1, d_2, f),$ not $sim^*(d_2, d_1, f).$
$\bot \leftarrow sim^*(d_1, d_2, f), sim^*(d_2, d_3, f),$ not $sim^*(d_1, d_3, f).$

Meta-constraint:
$\bot \leftarrow ok(X), ng(Y).$
This constraint expresses that the current case is not similar to OK case and NG case simultaneously.

Example 5 *The constraints in Example 1 are translated into the following. We show the instantiated constraints for simplicity.*

EC1: $\bot \leftarrow sim^*(5, 1, dl).$
EC2: $\bot \leftarrow sim^*(5, 7, dl),$ *not* $sim^*(5, 6, dl).$
$\bot \leftarrow sim^*(5, 8, dl),$ *not* $sim^*(5, 7, dl).$
$\bot \leftarrow sim^*(5, 8, dl),$ *not* $sim^*(5, 6, dl).$
EC3: $\bot \leftarrow sim^*(y, n, ex).$
IF: $\bot \leftarrow imp^*(rn),$ *not* $imp^*(ex).$

Among the above constraints, the constraints of sim^* is the source of cutting undesirable similarity among the values of the domain.

3.4 Translation for the position

If a user would like to show that the current case is OK then ask $? - ok(X)$ and if he would like to show that the current case is NG then ask $? - ng(X)$.

3.5 Relation between Formalism and Translation

Semantics of the above translated programs is based on generalized stable model [4] as follows. We extend the definition slightly so that negated abducibles can be included in abducibles.

Definition 4 *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.*

$\Pi_T^M = \{H \leftarrow B_1, ..., B_k |$
$H \leftarrow B_1, ..., B_k, \sim A_1, ..., \sim A_m \in \Pi_T$
and $A_i \notin M$ *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 $\bot \notin M$.*

Definition 5 *Let T be a logic program with abducible predicates and Θ be a set of ground instances or negated ground instances of abducible predicates (called* abducibles*). A* generalized stable model $M(\Theta)$ *for T is a stable model of $T(\Theta)$ where*

$T(\Theta) = T \cup \{H \leftarrow | H$ *is an atom,* $H \in \Theta\} \cup$
$\{\bot \leftarrow H | H$ *is an atom, not* $H \in \Theta\}$

The following theorem states that if there is a model for a translated program, then the abducibles in the model correspond with important factors and a set of the additional constraints for the equivalence relation to prove OK for the current case.

Theorem 1 *Let T be a translated logic program from $\mathcal{CB}$, the current case C and constraints. Suppose that there exists a generalized stable model $M(\Theta)$ for T 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 $IF_{OK} = \{f | imp^*(f) \in \Theta\}$ is a set of important factors and there exists a set of additional constraints for the equivalence relation to prove OK for C.*

The following theorem states that if there is a supporting case, a set of important factors and a set of additional constraints for the equivalence relation to prove OK for the current case, then there is a model which contains abducibles corresponding with the important factors and additional constraints for the equivalence relation.

Theorem 2 *Let T be a translated logic program from $\mathcal{CB}$, the current case C and constraints. Suppose that C_{ok} is a supporting case and IF_{OK} is a set of important factors and $EC_{OK(f)}$ is an additional constraints for the equivalence relation to prove OK for the current case C. Then, there exists a generalized stable model $M(\Theta)$ for T such that $M(\Theta) \models ok(id(C_{ok}))$ and $\Theta \supseteq \{imp^*(f) | f \in IF_{OK}\}$*
$\cup \{sim^*(f(C), f(C_{ok}), f) | f(C) \sim_f f(C_{ok}) \in EC_{OK(f)}\}$
$\cup \{sim^*(f(C), f(C_{ng}), f) | f(C) \not\sim_f f(C_{ng}) \in EC_{OK(f)}\}.$

A sketch of the proofs are found in Appendix.

Example 6 *Let T be a union of programs in Example 3, Example 4 and Example 5. The following shows the above correspondence in Theorems.*

Let Θ be a set $\{imp^*(dl), imp^*(ex), imp^*(rn),$
$sim^*(5, 6, dl), sim^*(5, 7, dl), sim^*(d, h, rn)$
not $sim^*(5, 1, dl),$ not $sim^*(5, 8, dl),$
not $sim^*(y, n, ex),$ not $sim^*(d, p, rn)\}$
plus a set of abducibles augmented
by the basic properties (reflexivity, symmetry, transitivity)
of equivalence relation.
Let $M(\Theta) = \Theta \cup$
$\{ok(c4), dl(6), dl(7), ex(y), rn(h)\}$
Then, a part of $\Pi_{T(\Theta)}^{M(\Theta)}$ becomes the following program:
$ok(c1) \leftarrow dl(1), ex(y), rn(p).$
$ng(c2) \leftarrow dl(2), ex(n), rn(n).$
$ng(c3) \leftarrow dl(6), ex(y), rn(p).$
$ok(c4) \leftarrow dl(7), ex(y), rn(h).$
$ng(c5) \leftarrow dl(8), ex(y), rn(d).$
$dl(1) \leftarrow sim^*(5, 1, dl).$
$dl(6) \leftarrow sim^*(5, 6, dl).$
$dl(7) \leftarrow sim^*(5, 7, dl).$
$dl(8) \leftarrow sim^*(5, 8, dl).$
$ex(y) \leftarrow sim^*(y, y, ex).$

$ex(n) \leftarrow sim^*(y, n, ex)$.
$rn(h) \leftarrow sim^*(d, h, rn)$.
$rn(p) \leftarrow sim^*(d, p, rn)$.
$imp^*(dl) \leftarrow$.
$imp^*(ex) \leftarrow$.
$imp^*(rn) \leftarrow$.
$sim^*(5, 6, dl) \leftarrow$.
$sim^*(5, 7, dl) \leftarrow$.
$sim^*(d, h, rn) \leftarrow$.
$\bot \leftarrow sim^*(5, 1, dl)$.
$\bot \leftarrow sim^*(5, 8, dl)$.
$\bot \leftarrow sim^*(y, n, ex)$.
$\bot \leftarrow sim^*(d, p, rn)$.

We can easily check that $min(\Pi_{T(\Theta)}^{M(\Theta)}) = M(\Theta)$. Therefore, $M(\Theta)$ is a generalized stable model for T. Then, $IF_{OK} = \{dl, ex, rn\}$ and newly added constraints coincide with the result of Example 2. □

We can use an existing procedure [6] based on the generalized stable model semantics which is an enhanced procedure of [3] with bottom-up checking of integrity constraints. The procedure is sound and complete for a finitely grounded program in the sense that every generalized stable model is exactly computed. Since the translated program can be finitely grounded, we can guarantee that every possible interpretation of a rule is exactly computed due to the above correspondence of abducibles with important factors and the value similarity.

4 CONCLUSION

In this paper, we propose a formalization of adversarial case-based reasoning with soft matching over various kind of values and give a translation method from the formalism into abductive logic programming with two level abducibles; the first for important factors and the second for the soft matching.

We need the following future work.

- We would like to incorporate with more powerful constraint reasoning for richer representation of constraints for additional equivalence relation. We might use abductive constraint logic programming [5] to abduce some constraints.
- We would like to pursue the other usage of this formalism such as hypothetical reasoning where some hypothetical scenario can be simulated.

APPENDIX: SKETCH OF PROOFS

Proof of Theorem 1:
Let the rule for C_{ok} be
$ok(c_{ok}) \leftarrow f_1(v_1), ..., f_n(v_n)$.
Since there is no other rule for C_{ok}, this rule should be satisfied. Therefore, for each $f_i(v_i)$, $M(\Theta) \models f_i(v_i)$. $M(\Theta) \models f_i(v_i)$ means that either $M(\Theta) \models sim^*(u_i, v_i, f_i)$ where $u_i = f_i(C)$ and C is the current case, or $M(\Theta) \not\models imp^*(f_i)$. Then, the following hold:

- $f_i(v_i)$ in the body of the rule of $ok(c_{ok})$ means $f_i(C_{ok}) = v_i$.
- $M(\Theta) \models sim^*(u_i, v_i, f_i)$ means that $sim^*(u_i, v_i, f_i)$ is consistent with constraints translated from EC_{f_i}. Therefore, $f_i(C) \sim_{f_i} f_i(C_{ok})$ is consistent with EC_{f_i}.
- $M(\Theta) \not\models imp^*(f_i)$ means $f_i \notin IF_{OK}$.

Thus, $f_i \in IF_{OK}$ implies that $f_i(C) \sim_{f_i} f_i(C_{ok})$ is consistent with EC_{f_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
$ng(c_{ng}) \leftarrow f_1(v_1), ..., f_n(v_n)$.
Then, there exists $f_i(v_i)$ such that $M(\Theta) \not\models f_i(v_i)$. This means that $M(\Theta) \not\models sim^*(u_i, v_i, f_i)$ and $M(\Theta) \models imp^*(f_i)$. Therefore, $f_i \in IF_{OK}$ and $f_i(C) \not\sim_{f_i} f_i(C_{ng})$ is consistent with EC_{f_i}. We take this f_i as the value of $UMF(C_{ng})$.

Since $M(\Theta)$ also satisfies every integrity constraint related to $sim^*(u_i, v_i, f)$, for every f_i, $\{f_i(C) \sim_{f_i} f_i(C_{ok})\} \cup \{f_i(C) \not\sim_{f_i} f_i(C_{ng}) | C_{ng} \in \mathcal{NG}$ and $M(\Theta) \not\models sim^*(f_i(C), f_i(C_{ng}), f_i)\}$ is consistent with EC_{f_i}. This set subsumes $\{f_i(C) \sim_{f_i} f_i(C_{ok})\} \cup \{f_i(C) \not\sim_{f_i} f_i(C_{ng}) | C_{ng} \in \mathcal{NG}$ and $f_i = UMF(C_{ng})\}$. □

Proof of Theorem 2:
Let $T' = T - \{f(X) \leftarrow$ not $imp^*(X) | f \in \mathcal{F}\}$
$\cup \{f(X) \leftarrow | f \notin IF_{OK}\}$
$\cup \{sim^*(u, v, f) \leftarrow |$
$u \sim_f v$ is derived from $EC_{OK(f)}$ and EC_f. $\}$
$\cup \{\bot \leftarrow sim^*(u, v, f) |$
$u \not\sim_f v$ is derived from $EC_{OK(f)}$ and EC_f. $\}$
and $M = min(\Pi_{T'})$.

We first show that $M \models ok(id(C_{ok}))$. Suppose that $M \not\models ok(id(C_{ok}))$. There is only one rule for $ok(id(C_{ok}))$ denoted as:
$ok(c_{ok}) \leftarrow f_i(v_i), ..., f_i(v_n)$.
where $c_{ok} = id(C_{ok})$. Then, since $M \not\models ok(c_{ok})$, there exists $f_i(v_i)$ such that $M \not\models f_i(v_i)$. This means that $M \models imp^*(f_i)$ and $M \not\models sim^*(f_i(C), v_i, f_i)$. And this contradicts with the condition that for every $f \in IF_{OK}$, $f(C) \sim_f f(C_{ok})$ is consistent with EC_f. Therefore, $M \models ok(c_{ok})$. Similarly, we can show that $M \not\models \bot$. Let Θ be a set of abducibles corresponding with IF_{OK} and abducibles corresponding with constraints derived from $EC_{OK(f)}$ and EC_f for all f, and $M(\Theta) = M \cup \Theta$. Then, since $\Pi_{T(\Theta)}^{M(\Theta)} = \Pi_{T'(\Theta)}$,
$min(\Pi_{T(\Theta)}^{M(\Theta)}) = min(\Pi_{T'(\Theta)}) = M \cup \Theta = M(\Theta)$.
Therefore, $M(\Theta)$ is a generalized stable model of T. □

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

[5] Kakas, A. C., Michael, A., "Integrating Abductive and Constraint Logic Programming", *Proc. of ICLP-95*, 399 – 413 (1995).

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

[7] Satoh, K., "Translating Case-Based Reasoning into Abductive Logic Programming", *Proceedings of ECAI-96*, 142 – 146 (1996).