知識工学と人工知能 64 － 3
（１９８９．５．１８）

ＧＬＰの理論に基づく学習システムの構築Ｉ

赤間　清

北海道大学工学部情報工学科

ＧＬＰ(generalized logic program)の理論は，文脈自由文法や論理プログラムや制約論理プログラムなどを含む多くの知識表現系を統一的に扱うことができる．文脈自由文法も Prolog プログラムも制約論理プログラムもすべてＧＬＰの例である．ＧＬＰの理論に基づいた学習システムの構築理論では，任意のＧＬＰを知識表現系とする学習システムが議論の対象となる．そして，それらの学習システムが共通に持つ基本的な構造が抽象的なレベルで論じられる．本論文では，ＧＬＰの理論に基づいた学習システムの理論の最も基礎的な部分を与える．ＧＬＰの理論が学習システムの理論の構築にとって非常に重要な役割を果す可能性を，ＧＬＰと学習システムの基本的な構造などに基づいて論じる．

Constructing Learning Systems
based on
the theory of generalized logic programs
PART I

Kiyoshi Akama
Dept. of Information Engineering, Faculty of Engineering, Hokkaido University

GLP (generalized logic program) theory gives a unified treatment of many knowledge representation systems, which include CFGs (context free grammers), LPs (logic programs) and CLPs (constraint logic programs). All of them can be regarded as GLPs. The theory of learning systems based on GLP theory deals with any learning system whose knowledge representation system is a GLP, and discusses their common features in the abstract framework of GLP theory. In this paper we give the most basic part of the learning system theory based on GLP theory. Several very important advantages of GLP based learning systems are discussed.

1．　まえがき

ＧＬＰ(generalized logic program)の理論は，文脈自由文法や論理プログラムや制約論理プログラム[Jaffar 86]などを含む多くの知識表現系を統一的に扱うことができる．文脈自由文法も Prolog プログラムも制約論理プログラムもすべてＧＬＰの例である．ＧＬＰの理論に基づいた学習システムの構築理論では，任意のＧＬＰを知識表現系とする学習システムが議論の対象となる．そして，それらの学習システムが共通に持つ基本的な構造が抽象的なレベルで統一的に論じられる．また必要なら知識表現系により強い構造を仮定して，知識表現系のクラスを限定した理論を構築すればよい．そのようにしてできる理論の全体は，多くの研究の相互関係と本質を明らかにする生産的で明快な理解の体系をなすだろう．

ＧＬＰの理論も，それに基づいた学習システムの理論も，既存の多くの研究成果を統合できる枠組みである．しかしもちろんそれらは既存の成果の単純な一般化をする「理論のための理論」ではない．それは，既存の成果を統合しながら，さらに深い理解と大きな可能性に我々が進むために是非とも必要な枠組みである．

ＧＬＰの枠組みのもとで我々がこれから達成しなければならない課題は山のようにある．それらは我々を真に効果的に前進させてくれる課題であると考えられる．本論文では，ＧＬＰの理論が学習システムの理論の構築にとって非常に重要な役割を果す可能性を，ＧＬＰと学習システムの基本的な構造などに基づいて論じる．それはＧＬＰの理論に基づく学習理論の可能性への簡単な introduction である．

2．　学習システムと知識表現系

2．1　学習システムの概形

ここでは便宜上，学習システムを３つの構成要素（Ｍ＋Ｉ＋Ｌ）に分けて考えることにする．

Ｍ：Memory element　　…　記憶部分

Ｉ：Interaction element …　相互作用部分

Ｌ：Learning element　　…　学習部分

Ｍは知識を蓄える部分，Ｉは外界からの質問が与えられたとき，Ｍの知識を用いて応答を作りだすアルゴリズム，ＬはＭの知識を更新するアルゴリズムである．

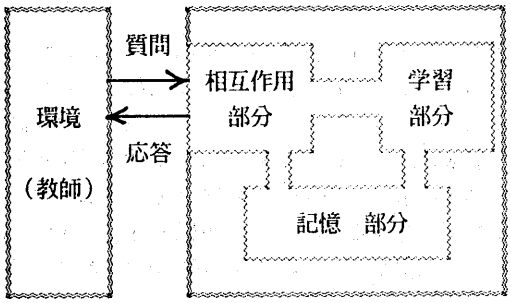

図１　学習システムの構成要素

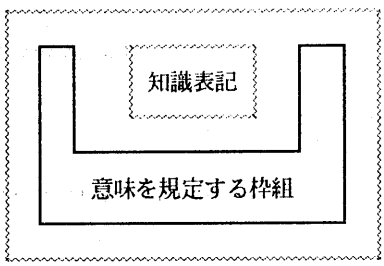

図２　知識表現系

2．2　知識表現系と学習システム

記憶部分は，知識表現系（＝知識表記＋意味を規定する枠組み）を基礎として構成されなければならない．知識表現系（図２）から学習システムをながめると図３のようになる．学習アルゴリズムは，意味を規定する枠組みに支えられた知識表記を変更するアルゴリズムである．重要なことは，知識表現系が学習システムの核であり，知識表現系の持つ性質が学習システムの能力に大きな影響を与えることである．

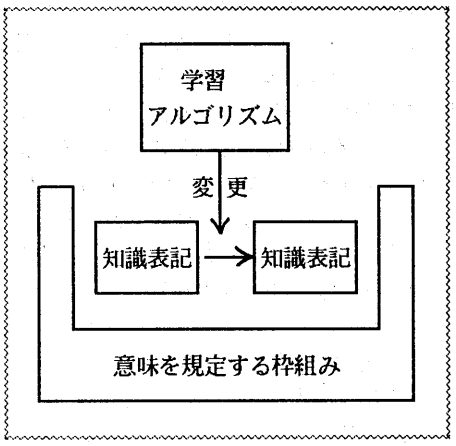

図3　知識表現系と学習システム

3．　学習システムのための知識表現系

本章では，学習システムのために使われてきたいくつかの知識表現系に言及し，それらをＧＬＰ（一般化された論理プログラム）という知識表現系に統合する．

3．1　文脈自由文法

すべての文脈自由文法は「論理プログラム」で表現できる．たとえば文脈自由文法：

```
A→ｒＡｓＢ
A→ｐｑ
B→ｋｌｍ
```

は，

```
(a [r $a s $b])
      ← (a [$a]), (b [$b]).
(a [p q]).
(b [k l m]).
```

と書ける．非終端記号Ａの生成する言語は，1引数述語 a を満足する文字列に対応する．それらは，明らかに，

$$\{(r)^n pq(sklm)^n \mid n=0,1,\cdots\}$$

である．

3．2　翻訳文法

佐藤ら[佐藤 86]は，翻訳の学習のための文法として，翻訳文法と呼ばれる知識表現を提案している（図４）．これは文脈自由文法を２つペアにしたような形の文法である．翻訳文法の表現は，そのままの形で「論理プログラム」に移し替えることができる（図５）．明らかに３つ以上の言語でも同様にできる（図６）．

```
I learn %01 -> %s  <-  私 は %01 を 学ぶ
    English -> %01 <- 英語
  Japanese -> %01 <- 日本語
```

図4　翻訳文法の例

```
(translate  [I learn $x]
            [私 は $y を 学ぶ])
  ← (nouns [$x] [$y]).
(nouns [English] [英語]).
(nouns [Japanese] [日本語]).
```

図5　上記の翻訳文法と等価な論理プログラム

```
(translate  [I learn $x]
            [私 は $y を 学ぶ]
            [Ich lerne $z])
  ← (nouns [$x] [$y] [$z]).
(nouns [English] [英語] [Englisch] ).
(nouns [Japanese] [日本語] [Japanisch] ).
(nouns [German] [ドイツ語] [Deutsch] ).
```

図6　３つ以上の言語の場合

3．3　ＬＳ／１の知識表現系

学習システムLS/1の知識表現系は標号網と呼ばれるネットワークが基本である．これはやはり「論理

プログラム」とみなすことができる．ネットワークの頂点が述語，ラベル付の辺が Horn clause に対応するのである．例を図７に示す．

```
(QA [what is $x] [it is $y])
        ← (ISA [$x] [$y]).
(QA [say example of $x] [$y])
        ← (ISA [$y] [$x]).
      …中略…
(QA [which is $x : $y or $z] [$z])
        ← (ISA [$z] [$x]).
  (ISA [dog] [mammal]).
  (ISA [cat] [mammal]).
  (ISA [pig] [mammal]).
  (ISA [hawk] [bird]).
  (ISA [dove] [bird]).
     …中略…
* (CAN [salmon] [swim]).
  (HAS [mammal] [leg]).
```

図７　LS/1の知識表現の例

３．４　ＬＳ／２の知識表現系

人間の学習では，主に視覚（情景）と聴覚（言語）という性質の異なる２つの情報源の間の関係を発見し，それをつぎつぎに組み立てることによって，世界の意味情報が知識体系として獲得されていくと考えられる．一方LS/1は任意の単語列（言語）だけを扱うという意味で，たとえて言えば，「聴覚」だけを持つ学習システムである．われわれは，LS/1の枠組みを自然に拡大して，「聴覚」だけでなく，「視覚」にあたる情報源を導入した学習システムLS/2を試作し，拡張を進めている．図８がLS/2の学習過程の例である．たとえば，この１７回目の会話では，

{(shape @18 circle) (color @18 blue)}

という「視覚」からの情報を見せられて，

[色 は 何 です か]

という「聴覚」からの質問を受け，

[青 です]

と答えている．

図９はこの学習過程の最後の時点でLS/2が獲得した知識（の一部）である．この知識もやはり「論理プログラム」とみなすことができる．ここには，引数として，３種類のデータが含まれている．それは，

単語列　… 文など
原子論理式の集合 … 絵など
Ｓ式　… 絵の要素

である．右に書いたのはそれらによって表現されるものの例である．新しいデータの導入により，LS/2が扱う変数は，単語列を表わす変数（$y など），原子論理式の集合に対応する変数（¥rest など），Ｓ式を表わす変数（*x など)の３種類になる．

```
?  [ 0] : [形 は 何 です か] {(shape @00 circle)} ??? [丸 です]
?  [ 1] : [形 は 何 です か] {(shape @01 triangle)} ??? [三角 です]
?  [ 2] : [形 は 何 です か] {(shape @02 square)} ??? [正方形 です]
?  [ 3] : [形 は 何 です か] {(shape @03 circle) (color @03 red)} ??? [丸 です]
                ……… 中略 ……
?  [15] : [形 は 何 です か] {(shape @15 triangle) (color @15 blue)} ??? [三角形 です]
o  [16] : [色 は 何 です か] {(shape @17 triangle) (color @17 yellow)} [黄色 です] [黄色 です]
o  [17] : [色 は 何 です か] {(shape @18 circle) (color @18 blue)} [青 です] [青 です]
```

図８　LS/2の学習過程の例

```
i_01 : 80 k2 p (v_r [形 は 何 です か]
                   {(shape *2_107^@ *2_176)}
                   [*2_185 です])
               (v_0 *2_176 [*2_185])
i_10 : 80 k2 p (v_r [色 は 何 です か]
                   {(shape *2_445^@ circle) (color *2_445^@ *2_532)}
                   [*2_541 です])
               (v_2 *2_532 [*2_541])
          ..... 中略 .....
i_02 : 80 k1 p (v_0 triangle [三角])        i_19 : 80 k1 p (v_1 circle yellow [丸])
i_03 : 80 k1 p (v_0 circle [丸])            i_21 : 80 k1 p (v_1 triangle blue [三角形])
i_04 : 80 k1 p (v_0 square [正方形])        i_11 : 80 k1 p (v_2 red [赤])
i_07 : 80 k1 p (v_1 square blue [正方形])   i_12 : 80 k1 p (v_2 blue [青])
i_08 : 80 k1 p (v_1 circle red [丸])        i_18 : 80 k1 p (v_3 yellow [黄色])
i_13 : 80 k1 p (v_1 square red [正方形])    i_17 : 50 k3 p (v_3 *2_1347 [*2_1356])
i_14 : 80 k1 p (v_1 triangle red [三角形])                 (v_2 *2_1347 [*2_1356])
```

図９　図８の学習によってLS/2が獲得した知識

LS/2が「視覚」からうける入力が論理式の集合であることに不満を感じる読者があるかも知れない．しかしここはこれでいいのである．「視覚」からの入力をもっともらしい形にして繁雑な学習問題を解くことはここでは考えない．我々の目標は高度な学習システムを達成するために都合のよい構造とは何か，その場合に共通に使える学習方式とはなにかを探すことである．

LS/2の扱う３種のデータがすべて「論理プログラム」で扱うことができて，その「論理プログラム」が学習にとって都合のよい構造をかなり一般的にとらえていることが重要なのである．また，LS/1やLS/2で使われた学習の枠組みや学習アルゴリズムが，そのまま一般的な「論理プログラム」を基礎とした学習理論の中で有効になることが重要なのである．LS/1やLS/2の翻訳学習[赤間 87]や質問応答学習の例の表面を見るのではなく，［背後にある本質的な構造］を見なければ学習研究は進展しない．

３．５　「論理プログラム」＝ＧＬＰ

このように学習研究に用いられた多くの知識表現系は「論理プログラム」とみなすことができる．しかしこの「論理プログラム」はこれまでは単に直観的に語られてきたにすぎない．それゆえに我々はいろいろな学習システムの［背後にある本質的な構造］に気づきながらも，それをLS/1やLS/2の学習の具体例などで間接的にしか表現することができなかった．

そのような困難を克服する鍵はＧＬＰの理論である．ＧＬＰとは generalized logic program の略である．これまでいろいろな学習システムの知識表現系の背後に見た「論理プログラム」とは，厳密にはＧＬＰである．ＧＬＰの理論は，論理プログラムの理論や文脈自由文法の理論やその他の興味深い知識表現系などの理論を統合する．この統合は，それらの個々の理論が記述されているレベルより抽象化されたレベルで理論を構成することによってなされる．［背後にある本質的な構造］はＧＬＰの言葉で明快に語ることができる．

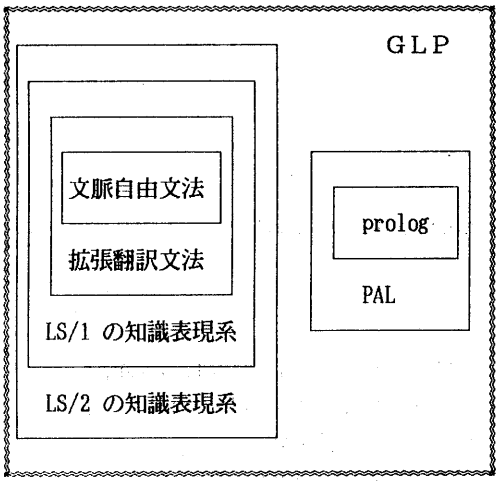

図１０　知識表現系の包含関係

４．　ＧＬＰの理論の基礎部分の概要

ＧＬＰを定義するための基礎となる構造を導入する．

４．１　縮小構造と縮小系

Prolog のパターンと代入の関係を見てみよう．パターン集合をＡ，基礎例集合をＧ，代入集合をＳとする．代入は，実はＡからＡへの写像であり，パターンは何回も代入をうけることができる．２つの代入を繰返し適用する変換は，等価的に，ある１つの代入と同じである．どんなパターンをも変化させない代入（恒等代入）が存在する．Ｇの元は任意の代入に対する不動点である．このような構造を一般化することによって，縮小構造と縮小系を定義する．

定義：縮小構造とは，つぎの条件を満たす３項組
＜Ａ，Ｓ，μ＞ である．

① μ：Ｓ→ partialmap(Ａ，Ａ)

② $\forall s_1, s_2 \in S, \exists s \in S$：
$\mu(s) = \mu(s_1) \circ \mu(s_2)$

③ $\exists s \in S, \forall a \in A$：
$\mu(s)(a) = a$
Ｓの元を縮小と呼ぶ．

定義：縮小系とは，つぎの条件を満たす４項組
＜Ａ，Ｇ，Ｓ，μ＞ である．

① ３項組 ＜Ａ，Ｓ，μ＞ は縮小構造である

② $A \supset G$

これではＧの制約が緩すぎる．我々は場合に応じてたとえば次のような条件をＧに課す．

$\forall s \in S, \forall g \in G : \mu(s)(g) = g$

Ｇに課す条件は場合によって変える必要がある．これに関しては，文献［赤間 89d］に譲る．

縮小系の定義には，変数の概念は出現せず，もっと抽象的なレベルで定義がなされている．この抽象度では代入という言葉は相応しくないので，縮小という言葉を用いている．縮小という呼び方は，ｓ∈ＳをＡの元ａに作用させることによってＡの元ｂが得られるとき，ａの表現する集合よりｂの表現する集合が小さくなることから来ている．縮小系の例をあげよう．

```
x
↓ ……x／（１　ｙ）
（１　ｙ）
↓ ……ｙ／（２　ｚ）
（１　（２　ｚ））
↓ ……ｚ／３
（１　（２　３））
```

ここでの縮小は代入にほかならない．つぎの例はクラス束縛変数の例である．

*x → *^物体 → *^動物 → *^犬 → ポチ

ここでの縮小とはクラス階層において制約をきつくすることである．

４．２　縮小系上の論理プログラム

縮小系という抽象的な構造の上で，論理プログラム（ＧＬＰ）を定義できる．ＧＬＰはもちろん Prolog プログラムを含む．

定義：縮小系<A，G，S，μ>上の program clause とは，

H ← B_1，…，B_n．

の形の式である．ここで，nは負でない整数であり，H，B_1，…，B_n は Aの元である．Hは head，B_1，…，B_n は body と呼ばれる．縮小系<A，G，S，μ>上の generalized logic program（GLP）とは，<A，G，S，μ>上の program clause の有限集合である．

詳細は省略するが，縮小系上の論理プログラムの意味（＝表現する集合）を，通常の理論の自然な拡張によって定めることができる[赤間 89a]．また縮小系上の論理プログラムに対して，ユニフィケーションやR計算解縮小（狭義の理論のR計算解代入に相当する）や，SLD導出や正解縮小（狭義の理論の正解代入に相当する）などが狭義の論理プログラムの場合とほぼパラレルに定義できて，健全性と完全性が証明できる[赤間 89ab]．

5． GLPの理論と学習システムの構築

GLPの理論を基礎として学習システム構築を議論すれば，以下に述べるようなさまざまな利点を享受できる．

5．1 学習システムの相互作用部分とSLD導出

知識表現系に対してSLD導出が定義されているので，それを利用することによって多くの場合，学習システムの相互作用部分のアルゴリズムを比較的効率的に実現することができる．たとえば，LS/1の設定において，LS/1の知識が図7の形で書かれているとする．そのとき，(what is dog) という質問に対する応答は，この論理プログラムに対する問い合せ：

(QA [what is dog] [$r])

を起動し，$r に対する解縮小（代入／束縛）をSLD導出で計算することによって得ることができる．また，次のような文法が学習されたとする．

(a [r $a s $b])
← (a [$a]), (b [$b]).
(a [p q]).
(b [k l m]).

このとき，[rrrpqsklmsklmsklm]が文法にかなっているか否かは，問い合せ：

(a [rrpqsklmsklmsklm])

を起動し，それが成功するか否かによって判定できる．

5．2 縮小系と一般化

縮小系は特殊化と一般化を論じるための枠組みを提供しする．特殊化は縮小演算そのものにほかならない．一般化はその逆の演算で次のように定義できる．この定義が ground でないパターンに対する一般化をも含んでいること，ユニフィケーションと双対の定義であることに注意したい．

定義：縮小系<A，G，S，μ>において，

p_1，p_2，x ∈Aとする．

$\mu(s_1)(x) = p_1$

$\mu(s_2)(x) = p_2$

を満たすs_1，s_2 ∈Sが存在するとき，xはp_1，p_2の一般化であるという．

n文字変数を含む文字列パターンの集合をもとにして，縮小系を構成できる[赤間 89c]．この縮小系において，文字列の一般化の例をあげる．

p_1＝”花子の髪の色は何色か”

p_2＝”馨の背の高さは何センチか”の２つに対する１つの一般化は，

x＝”$\alpha\beta\gamma$は何δか”

である．継承階層をさかのぼる一般化も，集合束縛変数の空間を縮小系として構成すれば[赤間 89d]，上記の一般化の１つの例として扱える．たとえば，

p_1＝*1^犬，p_2＝*2^猫

に対する１つの一般化は，x ＝*3^動物 である．

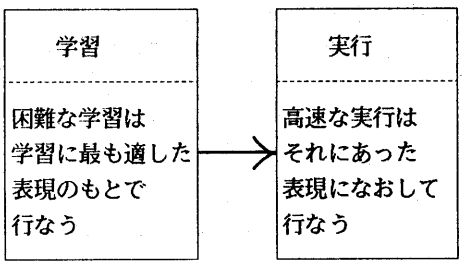

図１１　学習と実行の場を区別する

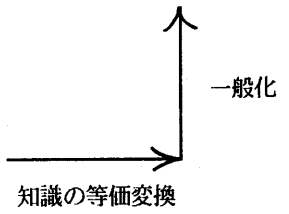

図１２　一般化の準備としての
知識の等価変換

５．３　論理プログラムの基礎的成果の継承

ＧＬＰの理論を基礎として学習システムの構築理論を作ることの利点の１つは，論理プログラムに関してこれまでに蓄積されてきた膨大な知見を自然に一般化して活用しながら議論を展開できることにある．とくに知識の等価変換（プログラム変換，部分計算，知識コンパイルなど）は論理プログラムの枠組みとの適合性が高く，研究が進みつつあるが，それらは学習システムの構築理論にとってもなくてはならない要素技術の１つである．それには少なくともつぎのような理由が考えられる．

① 学習に適した表記を用いて学習し，その後プログラム変換や部分計算や知識コンパイルなどでそれを等価変換して実行するための表記を得るような枠組みは，学習という情報処理の困難性を考えるとき，きわめて重要である（図１１）．

② 学習において知識を構造化，般化するするためには，その前段階として，知識の等価的な変換がしばしば有用である．知識の適切な等価変換は，しばしば価値ある一般化を準備するのである（図１２）．

５．４　適切な知識表現系の選択

ＧＬＰは基礎となる縮小系を変えることによって非常に多くの知識表現系（プログラム言語）を作りだすことができる．それはＣＬＰ（制約論理プログラム）の理論と同じ意味で知識表現系（プログラム言語）のスキーマだからである．そしてその自由度はＣＬＰを越えている．

この自由度は適切な学習システムの設計にとって必須のものである．たとえば，言語の学習を扱う場合には文を表現する必要がある．

```
[This is my pen]
[The girl is my sister]
```

これら２つの文から共通の構造を引出し，その構造を表現するのはたやすい．１つの答えは，

```
[$N is my $M]
```

である．ここで $N や $M は単語列を表す変数である．知識表現系として Prolog を選択し，文をリストで表現したらどうだろうか．

```
(This is my pen)
(The girl is my younger sister)
```

ほぼ同じに見えるので，これでも問題ないように思われるかも知れないが，これでは駄目なのである．

```
(This is my pen)
```

は実は，

```
(This . (is . (my . (pen . NIL))))
```

の略記であることを思いだそう．単語列が平坦な構造をしているのとちがって，リストは右に行くとどんどん深くなる．また $N が [The girl] を表現したのに対して，リストからはその構造をくずさないで，The girl　だけを取りだすことはできない．

リスト（Ｓ式）のなす構造は，学習対象である文の世界の構造とうまく整合しない．ＧＬＰの理論から見れば，これは学習対象と縮小系の不整合の問題と考えることができる．Ｓ式のなす縮小系ではなく，単語列のなす縮小系を選択すれば問題は解決する．

ＧＬＰの理論は縮小系の選択により非常に広い範囲の対象にうまく適合させることを可能にしている．また適合する縮小系を構成的に作りだすための方法も提供している[赤間　89cd]．

５．５　適切な抽象度の選択

ＧＬＰの理論に基づく議論では，縮小系の条件などにどの程度限定を加えるかを調整することによって適切な抽象度を選択することができる．上記の縮小系の定義に何の限定も加えなければ，すべてのＧＬＰに共通の性質を議論することになる．縮小系を特殊な構造，たとえば単語列のなす縮小系に限定すれば，それに特有の性質が議論できる．これは学習システム構築理論の体系を効果的に記述したり，本質的な構造を効果的に発見するためにとても有用である．

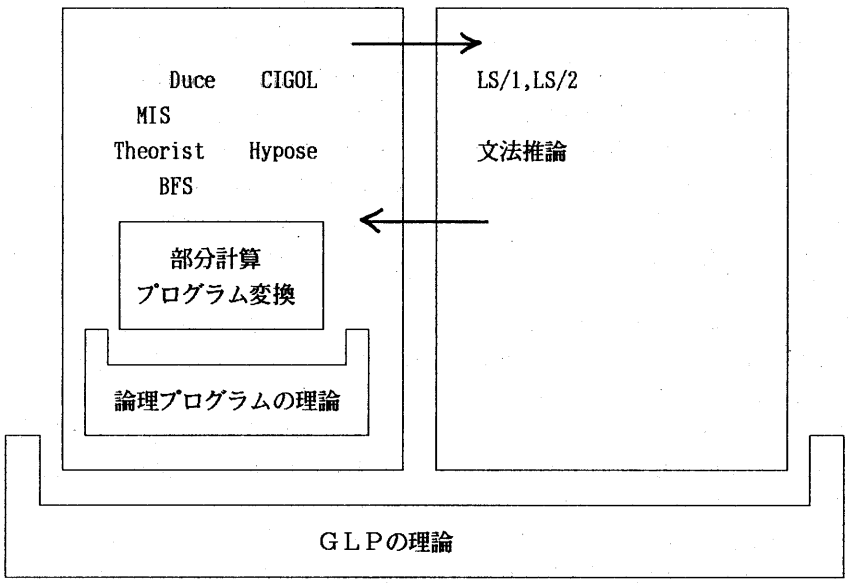

図１３　ＧＬＰの理論による学習研究発展の図式

５．６　明快な意味論

論理プログラムを核としたアプローチは，プログラム変換，部分計算，知識コンパイル，プログラムの虫の発見，学習などいろいろな場合に成果を挙げている．その成功は偶然ではない．成功の秘密の１つは明快な意味論にある．たとえば，プログラム変換や部分計算などは，それらがプログラムの意味を保存したままプログラムを別な（例えばもっと実行速度の速い）プログラムに変換するものである．そのような変換を大規模に実現するのは通常非常に難しい．ところが論理プログラムは単純明快な意味論を持つので，そのような変換の基礎をなす理論や手法を構築するのが比較的楽である．

このことは学習にも同様に当てはまる．学習ではプログラムを更新するが，プログラムは器であり，本当に問題なのはプログラムが表現している内容で

ある．理論的にはプログラムの内容はプログラムの意味論によって扱われる意味である．学習時に得られる情報は，意味に対する部分情報に他ならない．学習のアルゴリズムのなすべき仕事の核の部分は，不完全なプログラムを変更することによってその意味を修正したり，不十分なプログラムを変更して足りない意味を補ったりすることである．これらはいずれもプログラムの意味が鍵となっている．

これらのことから次の推測が導かれる．意味論を制する方法論が，今後の学習研究さらには知能一般の研究の大局を制することになるだろう．論理プログラムはこの意味で非常に有望であり，その長所はＧＬＰの理論に自然に継承されている．というより，ＧＬＰの理論は論理プログラムのこの長所を最大限拡大するために作られたというべきかもしれない．このように，ＧＬＰを基礎とした学習研究は，着実に発展する根拠を持つと言えるのである．

５．７　ＧＬＰによる統合と発展

ＧＬＰの理論による学習研究の発展の図式を図１３に挙げる．ＧＬＰは既存の多くの学習研究の知見を継承し，それらを統合し発展させることができる．たとえば，文法推論やLS/1,LS/2の学習は，文字列単語列などの世界を含むものであり，Prolog を基礎とする学習研究（Duce, CIGOL, MIS, Theorist, …）とは一応は別系統で展開されてきた．しかしＧＬＰを基礎とする研究法は，それらの共通性を発見したり，共通性を直接的に表現したり，また，一方での発見を他方に応用する可能性を拡大してくれる．我々はＧＬＰによって今までよりずっと広い視野にたって研究できることになる．ＧＬＰは，学習研究の発展に大きな力を発揮できる可能性がある．

文　献

[Jaffar 86] Jaffar,J. and Lassez,J.L.: Constraint Logic Programming,Technical Report, Department of Computer Science, Monash University, June (1986)

[Muggleton 87] Muggleton,S.: Duce, an oracle based approach to constructive induction, IJCAI 10, pp.287-292 (1987)

[Muggleton 88] Muggleton,S.: Machine invention of First-order Predicates by inverting resolution, Proc. 5th International confe rence on Machine learning pp.339-351 (1988)

[Poole] Poole,D.L.,Aleliunas,R. and Goebel,R : Theorist : a logical reasoning system for defaults and diagnosis, Knowledge Representa tion, N.J.Cercone and G.McCalla (eds.), Springer-Verlag, New York

[Shapiro 81] Shapiro,E.:Inductive Inference of Theories From Facts, Technical Report, Yale University, P.50 (1981)

[van Emden 76] van Emden,M.H. and Kowalsky,R.A. : The semantics of Predicate logic as a Prog ramming Language, J.ACM, vol.23, No.4 pp.733-742 (1976)

[赤間 87] 赤間清：帰納的学習システムＬＳ／１による翻訳の学習，人工知能学会誌，Vol.2 No.3, pp.341-349 (1987)

[赤間 88] 赤間清：知識はいかに獲得されうるか，認知科学の発展，日本認知科学会編，講談社 pp.161-188 (1988)

[赤間 89a] 赤間清：ＧＬＰの理論Ｉ，WOL'８９論文集 (1989)

[赤間 89b] 赤間清：ＧＬＰの理論II，情報処理学会，知識工学と人工知能研究会資料，63-9, pp.77-86 (1989)

[赤間 89c] 赤間清：ＧＬＰの理論III，情報処理学会，知識工学と人工知能研究会資料，63-10, pp.87-96 (1989)

[赤間 89d] 赤間清：ＧＬＰの理論IV，情報処理学会，知識工学と人工知能研究会資料，本号 (1989)

[佐藤 86] 佐藤理史，長尾真：文法推論に基づいた翻訳文法の学習方式，シンポジウム「学習の諸問題」報告論文集 pp.132-142 (1986)