

概念形成機構をもつ
フレーム型知識ベースシステムの開発

長町三生・丸中謙司・伊藤宏司・福場良之・辻敏夫
広島大学工学部

エキスパートシステム構築の際の問題の一つに知識獲得の問題があげられる。知識を自動的に獲得することを実現する方法として、フレーム型知識表現における上位-下位概念の関係に着目し、下位概念から上位概念を帰納することについて考察する。この過程は例題学習といわれる。現在、例題学習における最大の問題の一つは規則空間探索における組み合わせ的爆発である。従来、規則空間探索の制御は領域に依存しない構文的ヒューリスティクスと領域固有の知識の手続き的組み込みによって行われてきた。我々は汎用な知識ベース作成システムを実現するために領域固有知識の宣言的表現を採り入れ、知識を特定する属性または関係自体の一般的性質や意味的关系を表現利用する手法を提案する。

Frame-type Knowledge Base System
with
Inductive Learning Mechanisms

Mitsuo NAGAMACHI, Kenji MARUNAKA, Koji ITO, Yoshiyuki FUKUBA, Toshio TSUJI
Faculty of Engineering, Hiroshima University
Shitami, Saijo-cho, Higashihiroshima-shi, Hiroshima, 724, Japan

A method of inducing general descriptions of concepts from specific instances of these concepts using frame-type knowledge representation is described. This process is a concept-learning from examples (called concept aquisition). One of the most difficult problems on the inductive learning is combinational explosions in searching rule space. To solve the problem, syntactic heuristics, which are domain independent, and procedural knowledge, which are domain dependent, are used. We represent declaratively specific properties of concepts and relationships among concepts, and suggest that these declarative knowledge is used to generate descriptions for a knowledge base constructing system.

1. まえがき

エキスパート・システムの設計問題は、知識獲得、知識表現、知識利用の3つのテーマに分類される。エキスパート・システムの知識ベースにおける知識は、プロダクショナルルールのような組織化されたモジュールとして与えられることが多い。これは、知識の追加、削除、修正といった編集が容易であり、また、検索・推論に利用しやすいためである。従って、これらの知識は、問題領域に長年かかっている専門家から（通常知識工学者を介して）整えられた形式で与えられる。

エキスパート・システムの対象領域の多様化やシステム規模の増大に伴い、このような知識獲得の方法に次のような問題が生じてきた。

1) 獲得されるべき知識量の増大

システムが実際に問題解決にあたるには、些細な常識でも、もろさず与える必要がある。また、専門家の持つ知識は本来、明示的に体系化されたものではなく、それを利用できる形に処理することは大変な作業になってくる。このことから、人間の処理では、能力的にも経済的にも困難が生じる。

2) 専門家の存在しない分野でのシステム作成

専門家が存在しない領域でのデータ、或はまだ専門家が検討中のデータを、システムが整理、解釈して、知識ベース化できることが期待される。応用分野の拡大とともにこのような必要性はますます高まると思われる。このようなデータ解釈機構は逆に専門家を助けるツールともなる。

3) 知識の統合化問題

1)に関係するが、知識を追加することは、単に知識のモジュールを知識ベースに格納するだけでは済まない。加えられた情報と既存の情報との矛盾や冗長性を排除しなくてはならない。知識ベースの規模増大に伴って、すこしの修正でも、このチェック作業は膨大となってくる。

4) システム稼働中の知識獲得

システムは、例えばユーザーに関する知識を獲得する必要があるかもしれない。システムが問題解決を遂行中に知識を獲得することは十分考えられる。このことは、単にインターフェースの充実で済む問題でなく、知識ベース内の統合処理にかかわってくる。そして、その処理は、システム管理者に依存しない形で行うことができなければならない。

これらの問題に対する一つの解として、知識のシステムによる自動獲得が考えられる。すなわち、知識ベースに与えられた秩序に乏しい情報をそのままの形態で格納するのではなく、潜在する規則性を観察して、より圧縮された表現に書き替える、一種の帰納学習機構をシステムに与えるのである。

ところで、このような学習過程の一つに概念学習または例題学習と呼ばれるものがあり、古くから心理学において扱われている¹⁾。例題学習とは、ある概念の実例群（及び反例群）の記述からその概念の一般的記述を一般化して求める問題である。例題学習は70年代に入るとその定式化の比較的容易なことから人工知能（以下AI）の分野においても研究されはじめた。

今日までに開発された学習システムは問題領域の知識がプログラム内に手続き的に強く組み込まれている。従って各々の学習機構は各領域でしか働けない場合が大半を占めている。このような現状に対し、ソフトウェア資源の有効利用をはかるといふ実用的観点からして、また一方で、学習過程の解明への理論的な寄与という点からも、領域に依存しない学習システムについて考察することが要求されてくる。

本稿では、知識ベース・システムに領域と独立な学習機構を与えることを検討する。特にここで、フレーム型表現における上位-下位概念の關係に着目する。従来、フレーム表現における上位概念はシステム管理者により先験的に与えられているが、これをシステム自ら生成し知識を木構造に体系化してゆくことを考える。この過程は下位概念からの上位概念の帰納推論である。

2. 例題学習

例題学習とは、特定の事例群からそれらの振舞や特徴に潜む一般的規則を導く帰納推論過程をいう。SimonとLeaは、例題学習過程を2空間モデルで表現した⁶⁾。すなわち、例題学習とは学習者の訓練のための事例の空間と可能な一般的規則の存在する規則空間における協調的同時探索であるといえる。ここで、学習者は、

1) その時点での仮説群に誘導されながら事例空間を観察（探索）する。

2) その観察に基づいて仮説が生成・消去される。

3) すると今度は逆にそれらの仮説の変更を検証するため再び事例空間に新たな事例を求めに行く。

学習はこれらの手続きの反復で進められてゆく。このとき、2空間の表現に大きな隔りがあれば、両空間の探索は複雑な仮説の生成や検証手続きを行わねばならない。この複雑さを回避するために多く用いられる方法が単一表現のトリック²⁾である。単一表現トリックでは、事例と規則が構文上区別のない概念記述となる。このような工夫により、学習過程はある概念記述に一般化操作を加えて変形されたものが事例群すべてを包含するように導くことで進められる。概念記述の一般化は、定数の変数化のほか条件の削除、例外の導入、内在論理和(internal disjunction)等の操作により行われる⁹⁾。

一方、例題学習をいくらかシンプルに規定したものに単一概念の学習がある。この場合、学習されるものは一つの概念記述であり、事例はその概念に属するかどうか判っている。この問題は次のように表現される。

1) 概念及び実例を表現する言語 (これらは実例空間と規則空間を規定する) 及び

2) 求める概念に属する実例と反例 (負の実例: 与えられない場合もある) を与えられて、

すべての正の実例を含み、すべての反例を排除するような概念記述を求める。

なお、今日までのほとんどの研究では、求める概念記述の解は一意に決定されると仮定している²⁾。

さて、単一表現トリックを用いた単一概念学習過程において、尤もらしい (可能な) 仮説の集合 H は、H 中で最も一般的な記述の集合 G と最も特殊な集合 S との間の空間として表現できる。この仮説集合 H のことを Version-空間と呼ぶ³⁾。単一表現トリックを用いた単一概念の学習の問題は、H を一つの記述 C に収束させることに帰着される。この過程は候補消去アルゴリズムで記述できる。

しかし、このアルゴリズムは規則空間の全探索である。その上、ある程度豊かな表現力をもつ記述言語と一般化手続き群によって生成される規則空間は非常に大きくなる。したがって、このアルゴリズムをそのまま適用することは事実上不可能である。この問題に対して従来の学習システムがとってきた方策には次のようなものがあげられる。

- 1) 構文的なヒューリスティック規則に従って探索を制御する¹⁰⁾。
- 2) 領域固有のモデルを用いて探索を規定する¹¹⁾。
- 3) 表現言語に制約を加えることで探索量を軽減する。例えば一階述語論理から限量子や多項述語を除くことが行なわれる¹²⁾。

本稿では、従来手続き的にシステムに組み込まれてきた領域固有知識を宣言的に表現し、システムの核部分から分離することを試みる。また、3)に関連して、実例の特徴情報と構造情報を分離する2空間アプローチを用いる。そして従来、2空間それぞれにおいて別々の表現言語が用いられていたのに対し、フレームというひとつの表現の上で統一的に扱うこととする。

3. システム概要

フレーム及びそれに派生したスクリプト等の知識表現形式は、今日人工知能や知識工学において最もよく用いられているものである。このフレーム表現における上位・下位概念関係に着目し、従来システム管理者により先験的に与えられてきた上位概念を、前章で概説した概念学習の手法により生成することを考える。本システムは、単純なフレーム形式で表現した知識群に対して、ユーザーの管理のもとで種々の帰納推論手続きを適用してゆく学習システムである。

3-1. 知識表現

本システムにおけるフレーム表現はスロットに入れ子構造を許さない、単純な属性リストとする。すなわち、スロット名はストリング (アトム)、スロット値は任意の S 式である。各フレームは、最低限 "type" スロットを持ち、Table 1 のような、フレームのタイプを設定している。

知識は属性リスト、及びそれらによって構成されるネット・ワークにより表現される。このネット・ワーク情報はノードを中心にしたリスト形式で属性値に与えられる (Fig.1 参照)。システムが表現対象に対して先験的に持っている知識は、各 type の扱い方に関するものだけであり、領域固有の知識はシステム内部に手続き的には全く含まれていない。本システムの特徴のひとつは、フレームにおける属性自体もまたフレームとして宣言的に表現する点にある。relation および property 間の関係構造を把握することは、従来単なる無意味綴りにすぎなかった属性名に意味を与えることになる。このような知識は概念探索の制御や教師なしの学習において大きな役割を果たすと考えられる。

Table 1. フレームのタイプ

FRAME TYPE		ex.
object	: 個々の事物	blockA, person3
property	: 単項述語で表現されるもの	square (A), big (A)
relation	: 多項述語 "	contact (A, B)
construction	: FRAME で構成されるネット・ワーク	arch1, statusX
causal-pair	: 2つのFRAMEからなる construction	movement-1 (status1, status2)

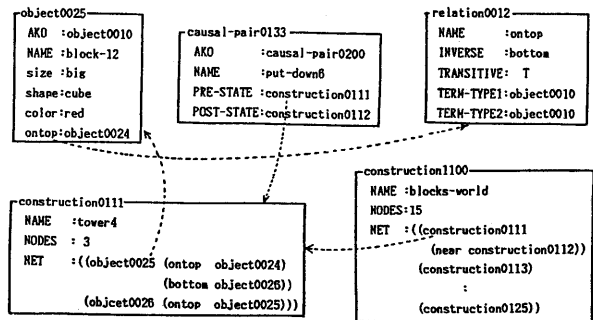


Fig.1 フレームの例

3-2. 一般化手続き

概念記述を述語表記した場合、述語群は単項述語と多項述語とに大きく分類することができる。述語の引数がある実体と考えると、単項述語は実体の特徴を表し、多項述語は実体間の関係を表すと意味付けすることができる。この2つの表現を組み合わせて記述されている概念を、それぞれ別個に一般化することを考える。この方法は規則空間を二つに分けることから、2空間アプローチ⁵⁾と呼ばれる (SimonとLeaの学習に対する2空間探索アプローチ⁶⁾と区別するために、以下、2空間分割アプローチと呼ぶ)。

フレーム記述の一般化に際して、本システムでは2空間分割アプローチをとる。この方法が構造を持つ対象に非常に効果的であり、本システムの扱うデータがまさにその類に属するからである。まず個々のフレーム記述に関する一般化を考え (特徴の学習)、次にある関係群により結合されたフレーム群の構造を一般化することについて述べる (構造の学習)。

a. 特徴の学習

個々のフレームは、特徴と特徴値の対 (以下 "属性対" と呼ぶ) から成っている属性リストの形をしている。この属性リストをひとつの実例とみて、あるフレーム群を一般化することを考える。ここでのフレーム群はすでに与えられているものとする。すなわち、一般化の対象は、ある特定の上位概念を有する (共通のAKO属性値をもつ) フレーム群、あるいは、ユーザー (教師) により指定されたフレーム群であるとする。一般化の操作としては、基本的なものである条件削除と定数の変数化を実行する。この2つの操作は、属性対が論理積で結合されていることから、共通の属性対及び共通の属性 (属性値は異なってもよい) を抽出するというように帰着できる。これは、実例群における属性対を通算した "分布リスト" を用いることで容易に実現できる。この "分布リスト" は Fig. 2 のように与えられる。

ここで実例群の要素数を C とすると、共通の属性対と属性の集合は以下のように与えられる。

$$\{(P_i, V_{ij}) \mid N_{ij}=C\} : \text{属性対}$$

$$\{P_i \mid \sum_{j=1}^{\nu_j} N_{ij}=C\} : \text{属性}$$

これらの集合を求めることは非常に容易である。また、ノイズに寛容にするには等号を緩めてやればよい。本システムでは、"Cの何パーセント以上" という設定が行える。この特徴の学習が対象としているのは、主に object フレームである。しかし他のタイプのフレームを対象とすることも考えられる。例えば construction フレームは、net: net-work list という属性対がその内容の中心である。(これは次に述べる構造の学習の対象となる) が、construction フレームはこの他に node 数、枝数、階層か?、treeか?、等ネットワークの一般的性質を示す属性をもっているかもしれない。これらは特徴の学習の対象となる。

[(P1 (V11 N11)
(V12 N12))
(P2 (V21 N21)
(V22 N22)
(V23 N23))
:
(Pm (Vm1 Nm1)
:
(Vmνm Nmνm))]
Pi : 属性
Vij : 属性 Pi の第 j 値
(但し j は出現順であり意味はない。)
Nij : 属性 Pi の値が Vij である実例の数
νj : 属性 Pi のとる値の数
m : 実例群中に見出される属性の種類の数

Fig. 2 分布リスト

b. 構造の学習

次に、個々のフレームから、2項関係で結合されたフレーム群へ視点を変えてみよう。ここでは、個々のフレームの特徴は考えず、フレーム群のなすネットワーク構造にのみ着目する。このような構造情報は construction フレームの "net" 属性値として与えられる。まずその表現形式を与える。

[ネットワーク・リスト]

ネットワークはノードを中心にみてリストで表現する。これと双対に、関係を中心にみた表現とあわせて Fig. 3 に例を示す。

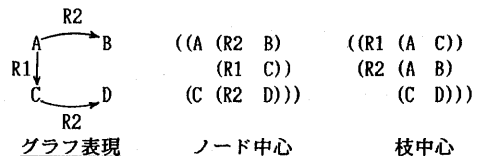


Fig. 3 ネットワークのリスト表現

[2つの有向グラフ G_x, G_y のマッチング]

問題は、このネットワーク群を一般化し、そこに共通して存在する構造を抽出することである。この一般化は以下に示す2つのネット間のマッチングを基本とする。

<2つのネットのマッチング手続き P1>

step 1 初期設定

$N_x = \{G_x \text{中の全ノード}\}, N_y = \{G_y \text{中の全ノード}\}, G_g = \phi$

step 2 終了判定

$N_x = \phi$ ならば終了。 G_g がマッチングの結果である。

step 3 マッチ・ノードの選択

N_x から1つノードを選び n_x とする。

N_x から n_x を除く。

step 4 被マッチ・ノードの選択

n_x とマッチさせるノード n_y を N_y から1つ選ぶ。

N_y から n_y を除き更新する。

n_x と n_y に共通のラベル n_{g_i} をつける

(このとき、 n_y は n_x に"束縛された"という。)

step 5 共通枝の収集

n_x と n_y が共通の枝 r_j ($j=1, \dots, m$; m は n_x, n_y に共通な枝の個数)をもってるとき、

各 r_j について以下の手続きを実行する。

- a) n_y から r_j でむすばれたノード (" n_y の r_j 先"と呼ぶ)が未束縛ならば、 n_y の r_j 先を n_x の r_j 先に束縛させて、 G_g に付ける。
- b) n_y の r_j 先が r_k によって既に束縛されているならば、

それが n_x の r_j 先と等しい時にのみ、 n_{g_i} に r_j と r_j 先のノードラベルを付け加えて、 G_g に付ける。

$j=0$ (共通の枝を持たない)ならば

n_{g_i} のみを G_g に付ける。

step 2~5を終了まで繰り返す。

このstep 3において、 n_y の選び方の順序をかえると結果として生成されるネットも変わってくる。全部のマッチングの組合せはノード数の増加とともに爆発的に増えてくる。本システムでは G_x 中のノード n_x について、 n_x がもつと同じ枝をもつノードが G_y 中に存在すれば、そのノード群 $\{n_y\}$ にのみ n_x をマッチさせるという制約を与えている。その結果生成されるネットがなければ全探索を行う。また生成されるネット群は枝をもつノードの数とすべての枝の数の和によりソートされ、指定した数だけ提出される。

この手続きにより単純な2つのネットをマッチングさせた例をFig.4に示す。

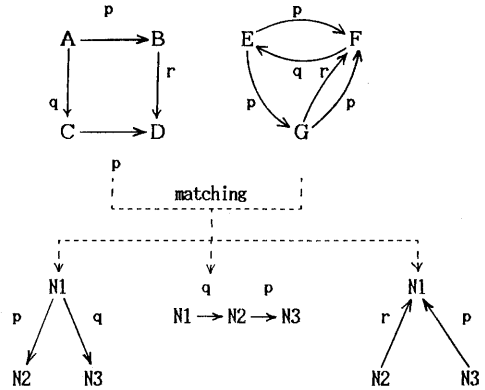


Fig.4 2つのネットワークのマッチング

[複数のグラフのマッチング]

2つのグラフのマッチングを繰り返すことにより、3つ以上のネットを般化できる。その手順を次に示す。手続きP2がすべて終了したときOが最初の入力Iを般化したネット集合である。

<ネット群I = $[N_1, N_2, \dots, N_n]$ を入力とし、般化されたネット群Oを出力とする再帰的手続きP2>

step 1 N_1 と N_2 をマッチングした結果のネット集合を $S_g = [g_1, g_2, \dots, g_m]$ とする。

step 2 $S_g = \phi$ ならば終了。

Iが2要素なら S_g をOに入れ終了。

step 3 Iから N_1, N_2 を除いた列に、 S_g の各要素を先頭に与えた列の群 I_i をつくる。

$$I_1 = [g_1, N_3, N_4, \dots, N_n]$$

$$I_2 = [g_2, N_3, N_4, \dots, N_n]$$

⋮

$$I_m = [g_m, N_3, N_4, \dots, N_n]$$

step 4 各 I_i を入力として手続きP2を実行する。

[階層ネットの般化]

ネット・ワークを構成するノードは、Fig.5に示すようにそれ自体がネット・ワークであることが考えられる。このような階層ネットの般化は、前に述べた複数ネットの般化を繰り返すことで実行できる。その手順を下に示す。

<階層ネット群の一般化手続きP3>

step 1 ネット群H = $[NET_1, NET_2, NET_3, \dots, NET_n]$ をトップレベルにおいて般化して、その結果得られた集合を

$$G = [g_1, g_2, g_3, \dots, g_m]$$

step 2 以下の手続きをGの各要素gについて行う。

gを構成するノードの集合を $NOD_g = \{n_1, n_2, \dots, n_k\}$ とする。

NOD_g が非ネットの要素を含んでいれば終了。

Hの各要素 NET_j について $n_j (j=1, k)$ に対応するノードを e_{ij} とし、各jについてまとめた集合 h_j をつくる。

$$h_1 = \{e_{11}, e_{21}, \dots, e_{n1}\}$$

$$h_k = \{e_{1k}, e_{2k}, \dots, e_{nk}\}$$

これら h_j をそれぞれをHとしてP3を適用する。

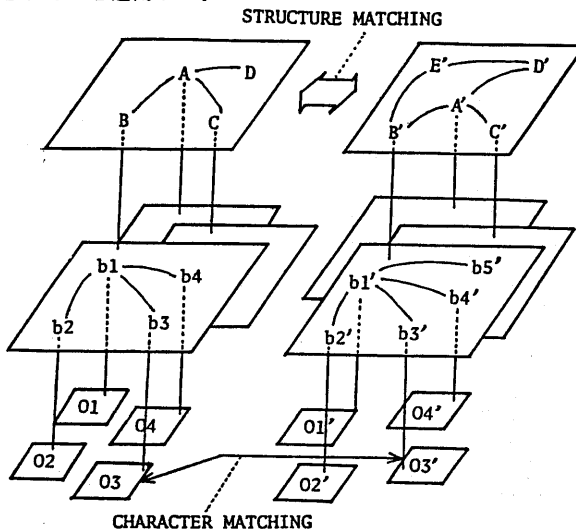


Fig.5 階層ネットワークの一般化

3-3. 教師なしの学習

これまでの議論は、与えられた実例群を一般化する方法に関するものであった。そのより発展した課題として、実例群の集合が定義されていない状況において概念を生成することが考えられる。このような課題の発生する場面のひとつに論理和表記の導入問題がある。概念の表現言語に無制限に論理和の使用を許すと、生成される概念が無意味なほど複雑になる可能性がある。ひとつの極端な例は実例群全体を論理和で結んだ記述である。この問題に対するひとつの解は、ある程度の実例群をカバーする論理積記述をいくつか生成し、それら全体の論理和表現で実例群全体を覆うようにするという方法である⁴⁾。このとき、個々の概念記述を生成する過程は教師なしの学習となる。ここでは属性および関係に着目した方法を提案する。

a. 特徴の学習

記述集合（実例でないのでこう呼ぶ）Cの部分集合を規定する概念記述の生成を考える。この課題はCの分類であるときとみなせる。分類ができればその結果としての各部分集合の要素を一般化することで概念記述が得られるからである。さらに分類を、特定の属性あるいは属性対を基準にして実行していくと、その基準の蓄積結果がすなわち概念記述となっていくことがわかる。本システムはこの分類過程を概念生成手続きとしてもっている。

<記述群Cの部分集合を実例としてもつ概念の生成手続きP4>

step 1 C中の記述群における適当な属性pを選択しそれによりCの部分集合 $\{C_p\}$ を求める

step 2 各 C_p に特徴の学習を実行し、その結果が分類記述pの蓄積より大きければ終了する。

step 3 Cが設定した大きさまで小さくなったら終了する。

step 4 各 C_p をCとしてP4を適用する。

ここで、step 1において適当なpを選択してゆくことには属性間に関する知識が有効である。例えば、物理性質という上位フレームに属する属性群に着目し、まず"温度"次に"湿度"の値によって分類してゆくといったことが考えられる。

b. 構造の学習

一般にフレーム間の関係構造は、そのごく一部がネットワーク・リストとして明示的に表現されているにすぎない。本システムは、特定のrelation群を指定すると、それらのうちのどれかを枝としてもつノード（フレーム）を収集し、そのノード群の構成するネットワークを作成する機構をもっている。

<フレーム群Fと関係群Rを入力に持つネット生成手続きP5>

step 0 $F = \{f_1, f_2, f_3, \dots, f_n\}$, $R = \{r_1, r_2, \dots, r_m\}$,
 $O = \emptyset$ とする。 f, r はそれぞれフレーム（ノード）及び関係（枝）を表す。 O は出力集合。

step 1 $F = \emptyset$ ならば終了。 O が生成されたネット群である。

step 2 f_1 を出発点としてRの要素で結合されたノードを集めてゆく。そのノード集合を F_r とする。

step 3 F_r をノードとするネットを作成し O へ入れる。
 ここの枝はRに限らず、 F_r の要素間に存在するすべての枝である。

step 4 $F = F - F_r$ としてstep 1へ

この手続きで作成されたネット群を般化すると、特定の实例群をもつ構造概念が生成されたことになる。Fig. 6の例では関係(p, r)を指定している。このような関係群の指定には、前述の特徴の学習と同様に関係間の関係に関する知識が有効である。

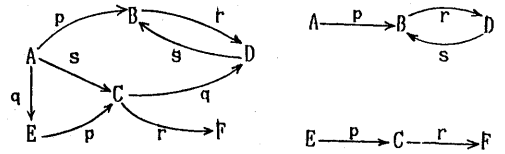


Fig.6 ネットワークの生成

3-4. 関係概念に関する問題

関係（属性）に関して何を宣言的に表現し、さらにそれらをどのように学習するかということは本システムにおける重要なテーマである。その端緒をつかむべくrelationフレームの以下の点について検討し特有の属性対を与えた。

1) 項の性質：関係R (term1, term2) の各項がある共通した性質をもっていることが予想される。例えば母子(x, y)においてx, yは人間でxの性別は女性である。つまり、

$$(\forall x)(\forall y)[\text{母子}(x, y) \supset \text{人間}(x) \wedge \text{人間}(y)]$$

$$(\forall x)[\text{母子}(x, y) \supset \text{女性}(x)]$$

$$(\forall x)(\forall y)[\text{母子}(x, y) \supset \text{年上}(x, y) \wedge \text{年下}(y, x)]$$

が成立するであろう。これらの情報はx, yを独自のフレームとして作成し、Rフレームがそれらを引用することで表現できる。

2) 関係間の関係：ある2つの関係間に、上位、下位、逆（被）、因果等の関係が存在する可能性がある。
 3) 一般的性質：推移的、反射的等の一般的性質が問われる。

これらの情報をFig.7に示すように特別な属性として与えた。

<pre> relation0123 name: mother AKO : relation0111(kinship) TERM1: object0002 TERM2: object0003 CO-REALATIVE: relation0222(strong) TRANSITIVE: NIL </pre>

Fig.7 Relationフレームの一例

3-5. インプリメンテーション

本システムはHITAC M200H上でUtilispにより開発中である。Fig.8に本システムの構成を示す。システムはフレーム・データ・ベースとそれに対する手続き群から成っている。bagとはユーザーの定義するフレーム集合のデータであり、ファイル管理や一般化操作を含め多くの手続きがこのbagを対象とする。USEとはUtilispの構造エディタである。システムのもつ簡易フレーム・エディタを補佐している。

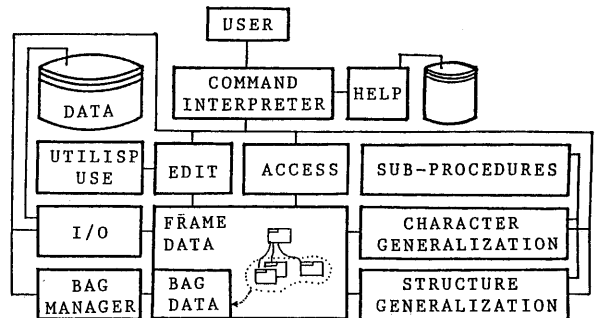


Fig.8 システム構成

4. まとめと今後の展望

本稿では、エキスパート・システム構築の際の知識獲得問題に着目し、システムによる知識獲得アルゴリズムとして概念学習のパラダイムを持ち込んだ一方法を提案した。本稿で与えた学習手続き群はプロトタイプであり、なお一層洗練される必要がある。

概念学習テーマにおける最も大きな問題は概念空間における探索制御である。本研究のアプローチにおいては特に構造記述のマッチングが問題となる。階層構造や多数のノードを持つネットを対象とした大規模なマッチングを可能にするためには順序に従って逐次的にノードを対応づけてゆくのではなく、人間がよく行うように、注視点をあちこちにしながら実例群を眺めていく過程を実現する必要がある。ノードのマッチにおいてノードの特徴情報を利用して探索をガイドすること、すなわち特徴の学習と構造の学習の協調的並列実行が効果的であろう。関係の性質に関する帰納推論過程などは概念学習の枠外の問題である。

人間の行う種々の帰納過程を定式化し体系化するためには、論理学や集合論から出発する公理的アプローチに加えて、人間の実際の思考過程を思考実験などにより観察・収集・分類・体系化してゆく地道な解析的アプローチが必要不可欠であると考えられる。

参考文献

- 1) H.A. Simon: "Why Should Machine Learn?", in Machine learning, R.S. Michalski, J.G. Carbonell, T.M. Mitchell (eds.), Springer-Verlag (1984).
- 2) P.R. Cohen, E.A. Feigenbaum 編 (田中, 淵 監訳): "人工知能ハンドブック第Ⅲ巻", 第XIV章, 学習と帰納的推論, p.417-664, 共立出版 (1984).
- 3) T.M. Mitchell: "Version spaces", IJCAI, 5, pp.305-310 (1977).
- 4) R.S. Michalski: "On the quasi-minimal solution methods to the general covering problem", Proceedings of the 5th International Federation on Automatic Control, 27, pp.109-129 (1969).
- 5) T.G. Dietterich and R.S. Michalski: "Inductive learning of structural descriptions", Artificial Intelligence, 16, pp.257-294 (1981).
- 6) H.A. Simon, and G. Lea: "Problem solving and rule induction: A unified view", in Knowledge and cognition, L. Gregg (ed.), pp.105-127, Lawrence Erlbaum, N.J. Hillsdale (1974).
- 7) B.G. Buchanan, and T.M. Mitchell, "Model-Directed Learning of production rules", in Pattern-directed inference systems, D.A. Waterman and F. Hayes-Roth (eds.), pp.297-312, Academic Press (1978).
- 8) D.B. Lenat, "AM: An artificial intelligence approach to discovery in mathematics as heuristic search", Rep. No. STAN-CS-76-570. Computer Science Dept., Stanford University. (1976) (Doctoral dissertation. Reprinted in R. Davis and D.B. Lenat. "Knowledge-based systems in artificial intelligence", McGraw-Hill (1980).)
- 9) R.S. Michalski: "A Theory and Methodology of Inductive Learning", in Machine learning, R.S. Michalski, J.G. Carbonell, T.M. Mitchell (eds.), Springer-Verlag (1984).
- 10) F. Hayes-Roth and J. McDermott: "An interference matching technique for inducing abstractions", Communications of the ACM, 26, 401-410 (1978).
- 11) P.H. Winston: "Learning structural descriptions from examples", Rep.No.TR-231, AI Laboratory, MIT (1970).
- 12) S.A. Vere: "Inductive learning of relational productions", in D.A. Waterman and F. Hayes-Roth (eds.), Pattern-directed inference systems, pp.281-296, Academic Press (1978).