

事例ベース推論における動的類似性の仮説論理プログラミングによる実現[†]

Implementing Dynamic Similarity in Case-Based Reasoning by Abductive Logic Programming

佐藤 健*
Ken Satoh

* 北海道大学大学院工学研究科電子情報工学専攻
Division of Electronics and Information Engineering, Hokkaido University, Sapporo 060, Japan.

1996年11月27日 受理

Keywords: case-based reasoning, abductive logic programming, HYPO.

Summary

In legal reasoning, judging from precedent cases is very important since precedent cases complements incompleteness of legal rules. Therefore, CBR systems such as HYPO have been studied. A main characteristics of such CBR systems in legal reasoning is that similarity between cases are dynamically changed according to contexts. For example, if the system is on the plaintiff side, it will retrieve similar cases which are in favor of the plaintiff side. On the other hand, if the system is on the defendant side, it will retrieve entirely different cases which are in favor of the defendant side based on another similarity measure which is different from the measure used for the sake of the plaintiff. This kind of similarity is naturally implemented in abductive logic programming since we can regard the dynamic similarity as abducible predicates and change these predicates by context. In this paper, we define a relevance criteria for dynamic similarity and show a translation method to abductive logic programming. Moreover, by using abducibles, we show how to construct an explanation why the current case is similar to the cited case.

1. はじめに

法的推論の分野では法律のみでは正しい答えを決められないときがあり、過去の判例によって判決が左右されるときがある。とくにこの傾向は英米法では顕著であり、このため、法的推論の研究においても英米では、HYPO [Ashley 90] のような事例ベース推論が重要となっている。Ashley [Ashley 90, p. 2] は、法的推論を計算機上で実装するにあたり障害となる以下の問題点があると述べている。

(1) 法的規則は、不完全であり、論理的、意味的に

曖昧であり、ときどき矛盾をはらんでいる。また現在の状況に適應する規則を見つけることは難しい場合がしばしばある。

(2) 法的推論は、対立的 (adversarial) である。すなわち、しばしば法的問題には唯一の正しい答えはなく、通常は少なくとも2つの対立した視点があり、互いの議論は、衝突を起こす。

(3) 法的推論は類推を使った推論である。法律家は過去の判例に対する類推を用い、仮説を提示することによって議論を行う。

これらの問題点を考慮し、Ashley は、法的推論システムの基本メカニズムである対立的な事例ベース推論 (*adversarial case-based reasoning*) を提案した。このメカニズムは視点に応じて類似過去判例が変化するような動的類推を実現している。Ashley は、このメカニズ

[†] 本論文は、第12回 European Conference on Artificial Intelligence において発表された論文 [Satoh 96b] の改訂版である。

ムに基づいたシステム HYPO [Ashley 90] を開発し、さらに、HYPO で用いられているこの動的類推のための関連基準 (*relevance criteria*) に関して論理的な表現 [Ashley 94] を与えている。この場合、過去の事例と現在の事例との類似度は、固定されたものではなく、与えられる事例と弁護士/検事の立場によって類似度が動的に変化することが大きな特徴となっている。

本論文では、そのような動的な類似度の変化する事例ベース推論を仮説論理プログラミングに変換することで実現することを提案する。本論文における動的類似性のアイデアは以下のものである。事例は、その事例において満たされる属性の集合として表される。そして、事例の集合を2つに分けて、原告側に有利な過去の事例および被告側に有利な過去の事例にしておく。さて、ある新しい事例が与えられたときに、その事例が原告側に有利であることを示そうとしたとする。このときに、属性を2つに分け、一方は無視することで新しい事例と原告側に有利な過去の事例の一つとマッチするようにし、もう一方はその事例の結果に貢献する重要な属性であると認識することで新しい事例が被告側に有利な過去のすべての事例とマッチできないようにする働きをする。

このような類似性の動的変化を仮説論理プログラミングで実現することを考える。仮説論理プログラミングにおいては、ゴールによって仮説が変化する。したがって、この仮説を類似度と考えれば、立場によって類似度が変化することを実現できる。具体的には、属性が重要であるかどうかを仮説とし、ある属性が重要であると仮定した場合には、その属性の成立/不成立の状態が新しい事例とマッチしたい事例で一致するときのみマッチできるようにし、属性が重要でないと仮定した場合には、その属性の状態に無関係にマッチできるようにした。また、本論文では、各事例ごとに変換を与え、他の部分に影響を与えないようにしたため、変換が局所的となり、漸進的な変換が可能となっている。

仮説論理プログラミングを用いた類推の研究は、[Iwayama 92, Orihara 91] によっても行われているが、それらの研究では類推すべき述語が最初に与えられているため、静的な類推といえる。これに対し、本研究では類推すべき述語の選択が、状況によって動的にかわるため動的な類推を行っているといえる。

2. 動的類似度の定義

【定義 1】 \mathcal{P} を有限集合とし、属性の集合と呼ぶ。 \mathcal{P} の部分集合を事例と呼ぶ。ある事例の集合 OK を正事

例集合と呼び、それと共通要素のない事例の集合 \mathcal{NG} を負事例集合と呼ぶ。CB を $OK \cup \mathcal{NG}$ とし、事例ベースと呼ぶ。

C を事例とする。 C が OK であるとは、ある正事例集合のなかの事例 C_{ok} と \mathcal{P} の部分集合 I_{ok} が存在し、以下が成り立つことをいう。

- $C \cap I_{ok} = C_{ok} \cap I_{ok}$
- 負事例集合のすべての事例 C_{ng} に対して、

$$C \cap I_{ok} \neq C_{ng} \cap I_{ok}.$$

このとき C_{ok} を C を OK と示唆する事例と呼び、 I_{ok} を C を OK と示唆する重要な属性の集合と呼ぶ。

また、対照的に、 C が \mathcal{NG} であるとは、ある負事例集合のなかの事例 C_{ng} と \mathcal{P} の部分集合 I_{ng} が存在し、以下が成り立つことをいう。

- $C \cap I_{ng} = C_{ng} \cap I_{ng}$
- 正事例集合のすべての事例 C_{ok} に対して、

$$C \cap I_{ng} \neq C_{ok} \cap I_{ng}.$$

このとき C_{ng} を C を \mathcal{NG} と示唆する事例と呼び、 I_{ng} を C を \mathcal{NG} と示唆する重要な属性の集合と呼ぶ。□

重要な属性の成立/不成立に関して、新しい事例と正事例(負事例)とが一致していれば、新しい事例は、正事例(負事例)と類似していると考えられる。この考えは、HYPO でも使われている。本論文では、新しい事例が正事例(負事例)であることを示すために、新しい事例と類似する正事例(負事例)を見つけるだけでなく、どの負事例(正事例)とも類似していないことを示すようにした。

もし新しい事例の重要な属性の状態がすべて正(負)事例と一致していれば、他の重要でない属性を無視することにより、新しい事例は正(負)事例と似ていると考えられる。したがって、新しい事例が OK であることを示すためには、新しい事例に似た $OK(\mathcal{NG})$ と示唆する事例が存在し、すべての負(正)事例が新しい事例と似ていないことを示せばよい。

このような定義を行うと以下の例で示すように、事例ベース、新しい事例、類似事例、目的が変化することで、重要な属性が存在しなかったり変化したりする。

事例ベース 事例ベース中で OK/\mathcal{NG} と示唆する事例が削除されれば、別な事例を探さなければならなくなり重要属性を変える必要がありうる。また、事例が追加されたときでも、前に使った重要属性によっては、新しい事例が正事例と負事例の両方に類似したものになることがありうる。これらの場合には重要属性を変えなければならない。

新しい事例 新しい事例の属性の成立/不成立の様子が異なれば、前の新しい事例に類似していた事例

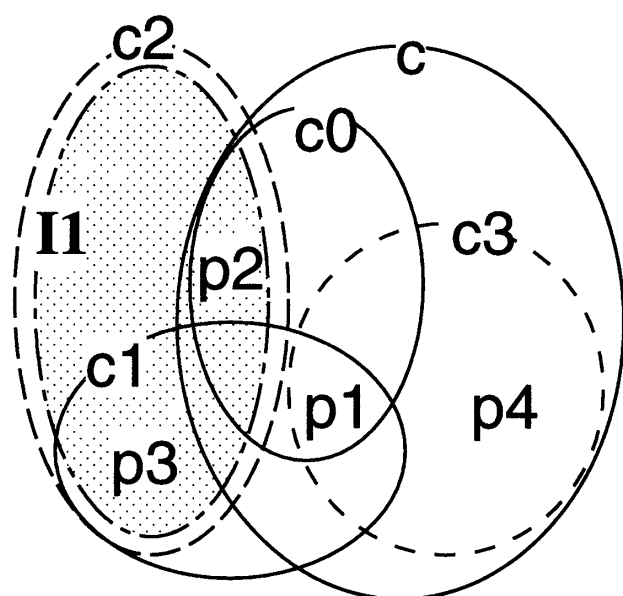


図1 例1の事例の状況

が類似しなくなる場合がある。そのような場合には、重要属性を変化させて類似させるか、類似事例自体も変化させなければならない。

類似事例 類似させたい事例が変わった場合に、重要属性を変化させないと新しい事例と類似しなくなる場合がある。

目的 目的とは、OK / NG のどちらを示したいかである。目的が異なれば、同じ重要属性を取ることとはできない。なぜならば、もし同じ重要属性によってOK / NG のどちらも示せるとすれば、新しい事例は正事例と負事例の両方に類似することになり定義に反するからである。

一般的な法的推論では、上記の変化だけでなく、価値判断や社会的な要請の変化によって重要属性が変化する。これに対してHYPOにおいては、法律のドメインの中には、解析がむずかしい価値判断や社会的な要請を扱わなくても、重要となりうる属性の集合を法律家が特定でき、その集合に基づいて法律家と同じようなことができることを示している。本論文の目的は、HYPOと同程度の推論能力をもつ事例ベース推論システムを実現することであるので、HYPOと同様のアプローチを取った。

〔例1〕 $P = \{p1, p2, p3, p4\}$, $OK = \{c0, c1\}$, $NG = \{c2, c3\}$, $c0 = \{p1, p2\}$, $c1 = \{p1, p3\}$, $c2 = \{p2, p3\}$, $c3 = \{p1, p4\}$ とする。

新しい事例 c を $\{p1, p2, p4\}$ とする。 $I_1 = \{p2, p3\}$ とすれば、正事例 $c0$ に対して、 $c0 \cap I_1 = \{p2\} = c \cap I_1$ でありすべての負事例 $c2, c3$ に対して、 $c2 \cap I_1 =$

$\{p2, p3\} \neq c \cap I_1$ かつ $c3 \cap I_1 = \emptyset \neq c \cap I_1$ であるから、 c をOKと示唆する事例 $c0$ と c をOKと示唆する重要な属性の集合 I_1 が存在するので、 c はOKである。図1にその状況を示す。図でわかるように、 $p2$ が $c, c0$ で成り立ち、 $p3$ が $c, c0$ で成り立たないため、これらを重要な属性とすれば、 $c, c0$ ではその成立/不成立の状態が一致するため類似していると考えることができる。この場合、 $p4$ は c で成立、 $c0$ で不成立であるが、 $p4$ を重要でないとすることでその違いを無視している。また、負事例である $c1, c2$ に対しては、 c とそれぞれ $p3, p2$ の成立/不成立の状態が異なるため差別化が可能となっている。また、この例では、 $c0$ をOKと示唆する事例としたときの重要な属性の集合として、他に $\{p1, p2, p3\}$ と $\{p1, p2\}$ が存在している。

一方、 $c1$ をOKと示唆する事例とすることは以下の理由で不可能である。 c と $c1$ で成立/不成立の状態が一致する属性は $p1$ のみであるが、 $p1$ を重要な属性とすると負事例 $c3$ に対しても $c3 \cap \{p1\} = c \cap \{p1\} = \{p1\}$ となり、負事例にも類似することになってしまうからである。□

次の例は、新しい事例や示唆する事例が変化した場合に重要な属性が変化することを示している。

〔例2〕 P, OK, NG を例1と同じとする。新しい事例 c が $\{p1, p2, p3\}$ であったとする。すると $c0 \cap I_1 = \{p2\} \neq c \cap I_1$ であるから、例1における I_1 は重要な属性となくなる。そこで、 I_2 として $\{p1, p4\}$ を取ったとする。すると、正事例 $c0$ に対して $c0 \cap I_2 = \{p1\} = c \cap I_2$ であり、負事例 $c2, c3$ のおのおのに対し、 $c2 \cap I_2 = \emptyset \neq c \cap I_2$ かつ $c3 \cap I_2 = \{p1, p4\} \neq c \cap I_2$ であるから、 c はOKである。また、 $c0$ に対応した重要な属性として、他に $\{p1, p2, p4\}, \{p1, p2\}$ があり、また、OKを示唆する事例として $c1$ があり、 $c1$ に対する重要な属性として $\{p1, p3, p4\}, \{p1, p3\}, \{p1, p4\}$ がある。□

次は、OKであることを示したいのか、NGであることを示したいのかの目的が違えば重要な属性が変化する例である。

〔例3〕 P, OK, NG, c を例1と同じとする。 $I_3 = \{p3, p4\}$ とすると負事例 $c3$ に対して、 $c3 \cap I_3 = \{p4\} = c \cap I_3$ であり、すべての正事例に対して $c0 \cap I_3 = \emptyset \neq c \cap I_3$ かつ $c1 \cap I_3 = \{p3\} \neq c \cap I_3$ であるので、 c をNGと示唆する事例 $c3$ と c をNGと示唆する重要な属性の集合 I_3 が存在するので、 c はNGである。この例では、 $c3$ をNGと示唆する事例としたときの重要な属性の集合として、他に $\{p1, p3, p4\}$ と $\{p1, p4\}$ が存在している。□

3. 仮説論理プログラムへの変換

本章では、事例ベース推論の仮説論理プログラミングへの変換について述べる。この変換は、事例ベースのルール変換、新しい事例の属性のルール変換、および立場の一貫性制約およびゴールへの変換の3つの部分からなる。

【定義 2】 事例ベース中の事例 C の基本変換は以下である。 p_1, \dots, p_k を C に含まれる属性とし、 n_1, \dots, n_m を C に含まれない属性とする。

もし C が正事例であれば、

$$\begin{aligned} ok(id(C)) &\leftarrow f(p_1), \dots, f(p_k), \\ &\quad nf(n_1), \dots, nf(n_m), \end{aligned}$$

もし C が負事例であれば、

$$\begin{aligned} ng(id(C)) &\leftarrow f(p_1), \dots, f(p_k), \\ &\quad nf(n_1), \dots, nf(n_m) \end{aligned}$$

に変換する。ここで $id(C)$ は、 C の識別子を表すものとする。また、以下のルールを追加する。

$$\begin{aligned} f(P) &\leftarrow cf(P) \\ f(P) &\leftarrow \sim imp^*(P) \\ nf(P) &\leftarrow ncf(P) \\ nf(P) &\leftarrow \sim imp^*(P) \end{aligned}$$

ここで、 P は変数、 \sim は失敗による否定を表し imp^* は仮説述語を表す。上記のプログラムをまとめて T_{basic} と表す。 □

【例 4】 P, OK, NG を例 1 と同じとする。事例ベースの基本変換は以下のようになる。

$$\begin{aligned} ok(c0) &\leftarrow f(p1), f(p2), nf(p3), nf(p4) \\ ok(c1) &\leftarrow f(p1), nf(p2), f(p3), nf(p4) \\ ng(c2) &\leftarrow nf(p1), f(p2), f(p3), nf(p4) \\ ng(c3) &\leftarrow f(p1), nf(p2), nf(p3), f(p4) \\ f(P) &\leftarrow cf(P) \\ f(P) &\leftarrow \sim imp^*(P) \\ nf(P) &\leftarrow ncf(P) \\ nf(P) &\leftarrow \sim imp^*(P) \end{aligned} \quad \square$$

上記の変換のアイデアを以下に示す。すべての事例ベース中の事例は、ヘッドでその事例の正/負の別を表し、ボディで各事例の各属性の状態を表す。新しい事例と事例ベース中の事例との照合は、属性 p が重要な属性であることを示す仮説述語 $imp^*(p)$ を介して行われる。たとえばある属性 p が重要でないとする。すると $\sim imp^*(p)$ が真となり、 $f(p)$ と $nf(p)$ は両方とも真となる。これは、 p が照合において無視されることを示す。一方属性 p が重要であるとすると、 $imp^*(p)$ が真となり $f(p) \leftarrow cf(p)$ と $nf(p) \leftarrow ncf(p)$ が $f(p)$ と

$nf(p)$ を真とする唯一のルールとなる。以下の定義が示すように、 $cf(p)$ と $ncf(p)$ は新しい事例の属性の状態を表しているの、結局、照合は、新しい事例と事例ベース中の事例の属性の状態が一致するときに成功することになる。

【定義 3】 新しい事例 C に対する変換は以下である。 p_1, \dots, p_k を C 内の属性とし、 n_1, \dots, n_m を C 外の属性とする。

$$\begin{aligned} cf(p_1) &\leftarrow \\ &\quad \vdots \\ cf(p_k) &\leftarrow \\ ncf(n_1) &\leftarrow \\ &\quad \vdots \\ ncf(n_m) &\leftarrow \end{aligned}$$

上記プログラムを $T_{current}$ と書く。 □

【例 5】 新しい事例 c を例 1 と同じとする。 c の変換は以下のようになる。

$$\begin{aligned} cf(p1) &\leftarrow \\ cf(p2) &\leftarrow \\ ncf(p3) &\leftarrow \\ cf(p4) &\leftarrow \end{aligned} \quad \square$$

【定義 4】 c が OK であることを示すためには、以下の一貫性制約をプログラム $T_{basic} \cup T_{current}$ に付け加える。

$$\perp \leftarrow ng(X)$$

付け加えたプログラムを T_{ok} と表し、この T_{ok} に対して、 $? - ok(C)$ と聞く。

また、 c が NG であることを示すためには、以下の一貫性制約をプログラム $T_{basic} \cup T_{current}$ に付け加える。

$$\perp \leftarrow ok(X)$$

付け加えたプログラムを T_{ng} と表し、この T_{ng} に対して、 $? - ng(C)$ と聞く。 □

上記プログラムの意味論は一般化安定モデル (generalized stable model) [Kakas 90] であり、以下のように定義される。

【定義 5】 T を論理プログラム、 Π_T を T のすべてのルールの変数をエルブラン空間の要素で置き換えた基底ルールの集合とする。 M を Π_T における基底アトム の集合とし、プログラム Π_T^M を以下のように定義する。

$$\begin{aligned} \Pi_T^M &= \{H \leftarrow B_1, \dots, B_k \mid \\ &\quad H \leftarrow B_1, \dots, B_k, \sim A_1, \dots, \sim A_m \in \Pi_T \\ &\quad \text{and } A_i \notin M \text{ for each } i = 1, \dots, m.\} \end{aligned}$$

$\min(\Pi_T^M)$ を Π_T^M の最小モデルとしたとき、 M が T の安定モデルであるとは、 $M = \min(\Pi_T^M)$ かつ $\perp \notin M$ の

ときである。□

【定義 6】 T を仮説述語をもつ論理プログラムとし、 Θ を仮説述語の基底アトム集合とする。 Θ を仮説と呼ぶ。 $T(\Theta)$ を $T \cup \{H \leftarrow |H \in \Theta\}$ としたとき、 $T(\Theta)$ の安定モデル $M(\Theta)$ を、 T の一般化安定モデルと定義する。 □

以下の定理がこの変換の健全性 (重要な属性の仮説が見つければ類似している) と完全性 (類似していれば重要な属性の仮説が必ず存在する) を表している。

【定理 1】 T_{ok} の一般化安定モデル $M(\Theta)$ が存在し、かつ $M(\Theta) \models ok(c_{ok})$ となる c_{ok} が存在したとする。すると、 $C_{ok} = id^{-1}(c_{ok})$ は新しい事例 C を OK と示唆する事例であり $\{p | imp^*(p) \in \Theta\}$ は新しい事例を OK と示唆する重要な属性の集合となる。

【定理 2】 C_{ok} を、新しい事例を OK と示唆する事例とし、 I を、新しい事例 C を OK と示唆する重要な属性の集合とする。すると、 T_{ok} に対して一般化安定モデル $M(\Theta)$ が存在し、 $M(\Theta) \models ok(id(C_{ok}))$ かつ $\Theta = \{imp^*(p) | p \in I\}$ である。

証明は付録 A を参照されたい。

【例 6】 T_{ok} を例 4 と例 5 の和集合とする。 $\Theta = \{imp^*(p2), imp^*(p3)\}$ とする。

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

$\Pi_{T_{ok}(\Theta)}$ は、 T_{ok} の変数を具体化したものと Θ をルール化した以下のプログラムとなる。

$$\begin{aligned} ok(c0) &\leftarrow f(p1), f(p2), nf(p3), nf(p4) \\ ok(c1) &\leftarrow f(p1), nf(p2), f(p3), nf(p4) \\ ng(c2) &\leftarrow nf(p1), f(p2), f(p3), nf(p4) \\ ng(c3) &\leftarrow f(p1), nf(p2), nf(p3), f(p4) \\ f(p1) &\leftarrow cf(p1) \\ f(p2) &\leftarrow cf(p2) \\ f(p3) &\leftarrow cf(p3) \\ f(p4) &\leftarrow cf(p4) \\ f(p1) &\leftarrow \sim imp^*(p1) \\ f(p2) &\leftarrow \sim imp^*(p2) \\ f(p3) &\leftarrow \sim imp^*(p3) \\ f(p4) &\leftarrow \sim imp^*(p4) \\ nf(p1) &\leftarrow ncf(p1) \\ nf(p2) &\leftarrow ncf(p2) \\ nf(p3) &\leftarrow ncf(p3) \\ nf(p4) &\leftarrow ncf(p4) \\ nf(p1) &\leftarrow \sim imp^*(p1) \\ nf(p2) &\leftarrow \sim imp^*(p2) \end{aligned}$$

$$\begin{aligned} nf(p3) &\leftarrow \sim imp^*(p3) \\ nf(p4) &\leftarrow \sim imp^*(p4) \\ cf(p1) &\leftarrow \\ cf(p2) &\leftarrow \\ ncf(p3) &\leftarrow \\ cf(p4) &\leftarrow \\ \perp &\leftarrow ng(c0) \\ \perp &\leftarrow ng(c1) \\ \perp &\leftarrow ng(c2) \\ \perp &\leftarrow ng(c3) \\ imp^*(p2) &\leftarrow \\ imp^*(p3) &\leftarrow \end{aligned}$$

すると、 $\Pi_{T_{ok}(\Theta)}^M$ は、 $\Pi_{T_{ok}(\Theta)}$ のルールの中で、 $M(\Theta)$ に含まれるアトム ($imp^*(p2), imp^*(p3)$) を否定したりテラルをボディに持つ以下のルール。

$$\begin{aligned} f(p2) &\leftarrow \sim imp^*(p2) \\ f(p3) &\leftarrow \sim imp^*(p3) \\ nf(p2) &\leftarrow \sim imp^*(p2) \\ nf(p3) &\leftarrow \sim imp^*(p3) \end{aligned}$$

を消去し、それ以外のルールから否定リテラルをとったものになる。つまり、

$$\begin{aligned} ok(c0) &\leftarrow f(p1), f(p2), nf(p3), nf(p4) \\ ok(c1) &\leftarrow f(p1), nf(p2), f(p3), nf(p4) \\ ng(c2) &\leftarrow nf(p1), f(p2), f(p3), nf(p4) \\ ng(c3) &\leftarrow f(p1), nf(p2), nf(p3), f(p4) \\ f(p1) &\leftarrow cf(p1) \\ f(p2) &\leftarrow cf(p2) \\ f(p3) &\leftarrow cf(p3) \\ f(p4) &\leftarrow cf(p4) \\ f(p1) &\leftarrow \\ f(p4) &\leftarrow \\ nf(p1) &\leftarrow ncf(p1) \\ nf(p2) &\leftarrow ncf(p2) \\ nf(p3) &\leftarrow ncf(p3) \\ nf(p4) &\leftarrow ncf(p4) \\ nf(p1) &\leftarrow \\ nf(p4) &\leftarrow \\ cf(p1) &\leftarrow \\ cf(p2) &\leftarrow \\ ncf(p3) &\leftarrow \\ cf(p4) &\leftarrow \\ \perp &\leftarrow ng(c0) \\ \perp &\leftarrow ng(c1) \\ \perp &\leftarrow ng(c2) \\ \perp &\leftarrow ng(c3) \\ imp^*(p2) &\leftarrow \end{aligned}$$

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

であり $M(\Theta)$ は T_{ok} の一般化安定モデルとなる。このモデルは、例 1 における重要な属性 I_1 に対応している。

さらに T_{ok} には以下の一般化安定モデルがあることも示すことができ、それらは他の重要な属性の集合に対応している。

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

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

□

4. 説明の生成

上記の変換された仮説論理プログラムに対する証明手続きとして、本論文では、[Sato 96a] で提案した手続きを拡張して、事例がなぜ OK / NG であるかについての説明を生成させるようにした。

簡単に手続きの動作について述べる。事例を OK にするためには $?-ok(C)$ と聞く。まず、手続きは $ok(C)$ と照合が成功するルールを探し、次にボディを満足させようとする。ボディの属性が、新しい事例の属性と同じ状態の場合には、無条件に満足され、そうでなければ、 $\sim imp^*(p_i)$ を真にする（その属性を重要でないと仮定する）。属性が満足されるごとに、反例がないかをボトムアップな一貫性制約検査によって調べる。反例がある場合には、新しい事例の属性のうち、反例の属性と状態が異なる属性 p_i に対して、 $imp^*(p_i)$ を真とすることで、その反例の充足を妨げる。この実行の間に仮定した仮説やほかの導かれたアトムを蓄積することで説明が作られる。定理 2 と仮説の有限性より、すべての説明を生成することができる。

また説明は以下のように作られる。

類似事例の類似理由 これは、蓄積された仮説 $imp^*(p)$ に対応する属性を列挙することで説明される。

事例一般化ルールの提示 事例のマッチングは重要な属性の成立 / 不成立のみで決まる。いいかえると、

マッチされた事例の属性の状態の一部を落として抽象化したルールが生成され、適用されているとも考えられる。蓄積された仮説 $imp^*(p)$ に対応する属性の新しい事例における状態を用いて、このルールを生成し使用者の参考として提示している。

対抗事例の非類似理由 各対抗事例に対して、蓄積された仮説 $imp^*(p)$ に対応する属性のうち新しい事例と対抗事例との間で状態が不一致であることを列挙することで非類似理由が説明される。

援護事例の非類似理由 援護事例であっても、重要な属性の指定によって類似しなくなるものがある。これらの事例に対して、蓄積された仮説 $imp^*(p)$ に対応する属性のうち新しい事例と援護事例との間で状態が不一致であることを列挙することで説明を生成し、使用者に参考のため提示している。

重要としても構わない他の属性の情報 上記手続きでは、 $\sim imp^*(p)$ も仮説として蓄積しており、 $imp^*(p)$ と $\sim imp^*(p)$ のどちらにも入っていない属性は、重要としても結果はわからない。そのような属性がある場合は、使用者の参考のために提示される。

図 2 では例 1 に対応する説明をシステムが生成している。この説明では、事例一般化ルールとして

不成立 (p3), 成立 (p2) → OK

が提示されている。

また、実際の法的推論に適用した例として、図 3 に示した企業秘密漏洩法に関する HYPO のクレーム束の例 [Ashley 94] を用いる。属性の意味は以下のようである。

- f3:** 被告が一人で開発した製品である。
- f4:** 被告が守秘義務遵守について同意していた。
- f5:** 守秘義務遵守同意が原告の問題製品を特定して行われていない。
- f6:** 原告は、秘密保持のための手段を講じていた。
- f7:** 原告の会社の以前の雇用者が被告に問題製品に関係したツールを紹介した。
- f18:** 被告はまったく同じ製品を作っている。

Structural Dynamics 以外を事例ベースとし、*Structural Dynamics* を新しい事例として与えたとする。

図 4 は *Structural Dynamics* を OK とする説明である。

この説明では、*Analogic* 社の裁判の結果を理由として OK と主張している。このとき、*Analogic* 社の裁判での属性 $f3$ と $f5$ の状態と *Structural Dynamics* 社の裁判における状態は異なっているため、これらを見捨て、 $f4$ と $f7$ が重要であると仮定している。すると、負事

- 事例名：c0(OK) 成立属性 p1,p2
- 事例名：c1(OK) 成立属性 p1,p3
- 事例名：c2(NG) 成立属性 p2,p3
- 事例名：c3(NG) 成立属性 p1,p4

与えられた事例名：c 成立属性 p1,p2,p4
 与えられた事例が OK かどうか調べます。

与えられた例は以下の理由により OK と考えられます。

与えられた事例が正事例 c0 に以下の理由により類似しているからです。

その理由とは

属性 p3,p2 が重要であると考えており、それらの状態が共通であるからです。

この解釈で使われたルールは以下です。

不成立 (p3), 成立 (p2) → OK

すると

与えられた事例は負事例 c2 に関係していますが、
 以下の理由で類似していないと考えます。

その理由とは属性 p3 の状態が共通ではなく

我々はその相違が重要であると考えているからです。

また、

与えられた事例は負事例 c3 に関係していますが、
 以下の理由で類似していないと考えます。

その理由とは属性 p2 の状態が共通ではなく

我々はその相違が重要であると考えているからです。

この解釈では、

与えられた事例は正事例 c1 とも類似してはなりません。

なぜなら属性 p3,p2 の状態が異なるからです。

図2 例1のOKの説明

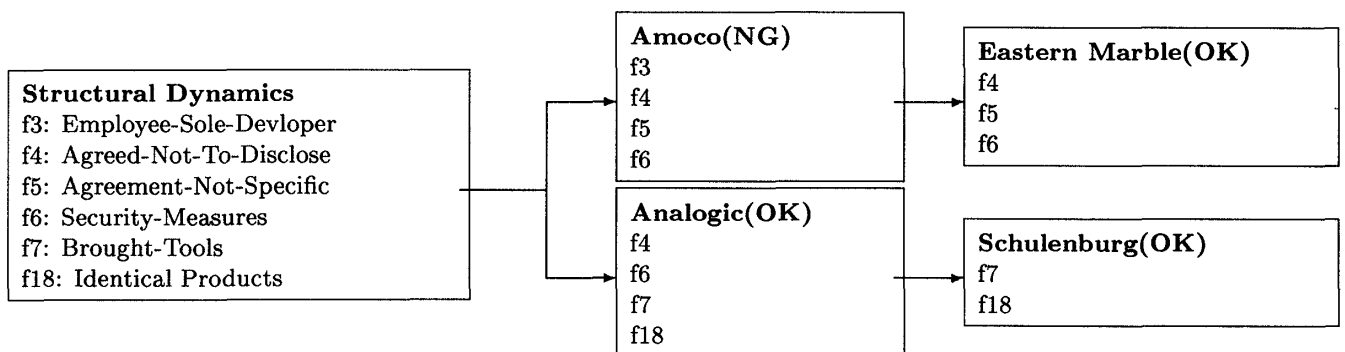


図3 HYPOのクレーム束による事例表現

例である Amoco 社の裁判での f4 の状態が Structural Dynamics 社の裁判における状態と一致するので NG であると反論される可能性があるが、f7 の状態が異なるという理由で、反論を却下している。しかし、この代償として、OK 事例である Easter Marble 社の裁判の

結果と Schulenburg 社の裁判の結果を OK を示唆する事例として使うことができなくなっている。最後の行は f6 と f18 を重要な属性としてもよいことを示している。

図5は Structural Dynamics を NG とする説明で

事例名：amoco(NG) 成立属性 f3,f4,f5,f6

事例名：analogic(OK) 成立属性 f4,f6,f7,f18

事例名：eastern_marble(OK) 成立属性 f4,f5,f6

事例名：shulenburg(OK) 成立属性 f7,f18

与えられた事例名：structure_dynamics 成立属性 f3,f4,f5,f6,f7,f18

与えられた事例が OK かどうか調べます。

与えられた例は以下の理由により OK と考えられます。

与えられた事例が正事例 analogic に以下の理由により類似しているからです。

その理由とは

属性 f4,f7 が重要であると考えており、それらの状態が共通であるからです。

この解釈で使われたルールは以下です。

成立 (f4), 成立 (f7) → OK

すると

与えられた事例は負事例 amoco に関係していますが、

以下の理由で類似していないと考えます。

その理由とは属性 f7 の状態が共通ではなく

我々はその相違が重要であると考えているからです。

この解釈では、

与えられた事例は正事例 eastern_marble とも類似してはいません。

なぜなら属性 f7 の状態が異なるからです。

また、

与えられた事例は正事例 shulenburg とも類似してはいません。

なぜなら属性 f4 の状態が異なるからです。

また、この解釈では、

属性 f6,f18 は重要であると考えても重要でないと考えても答えは同じです

図4 OKの説明

ある。

この説明では、Amoco 社の裁判の結果を理由として NG と主張するため属性 f3 を重要な属性と仮定している。すると正事例はすべて、重要な属性 f3 に関して、Structural Dynamics 社の裁判における状態と一致していないので類似していないと考えることができる。

5. 結 論

この論文の成果は以下である。

- 事例ベース推論から仮説論理プログラミングへの変換を示し、事例ベース推論における類似性に寄与する重要な属性と仮説論理プログラミングの仮説との 1 対 1 対応を示した。
- 仮説論理プログラミングの手続きから得られた重

要属性の仮説から、事例ベース推論の説明を生成する方法について示した。

今後の課題としては以下があげられる。

- CABARET [Rissland 91] のようなルールベース推論と事例ベース推論を仮説論理プログラミング上で融合する。最も簡単な融合は、ルールベースで使われる一貫性制約を導入することである。たとえば、例 1 において属性 p1 と属性 p2 が同時には重要な属性にならない場合を考える。これは、例 6 の T_{ok} に $\perp \leftarrow imp^*(p1), imp^*(p2)$ を加えることに対応し、属性 p1 と属性 p2 が同時に重要な属性となるモデルが排除される。例 6 では、排除された結果、重要属性が {p2, p3} である唯一の一般化安定モデルを選択できる。
- 属性として量的属性のような一階の属性を扱える

事例名：amoco(NG) 成立属性 f3,f4,f5,f6

事例名：analogic(OK) 成立属性 f4,f6,f7,f18

事例名：eastern_marble(OK) 成立属性 f4,f5,f6

事例名：shulenburg(OK) 成立属性 f7,f18

与えられた事例名：structure_dynamics 成立属性 f3,f4,f5,f6,f7,f18

与えられた事例が NG かどうか調べます。

与えられた例は以下の理由により NG と考えられます。

与えられた事例が負事例 amoco に以下の理由により類似しているからです。

その理由とは

属性 f3 が重要であると考えており、それらの状態が共通であるからです。

この解釈で使われたルールは以下です。

成立 (f3)→NG

すると

与えられた事例は正事例 shulenburg に関係していますが、

以下の理由で類似していないと考えます。

その理由とは属性 f3 の状態が共通ではなく

我々はその相違が重要であると考えているからです。

また、

与えられた事例は正事例 eastern_marble に関係していますが、

以下の理由で類似していないと考えます。

その理由とは属性 f3 の状態が共通ではなく

我々はその相違が重要であると考えているからです。

また、

与えられた事例は正事例 analogic に関係していますが、

以下の理由で類似していないと考えます。

その理由とは属性 f3 の状態が共通ではなく

我々はその相違が重要であると考えているからです。

また、この解釈では、

属性 f4,f5,f6 は重要であると考えても重要でないと考えても答えは同じです

図5 NGの説明

ようにする。例えば、年齢に関する属性であれば、ある年齢以上かどうかということが問題となるため、変数を含んだ属性を扱わなければならない。

- HYPO との理論的關係を考える。HYPO では、共通仮説の集合における包含關係を用いて仮説間の順序付けも行っている。これと対応付けるためには、仮説論理プログラミングにおいても仮説の集合の間に順序付けを行う必要があると思われる。
- 法的推論における事例修正を研究する。本論文では、類似事例検索を行い、類似事例の正/負によって NG か OK かを決めるだけであったが、損害の

賠償金額、交通事故の示談金額を決めるようなシステムや JUDGE システム [Bain 86] のような懲役期間を決めるようなシステムにおいては、類似事例を検索した後、新しい事例との差を考えて量的な結論を修正するメカニズムが必要となる。法的推論における事例修正のメカニズムを研究することによって、事例ベース推論における一般的修正問題への示唆を与える可能性があるため、この研究は非常に重要であると考えられる。

謝 辞

本研究は、文部省科学研究費補助金重点領域研究(課題番号 08204202)の補助を受けている。記してここに謝意を表す。

◇ 参 考 文 献 ◇

- [Ashley 90] Ashley, K. D., *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*, MIT Press (1990).
- [Ashley 94] 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*, pp.338 - 352 (1994).
- [Bain 86] Bain, W., "Case-Based Reasoning: A Computer Model of Subjective Assessment", Ph.D. Thesis, Yale University (1986).
- [Eshghi 89] Eshghi, K. and Kowalski, R., "Abduction Compared with Negation by Failure", *Proceedings of ICLP'89*, pp.234 - 254 (1989).
- [Iwayama 92] Iwayama, N., Satoh, K. and Arima, J., "A Formalization of Generalization-based Analogy in General Logic Programs", *Proceedings of ECAI-92*, pp.129 - 133 (1992).
- [Kakas 90] Kakas, A. C., Mancarella, P., "Generalized Stable Models: A Semantics for Abduction", *Proc. of ECAI-90*, pp.385 - 391 (1990).
- [Orihara 91] Orihara, R., "Analogical Reasoning as a Form of Hypothetical Reasoning and Justification-based Knowledge Acquisition", *Proc. of the Second Workshop on Algorithmic Learning Theory*, pp.243 - 254 (1991).
- [Rissland 91] Rissland, E. L., and Skalak, D. B., "CABARET: Statutory Interpretation in a Hybrid Architecture", *International Journal of Man-Machine Studies*, 34(6), pp.839 - 887 (1991).
- [Satoh 96a] Satoh, K. and Iwayama, N., "A Query Evaluation Method for Abductive Logic Programming Based on Generalized Stable Models", *人工知能学会誌*, Vol. 11, No.1, pp.137 - 147 (1996).
- [Satoh 96b] Satoh, K., "Translating Case-Based Reasoning into Abductive Logic Programming", *Proceedings of European Conference on Artificial Intelligence*, pp.142 - 146 (1996).

[査読者：榎木哲夫]

◇ 付 録 ◇

A. 定理の証明

定理1の証明:

C_{ok} を変換したルールを

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

とする。 C_{ok} に関するルールは他にないので、このルールのボディが $M(\Theta)$ で満たされなければならない。ゆえに、各 $f(p_i)$ に対して $M(\Theta) \models f(p_i)$ であり、各 $nf(n_i)$ に対して $M(\Theta) \models nf(n_i)$ である。

ここで $M(\Theta) \models f(p_i)$ であるということは、 $M(\Theta) \models cf(p_i)$ の $M(\Theta) \not\models imp^*(p_i)$ どちらかが成り立つということである。また、以下が成立している。

- $ok(c_{ok})$ のルールのボディに $f(p_i)$ があるということは、

$p_i \in C_{ok}$ ということである。

- $M(\Theta) \models cf(p_i)$ ということは、新しい事例 C に対して $p_i \in C$ ということである。
- $M(\Theta) \not\models imp^*(p_i)$ ということは、 $I = \{p \mid imp^*(p) \in \Theta\}$ としたときに、 $p_i \notin I$ ということである。

したがって、 $p_i \in C_{ok}$ ならば $p_i \in C$ または $p_i \notin I$ が成立している。いいかえれば、 $p_i \in C_{ok}$ かつ $p_i \in I$ のときは $p_i \in C$ が成立している。これはすなわち、 $C_{ok} \cap I \subseteq C$ ということを意味し、 $C_{ok} \cap I \subseteq C \cap I$ が成立している。

同様に、 $M(\Theta) \models nf(p_i)$ であることから、 $p_i \notin C_{ok}$ かつ $p_i \in I$ のときは $p_i \notin C$ より $C \cap I \subseteq C_{ok} \cap I$ が導かれ、 $C \cap I = C_{ok} \cap I$ が成り立つ。

また、 $M(\Theta)$ はすべての一貫性制約を満たすため、各 $C_{ng} \in NG$ に対して、 $c_{ng} = id(C_{ng})$ とすれば、 $M(\Theta) \models ng(c_{ng})$ である。

C_{ng} のルールを

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

とする。すると、このルールのボディのいずれかが $M(\Theta)$ において成立していないため、 $M(\Theta) \not\models f(p_i)$ なる $f(p_i)$ または、 $M(\Theta) \not\models nf(n_i)$ なる $nf(n_i)$ が存在する。

ここで $M(\Theta) \not\models f(p_i)$ なる $f(p_i)$ が存在したとする。すると、 $M(\Theta) \not\models cf(p_i)$ であり、かつ $M(\Theta) \models imp^*(p_i)$ である。したがって、 $p_i \in C_{ng}$ であり $p_i \in I$ であるが $p_i \notin C$ となる。これは、 $C \cap I \neq C_{ng} \cap I$ であることを意味する。一方 $M(\Theta) \not\models nf(n_i)$ なる $nf(n_i)$ が存在したとしても同様に、 $C \cap I \neq C_{ng} \cap I$ であることを示せるため、 $C \cap I \neq C_{ng} \cap I$ である。したがって、どちらの場合でも $C \cap I \neq C_{ng} \cap I$ である。

ゆえに定義より C_{ok} は新しい事例 C を OK と示唆する事例となり、 I が重要な属性となる。したがって C は OK となる。 □

定理2の証明:

T' を

$$T' = T_{ok}$$

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

とし、 $M = \min(\Pi_{T'})$ とする。

まず $M \models ok(id(C_{ok}))$ を示す。結論を否定し、 $M \not\models ok(id(C_{ok}))$ と仮定する。 $ok(id(C_{ok}))$ には唯一のルール

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

が存在している。ここで、 $c_{ok} = id(C_{ok})$ とする。すると、 $M \not\models ok(c_{ok})$ であるから、 $M \not\models f(p_i)$ なる $f(p_i)$ が存在するか、 $M \not\models nf(n_i)$ なる $nf(n_i)$ が存在する。

$M \not\models f(p_i)$ なる $f(p_i)$ が存在するとする。すると、 $p_i \in I$ であり、 $M \not\models cf(p_i)$ である。すなわち $p_i \in C_{ok}$ かつ $p_i \in I$ であるが、 $p_i \notin C$ となり、 $C \cap I \neq C_{ok} \cap I$ となり、これは I の定義に矛盾する。同様に、 $M \not\models nf(n_i)$ なる $nf(n_i)$ が存在する場合も矛盾を示すことができる。したがって、 $M \models ok(c_{ok})$ であり、同様に $M \models \perp$ を満たす。

さらに $M(\Theta) = M \cup \Theta$ とすれば、 $\Pi_{T_{ok}(\Theta)}^{M(\Theta)} = \Pi_{T'(\Theta)}$ であるから、

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

となり、 $M(\Theta)$ は T_{ok} の一般化安定モデルとなる。 □

— 著 者 紹 介 —

佐藤 健(正会員)は、前掲(Vol.12, No.4, p.606)参照。