ICOT Technical Report: TR-685

TR-685

一般化に基づく類推の論理プログラミングによる実現

岩山 登、佐藤 健、有馬 淳

September, 1991

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

一般化に基づく類推の論理プログラミングによる実現

岩山 登, 佐藤 健, 有馬 淳

新世代コンピュータ技術開発機構

概要

類推は, 人間の発見や問題解決において重要な役割を担っている. 類推の研究には, 大きく分けて二種類ある. 一方は, 置換に基づくものであり, 他方は, 一般化に基づくものである. ここでは, 一般化に基づく類推の一つの実現として, 一貫性制約をもつ論理プログラムにおける類推について述べる.

一般化に基づく類推では, 類似性, 投射性を S, P で表すと, $(\exists x.\ S(x) \wedge P(x))$ から $(\forall x.\ S(x) \supset P(x))$ (類推規則と呼ぶ) を帰納することが本質になる. 従って, このタイプの推論では次の2つの前提条件が満足されていることが必須と考えられる:(1) **ベース存在条件**: $(\exists x.\ S(x) \wedge P(x))$, (2) **類推無矛盾条件**: $(\forall x.\ S(x) \supset P(x))$ が無矛盾. (1), (2) の2条件が満たされる時, 類推規則を導くことを一般化に基づく類推の**基本形式**と呼ぶことにする. このような条件を扱う研究では否定情報が表現できる知識表現系が必要になる.

ここでは, この基本形式に基づく類推の (エルブランモデルによる) モデルを与え, それがもとの知識表現としての論理プログラムにある簡明なルールを付け加えたプログラムの (安定モデルによる) モデルと一致することを示し, それを計算する非決定的手続きを示す. この手続きはそれ自体すでに一つの類推システムとみることができるが, 一般化に基づく類推の基本動作を実現したものであり, そのタイプの研究成果の実現のための汎用的な道具として使用することができる. また, 知識表現に陽に否定情報が存在する場合も扱えるため, 論理に基づくより一般的な類推研究の一助になると考えられる.

1 はじめに

類推を論理的システムとして捕らえる研究は, 類推の客観的解明に役立つと考えられ, 今後も重要なアプローチの一つであり続けると思われる. そういった研究では, 通常, 演繹的システムを拡張する形で類推が捕らえられるため, システムの実験的実現には, 実際的な見地から一階述語論理式の部分クラスに表現を限定した論理プログラム (たとえば Prolog) が使われることが多く, 知識表現もそのシンタックス (確定節と呼ばれるクラス[1]) に制限されたものになっている [7, 9, 10].

ある非演繹的な推論の結果は (必ずしも与えられた事実から成り立つとは限らないという点で) 一つの仮説としてみることができる. 科学的な理論形成過程ではこのような仮説に対して必ず検証が行われる. この検証のフェーズでは反例がない (現情報に対して無矛盾である) ことが必須の条件になり, 反例を持つ仮説は修正が行われたり, 捨てられたりすることで, “正しい” 仮説の獲得に貢献することになる. 帰納推論, 学習の研究ではこの考え方はまったく珍しくない. 行き過ぎた汎化を防ぐのは反例 (負例) の存在による. もし, 反例が与えられることがなければ理論形成は非常に簡単になってしまう. 完全な汎化ルールはすべてを説明できるからである. 実は, 確定節のみを扱う非演繹的システムはすべてこの点で問題がある. 確定節で表現される知識はどんなものも無矛盾であり, 反例を記述することができないからである[2]. よって, 確定節の知識のみを扱う研究では本来扱わねばならない検証過程が (論理的意味を持つ議論からは) 漏れ落ちることになり, このことが確定節に制限した場合に大きな見落しになりかねない. 確定節が無矛盾であることの最大の要因は, 否定情報を表すのに必要な負節 (負リテラルのみからなる節) が記述できない点にある[3]. 従って, 負節に対して論理的意味を与えられる論理プログラムへの拡張は類推研究にとって切に必要と考えられる.

近年, 論理プログラムの分野では, プログラムの論理的意味を明確に保ったまま従来の論理プログラムを拡張する多くの優れた研究がなされている. そのような論理プログラムの一つの拡張である, 一貫性制約を用いた論理プログラム [12, 4] を使えば, 上記のような否定情報を直接知識表現上で扱うことができる[4]. 本稿では, 論理的類推研

[1] SLDNF 戦略を取る通常の Prolog 処理系では確定節を含む局所的層状 (*locally stratified*) プログラムと呼ばれるクラスに対して意味論が与えられている [11]. しかし, このクラスに対しても以下の議論が成り立っている.

[2] この抜け道がないわけではない. この問題が決定的である帰納推論の研究, 例えば確定節のプログラム合成では, 負例であることはメタな方法 (論理の外) で記される. しかし, この方法では “負例である (仮説に対し矛盾する)” ことの厳密な論理的意味を欠くことになってしまう.

[3] 例えば, 個体 T が性質 P を満足しないという否定情報は, 最も簡単な負節の例になる. すなわち, $\neg P(T)$ で表される.

[4] 負節: $\neg A_1 \vee \ldots \vee \neg A_N$ は一貫性制約: $\leftarrow A_1, \ldots, A_N$ で表される. ($A_1, \ldots, A_N$ はアトム (正リテラル))

究．特に，一般化に基づく類推の研究 [5, 1, 2] から得られた類推の基本原理が，その論理プログラムで非常に簡明に記述 (すなわち実現) できることを示す．

このシステムはそれ自体すでに一つの類推システムとみることができるが，一般化に基づく類推の基本動作を実現したものであり，そのタイプの研究成果の実現のための汎用的な道具として使用することができる．また，知識表現に否定情報が存在する場合も扱えるため，論理によるより一般的な類推研究の一助になると考えられる．さらに，論理プログラムの観点からは，一貫性制約をもつ論理プログラムの一つの重要な応用域を示したことになるだろう．

以下では，一般化に基づく類推の基本的な形式について述べ，その定義を一貫性制約を伴う論理プログラム上のエルブランモデルを用いた類推の定義に翻訳する．さらに，その定義が，元の論理プログラムにある簡明なルールを付け加えたときの安定モデル (stable model) に一致することを述べる，また，筆者らが先に提案した，安定モデルの非決定的計算手続き [13] を用いた類推の計算を示す．

2 一般化に基づく類推の基本形式

2 つの事柄がある共通の性質 (S: 類似性) を有している時，一方 (B: ベース) の持つ性質 (P: 投射性) を他方 (T: ターゲット) も持つと推定する類推について考察する．このような推論は次のスキーマを使って表せると考えるかもしれない．

$$\frac{\begin{array}{l} S(B) \wedge P(B) \\ S(T) \end{array}}{P(T)} \tag{1}$$

これまでの類推研究における "類推" の捕らえ方は大きく以下の 2 つの型に分類することができるだろう．一つは，類推をベースの性質をターゲットに置き換える操作であると捕らえるもの (伝統的な殆どの類推研究はこの型である) と，類推を一例による一般化と捕らえるもの [5, 1, 2] である．前者の型での類推は置換が本質的な操作になる．すなわち，ともにある種の S を満たす B を T に置き換えることによって "直接" $P(B)$ から $P(T)$ が得られる．これに対して後者の型では，$S(B) \wedge P(B)$ ($(\exists x.\, S(x) \wedge P(x))$) から一般的な知識，$(\forall x.\, S(x) \supset P(x))$，(類拠規則と呼ぶ) を得ることが本質的になる ($P(B)$ は類拠規則と $S(T)$ からの演繹で得られる)．ここでは後者の立場に立った議論を展開する．

さて，(1) 式のスキーマでは類推を表すのは不十分であり，スキーマの前提部にはいくらかの条件が存在する [5, 3]．このような暗黙的条件を明らかにするのは類推研究の中心的課題であり，多くの研究がなされているが，残念ながら未だ完全に同意がとれる解が見つかっていないのが実情である．そこでここでは類推を一例からの一般化と捕らえた場合に必要最低限の 2 つの条件を記し[5]，この 2 条件が満たされた時に類拠規則を導出することを一般化に基づく類推の基本形式と呼び，基本形式に基づく類推を扱うことにする．

以下に一般化に基づく類推に必須の 2 条件を示す．

まず第一に，一般化のもとになる事例，すなわち，ターゲットと共通の性質 (類似性) を満たし，投射性を満たすベースとなる個体が存在しなくてはならない．

ベース存在条件:

A1: $$A \vdash (\exists x.\, S(x) \wedge P(x)) \tag{2}$$

また，推論が無矛盾であるために，一例からの一般化が無矛盾でなければならない．

類拠無矛盾条件:

A2: $$A \cup \{(\forall x.\, S(x) \supset P(x))\} \text{ は無矛盾} \tag{3}$$

ここで類拠無矛盾性は

[5]恐らくこのレベルでは合意がとれるものと思われるが，詳しい議論は本論文の対象外である．別の文献を参照されたい [2]

A2': $$A \cup \{\neg(\exists x.\ S(x) \wedge \neg P(x))\} \text{ は無矛盾} \tag{3'}$$

と等価であることに注意せよ.

3 論理プログラムにおける類推

3.1 準備

最初に, 一貫性制約をともなう論理プログラムとそのプログラムのモデルの定義を行う. この定義は, [12] に基づくが, 一貫性制約の目的が異なることに注意されたい. [12] では, 一貫性制約はデータベースの更新のさいの正当性をチェックするために使われるが, ここでは, 類推が可能かどうかの検証に用いられる.

定義 1 A をアトム, $L_1, ..., L_m (m \geq 0)$ をリテラルとする. **論理プログラム**は以下の形をしたルール

$$A \leftarrow L_1, L_2, ..., L_m.$$

または, 以下の形をした**一貫性制約** *(integrity constraint)*

$$\leftarrow L_1, L_2, ..., L_m.$$

からなる集合である.

ルール中の A をヘッドと呼び, $L_1, ..., L_m$ をボディと呼ぶ. また, R をルールとするときに, R のヘッドを $head(R)$ で表し, また, R のボディにある正リテラルの集合を $pos(R)$ で表し, また, R のボディにある負リテラルから否定記号をとりさった集合を $neg(R)$ で表す. また, 一般論理プログラムの中のルールのボディならびに一貫性制約の中に負リテラルがない場合には, そのプログラムを**正プログラム**と呼ぶ. あとで示すように, 本稿では正プログラムを類推の対象とし, 類推の計算のために一般論理プログラムを使う.

本稿で用いる一般論理プログラムの意味論は安定モデル [6] に基づくものである.

定義 2 論理プログラムを K とする. θ を, K 中のルールまたは, 一貫性制約に存在するすべての変数への K のエルブラン領域の要素の代入とする. K の安定モデル *(stable model)* M は, 以下の条件を満たすエルブランモデル *(Herbrand Model)* である.

1. M は, K^M の最小エルブランモデルに一致する.
   ここで $K^M = \{R'\theta | head(R'\theta) = head(R\theta)\ and\ pos(R'\theta) = pos(R\theta)$
   $and\ neg(R') = \emptyset$ ただし R は K 中のルールであり, $neg(R\theta) \cap M = \emptyset\}$
2. すべての K 中の一貫性制約 C とすべての θ について, $pos(C\theta) \not\subseteq M$ または, $(neg(C\theta) \cap M) \neq \emptyset$ である.
   このとき, 「M は C に違反しない」という.

3.2 類推モデルと類推スキーマ

さて, 以下では, 正プログラムに関する類推の定義を述べる.

定義 3 K を正プログラム, S, P を K に現れる述語記号とする. また, $HU(K)$ を K のエルブラン領域とし, $min(K)$ を K の最小エルブランモデルとする. K が, S **を類似性として** P **を類推可能である**とは, ある $t \in HU(K)$ が存在して, $min(K) \models S(t) \wedge P(t)$ であって, すべての $t \in HU(K)$ に対して, $M \models S(t) \rightarrow P(t)$ なる K のエルブランモデル M が存在することをいう.

K が正プログラムであるので, $min(K)$ で真であるアトムは K のすべてのモデルで真である. したがって最初の条件からベース存在条件が成り立つ. また, 2 番目の条件は, 類拠無矛盾条件が成り立つことである. したがって, 一般化に基づく類推の基本形式の 2 つの条件を満たしている.

定義 4 S, P を論理プログラム K に現れる述語記号とし, K を S を類似性として P を類推可能であるとする. K のエルブランモデル M が, S **を類似性とする** P **に関する類推モデル**であるとは, M が $K \cup \{P(x) \leftarrow S(c)\}$ の最小モデルであることをいう.

さて，上の定義は K のエルブランモデル上のメタ的な定義であったが，次に示す新たなルールを K に加えたときの安定モデルとしても定義できる．

定義 5 K を正プログラム，S, P を K に現れる述語記号，$appli, contra$ を K に現れない述語記号とする．以下のルールの集合を，S を類似性とする P に関する**一般化類推スキーマ**と呼び，$\mathcal{A}(S,P)$ と書く．

$$P(x) \leftarrow \neg contra, appli, S(x).$$
$$appli \leftarrow S(x), P(x).$$
$$contra \leftarrow S(x), \neg P(x).$$

上のスキーマにおいて，$appli$ は，ベース存在条件が成り立てば真であり，$contra$ は，類拠無矛盾条件が成り立たなければ真となる．$appli$ が真で，$contra$ が偽であれば，スキーマの第 1 条件は類拠規則と等しくなるので，このスキーマは一般化に基づく類推の基本形式を表しているといえる．

定義 6 K を正プログラム，S, P を K に現れる述語記号とする．S を類似性とする P に関する類推スキーマを K に付け加えた $K \cup \mathcal{A}(S,P)$ を K の**類推のための拡大**という．S と P が文脈から明らかなときは，$K \cup \mathcal{A}(S,P)$ を K_+ とかく．

ここで注意すべきなのは，K_+ は局所的層状 (locally stratified) プログラムでないということである．層状プログラム以外のプログラムに意味を与えることができるのは，有底モデル (well-founded model) と安定モデルがあるが，有底モデルでは，意図したモデルが複数ある場合にそこで共通に成り立つものを考えるのに対し，安定モデルでは，個々のモデルを考えることができる．類推スキーマから得られるモデルは，一般には下で示すように，複数ある意図したモデルのうちの一つであるので，安定モデルを本稿における類推を捕らえるために使っている．

以下の 2 つの定理は，類推モデルと類推のための拡大に対する安定モデルとの対応を示している．

定理 1 K を正プログラムであるとする．$appli \in M$ かつ $contra \notin M$ なる K_+ の安定モデル M が存在するならば，K は S を類似性として P を類推可能であり，$M - \{appli\}$ は，K の類推モデルである．

定理 2 K を正プログラムであるとする．K が S を類似性として P を類推可能な論理プログラムであり，M が K の類推モデルならば，$M \cup \{appli\}$ は K_+ の安定モデルである．

したがって，K における類推モデルを求めるかわりに，K_+ の安定モデル M を求め，$appli \in M$ かつ $contra \notin M$ なるものがあるときかつそのときにかぎり，$M - \{appli\}$ が，類推モデルとなる．

例 1 K が類推モデルをもっても，K_+ の安定モデルはその類推モデルに対応するモデルだけでない場合がありうる．たとえば，次の K を考えてみる．

$S(B).$
$P(B).$
$S(T).$

これに対する K_+ の安定モデルは，2 つ存在する．一方は，$\{S(B), P(B), S(T), P(T), appli\}$ であり，他方は，$\{S(B), P(B), S(T), appli, contra\}$ である．このうち前者の方が $appli$ を含み，$contra$ を含まないので類推モデルに対応する．これに対し，有底モデルだと，$P(T)$ が成り立つかどうかがわからなくなる．この例が示すように意図した複数のモデルの中から類推モデルを選べることが，安定モデルの特徴となっている．

4 類推の計算方法

ここでは，非決定的ボトムアップ手続きによる類推の計算方法を示す．停止性を保証するため，以下の制限を置く．

1. 定数記号のみで関数記号は存在しない．
2. range-restricted，すなわち，rule に関しては head に現れる変数は，body に必ず現れる．

この制限から，論理プログラムは，ボトムアップな手続きによってすべてground化できる．このとき安定モデルを計算する非決定手続きは以下のようになる．これは，[13]の命題論理プログラムに対する手続きを，上の制限のプログラムを扱えるように拡張したものである．

安定モデルを計算する手続き

K を一般論理プログラムとする．θ はルールまたは一貫性制約に存在するすべての変数への $HU(K)$ の代入である．

$i := 0,$
$M_0, \widetilde{M}_0 := propagate(\emptyset, \emptyset).$
If $M_0 \cap \widetilde{M}_0 \neq \emptyset$ then **fail**.

Step 1:

Select a rule R and θ in K such that $head(R\theta) \notin M_i$ and $pos(R\theta) \subseteq M_i$ and $(neg(R\theta) \cap M_i) = \emptyset$ and go to **Step 2**.
If such a rule is not found and there exists an integrity constraint C in K and θ s.t. M_i violates $C\theta$
then **fail** else **return** M_i.

Step 2:

$i := i + 1,$
$M_i, \widetilde{M}_i := propagate(M_{i-1} \cup \{head(R)\}, \widetilde{M}_{i-1} \cup neg(R))$
If $M_i \cap \widetilde{M}_i \neq \emptyset$ then **fail** else go to **Step 1**.

$propagate(M, \widetilde{M})$
begin
$k := 0,\ M_i^0 := M,\ \widetilde{M}_i^0 := \widetilde{M}.$
do
$k := k + 1,\ M_i^k := M_i^{k-1},\ \widetilde{M}_i^k := \widetilde{M}_i^{k-1}.$

For every rule R in K

1. If there is a θ such that $head(R\theta) \notin M_i^{k-1}$ and $pos(R\theta) \subseteq M_i^{k-1}$ and $neg(R\theta) \subseteq \widetilde{M}_i^{k-1}$, then add $head(R\theta)$ to M_i^k.
2. If there is a θ such that $head(R\theta) \in \widetilde{M}_i^{k-1}$ and there exists $P \in pos(R\theta)$ s.t. $P \notin M_i^{k-1}$ and $(pos(R\theta) - \{P\}) \subseteq M_i^{k-1}$, and $neg(R\theta) \subseteq \widetilde{M}_i^{k-1}$ then add P to $\widetilde{M}_i^k$.
3. If there is a θ such that $head(R\theta) \in \widetilde{M}_i^{k-1}$ and $pos(R\theta) \subseteq M_i^{k-1}$ and $neg(R\theta) \subseteq \widetilde{M}_i^{k-1}$, then **fail**.

For every integrity constraint C in K,

4. If there is a θ such that there exists $P \in C\theta$ s.t. $P \notin M_i^{k-1}$ and $(C\theta - \{P\}) \subseteq M_i^{k-1}$, then add P to $\widetilde{M}_i^k$.
5. If there is a θ such that $C\theta \subseteq M_i^{k-1}$, then **fail**.

until $M_i^k = M_i^{k-1}$ **and** $\widetilde{M}_i^k = \widetilde{M}_i^{k-1}$.
return $M_i^k, \widetilde{M}_i^k$
end

上で **select** の部分が非決定性を表しており，**fail** が直前の選択点までバックトラックすることを表す．この手続きは上で述べた制限を持つプログラムに対して，出力は安定モデルであり(健全)，すべての選択を尽くせばすべての安定モデルが得られる(完全)．証明は[13]を見よ．

例 2 K が次のようであるとする．

$S(B).$ (1)
$P(B).$ (2)
$S(T).$ (3)

K に次の一般化類推スキーマを加えたときの非決定的手続きの動きをルールの選択ごとに示す.

$P(x) \leftarrow \neg contra, appli, S(x).$ (4)
$appli \leftarrow S(x), P(x).$ (5)
$contra \leftarrow S(x), \neg P(x).$ (6)

Selection 1.

0. $M_0 = \{S(B), P(B), S(T), appli\}$, $\widetilde{M}_0 = \emptyset$

1. θ を $\{x = T\}$ として, ルール (6) を選ぶ.
   $M_1 = \{S(B), P(B), S(T), contra\}$, $\widetilde{M}_1 = \{P(T)\}$
2. これ以上選ぶルールがないので M_1 が安定モデルとして返される. このモデルは, $contra$ を含むので, 類推モデルに対応しない.

Selection 2.

0. $M_0 = \{S(B), P(B), S(T), appli\}$, $\widetilde{M}_0 = \emptyset$

1. θ を $\{x = T\}$ として, ルール (4) を選ぶ.
   $M_1 = \{S(B), P(B), S(T), appli, P(T)\}$, $\widetilde{M}_1 = \{contra\}$
2. これ以上選ぶルールがないので M_1 が安定モデルとして返される. このモデルは, $appli$ を含み, $contra$ を含まないので, $M_1 - \{appli\}$ は類推モデルとなる. またこのとき, 類推の結果として $P(T)$ が成り立っている.

例 3 K が次のようであるとする. (4) が一貫性制約であることに注意されたい.

$S(B).$ (1)
$P(B).$ (2)
$S(T).$ (3)
$\leftarrow R(T).$ (4)
$R(x) \leftarrow P(x).$ (5)

K に次の一般化類推スキーマを加えたときの非決定的手続きの動きをルールの選択ごとに示す.

$P(x) \leftarrow \neg contra, appli, S(x).$ (6)
$appli \leftarrow S(x), P(x).$ (7)
$contra \leftarrow S(x), \neg P(x).$ (8)

Selection 1.

0. $M_0 = \{S(B), P(B), S(T), appli, contra\}$, $\widetilde{M}_0 = \{R(T), P(T)\}$
1. これ以上選ぶルールがないので M_0 が安定モデルとして返される. このモデルは, $contra$ を含むので, 類推モデルに対応しない.

これ以外のルールの選択ができないので, 類推モデルは存在しないことになる. これは, (4) によって類拠無矛盾条件が成り立たないことによる.

例 4 類推モデルや類推スキーマの定義では，1引数述語のみを扱っているが，この制限は本質的なものではなく容易に拡張できる．これを以下の例で示す．K が次のようであるとする．

$$ApartFrom(Ball, Block). \tag{1}$$
$$ApartFrom(Planet, Sun). \tag{2}$$
$$ApartFrom(Electron, Neucleus). \tag{3}$$
$$Attracts(x,y) \leftarrow AstroHeavy(x), Object(y). \tag{4}$$
$$Attracts(x,y) \leftarrow PosElect(x), NegElect(y). \tag{5}$$
$$Attracts(x,y) \leftarrow NegElect(x), PosElect(y). \tag{6}$$
$$AstroHeavy(Sun). \tag{7}$$
$$Object(Planet). \tag{8}$$
$$PosElect(Neucleus). \tag{9}$$
$$NegElect(Electron). \tag{10}$$
$$Revolves(Planet, Sun). \tag{11}$$
$$\leftarrow Revolves(Ball, Block). \tag{12}$$

上記のプログラムに対して，非演繹的結論である $Revolves(Electron, Neucleus)$ (“電子は原子核の廻りを回っている”) が，いろいろな類似性候補に対して類推できるかどうかをみる．すなわち，$P(x,y)$ を $Revolves(x,y)$ とし，T を $\langle Electron, Neucleus\rangle$ とする．

例 4-1．$S(x,y)$ が $NegElect(x) \land PosElect(y)$ の場合:
K に対して以下の類推スキーマを付加する．

$$Revolves(x,y) \leftarrow \neg contra, appli, NegElect(x), PosElect(y). \tag{13}$$
$$appli \leftarrow NegElect(x), PosElect(y), Revolves(x,y). \tag{14}$$
$$contra \leftarrow NegElect(x), PosElect(y), \neg Revolves(x,y). \tag{15}$$

このときの非決定的手続きの動きをルールの選択ごとに示す．

Selection 1.

0. $M_0 = \{$
$ApartFrom(Ball, Block), ApartFrom(Planet, Sun), ApartFrom(Electron, Neucleus),$
$AstroHeavy(Sun), Object(Planet), PosElect(Neucleus), NegElect(Electron),$
$Revolves(Planet, Sun), Attracts(Sun, Planet),$
$Attracts(Neucleus, Electron), Attracts(Nuecleus, Electron)\},$
$\widetilde{M_0} = \{Revolves(Ball, Block)\}$

1. θ を $\{x = Electron, y = Neucleus\}$ として，ルール (15) を選ぶ．
$M_1 = M_0 \cup \{contra\}$, $\widetilde{M_1} = \widetilde{M_0} \cup \{Revolves(Electron, NeuCleus)\}$.
これ以上選ぶルールがないので M_1 が安定モデルとして返される．このモデルは，$contra$ を含むので，類推モデルに対応しない．

これ以外のルールの選択ができないので，類推モデルは存在しないことになる．これは，類拠規則に対して $(NegElect(Electron) \land PosElect(Nuecleus)) \land \neg Revolves(Electron, Nuecleus)$: (“負電荷を持つ Electron が正電荷を持つ Nuecleus の廻りを回っていない”) なる反例が存在し，類拠無矛盾条件が満たされないことが原因である．

例 4-2．$S(x,y)$ が $ApartFrom(x,y)$ の場合:
K に対して以下の類推スキーマを付加する．

$$Revolves(x,y) \leftarrow \neg contra, appli, ApartFrom(x,y). \tag{16}$$
$$appli \leftarrow ApartFrom(x,y), Revolves(x,y). \tag{17}$$
$$contra \leftarrow ApartFrom(x,y), \neg Revolves(x,y). \tag{18}$$

このときの非決定的手続きの動きをルールの選択ごとに示す.

Selection 1.

0. $M_0 = \{$
$ApartFrom(Ball, Block), ApartFrom(Planet, Sun), ApartFrom(Electron, Neucleus),$
$AstroHeavy(Sun), Object(Planet), PosElect(Neucleus), NegElect(Electron),$
$Revolves(Planet, Sun), Attracts(Sun, Planet),$
$Attracts(Neucleus, Electron), Attracts(Nuecleus, Electron), appli, contra\}$,
$\widetilde{M}_0 = \{Revolves(Ball, Block)\}$

1. これ以上選ぶルールがないので M_0 が安定モデルとして返される. このモデルは, $contra$ を含むので, 類推モデルに対応しない.

これ以外のルールの選択ができないので, 類推モデルは存在しないことになる. これは, 類推規則に対して $ApartFrom(Ball, Block) \wedge \neg Revolves(Ball, Block)$ ("$Ball, Block$ は離れたままであるが, $Ball$ は $Block$ のまわりを回っていない") なる反例が存在し, 類推無矛盾条件が満たされないことが原因である.

例 4-3. $S(x, y)$ が $ApartFrom(x, y) \wedge Attracts(y, x)$ の場合:
K に対して以下の類推スキーマを付加する.

$$Revolves(x, y) \leftarrow \neg contra, appli, ApartFrom(x, y), Attracts(y, x). \tag{19}$$
$$appli \leftarrow ApartFrom(x, y), Attracts(y, x), Revolves(x, y). \tag{20}$$
$$contra \leftarrow ApartFrom(x, y), Attracts(y, x), \neg Revolves(x, y). \tag{21}$$

このときの非決定的手続きの動きをルールの選択ごとに示す.

Selection 1.

0. $M_0 = \{$
$ApartFrom(Ball, Block), ApartFrom(Planet, Sun), ApartFrom(Electron, Neucleus),$
$AstroHeavy(Sun), Object(Planet), PosElect(Neucleus), NegElect(Electron),$
$Revolves(Planet, Sun), Attracts(Sun, Planet),$
$Attracts(Neucleus, Electron), Attracts(Nuecleus, Electron), appli\}$,
$\widetilde{M}_0 = \{Revolves(Ball, Block)\}$

1. θ を $\{x = Electron, y = Neucleus\}$ として, ルール (21) を選ぶ.
$M_1 = M_0 \cup \{contra\}$, $\widetilde{M}_1 = \widetilde{M}_0 \cup \{Revolves(Electron, NeuCleus)\}$.
これ以上選ぶルールがないので M_1 が安定モデルとして返される. このモデルは, $contra$ を含むので, 類推モデルに対応しない.

Selection 2.

0. $M_0 = \{$
$ApartFrom(Ball, Block), ApartFrom(Planet, Sun), ApartFrom(Electron, Neucleus),$
$AstroHeavy(Sun), Object(Planet), PosElect(Neucleus), NegElect(Electron),$
$Revolves(Planet, Sun), Attracts(Sun, Planet),$
$Attracts(Neucleus, Electron), Attracts(Nuecleus, Electron), appli\}$,
$\widetilde{M}_0 = \{Revolves(Ball, Block)\}$

1. θ を $\{x = Electron, y = Neucleus\}$ として, ルール (19) を選ぶ.
$M_1 = M_0 \cup \{Revolves(Electron, NeuCleus)\}$, $\widetilde{M}_1 = \widetilde{M}_0 \cup \{contra\}$.
これ以上選ぶルールがないので M_1 が安定モデルとして返される. このモデルは, $appli$ を含み $contra$ を含ま

ないので，$M_1-\{appli\}$ は類推モデルとなる．またこのとき，類推の結果として $Revolves(Electron, NeuCleus)$ が成り立っている．これは，類拠規則，

$$(\forall x, y.\ Attracts(y,x) \wedge ApartFrom(x,y) \supset Revolves(x,y))$$

（"y が x を引っ張っており，離れたままであるとすると，x は y の廻りを回っている"）が基本形式に基づき得られ，

$$Attracts(Neucleus, Electron) \wedge ApartFrom(Electron, Neucleus)$$

（"原子核は電子を引っ張っており，電子は原子核から離れたままである"）が演繹されるので，結局，$Revolves(Electron, Neucleus)$ が導出されることに対応している．

5 関連研究との比較－折原らの方法

安定モデル（正確には，generalized stable model）に基づく類推を提案した先駆的研究として，[10] がある．そこでは，一貫性制約のない局所層状プログラム上での類推を扱っている．彼らの形式化は，我々の場合と同様に，元のプログラムに類推のためのルールを付加する方法である．

[10] の形式化と我々の形式化の大きな相違点は次の 2 点である．

1. 否定情報の扱い

   [10] では局所層状プログラムを類推の対象としており，1 節で述べたように，局所層状プログラムでは論理的否定事実を記述することはできない．そのため，望ましい類推モデルを選ぶためにはなんらかの外部からの知識が必要である．我々の形式化では一貫性制約を用いることにより，不必要な類推を抑制することが可能となった．

2. 目的とする類推

   [10] では，なるべく多くの事実を類推によって導き出すために，類似性と投射性を特に区別しない点である．例えば，$\{S(B), P(B), S(T_1), P(T_2)\}$ に対して，$\{S(B), P(B), S(T_1), P(T_1), S(T_2), P(T_2)\}$ は [10] で意図された類推モデルであるが，我々の意図したモデルは $\{S(B), P(B), S(T_1), P(T_1), P(T_2)\}$ で，ターゲット T_2 に関する事実 $S(T_2)$ は類推されない．

上記の 2 に関し，さらに考察する．[10] では，プログラム K

$S(B).$
$P(B).$
$S(T).$

に対して，以下のルール及び一貫性制約が付加される．

$C(x) \leftarrow S(x), P(x).$ (1)
$S(x) \leftarrow C(x).$ (2)
$P(x) \leftarrow C(x).$ (3)
$A_S(x) \leftarrow P(x), S^*(x).$ (4)
$A_P(x) \leftarrow S(x), P^*(x).$ (5)
$C(x) \leftarrow P(x), A_S^*(x).$ (6)
$C(x) \leftarrow S(x), A_P^*(x).$ (7)
$\leftarrow S(x), S^*(x).$ (8)
$\leftarrow P(x), P^*(x).$ (9)
$\leftarrow A_S(x), A_S^*(x).$ (10)
$\leftarrow A_P(x), A_P^*(x).$ (11)
$S(x) \vee S^*(x).$ (12)
$P(x) \vee P^*(x).$ (13)
$A_S(x) \vee A_S^*(x).$ (14)
$A_P(x) \vee A_P^*(x).$ (15)

以下のルール

$P(x) \leftarrow S(x), A^*_P(x).$ (3) を (7) で展開したルール

$A_P(x) \leftarrow S(x), P^*(x).$ (5)

は, $A_P(x)$ を $contra$ とみなす (P^* は P の否定を表す) と, 我々の一般化類推スキーマにおける第1条件と第3条件に対応すると考えることができる.

そこで, 我々の類推スキーマに変更を加えた次のような類推スキーマ $\mathcal{A}'(S,P)$ について考える.

$$P(x) \leftarrow \neg contra(x), appli, S(x).$$

$$appli \leftarrow S(x), P(x).$$

$$contra(x) \leftarrow S(x), \neg P(x).$$

すると, $K \cup \mathcal{A}'(S,P) \cup \mathcal{A}'(P,S)$ によって, [10] の意図する類推が計算できるのではないかと考える.

6 今後の課題

今後の課題として, 以下のことが挙げられる.

1. 1つのプログラムに対して, 複数のソースおよびターゲットの組み合わせについて同時に類推を行う場合についての考察.
2. 対象とするプログラムの, 一般論理プログラムへの拡張.
3. ある事実が類推モデルで成り立つかどうかという, 問い合わせに対する Top-down 手続きの検討 ([8] の, 一貫性制約つきの一般論理プログラムに対する Top-down 手続きが利用できると思われる).
4. 本稿で述べたテクニックの, 他の類推の枠組 (例えば, 例証的基準に基づく類推 [3]) への応用.

Appendix

定理 1の証明: K_+ の安定モデル M は, K_+^M すなわち, $K \cup \{P(t) \leftarrow appli, S(t) | t \in HU(K)\} \cup \{appli \leftarrow S(u), P(u) | u \in HU(K)\} \cup \{contra \leftarrow S(v) | v \in HU(K)$ and $P(v) \notin M\}$ の最小モデルに一致し, かつ K 中の一貫性制約に矛盾しない.

したがって, $M - \{appli\}$ は K のモデルである. このとき, K が類推可能でないとする.

1. $min(K) \not\models S(t) \wedge P(t)$ なる $t \in HU(K)$ が存在しないとき.
   $appli \in M$ であるから, $appli$ は K_+^M から導かれねばならないが, K からは, $S(u) \wedge P(u)$ を導くことができず, 他のルールも $S(u) \wedge P(u)$ を導くことができないので, 矛盾である.
2. すべての $t \in HU(K)$ に対して $M \models S(t) \rightarrow P(t)$ なる K のモデル M が存在しないとき.
   $M \models S(t) \wedge \neg P(t)$ なる $t \in HU(K)$ が存在するので $contra$ が導かれ, $contra \notin M$ に矛盾する.

appli が M で真であるので, K_+^M の最初の式から $P(x) \leftarrow S(x)$ が導かれ, また, M は K_+^M の最小モデルになっているため $M - \{appli\}$ は $K \cup \{P(x) \leftarrow S(x)\}$ の最小モデルとなっており, したがって類推モデルともなっている. □

定理 2の証明:

K の類推モデルを M とする.

$m = M \cup \{appli\}$ が, K_+ の stable モデルであること, すなわち, $m = min(K_+^m)$ を示す.

contra は K に含まれない述語記号なので, $contra \notin m$ となり, $K_+^m = K \cup \{P(t) \leftarrow appli, S(t) | t \in HU(K)\} \cup \{appli \leftarrow S(u), P(u) | u \in HU(K)\} \cup \{contra \leftarrow S(v) | v \in HU(K)$ and $P(v) \notin M\}$ となる.

1. $M \subset m$ より, m は K を満たす.

2. すべての $t \in HU(K)$ に対して, $M \models S(t) \rightarrow P(t)$ より, $m(= M \cup \{appli\})$ は $\{P(t) \leftarrow appli, S(t) | t \in HU(K)\}$ を満たす.

3. $appli \in m$ より, m は $\{appli \leftarrow S(u), P(u) | u \in HU(K)\}$ を満たす.

4. すべての $t \in HU(K)$ に対して, $M \models S(t) \rightarrow P(t)$ より, $P(v) \notin M$ のとき $S(v) \notin M$ であるから, m は $\{contra \leftarrow S(v) | v \in HU(K)$ and $P(v) \notin M\}$ を満たす.

したがって, m は K^m_+ のモデルになっている.

また, M が $K \cup \{P(x) \leftarrow S(x)\}$ の最小モデルとなっており, ある $t \in HU(K)$ が存在して $min(K) \models S(t) \wedge P(t)$ より $appli \in min(K^m_+)$ となっているため, 少なくとも m で真であるものが, $min(K^m_+)$ に含まれているため, $min(K^m_+)$ の最小性より $m = min(K^m_+)$ となる.

さらに, M が K のモデルになっていることから m は K 中の一貫性制約に違反しないので, m は K_+ の安定モデルとなる. □

参考文献

[1] 有馬 淳, 含意限定: 非単調推論による高次推論の一形式, 情報処理学会論文誌, Vol. 30, No. 11, pp. 1424 -1433 (1989).

[2] 有馬 淳, 類推の形式化に関する考察 (研究メモ), ICOT-TM706 (1989).

[3] Arima,J., A Logical Analysis of Relevance in Analogy, to appear in *Proc. of ALT'91* (1991).

[4] Eshghi, K., Kowalski, R. A., Abduction Compared with Negation by Failure, *Proc. of ICLP'89*, pp. 234 254 (1989).

[5] Davies,T., Russel, S.J., A Logical Approach to Reasoning by Analogy, *Proc. of IJCAI-87* pp. 264 270 (1987).

[6] Gelfond, M., Lifschitz, V., The Stable Model Semantics for Logic Programming, *Proc. of LP'88*, pp. 1070 - 1080 (1988).

[7] 原口 誠, 有川 節夫, 類推の定式化とその実現, 人工知能学会誌, Vol.1, No.1, pp. 132 - 139 (1986).

[8] Kakas, A. C., Mancarella, P., On the Relation between Truth Maintenance and Abduction, *Proc. of PRICAI'90*, pp. 438 – 443 (1990).

[9] Keder-Cabelli,S., Purpose-Directed Analogy, *Proc. of the 7th Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 150 – 159 (1985).

[10] 折原良平, 大須賀昭彦, 楠井洋一, 論理プログラムの類推的モデル, 情報処理学会研究報告, 91-AI-75 (1991).

[11] Przymusinski,T.C., On the Declarative Semantics of Deductive Databeses and Logic Programs, *Foundations of Deductive Databeses and Logic Programs* (J. Minker, ed.), Morgan Kaufman, Chapter 5, pp. 193 – 216 (1988).

[12] Sadri. F., Kowalski, R., A Theorem-Proving Approach to Database Integrity, *Foundations of Deductive Databeses and Logic Programs* (J. Minker, ed.), Morgan Kaufman, Chapter 9, pp. 313 362 (1988).

[13] Satoh, K., Iwayama, N., Computing Abduction by Using the TMS, *Proc. of ICLP'91*, pp. 505 – 518 (1991).