人　工　知　能　75−14
（1 9 9 1.　3.　7）

高階一般化による類推

原尾　政輝

九州工業大学情報工学部 知能情報工学科
harao@dumbo.ai.kyutech.ac.jp

科学的問題を対象とした類推を高階論理の枠組で実現する一手法について述べる．まず，対象とする知識を一階の論理式で，その抽象化された知識である一般化知識を二階の論理式で表現する．対象知識の間の類似性を単一化可能性によって，例題や観測された事実から一般化規則を求める一般化を高階表現への抽象化によって，それぞれ定式化する．与えられた問題に対する類推は，その問題と類似の解法の分かっている問題を基に，その一般化規則を探索し適用することによって実現される．

ANALOGICAL REASONING BASED ON HIGHER ORDER GENERALIZATION

Masateru HARAO
Kyushu Institute of Technology, Department of Artificial Intelligence
Kawatsu 680-4 Iizuka , 820 Japan

An approach of realizing an analogical reasoning system for scientific problrms in the framework of higher order logic is presented. Each object knowledge is represented as a first order formula, and a generalized knowledge which characterizes a set of object level rules is represented as a second order formula. The similarity of target problems is defined based on their unifiability, and the generalization procedure is defined as an abstraction to a higher order expression. The analogical reasoning is formalized as a procedure to infer a solution of a given problem by finding some similarity with some observed known problem and applying its generalized rule to the problem.

1．はじめに

機械に，より知的で柔軟な人工知能を実現するためには，従来の演繹的推論の枠を越えた推論機構の実現することが重要な課題となっている．類推は，人間の発見的推論や独創性を支えている最も基本的な思考法の一つであって，類推を推論機能として取り込む事は有用と考えられる．類推には色々な捉え方があるが，次はＭｉｎｓｋｙ[17]の与えた定義である：

Ｉｎ ｏｒｄｅｒ ｔｏ ｓｏｌｖｅ ａ ｎｅｗ ｐｒｏｂｌｅｍ，ｏｎｅ ｓｈｏｕｌｄ ｔｒｙ ｕｓｉｎｇ ｍｅｔｈｏｄｓ ｓｉｍｉｌａｒ ｔｏ ｔｈｏｓｅ ｔｈａｔ ｈａｖｅ ｗｏｒｋｅｄ ｏｎ ｓｉｍｉｌａｒ ｐｒｏｂｌｅｍｓ．

即ち類推とは，与えられた問題の解を，似ていると思われる問題の解法を援用して推論し、求める事と捉えることができる．また，類推は一般的概念や抽象的な性質に関するグローバルな推論であって，人間に見られるような発見的な処理や多相性なども取り込める大きな枠組みであると考えられる．

類推に関する研究は，古くから報告されているが，科学的な考察がなされた先駆的なものとしてＰｏｌｙａ[21]の数学における帰納と類比の研究がある．人工知能的な立場からは，初期にはＥｖａｎｓ[5]の知能テストに用いられるような幾何学における類推の問題を取り扱った仕事，Ｋｌｉｎｇ[16]の類推による定理証明など，がその先駆けである．その後も幾つかの新しい概念の導入され，第二世代の研究が始る．その代表的なものとしては，Ｃａｒｂｏｎｅｌｌ[3]の誘導類推，Ｗｉｎｓｔｏｎ[24]の因果関係や意味ネット表現による類推方式，Ｇｅｎｔｎｅｒ[6]の構造写像法，Ｂｕｒｎｓｔｅｉｎ[2]の類推によるＢＡＳＩＣの理解，などがある．それらの研究を引継いで，現在はＫｅｄａｒ&Ｃａｂｅｌｌｉ[14]のＰｕｒｐｏｓｅ-ｄｉｒｅｃｔｅｄの方法，Ｇｒｅｉｎｅｒ[7]の抽象化法，原口&有川[12]による論理的枠組みでの定式化，等が代表的なものとして挙げられる．これらの詳細は[15]等を参照されたい．

これらの研究の対象領域としては，言語などの比喩理解，問題解決，学習や概念形成，などが主である．また，方法論としては，個別の問題に対してその問題特有の性質を取り扱うといった傾向にあり，形式的に汎用的な定式化を行った研究は少ないようである．

本研究では，計算科学ＡＩ的観点から科学的な問題解決を対象として，形式的定式化のもとに，次の課題に対して類推の問題を考察する．

(1)汎用的で理論的にも裏付けされた類推の枠組みをどう定式化するか．

(2)知識の獲得や学習の機能をどのように類推に取り込むか．

(3)類推の処理機構を定理証明等の手法を用いてどのように機械化するか．

２章で，類推の基本的枠組みを与える．言語としては，型付λ-計算に基づく高階の論理式を用い，知識間の類似性や規則の抽象化を高階単一化の概念を用いて記述する．３章は，観測された事実や例題から一般的知識を導出する，一般化について考察する．特に，具体的入出力例から高階単一化を用いた一般化と，証明過程からの一般化技法について述べる．４章では，ＬＫにおける論理式の証明を例に，この枠組みでの類推による証明の可能性と問題点について考察する．５章は結論である．

2．類推の定式化

類推とは，すでに分かっている問題の解法や知識を用いて，類似性に基づいて解を推論する事とする．本研究では，科学的問題を対象に，出来るだけ形式的な手法で類推を自動化する事を目標にする．自動化手法としては，導出証明や高階単一化を基礎にする．

2．1基本的枠組み

類推においては，問題の類似性をどう捉えるか，一般化知識をどう与えるか，そして類推の結果の正当性の問題，等が重要な要素となる．

(1)類似性：対象問題の領域を特性化する理論Ｔｈの下で式が類似であるとは，理論Ｔｈにおける変換規則の下でそれらが等価なことと考える．変換規則は，対象とする問題空間と意味構造に従って定義される．例えば，可換則が成り立つ世界では，式ｘｙとｙｘとは類似な式となる．式ＰとＱが類似である事を，Ｐ∽Ｑで表す．また，類似な式よりなる一つの集合を概念と呼ぶ．即ち，Ｓ（Ｐ）＝{Ｑ｜Ｐ∽Ｑ}は一つの概念に対応する．

(2)一般化：すでに分かっている問題の解法から，それと類似の問題にも適用可能な一般的な規則を構成する手続きである．一般化は，概念に関する記述であり，高階表現を含む規則への抽象化によって実現する．

(3)正当性：類推は一般的な推論機構であり，観点によって複数の解釈が可能な多相性を持つ．しかし，問題を解く観点を定めたとき，類推処理によって得られた結果はそれに関して正当でなければならない．

対象とする体系の基礎となる理論Ｔｈの知識（公理，推論規則，定理）をバックグラウンド知識と呼びＢで表す．基本的な解法を与える問題を誘導問題とよぶ．問題ｇの証明をｐｒｏｏｆ（ｇ），問題ｇの一般化をφ（ｇ），証明ｐｒｏｏｆ（ｇ）の一般化をｐｒｏｏｆ（φ（ｇ））＝Φ（ｇ）と書く（簡単なためにΦと書くときもある）．基礎となる理論Ｔｈの下での証明を⊢で，一般化を $\vdash\!\!\!\!\sim$ で表す。この時，類推は理論Ｔｈの下で似通った問題の解法も似通っているとする仮定の下での⊢に関する推論である．

【仮定】式の類似性と証明の類似性

ｔ∽ｇ ⇔ｐｒｏｏｆ（ｔ）∽ｐｒｏｏｆ（ｇ）

＞似ている問題は似た解法を持つ＜

【定義2.1】バックグラウンド知識Ｂの下での，誘導問題ｇとその一般化を用いた，対象問題ｔに関する類推を次のように定義する：

(1)Ｂ，ｇ⊢ｐｒｏｏｆ（ｇ）

(2)Ｂ，Ｅ，ｇ $\vdash\!\!\!\!\sim$ φ（ｇ）

Ｂ，ｐｒｏｏｆ（ｇ），φ（ｇ）$\vdash\!\!\!\!\sim$Φ（ｇ）

(3)Ｂ，Φ（ｇ），ｔ∽ｇ⊢ｐｒｏｏｆ（ｔ）

ただし，Ｅはｇの一般化のための知識である．

定義の各手順は概略次のような処理である．

(1)は，バックグラウンド知識Ｂの下での誘導問題ｇの解法を求める操作である．誘導問題としては，基本的で代表的な問題を選んでおく．

(2)は，誘導問題ｇの解法ｐｒｏｏｆ（ｇ）を，ｇの一般化で特性化される類似の問題に対しても適用可能なように一般化する操作である．

(3)は，与えられた問題ｔに対して，それと類似な問題ｇを探し，一般化解法Φ（ｇ）とＢよりｐｒｏｏｆ（ｔ）を求める操作である．ｐｒｏｏｆ（ｔ）を直接求めれば，非決定性が含まれ複雑な操作になるかもしれないが，類推ではΦ（ｇ）を用いるため，決定的な操作となり効率化が図れる．

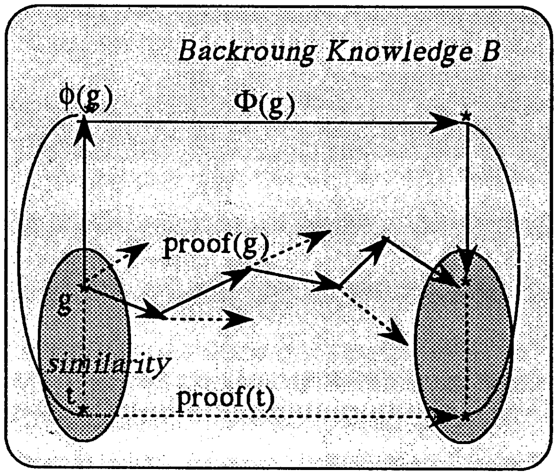


図１類推の基本的枠組み

２．２高階言語に基づく類推の定式化

定式化に用いる言語としては，高階言語を用いる．計算可能性の観点からは一階述語言語の範囲で十分であるが，類推には抽象化や概念などの高階性が本質的であり，高階言語の方が自然で，次のような効果が期待できる．

(1)知識表現の明快さ：型理論に基づく高階論理は，理論的にも明快に集合間の階層性や依存関係を記述可能であり，簡潔で自然な知識表現が可能である．
(2)証明過程の明快さ：高階の概念を用いることによって，意味論的な処理と構文論的な処理を統一的に処理出来，証明が明快になる．
(3)メタ推論機構：高階の概念を用いることによって，メタ推論などの機能を自然な形で定式化可能である．
λ式に型を導入したものを型付λ式という。式Pの型をτ(P)で表す．基本となる型の有限集合をT_0とする．型の集合Tは，$T_0 \subseteq T$と次の規則から帰納的に構成される．

$\alpha, \beta \in T$ならば$\alpha \rightarrow \beta \in T$.

また，型αの**階数**を$Odr(\alpha)$で表し，帰納的に次のように定義する：

(a) $\alpha \in T_0$に対し，$Odr(\alpha)=1$.

(b) $\alpha \rightarrow \beta \in T$ならば

$$Odr(\alpha \rightarrow \beta)=\max\{Odr(\alpha)+1, Odr(\beta)\}$$

型の例としては，整数型；int，論理型；o等があり，τ (equal) =int→inbt→oなら Odr(equal)＝2，τ (⊃) ＝$o \rightarrow o \rightarrow o$ ならOdr(⊃)＝2となる．特に，型oを持つ項を論理式という．以下，Pxy...zをP(x,y,...,z)と書くこともある．x が変数、t および A を項とするとき，A において自由変数xの出現を全て t で置き換えたものを A[x/t]で表す。ただしτ(t)＝τ(x)とする。

【定義 2.2】 λ-変換

(1) α-変換： y が A で x に関して自由なとき(λx.A) を(λy A[x/y])で置き換える。
(2) β-変換： t が A で x に関して自由なとき(λx.A)t を A[x/t])で置き換える。
(3) η*-変換： $\tau(A)=(\alpha \rightarrow \beta)$，$\tau(z)=\alpha$で z は A で自由でないならば，A を λz.(Az)で置き換える．

本稿ではβ，η*-変換でこれ以上変換出来ないλ式へ変換したものをη*-正規形または単に**正規形**とよぶ．型付λ式は，常に正規形を持ちセマンティクスも簡潔である。また，自然演繹法などの証明法とも自然な対応があり，高階論理の定式化に用いられる[17]．本稿の高階言語はこの言語を基本とする．

対象とする知識を**対象知識**(Object-level Knowledge)，その対象知識の集合を特性化する知識を**一般化知識**または**スキーマ**と言う．対象知識は高々１階の変数しか含まないλ式（以下１階の式という）で，一般化式は２階の自由変数を含む２階のλ式（以下２階の式という）でそれぞれ表す[10]．スキーマは型を持ち，対象知識はその型のインスタンスである．

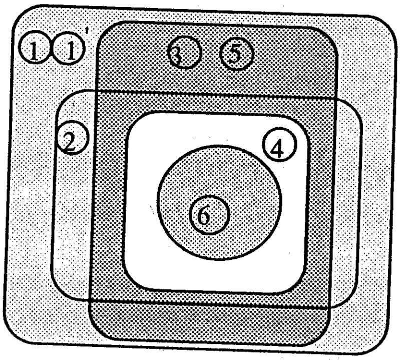
２．３類似性と単一化可能性

次の形をした置き換えの有限集合σを**代入**と呼ぶ：$\sigma = \{x_1/u_1, x_2/u_2, ..., x_n/u_n\}$
ただし，$\tau(x_1)=\tau(u_1), ..., \tau(x_n)=\tau(u_n)$とする．また，項tに代入$\sigma$を施したものを$\sigma(t)$と書き，$[\lambda x_1 x_2 ... x_n.t](u_1, u_2, ..., u_n)$の正規形として定義する。

【定義2.3】 t_1, t_2を$\tau(t_1)=\tau(t_2)$なる項とする。このとき、$\sigma(t_1)=\sigma(t_2)$ ($\sigma(t_1)=t_2$)なる代入σを t_1, t_2の単一化子 **(マッチング)**と呼ぶ。

一般に単一化は，ある演算の下での等価な項を判定する事に相当する．λ項の場合は，λ変換の下での等価性判定でありλ-変換の操作が入るために非常に複雑になる。しかし，極汎単一化子のλ-変換に関する同値類の代表元（正規形）を枚挙する，健全かつ完全な部分アルゴリズムが与えられている．そこでは，模倣規則と射影規則の２種の基本演算があり，それを非決定的に適用することによってアルゴリズムは設計されている[13, 22]．階数がmとnである項sとtの単一化問題を＜m，n＞で表す．

【命題2.1】 単一化問題＜m，n＞の計算可能性，計算の複雑さには図２の性質が成立する．



①:=<m, n>:非可解

①':=射影規則だけなら線形時間で解ける[10].

②:=<m, 1>:可解かどうかは未解決.

③:=<2, 2>:非可解 [8]

⑤:= ③において,変数のネスト出現がないならば可解 [10].

④:=<2, 1>:可解 [13]

⑥:= ④において,2階変数は1種だけで,変数のネスト出現がないならNP-完全 [10].

図2 λ項における単一化可能性

一般に知識は,構文的側面と意味論的な側面を持っており,単一化はその間を結び付ける操作である.そして,類似性を持つ知識とは,ある観点の下で同じ概念に属すものと考えることが出来る.

【定義2.4 (類似性)】式PとQが理論Thの下で類似であるとは,ある概念 (Concept) に対して,置換σ_1,σ_2が存在して,P=σ_1 (Concept),Q=σ_2 (Concept)が成立することである.すなわち,

$$P \backsim Q \Leftrightarrow \exists \text{Concept}, \exists \sigma_1, \exists \sigma_2,$$
$$P = \sigma_1 (\text{Concept})$$
$$Q = \sigma_2 (\text{Concept})$$

理論Thとしては,距離,位相等や各種の代数演算のようなものが考えられるが,ここでは型付λ計算でのα,β,η変換の下での等価性のみを仮定する.

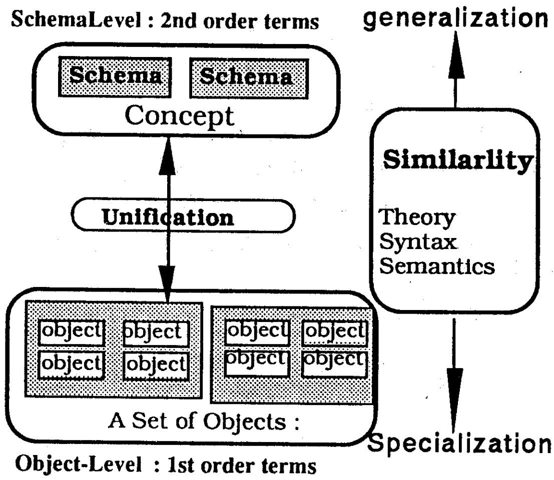


図3 単一化と類似性

3. 一般化

観測された事実や例題から抽象化によって一般的な知識を求める問題を一般化という[19, 20].これは一種の知識獲得や学習の形態でもある.本稿では,高階性を用いた具体例からの一般化と誘導問題からの一般化について述べる。

3. 1具体例集合からの一般化

対象とする関数の具体例や入出力の集合から,それを特性化する一般的な知識を求める一般化技法である.ここでは,高階単一化の手法に基づく方法について述べる.

【定義3.1】Sを項(具体例)の有限集合とする.このとき,項ΦがSの一般化とは,Sの任意の要素s_iに対して,ある置換θ_iが存在してθ_i (Φ) $\backsim s_i$ が成立する事である.ここで,$\backsim$は置き換えの下でのβ,η等価性とする.

Sを閉じた項の有限集合とする.Sの一般化をΦとすると,あるθに対して

θ (Φ) $= f\, t_1 t_2 \ldots t_m \backsim s$, $s \in S$

だから,概略次のように求める事ができる[9].

(1) S= $\{s_1, s_2, \ldots, s_n\}$ とする.Sの型より関数変数Fと変数$y_1 y_2 \ldots y_m$を導入し次の連立方程式をつくる.

$F y_1 y_2 \cdots y_m \infty s_1$
$F y_1 y_2 \cdots y_m \infty s_2$
$\cdots$
$F y_1 y_2 \cdots y_m \infty s_n$

(2)高階単一化アルゴリズムを用いてこの連立方程式の解を求める．解θは，Fおよび$y_1, y_2, \ldots, y_m$に対する置き換えよりなる．

1) $s_1, s_2, \ldots, s_n$に関する解を，$\theta_1, \theta_2, \cdots, \theta_n$とする．

2)その極小上界を$\theta = \bigcap\{\theta_i \mid i=1,\ldots,n\}$とすると，$\theta$が解である．

(3)解θを用いて　$\theta(F y_1 y_2 \cdots y_m) = \Phi$が求まる．

【例題3.1】 $S=\{g a b,\ g b c,\ g b b\}$としたときの一般化を求める．まず，Sの型より次の連立方程式を作る．

$F x_1 y_1 = g a b = s_1$
$F x_2 y_2 = g b c = s_2$
$F x_3 y_3 = g b b = s_3$

この方程式を，高階単一化アルゴリズムを用いて解くと，単一化子が次のように求まる．

$F x_1 y_1 = g a b$に対する置き換え：θ_1

◎	$F=\lambda x y.g\ x\ y$	$x=a,\ y=b$
	$F=\lambda x y.g\ a\ y$	$x\quad y=b$
	$F=\lambda x y.g\ x\ b$	$x=a,\ y$
	$F=\lambda x y.g\ a\ b$	

$F x_2 y_2 = g b c$に対する置き換え：θ_2

◎	$F=\lambda x y.g\ x\ y$	$x=b,\ y=c$
	$F=\lambda x y.g\ a\ y$	$x\quad y=c$
	$F=\lambda x y.g\ x\ b$	$x=b,\ y$
	$F=\lambda x y.g\ b\ c$	

$F x_3 y_3 = g b b$に対する置き換え：θ_3

◎	$F=\lambda x y.g\ x\ y$	$x=b,\ y=b$
	$F=\lambda x.g\ x\ x$	$x=b$
	$F=\lambda x y.g\ b\ y$	$x\quad y=b$
	$F=\lambda x y.g\ x\ b$	$x=b,\ y$
	$F=\lambda x y.g\ b\ b$	

この解のうち，極小単一化子を求める事によって解$\lambda x y.g\ x\ y$を得る．この$g x y$を一般化の本体とよぶ．□

【命題3.2】 s_1，s_2を任意の1階の項とする．このとき，次の連立方程式

$F y_1 y_2 \ldots y_m = s_1 = @_1(\cdots)$
$F y_1 y_2 \ldots y_m = s_2 = @_2(\cdots)$

を満たすs_1，s_2の一般化の本体$HG(s_1, s_2)$は次のように与えられる．

(1) s_1とs_2のヘッドが定数で同じとする：
$HG(s_1, s_2) = @(HG(s_{11}, s_{21}), HG(s_{12}, s_{22}), \cdots, HG(s_{1k}, s_{2k}))$

(2) s_1とs_2のヘッドが定数で異なるとき：

1) s_1とs_2のヘッドが1階の定数ならば，
$HG(s_1, s_2) = x$　（xは1階の変数）．

2) s_1とs_2のヘッドが2階の定数ならば，ヘッドを2階の関数変数Hとし，
$HG(s_1, s_2) = H s_1 s_2$…射影規則．
$HG(s_1, s_2) = H(HG(s_{11}, s_{21}), HG(s_{12}, s_{22}), \cdots, HG(s_{1k}, s_{2k}))$
…模倣規則

(3) s_1とs_2のヘッドのどちらかが1階変数なら$HG(s_1, s_2) = z$（zは新しい変数）．

（証明）（2）の2)：射影規則による解は，
$\theta_1 = \{F := \lambda x y.x,\ y_1 := s_1, y_2 := s_2\}$
$\theta_2 = \{F := \lambda x y.y,\ y_1 := s_1, y_2 := s_2\}$
従って，$\theta = \{y_1 := s_1, y_2 := s_2\}$となり，$\Phi = \theta(F x y) \infty H s_1 s_2$．ここで，Hは2階変数である．他も同様にして示せる．□

【例題3.2】 一般化本体の導出例を示す．但し，$\tau(s_i) = \mu$，$i = 1,2$，と仮定する．

(1) $s_1 = f a$，$s_2 = g b a$

1) $HG(s_1, s_2) = H(f a)(g b a)$…射影規則で，$\tau(H) = \mu \to \mu \to \mu$．

2) $HG(s_1, s_2) = H a$，$\theta_1 = \{F := \lambda x.f x,\ x := a\}$ $\theta_2 = \{F := \lambda x.g b x,\ x := a\}$．$\Phi = H a$，$\tau(H) = \mu \to \mu$．

3) $HG(s_1, s_2) = x$，$\theta_1 = \{F := \lambda x.x,\ x := f a\}$ $\theta_2 = \{F := \lambda x.x,\ x := g b a\}$．$\Phi = \lambda x.x$．□

【命題3.3】 オブジェクトの集合Ｓを一階の項，一般化の項を２階の項に限れば，一般化手続は計算可能である．

（略証）２階の項と１階の項の間の高階単一化（マッチング）は計算可能である．各s_iに対する単一化子も高々有限だから，θは計算可能である．□

同様な手法を用いる事によって，関数の入出力の例を与えた場合にも一般化が可能である．

【例題3.3】 次のような入出力条件が与えられたとき，それを満たすｆを求める．ｆが一般化に対応する．

$f(g\ a\ b)=g\ b\ a$

$f(g\ c\ d)=g\ d\ c$

Ｆを３引数ｇ，ｘ，ｙに関する関数変数と考えて次のような式を考える：

$F(g\ a\ b)=g\ b\ a$

$F(g\ c\ d)=g\ d\ c$

この方程式を満たす単一化子を求めると，次のようになる：

(1)○$F=\lambda f x y.f y x$,
　　$F=\lambda f x y.f y x$

(2)◎$F=\lambda f x y.g y x$
　　$F=\lambda f x y.g y x$

(3)×$F=\lambda f x y.g b x$
　　$F=\lambda f x y.g d x$

(4)×$F=\lambda f x y.g y a$
　　$F=\lambda f x y.g y c$

(5)×$F=\lambda f x y.g b a$
　　$F=\lambda f x y.g d c$

連立方程式の条件を満たすものには(1)と(2)があるが，最小性の条件より(2)を解とする．□

３．２誘導問題からの一般化

解法や例題からの一般化については，数式処理規則の一般化[4]，述語記述の一般化[19]等があるが，本稿では[11]や[23]の手法を基礎にした，代表的な誘導問題の解法からの一般化手法について述べる．特に，バックグラウンド知識としてＬＫ論理系を考え，証明図に関する一般化について考察する．

(1)基本的バックグラウンド知識はＬＫの推論規則であり，用いる幾つかの規則を次に示す．

$$\frac{A,\Gamma\vdash\Theta\quad B,\Gamma\vdash\Theta}{A\vee B,\Gamma\vdash\Theta}(\vee L)\qquad \frac{\Gamma\vdash\Theta,A,B}{\Gamma\vdash\Theta,A\vee B}(\vee R)\qquad \frac{A,B,\Gamma\vdash\Theta}{A\wedge B,\Gamma\vdash\Theta}(\wedge L)$$

$$\frac{\Gamma\vdash A,\ \Pi\vdash B}{\Gamma,\Pi\vdash A\wedge B}(\wedge R)\qquad \frac{\Gamma\vdash\Theta,A\quad B,\Delta\vdash\Lambda}{A\to B,\Gamma,\Delta\vdash\Theta,\Lambda}(\to L)\qquad \frac{A,\Gamma\vdash\Theta,B}{\Gamma\vdash\Theta,A\to B}(\to R)$$

$$\frac{P[x/t],\Gamma\vdash\Theta}{\forall xP,\Gamma\vdash\Theta}(\forall L)\qquad \frac{\Gamma\vdash\Theta,A}{\Gamma\vdash\Theta,\forall x.A}(\forall R)^{*}\qquad \frac{\Gamma\vdash\Theta,P[x/t]}{\Gamma\vdash\Theta,\exists xP}(\exists R)$$

$$\frac{A,\Gamma\vdash\Theta}{\exists x.A,\Gamma\vdash\Theta}(\exists L)^{*}\qquad \frac{\Gamma\vdash\Theta}{A,\Gamma\vdash\Theta}(\text{thin\_L})\qquad \frac{\Gamma\vdash\Theta}{\Gamma\vdash\Theta,A}(\text{thinR})\qquad \frac{}{\Gamma\vdash\Gamma}(\text{Axiom})$$

ただし，＊印の規則には自由変数の条件がつく．

(2)誘導例題として，$g=(p(a)\vee q(b))\wedge\forall x(p(x)\to q(x))\to\exists xq(x)$を考える．この証明図は図５になり，その項表現は次のように与えられる．

```
proof (g) =imp_R(and_L(or_L(all_L(imp_L(axiom(p(a)),
        some_R(axiom(q(a))))),thin_L(some_R(axiom(q(b)))))))
```

q(a)⊢q(a)　(∃R)
p(a)⊢p(a)　q(a)⊢∃xq(x)(→L)　　　q(b)⊢q(b)　(∃R)
p(a),p(a)→q(a)⊢∃xq(x)(∀L)　　　q(b)⊢∃xq(x)(thin L)
p(a),∀x(p(x)→q(x))⊢∃xq(x)　q(b),∀x(p(x)→q(x))⊢∃xq(x)(∨L)
p(a)∨q(b),∀x(p(x)→q(x))⊢∃xq(x)　(∧L)
(p(a)∨q(b))∧∀x(p(x)→q(x))⊢∃xq(x)　(→R)
⊢(p(a)∨q(b))∧∀x(p(x)→q(x))→∃xq(x)

図4　ＬＫ系での証明図

(3) 論理式ｇを述語変数を導入して一般化式φ（ｇ）を次のように定義する．
φ（ｇ）＝（Ｐ（ｖ）∨Ｑ（ｗ））∧∀ｘ（Ｐ（ｘ）→Ｑ（ｘ））→∃ｘＱ（ｘ）
この過程では，色々な一般化が可能である．次いで，一般化証明Φ（ｇ）を求める．Φ（ｇ）の証明図は，誘導問題と異なるＰ（ｖ），Ｑ（ｗ），Ｐ（ｘ），Ｑ（ｘ）の部分を新しい証明で置き換えてやればよい．すなわち，次のようになる．

Φ（ｇ）⇒imp_R(and_L(or_L(all_L(imp_L(proof(P(v)),
　　some_R(proof(Q(v))))),thin_L(some_R(proof(Q(w)))))))

(4)変換Ｍｅｔａ：φ（ｇ）→Φ（ｇ）が，ｇの証明に関する一般化規則となる．また，ＬＫの推論規則は論理式の恒真性を保存するから，得られた一般化規則は恒真な式となっている．一般に，より汎用的で厳密な一般化式を求めることが重要な課題である．

４．類推による知識処理

本節では，これまで述べてきた枠組みで，類推処理がどのように実現されるかを，ＬＫ論理系での類推証明の問題を例に考察する．以下の類推システムでは，誘導問題の集合Ｇに対して有効な一般化規則が与えてあるとする．

４．１高階単一化に基づく類推処理

高階一般化に基づく知識処理の概念図を図５に示す．処理アルゴリズムは次のようである．
(1)対象問題ｔに対して，類似問題ｇを探し，この一般化となっている一般化式φ（ｇ）を探す．これは，ｔ，ｇ共に単一化可能な一般化式を探索する事によって実現する．
(2)φ（ｇ）とｔとの単一化子をθとする．一般化変換Ｍｅｔａ：φ（ｇ）→Φ（ｇ）に対してθ（Φ（ｇ））＝ｐｒｏｏｆ（ｔ）を求める．

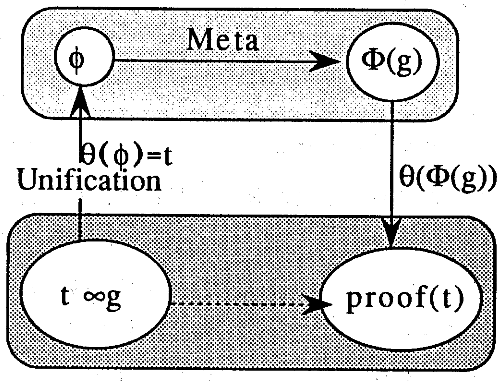


図5．類推アルゴリズム

【命題4.1】 対象知識を１階の項，スキーマを２階の項とする．このとき，類推アルゴリズムは可解である．

ＦＶ（ｆ）で式ｆに現れる自由変数の集合を表す．式ｔに関する類推の結果をｐｒｏｏｆ （t）とするとき，FV（t）⊇FV（proof（t））ならば，tに関する類推は確定的であるという．確定的な類推は，冗長な自由変数を持たないので，ある意味で証明が一意的に定まる．

【命題4.2】 一般化変換Meta：$\phi\to\Phi$がFV（ϕ）⊇FV（Φ）の条件を満たすならば，任意の1階の対象式tに関するMetaの下での類推は確定的である．

（証明）類推は，θ（ϕ）＝tなる単一化子θを計算し，proof（t）＝θ（Φ）を導出することによって行われる．tは1階のλ式であり，θ（ϕ）＝tよりFV（t）＝FV（θ（ϕ））となる．FV（ϕ）⊇FV（Φ）の条件より，FV（t）⊇FV（θ（Φ））．

一般化規則Meta：$\phi\to\Phi$による類推は，すべての一階の式tに対して，ある代入θが存在してθ（ϕ）＝tならばθ（Φ）がtの証明となるとき，正当であるという．LKの証明の一般化によるMeta規則を用いた類推は正当である事が分かる．

4．2類推による論理式の自動証明

LKにおける次のような一般化変換規則Meta：$\phi\to\Phi$を用いた類推の例を用いて処理の概要を説明する．

ϕ＝（P（v）∨Q（w））∧∀x（P（x）→Q（x））→∃xQ（x）

Φ＝imp_R(and_L(or_L(all_L(imp_L(proof(P(v)), some_R(proof(Q(v))))),thin_L(some_R(proof(Q(w))))))

この様な一般化知識を基にして次の証明を求める．

?-t＝((p(a)→s(a))∨q(c))∧∀x((p(x)→s(x))→q(x))→∃xr(x)

(1)ϕと高階ユニフィケーションによってP，Qに関するユニファイアを求める．

θ＝{P:=λx.［p（x）→s（x）］，Q:=λx.q（x）}

(2)Meta：$\phi\to\Phi$から，変換された証明式θ（Φ）を導出する．

imp_R(and_L(or_L(all_L(imp_L(proof(P(v)),
 some_R(proof(Q(v))))),
 thin_L(some_R(proof(Q(w))))))

ただし，

proof（P(v)）＝imp_R(imp_L(axiom(p(a),
 axiom(s(a))))

proof（Q（w））＝q(c)

新しく得られたtの証明図は，ϕの証明図の一部分を置き換えて構成されている．論理式の構文上の類似性と証明の類似性に着目した，自明でない一般化規則を構成することができる．また，そのような一般化規則をバックグラウンド知識に蓄えることによって，類推による自動証明システムが構築できる．

5．考察

本論文では，LK論理系での類推による証明について主に述べたが，この手法は数式処理システム，回路自動合成，プログラム変換といった分野へも同様に応用可能である．しかし，これを実用的なレベルまで発展させるには，次のような課題がある．

まず，具体例や証明過程から一般的な知識情報を導出する，一般化手続きの自動化がある．これは，知識獲得や学習とも関係して重要な課題である．本稿では，高階単一化による一般化など新しい枠組みを提案したが，もっと多面的な角度から検討が必要である．

本稿で提案した類推では，高階単一化が基礎となっている．高階単一化は，解の一意性や最汎性が保証されず，自明な単一化子を多数導出する場合がある．それを避けるためには，単一化子への制約条件等を付加して，無意味な探索の除去の方策が必要である．また，曖昧性を含まない一般化式を定義する事が重要である．

類推は，対象問題の類似性から解を導出する

推論方式であるが，類推過程を機械化するという観点からは，それは探索問題に帰着される．従って，問題間の類似性を用いた探索空間の制限や探索戦略の手法が大きな問題である．

文献[1] P.B.Andrew : An Introduction to Mathematical Logic and Type Theory,Academic Press.Inc,1986.
[2]M.H.Burnstein:Concept Formation by Incremental Analogical Reasoning and Debugging, in Machine Learning II , Morgan Kaufmann, 1986,pp.351-369.
[3]J.G.Carbonell:Derivational Analogy , A theory of reconstructive problem solving and expertise acquisition,in Machine Learning II ,Morgan Kaufman,1986,pp.371-392.
[4]M.R.Donat,L.A.Wallen, Learning and Applying Generalised Solutions Using Higher Order Resolution,LNCS No.310,1988,pp.41-60.
[5]T.G.Evans:A Program for the Solution of Geometric Analogy Intelligence Test Questions,in Minsky,(ed.),Semantic Information Processing,MIT Press,Cambridge,1968.
[6]D.Gentner:Structure Mapping:a theoretical framework for analogy,Cognitive Science, Vol.7,No.2,1983,pp.155-170.
[7]R.Greiner:Learning by Understanding Analogy,Proc.of1stInt.Workshop on AnalogicalReasoning,Morgan Kaufmann,1987.
[8]W.D.Goldfarb: The Undecidability of the Second-Order Unification Problem,TCS,Vol.13,1981,pp.225～230.
[9]M.Hagiya:Synthesis of Rewrite Programs by Higher-Order and Semantic Unification,Proc.of 1st Int.Workshop on Algorithmic Learning Theory,1990,pp.396-410
[10]原尾,岩沼：高階ユニフィケーションアルゴリズムの複雑さについて：コンピュータソフトウェア,Vol.8 No.1 Jan. 1991 ,pp41～53.
[11]M.Harao:Analogical Reasoning Based on Higher Order Unification,Proc.of 1st Int. Workshop on Algorithmic Learning Theory,1990,pp.151-163.
[12]原口，有川：類推の定式化とその実現，人工知能学会誌，Vol.1 No.1, 1986,pp.132-139.
[13]G.Huet and B.Lang: Proving and Applying Program Transformations Expressed with Second-Order Patterns,Acta Informatica,11, 1978,pp31～55.
[14]S.Kedar-Cabelli:Purpose-directed Analogy,Proc.Cognitive Science Society,1985,pp.150-159.
[15]S.Kedar-Cabelli:Analogy-from a unified perspective,Herman(ed.),Analogical reasoning,1988,pp.65-103.
[16]R.E.Kling: A Paradigm for Reasoning by Analogy, Artif.Intell.,Vol.2 No.2,1971,pp.147-178
[17]D.A.Miller,G Nadathur: Higher Order Logic Language,LNCS no.225,1987,pp.449- 462.
[18]M.Minsky:Computer and Thought,Introduction,McGraw-Hill,1963
[19]D.A.Plaisted:Theorem Proving with Abstraction,Artif.Intell.,Vol.16,No.1,1981,pp.47-108.
[20]G.D.Plotlin:A note on Inductive Generalization,Machine Intelligence 5,1970,pp.153-163.
[21]G.Polya:Induction and Analogy in Mathematics,Princeton University Press,1953
[22]W.Snyder,J.Gallier:Higher Order Unification Revised, J.of Symbolic Computation, No.8,1989,pp.101-140.
[23]T.B.de la Tour,R.Caferra:Proof Analogy in Interactive Theorem proving,IJCAI'87, 1987,pp.95-99.
[24]P.H.Winston: Learning New Principles from Precedents and Exercises,Artf. Intell., Vol.19,No.3,1982.