ICOT Technical Memorandum: TM-0706

TM-0706

類推の形式化に関する考察(研究メモ)

有馬　淳

March, 1989

ICOT

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

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

類推の形式化に関する考察 (研究メモ)

ICOT 第一研究室
有馬 淳

1. 類推であるための必要条件

本研究では，類推が持つさまざまな特徴を探ることから始めて，その特徴を満足する類推の形式化を目指す.

1) 類推の特徴は概して以下のスキーマで表されるであろう.

$$\frac{P(s) \wedge Q(s)}{P(t)} \qquad \overline{\qquad Q(t)}$$

ここで，s, t はそれぞれsource，targetと呼ばれる，特殊な個体であり，tと類似した性質Pをsが持っているという事実からsの持っている性質Qをtも持つと推論する.

しかし，上記スキーマによる推論は類推の疑似的な表現でしかない．ある形式が，類推の形式化として適当であるためには，類推の持つ性質として認められるさらに多くの条件を，満たされなければならないと考える．現在，その条件の完全なリストは持ち合わせていないが，部分的に以下の条件をあげることができる.

2) 類推は無矛盾である(演えきが類推に優先する).
類推の結果は，事実と無矛盾でなければならない.

EX1) ''sとtはともにリンゴである．また，sは赤く，tは赤くない''，という事実Aから，

''sとtはともにリンゴである''という点で似ている．''sは赤い''，よって，''tも赤い''，

といった類推を我々は行なわない．つまり，$A \models Q$ によって，QがAの論理的帰結であることを表すとすると，前提Aに対しQが類推の結果を表すためには

a) $A \not\models \neg Q$

であることが必要である．さらに，targetの持つ性質が分かっている場合に，類推を働せることはないという立場に立つと，さらに

b) $A \not\models Q$

でなければならない. a) をconsistency condition, b) をunknown conditionと呼ぶことにする. unknown conditionは論理的帰結を導く演えきとの違いを明確にするが, 類推の結果と演えきによる結果を区別する必要のない場合, 演えき推論システムの類推による強化, 拡張こそが重要である場合はunknown conditionを無視できる. 一方, concistency conditionは論理を基にした推論システムでは欠くことのできない重要な条件である. 類推は非演えき的な推論であるがゆえに, Qが現在の知識Aに無矛盾な推論結果であっても, 新たな知識の増加によって, Qが矛盾する場合がある. この場合は, Qを推論結果としないようにしなければならない. このことは類推が非単調であることを示す(実際, 非演えき的な推論はすべて同じ理由により非単調にならざるをえない.) 非単調であることはシステムの仮説を含めた知識全体の健全な拡大に重要な役割を果たしている. Qを仮説だとしてみると仮説の生成, 消滅の過程は論理における''淘汰''と考えることができる. 事実が単調に増加するにつれ, これまでの知識に無矛盾な仮説だけがもっともらしさを増して''生き残る''のである. この考え方は, 非演えきシステムである, 枚挙による帰納推論の分野でもみつけることができる. その分野では, 帰納推論が生成した仮説のもっともらしさの度合は, 例外なしにその仮説の正の例が増せば増すほど高まると考えている.

2) 問題解決指向であること

targetに関する知識Q(t)かどうかを知りたい, すなわち, Q(t)が類推として説明可能かどうかを調べる. Q(t)を説明する無矛盾な仮説は一般に無数にある. そのうちで何をもっともらしい仮説とし選択するかという問題は未だに残されている. ''人間の知りたいことが重要な定理である''と仮定すると, 問題解決を契機としてはじめて推論を働かせる方法は, 推論過程で生じる仮説間に一種の重要さの尺度を持ち込むことができる.

この原理は, 類推に限らず他の非演えき的な推論仮説選択の指針として一般的なものであるかも知れない.

さて, 類推が問題解決指向であることは, 副次的に次のことを要求する.

''sourceがあらかじめ与えられるのではなく, 類推の過程で適当なsourceが選択され, そのsourceをもとに類推が行なわれることを説明できねばならない''.

類推を問題解決の手段としてみる時, sourceがあらかじめ与えられるという前提をとることはできない. 与えられるものはQ(t)のみであり, これを既存の知識から類推によって引き出すことを表現できる形式を求めなければならない.

sourceに関係した条件としてはa.1)がある.
a.1) T.R.Davies & S.J.Russel は類推にはtargetの存在だけでなく, sourceの存在が欠かせないことをthe non-redundancy problemという問題として指摘した[2]. つまり, sourceによって与えられる情報が推論に影響を及ぼ

す事実が説明されなければならないと主張した.

ここではさらに, sourceは与えられないという観点から次のことを説明するものでなければならないと考える.

a.2) ''得たい結論がsourceの選択に影響する''

EX2) ''ある日, ある所で, 動物たちが掃除をすることになった. 掃除はほ乳類チームと鳥類チームが交互に行なうことになった. 初日, ほ乳類チームの掃除の番で, ライオンのs1がコウモリであるtをさそった. tはカナリアのs2と類比をとった. '私とs2は飛ぶという点で似ている. よってわたしは鳥類に属すので, 今日は掃除をしない. ' 次の日, 鳥類チームの掃除の番で, カナリアのs2がtをさそった. tはライオンのs1と類比をとった. '私とs1は卵を生まないという点で似ている. よってわたしはほ乳類に属すので, 今日は掃除をしない. '. . . ''

この例は, 得たい結論がsourceの選択に影響し, 類比の取り方が異なってくることを示していると考える. この形式化では, target に対し特定のsourceを与えることを前提としない. sourceは問題解決の過程で見つけられ利用されるべきものである.

3) Similarity PとProjected property Q の間の関連を明らかにせねばならない.

T.R.Davies & S.J.Russel は次の例を示し, 類似性Pと, sourceからtargetに投影(project)される性質Qの間の関係によって, 類推のもっともらしさが変わることを示した.

EX3) BobとJohnはともに1982年ムスタングGLX V6 ハッチバック車を持っている. Johnの車体の色が赤だからと言って, Bobの車の色が赤だとは我々は推論しない. しかしながら, Jhonの車が約$3500だったという事実からは, Bobの車も$3500程度であろうと推測するのである.

この例は同じ前提からの推論が, 前者の結論はあやしく, 後者の結論はありそうであることを示している. もっともらしい結論を導くために, PとQの間になんらかの制約をもたねばならない.

4) Similarity は制限されなければならない.
''類似性は無数に見つけることができる. ''次の例を考えよう.

EX4) Peterはうさぎである. Psiはコンピューターである. 以下の理由で2者は似ている.

1) ともに，PeterかPsiであるという性質(x=Peter v x=Psi)を持つ.
2) ともに，PeterかPsiかJunであるという性質(x=Peter v x=Psi v x=Jun)を持つ.

. . .

3) ともに，うさぎかコンピューターであるという性質を持つ.

. . .

以上の理由でPeterとPsiは似ているからといってPsiはうさぎであるなどとは推論しない.

この事実は 2者間の類似性であろうと(有限)多数者間での類似性であろうと本質的に変わりがないことに注意すべきである．類似性が無数に見いだしうるにもかかわらず，われわれはいくつかの決まった類似性しか見い出していないように思える．そこには人間の好み，個々の特性，意図，あるいは偶然性さえも関与しているようにおもえる．問題は類似性の候補を絞り込む一般的なヒューリスティックの発見であろう.

この問題の多少アドホックな解決法は言語の制限にある．例題に関して言えば，選言記号(∨)を含んだ性質は類似性にはならないというものが考えられるが，その場合，もし連言記号(∧)による類似性の表現は許すと，否定(¬)，含意(⊃)等も除外しなければならなくなる．これはあまりにきゅうくつである．実は例の3)は論理的淘汰にまかせられる．1)はtargetの投射される性質について実際に分からない限り論理淘汰されないため，1)の除外こそがもっとも重要である．このためには次の仮定をおくことも考えられる．“等号を含んだ性質は類似性にはならない”

2. 類推の形式化 (暫定版)

[定義] 類推含意推定(analogical ascription) (仮名)
Aをιが現れる与えられた論理式，Qを述語記号，Pを述語の組，ZをAに現れるQ以外のすべて述語記号または関数記号の組，tをtargetを表す個体記号とする．またA[r/R]はAにおける述語記号または関数記号の組Rを対応するrの要素で置き換えたものとすると

A における Q(ι) の P への analogical ascription, $Anl-asc(A;Q(t);P)$は

$$Anl-asc(A;Q(t);P) \equiv$$
$$A \wedge$$
$$\forall p \in P.(p(t) \wedge \exists x.(p(x) \wedge Q(x)) \wedge \exists z.A[\lambda x.(p(x) \vee Q(x))/Q; z/Z]$$
$$\supset \forall x.(p(x) \supset Q(x)))\ ^{1}$$

で表される.

analogical ascriptionが1. で述べた必要条件をいかに満足するかをみる.

[1] ascription[1] において，前提条件を強化し，含意限定する対象 (Ψ) を制限したものになっている.

EX1)

$A \equiv Apple(s) \wedge Red(s) \wedge Apple(t) \wedge \neg Red(t))$

Red(t)がtもsもAppleと言う点でにているからと言う理由で導けるかどうかをみる.

$Anl - asc(A; Red(t); Apple) \equiv$

$A \wedge (Apple(t) \wedge \exists x.(Apple(x) \wedge Red(x)) \wedge A[\lambda x.(Apple(x) \vee Red(x))/Red]$

$\supset \forall x.(Apple(x) \supset Red(x))).$

ここで,

$A[\lambda x.(Apple(x) \wedge Red(x))/Red] \equiv$

$Apple(s) \wedge (Apple(s) \vee Red(s)) \wedge Apple(t) \wedge \neg(Apple(t) \vee Red(t))$

より偽. すなわち,

$Anl - asc(A; Red; Apple) \equiv A$

となる. 明らかにAからの論理的帰結しか得られないので, Red(t)は導かない. (実際, Aが無矛盾な論理式である時, analogical ascriptionは無矛盾であることが証明できる.)

EX2) 次のように簡単化する

$A \equiv Bat(t) \wedge Fly(t) \wedge \neg YieldEggs(t)$

$\wedge Lion(s1) \wedge Mammal(s1) \wedge \neg YieldEggs(s1)$

$\wedge Canary(s2) \wedge Bird(s2) \wedge Fly(s2)$

tの持つ性質からP として$Bat, Fly, \lambda x.(\neg YieldEggs(x)), ...$が考えられる. ここで, コウモリは初日には鳥類であることを主張する. Pに対してanalogical ascriptionをおこなうと,

$Anl - Asc(A; Bird(t); Bat, Fly, \lambda x.(\neg YieldEggs(x)), ...)$

$\vdash \exists x.(Bird(x) \wedge Bat(x)) \supset \forall x.(Bat(x) \supset Bird(x))$

$\vdash \exists x.(Bird(x) \wedge Fly(x)) \supset \forall x.(Fly(x) \supset Bird(x))$

$\vdash \exists x.(Bird(x) \wedge \neg YieldEggs(x)) \supset \forall x.(\neg YieldEggs(x) \supset Bird(x))$

. . .

2つめの結論の前件, $\exists x.(Bird(x) \wedge Fly(x))$, はs2の存在によって, 満足されるから, 結論, $\forall x.(Fly(x) \supset Bird(x))$, がanalogical ascriptionによって導かれることになる. すなわち, 飛ぶコウモリは鳥である.

次の日にはコウモリは哺乳類であることを主張する. 同様にして, s1の存在によって,

$Anl - Asc(A; Mammal(t); Fly, \lambda x.(\neg YieldEggs(x)), ...)$

$\vdash \forall x.(\neg YieldEggs(x) \supset Mammal(x)).$

これは類比のとり方で結論が変わることを説明している.

EX3)

$A \equiv M(Cb) \wedge M(Cj) \wedge Red(Cb) \wedge \$3500(Cb)$

Aにおけるanalogical ascriptionからは

$\forall x.(M(x) \supset Red(x)),$

$\forall x.(M(X) \supset \$3500(x))$

が両者とも導きうるので, Johnの車が赤いということも$3500であることも同様に確からしい. これはEX3の問題を解決しないようにみえるが, 論理的淘汰の考え方に基

づけばさほど問題ではないように思われる．われわれが前者の結論がおかしく感ずるのはムスタングで青い車や白い車，つまり赤い車でないものが存在することを知っているせいである．約$3500であることが奇異でないのはその例外を知らないからである．(若干の例外が存在しても多数が満足される仮説はもっともらしいとも言えるが，本考察ではその問題を扱わない．)実際，もしムスタングで赤でない車を知っていると，$A \wedge \exists x.(M(x) \wedge \neg Red(x))$, analogical ascriptionは先の“ムスタング車であれば赤い”，$\forall x.(M(x) \supset Red(x))$,という結論は取り下げられる．

EX4)
analogical ascriptionでは類似性の候補Pを与える．これにより，人間のヒューリスティクスを与えることができる．例えば，Pの中に等号は含まれないように設定する．

3．今後の課題

analogical ascriptionは無矛盾な仮説のうち，妥当な仮説の一候補を与えていると考える．この点で帰納推論の研究にも示唆を与えることができると考えている．

類推としての必要条件のリストをより完全なものにしたい．

[1]Arima,J.: Ascription: An Approach to Formalization of Non-monotonic Reasoning in the Wider Sence, WOL'87.(現在改訂中)
[2]Davies,T.R & Russel,S.J.: A Logical Approach to Reasoning by Analogy, IJCAI-87.
[3]Greiner,R.: Learning by Understanding Analogies, AI 35,1988.

付録

以下に実行例を示す．テスト番号は本文中の各例題にほぼ対応する．一階述語 prover はICOTで開発されたもの(by坂井公)を使用.

```
test1 :-
        A1 = (apple(s) /\ red(s) /\ apple(t) /\ ~red(t)),
        anlasc(A1,red(t),[lambda([x],apple(x))]).

test2a :-
        A2 = (bat(t) /\ fly(t) /\ ~yield-eggs(t)
           /\ lion(s1) /\ mammal(s1) /\ ~yield-eggs(s1)
           /\ canary(s2) /\ bird(s2) /\ fly(s2) ),
        Sim = [ lambda([x],fly(x)),
                lambda([x],~fly(x)),
                lambda([x],yield-eggs(x)),
                lambda([x],~yield-eggs(x))],
        anlasc(A2,bird(t),Sim).

test2b :-
        A2 = (bat(t) /\ fly(t) /\ ~yield-eggs(t)
           /\ lion(s1) /\ mammal(s1) /\ ~yield-eggs(s1)
           /\ canary(s2) /\ bird(s2) /\ fly(s2) ),
        Sim = [ lambda([x],bat(x)),
                lambda([x],fly(x)),

                lambda([x],yield-eggs(x)),
                lambda([x],~yield-eggs(x))],
        anlasc(A2,mammal(t),Sim).


test3 :-
        A3 = ( mustang(car_b) /\ mustang(car_j)
            /\ red(car_b)
            /\ d3500(car_b) ),
        Sim = [lambda([x],mustang(x))],
        anlasc(A3,red(car_j),Sim),
        anlasc(A3,d3500(car_j),Sim).
```

```
LINK FROM ARIMA, TTY 332
[PH: Initiation. 15-Mar-89  6:58PM]

 TOPS-20 Command processor 6.1(3067)
@prove

yes
| ?- [anl].

anl consulted   950 words      0.53 sec.

yes
| ?- [anltes].

anltes consulted   546 words      0.28 sec.

yes
| ?- test1.

Analogical Ascription is
(apple(t)/\some([x],apple(x)/\red(x))/\apple(s)/\(apple(s)\/red(s))/\apple(t)/\
 ~(apple(t)\/red(t))->all([x],apple(x)->red(x)))/\apple(s)/\red(s)/\apple(t)/\ ~r
ed(t)
no
| ?- test2a.

Analogical Ascription is
(fly(t)/\some([x],fly(x)/\bird(x))/\bat(t)/\fly(t)/\ ~yield-eggs(t)/\lion(s1)/\m
ammal(s1)/\ ~yield-eggs(s1)/\canary(s2)/\(fly(s2)\/bird(s2))/\fly(s2)->all([x],f
ly(x)->bird(x)))/\(~fly(t)/\some([x],~fly(x)/\bird(x))/\bat(t)/\fly(t)/\ ~yield-
eggs(t)/\lion(s1)/\mammal(s1)/\ ~yield-eggs(s1)/\canary(s2)/\(~fly(s2)\/bird(s2)
)/\fly(s2)->all([x],~fly(x)->bird(x)))/\(yield-eggs(t)/\some([x],yield-eggs(x)/\
bird(x))/\bat(t)/\fly(t)/\ ~yield-eggs(t)/\lion(s1)/\mammal(s1)/\ ~yield-eggs(s1
)/\canary(s2)/\(yield-eggs(s2)\/bird(s2))/\fly(s2)->all([x],yield-eggs(x)->bird(
x)))/\(~yield-eggs(t)/\some([x],~yield-eggs(x)/\bird(x))/\bat(t)/\fly(t)/\ ~yiel
d-eggs(t)/\lion(s1)/\mammal(s1)/\ ~yield-eggs(s1)/\canary(s2)/\(~yield-eggs(s2)\
/bird(s2))/\fly(s2)->all([x],~yield-eggs(x)->bird(x)))/\bat(t)/\fly(t)/\ ~yield-
eggs(t)/\lion(s1)/\mammal(s1)/\ ~yield-eggs(s1)/\canary(s2)/\bird(s2)/\fly(s2)
Time Used : 5539msec = 5.539sec

  |fly(t)|all([x],fly(x)->bird(x))| => |bird(t)|
    o |fly(s2)|bird(s2)| => |some([x],fly(x)/\bird(x))|

Analogical Ascription Succeed

yes
| ?- test2b.

Analogical Ascription is
(bat(t)/\some([x],bat(x)/\mammal(x))/\bat(t)/\fly(t)/\ ~yield-eggs(t)/\lion(s1)/
\(bat(s1)\/mammal(s1))/\ ~yield-eggs(s1)/\canary(s2)/\bird(s2)/\fly(s2)->all([x]
,bat(x)->mammal(x)))/\(fly(t)/\some([x],fly(x)/\mammal(x))/\bat(t)/\fly(t)/\ ~yi
eld-eggs(t)/\lion(s1)/\(fly(s1)\/mammal(s1))/\ ~yield-eggs(s1)/\canary(s2)/\bird
(s2)/\fly(s2)->all([x],fly(x)->mammal(x)))/\(yield-eggs(t)/\some([x],yield-eggs(
x)/\mammal(x))/\bat(t)/\fly(t)/\ ~yield-eggs(t)/\lion(s1)/\(yield-eggs(s1)\/mamm
al(s1))/\ ~yield-eggs(s1)/\canary(s2)/\bird(s2)/\fly(s2)->all([x],yield-eggs(x)-
>mammal(x)))/\(~yield-eggs(t)/\some([x],~yield-eggs(x)/\mammal(x))/\bat(t)/\fly(
t)/\ ~yield-eggs(t)/\lion(s1)/\(~yield-eggs(s1)\/mammal(s1))/\ ~yield-eggs(s1)/\
canary(s2)/\bird(s2)/\fly(s2)->all([x],~yield-eggs(x)->mammal(x)))/\bat(t)/\fly(
t)/\ ~yield-eggs(t)/\lion(s1)/\mammal(s1)/\ ~yield-eggs(s1)/\canary(s2)/\bird(s2
)/\fly(s2)
Time Used : 5371msec = 5.371sec

  |~yield-eggs(t)|all([x],~yield-eggs(x)->mammal(x))| => |mammal(t)|
    o |mammal(s1)| => |yield-eggs(s1)|some([x],~yield-eggs(x)/\mammal(x))|
```

```
Analogical Ascription Succeed

yes
| ?- test3.

Analogical Ascription is
(mustang(car_j)/\some([x],mustang(x)/\red(x))/\mustang(car_b)/\mustang(car_j)/\(
mustang(car_b)\/red(car_b))/\d3500(car_b)->all([x],mustang(x)->red(x)))/\mustang
(car_b)/\mustang(car_j)/\red(car_b)/\d3500(car_b)
Time Used : 102msec = .102sec

  |mustang(car_j)|all([x],mustang(x)->red(x))| => |red(car_j)|
    o |red(car_b)|mustang(car_b)| => |some([x],mustang(x)/\red(x))|

Analogical Ascription Succeed

Analogical Ascription is
(mustang(car_j)/\some([x],mustang(x)/\d3500(x))/\mustang(car_b)/\mustang(car_j)/
\red(car_b)/\(mustang(car_b)\/d3500(car_b))->all([x],mustang(x)->d3500(x)))/\mus
tang(car_b)/\mustang(car_j)/\red(car_b)/\d3500(car_b)
Time Used : 113msec = .113sec

  |mustang(car_j)|all([x],mustang(x)->d3500(x))| => |d3500(car_j)|
    o |d3500(car_b)|mustang(car_b)| => |some([x],mustang(x)/\d3500(x))|

Analogical Ascription Succeed

yes
| ?- halt.

[ Prolog execution halted ]

EXIT
@pop
[PH: Termination. RRR..2. 15-Mar-89  7:06PM]
```