

## 人工知能向言語 (I)\*

横井俊夫\*\*

### 1. まえがき

計算機の応用分野が拡大すると共に、いろいろな何々向(用)言語と呼ばれるものが考案されてきた。例えば、数値計算用、事務計算用、シミュレーション用、コンパイラ作成用などである。いずれにしても、計算機に対する新しい応用分野が確認されてくると、既存のプログラミング言語を用いて、種々の実験システムが作られる。既存言語では、表現が難しく、その分野に共通の要素、機構が発見されてくると、それらを基本要素として取り込んだ新しい言語が考案される。次には、この言語を用いて、さらに大規模な実験が繰り返される。

この新しい言語が良いものであれば、実験を容易にし、さらに新しい機構の発見を促す。この過程を繰り返すうちに、悪いものは淘汰され、改良が施こされ、その分野向けの言語として、広く使用されるようになる。したがって、人工知能向言語が考案され、使用されているという事実は、人工知能研究の対象が計算機応用の大きな分野となってきたことと、ある程度、共通の要素や機構が確認されてきたことを反映している。

しかし、人工知能研究は、新しい展開の初期の段階にあり、多くの不確定要素を含んでいる。したがって、人工知能向と称され、提案のみも含めると 20 件あまりになる言語やシステム (付録 (585 ページ参照)) の細かな比較を行っても無意味である。実際に製作され、かなりの使用実績のあるものに限定する。GPS (General Problem Solver), Micro-PLANNER, CONNIVER という一番大きな流れに着目し、最後に現状と今後の動向として、筆者が開発中のシステム

の紹介を含めて説明する。

説明を進めるに先だち、もう少し、人工知能向の人工知能の中身を限定する。ここでいう人工知能とは、問題の記述と解決に関する分野、最近の言葉でいえば、もう少し広義に、知識の表現と利用の機構を計算機上で研究しようという立場である。知識も、かなり一般的な、まだそれ程構造が明確になっていないものを対象とする。したがって、内部構造としては、リストとリストに対するパターン照合を基本とする。ゲームやパズルのように、それぞれ固有のデータ構造と処理機構を設定した方がよいものは対象としない。チェス盤は、リストより 2 次元配列と表わした方がよい。また、構文解析のように、文字列を対象とするには、より適した言語<sup>57), 64)</sup>が作られているので、対象外とする。

### 2. 単位プログラム

関連を明確にし、説明を容易にするため、言語やシステムが、記述対象とする問題解決システムの一般的な枠決めを行う。システムを構成する基本的な単位として、図-1 に図示したものを考える。これは最も一般的なものとしてまとめたものであり、実際には、もっと簡単な場合もあり、システムもこの単位プログラム一つから成る場合もあるし、何段かの階層構造を成す

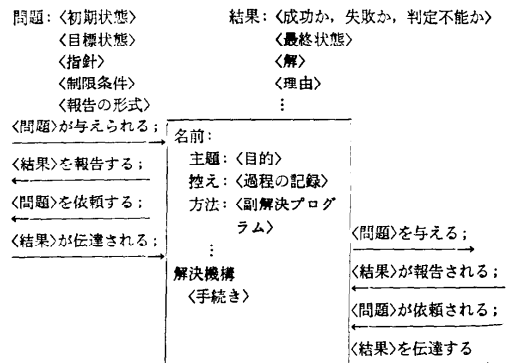


図-1 単位プログラム (単位プロセス)

\* Programming Languages for Artificial Intelligence (1)  
by Toshio YOKOI (Machine Inference Section, Information  
Sciences Division, Electrotechnical Laboratory)

\*\* 電子技術総合研究所パターン情報部推論機構研究室

場合もある。問題は次のようなものである。

“〈初期状態〉から出発して、〈目標状態〉を満たす状態になるようにせよ。その時、〈指針〉の情報を利用して解け。さらに〈制限条件〉を守り、結果を〈報告の形式〉に従がい報告せよ。”

結果は、成功か、失敗か、判定不能のいずれか。成功の場合は、最終的な状態とその状態に至るまでの過程、すなわち解、失敗や判定不能の場合は、その原因を報告する。この単位プログラムは、SIMULA-67のクラス<sup>4)</sup>と同じ様なコルティンで、一つの単位プログラムから多数の単位プロセスが生成される。

### 3. GPS<sup>5), 59)</sup>

計算機研究の初期の頃から、問題の表現と解決過程の一般性を追求して行われた代表的な研究である。多くの版のプログラムが作られたが、プログラミング技術も未発達時代に行われたものだけに、内部構造の整理も完全とはいえず、説明もかなり難解で古典的なものである。そこで、図-1の単位プログラムに照し合わせ、その骨組を図-2、図-3に整理して示す。実際のGPSは、もっと複雑な構造を持っており、本図

問題: <RENAME>  
 <DECLARE>  
 <TASK-STRUCTURE:  
 <TOP-GOAL: 〈初期状態〉を〈目標状態〉に transform せよ  
 <DIFF-ORDERING>  
 <TABLE-OF-CONNECTION>  
 <宣言された各データ構造の詳細な記述>  
 結果: 〈成功か、失敗か、判定不能か〉  
 <解>

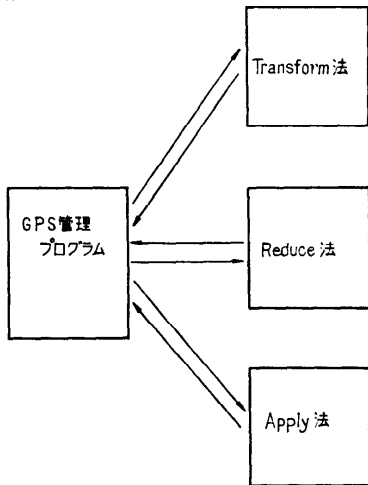


図-2 GPS の内部構造

#### GPS 管理プログラム

主題: 手段-目標解析法による問題解決。

控え: 〈記録〉

方法: Transform 法。

Apply 法。

Reduce 法。

解決機構:

〔〈問題〉が依頼される〕

〈問題〉を〈記録〉に照し、評価する;

すでに拒否→……………;

すでに失敗→……………;

複雑になりすぎる→……………;

続行は危険→……………;

〈問題〉を解くに適した〈方法〉をさがす;

〈方法〉に〈問題〉を与える;

〔〈結果〉が報告される〕

〈結果〉が

失敗→……………;

成功→……………;

(a) GPS 管理プログラム

Transform 法:

主題: 対象の変換

控え: 〈内部変数〉

解決機構:

〔問題〔〈対象〉を〈希望する対象〉に transform せよ〕が与えられる〕

〈対象〉と〈希望する対象〉を比較する;

違いがない→結果〔成功〕を報告する;

問題〔〈対象〉において〈違い〉を reduce せよ〕を依頼する;

問題〔〈結果〉を〈希望する対象〉に transform せよ〕を依頼する;

(b) Transform 法

Reduce 法:

主題: 違いを無くする。

控え: 〈内部変数〉

解決機構:

〔問題〔〈対象〉において〈違い〉を reduce せよ〕が与えられる〕

〈違い〉に関連する〈オペレータ〉をすべてさがす;

ない→結果〔失敗〕を報告する;

さがしたオペレータから一つ〈オペレータ〉を取り出す;

無→結果〔失敗〕を報告する;

有→問題〔〈オペレータ〉を〈対象〉に apply せよ〕を依頼する;

〔結果〔失敗〕が伝達される〕

取り出しを続ける;

〈結果〉を報告する;

〔問題〔他の解を求めよ〕が与えられる〕

取り出しを続ける;

(c) Reduce 法

Apply 法:

主題: オペレータを適用する。

控え: 〈内部変数〉

解決機構:

〔問題〔〈オペレータ〉を〈対象〉に apply せよ〕が与えられる〕

問題〔〈対象〉を〈オペレータ適用可能な状態〉に transform せよ〕を依頼する;

〈結果〉に〈オペレータ〉を適用して〈新しい対象〉を作る; 結果〔〈新しい対象〉〕を報告する;

(d) Apply 法

図-3 GPS 単位プログラム

は、かならずしも正確にGPSを反映したものではない。しかし、GPSの基本的な問題解決機構である、手段-目標解析法(mean-end analysis)の機構は十分

に反映している。解決過程全体の管理を行う GPS 管理プログラム (GPS Executive) と手段一目標解析法の 3 つの基本操作に対応する 3 つの方法 (Method) という 4 種類の単位プログラムからなる。管理プログラムは、プロセスとして唯一であるが、3 つの方法は、管理プログラムの制御のもとに、多数のプロセスが生成される。このシステムに対し、個々の問題が次のようなものを用いて表現される。

**RENAME:** 状態 (対象) の記述は、最も一般的なものとして、木構造を用いる。枝の名前として、FIRST, SECOND, THIRD, ……が仮定されるが、より問題に適した名前に換えるのがこの RENAME である。

**DECLARE:** 通常の宣言と同様に、それぞれの記号がどのデータ構造に対応するかを宣言する。FEATURE, TEST, OBJECT-SCHEMA, SET, MOVE-OPERATOR, RELATION などのデータ構造がある。さらに、構造の特殊なもの、例えば、RELATION の特殊なもの、EQUALS, NOT-EQUALS などを含めると、システムの手掛り語として、80 個程が用意されている。

**TASK-STRUCTURE:** 宣言された各記号に対し、データ構造の細部の記述を行う。さらに、〈指針〉として、DIFF-ORDERING と TABLE-OF-CONNECTION が与えられる。DIFF-ORDERING は、多くの“違い”を重要な順にリストしたもので、その“違い”とそれをなくす“オペレータ”との対応が TABLE-OF-CONNECTION である。

この問題が、GPS に与えられる。つまり管理プログラムの〔〈問題〉が依頼される〕が起動される。TOP-GOAL は常に、〔 $\sim$ transform せよ〕というものであるから、Transform 法を選び、〈問題〉を与える。後は、3 種類の方法 (多数のプロセス) が管理プログラムの制御のもとに、人間の最も基本的な問題解決機構である手段一目標解析法をシミュレートしていく。

GPS が後の人工知能向言語に影響を与えた主な機能を列挙する。

#### (I) 管理プログラムとモジュール化されたプログラム群\*

3 つの方法は、直接呼び合っているのではない。常

に管理プログラムを経由して行われる。例えば、〔オペレータ〈A〉を対象〈B〉に apply せよ〕という問題が依頼されると、管理プログラムは、〈記録〉に照し合わせ、4 種類位の判定を行う (図-3 の④)。同じ問題の依頼を既に受けているかどうかの判定をまず行う。同じものがある場合は、既に失敗、あるいは、他の理由で続行が拒否されているかの判定をし、〈記録〉に記し、他の実行可能な問題の選択を行う。同じものがない場合には、その目標が解を得る方向に向っているか否かを調べる。例えば、2 つの命題論理式、

$$L0: \sim(\sim Q \cdot P)$$

$$L1: (R \supset \sim P) \cdot (\sim R \supset Q)$$

が等しいことを証明する問題の場合、すなわち〔L1 を L0 に transform せよ〕が TOP-GOAL の場合を考える。オペレータは、命題論理の推論規則と論理記号に関する公理からなる。GPS は、対象である式の間の違いを見ながら、L1 を L0 に近づけるようにオペレータを選び適用していく。しかし、管理プログラムは、論理式があまりに長くなると、これ以上進めても解を得る見込みがないので、複雑すぎるという理由で他の問題の選択に移る。また、ある命題記号を取り除くというオペレータを適用した直後に、加えるというオペレータを適用するという問題が依頼された場合は、ループに陥る危険があるので、続行を一旦拒否し、他の問題の選択に移る。危険の少ない問題をすべて試した後、すべてが失敗した場合は、続行を拒否した問題を取り上げる。管理プログラムの探索の基本戦略は、単純な深さ優先 (depth first) ではなく、依頼された〈問題〉と過去の〈記録〉とを照し合わせ、評価を行い選択を決めて行く。

以上の GPS のプログラム構造の分析から、一般的に、問題解決システムの構造として、次の様なものを考えることができる。

“解決過程全体を監視する管理プログラムと、過程の部分部分を担当する専門プログラム群からなる。各専門プログラムは、それぞれの担当する処理についてのみ着目してプログラムされ、相互の関連の処理は、すべて管理プログラムにまとめられる。”

この考え方は、OS の制御プログラムの階層化とモジュール化の考え方と同一のものである。

#### (II) 対象の一般的な表現法と基本操作

GPS のほとんどのデータ (対象) は、木構造で、内部的には、属性一値対を基本要素とするリスト構造として表わされる。この木構造に対し、位置の指定法

\* 実際の GPS システムが、きれいにモジュール化されているわけではない。しかし、最大限の努力がなされていることと、その必要性を早くから指摘した点を評価する。

(LOC-PROG) と、その位置に対する読み出し、書き込み、比較などの基本操作が用意されている。さらに、木構造に対するパターン照合の操作が頻繁に使用される。例えば、先の命題論理式の証明の問題で、交換則を表わすオペレータ (FORM-OPERATOR) は、

$$((A \vee B) \text{ YIELDS } (B \vee A))$$

と表わされる。(A ∨ B) という木構造は、A, B を変数として、対象となる論理式 (同じく木構造) にパターン照合が行われる。このように簡単な場合ならばよいが、もっと複雑な木構造同志、しかも頻繁に照合が必要な場合は、照合を高速化する機構が必要である。管理プログラムは、〈問題〉 (木構造) が依頼されるたびに、〈記録〉に留められているすべての〈問題〉に対し、同一か否かの判定、すなわちパターン照合を行う。この照合を高速化する方法として、EPAM ネットと同様な弁別ネット (Discrimination Net) を用いる。

#### 4. Micro-PLANNER<sup>1), 74)</sup>

定理の証明やロボットの行動計画作りなどを目的として提案された PLANNER は、そのサブセットが Lisp 上に製作され、Winograd の著名なるシステム SHRDLU<sup>83)</sup> で使用された。これが、Micro-PLANNER と呼ばれるものである。自動後戻り (automatic backtracking) を基本解決機構とし、GPS の持つ内部のいろいろな機能を一般化し、言語化したものである。Micro- の方は、後に批判を受け、次に説明する CONNIVER にとってかわられるが、PLANNER 自身の方は、その後幾多の変遷を経て、呼び名も変わり、はなはだ難解な論文<sup>42), 43)</sup> がいくつか発表されているが、いまだ、使用に耐えるシステムは出来ていないようである。

Micro-PLANNER を先の単位プログラムで表わすと、問題解決システムとして最も自然な形は、**図-4** のようになる。GPS と対比させ、何が一般化され、拡張されたかを列挙する。

(1) 手続き型定理とリスト GPS 管理プログラムが、インタプリタの管理部に対応し、3つの方法が、定理と称する副プログラム群に対応する。GPS では、システムに組み込みであった方法が、自由にプログラムすることができる定理に一般化された。ただし、この定理も3種類の型に分けられる。CONSEQUENT, ANTECEDENT, ERASING である。これらは、それぞれ GOAL, ASSERT, ERASE という関数、つまり管理部に依頼される問題の型に対応する。知識の表

問題: 〈初期状態〉データ・ベース      結果: 〈SUCCESS (FAIL)〉  
 〈目標状態〉〈GOAL [パターン]〉      〈解〉パターン変数の値  
           〈ASSERT [パターン]〉      〈伝言〉  
           〈ERASE [パターン]〉  
 〈指針〉推せんリスト  
 〈制限条件〉  
 問題: 〈初期状態〉データ・ベース      結果: 〈SUCCESS, FAIL〉  
 〈目標状態〉(呼び出しパターン)      〈解〉パターン変数の値  
 (具体的には書かれるプログラムによる) 〈伝言〉 MESSAGE

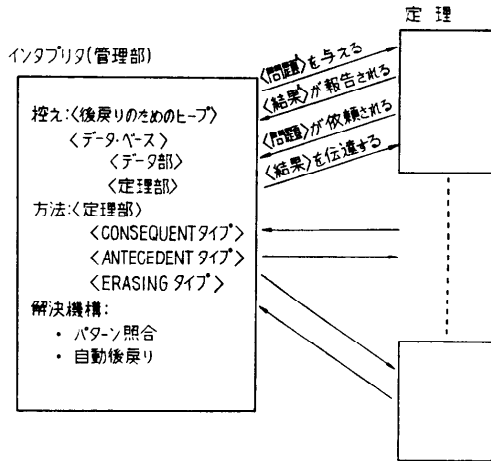


図-4 Micro-PLANNER の構造

現手段として、世界を支配する法則を表わすのに、プログラムによって任意の意味をもたせることのできるこの手続き型定理を、世界のあるがままの事実に対しては、最も一般的なデータ構造であるリストを用意した。

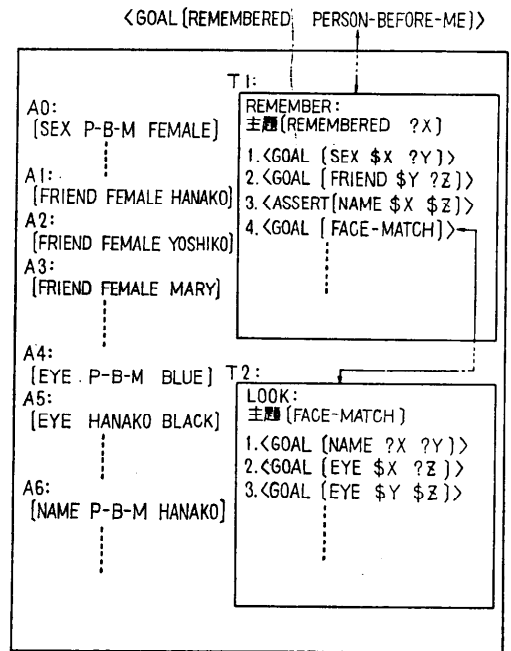
(2) データ・ベース GPS では、管理プログラムの控えとして局在化していたものを、データ・ベースとして顕在化させた。世界は変化する、その世界を反映する内部のモデルも、何らかの方法でその世界をシミュレートする必要がある。新しい事実が現われれば、データ・ベースに書込み (ASSERT)、なくなれば削除する (ERASE)。世界の現状は、各事実がデータ・ベースにあるか否かで判断できる。定理証明の枠の問題 (Frame Problem)<sup>38), 50), 67)</sup> の単純な解決法である。世界を構成する個々の事実は独立ではない。一部が変化すれば、全体としての整合性を保つため、他の部分も変化する。家族は構成員の幸・不幸を分かち合う小さな世界である。この相互の関連を反映するのが、ANTECEDENT 定理と ERASING 定理である。

(3) パターン照合と自動後戻り 管理部へ依頼する問題は、データ・ベースへの操作、さらにリストに対するパターン照合を伴う操作に一般化され、照合に

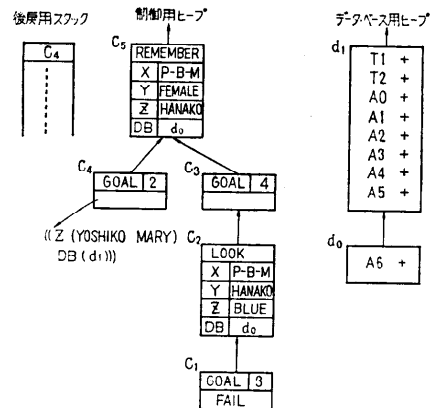
より生ずる多重一致の処理も、自動後戻りという一般的な制御機構で行われる。手続き型定理もデータベース中に置かれ、管理部が依頼された問題を適切な定理に与えるという過程は、パターン照合と一体化されたパターンによる呼び出し (pattern directed invocation) という機構に一般化される。

以上の機能を具体的に説明し、さらに内部の機構を明らかにするために、図-5 に1つの例を挙げる。この図も、正確に Micro-PLANNER の内部構造を表わしたものではない。次の CONNIVER の説明を容易にするため、整理し、少々変更をしてある。この例は、眼前の人物の名前を思い出すという、若干作為的なものである。データベース中には、友人の名前と特徴、眼前の人物の特徴が事実として記憶されている (A0~A5)。思い出すという行為 (T1) と特徴を見比べるという行為 (T2) が、CONSEQUENT 型の定理として存在する。REMEMBER 定理は、まず、人物の性別を確かめ、その性別に合う友人の名前を仮定し、顔の特徴を見比べよう依頼する。Look 定理は、仮定を取り出し、それに基づき、顔の各部分比べる。図-5の(2)は、Look 定理の3の GOAL 文が失敗した時点の状態を示す。細かな動作の説明は、紙面の都合上、割愛する。

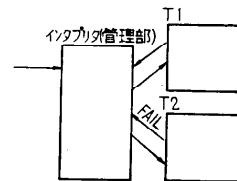
さて、ここでもう一度、GPS と Micro-PLANNER を見比べていただきたい。GPS では作りつけであった3つの方法が、Micro-PLANNER では、プログラム可能な定理になり、いろいろな解決システムを記述する自由度が増大した。しかし、全体を管理する部分、GPS の管理プログラム、Micro-PLANNER のインタプリタ (管理部) とともに作りつけである。人工知能向言語は、種々の問題解決過程を表現するための一般的な手段を提供するのを目的に作られた。Micro-PLANNER が作りつけとした自動後戻りの機構が、その目的に一致する、すなわち、すべての解決機構の根幹となるというのであればよいが、むしろ制約になるとか、実際の問題解決過程は、もっと異なったいろいろの機構を持つというのであれば、人工知能向言語として、重大な誤りを犯したことになる。現在、この管理部に固定してしまっても良い程、よく研究され、洗練されたアルゴリズムはない。むしろ、問題解決過程の動的モデルの重点は、この管理部にあり、人工知能向言語としては、この部分の記述に適した要素を用意しなければならない。この誤りは、厳しい批判<sup>75), 76)</sup>を受け、次に CONNIVER が提案される。もちろん、



(1) データベース



(2) 控文の内部



(3) システムの構造

図-5 モデル化の例 (Micro-PLANNER による)

Micro-PLANNER も出来るだけの努力はしている。盲目的な自動後戻りに対して、利用者が付け加え得る

修飾として、次の様なものが用意されている。

- ① データ・ベースの検索範囲を制限：データ・ベース操作関数には、推せんリスト (recommendation list) という〈指針〉を付けることができる。
- ② 監視要求：ループに陥らぬよう、監視するよう管理部に要求することができる。
- ③ 後戻り地点を指定：〈結果〉を伝える先を明示することができる。
- ④ 管理部の記録の変更：後戻りする必要がなくなった大部分を控えの記録から削除することができる。

①と効率を考えると②は、定理の中に記述すべきであるが、③、いわんや④にいたっては、管理部の中に記述すべきことである。モジュール化は、すべての鉄則である。Micro-PLANNERでは、実際のシステム作成にあたって、このモジュール化を崩す記述を強いられるため、プログラム全体の見通しを悪くし、虫取りを非常に困難にした。しかし、知識利用の動的モデルとして、積極的に評価すべきものも多数ある。若干の注釈をつけ列挙する。

(I) 手続き型の表現 プログラムする実行文のレベルをどうするかは議論は別にして、手続きを知識表現の有効な手段として明らかにした点は、大きな功績である。何故なら、我々の持っている知識にも手続き型が有効な役割を果しているからである。ホットケーキの作り方という知識は、材料と道具という初期状態と、出来上がったホットケーキという目標状態の記述によって表現されてはいない。ある抽象化されたレベルでの手続きとして表現されているはずである。

(II) データ・ベース 通常の意味とはことなり、ここでいうデータ・ベースは、枠の問題における、枠(Frame)に対応するもので、この枠と枠に対する読み、書き、削除の3つの操作が一般的な表現手段となることを明らかにしたこと。具体的な解釈は、枠はある地点でのある状態を表わす。読み操作は、その状態での事実の確認、書き、削除操作は、状態の遷移を表わす。しかし、GOAL, ASSERT という言葉は適切ではない。人間は、知的活動にあたり、多種類、多数の枠を用いる。ある種の枠に対する読み操作が、GOAL, すなわち、“ある目標を達成せよ”という意味に対応する、さらに、3つの基本操作に対応する法則(定理)を設け、パターンによる呼び出しという機構を設けたこと。当然のことながら、法則自体も、枠への書き込み、削除の対象となるし、法則を能動的に解釈して、枠へ

の操作を監視するもの、すなわち、デモン(Demon)という見方もできる。適、不適は別にして、この事実を、物語理解システムの文脈処理に適用しようとした例<sup>8)</sup>もある。

(III) 非決定的アルゴリズム<sup>26)</sup> データ・ベースへの操作を、手続きの中に埋込む方法として、パターン照合を選択文とする非決定的アルゴリズム(Non-Deterministic Algorithm)を用いたこと。ある範囲内では、かなり手続きの見通しを良くする。非決定的アルゴリズムは、Floydの定義より、一般化でき<sup>60)</sup>、さらに重要なことは、決定的アルゴリズムに変換する方法は、組み合わせも含めるといろいろあり、自動後戻りもその内の一つにすぎない。

### 参考文献

- 1) Baumgart, B.G.: "Micro-Planner Alternate Reference Manual." Stanford AI Lab. Operating Note no. 67. (Apr. 1972)
- 2) Bell, C.G.: "C. ai-A Computer Architecture for AI Research," Proc. FJCC, pp. 779-790, (1972).
- 3) Bobrow, D.G. and Raphael, B.: "New Programming Languages for AI Research," Computing Surveys, Vol. 6, No. 3, (September 1974).
- 4) Bobrow, D.G. and Wegbreit, B.: "A Model for Control Structures for Artificial Intelligence Programming Languages," Proc. 3d IJCAI, (Aug. 1973).
- 5) Bobrow, D.G. and Wegbreit, B.: "A Model for Control Structures of Multiple Environments," CACM. Vol. 16, No. 10. (Oct. 1973).
- 6) Burstall, R.M. et al.: Programming in POP 2, Edinburgh Univ. Press, (1971).
- 7) Chang, C. -L. et al.: Symbolic Logic and Mechanical Theorem Proving. Academic Press, New York, (1973).
- 8) Charniak, E.C.: "Toward a Model of Children's Story Comprehension," AI TR-266, MIT, (1972).
- 9) Dahl, O. -J.: Dijkstra, E.W. and Hoare C. A.R., Structured Programming, Academic Press, London, (1972). (日本語訳あり)
- 10) Davies, D.J.M.: "POPLER A POP-2 Planner," Memo MIP-R-89, School of Artificial Intelligence. (1971).
- 11) Deksen, J. et al.: "The QA 4 Language Applied to Robot Planning," Proc. FJCC, pp. 1181~1187 (1972).
- 12) Deutsch, L.P.: "A Lisp Machine with Very Compact Programs." Proc. 3d IJCAI, Aug. pp.

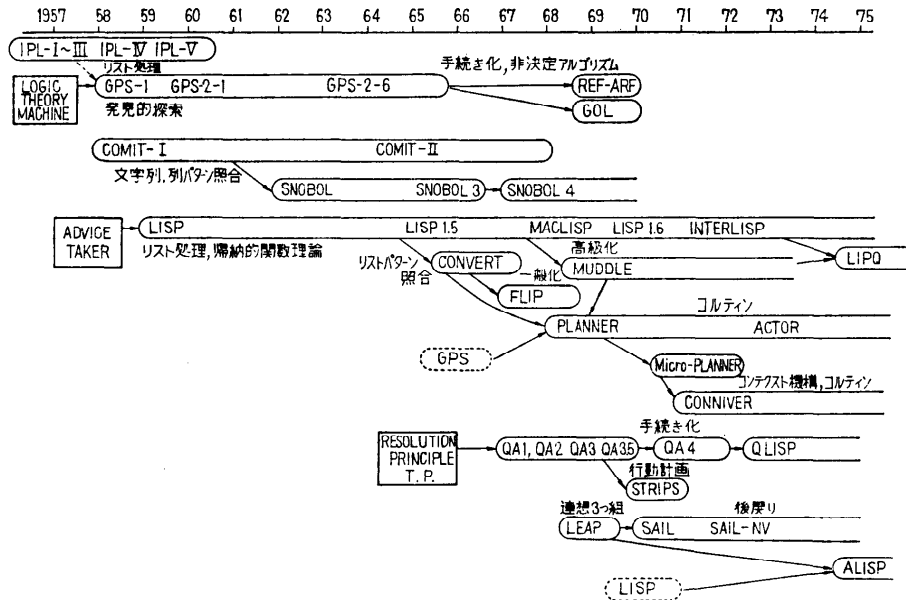


図-6 人工知能向言語とシステムの推移

697~703 (1973)

- 13) Dijkstra, E. W.: "Programming Considered as a Human Activity." Proc. IFIP. pp. 213~217 (1965).
- 14) Elcrok, E. W. et al.: "ABSET: A Programming Language Based on Sets, Motivation and Examples," Machine Intelligence, Vol. 6. (1970)
- 15) Ernst, G. W. and Newell, A.: "GPS: A Case Study in Generality and Problem Solving," Academic Press, New York, (1969).
- 16) "ALISP 利用者説明書", 電総研内部資料, (Mar. 1975).
- 17) "LISP User's Manual," EPICS-5-ON-2, 電総研, (Mar. 1976).
- 18) Fahlman, S.: "A Planning System for Robot Construction Tasks," AI-TR-283, AI Lab. MIT, May 1973, also Artificial Intelligence, Vol. 5, pp. 1~49 (Spring 1974).
- 19) Feldman, J. A. and Rovner, P. D.: "An ALGOL-Based Associative Language," CACM, pp. 434~449 (Aug. 1969).
- 20) Feldman, J. A. et al.: "Recent Developments in SAIL --An ALGOL Based Language for Artificial Intelligence," Proc. FJCC, pp. 5~7 (Dec. 1972).
- 21) Fikes, R. E.: "REF-ARF: A System for Solving Problems Stated as Procedures," Artificial-Intelligence, Vol. 1, (1970).
- 22) Fikes, R. E.: "STRIPS: A New Approach to the Application of Theorem Proving in Problem Solving," Artificial Intelligence. Vol. 2, pp. 189-208, (1971).
- 23) Fikes, R. E. et al.: "Learning and Executing Generalized Robot Plans." Artificial Intelligence, Vol. 3, pp. 251~288 (1972).
- 24) Findler, N. U. and Mckinzie, W. R.: "On a New Tool in Artificial Intelligence Research, An Associative Memory, Parallel Processing Language AMPPL-II," Proc, 1st IJCAI, pp. 259~270 (1969).
- 25) Fisher, D. A.: "Control Structures for Programming Languages," Ph. D. Thesis, Dept. of Computer Science, CMU, (May 1970).
- 26) Floyd, R. W.: "Nondeterministic Algorithm," JACM, Vol. 14, No. 4, Oct. pp. 636~644, (1967).
- 27) 淵, 横井, "アルゴリズムと試行錯誤", 数理科学, Vol. 12, (1972).
- 28) 淵, 横井, "推論機械", 信学会電子計算機研究会資料, EC 74-25, (1974).
- 29) Garney, T. D. and Kling, R. E., "User's Guide to QA 3.5 Question-Answering System," SRI AI-Group Technical Note 15, (Dec. 1969).
- 30) Goldstein, I. P.: "Understanding Simple Picture Programs," AI-TR-294, AI Lab. MIT, Sep. 1974, also Artificial Intelligence, Vol. 6, pp. 249~288 (1975).
- 31) Green, C. C. and Raphael, B., eds.: "The Use of Theorem-Proving Techniques in Question-Answering Systems," Proc. ACM National Conference. pp. 169~181 (1968).

- 32) Green, C.: "Theorem-Proving by Resolution as a Basis for Question-Answering Systems," *Machine Intelligence*, Vol. 4, (1969).
- 33) Green, C.: "The Application of Theorem-Proving to Question-Answering Systems," PhD Thesis, Stanford Artificial Intelligence Project Memo AIM-96, (June 1969).
- 34) Green, C.: "Application of Theorem-Proving to Problem Solving," *Proc. 1st IJCAI*, (May 1969).
- 35) Griswold, R.E.: "The SNOBOL4 Programming Language: 2nd Edition," Prentice-Hall, (1971).
- 36) Guzman, A. and McIntoch, H.V.: "CONVERT," *CACM*, Vol. 19, No. 8, pp. 604~615, (Aug. 1966).
- 37) Hawkinson, L.: "The Representation of Concepts in OWL," *Proc. 4th IJCAI*, Sept. pp. 107~114 (1975).
- 38) Hayes, P.F.: "The Frame Problem and Related Problems in Artificial Intelligence," *Artificial and Human Thinking*, A. Elithorn and D. Jones, eds., Elsevier Scientific Publishing Co., New York, (1973).
- 39) Hewitt, C.: "PLANNER A Language for Proving Theorems in Robots," *Proc. 1st IJCAI*, (1969).
- 40) Hewitt, C.: "Procedural Embedding of Knowledge in PLANNER," *Proc. 2d IJCAI*, pp. 167~182 (1971).
- 41) Hewitt, C.: "Description and Theoretical Analysis (Using Schemata) of PLANNER A Language for Proving Theorems and Manipulating Models in a Robot," Ph. D. Thesis, MIT AI-TR-258, (Apr. 1972).
- 42) Hewitt, C. et al.: "A Universal Modular Actor Formalism for Artificial Intelligence," *Proc. 3d IJCAI*, (Aug. 1973).
- 43) Hewitt, C.E. and Smith B.: "Towards a Programming Apprentice," *IEEE Trans. on SE*, Vol. SE-1, No. 1, pp. 26~45 (Mar. 1975).
- 44) Ichibiah, J.D. and Morse, S. P.: "General Concepts of the Simula 67 Programming Language," *Annual Review in Automatic Programming*, Vol. 7, pp. 65~93 (1972).
- 45) Kuipers, B.J., "A Frame for Frames Representing Knowledge for Recognition," *Representation and Understanding*, D. G. Bobrow and A. Collins, eds., Academic Press, New York, (1975).
- 46) 黒川: "LISP のデータ表現—TOSBAC 5600 LISP を中心にして—", *情報処理*, Vol. 17, No. 2, pp. 100~109 (Feb. 1976).
- 47) Liskov, B., and Zilles, S.: "Programming with Abstract Data Types," *Proc. of a Symposium on Very High Level Languages*, SIGPLAN Notes, pp. 50~59 (Mar. 1974).
- 48) Martin, W. A.: "OWL: A System for Building Expert Problem Solving Systems Involving Verbal Reasoning," MIT OWL-Note, (1974).
- 49) McCarthy, J. et al.: "LISP 1.5 Programmer's Manual," MIT Press, Second Edition, (1966).
- 50) McCarthy, J. and Hayes, P.: "Some Philosophical Problems from the Standpoint of Artificial Intelligence," *Machine Intelligence*, "Vol. 4, (1969).
- 51) McDermott, D. V.: and Sussman, G. J., "The Conniver Reference Manual," MIT AI Lab. AI-Memo No. 259, May 1972, Updated No. 259a, (Jan. 1974).
- 52) McDermott, D. V.: "Assimilation of New Information by a Natural-Language Understanding System," AI Lab., MIT, TR-291, (Feb. 1974).
- 53) Minsky, M.: "A Framework for Representing Knowledge," *The Psychology of Computer Vision*, P. Winston, ed., McGraw-Hill, (1975).
- 54) Moon, D. A., *MACLISP Reference Manual*, Project MAC, MIT, (Apr. 1974).
- 55) Moore, J. and Newell, A.: "How can Merlin Understand?," *Knowledge and Cognition*, L. Gregg, ed., Laurence Erlbaum Associates, Potomac, Md., (1973).
- 56) Moses, J. et al.: "Proposal for Personal Computer Capable of Executing Large LISP Programs," *Proposal to ARPA*, (Mar. 1975).
- 57) 長尾, 辻井: "自然言語処理のためのプログラミング言語 PLATON," *情報処理*, Vol. 15, No. 9, pp. 654~661 (1974).
- 58) Newell, A., ed.: *Information Processing Language V Manual*, Prentice-Hall, Englewood Cliffs, (1961).
- 59) Newell, A. and Simon, H. A.: *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, (1972).
- 60) Nilsson, N. J.: *Problem-Solving Methods in Artificial Intelligence*, McGraw-Hill, New York, (1971). (日本語訳あり)
- 61) Nilsson, N. J.: "Artificial Intelligence," *Proc. IFIP*, pp. 778~801 (1974).
- 62) Pfister, G.: "A MUDDLE Primer," *Project MAC SYS. 11.01*, MIT (May 1972).
- 63) Pople, Jr. H.E.: "A Goal-Oriented Language for the Computer," *Representation and Meaning*, H. A. Simon et al., eds., Prentice-



Hall, (1972).

64) Pratt, V. R.: "A Linguistics Oriented Programming Language," MIT AI-Lab, AI Memo No. 277, (Feb. 1973).

65) Quam, L. H.: "Stanford LISP 1.6 Manual," Stanford AI Lab. Operating Note No. 28.4, (Sept. 1970).

66) Quillian, M. R.: "Semantic Memory," Semantic Information Processing MIT Press, (1968).

67) Raphael, B.: "The Frame Problem in Problem-Solving System," Artificial Intelligence and Heuristic Programming, N. V. Findler, ed., Edinburgh Univ. Press, (1971).

68) Reboh, R. and Sacerdoti, E.: "A Preliminary-QLISP Manual," SRI AI Center Technical Note 81, (Aug. 1973).

69) Rulifson, J. F. et al.: "A Language for Writing Problem-Solving Programs," Proc. IFIP. pp. 201~205 (1971).

70) Rulifson, J. F. et al.: "QA 4 A Procedural Calculus for Intuitive Reasoning," SRI Technical Note 73, (Nov. 1972).

71) 島田: "定理証明用言語 MICRO-PLANNER について", 情報処理, Vol. 14, No. 5, pp. 341~348 (May 1973).

72) 島田, 他, "LISP マシンとその評価", 信学誌, 掲載予定.

73) Smith, D. C. et al.: "MLISP 2," Stanford AI Lab. Memo AIM-195. (May 1973).

74) Sussman, G. J.: "Micro-Planner Reference Manual," MIT AI-Memo, No. 203 A (Dec. 1971).

75) Sussman, G. J.: "Why Conniving is Better than Planning?," MIT AI-Memo No. 255. (Feb. 1972).

76) Sussman, G. J. and McDermott, D. V.: "From-PLANNER to CONNIVER--A Genetic Approach," Proc. FJCC, pp. 1171~1180(1972).

77) Sussman, G. J.: "A Computational Model of Skill Acquisition," AI-TR-297, Artificial Intelligence Lab., MIT, (Aug. 1973), Published as A Computer Model of Skill Acquisition, American Elsevier Publishing Co., (1975).

78) Takeuchi, I. and Okuno, H.: "A List Processor LIPQ," 2nd Proc. USA-Japan Computer Con., pp. 416~421 (Aug. 1975).

79) Teitelman, W.: "Design and Implementation of FLIP A Lisp Format Directed List Processor," AFCRL-67-0514, BBN Report No. 1495 (Jul. 1967).

80) Teitelman, W.: INTERLISP Reference Manual, Xerox PAPC, (1974).

81) Tesler, L. G.: "The LISP 70 Pattern Matching System." Proc. 3d IJCAI, pp. 671~676 (Aug. 1973).

82) Vanlehn, K. A., ed.: SAIL User Manual, Stanford AI Lab. Memo AIM-204, (July 1973).

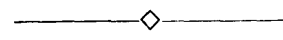
83) Winograd, T.: "Procedures as Representation for Data in a Computer Program for Understanding Natural Language," AI-TR-17, MIT, (1971), Published as Understanding Natural Language, Edinburgh Univ. Press, (1972).

84) Winograd, T.: "Five Lectures on Artificial Intelligence," PIPS-R-No. 5, ETL, (Aug. 1974).

85) Winograd, T.: "Frame Representations and the Declarative-Procedural Controversy," Representation and Understanding, D.G. Bobrow and A. Collins, eds., Academic Press, New York, (1975).

86) Yngve, V. H., "Computer Programming with COMIT-II." MIT Press. (1972).

87) 横井, 他, "定理証明のファームウェア化", 情報処理学会全国大会論文集, 57, (1975).



〔付録〕

提案のものも含め, 代表的な人工知能向言語と関連するシステム, さらに基盤となる記号処理用言語を表-1 にあげる。それらの相互の関連の概略とおおよその年代を図-6 (583 ページ参照) に示す。初歩的な解説は, 文献 3) が最適で, 広く人工知能の種々の分野全般の解説には, 文献 61)がある。なお, Micro-PLANNER の紹介には, 文献 27), 71) がある。

表-1 人工知能向言語

名 前	参考文献	名 前	参考文献
ABSET	14,	MLISP2	73,
ALISP	16,	MUDDLE	62,
AMPPL-II	24,	PLANNER	39, 40, 41, 42, 43
COMIT-II	86,	POP-2	6,
CONNIVER	51, 75, 76	POPLER	10,
CONVERT	36,	QA 1, QA 2	31,
FLIP	79,	QA 3	32, 33, 39
GOL	63,	QA 3.5	29,
GPS	15,	QA 4	11, 69, 70
IPL-V	58,	QLISP	68,
LIPQ	78,	REF-ARF	21,
LJSP	17, 54, 65, 80	SAIL	20, 82
LISP 70	81,	SNOBOL 4	35,
Micro-PLANNER	1, 74	STRIPS	22, 23

(昭和 51 年 4 月 5 日受付)