

技 報

UNIVAC TECHNOLOGY REVIEW

1986年2月 第10号

論 文

- 知的システムで使う問題解決戦略..... M. D. B-Janse 59
 変分原理による軸対称殻弾性方程式の導入..... 藤野 勉 70

報 告

- 線形計画法のためのモデル記述言語..... 前田英次郎・森本皓夫・保科 剛 1
 仕様記述法の味見..... 加藤潤 14
 データ構造に基づくテスト・ケース設計法..... 峰尾欽 26
 CAD エキスパート・システム..... G.D.Nelson, R.A.Higgins 35
 漢字の意味を利用した係り受け辞書による
 カナ漢字変換システムについて..... 稲永紘之・橋本和博 48

TECHNOLOGY TREND

- CAD/CAM 技術に欠けているもの..... 川合平夫 83
 機械翻訳の現状と将来..... 桑野龍夫 84

NEW PRODUCTS..... 89

EDITORS' NOTE 表2

応用人工知能の主な目標は、ある仕事について人間の行動に似た行動を示す問題解決システムを構築することである。Brouwer-Janse の「知的システムで使う問題解決戦略」は、この問題解決の研究分野を概説したものである。まず、人工知能と認知心理学における問題解決の関係を考察し、次に人工知能技法から生まれ、人間システムに適用される問題解決戦略、および問題解決モデルを論じている。最後に方法的問題を述べ、人間の専門家と非専門家を使う問題解決の戦略についての研究とその分析データを述べる。

任意形状の弾性体について、その荷重による変形を支配する弾性方程式を導入するためには、一般にベクトル（微分方程式）法とエネルギー（変分原理）法とがあるが、形状の複雑化と共に後者のほうが容易となる。とくに高い精度の弾性方程式を必要とするとき、力およびモーメント以上の力学的概念が必要となり、前者による弾性方程式の導入は困難となる。藤野の「変分原理による軸対称殻弾性方程式の導入」は、任意形状の軸対称殻について変分原理によって支配方程式の導入を行ったもので、さらに差分法によるその数値解析法について述べ、その一つの例として円環殻をとりあげている。

数値計画システム（主として、線形計画を扱うソフトウェア・パッケージ）への入力は、MPS 形式と呼ばれるものに統一されている。MPS 形式をマクロ化したマトリックス・ジェネレータは、MPS 形式の欠陥のある部分を補うものとして、広く使用されているが、やはり MPS 形式の不自然さを完全に逃れることはできない。前田らの「線形計画法のためのモデル記述言語」は、MPS 形式と、マトリックス・ジェネレータの短所と、その対策として新らしく開発した線形計画モデル記述法とについて述べている。

Liskov-Zilles 以来、プログラムの仕様記述についての議論が盛んである。しかし、仕様の大切さや記述方法の持つべき特質などを論じたものが多く、どのように書いたかといったような、現場のプログラミングの際に役立つ具体的な方針となるようなものが少ない。加藤の「仕様記述法の味見」は、よい

仕様の記述ないしは課題の提示のための考え方や書き方を模索して、そこで学んだ記述の枠組みや記述を小さな課題に適用した報告である。いくつかの記述法を試み、その結果に対して、書きやすさ、読みやすさ、整合性検査やデバッグ、実現のための手掛り、適用範囲などについて比較検討している。

プログラムの生産現場において、いかにテスト・データを準備するかということは常にプログラマを悩ませる問題である。峰尾の「データ構造に基づくテスト・ケース設計法」は、プログラマの主観や経験に左右されない系統的なテスト・データ選択法について論じている。この方法のアイデアは、入出力データの構造を状態推移図に変換することによって、テスト・データの選択基準を選定することに基づいている。この方法は、データ構造からプログラム構造を導く Jackson 構造プログラミングのようなプログラミング方法論にとくに効果的である。

VLSI チップ設計をより効果的にコンパクトなレイアウトを自動的に行わせるため、専門家の知識をエキスパート・システム化し利用する考え方がある。Nelson らの「CAD エキスパート・システム」は Sperry 社の同プロジェクトでのいくつかの試行的あるいは実用開発的な作業、すなわち知識獲得・探索・制御法などに関する基礎研究作業、チップ配置設計のためのアルゴリズム/エキスパート混合システムの開発作業、企業 CAD 知識ベースの開発作業についての報告である。

カナ漢字変換処理においては、文節分かち書きのカナ漢字変換から、最近では連文節あるいはべた書き入力文のカナ漢字変換への移行が見られるが、これらはいずれも文節単位にカナ漢字変換を行うことが基本である。稲永らの「漢字の意味を利用した係り受け辞書によるカナ漢字変換システムについて」は、接尾語や同形異義語の曖昧さを解消するために「漢字（の意味）情報を用いた係り受け処理」を行うカナ漢字変換処理に焦点を絞り、処理の概要と実験結果および今後の課題について述べている。

報告

線形計画法のためのモデル記述言語

A Model Description Language for Linear Programming

前田 英次郎, 森本 皓夫, 保科 剛

要約 数理計画システム（主として線形計画，LP を扱うソフトウェア・パッケージ）への入力
は，現在，MPS 形式と呼ばれるものに統一されている。LP は最適化手法の中でとくに強力な
ものであるが，その解法は特殊な内部データ形式を必要とする。MPS 形式はこの内部形式をほ
ぼ忠実になぞったもので，そこからモデルの意味を読みとることは不可能に近い。MPS 形式を
マクロ化した Matrix Generator は，MPS 形式の欠陥のある部分を補うものとして，広く使用さ
れているが，やはり MPS 形式の不自然さを完全に逃れることはできない。

LP モデルは，添字つき変数による数式で表すのが普通である。この形式をそのまま入力して
解が得られれば非常に便利である。本稿では，MPS 形式と Matrix Generator の短所とその対策
として新しく開発した LP モデル記述言語について述べる。

Abstract MPS format is the present industry standard of input data format for mathematical
programming systems. It is almost impossible to understand MP models written in the format,
because it is designed by the request from the simplex algorithm for LP. Most of matrix generators
are not free from the structure of the format.

This paper describes an MP system for 1100 series which accepts models written in usual
mathematical notation with subscripted variables and summations. Model parameterers are maintained
separately and combined with a model to generate an LP problem.

1. はじめに

$$\begin{aligned} l_i \leq g_i(x) \leq u_i \quad & i = 1, \dots, m \\ x \in X \end{aligned}$$

の制約の下で，

目的関数： $f(x)$

を最大（あるいは最小）にする変数 x の値を求める最適化問題を数理計画問題，これを解
く手法を数理計画法 (Mathematical Programming, MP) という。

g_i, f, X の性質によって異なったアプローチを必要とし，線形計画 (Linear Programming,
LP), 非線形計画 (Nonlinear Programming, NLP), 整数計画 (Integer Programming, IP)
などに分類される。 g_i, f が線形で $X = R^n$ であるもの，

$$\begin{cases} \text{maximize} & \sum_{j=1}^n c_j x_j & (\text{または minimize}) \\ \text{subject to} & l_i \leq \sum_{j=1}^n a_{ij} x_j \leq u_i & i = 1, \dots, m \end{cases}$$

が線形計画問題である。LP にはシンプレックス法という強力な解法があり，大型の問題
(m, n が 1000 以上) も短時間で解くことができる。非線形計画や整数計画は LP よりも
はるかにむずかしく，あまり大きな問題は解けない。

シンプレックス法の存在によって LP は強力な最適化手法として広く用いられている。
数理計画システム (Mathematical Programming System, MPS) の名前で数多くのアプリケ

ーション・パッケージが提供されているが、これらはすべて、LP のためのシンプレックス法とシンプレックス法の変形によって解くことのできる特殊な非線形計画 (Separable Programming, SEP) と混合整数計画 (Mixed Integer Programming, MIP) を対象とするものである。

これらの MPS に問題を与える方法、すなわち入力データは MPS 形式と呼ばれるものにはほぼ統一されている。これは、60 年代の代表的な LP コードである IBM 7090 のための LP/90 の入力形式を基礎として、IBM 360 のために MPS/360 が開発されるときに LP 解法の進歩を取り入れ、改良を加えたものである。シンプレックス法のアルゴリズムの要求するデータ構造に近い形をしていて、プログラムでの処理を容易にしているが、普通のモデルの記述法とはまったく異なる順序に情報を配置しなければならず、データ・リストからモデルを読み取ることが非常に困難である。モデルが大型になると入力データも大量になり、取扱いは超人的な労苦となる。そのため大型のモデルを手作業で作ることは次第に少なくなり、モデルを生成するプログラムが FORTRAN や COBOL で作られるようになった。

そのうち MPS 形式をマクロ化した Matrix Generator (MG) と称するパッケージが出現した。最初は計算時間が相当大きいこともあってあまり普及しなかったが、コンピュータの高速・低価格化とともに利用が盛んになってきた。Matrix Generator は大変便利なものであるが、モデルの記述とまったく異なった原理に基づく MPS 形式を土台としているためにわかりにくい部分も少なくない。

LP モデルを考えるには、普通の数式の形で記述したものが用いられる。この形そのままの入力から、シンプレックス法が要求するデータ形式を作成するプログラムがあれば、MPS 形式のように理解し難いものと格闘しなくてもすむ。われわれはモデル記述言語を設計し、UNIVAC シリーズ 1100 の上に開発した。その経験に基づいて、モデル記述言語の有用性を述べるのが本稿の目的である。

2. MPS 形式と Matrix Generator

LP モデルの記述法として、MPS 形式と Matrix Generator の持つ欠陥を例題によって説明しよう。

図 1 の問題 1 を考える。ごく普通の輸送問題を生産費を含む形に拡張し、それに他の要因による制約を一つ加えたものである。これを LP として定式化したものが図 2 である。MPS 形式では制約式が名前を持つ必要があるので、 s_i, d_j, z を制約式の名前として採用した。係数に実際の値を与えると、図 3 の形になる。図 3 の LP に対して、MPS 形式データは図 4 のようになる。 n 変数・ m 制約式のモデルが制約式の集まりと目的関数とで表現されるのに対して、シンプレックス法のアルゴリズムは問題を変数の集まりとして扱う。それぞれの変数の情報は一つにまとまっている必要があるので、MPS 形式では制約の係数行列 (a_{ij}) は列ごとに分割して記述されることになる。

NAME 行は問題に名前を与える。

ROWS 行に続く部分で制約式の型と名前を与える。名前は 8 文字以内、英字の後に英数字を並べたものである。名前前の N, L, G, E が制約式の型を表していて、

$$N: \sum_j a_{ij} x_j \quad \text{not constrained}$$

$$L: \sum_j a_{ij} x_j \leq b_i \quad \text{less than}$$

G : $\sum_j a_{ij}x_j \geq b_i$ greater than

E : $\sum_j a_{ij}x_j = b_i$ equal to

を意味する。つまり、COST は目的関数を表すもので、制約ではない。

COLUMNS 行に続いて制約式の係数を変数ごとにまとめて記述する(図4)。変数 X 11 は目的関数行 COST に係数 130, 制約式 S 1 と D 1 に係数 1 を持つ。変数 X 12 は COST に 150, S 1, D 2, Z に 1 を持つ。

RHS 行に続いて各制約式の定数項が与えられる。

図4の MPS 形式データを眺めると、いくつかの不都合に気付く。

問題 1

2工場, 3営業所を持つ会社がある。工場 i の製品 1 個当たりの原価 (p_i) と生産能力 (s_i), 営業所 j の需要量 (d_j), 工場 i から営業所 j への輸送費 (t_{ij}) は次のようである。

		i					
		p	1	2	3	s	
1	80	1	50	70	60	1	180
2	100	2	30	40	50	2	120
		d	90	135	70		

また、特別の事情で、工場 1 から営業所 2, 工場 2 から営業所 3 への輸送量の和が 50 でなければならない。

このとき、費用最小の生産輸送計画を作成せよ。

図 1 生産輸送計画

Fig. 1 Problem 1...Production and transportation

x_{ij} で工場 i から営業所 j への輸送量を表す。

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^2 \sum_{j=1}^3 (p_i + t_{ij}) * x_{ij} \\ &\text{subject to} && s_i : \sum_{j=1}^3 x_{ij} \leq s_i \quad i=1, 2 \\ &&& d_j : \sum_{i=1}^2 x_{ij} \geq d_j \quad j=1, 2, 3 \\ &&& z : x_{12} + x_{23} = 50 \quad x_{ij} \geq 0, i=1, 2, j=1, 2, 3 \end{aligned}$$

図 2 問題 1 の LP モデル (その 1)

Fig. 2 LP model 1 for problem 1

COST : $130x_{11} + 150x_{12} + 140x_{13} + 130x_{21} + 140x_{22} + 150x_{23} \rightarrow$ 最小

$$\begin{aligned} S_1 : & x_{11} + x_{12} + x_{13} && \leq 180 \\ S_2 : & && + x_{21} + x_{22} + x_{23} \leq 120 \\ D_1 : & x_{11} && + x_{21} && \geq 90 \\ D_2 : & && + x_{12} && + x_{22} && \geq 135 \\ D_3 : & && && + x_{13} && + x_{23} \geq 70 \\ Z : & && && + x_{12} && + x_{23} = 50 \end{aligned}$$

$$x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23} \geq 0$$

図 3 問題 1 の LP モデル (その 2)

Fig. 3 LP model 2 for problem 1

- 1) 制約式が読み取りにくい。

$$S1: X11+X12+X13 \leq 180$$

を知るためには、ROWS 部の第4行、COLUMNS 部の第12, 15, 19行、RHS 部の第32行の5行を見つけ、さらにそれ以外に S1 が現れないことを確認しなければならない。このような小規模な問題ならばともかく、100 を越える変数を扱う場合、デバッグは大変な苦勞である。LP は目的関数と制約式でできているものであるから、これらが直接認識できない記述法は不自然というほかない。

- 2) 変数名、制約式名に添字が使用できない。

変数名 X 13 は x_{ij} の $i=1, j=3$ の場合であるが、 x_k の $k=13$ の場合と考えられなくもない。X 1221 は、 $x_{1,2,2,1}, x_{12,2,1}, x_{12,21}, \dots$ のどれなのだろうか。これらの区別には別に文書が必要である。図2の記法ならば誤解は生じない。現実には変数名 AHB37XJ は AH [B 3, 7, XJ] を表すといった曲芸が常用されているのである。

- 3) 係数を直接数値で与えなければならない。

図2では、 $p_i + t_{ij}$ と書いたものが、図4では COST の係数として 130, 150, ... と他所で計算した結果を与えるしかない。また、第 t 期の在庫量を st_t 、購入量を p_t 、販売量を s_t としたときのバランス式

$$st_{t-1} + p_t - s_t = st_t$$

は移項して、次式とした上で MPS 形式に表さなければならない。

$$st_{t-1} + p_t - s_t - st_t = 0$$

	NAME	TRANS	
1.			
2.	ROWS		
3.	N COST		
4.	L S1		
5.	L S2		
6.	G D1		
7.	G D2		
8.	G D3		
9.	E Z		
10.	COLUMNS		
11.	X11	COST	130.000000
12.	X11	S1	1.000000
13.	X11	D1	1.000000
14.	X12	COST	150.000000
15.	X12	S1	1.000000
16.	X12	D2	1.000000
17.	X12	Z	1.000000
18.	X13	COST	140.000000
19.	X13	S1	1.000000
20.	X13	D3	1.000000
21.	X21	COST	130.000000
22.	X21	S2	1.000000
23.	X21	D1	1.000000
24.	X22	COST	140.000000
25.	X22	S2	1.000000
26.	X22	D2	1.000000
27.	X23	COST	150.000000
28.	X23	S2	1.000000
29.	X23	D3	1.000000
30.	X23	Z	1.000000
31.	RHS		
32.	RHS	S1	180.000000
33.	RHS	S2	120.000000
34.	RHS	D1	90.000000
35.	RHS	D2	135.000000
36.	RHS	D3	70.000000
37.	RHS	Z	50.000000
38.	ENDATA		

図4 問題1の MPS 形式データ

Fig. 4 Model in MPS format for problem 1

4) 制約式の上下限の指定が不自然である。

例にはないが、MPS 形式には変数 x_j のとりうる値の範囲を示す BOUNDS 部と上下から制約を受ける式

$$l_i \leq \sum_j a_{ij} x_j \leq u_i$$

を表現する RANGES 部とがある。BOUNDS 部は素直な仕様であるが、RANGES 部では、 l_i と u_i を指定しないで

$$l_i \text{ と } u_i - l_i$$

$$u_i \text{ と } u_i - l_i$$

のどちらかを指定するという不自然な形になっている。実はスラック変数の範囲が $[0, u_i - l_i]$ になるというシンプレックス法のアルゴリズムを意識しすぎた結果なのである。

5) データ量が膨大である。

係数を直接数値で与えるので、500 制約式、1000 変数という中規模の問題でも、各変数が平均 10 個のゼロでない係数を持つとして、ROWS 部 500 行、COLUMNS 部 5000 行のデータが必要になる。これ以外に RHS, BOUNDS, RANGES の各部が加わる。

6) 文書性に乏しい。

上記の 1)~5) の理由で、MPS 形式のデータを読んでモデルを理解することは、ほとんど不可能である。問題そのものの説明と定式化の結果、すなわち図 1 と図 2 に当たる文書を用意することが避けられない。

この問題を Matrix Generator, GAMMA で書いたものが図 5 である。

```

1.....DATA
2..... LIST (I)
3..... 1 2
4..... LIST (J)
5..... 1 2 3
6..... TABLE TAB
7..... * P 1 2 3 S
8..... 1 80 50 70 60 180
9..... 2 100 30 40 50 120
10..... D 90 135 70
11.....PROBLEM TRANS
12..... COST,OBJ
13..... S(I),MAX
14..... RHS,RHS=(TAB,(I),S)
15..... D(J),MIN
16..... RHS,RHS=(TAB,D,(J))
17..... Z ,FIX
18..... RHS,RHS=50.0
19.....MATRIX
20..... X(I)(J)
21..... COST=(TAB,(I),P)+(TAB,(I),(J))
22..... S(I)=1.0
23..... D(J)=1.0
24..... Z =1.0 ,IF((I).IM.1.AND.(J).IM.2)
25..... Z =1.0 ,IF((I).IM.2.AND.(J).IM.3)
26.....ENDATA
    
```

図 5 問題 1 の MG プログラム
Fig. 5 MG program for problem 1

LIST コマンドでリスト $I = \{1, 2\}$, $J = \{1, 2, 3\}$ を、TABLE コマンドで表 TAB を定義する。TAB は図 1 の p_i, t_{ij}, s_i, d_j をまとめたものである。表の左端と上端が見出しで、要素は

- (TAB, 1, P): $p_1 = 80$
- (TAB, 2, 3): $t_{23} = 50$
- (TAB, 2, S): $s_2 = 120$

(TAB, D, 3): $d_3=70$

といった形で引く。

PROBLEM コマンドで問題の名前 TRANS を与え、それに続いて制約式の名前、型、定数項を与える。(I)と(J)はその場所にリスト I と J の要素を代入したものを生成することを意味する。つまり、第13, 14行は、

```
S1, MAX
RHS, RHS=(TAB, 1, S)
S2, MAX
RHS, RHS=(TAB, 2, S)
```

をまとめたものである。結局、7個の制約式 COST, S1, S2, D1, D2, D3, Z が定義される。OBJ, MAX, MIN, FIX は制約式の型で、それぞれ, N, L, G, E 型に当たる。定数項は

```
RHS, RHS=...
```

の形で与える。

MATRIX コマンドに続いて、第20行に変数名 X(I)(J)つまり X11, X12, ..., X23 が与えられ、第21~25行にその係数が与えられている。たとえば X12 は制約式 COST に係数 (TAB, 1, P)+(TAB, 1, 2)=80+70=150, S1, D2, Z に係数1を持つ。第24行は、Iの値が1で、Jの値が2ならば、制約式Zに係数1を持つ、と読む。制約式Zの部分が混み入っているのは、系統的に表現しにくいものだからである。

この例で見ると、MPS形式の欠陥としてあげたもののうち、5)のデータ量は完全に

問題 2

ある商品の第 t 期の購入単価 (pp_t) と販売単価 (sp_t) が次のように予測されている。

t	pp_t	sp_t
1	40	40
2	60	50
3	30	40
4	70	60
5	50	70

在庫単価は1期につき5、在庫能力は300である。初期在庫はないものとする。購入量は各期とも500以下とする。

このとき、利益を最大とする購入・在庫・販売計画を作成せよ。

図 6 購入・在庫・販売計画

Fig. 6 Problem 2...purchase, stock and sales

p_t : 第 t 期の購入量
 s_t : 第 t 期の販売量
 st_t : 第 t 期の期末在庫量

変数を上のように定義する。

$$\begin{aligned} & \text{maximize} && \sum_{t=1}^5 (sp_t * s_t - pp_t * p_t - 5 * st_t) \\ & \text{subject to} && b_t: st_{t-1} + p_t - s_t = st_t \quad t=1, 2, \dots, 5 \\ & && p_t: 0 \leq p_t \leq 500 \quad t=1, 2, \dots, 5 \\ & && s_t: s_t \geq 0 \quad t=1, 2, \dots, 5 \\ & && st_t: st_0 = 0 \\ & && st_t: 0 \leq st_t \leq 300 \quad t=1, 2, \dots, 5 \end{aligned}$$

図 7 問題 2 の LP モデル (その 1)

Fig. 7 LP model 1 for problem 2

	P ₁	P ₂	P ₃	P ₄	P ₅	S ₁	S ₂	S ₃	S ₄	S ₅	ST ₀	ST ₁	ST ₂	ST ₃	ST ₄	ST ₅	
PROFIT	-40	-60	-30	-70	-50	40	50	40	60	70		-5	-5	-5	-5	-5	
B1		1				-1					1	-1					= 0
B2			1				-1					1	-1				= 0
B3				1				-1					1	-1			= 0
B4					1				-1					1	-1		= 0
B5						1				-1					1	-1	= 0
変数の上限			500					∞			0			300			
変数の下限			0					0			0			0			

図 8 問題 2 の LP モデル (その 2)

Fig. 8 LP model 2 for problem 2

```

1.....DATA
2..... LIST (T)
3..... 1 2 3 4 5
4..... LIST (P)
5..... 0 1 2 3 4 5
6..... LIST (S)
7..... 1 2 3 4 5 6
8..... TABLE PRICE
9..... * 1 2 3 4 5
10..... P 40 60 30 70 50
11..... S 40 50 40 60 70
12.....PROBLEM INVENTORY
13..... PROFIT,OBJ
14..... B(T) ,FIX
15..... RHS,RHS=0.0
16.....MATRIX
17..... P(T)
18..... PROFIT =-(PRICE,P,(T))
19..... B(T) = 1.0
20..... BOUND,MAX = 500
21..... S(T)
22..... PROFIT = (PRICE,S,(T))
23..... B(T) =-1.0
24..... ST(P)
25..... PROFIT =-5.0, IF((P) .NM.0)
26..... B(S/P) = 1.0, IF((S/P).NM.6)
27..... B(P) =-1.0, IF((P) .NM.0)
28..... BOUND, FIX = 0.0, IF((P) .IM.0)
29..... BOUND,MAX = 300, IF((P) .NM.0)
30.....ENDATA
    
```

図 9 問題 2 の MG プログラム

Fig. 9 MG program for problem 2

解決され、3)は式の計算が可能になり(移項はだめだが)、2)の添字はスマートとはいえずなくとも一応解決されている。

欠陥 1)の制約式の分散については、ROWS と RHS が結合され、COLUMNS もまともっていて大変見やすくなっている。GAMMA の効果は大である。

しかし、残された問題も実は大きい。図 6 の問題 2 はごく単純な在庫管理問題である。図 7 のように定式化すると、図 8 の係数行列が得られる。これを GAMMA で書いたものが図 9 である。このモデルでは変数 P(T) と ST(P) に上限が付加されていて、

$$\text{BOUND, MAX}=500$$

などの形で表現されている。ここで (P). NM. 0 は「P が 0 でなければ」と読む。

$$\text{B(S/P)}=1.0, \text{IF}((\text{A}).\text{NM}.6)$$

の中の (S/P) は、「P の現在の値のリスト P での順位と同順位の S の要素」を意味する。すなわち、P=0 なら S=1, P=1 なら S=2, である。リスト P や S は、実は文字列であって整数ではないので、P+1 とは書けない。

図 9 の GAMMA プログラムから

B3: P3-S3+ST2-ST3=0

であることを読みとるのは、やはり楽ではない。結局 Matrix Generator プログラムの文書性も高いとは言いがたい。図7はやはり必要な文書である。また、図9は図7を GAMMA で実現する一つのやり方であって、等価な GAMMA プログラムは他にいろいろ考えらる。大型の問題では、普通のプログラム開発のように、図7から図9を導くための文書も必要になるだろう。

3. LP モデル記述言語

Matrix Generator にもっと便利な機能を追加してモデル記述を楽にすることは可能だし、現に GAMMA にもまだまだ道具はある。しかし、Matrix Generator は MPS 形式を生成するマクロ言語という性格を離れることがむずかしい。MPS 形式は、シンプレックス法が要求する内部データ形式をある程度利用者にも読めるように手直したものである。Matrix Generator の発想の原点には、アルゴリズムの要求する内部データ形式がある。そこを出発点として利用者に歩み寄ったものが MPS 形式であり、Matrix Generator であった。

しかし、考え直せば、内部データ形式を利用者が気にする必要はない。利用者が問題を正確に記述すれば、コンピュータがそれに対応した内部データ形式に変換してくれればよい。

図2や図7をできるだけそのままの形で入力できるには、LP モデル記述言語はどのような機能を必要とするかを考えてみよう。

- 1) データ記述はモデル記述から分離されている。
- 2) モデルを制約式の集まりととらえる。
- 3) 変数名、制約式名、定数に添字が使用できる。
- 4) 一つの制約式の記述で、一連の制約式を生成できる。
- 5) 式の中に総和が書ける。
- 6) 添字に式が使えらる。
- 7) $l_i \leq \sum_j a_{ij}x_j \leq u_i$ が書ける。
- 8) $\sum_j a_{ij}x_j + b_i \leq \sum_j d_{ij}x_j + e_i$ が書ける。
- 9) 添字の範囲を示す集合を指定できる。

などがあげられる。1), 3), 4), 9) は Matrix Generator でも実現されているものである。

問題1のモデル(図2)をわれわれのモデル記述言語(Model Description Language,

```

1  0  set    i,      /* 工場 */
2  0  set    j,      /* 営業所 */
3  0
4  0  const  p[i],    /* 工場原価 */
5  0  const  s[i],    /* 生産能力 */
6  0  const  d[j],    /* 需要量 */
7  0  const  t[i,j],  /* 輸送費 */
8  0
9  0  var    x[i,j]   /* 輸送量 */
10 0
11 0
12 0  min   :  sigma( i, j: ( p[i] + t[i,j] ) * x[i,j] )
13 0
14 0  s[i]   :  sigma( j : x[i,j] ) < s[i]
15 0  d[j]   :  sigma( i : x[i,j] ) > d[j]
```

図10 問題1の MDL 表現

Fig. 10 Model in MDL for problem 1

```

1  0  set      i      := { 1,2 }      /* 工場 */
2  0  set      j      := { 1:3 }      /* 営業所 */
3  0
4  0  const    p[i]   := 80, 100      /* 工場原価 */
5  0  const    s[i]   := 180, 120     /* 生産能力 */
6  0  const    d[j]   := 90, 135, 70  /* 需要量 */
7  0
8  0  table    t      /* 輸送費 */
9  0      *      1      2      3
10 0      1      50     70     60
11 0      2      30     40     50
    
```

図 11 問題 1 のデータ

Fig. 11 Data in DDL for problem 1

MDL) で表現したものが図 10 である。データはモデルとは分離して、データ記述言語 (Data Description Language, DDL) によって図 11 のように表現される。この二つを組み合わせることによって、図 3、図 4 の LP モデルが作成されて、解かれることになる。

/*...*/ の部分はコメントである。

```
set i,j
```

は、 i と j が添字集合であることを宣言し、

```
const p[i], s[i], d[j], t[i,j]
```

で、 p, s, d が 1 次元のベクトル、 t が 2 次元のマトリックスであることを宣言する。 i, j の要素と p, s, d, t の値は、図 11 で与えられる。図 10 はモデルの枠を与えるもので、図 11 を修正あるいは差し替えれば、新しい問題が簡単に作り出せる。

```
var x[i,j]
```

は、 $x[i, j]$ が値を決定すべき変数であることを示す。

```
min : sigma(i,j : (p[i] + t[i,j])*x[i,j])
```

は、目的関数

$$\sum_{i,j} (p_i + t_{ij})x_{ij}$$

を最小化する問題であることを意味する。

```
s[i] : sigma(j : x[i,j]) < s[i]
```

は、最初のコロン (:) まです、添字集合 i の各要素に対応して、 s_i と名付けた制約式が作られることを意味し、コロンの右の部分で、その制約が

$$\sum_j x_{ij} \leq s_i$$

であることを示している。総和の添字 i, j は、それぞれ図 11 で与える集合 $\{1, 2\}, \{1, 2, 3\}$ の上を動く。動く範囲を明示して

```
sigma(j = {1, 2, 3} : x[i,j]) < s[i]
```

または、

```
sigma(j = {1 : 3} : x[i,j]) < s[i]
```

と書くこともできる。図 10 の表記は

```
sigma (j=j : x[i,j]) < s[i]
```

の省略形なのである。 $j=j$ の右辺が添字集合で、左辺はその上を動く添字変数の名前を意味している。< は本来 \leq と書くべきところだが、文字がないので < を使用している。

図 11 では、添字集合 i, j とデータ p, s, d, t に値を与えている。

```
set j := {1 : 3}
```

は

```
set j := {1, 2, 3}
```

```

1 0 set t /* 期 */
2 0
3 0 const pp[t], /* 購入単価 */
4 0 sp[t] /* 販売単価 */
5 0
6 0 var p[t], /* 購入量 */
7 0 s[t], /* 販売量 */
8 0 st[{0}+t] /* 在庫量 */
9 0
10 0
11 0 max : sigma( t : sp[t]*s[t] - pp[t]*p[t] - 5*st[t] )
12 0
13 0 b[t] : st[t-1] + p[t] - s[t] = st[t]
14 0 p[t] : 0 < p[t] < 500
15 0 s[t] : 0 < s[t]

```

図 12 問題 2 の MDL 表現

Fig. 12 Model in MDL for problem 2

```

1 0 set t := { 1:5 } /* 期 */
2 0
3 0 const pp[t] := 40, 60, 30, 70, 50 /* 購入単価 */
4 0 const sp[t] := 40, 50, 40, 60, 70 /* 販売単価 */

```

図 13 問題 2 のデータ

Fig. 13 Data in DDL for problem 2

と同じである。

```

parameter nw=3
set j:= {1: nw}

```

と書くこともできる。ベクトルである p, s, d は、

```
const p[ i ]:=80, 100
```

の形で、行列である t はテーブル形式で値を指定することができる。他にスパース行列用の形式もある。

問題 2 は、図 12、図 13 のように書ける。

```
var p[t], s[t], st[{0}+t]
```

の $st\{0\}+t$ は、 st の添字が $\{0\}$ と t の和集合の上を動くことを示している。

```
st[t-1]
```

と添字に式を書いたり、

```
st[t-1]+p[t]-s[t]=st[t]
```

のように比較演算子の両辺に変数を含む式が現れたり、

```
0<p[t]<500
```

のように上下からはさんだり、が自由にできることも読みとれるであろう。つまり、図 2、図 7 がほとんどそのままの姿でコンピュータに入力されることがわかる。

図 10、図 12 は、文書として図 2、図 7 の代わりになりうるものである。問題をモデル化するとき、図 2、図 7 (あるいは図 10、図 12) は必ず必要である。MDL を使用すれば、それ以外の文書を不要にすることが可能である。

4. おわりに

われわれのモデル記述言語は通常の MPS で扱える Parametric LP, SEP, MIPなどを扱う形にまだなっていない。これらに適した形式を見出して取り入れる必要がある。LP モデルを扱うには、モデルを記述し、解が得られればそれでよいという訳にはいかない。得られた解からレポートをまとめることが非常に重要な仕事である。このためのサポートも必要になる。

記述された LP モデルの解が得られればそれで終わりとなるわけでもない。モデル化の誤り、無視した部分の重要性が判明する、データの誤り、単純な書き損じ、などと普通のプログラム作成と同様のトラブルはいくらでもありうる。モデルのデバッグも重要な作業である。LP 解が意外な解を与えた場合、その原因をさぐるためにコンピュータのできるサポートはいろいろ考えられる。

モデル記述の外側の問題としたデータの記述もまた大きな問題である。疎行列の扱いには十分スマートな方法を考案する必要がある。

LP を始めとする数理計画手法は、広く知られている割合に利用されていないように見受けられる。取り付けやすいソフトウェアがないことも原因の一つである。

モデル記述言語
 データ管理システム
 最適化プログラム
 レポート・ライタ
 デバッグング・サポート

などを整備して、思い立ったら気軽にモデルを作り、解いてみることでできる状況を作り出したいものである。

付 録

モデル記述言語の簡易構文規則

```

モデル ::= 宣言... 目的関数 制約定義...
宣言   ::= パラメータ宣言
        |   集合変数宣言
        |   定数宣言
        |   変数宣言
パラメータ宣言
        ::= par パラメータ定義 [ , パラメータ定義 ] ...
パラメータ定義
        ::= 名前 [ := 整数式 ]
集合変数宣言
        ::= set 集合変数定義 [ , 集合変数定義 ] ...
集合変数定義
        ::= 名前 [ := 集合式 ]
定数宣言
        ::= const 定数定義 [ , 定数定義 ] ...
定数定義
        ::= 名前 [ [ 添字 [ , 添字 ] ... ] ]
添字   ::= * | 集合式
変数宣言
        ::= var 変数定義 [ , 変数定義 ] ...
変数定義
        ::= 名前 [ [ 集合式 [ , 集合式 ] ... ] ]
目的関数
        ::= { min | max } : 式
制約定義
        ::= [ 名前 ] [ [ 制御並び ] ] : 制約式
    
```

制御並び
 ::= 制御 [, 制御]...
 制御 ::= 名前 [:= 集合式]
 | 整数式
 制約式 ::= 式
 | 式 = 式
 | 式 > 式 [> 式]
 | 式 < 式 [< 式]
 集合式 ::= 集合式 + 集合式
 | 集合式 - 集合式
 | 集合式 * 集合式
 | (集合式)
 | 集合変数
 | 集合定数
 集合定数
 ::= {要素 [, 要素]... }
 要素 ::= 整数式
 | 整数式: 整数式
 式 ::= 式 + 式
 | 式 - 式
 | 式 * 式
 | 式 / 式
 | - 式
 | 式 ** 式
 | (式)
 | 総和
 | 関数
 | 定数
 | 変数
 | 制御変数
 | パラメータ
 | 実数
 | 整数
 総和 ::= σ (制御並び: 式)
 関数 ::= 名前 (式 [, 式]...)
 定数 ::= 名前 [[添数 [, 添数]...]]
 変数 ::= 名前 [[添数 [, 添数]...]]
 添数 ::= 整数式
 整数 ::= 数字 [数字]...
 実数 ::= [整数]. [整数] [E [+|-] 整数]
 名前 ::= 英字 [英字 | 数字 | _]
 英字 ::= A | B | ... | X | Y | Z
 数字 ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

注: 整数式は整数値を取る項目だけで書かれた式である。

- 参考文献 [1] 小野勝章, 計算を中心とした線形計画法, 日科技連出版社, 1976.
 [2] 刀根 薫, 数理計画, 朝倉書店, 1978.
 [3] 日本ユニパック(株), 数理計画システム FMPS 概説書, 資料コード 483203103, 1980. 同数理計画システム FMPS. 資料コード 483205143, 1979. 同, 数理計画システム GAMMA, 資料コード 483205144, 1980.
 [4] IBM World Trade Corporation, Matrix Generator and Report Writer (MGRW) Program Reference Manual, No. SH 19-5014, 1972.

- [5] S. Katz, L. J. Risman, M. Rodeh, "A system for constructing models", *IBM Systems Journal*, Vol. 19, 1980, pp. 505~520.
- [6] J. Bisschop, A. Mearaus, "On the development of a general algebraic modeling system in a strategic planning environment", *Mathematical Programming Study*, Vol. 20, 1982, pp. 1~29.
- [7] R. Fourer, "Modeling languages versus matrix generators for linear programming", *ACM Trans. Math. Softw.* Vol. 9, 1983, pp. 143~183.
- [8] W. G. Kurator, R. P. O' Neill, "PERUSE: An interactive system for mathematical programs," *ACM Trans. Math. Softw.* 6, Vol. 6, 1980, pp. 489~509.
- [9] H. J. Greenberg, "A functional description of ANALYZE: A computer-assisted analysis system for linear programming models," *ACM Trans. Math. Softw.* Vol. 9, 1983, pp. 18~56.
- [10] H. P. Williams, *Model building in mathematical programming*, John Wiley & Sons, Ltd. 1978.

執筆者紹介 前田英次郎 (Eijiro Maeda)

昭和14年生, 38年東京大学応用物理学数理工学コース卒業, 47年Cornell大学OR学科にてMaster of Science, 44年(株)日本ユニバック総合研究所入社. 52年日本ユニバック(株)入社. 数理計画の研究開発, プログラム理論の応用などに従事.



森本皓夫 (Akio Morimoto)

昭和20年生, 44年東京電機大学電子工学科卒業, 同年(株)日本ユニバック総合研究所入社. 52年日本ユニバック(株)入社. 主として数理計画の開発応用に従事.



保科 剛 (Tsuyoshi Hoshina)

昭和32年生, 56年東京理科大学理学部応用数学科卒業, 同年日本ユニバック(株)入社. 主として数理計画の開発・応用に従事.



報告 仕様記述法の味見**A Sampler of Specification Description**

加藤潤三

要約 仕様記述ないし課題の提示の具体的方法を求めてボウリング・ゲームの得点計算に対し、下記のような記述を試みた。その結果に対して、書きやすさ、読みやすさ、整合性検査や虫とり、実現のための手掛り、適用範囲などについて比較検討を加えた。

試みた記述方法は、IPL approaches, Definite Clause Grammar, 属性文法, 表示的意味記述, Vienna Development Method, HDM, HISP, Concurrent Prolog, 染谷法, Z 言語法の各記法である。

Abstract A good discipline for program specification descriptions is yet a desideratum. As a result of writing 10 different descriptions of the score calculation for a bowling game, this paper reports our discussions about the taste of description methods in the following respects: ease of writing, reading, verifying, modifying, implementing, and applying. Frames and languages employed were: IPL Approaches, Definite Clause Grammar, Attribute Grammar, Denotational Semantics, Vienna Development Method, HDM, HISP, Concurrent Prolog, Someya and Z.

1. はじめに

Liskov-Zilles^[1] 以来、プログラムの仕様記述についての議論が盛んである。しかし、仕様の大切さや記述方法の持つべき特質などを論じたものが多く、どう書いたかといった現場のプログラムにとって具体的な指針となるものが少ないようである。

よい仕様の記述ないしは課題の提示のための考え方や書き方を模索して、そこで学んだ記述の枠組みや記述を小さな課題に適用してみた。理解を確かめるためには実例への適用が一番であると考えたからである。これはその報告である。

2. 課題

課題としてボウリング・ゲームの得点計算を選んだ。この課題を選んだ理由は強いて言うなら、小さいが自明でない程度の複雑さを持っていると考えたからである。規則集^[2]から関係のある部分を以下に抜粋する。

(ゲームの構成)

第101条 ボウリングの1ゲームは、10個のフレームをもって構成する。

競技者は、ストライクの場合を除き、それぞれのフレームで2回ずつ投球する。ただし、第10フレームがストライクまたはスペアーの場合には、サービス・フレームとし、ストライクの場合は、さらに2回投球でき、スペアーの場合には1回投球できる。

ゲームの成績は、適正な投球によって、倒されたピンの数をもって計算し、10個のフレームの合計点によって、これを表す。

投球されたボールとは、競技者の持っているボールが手から放れ、フェール・ラインを越えたものを投球されたボールという。

(ストライク)

第102条 第1回の投球によって、完全に配置された10本のピンを全部倒した場合は、これをストライクという。

ストライクの場合は、そのフレームの小さな枠の左に (×) 印をつけて表す。

ストライクの場合、そのフレームのピンの得点は、競技者がさらに次のフレームの得点
が加算されるので次のフレームの投球を終るまで空欄としておく。

ストライクに続いて、次のフレームでスペアーの場合には、その得点は 20 となる。

(ダブル)

第 103 条 続けて 2 回ストライクの場合は、これをダブルという。

この場合、第 1 のストライクのフレームにおける得点は、競技者がさらに次の投球を終
るまで空欄としておく。

ダブルの場合、最初のストライクの得点は、20 にそれに続く第 3 のフレームの第 1 球で
倒されたピンの数を加える。したがって、第 2 のストライクに続いて次の第 1 球で 9 本の
ピンを倒した場合には、第 1 ストライクのフレームには 29 を加える。

(トリプルまたはターキー)

第 104 条 続けて 3 回ストライクの場合は、トリプルまたはターキーという。

この場合、第 1 のストライクのフレームにおける得点は 30 となる。このようにして 10
フレーム全部について連続 12 回のストライクのときは、そのゲームは 300 の得点となり、
パーフェクト・ゲームという。

(スペアー)

第 105 条 どのフレームにおいても、第 2 回の投球によってピンの全部を倒した場合は、これをス
ペアーという。

スペアーの場合は、そのフレームの小さな枠の右に (／) 印をつけて、これを表す。第 1
回の投球で倒されたピンの数は、スペアーをとる前に小さな枠の左に記入し、そのフレ
ームの得点は競技者が次のフレームで第 1 回の投球を終えるまで空欄とし、次の第 1 球の得
点をスペアーとしての得点 10 に加え、その合計をもって記入する。第 10 フレームにおい
てスペアーの場合は、サービスフレームとして続いて第 3 回を投球できる。

(エラー)

第 106 条 一つのフレームで 2 回投球し、10 本のピンを全部倒すことができなかった場合は、これ
をエラーという。

エラーの場合には、第 1 回の投球で倒したピンの数を小さな枠の左に記入し、第 2 回の
投球で残ったピンを 1 本も倒さなかったとき小さな枠の右に (-) 印をつけて表す。また、
第 2 回の投球で倒したピンの数は小さな枠の右に記入し、そのフレームの得点は第 2 回の
投球が終了すれば直ちに合計し記入する。

(ファール)

第 108 条 第 1 回あるいは第 2 回の投球のとき、ファールした場合、第 1 投球のファールは左の枠、
第 2 投球のファールは右の枠の中へ (F) 印をつけてこれを表す。

フレームの第 1 回の投球でファールをした場合には、倒したピンは全部立て直され、第
2 回の投球で倒したピンだけをそのフレームで得点として計算する。

もし、第 1 投でファールをし、第 2 投で全部倒した場合はスペアーとなり。第 2 投で全
部倒さなかった場合はエラーとなる。

第 2 投でファールをおかした場合には、第 1 投で倒したピンの数だけをそのフレームで
計算する。

第 10 フレームにおいて、第 1 投でファールをし、第 2 投で全部のピンを倒してスペアー
となった場合は、サービス・フレームとして第 3 投を投げ、このフレームの計算はスペア
ーに第 3 投で倒したピンの数を加えたものとなる。

第 10 フレームで第 3 投がファールとなった場合は、最初の 2 回の投球により倒したピン
の数だけを計算する。

(ガター)

第 109 条 第 1 回の投球でボールがガターに入った場合は、左の枠の中に (G) 印をつけて表す。

スペアーをとるために投球した第2投がガターに入った場合は、右の枠の中に(-)印をつけてこれを表す。

ガターの場合は得点は0となるが、第2投により10本のピンを全部倒した場合はスペアーとして計算する。

スコアの記入例

フレーム	1	2	3	4	5	6	7	8	9	10						
	X	X	X	⊙/	F	8	9	F	X	8	-	G	9	X	X	8
得点	30	58	78	88	96	105	123	131	140	168						

3. 仕様の記述

ボウリング・ゲームの得点計算の仕様を記述する。ここで「仕様」というのは、「課題」、「算法」、「プログラム」を漠然と区別して、「課題の提示」ないしは「課題の背景」「課題のモデル化」といった程度の意味である。さて、われわれは課題の提示として次のような記述方法を試みた。

- 1) 述語論理の言葉によって：IPL approaches^[3]
- 2) 文法記述にならって：DCG^[4]，属性文法^[5]，表示的意味記述^[6]，VDM^[7]
- 3) 抽象データ型によって：HDM^[8]，HISP^[9]
- 4) 並列処理的に：Concurrent Prolog^[10]
- 5) 一般的：染谷^[11]，Z^[12]

われわれはゲーム体験から、得点計算を手続き的に理解していた。それをそのまま書けば、たとえば以下のような Pascal プログラムが完成する。

```

program bgame(input,output);
  label 999;
  type ftype=(strike,spare,error,out);
  var score,fn,f,s,n,index:integer;

  function gt:integer;
    var ch:char;
    begin
      read(ch);
      case ch of
        'X' : gt:=10;
        '/' : gt:=-10;
        'F','G','-': gt:=0;
        '1'..'9' : gt:=ord(ch) - ord('0')
      end;
    end;

  function kind(f,s:integer):ftype;
    begin
      if f=10 then kind:=strike
      else if (f<10) and (s=-10) then kind:=spare
      else if (f+s)<10 then kind:=error
      else kind:=out
    end;

  begin
    score:=0;
    f:=gt; s:=gt;
    for fn:=1 to 10 do
      begin
        case kind(f,s) of
          strike : begin

```

```

        n:=gt;
        if n=-10 then score:=score+20
        else score:=score+10+s+n;
        f:=s; s:=n;
    end;
    spare : begin
        n:=gt;
        score:=score+10+n;
        if fn<10 then
            begin f:=n; s:=gt; end;
        end;
    error : begin
        score:=score+f+s;
        if fn<10 then
            begin f:=gt; s:=gt; end;
        end;
    out : begin
        writeln('errorinput',f,s);
        goto 999
    end
    end
    end;
    writeln('score=',score):
    999:
end.

```

そうではなくて書くべきものは「仕様」であり、また記法として既出するようなものを考えていたから、それらによく合致するような課題の理解の仕方を考えることになった。そこで、ゲームを大略次のように理解しようとした。

- 1) ほとんどスクラッチから集合、関係を述語論理の言葉によって構成しながら（抽象）ゲームを記述する。
- 2) 文字列である入力データに注目し、課題記述を文法記述にならない、その構文的側面と意味的側面をほどよく調和させることを工夫する。
- 3) 抽象データ型として課題を把握するためにひそかに状態機械を考える。
- 4) 勝手に集合や関数（逆関数）を定義して（抽象）ゲームを構成する。スコアはゲームの「表現」と考え、表現に対しゲームが一意に存在する（あるいは存在しない）ことを注意して「仕様」を完成する。

以下、個別に記述を概観する。

3.1 IPL approaches

IPL approaches では、entities と propositions によって課題の世界をモデル化する。ボウリング・ゲームでは、Game, Frame, Throw, ... を entities として導入し、投球ルール、スコアリング・ルール、... をこれらの間に成り立つ propositions として与える。“集合”と“関係”という単純な概念によってのみモデル世界を構築できる。しかしその反面、記述（思考）のための道具が原始的なものしかないため記述量が増えた。

実現のための手掛かりを得るためには、記法の洗練、より簡便な組込み述語の導入などが考えられる。つぎに IPL approaches による仕様の一部を示す。

```

For all I For all t For all P For all G
    (G throws ItP iff Prb''''GtP)
For all I For all t For all P For all G
    (If G Throws Itp Then t Is among Nn
        & P Is among Bowlingpoint)
Games = {G/For some I for some t for some P (G throws ItP)}
Throws={I/For some G for some t for some P (G throws ItP)}
Games In I

```

```

Throws In I
Throw Number = {tI/For some G For some P (G throws TtP)}
Throw Number Is Among Fcn
Dr Throw Number = Throws
For all G Among Games
  ({tG throws TtP} = {1,2,...,K : {I/G throws TtP}})
  .
  .
  .
For all G For all F For all f (G frame Ff iff Prc''Gff)
For all G For all F For all f
  (If G frames Ff Then G Is among Games & f Is among Nn)
Frames = {F/For some G For some f (G frame Ff)}
Frames In I
Frame Number = {fF/For some G (G frames Ff)}
Frame Number Is among Fcn
Dr Frame Number = Frames
For all G Among Games
  ({fG frame Ff} = {1,2,...,10})
  .
  .
  .
For all I Among Throws For all F Among Frames
  (If I is first throw of F
   Then Throw Point : I is among digit U {strike})
For all I Among Throws For all F Among Frames
  (If I is second throw of F
   Then Throw Point : I is among digit U {spare})
For all I Among Throws For all F Among Frames
  (If I is first throw of F & Throw Point : I = Strike
   Then K : {I/I is second throw of F} = 0)

```

3.2 属性文法

記述の様式は Knuth^[5] をそのまま真似ている。構文部分を次の程度にとどめる。

```

bgame -> body
body   -> frame body | frame
frame  -> throw
        | throw throw
        | throw throw throw
throw  -> 'G' | 'F' | '-' | 'X' | '/'
        | '1' | '2' | '3' | '4' | '5'
        | '6' | '7' | '8' | '9'

```

上の構文範ちゅうに以下のような属性を付随させる。

symbols	attributes		
	inherited	synthesized	domains
bgame		score	{0, ..., 300}
body	no	score first second	{1, ..., 10} {0, ..., 10} {0, ..., 10}
frame	no	type first score second	{error, spare, strike}
throw	no order	score type	{1, 2, 3}

以上によって,

条件, 生成規則, 相続属性, 合成属性

の形式で意味部分を補足する. このとき, body の合成属性として body 内第一投 first, 第二投 second を用意したので, 実際のゲームに際しての得点計算と同様の計算規則の提示になっている.

3.3 DCG

3.2 節の属性文法の場合と同様のアイデアを DCG 処理を備えた Prolog によって課題を記述する. 具体的には Prolog プログラムである. 一部を次に示す. 最大の特徴は「実行可能である」ことである.

```
bowling(X) :- bowling_score(Total,X,[]),
              write('Total Score is '),
              write(Total).nl.
bowling_score(Total) --> frames(Total,1,_,_).
frames(T,FN,FTV,STV)
  --> { FN<10 },error_frame(FTV,STV),
      { Next_FN is FN+1 },!,
      frames(Next_T,Next_FN,Next_FTV,Next_STV),
      { T is Next_T+FTV+STV }.
frames(T,FN,FTV,STV)
  --> { FN<10 },spare_frame(FTV,STV),
      { Next_FN is FN+1 },!,
      frames(Next_T,Next_FN,Next_FTV,Next_STV),
      { T is Next_T+FTV+STV+Next_FTV }.
frames(T,FN,FTV,STV)
  --> { FN<10 },strike_frame(FTV),
      { Next_FN is FN+1 },!,
      frames(Next_T,Next_FN,Next_FTV,Next_STV),
      { STV is Next_FTV,
        T is Next_T+FTV+Next_FTV+Next_STV }.
frames(T,FN,FTV,STV)
  --> { FN=10 },ten_frame(FTV,STV,TTV),
      { T is FTV+STV+TTV }.
error_frame(FTV,STV) --> error_1st_throw(FTV),
                        error_2nd_throw(STV),
                        { FTV+STV < 10 }.
spare_frame(FTV,STV) --> error_1st_throw(FTV),
                        spare,
                        { STV is 10 - FTV }.
strike_frame(FTV) --> strike,{ FTV is 10 }.
error_1st_throw(FTV) --> ("G",!,{ FTV is 0 });
                        ("F",!,{ FTV is 0 });
                        digit(FTV).
error_2nd_throw(STV) --> ("-",!,{ STV is 0 });
                        ("F",!,{ STV is 0 });
                        digit(STV).
```

3.4 表示的意味記述

これは直接的な意味記述 (direct semantics), すなわち状態推移のみによるものを Oxford 流で行ったものである. その一部を次に示す.

SYNTACTIC DOMAINS

- $\Sigma \in \text{Sc}$ (Scores)
- $\Phi \in \text{Bd}$ (Bodies)
- $\Psi \in \text{Fr}$ (Frames)
- $T \in \text{Th}$ (Throws)

SYNTAX

- $\Sigma ::= \Phi$
- $\Phi ::= \Psi\Phi \mid \Psi$
- $\Psi ::= T \mid T T \mid T T T$
- $T ::= 'X' \mid '1' \mid '2' \mid '3' \mid '4' \mid '5'$
 $\quad \mid '6' \mid '7' \mid '8' \mid '9'$
 $\quad \mid 'G' \mid '1' \mid '-' \mid 'F'$

SEMANTIC DOMAINS

- Z (Integers)

$Ty = \{error, spare, strike, double\}$ (Types)
 $No = \{1, 2, \dots, 10\} \in Z$ (Frame Nos)
 $Pt = \{0, 1, \dots, 300\} \in Z$ (Points)
 $\sigma \in S = No \times Ty \times Pt$ (States)

SEMANTIC FUNCTION

$S : Sc \rightarrow Pt$
 $B : Bd \rightarrow S \rightarrow S$
 $F : Fr \rightarrow S \rightarrow S$
 $N : Th \rightarrow Z$

ここでは直接的意味記述によりたかたため、モデルとしてはやや人為的な変形を加えたものになっている。

3.5 VDM

以下は 3.4 節を VDM の記述言語 Meta IV によって書いたものの一部である。

Syntactic Domains

1. Game :: Properframes [Serviceframe]
2. Properframes = Properframe
3. Serviceframe = Onemorethrow | Twomorethrow
4. Properframe = Strike | Spare | Error
5. Onemorethrow = STRIKECHAR | (Errorchar - {Q})
6. Twomorethrow :: Onemorethrow (Onemorethrow | SPARECHAR)
7. Error :: Errorchar Errorchar
8. Spare :: Errorchar SPARECHAR
9. Strike :: STRIKECHAR
10. Errorchar = Q | ı | E | ı | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Context conditions

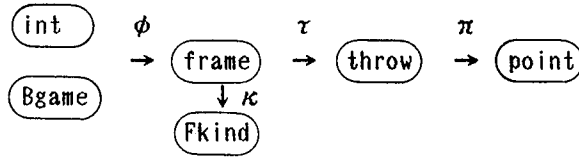
11. is-wf-Game : Game \rightarrow Bool
12. is-wf-Game (mk-Game(pf, sf)) =
13. len(pf) = 10
14. $\wedge (\forall i \in \{1:10\})(\text{cases } pff[i])$
15. (mk-Error(d1, d2) \rightarrow int(d1)+int(d2)<10 \wedge d1 \neq ı \wedge d2 \neq ı),
16. mk-Spare(d, s) \rightarrow d \neq ı,
17. mk-Strike(x) \rightarrow TRUE,
18. T \rightarrow FALSE)
19. $\wedge (\text{cases } sf$
20. (NIL \rightarrow is-Error(pf[10]),
21. mk-Onemorethrow(t) \rightarrow is-Spare(pf[10]),
22. mk-Twomorethrow(t1, t2) \rightarrow
23. is-Strike(pff[10])
24. $\wedge (t1 = \text{STRIKECHAR} \rightarrow t2 \in \text{Onemorethrow},$
25. $t1 \neq \text{STRIKECHAR} \wedge t2 \neq \text{STRIKECHAR} \rightarrow$
26. $(t2 \neq \text{SPARECHAR} \wedge \text{int}(t1) + \text{int}(t2) < 10) \vee t2 = \text{SPARECHAR},$
27. T \rightarrow FALSE),
28. T \rightarrow FALSE)

3.6 HDM

HDM は抽象データ型によるモジュール作成用言語とのことであるが、 \circ , \vee 関数などによって相当状態機械を意識させられるものである。表現の歪（過剰仕様）がでる。

3.7 HISP

染谷 (3.9 節後出) の仕様を抽象データ型として書いたものであり、種 (sort) とそれらの間の演算子を Gougen 図^[13] で次に示す。



3.8 Concurrent Prolog

構文部分の記述 `throws` と意味部分の記述 `scores` に分け、それぞれをプロセスとみなし、ゲームの各フレーム結果の流れをプロセス間通信とみなした。一部を次に示す。

```

bowling(X) :- throws(Ps,X?,[]),
              scores(Ps?,0,GT ),
              total_score(GT?).
throws(Ps,S0,S) :- frames(1,Ps,S0?,S).
frames(FN,[[FN,error,FTV,STV]|Ps],S0,S)
  :- FN<10,error_frame(FTV,STV,S0,S1) |
     Next_FN is FN+1,
     frames(Next_FN,Ps,S1?,S).
frames(FN,[[FN,spare,FTV,STV]|Ps],S0,S)
  :- FN<10,spare_frame(FTV,STV,S0,S1) |
     Next_FN is FN+1,
     frames(Next_FN,Ps,S1?,S).
frames(FN,[[FN,strike,FTV]|Ps],S0,S)
  :- FN<10,strike_frame(FTV,S0,S1) |
     Next_FN is FN+1,
     frames(Next_FN,Ps,S1?,S).
frames(FN,[[FN,ten,FTV,STV,TTV]],S0,S)
  :- FN=10 | ten_frame(FTV,STV,TTV,S0,S).
scores([[FN,strike,FTV]|Ps],T,GT)
  :- FN < 8,wait(Ps),Ps=[P1,P2|_],
     P1=[_,strike,STV],wait(P2),
     P2=[_,_,TTV|_],wait(TTV) |
     New_T is T+FTV+STV+TTV,
     scores(Ps,New_T,GT).
scores([[FN,strike,FTV]|Ps],T,GT)
  :- wait(Ps),Ps=[[_,_,STV,TTV|_] | _],
     wait(TTV) |
     New_T is T+FTV+STV+TTV,
     scores(Ps,New_T,GT).
scores([[FN,spare,FTV,STV]|Ps],T,GT)
  :- wait(Ps),Ps=[[_,_,TTV|_] | _],wait(TTV) |
     New_T is T+FTV+STV+TTV,
     scores(Ps,New_T,GT).
scores([[FN,error,FTV,STV]|Ps],T,GT)
  :- wait( STV ) |
     New_T is T+FTV+STV,
     scores(Ps?,New_T,GT).
scores([[FN,ten,FTV,STV,TTV]],T,GT)
  :- wait( TTV ) | GT is T+FTV+STV+TTV.
total_score(GT) :- wait(GT,T) |
                  write('Total Score is '),write(T),nl.
    
```

3.9 染 谷

抽象ボウリング・ゲームの母体(枠組)を集合 `Bgame`, `Frame`, `Throw`, `Point`, `Fkind` と、それらの間の演算 π (投球による得点), τ_1 , τ_2 (フレームにおける第一投, 第二投), κ (フレーム型すなわち, ストライク, スペア, エラー), ϕ (ボウリング・ゲームのスコアフレーム分け) によって定義する。これらに成立する公理を羅列する。ついでフレーム得点をゲームに対して、フレームから得点への関数を対応させる関数として定義する。さらにゲーム得点を、フレーム得点を用いてゲームから得点への関数として定義する。

1. 枠組

```

.0 <Bgame,Frame,Throw,Point,Fkind, pi, tau1,tau2,kappa, phi>
.  is BowlingGame <==>def
.1 Bgame ≠ ∅
.2 ∧Frame ≠ ∅
.3 ∧Throw ≠ ∅
.4 ∧Point = NaturalNumber
    
```

```

.5  $\wedge$ Fkind = {error,spare,strike,service1,service2}
.6.0  $\wedge$  $\pi$ :Throw $\rightarrow$ Point
.1  $\wedge$ ( $\forall t \in$ Throw)( $0 \leq \pi(t) \leq 10$ )
.7.0  $\wedge$  $\tau$ 1:Frame  $\rightarrow$ Throw
.1  $\wedge$  $\tau$ 2:Frame  $\rightarrow$ Throw U {undefined}
.2  $\wedge$  $\kappa$ :Frame  $\rightarrow$ Fkind
.3  $\wedge$ ( $\forall f \in$ Frame)
.4 { [  $\kappa(f)$ =error  $\Rightarrow$  ( $\tau$ 2( $f$ ) $\in$ Throw
.5  $\wedge$  $\pi$ ( $\tau$ 1( $f$ )) +  $\pi$ ( $\tau$ 2( $f$ )) < 10)]
.6  $\wedge$ [  $\kappa(f)$ =spare  $\Rightarrow$  ( $\tau$ 2( $f$ ) $\in$ Throw
.7  $\wedge$  $\pi$ ( $\tau$ 1( $f$ )) +  $\pi$ ( $\tau$ 2( $f$ )) = 10)]
.8  $\wedge$ [  $\kappa(f)$ =strike  $\Rightarrow$  ( $\tau$ 2 = undefined
.9  $\wedge$  $\pi$ ( $\tau$ 1( $f$ )) = 10)]
.10  $\wedge$ [  $\kappa(f)$ =service1  $\Rightarrow$   $\tau$ 2( $f$ )=undefined ]
.11  $\wedge$ [  $\kappa(f)$ =service2  $\Rightarrow$  [  $\tau$ 2( $f$ ) $\in$ Throw
.12  $\wedge$ ( $\pi$ ( $\tau$ 1( $f$ ))<10  $\Rightarrow$   $\pi$ ( $\tau$ 1( $f$ )) +  $\pi$ ( $\tau$ 2( $f$ ))  $\leq$  10 )]]
.13 }
.8.0  $\wedge$  $\phi$ :Bgame $\rightarrow$ Frame* /* the set of sequences in Frame */
.1  $\wedge$ ( $\forall b \in$ Bgame)
.2 { [ len( $\phi$ ( $b$ ))=10  $\vee$  len( $\phi$ ( $b$ ))=11 ]
.3  $\wedge$ ( $\forall i \in$ {1:10}) [  $\kappa$ ( $\phi$ ( $b$ )( $i$ ))  $\in$  {error,spare,strike}]
.4  $\wedge$ [  $\kappa$ ( $\phi$ ( $b$ )(10))=error  $\Rightarrow$  len( $\phi$ ( $b$ ))=10 ]
.5  $\wedge$ [  $\kappa$ ( $\phi$ ( $b$ )(10))=spare  $\Rightarrow$  len( $\phi$ ( $b$ ))=11
.6  $\wedge$  $\kappa$ ( $\phi$ ( $b$ )(11))=service1]
.7  $\wedge$ [  $\kappa$ ( $\phi$ ( $b$ )(10))=strike  $\Rightarrow$  len( $\phi$ ( $b$ ))=11
.8  $\wedge$  $\kappa$ ( $\phi$ ( $b$ )(11))=service2]
.9 }
2. 得点
2.1 frame-point:Bgame $\rightarrow$ [[{1:10}]  $\rightarrow$ Point]
.0 frame-point( $b$ )( $i$ ) =def
. /* defined by conditional expression */
.1 [  $\kappa$ ( $\phi$ ( $b$ )( $i$ ))=error  $\rightarrow$   $\pi$ ( $\tau$ 1( $\phi$ ( $b$ )( $i$ ))) +  $\pi$ ( $\tau$ 2( $\phi$ ( $b$ )( $i$ )));
.2  $\kappa$ ( $\phi$ ( $b$ )( $i$ ))=spare  $\rightarrow$  10 +  $\pi$ ( $\tau$ 1( $\phi$ ( $b$ )( $i$ +1)));
.3  $\kappa$ ( $\phi$ ( $b$ )( $i$ ))=strike $\rightarrow$ 
.4 [  $\tau$ 2( $\phi$ ( $b$ )( $i$ +1))  $\in$  Throw $\rightarrow$ 
.5 10 +  $\pi$ ( $\tau$ 1( $\phi$ ( $b$ )( $i$ +1))) +  $\pi$ ( $\tau$ 2( $\phi$ ( $b$ )( $i$ +1)));
.6  $\tau$ 2( $\phi$ ( $b$ )( $i$ +1)) = undefined  $\rightarrow$ 
.7 10 +  $\pi$ ( $\tau$ 1( $\phi$ ( $b$ )( $i$ +1))) +  $\pi$ ( $\tau$ 1( $\phi$ ( $b$ )( $i$ +2)))
.8 ] ]
2.2 score:Bgame $\rightarrow$ Point
.0 score( $b$ )=def  $\sum_{1 \leq i \leq 10}$  frame-point( $b$ )( $i$ )

```

3.10 Z

これは 3.9 節を Z 言語で書き直したものである。その一部を次に示す。

```

BOWLING-POINT =
def
strike = {'X'}
spare = {'/'}
digit = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'}
bowlingpoint = {strike, spare} U digit
int = {'0' $\rightarrow$ 0, '1' $\rightarrow$ 1, '2' $\rightarrow$ 2, '3' $\rightarrow$ 3, '4' $\rightarrow$ 4, '5' $\rightarrow$ 5,
'6' $\rightarrow$ 6, '7' $\rightarrow$ 7, '8' $\rightarrow$ 8, '9' $\rightarrow$ 9}
frame = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
throw = set f for f : NAT where f $\in$ segment[NAT] end
end BOWLING-POINT

BOWLING-GAME =
use BOWLING-POINT def
throw-rule = class

```



```

 $\phi$  : throw  $\rightarrow$  frame;
 $\gamma$  : throw  $\rightarrow$  bowlingpoint
end;
frame-order = subclass
  throw-rule where forall t for t : throw
  then
    (t+1)  $\in$  throw  $\Rightarrow$   $\phi$  (t+1) =  $\phi$  (t)
    or  $\phi$  (t+1) =  $\phi$  (t) + 1 end
  end;
frame-rule =
  subclass throw-rule where
  forall f for f : frame - {10} then
  exist t for t : throw where
  inverse( $\phi$ )(f) = {t} and  $\gamma$  (t) = strike
  or
  inverse ( $\phi$ )(f) = {t,t+1} and  $\gamma$  (t)  $\in$  digit and
  [[  $\gamma$  (t+1)  $\in$  digit and int( $\gamma$  (t))+int( $\gamma$  (t)) < 10]
  or  $\gamma$  (t+1) = spare]
  end
end

```

4. 味見と仕様記述の比較

小さな課題についていろいろ仕様を書き、また読んでみた。当然ながら課題の認識とその記述は書き手の主観と記述に使った規範に依存し、読み手の趣味もからんでくる。このような書き手の主観とか読み手の趣味を割り引いて“書きやすさとか読みやすさ”を主眼にして味見した結果を述べる。

- 1) 予備知識が不要なものは書きやすく、読みやすい。いい例が染谷 (3.9節) である。
- 2) 文法記述の方法は表層構造 (構文) から深層構造 (意味) への関数を定義するものであったが、逆に深層構造の表現関数によって表層構造を定義することも可能である。これが結構わかりやすく、また書きやすいものになる。これは深層構造が内在的 (intrinsic) な性質だけに注目して議論を展開できることに起因するためと考えられる。
- 3) 記述言語の背景にあるものの見方、いわば基本原理がわかりやすいと書きやすく、読みやすくなる。たとえば HISP がそうである。
- 4) 構造的プログラミングの精神すなわち抽象化と分割はわかりやすさの根本である。課題の分割の強力な原理として並行性 (Parallelism) を挙げることができる。試みてはいないが算体主導型もそうであろう。
- 5) 特定の記述言語を用いる場合と、そうでない場合では、書き手の能力に強く依存して相違が生ずる。すなわち、よく書けていればいいし、へたをすると読めないものになる。記述言語が一般的であればあるほど書き手に左右され、読みやすい仕様になったり、ならなかったりすることになる。言語を固定することは、実は言語が思考の道具として働くので課題の認識の手掛かりを与えてくれるから一面役立つ。しかし、記法が課題にとって有力な概念を支援していないと、仕様に表現の歪が生じ過剰仕様になりがちである (例, HDM)。固定した言語が簡素さを尊重して原始的な概念とわずかな構成法だけを与える場合、実際的でない面が生じる (例, IPL approaches)。
- 6) 「自由に書く」ことは多種論理 (Manysorted Logic) や、高階論理を勝手に使えるこ

とを意味し、とくに高階の関数などが有効であるようだ。したがって、記述言語がこのような概念を許容するかどうか記述能力とともに、わかりやすさの根元に影響する。

なお、次のような項目についても議論した。

- 1) 整合性の検査や虫とりのしやすさ……仕様がある程度の量の記述を伴うとき、書いた仕様の虫とりや整合性の検査がプログラムと同様やはり問題になる。その面で DCG/Prolog, HISP などの優位は明らかである。整合性の検査として強力な手段は型検査であろうから、LCF/ML などのような多様型 (Polymorphic type) を許す記述言語 (プログラム言語) が好ましいものになる。
- 2) 実現への手掛かり……文法記述にならった場合は、コンパイラの作成の要領の実現手段を示唆している。染谷は一見実現手段を隠しているようであるが、よく眺めると表現関数の逆関数を意味関数とみなすことによって実現の手掛かりが見えてくる。IPL によるものもデータベース構築のためのものであると考えれば、データベースに付随する手続き (意味ルーチン) を作成すればいいといった実現へのヒントが得られることになる。ところで、課題を構成的数学の枠組みで定理として述べ、その証明にしながらプログラムを作成 (自動作成) することは美しいやり方である。
- 3) 適用範囲……記述の適用範囲はそれが一般的かつ強力な概念を含有しているかどうかによって定まる。したがって勝手に書いた染谷が一番普遍的である。これから他の記述法への翻訳が容易であった。

5. おわりに

算体主導型など他にも見るべき規範は多い。残念ながら、まだこれらによる記述を試みていない。

あたりまえであるが、課題の持つ“構造”はその理解者の認識に依存する。しかし、その構造にはいくつかの“型”があり、それらを下敷にして課題を眺めることができる。ある種の句構造を見いだすとき、文法記述にならって課題を提示することができる。しゃにむに抽象データ型として提示することも勉強になる。前件と後件を述語論理の言葉で書くこともよい。これらの多くがいずれも実現への手掛かりを与えてくれることも特筆できる。

なんといっても一般的な記述方法は数学理論の展開を見習うことである。わかりやすくまた厳密な課題提示にとって、数学理論が持つ抽象性、論理性、普遍性、形式性といった特性はきわめて好ましい。課題を枠組みや書式にとらわれることなく、自由かつ数学的に示すことがよい仕様記述につながることになる。このような精神と態度で、課題に応じてまた読み手を考えて書き方を工夫することが仕様の記述にとって最も大切である。このあたりまえのつまらない事実が結論である。

- 参考文献 [1] B. Liskov, S. Zilles, “An Introduction to Formal Specifications of Data Abstractions”, Current Trends in Programming Methodology Vol. 1, Prentice-Hall, 1977.
 [2] 全日本ボウリング協会, ボウリング規定集, (財)全日本ボウリング協会, 1981.
 [3] J. J. Van Griethuysen, (ed), CONCEPTS and TERMINOLOGY for the CONCEPTUAL SCHEMA and the INFORMATION BASE, ISO/TC97/SC5-N695, 1982.
 [4] F. Pereira, D. Warren, “Definite Clause Grammars for Language Analysis”, *Artif. Intell.*, Vol. 13, 1980, pp. 231~278.
 [5] D. E. Knuth, “Examples of Formal Semantics”, Lecture Notes in Mathematics No. 188,

- Springer-Verlag, 1971, pp. 212~235.
- [6] 中島玲二, 数理情報学入門, 朝倉書店, 1982.
 - [7] D. Bjorner, C. B. Jones, (eds), "The Vienna development Method", Lecture Notes on Computer Science No. 61, Springer-Verlag. 1978.
 - [8] B. A. Silverberg, et al., The HDM Handbook 3 vols, SRI International, 1979.
 - [9] K. Futatsugi, K. Okada, "A Hierarchical Structuring Method for Functional Software Systems", 6th International Conference on Software Engineering, IEEE, 1982, pp. 393~402.
 - [10] E. Y. Shapiro, "A Subset of Concurrent Prolog and Its Interpreter", ICOT TR-003, 1983.
 - [11] 日本ユニバック, RSDM プログラム仕様記述技法, 日本ユニバック(株), 1983.
 - [12] J. R. Abrial, S. A. Schuman, B. Meyer, "Specification language (Draft)", Proceedings of Summer School on Program Construction, Cambridge University Press, 1979.
 - [13] J. A. Goguen, et al., "An Initial Algebra Approach to the Specification, Correctness, and Implementation of Abstract Data Types", Current Trends in Programming Methodology Vol. 4, Prentice-Hall, 1978.

執筆者紹介 加藤 潤三 (Junzo Kato)

昭和46年岡山大学理学部物理学科卒業。同年4月日本ユニバック(株)入社。現在、システム本部生産技術一部所属。



編集者注 本稿は情報処理学会・第40回ソフトウェア工学研究会(1985年2月7・8日)で発表した「仕様記述法の味見」(加藤潤三, 染谷 誠, 板倉 教, 田端福雄, 森沢好臣, 山崎利治)に若干の修正を加えたものである。

報告 データ構造に基づくテスト・ケース設計法

Test Case Design Based on Data Structures

峰 尾 欽 二

要 約 テスト・データを選択する適切な基準がないことが、プログラムのテストを困難な作業にしている一つの原因である。

本稿では、プログラマの主観や経験に左右されない系統的なテスト・データ選択法について論じる。この方法のアイデアは、入出力データの構造を状態推移図に変換することによってテスト・データの選択基準を選定することに基づいている。この方法は、データ構造からプログラム構造を導く、JSP (Jackson Structured Programming) のようなプログラミング方法論にとくに効果的である。

Abstract: One of the most difficult problems in program testing is that we cannot find out an adequate criteria to select the reliable test data.

This paper presents a systematic method of selecting test data free from programmers' experience. The idea of this method is based of fixing the criteria to select the test data by transforming the structures of input and output data into the state transition diagram. This method is effective for the programming method like JSP by which we can get program structure from the data structures.

1. はじめに

プログラムの生産現場において、いかにテスト・データを準備するかということは常にプログラマを悩ませる問題である。プログラム・テストのためのデータ例の選択は確かにむずかしい問題である。計算対象が複雑な構造をもっているとその選択は一層困難になる。事務計算の対象であるファイルに対してさえこの選択はやさしくない。本稿は、事務計算プログラムのテスト・データ選択についての一方法を提示する。この方法は課題から直接にデータ例を選択するもので、コードに基づいたものではない。したがって、このデータ例によってコード・カバレッジを検査し、そこで良い結果が得られればコード自身も適切に記述されているとみなしても良いといえるようなものである。

事務計算の一つの典型として次のようなプログラムを与えよう。

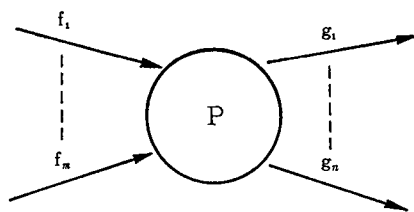


図1 プログラムとデータ
Fig. 1 Program and data

プログラムPは、レコード型 R_i のファイル $f_i (i=1, 2, \dots, m)$ を入力とし、レコード型 R_j のファイル $g_j (j=1, 2, \dots, n)$ を出力とする。ここで入力データ領域は、

$$I = \{f_1 + \dots + f_m \mid f_i: \text{レコード型 } R_i \text{ のファイル}\}$$

あるいは、

$$I = \{r_1^* + \dots + r_m^* | r_i: \text{レコード型 } R_i \text{ のレコード}\}$$

と考えるよい。(ここで+は、直和を表す。)

同様に、出力データ領域Oは、

$$O = \{r_1'^* + \dots + r_n'^* | r_j': \text{レコード型 } R_j' \text{ のレコード}\}$$

と考える。このままでは入力領域をテスト・データを作成するための副領域に分割することは困難であるから、データ例選択のためにIを適切な基準で類別したい。そのためにはIをアルファベット $\{r_1, \dots, r_m\}$ に関する句構造言語、とくに正規言語とみなすと都合がよい。このとき、プログラムPはその構文的側面を抽象化してIからOへの変換系となる。この点に着目してデータ例をつくらうというわけである。以下にその例を示す。

2. 例1: テキスト編集問題

やや現実的でない点もあるが、入力テキストを出力テキストに変換する小さなプログラムを考える。

入力テキストの文章は、単語に続くピリオドと空白文字で識別される。出力テキストでは、文章中の連続した空白文字は1個の空白文字に変換される。ピリオドに続く連続した空白文字は改行文字に変換される。入力テキストの最初に空白があってもそれは出力されないものとする。

この課題の入力領域に正規文法を見い出して正規言語 IL として修正 BNF で書くと以下のようになる。

$$\text{IL} \left\{ \begin{array}{l}
 \langle \text{input-text} \rangle ::= \langle \text{space-seq} \rangle \langle \text{sentence} \rangle^* \text{end-of-text} \\
 \langle \text{sentence} \rangle ::= \langle \text{word} \rangle \langle \text{word-set} \rangle^* \text{period} \langle \text{space-seq} \rangle \\
 \langle \text{word-set} \rangle ::= \text{space} \langle \text{space-seq} \rangle \langle \text{word} \rangle \\
 \langle \text{word} \rangle ::= \text{not-space-character not-space-character}^* \\
 \langle \text{space-seq} \rangle ::= \text{space}^*
 \end{array} \right.$$

同様に、出力領域を正規言語 OL として修正 BNF で書く。

$$\text{OL} \left\{ \begin{array}{l}
 \langle \text{output-text} \rangle ::= \langle \text{sentence} \rangle^* \text{end-of-text} \\
 \langle \text{sentence} \rangle ::= \langle \text{word} \rangle \langle \text{word-set} \rangle^* \text{period new-line-char} \\
 \langle \text{word-set} \rangle ::= \text{space} \langle \text{word} \rangle \\
 \langle \text{word} \rangle ::= \text{not-space-character not-space-character}^*
 \end{array} \right.$$

この課題は、入力領域 input-text から出力領域 output-text への変換系として定義される。出力の終端記号を生成する入力側の記号列を対応させると以下のようになる。

$$\begin{array}{l}
 \text{period new-line-char} \longleftrightarrow \text{period space}^* \\
 \text{space} \longleftrightarrow \text{space space}^* \\
 \text{not-space-character} \longleftrightarrow \text{not-space-character} \\
 \text{end-of-text} \longleftrightarrow \text{end-of-text}
 \end{array}$$

ここで変換系として定義された課題を、入力データを出力データに変換する状態推移図で表す。図2において、εは空データを、スラッシュ(/)の左側は入力終端記号を、右側は出力終端記号を表す。入力側に対応した出力が存在しない場合はスラッシュと出力終端記号を省略し、同様に ε/ε は ε に省略している。

(正規言語として表現された入力領域と出力領域から入出力状態推移図に変換するプロセ

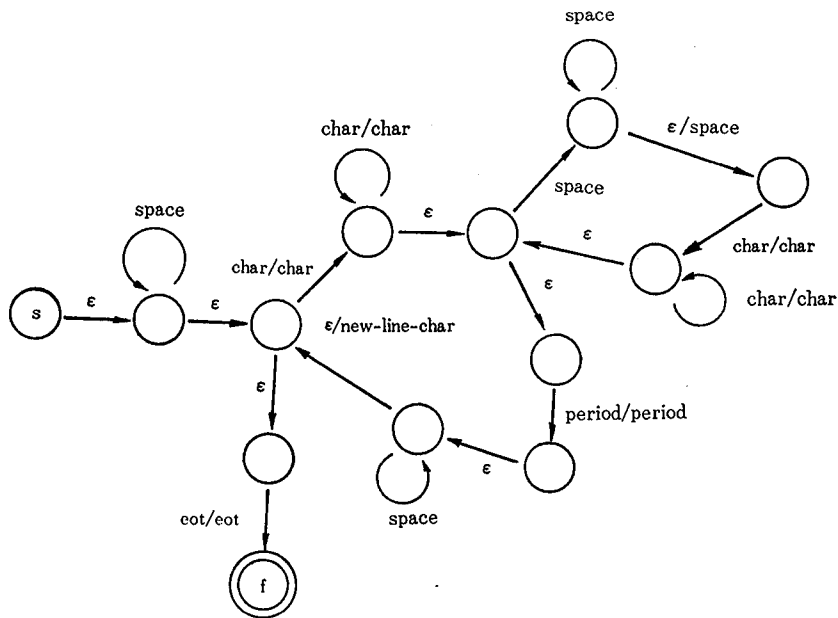


図 2 入出力状態推移図
Fig. 2 Input and output state transition diagram

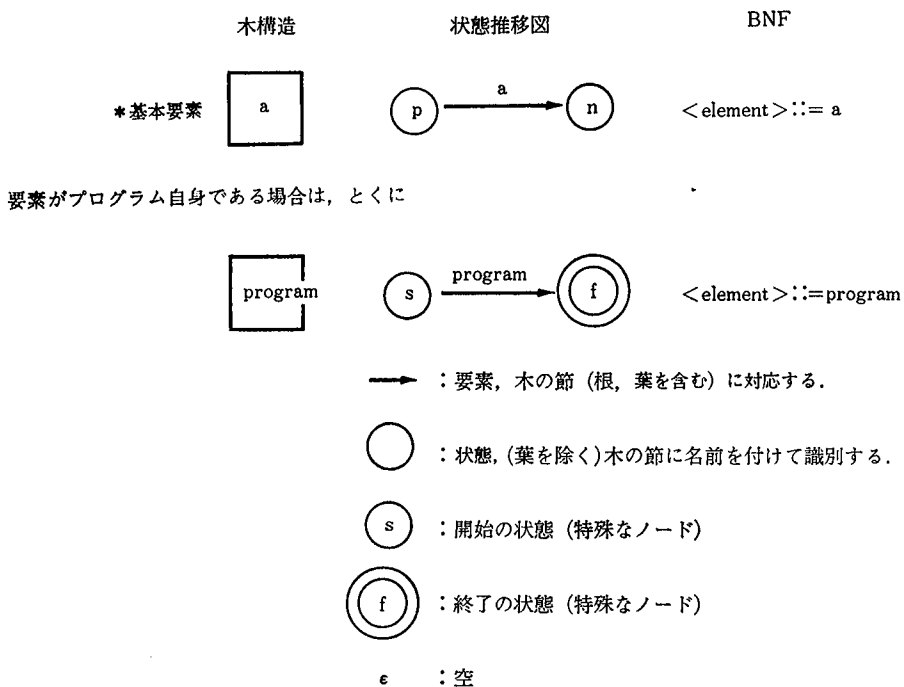


図 3 木構造と状態推移図-1
Fig. 3 Tree structures and state transition diagrams-1

スは次章で詳しく触れる.)

課題を適切に反映したプログラムは入出力状態推移図が示す入出力領域の関係を満たしている。コード化されたプログラムが課題を満たしていることを検査するために、入出力状態推移図から適切なデータ例を抽出してテストすることは効果的である。

テスト・ケースを開始ノード s から終了ノード f に至るパスとして選定する。このとき、少なくともすべてのアークは網羅されなければならない。

開始ノード s から終了ノード f に至る、最短距離ですべてのアークを網羅するパスとして、テスト・ケース

space char char space space char char period space eot

が選定される。このとき出力されるデータは、

char char space char char period new-line-char eot

である。以上が基本的なアイデアである。

3. テスト・ケース選定法

以上みてきたアイデアを以下で詳細に述べる。

選定法は三つのステップから成り立っている。すなわち、

1) データ構造の記述……最初に、JSP (Jackson Structured Programming) の流儀にし

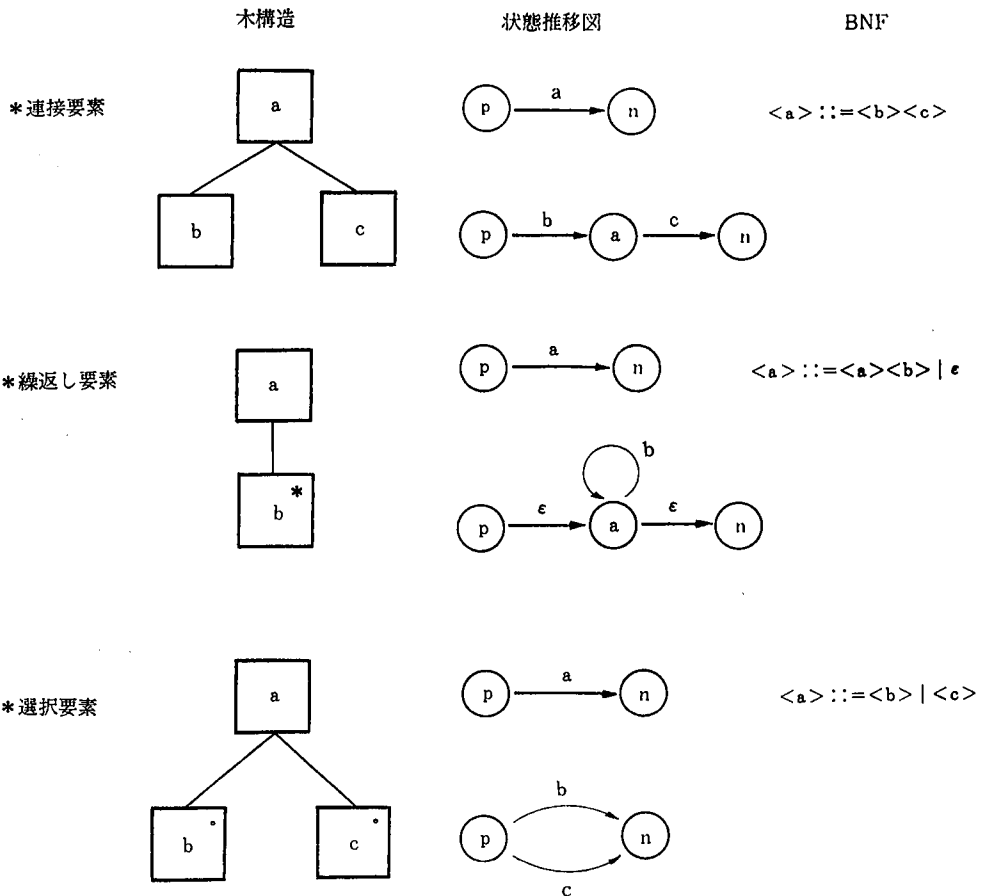


図 4 木構造と状態推移図-2

Fig. 4 Tree structures and state transition diagrams-2

たがって、入出力データの構造を木構造図で描く。入力ファイルが複数本ある場合には1本の論理の入力ファイルに統合する。(それは、複数本のファイルを1本に組み合わせられたファイルとみなすことができる。)

2) データ構造図から状態推移図への変換……まず、(論理的) 入力ファイルと出力ファイルたちから JSP 流のプログラム構造図をつくる。つぎに、そのプログラム構造図を入出力状態推移図に変換する。

3) テスト・ケースの選択……入出力状態推移図からテスト・ケースを選択する。

ここではデータ/プログラムの構造の表記は、JSP の木構造図のそれに従う。木構造の要素と状態推移図との対応を以下に示す。補足的に BNF の表記も添書きする。

図3は、プログラムの要素 a によって、状態 p が状態 n に変化することを示している。

これら表記法を用いれば、いかなる JSP 流の木構造図も状態推移図に変換することができる。一例を図5に示す。

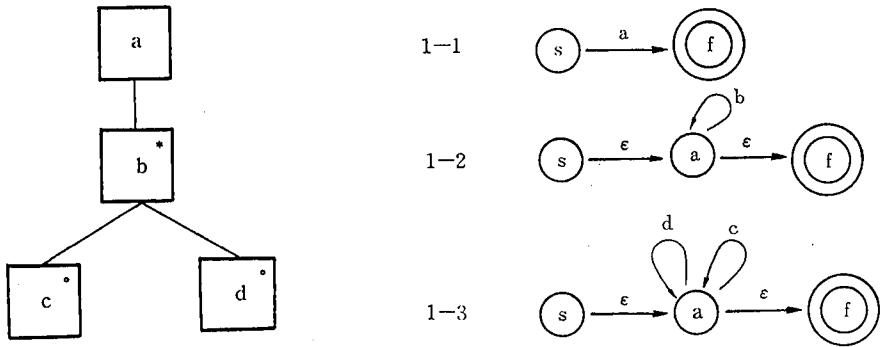


図5 状態推移図の段階的作成
Fig. 5 Stepwise implementation of state transition diagram

4. 例2：売上表作成問題

当期の売上表を作成する。入力ファイルは製品コードで整列された売上レコードから成っている。この売上ファイルは、当期の売上レコードと旧期の売上レコードを含んでいる。当期の売上レコードに関する売上表を作成する。最初に製品ごとの売上明細を印書し、続いて製品合計を印書する。最後に当期の総合計を印書する。

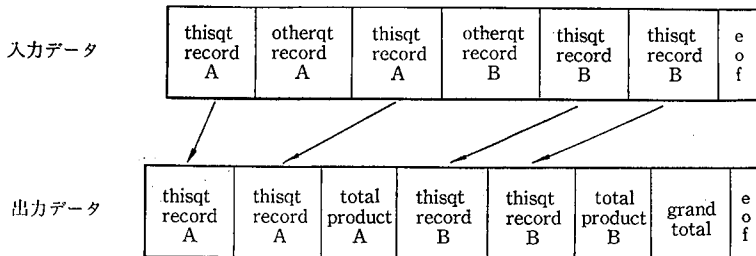


図6 データ例
Fig. 6 Sample data

最初に、入出力データの構造図をつくる(図7)。

つぎに、入力の木を出力の木に変換するプログラムの木をつくる(図8)。

図8において、四角で示されているプログラムの木の節の中の名称 a/b は、入力 a に対

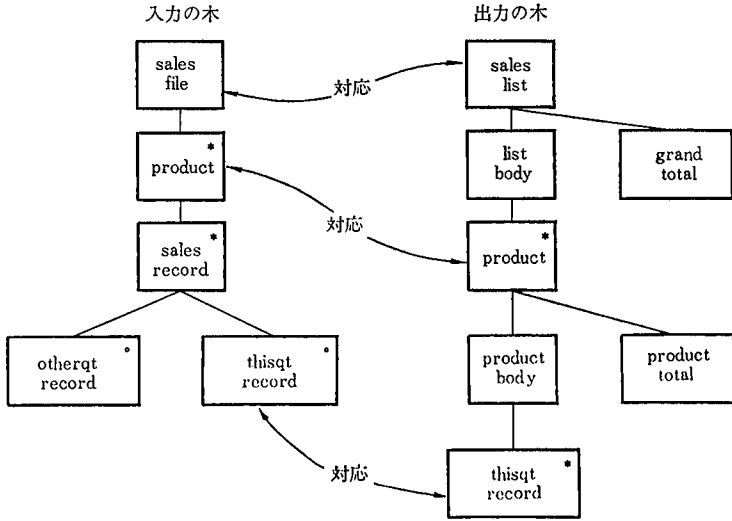


図 7 データ構造図
Fig. 7 Data structures

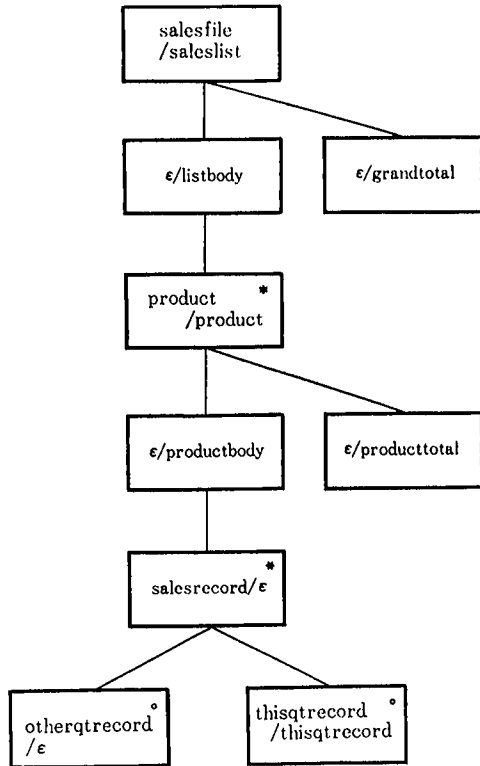


図 8 プログラム構造図
Fig. 8 Program structure

してbが出力するプログラムのプロセスを表している。otherqtrecord/εは、otherqtrecordが入力した時点では何も出力しないことを、ε/grandtotalは、それ以前のプロセスが完了した後に grandtotal が出力することを示している。

さらに、図8のプログラムの木を順次入出力状態推移図に変換する(図9)。

図9を簡略化して結局, 図10が得られる.

thisqtrec/thisqtrec は, thisqtrecが入力したときに thisqtrecが出力することを示している. ϵ , otherqtrec は, 各々 ϵ/ϵ , otherqtrec/ ϵ の省略形である.

入出力状態推移図の開始ノード s から終了ノード f までたどることによって, 入力データと対応する出力データの組合せを組織的に準備することができる.

この例題でテスト・ケースを選択してみよう. 図10において, 各アークに自然数が付いているが, この数値を列挙することによって入出力データの組合せの軌跡をたどることができる.

すべてのアークを網羅すべきであるというパス・カバレッジの観点からすれば, 図10か

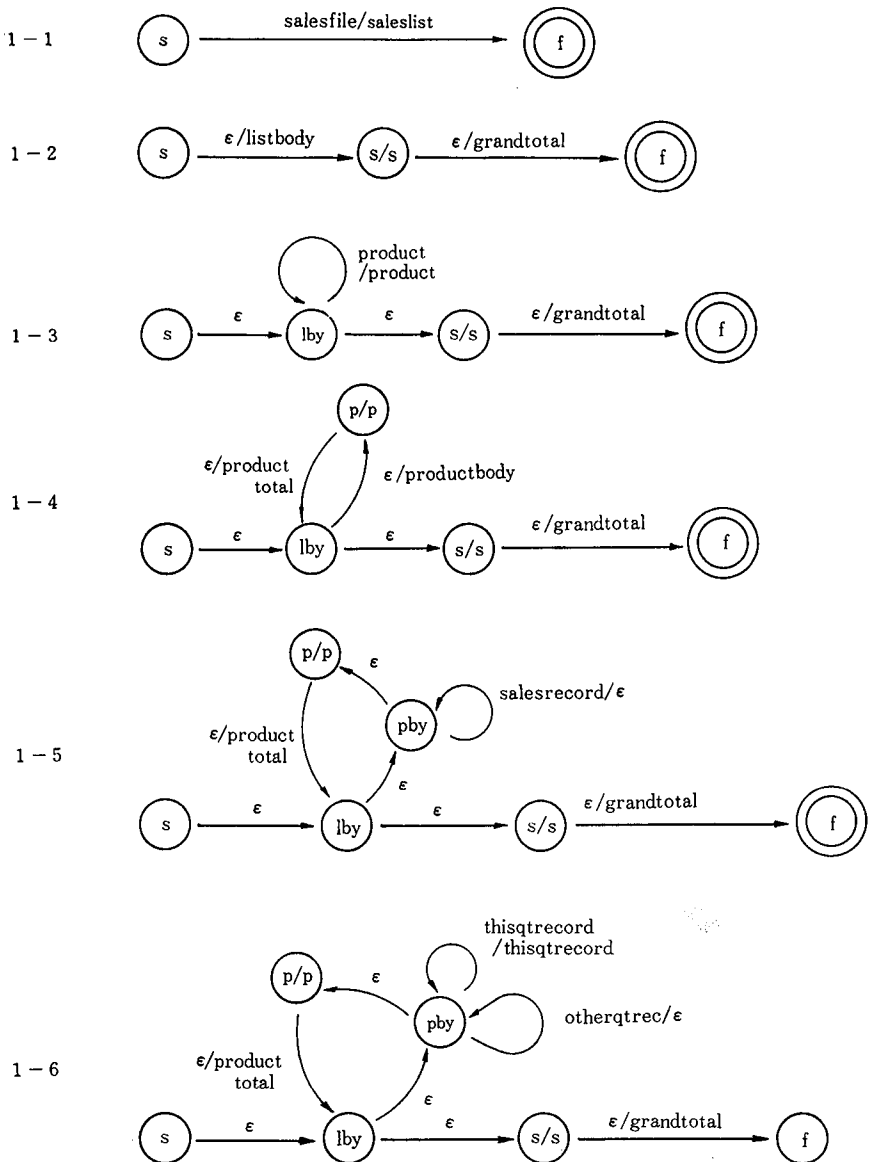


図9 入出力状態推移図の作成

Fig. 9 Implementation of input and output state transition diagrams

反映したものであるという事実に着目して、プログラムの課題を入力領域を出力領域に変換する変換系と解釈する。さらに、入力領域、出力領域を正規言語とみなし、正規文法を満たす文章としてテスト・ケースを選択する。このとき正規文法から変換された状態推移図のアークをすべて網羅するパス・カバレッジを基本に、さらに若干のプログラムの意味を考えて、テスト・ケースを選択する。選択されたテスト・ケースはプログラムが変換系を実現しているかどうか、すなわち入力終端記号（入力の木の葉）を出力終端記号（出力の木の葉）に変換するプログラムの木の葉がプログラム中に適切に実現されているかどうかをテストするのに役立つ。つまり、作成されたテスト・データがプログラム内部のパスを網羅していれば、そのプログラムは良いプログラムであると見なすことができる。したがって、テスト実行時にプログラムの木の葉に関するパス・カバレッジ測定ツールを用いれば一層効果的となるであろう。

最後に、適切な助言をいただいた山崎利治、前田英次郎、染谷誠各氏に感謝する。

- 参考文献 [1] M. A. ジャクソン著、鳥居宏次訳、構造的プログラム設計の原理、日本コンピュータ協会、1980. (M. A. Jackson, Principles of Program Design, Academic Press, 1975.)
 [2] 峰尾欽二、“プログラミング方法論（ジャクソン法）”，情報処理，Vol. 23, No. 11, 1982, pp. 1063~1074.

執筆者紹介 峰尾欽二 (Kenji Mineo)

昭和38年東京工業大学経営工学科卒業，41年日本ユニバック(株)入社，教育部所属プログラミング教育担当。



報告 CAD エキスパート・システム

Computer-Aided Design Expert Systems

G. D. Nelson, R. A. Higgins

概要 本稿では CAD に関連したエキスパート・システムに関する、いくつかの試行的なあるいは実用開発的な作業について述べる。それらは、1) 半導体チップの配線設計システム、2) チップ配置設計のためのアルゴリズム/エキスパート混合システム (ASHLAR) および、3) 全社 CAD 知識ベースの開発である。知識の獲得法や探索(推論)、制御法に関する経験を積んで、エキスパート・システムを持ち込むことによってアルゴリズム主体の配置配線設計の効率を改良したり、Sperry 社内に氾濫する CAD ツールの理解と効果的利用の促進をはかるための努力がなされた。全体にわたる目的は、Sperry 社のディフェンス・プロダクト・グループ (DPG) がエキスパート・システム技術を利用して、より複雑な VLSI チップや他の DPG 製品の設計能力を整えることである。そのためには、知識ベースの確立が必須であり、とりわけ、さまざまな専門家からの情報の獲得手法の開発が急務である。配線設計のための知識ベースを作り上げる過程で得られた経験は、他のプロジェクトにおける知識獲得作業の助けとなろう。KEE のような知識工学ツールは、知識ベースの開発の有用な手段として期待できるし、Reston と Eagan にある CTC の組織との共同作業も可能である。

Abstract This paper describes educational and developmental CAD-related activities with several expert systems, including: 1) a semiconductor chip wire router, 2) a hybrid algorithmic-expert chip layout system (ASHLAR), and 3) development of a corporate CAD knowledge base. These efforts were undertaken to develop experience with knowledge acquisition, search and control methods; to improve the performance of an algorithmic place-and-route system through integration with an expert system; and to promote the understanding and effective use of Sperry's diverse CAD tools. The overall objective is to provide Sperry Defense Products Group (DPG) with a capability for applying expert system techniques to the design of more complex VLSI chips and other DPG products. Establishment of knowledge bases is essential to the technical objectives in that techniques must be developed for acquiring information from various experts. Experience gained in creating a knowledge base for the wire router should aid in this acquisition task for other projects. Knowledge engineering tools, such as (KEE), are expected to assist in development of knowledge bases, and enable cooperative efforts between CTC organizations at Reston and Eagan.

1. はじめに

エキスパート・システムを用いた CAD 関連研究開発作業は、知識の獲得、探索、制御方法などに関する基礎研究作業；エキスパート・システムを利用した配置配線設計システムのアルゴリズムの効率改善作業；Sperry 社内に多数ある CAD ツールの普及および効果的利用の促進、などをそれぞれの目標として現在まで分野別に進められてきた。それらの作業の全体的目標は、Sperry Defense Products Group (DPG) に対し、より集積度の高い VLSI チップなど DPG 製品の設計にエキスパート・システムを適用した機能を提供することである。

1.1 エキスパート・システムの CAD への適用

設計部門にとって必要なのは、エキスパート・システム開発に関する最新技術に精通し、また人工知能的な手法、とりわけ CAD ツール開発へのエキスパート・システム技術の応用に関して、経験を積むことである。ここで重要な課題となるのは、設計部門が人工

知能やエキスパート・システムを用いた設計能力を持ち、将来の軍需用／民生用製品の開発にこの新技术を応用してゆくために必要な専門知識を持つのにどんな技法が有用であるかを見定めることである。

VLSI チップ設計者は、自社内の CAD ツール情報を効率よく入手してそれを活用し、さらに高性能軍需用電子システムを設計するために、さらに高度な CAD システムを必要とする。複雑なチップについて、要求水準に合う割付配置あるいは、より効率のよいコンパクトな割付配置を自動的に行わせるために、専門家の知識を最近のエキスパート・システムに登録して利用するという考え方が提案されている。

自動 CAD システムに関して当面の課題は、高度のエキスパート・システム技術と従来の CAD ツールを統合するための方法の確定である。配置／配線用の通常の CAD アルゴリズムが人間設計者と同じ程度の効率を達成できないことが明らかであるが、CAD ツールはエキスパート・システムとアルゴリズム主体の CAD システムを結合して熟練技術者の知識を援用することによって改善可能であろう。図 1 は、手操作で設計したチップの一部分拡大写真であるが、可能な限りコンパクトなセル密度が達成されていることがわかる。つぎに、図 2 に、Sperry 社の SCROL システムによるセミ・カスタム・チップの自動配置配線設計結果を示す。これには配置配線設計用 CAD ツールに共通する弱点、すなわち未使用チップ域の存在や非効率的なセル配置が散見できる。

また表 1 には、標準セル方式のチップについての手動配置配線設計と自動配置配線設計の、セル密度（セル全体領域／チップ領域）の比較表を示す。ここでは、流通ソフトウェア (CAL-MP) を含む、数種類の配置配線設計システムを比較している。自動設計で得られたセル密度は、手動設計の場合よりも 35 パーセントないし 50 パーセント低いことが、これら互いにタイプの異なる数種のセミカスタム・チップについてのデータからわかる。また ASHLAR, CAL-MP, SCROL がセル密度に関して大差ないことが注目される。第 3 章

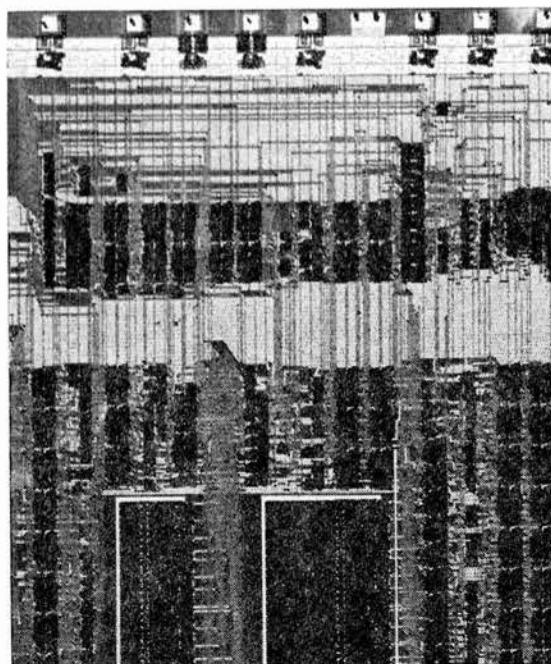


図 1 手動配置配線設計されたチップの顕微鏡写真

Fig. 1 Die photograph of manually-laid-out chip

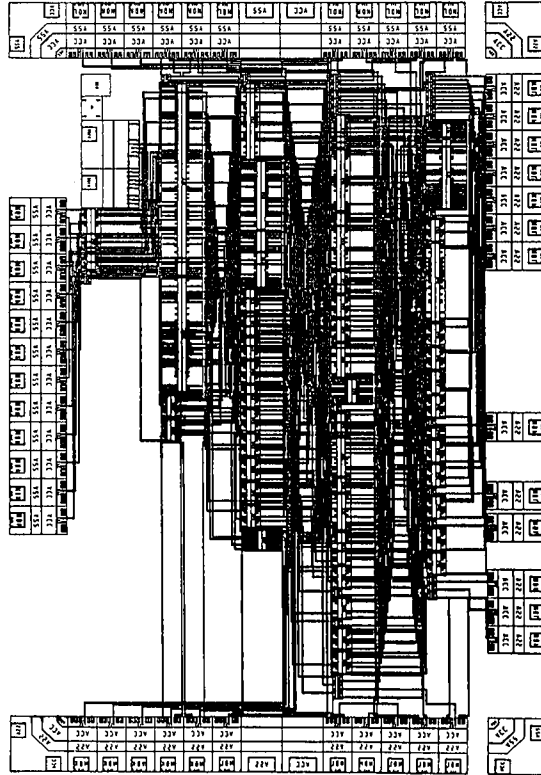


図 2 自動配置配線設計による標準セル型半導体チップ

Fig. 2 Automated layout-standard cell semiconductor chip

表 1 手動配置配線設計と自動配置配線設計のセル密度の比較表

Table 1 Comparison of manual and automated chip layout cell density * 設計者の手作業を一部含む.

チップ名	ASHLAR	CAL-MP	SCROL	手動
SUPER SEQUENCER	0.20*	0.22	0.26	0.40
GEM	0.28*	—	—	0.42
PROTEUS	—	0.34	—	0.59
DAMT	—	—	0.27	—

表 2 手動と自動の配置配線設計のチップ面積比の比較表

Table 2 Comparison of manual and automated layout-chip area ratio * 設計者の手作業を一部含む.

チップ名	ASHLAR	CAL-MP	SCROL
SUPER SEQUENCER	2.0*	1.8	1.5
GEM	1.4*	—	—
PROTEUS	—	1.4	—

で述べるように、非公開の研究用ツールである ASHLAR は、他のチップ・レイアウト・システムに対し、比較的効率が良いので、ハイブリッド・エキスパート・システムの開発ベースとして選択された。

また表2では、チップ・サイズに関して、数種の異なるセミ・カスタム・チップについて自動設計と手動設計の面積比で考察している。自動設計でのチップ面積は、手動設計でのチップの場合よりも40パーセントから100パーセント広い。場合によっては、これらの結果は、自動設計によるチップの製造を否定しかねない。というのも自動配置配線手法で製造したら、生産歩止まりが大幅に減少し、その結果として生産コストが高騰するからである。

現行CADシステムについても一つの重要な問題は、これらのシステムではデータベースやインタフェースの互換性がなく、可用性、運用データ、機能などに関する情報不足のために、どのデザイン・センタでもすべてのCADツールをアクセスできたり手頃に利用できるという訳にはいかないことである。さまざまな設計機関で利用されているツールの中からあるプロジェクト用に最善のCADツールを選択する場合には、エキスパート・システム技術を利用した学習システムがかなり役立ちそうである。この技術が開発されれば全社CAD知識ベースを用いて、エキスパート・ベースで設計プロジェクト用CADツールを選択することが実現できる。

1.2 専門技術者からの知識の獲得

さて熟練チップ設計者から知識を獲得する場合にはかなりの努力を要すること、また体系的な機械化された知識獲得法がないことが、これまでエキスパート・システム開発を阻害してきた主因である。だが専門技術者の知識をLISP^[1]関数やKEE^[2]などで取り扱い可能な形式に変換する作業より、知識獲得手順の展望が開けてきた。CAD用エキスパート・システム構築に十分な設計者知識を獲得することは困難な作業だが、とりあえず専門技術者から入手可能な断片的な情報だけを用いて、エキスパート・システム設計の第一段階の開発の開始が決定された。その開発手順は、製造設備に伴う諸問題を現場で発生させて確認するための「パイロット・ライン」の稼動に類似していた。

最初の聞き取り作業中に、画像強調処理モジュール(GEM)チップを開発してきた設計の専門技術者は、自分達がどんなルールに従っているかに気づいていないにもかかわらず、GEM配置配線設計の作成ルールの定式化を試みた。何回かの討論の場が持たれ、GEMに利用された知識や手法が類別され、そこから配線経路生成エキスパート・システムの開発を始めるのに十分な、40のルールの集合が引き出された。

1.3 エキスパート・システム開発に必須な学習過程

現在、エキスパートの規則を活用する最善の方法は、それをプロダクションの形に変換し、LISP関数を生成し^[1]配線設計の結果を見るためにグラフィック画像を生成する機能を用意し、それらのルールを用いて得られた結果を専門技術者に示すことである。エキスパート概念がまだ生まれて間もないものであり、エキスパート・システム開発は反復改良が必要な性格のものなので、専門技術者もシステム設計者も共に学習過程が必要である。当初は、プロダクション・ルールの数が限られているため、エキスパート・システムはそれほどエキスパート的に稼動することを期待できない。しかし、その運用上表れた欠点を調査し、専門技術者は、そのエキスパート・システムについてより理解を深め、最初は明らかでなかった補足的ルールを認識することができるようになる。

2. エキスパート・システム開発ツールの利用

本計画が達成されれば、Sperry Defense Products Group (DPG) は、エキスパート・システム技術をきわめて複雑な電子システムやその他の高度な製品の設計に適用できる能力を

獲得できる。エキスパート・システム方法論の採用により、CAD ツールの中で人間の知的機能の利用が可能となり、またそれによりさらに複雑な VLSI 設計の実現可能性が高められる。

フレーム型モデル、黒板型モデル、FLAVOR モデルなどを調査することにより、KEE (Knowledge Engineering Environment)^[2] を効果的に利用するための基本的な土台が得られる。上記三つの異なった性質を有するエキスパート・システムを Symbolics または LMI-LISP マシン上に用意された KEE を利用して開発してみることによって、各分野での実現可能性が評価でき、緊急かつ重要な応用のためどの方向にプロジェクトの努力を向けるべきかがわかるであろう。これを全社 CAD 知識ベース・エキスパート・システムへ高度に発展させるには、まず知識ベースの確立から始めなければならない。続いてエキスパート・システム用のルールを作り上げて、設計者が CAD ツールに関する知識を引き出せるようにする必要がある。ツール選択用の情報には、ホストコンピュータの環境、バイポーラ、MOS などの半導体テクノロジー、データベース構造、インタフェース要件、ツールの説明、機能、利用法、可用性、ツールの保管場所、などが含まれる。

3. アプリケーション

本プロジェクトは、エキスパート・システム技術を開拓し、以下の三つの領域における CAD 用 AI 技術の利用を推進する。

- 1) 入門用アプリケーションとしての半導体チップ配線経路生成機構
- 2) より実用 CAD ツールを作り出すことを目標とした、エキスパート-ASHLAR 回路配置配線設計システム^[3]
- 3) 全社 CAD 知識ベースの確立（これによって、将来の設計プロジェクト用 CAD ツールをエキスパート・ベースで選択することが可能となる。）

それぞれの研究についてプロジェクトの管理と完了までの作業に関する適切なドキュメントが公表される。図 3 には、この CAD エキスパート・システム・プロジェクトによって行われる作業を要約し、使用するコンピュータやソフトウェアを示している。

3.1 エキスパート配線経路生成機構

まず最初に、配置配線設計用エキスパート・システムが、KEE を利用したエキスパート知識の獲得、フォーマット化、およびその利用を実証するための応用対象として採用さ

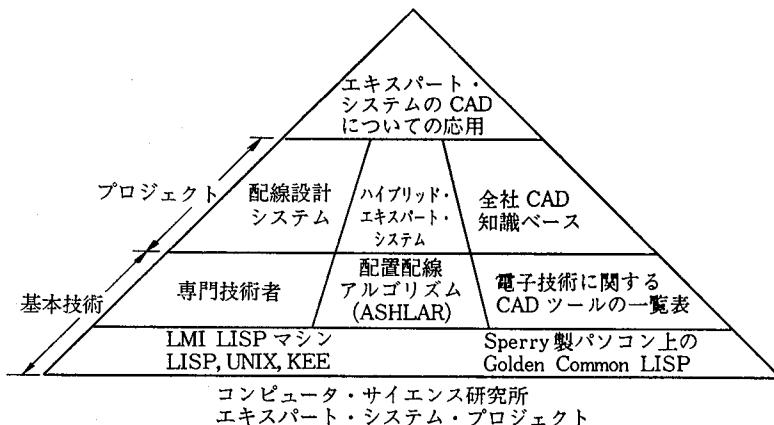


図 3 CAD エキスパート・システム・プロジェクト全体図

Fig. 3 CAD expert systems project overview

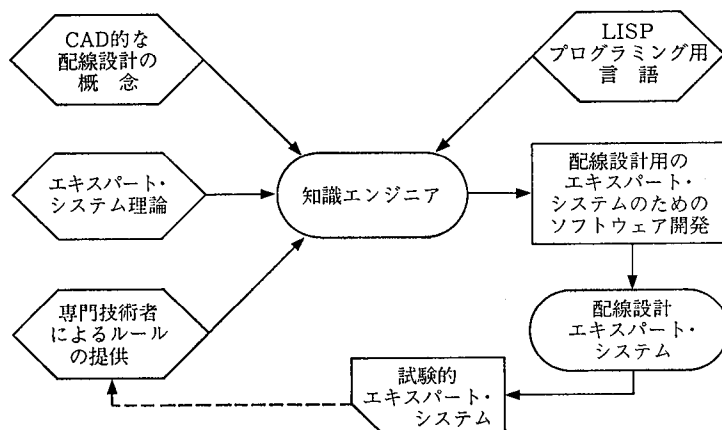


図 4 配線設計エキスパート・システムの開発

Fig. 4 Development of wire-router expert system

れる。配線経路生成エキスパート・システム開発用の基本的ハードウェアおよびソフトウェア構成要素は、図 4 に示される。開発は反復的に初期システムおよび、それに続く改訂版を評価し、次の改良版のための修正データを得ながら進められる。あらかじめ専門技術者から獲得された配線経路生成のためのルールは、定式化され KEE によってコンパイルされる。また、グラフィック表示装置が、エキスパート・システムが選択した配線経路を示すために導入される。こうしてできた配線経路生成機構の試験結果に基づいて、ルールの改良/追加を、専門技術者の助言を得ながら決定する。KEE にはさまざまなルートを選択する際に、あわせてその選択理由を表示する機能があるので、ルールの変更に必要な労力が軽減され、知識ベースの改善が楽になる。

知識ベースにルールを追加された改良エキスパート・システムは、試験チップの自動配線設計に適用して、同一チップに関して実施した手動設計による配線結果と比較することによって試験できる。この反復改良過程はまだ有意な改善が得られる可能性があり、コスト的に意味がある場合には継続される。

3.2 アルゴリズム的配置配線設計ツールとエキスパート・システムの統合

エキスパート・システムとアルゴリズム的配置配線設計ツールの結合システムの開発において、SCROL, ASHLAR^[3] などの、既存のアルゴリズム的方法によっては適切に処理できない配置配線設計時のさまざまな局面について、一連のエキスパート・システムが検討されてきた。まず手始めに、これらのシステムはエキスパート・システム開発支援用に設計されたプログラミング環境を利用して、単一目的のプロトタイプとして現行アルゴリズム的ツールと切り離して開発されるであろう。そして実験的段階を越えたところで、これらのプロトタイプ・システムは、アルゴリズム的ツールとさらに緊密に統合される。

汎用セル配置用エキスパート・システムの開発は、LMI-Lambda システム上で行われる。このシステムは、ZETALISP, PROLOG, FLAVORS 言語および KEE などのエキスパート・システム・サポートを装備した LISP マシンである。インタフェースは、設計情報交換のために提案された標準仕様の EDIF (Electronic Design Interchange Format) を用いて設定されるものとする。このインタフェースにより、UNIVAC シリーズ 1100 の ASHLAR による設計システムから LMI-Lambda システムへの試験チップ設計データの転送、およびこのエキスパート・システムから ASHLAR へ配置結果を戻して配線設計を行わせることが可能となる。

この開発では、二つのキーとなる技術的問題が着手される。その一つは、エキスパート・システム開発に固有な知識工学の課題すなわち、知識ベースの獲得、表現、組織化、修正などである。もう一つは、エキスパート・システムを ASHLAR にインタフェースするという問題である。前に概述したように当アプローチでは、人工知能的支援環境で開発を行うことにより、知識工学上の問題を最小限に抑えた。また配置問題のみを、配線問題と切り離して対象とすることによって、インタフェースのむずかしさを軽減した。本アプローチ以外に考察された代替案を以下に説明して、これらの問題点を浮きぼりにしてみよう。

UNIVAC シリーズ 1100 上で開発された ASHLER は PASCAL で 100,000 行のシステムであるから、1100 Pascal を用いてエキスパート・システムを開発すれば親和性が良いことが明らかであろう。そうすれば ASHLER との統合が比較的容易だが、その開発は専用 LISP マシンの検索プログラム機能や知識工学環境を利用できないため困難であろう。そこでまず、知識表現や推論機構を、UNIVAC シリーズ 1100 上で開発する案が提案された。つぎに ASHLER を MI-Lambda シリーズに移植する代案が考えられたが、これはインタフェースに関する問題が最も少なく、また望ましい知識工学環境が利用できるが、ASHLAR の移植の作業は UNIVAC システム 1100 上に知識工学環境を整える作業に規模の点で匹敵しよう。

さらに、配置設計業務、すなわちチップの縦横比率やセル配置時の接続ポート位置決めなどの制約条件の定式化など、比較的高水準の考察に基づく作業を選択的に移植するというもう一つの代案が考えられた。これらの作業は、階層的な ASHLER による配置設計にとっては重要だが、知識の獲得が困難である。なぜなら、ASHLER による配置設計の経験の蓄積が不十分なため、必要な専門知識が集約されていないからである。さらに、また必要なデータが ASHLER システムからは容易に取り出せないし、またエキスパート・システムによる解が ASHLER システムに簡単にインタフェースできないのである。

要約すれば、エキスパート/アルゴリズム混合配置配線ツールを開発するという問題は、次の三つの段階に分解できる。まず最初に、ASHLER とエキスパート・システム開発で利用する LMI-Lambda システムの間で、適当なインタフェースを確立する。第 2 に、汎用セル配置設計に関する知識ベースの獲得、表現、改良などの知識工学的作業が行われる。最後に、でき上がったエキスパート・システムを、ASHLER による自動配置結果や ASHLER ユーザによる会話型の設計結果と比較することにより、評価する。

3.3 CAD 知識ベース・エキスパート・システム

全社 CAD 知識ベース・エキスパート・システムの開発というより先進的なプロジェクトでは、まず CAD 知識ベースを確立し、そして CAD ツールに関する情報のアクセスを可能とするルール集合を作成する。このエキスパート・システムの目的は、多くの異なる組織によって利用されるツールの中からプロジェクト用の最善の CAD ツールを選択する際に、設計者を補助することである。

全社 CAD 知識ベース・プロジェクトは、LMI-LISP マシンを用いた知識ベースを生成する際の KEE の適切な応用について述べたレポートから引き出されたものである。KEE を用いて生成された知識ベースの利用を可能とするためには、まずエキスパート・ルールの獲得が必要となり、また有限のルール集合によって KEE をうまく利用した全社 CAD ツール用のエキスパート・システムを確立しなければならない。

全社 CAD 知識ベース用のエキスパート・システム開発では、次のような三つのフェーズが考えられる

- 1) ツールに関して知識ベースにアクセスするための予備的なルール集合を用いた、CAD 設計プロセス構造の設定。
- 2) CAD システムの専門家にプロジェクトの特性を満たす最適な CAD ツールを選択する際のルール集合を指定させて、この設計者の知性を把握してエキスパート・システム用のプロダクション・ルールとする。
- 3) 提案要求 (RFP, request-for-proposal) に述べられている要件を入力として、RFP 要件を満たすように選択した CAD システム・ツールを説明する仕様を出力とするようなエキスパート・システムを開発する。

4. 実施と結果

知識の獲得および表現に関する技術の研究過程で、半導体チップ配線経路生成用の LISP コード、それに伴うグラフィック操作機能用の LISP コードなどが開発され、また知識の獲得、知識表現、および人工知能的な探索/制御手法に関する最新技術の全般について評価することができた。LISP, Symbolics 3670 LISP マシン, KEE などに関する専門知識も取得できた。Symbolics 3670 は、KEE システムと共に使用した場合、エキスパート・システムの開発促進の有用な手段となることがわかった。

4.1 技術評価

4.1.1 知識表現リサーチ

知識表現用のいくつかの技法が本プロジェクトの研究者によって調査され、プロダクション・ルール、フレーム型モデル、一階の述語論理、および意味解析ネットなどに関するセミナーが開かれた。これらの知識表現方法、とくにフレーム型モデルとプロダクション・ルールに精通することは、KEE を理解するうえで非常に役に立った。オプションを考慮した LISP 関数集合が、プロダクション・ルール・システムで用いるために開発された。これについては 4.3 節で述べる。

研究者達は、Sperry 社の Corporate Square 研究所に設置された Symbolic 3670 を用いて限られた範囲での使用ではあるが、LISP マシンや KEE の強力な機能に親しむことができた。この体験は、配線経路生成用エキスパート・システム・プロジェクトの将来方向に関して判断を下すのに役立った。

Symbolic マシンで得た経験に基づいて、LMI-Lambda 2X2/PLUS プロセッサが発注された。このプロセッサの提供する三つの独立したプログラミング環境あるいは実行環境、すなわち LISP, UNIX, PROLOG が供給する強力なプログラミング・ツールは、プログラマの生産性を著しく向上させた。Lambda システムに KEE を導入することによって、LISP 機能だけを利用した場合に比べて、迅速な配線経路生成機構やその他のエキスパート・システムの開発が可能となる。フレーム型モデル指向の構造を持つ KEE では、LISP で実現した配線経路生成機構エキスパート・システムを呼び出すことができる。専門家のルールを KEE が解決できる形式に変換する方法論の確立を優先させるため、これ以外の知識表現に関する考察は行われなかった。Sperry パーソナル・コンピュータに搭載されたもう一つの有益なツールである Golden Common LISP (GCLISP)^[4]により、研究の生産性は、大いに高められた。GCLISP は LISP 言語の習得を著しく促進した。大型汎用コンピュータに対するパーソナル・コンピュータの主な利点であるデバッグ用知識ベースの構築やの支援ツールと、マルチウィンドウを利用したコードの作成や参照機能が活用された。知エキスパート・システムの開発は、KEE を搭載した LMI-Lambda システムと合わせて

GCLISP の利用によっても加速されるであろう。

4.1.2 探索と制御

さまざまな探索手法が、エキスパート・システムの探索の負荷を軽減するために考案されている。

- 1) トップ・ダウンの逆方向探索による目標指向探索。
- 2) ボトム・アップの順方向探索によるデータ指向探索。
- 3) 1) および 2) の結合である双方向探索など。

多様な探索方法が可能である。探索方法は、知識ベース上におかれた既知の探索限定領域を利用した、いわゆる“メタ”知識の適用によってその探索効率を向上させることができる。

縦型探索および水平探索は、ヒューリスティックな手法（発見的方法）や評価関数、階層的探索計画、あるいは階層的生成/試験法によって制御できる。探索過程を制御し、演算負担を軽減するためには探索領域に特有なヒューリスティックな情報を確立せねばならない。そのためには、性質の良い評価関数、あるいはそれに類する目標点への最適経路を精度よく指示できる指標が必要となる。探索経路を制御するこれらのヒューリスティックな関数は、その形を決定するのに相当量の探索領域特有な知識を必要とするので、不明確であるかも知れない。

探索問題は複雑で、多分にその探索領域の特性に依存する。実用的なシステムでは、探索、マッチング、重複した項目の分離、作業段階などの制御が必要である。探索法は目標指向的であり、階層的な探索計画やヒューリスティックな手法が探索時間を軽減するために採用される。マッチング処理を行ったのち、コンフリクトを生じた情報の集合は単一で最善のコンフリクト解決ルールで処理可能なように簡約化される。

4.2 ルール構造の実例

配線経路生成機構用のエキスパート・ルールを図5に示す。エキスパートによって提供された、これらの規則の解釈およびそれらをプロダクション・ルールとして公式化するには、LISP 関数あるいは KEE を用いる必要がある。ルールの登録のために LISP 関数、知識表現構造、探索手法などを開発するのではなく、LISP マシンでの KEE を用いる利点は、KEE には、エキスパート配線経路生成機構システムを開発するために必要なルール、広範なデバッグ用ツール、グラフィック機能、適切なルール構造などがすでに用意されていることである。

配線経路生成機構システムのために KEE に規則を入れるために従う手順の例として、図6のルール15および10を考察する。ルール15の規定では、もし同じ垂直配線チャンネル上に始点および終点を持つネットを探索し、またネット間の行をブロックするセルが他になければ、配線は行を横切って直接ピンとピンをつなぐように実行されることになる。ルール10の規定では、セルが背中合わせに配置されていれば、ネットはセル領域を通過して、配線され、「背中合わせ配線」としてラベル表示される。この実例では、KEE はネットの始点および終点の座標、配線に使用すべき金属層の指定を受けて、ネットを定義する関数が用意される。KEE の推論機関は、垂直チャンネルを共有するこれらのネットを決定するべく探索を実行する。さらに、KEE には、セルを記述するすべてのパラメタを与える関数が与えられているものとする。KEE の探索ルーチンは、他のセルが二つのネット間の行をブロックするかどうかを決定する。このような障害となるセルが存在しなければ、KEE はその配線経路ピン間を直接セル行を横切る経路に選択する。障害となるセル

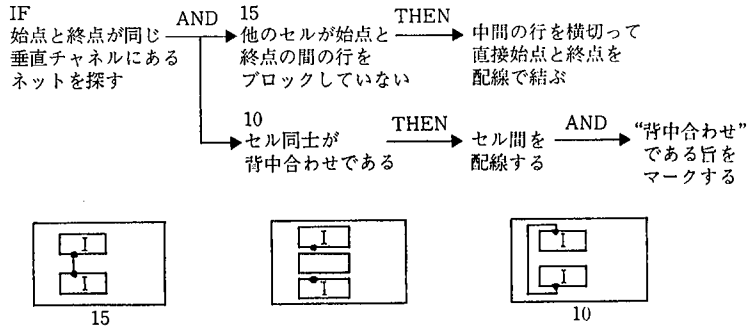


図 6 ルールの図解例

Fig. 6 Rule example with graphic detail

がネット間に存在すれば、配線経路決定用の別のルールを援用する。さらに、配線の障害となるセルが存在せずセルが背中合わせであれば、配線が完了し「背中合わせ配線」としてフラグが立てられる。セル記述項目の他の情報フィールドは、セル同士が背中合わせ配置であるか否かを確定するために参照されなければならない。

4.3 配線経路生成機構エキスパート・システム・シェル

記述された配線経路生成機構用のルールをすべて考察した結果、エキスパート・システム用にいくつかの関数を作成しなければならないことがわかった。これらの LISP 関数を探き出すための分析作業により、配線経路生成機構エキスパート・システムを開発する際に役立つかなりの知識が獲得された。そして、それは他のエキスパート・システム開発のための基礎知識となるものである。それらの関数は LISP の形式で表現され操作の実行に必要な関数名やパラメタ、および推論と処理の目的などと共に一覧表にされた。以下は、実験的な配線経路生成機構エキスパート・システム用に生成された 32 の関数群の一部分である。

1) 推論および処理操作機能

- MAX-DIST NODE DIRECTION……この関数は、ある NODE の位置から指定された方向へ移動できる最大距離を算定する。障害物との重なり合いは考慮しないが、基板の境界については考慮する。
- (ADD-CONSTRAINT NODE ADDED-CONSTRAINT)……NODE とは、格子点のこと。ADDED-CONSTRAINT とは、(方向修飾子付きの距離)の関数である。修飾子は、=, <, >である。>という修飾子は、指定以上の距離はすべて禁止領域であることを示す。また<という修飾子は、指定以下の距離はすべて禁止領域であることを示す。=という識別子は、ある距離が配線禁止であることを意味する。この関数は、指定の方向の制約条件範囲を更新または追加する。
- REPLACE-CONSTRAINT NODE CONSTRAINT……この関数は、関数 ADD-CONSTRAINT により呼び出される。この関数は、新しい制約条件が与えられた時に制約条件を構成する NODE のリストの物理的更新を行う。所定方向に関する旧制約条件は取り除かれ、リストに新しい制約条件が追加される。
- ALL-CONSTRAINT NODE……この述語はもし与えられた NODE がすべての方向について完全に制約されていれば、T を返す。そうでなければ NIL を返す。
- ROUTER BOARD-FILE CELL-LIBRARY-FILE……この関数は、配線経路探索過程を制御する。すべてがうまく配線できればTを返し、そうでなければ NIL を返す。

す。引数は外部ファイル名であり、“(<ファイル><エレメント>)”の形で与えられる必要がある。

これらの関数の使用法をさらによく理解するには、いくつかのパラメタに関する定義法についてさらに説明をしておく必要がある。東、西、南、北、上、下で表される、ボード上の方向は、北は、正の y 方向、東は正の x 方向、南は負の y 方向、西は負の x 方向である。上下は、金属配線層間の移動である。

判断リスト上の各格子点には、制約条件属性が与えられている。制約条件リストは、六つの各方向にたかだか一つのエントリを持つ。リストの各方向の制約条件は、(方向 (rl)... (yn)) という形式で表される。各 Ri は二つの数値で両端をおさえられた範囲である。範囲内のどの数も制約距離である（つまり、その方向にはその距離だけ移動することは禁じられる）。起こりうる最小下限束縛値は 0 であり、最大上限値はその格子点に関するその方向での最大距離である。

一例として、(NORTH (0 10) (50 100)) という制約条件を考察してみよう。距離 0 から 10、および 50 から 100 の移動が禁止される。だが距離 11 から 49、および 101 から最大距離までの移動は許される。

5. おわりに

知識獲得および表現技術を調査に始まって、半導体チップ配線経路生成機構用 LISP コード、およびそれに関連したグラフィック機能、初期化機能、推論機構、および処理関数などの開発に至った。配線経路生成機構エキスパート・システム・プロジェクトの全体的目的は、1) 知識獲得、2) 知識表現、3) AI 探索/制御手法、などに対する最新技術を評価することであった。その作業によって、LISP, Symbolics 3670 LISP マシン、および知識工学環境 (KEE) に関する専門知識が修得された。Symbolics 3670 プロセッサ上で稼動する KEE に精通することによって、配線経路生成機構エキスパート・システムの開発アプローチが改善された。Symbolics 3670 を、KEE システムと共に用いるとエキスパート・システム、とくにチップ配線経路生成機構の開発を促進する手段となりうる。

UNIVAC シリーズ 1100 での配線経路生成機構エキスパート・システム用の LISP ソフトウェアの初期開発は、Tektronix モデル 4010 のやや機能不足のグラフィック機能を利用して行われた。エキスパート・システムの次フェイズの拡張は、Sperry パーソナル・コンピュータの Golden Common LISP ソフトウェアを利用して行われた。

KEE は、エキスパート・システムの開発過程では非常に価値あることが実証された。KEE により、専門技術者から得られたすべての規則は KEE によって必要とされるフォーマットに変換される。KEE に対して提供された知識ベースは、セル・タイプ、セル・サイズ、チップ・サイズ；ピン位置、ピン名前、ピン間隔、フィードスルー；配線用チャンネル；その他 VLSI 設計で通常用いるパラメタ、などで構成される。

配線経路生成機構開発に際しての Symbolics 3670 LISP プロセッサおよび KEE システムの有用性は、全社 CAD 知識ベースなどの、他の潜在的 CAD アプリケーションも KEE の恩恵をこうむれることを示唆している。全社 CAD レポートに含まれる情報によって、KEE を用いて Symbolic 3670 で知識ベースを生成することができる。専門技術者からのルールの獲得のために、KEE を用いて生成された知識を利用することができる。全社 CAD ツール用にエキスパート・システムを KEE を用いて確立するための実現可能性は、エキスパート・ルール集合の大きさによって決まってくる。

- 参考文献 [1] P.H. Winston, B.K.P. Horn, *LISP, Second Edition*, Addison-Wesley Publishing Company, Reading MA, 1984, p. 434
- [2] Intellicorp, *KEE Software Development System User's Manual*, Intellicorp, 1985.
- [3] J.E. Hassett. "Automated Layout in ASHLAR: An Approach to The Problems of 'General Celi' Layout for VLSI", Conference Paper, 19th Design Automation Conference, IEEE-ACM, Las Vegas, NV, June 14-16, 1982, p. 8
- [4] *Goldem Common LISP Retereace Manual, Version 1.00*, Gold Hill Computers, Cambridge, MA, 1984, p. 234

執筆紹介 G.D. Nelson

Sperry 社に7年間在職。現在、エキスパート・システム研究プロジェクトのプロジェクト・エンジニアであり、信号処理プロジェクトに従事。最初、連邦鉄道機関主催の Rail-Flaw (ひび割れレール) プロジェクトに参加、その後、ジョセフソン接合グラジオメータおよびミリ波画像のアプリケーションを担当。

South Dakota 州立大学電気工学教授を9年間勤め、同大学のリモート・センシング研究所のパターン認識のスタッフ・サイエンティストであった。

Minnesota 州立大学で B.E.E., Iowa 州立大学で電気工学で M.S. および Ph. D を取得。IEEE 機関誌にフィーチャ・セレクション関連の論文を発表。

IEEE, Sigma Xi, Eta Kappa Nu のメンバであり、また、IEEE の電気工学における大学資格認定プログラムの特別審査員である。

R. A. Higgins

Sperry 社の半導体部門に3年間在職。また、Defense Product 部門に1年間在職。現在、Defense Products グループで、CAD システムに関連したエキスパート・システムを開発中。半導体部門では、SCROL およびその他の CAD プロジェクトに従事。入社前は、CONTROL DATA 社に在職し、イメージ・システム設計、エネルギー・マネジメントおよびディスク・メモリの開発に従事。また South Dakota 州立大学で、8年間、電気工学の教授を勤めた。

Minnesota 州立大学で B.E.E., Wisconsin 大学で M.S.E.E., Missouri 大学で電気工学で Ph. D. を取得。審査論文誌に、制御と制御機器に関する論文を数種類発表。

IEEE および Sigma Xi のメンバであり、公認のプロフェッショナル・エンジニア。また、Minnesota 州の Eden Prairie 教育委員会の理事。

報告

漢字の意味を利用した係り受け辞書による
カナ漢字変換システムについてA Kana-kanji Conversion System with the Kakari-uke
Dictionary Utilizing the Meanings of Kanji

稲永 紘之, 橋本 和博

要 約 カナ漢字変換処理においては、文節分かち書きのカナ漢字変換から、最近では連文節あるいはべた書き入力文のカナ漢字変換への研究の移行が見られるが、これらはいずれも文節単位にカナ漢字変換を行うことがその基本にある。

文節単位のカナ漢字変換処理では、自立語と付属語の文法解析のほかに、接頭語や接尾語の処理、複数個の単語が連なった複合語の処理、同形異義語の処理、未登録語の処理などが必要である。

本稿では、接尾語や同形異義語の曖昧さを解消するために「漢字(の意味)情報を用いた係り受け処理」を行うカナ漢字変換処理に焦点を絞り、処理の概要と実験結果および今後の課題について述べる。

Abstract The current Kana-Kanji conversion has been executed with the input of the Japanese sentences which are manually separated into sentence elements in advance. However, recently, the emphasis has been placed on the study of the algorithm which enables the solid written sentences to be separated automatically into the sentence elements.

All of these are both based on the element-by-element Kana-Kanji conversion of Japanese sentences. In addition to the syntactical analyses of independent words and adjuncts, such conversion requires the processing of prefixes, suffixes, compound words, i. e., the consecutively linked words, homographs, and unknown words.

This paper contains the outline, the test results, and the future problems of the Analyses of Word Relations using the meanings of Kanji which intends to eliminate the ambiguity of the suffixes and homographs in the Kana-Kanji conversion processing.

1. はじめに

日本語文を機械処理する場合、同形異義語の判別や機械翻訳における多義語の判別を自動化するためには、単に単語辞書の文法情報を用いるだけでは限界がある。

この問題を解決するためには、「語」が単独に持っている特性（表記、品詞、活用型等）だけでなく、語と語の関係に関する特性（格関係、慣用表現、上位/下位の関係等）を定義し、利用する方法が考えられる。

語と語の関係によって語の特性を明確にしようとする試みには、連想方式（京都大学、国立国語研究所）、国語辞典の語釈義文の利用（長崎大学）、構文解析結果の利用（富士通、日電）、係り受け辞書の利用（九州芸術工科大学）などがある¹⁾。

このうち、九州芸術工科大学（以下、芸工大と略称）では、使用頻度の高い係り受けの句あるいは慣用句を標準形で登録した辞書「標準の係り受け辞書」を用いて、同形異義語や接尾語、未登録語などの処理をする実験を行っている。さらに、この標準の係り受け辞書をもとに「漢字情報を用いた係り受け辞書」、「名詞の大分類を用いた係り受け辞書」を作成して、さらに精度のよい実験システムの開発も行っている。

本稿においては、漢字情報を用いた係り受け処理に焦点を絞り、その処理のカナ漢字変

換処理システム全体の中での位置付け, 処理の概要, 実験結果および今後の課題について述べる.

2. 漢字情報を用いた係り受け処理の位置付け

ここでは, 漢字情報を用いた係り受け処理の, カナ漢字変換処理システム全体の中での位置付けについて述べる.

カナ漢字変換処理システムの全体を, 図1に示す.

システム全体の処理の流れは, 次のとおりである.

入力文章(文節単位に分ち書きされた複数文. このシステムでは, べた書き入力文の自動分ち書きは行っていない)は一括入力後降順にソートされ, 自立語辞書, 付属語辞書, 接頭語接尾語辞書を用いて各文節単位に文法解析が行われ, マッチングのとれた自立語, 付属語(例), 接頭語, 接尾語(列)はすべてファイルに格納される.

この後, 「標準の係り受け辞書」を用いた係り受けの検定(自立語間の係り受けが成立

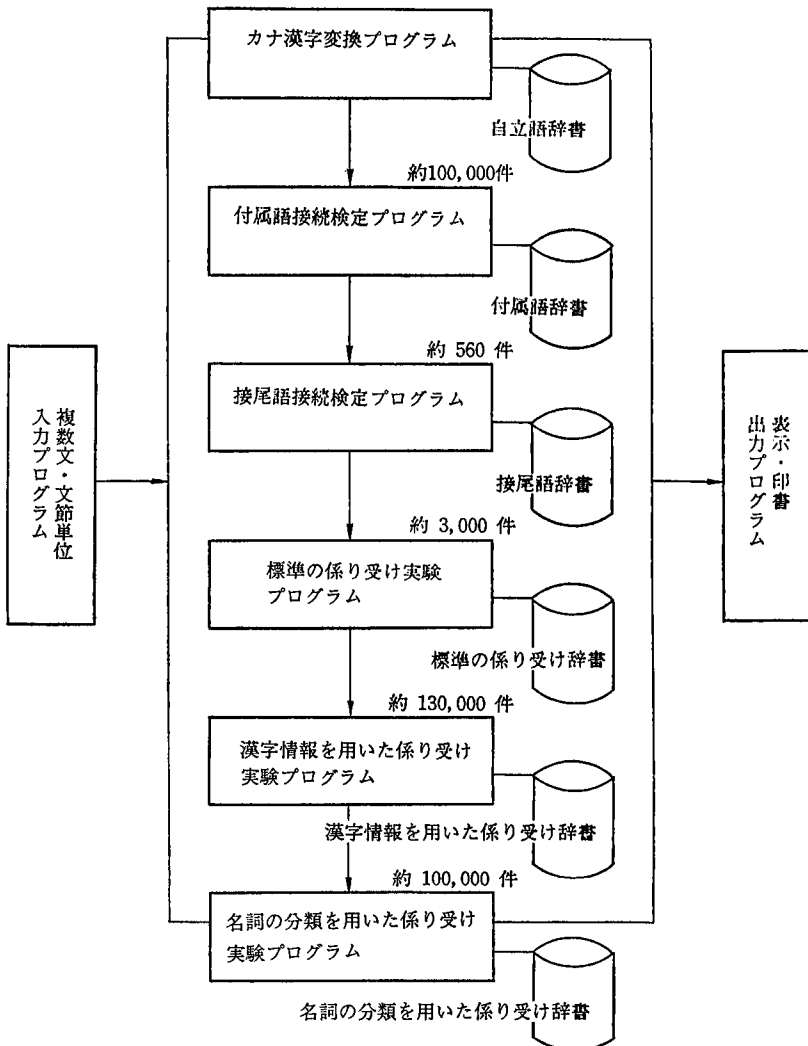


図1 カナ漢字変換システム全体図

Fig. 1 Kana-Kanji conversion system

するか否か)が行われ、係り受け関係の成立した自立語が同形異義語群中で第一義のものと認定される。

つぎに、前記の各処理を行ってもまだ曖昧さの残った同形異義語や接尾語に対して、「漢字情報を用いた係り受け辞書」が「標準の係り受け辞書」に代わって用いられる。

この処理は、同形異義語や接尾語の末尾の漢字情報をもとに、辞書（漢字情報を用いた係り受け辞書）を用いて係り受けの検定を行い、係り受け可能であればそれを第一義のものと認定してこれらの曖昧さを減らそうとするものである。

3. 係り受け辞書の構成

この章では、「標準の係り受け辞書」の概要を説明した後この辞書をもとに作成され、本実験で使用する「漢字情報を用いた係り受け辞書」について述べる。

3.1 標準の係り受け辞書

この辞書は、慣用句や結び付きの強い文節間の係り受け関係を、係り受けの句として登録したものである。

係り受けの句には二つのタイプがある。一つは短単位の自立語が直接接続した、いわゆる複合語である。

この複合語は入力文中では助詞を介して2単位以上に分かち書きされる可能性もあるため、辞書中の「読み」(カナ見出し)の項目中で、分かち書きされる可能性のある個所に“*”のマークを入れている(図2)。

【例】 ブンカ*イサン (文化遺産)
ゴ*コウセイ (語構成)

二つめのタイプは、主として和語の用言を用いた係り受けの句で、2単位以上に分かち書きされて入力されることが明らかな係り受けの句を辞書に登録したものである。

この場合、辞書中の「読み」(カナ見出し)の項目中、係りの語と受けの語の間には文節の切れ目を示す“+”のマークを入れている。

カード番号	読み	切れ目	順位	漢字	文法情報
A1702	アンシ ^ハ ン*リウメイ	1 2332		安心*立命・(する)	昔名1【9】 サ変(二自)
07906	アンシ ^ハ ウノ*チ	1 2411		安住の*地	昔名X【1】
07908	アンシ ^ハ ウ#+モトメ	1 24121		安住を 求め(る)	下一段(二自)
07909	アンシ ^ハ ウ#+ハカ	1 2412		安住を 凶(る)	五段(二自)
07911	アンシ ^ハ ウ#+ネカ ^ハ	1 2413		安住を 煩(う)	五段(二自)
07910	アンシ ^ハ ウ#+エ	1 2411		安住を 得(る)	下一段(二自)
07885	アンシ ^ハ ニ+ヨウ	2212		暗示に 弱(い)	形容詞
07886	アンシ ^ハ ニ+マト ^ハ	2213		暗示に 感(う)	五段(自)
07884	アンシ ^ハ ニ+カケ	22111		暗示に 掛け(る)	下一段(他)
07883	アンシ ^ハ ニ+カカ	22111		暗示に 掛か(る)	五段(自)
07892	アンシ ^ハ カ ^ハ +ヨウ	2222		暗示が 弱(い)	形容詞
07891	アンシ ^ハ カ ^ハ +ツヨ	2222		暗示が 強(い)	形容詞
07890	アンシ ^ハ カ ^ハ +クワウ	22221		暗示が 加わ(る)	五段(二自)
Z4351	アンシ ^ハ *リョウホウ	1 2232		暗示*療法・(する)	昔名1【1】 サ変(二自)
07889	アンシ ^ハ #+クワエ	22121		暗示を 加え(る)	下一段(二自)
07888	アンシ ^ハ #+アタエ	22121		暗示を 与え(る)	下一段(二自)

図2 標準の係り受け辞書

Fig. 2 Standard Kakari-uke dictionary

3.2 漢字情報を用いた係り受け辞書

3.1節で述べた単語レベルでの係り受けの句を用いて同形異義語などの処理を行う方法はきわめて有効であるが、この方法だけでかなりの成果を上げようとする、少なくとも見積っても数百万の係り受けの句を登録する必要がある、実用的ではない。

ここでわれわれは、標準の係り受け辞書に係り受けの句を登録していく過程で、次のような事実に着目した。

すなわち、名詞を漢字表記したとき末尾の漢字が意味の主体となる場合がかなり多く、そのような構造を持つ名詞が格助詞を介して他の語に係るとき、どのような語に係るかは末尾の漢字の性質（意味）によって決まる割合が多く、しかも末尾の漢字が同一の名詞群は同じ語に係ることが多いという事実である。

片カナ書き外来語 4,210 を含む名詞 72,797 について調べた結果では、末尾の漢字が意味の主体を担う名詞性の名詞（廃“液”，組“曲”，恩“人”，など）は 58.3 パーセント、末尾の漢字が動詞性のものであるが、意味の主体を成すと考えられる名詞（溺“死”，転“落”，昼“寝”，など）まで含めると 67.7 パーセントになる。

さらに、接尾語がついて漢字 3 文字で表される名詞（言語学，転落死，など）では、圧倒的に末尾の漢字が意味の代表である場合が多く、これらを含めて考えると実際には末尾の漢字がその名詞の意味の中心をなす割合は 70 パーセントを越えると推測される。

そこで、標準の係り受け辞書から末尾の漢字をキーとして名詞を抜き出し、得られた名詞の集合から上記の構造を持つ名詞を選択し、その末尾の漢字と受けの語を辞書（漢字情報をういた係り受け辞書と呼ぶ）に登録することによって、辞書をコンパクトにし、また、きめの細かい係り受けの句を作成することができる。

たとえば、末尾の漢字が「酒」であり、これが意味の主体をなしている名詞の集合には、「洋酒」，“日本酒”，“梅酒”，“御神酒”，“地酒”，“密造酒”などがあり、この集合に対して「に弱（い）」，“を飲（む）」… のような受けの語を登録する。このとき、「酒」の読みは“シュ”，“サケ”，“ザケ”のいずれでもよい。また「造り」，“並み”のような接尾語もこの「酒」の集合に対して登録しておけば、未登録自立語の問題解決にも役立つことになる。

この「漢字情報を用いた係り受け辞書」は、「標準の係り受け辞書」と「名詞の分類を

	7		24		34		65		73	74		77		101
…	カナ見出し	…	漢字表示部	…	自	他	…	サ	……	代	……			

カナ見出し：カナ見出し部の“*”は、その前後の文字列が格助詞を介さなくても接続可能であることを示し，“+”は文節の切れ目であり、文字列が2文節に分解されて入力されるときは、この位置に格助詞あるいはその変化した助詞（列）を必要とすることを示している。

〔例〕 ガク*センコウ（～学（を）専攻「する」）

 ホウ+サガ（～宝（を）探「す」）

漢字表示部：JIS 非漢字/第一水準漢字コードを用いた漢字コード列

自 ：自動詞を、その要求する格助詞によって分類したコード

他 ：格助詞“ヲ”を介して対象となる名詞を要求する動詞を示すコード

サ ：サ変動詞表示

代 ：普通名詞代表漢字数部

その名詞の末尾から何個の漢字表記部の漢字が意味の主体をなすかを示す。

図 3 漢字情報を用いた係り受け辞書の構成

Fig. 3 Format of Kakari-uke dictionary which utilizing the meanings of Kanji

用いた係り受け辞書][†]を補完するものであり、現在約 10 万件の係り受けの句が仮登録されているが、今回はこのうちチェックの終了した約 5,000 件の係り受けの句を用いて実験を行った。

4. 処理の流れ

4.1 係り受けの検定

前章までの処理で文節単位に文法構造が解析され、自立語部になりうるすべての単語は自立語辞書より選ばれ、ファイルに格納されている。そして、そのファイルに対するポインタを持った入力テーブル上の文が、漢字情報を用いた係り受け辞書を使用した係り受け検定の対象となる(図4)。

係り受けの検定は、自立語または接尾語の末尾の漢字が意味の主体を成すと考えられるものに対して行われる。このとき、末尾の漢字の漢字コードと、漢字情報を用いた係り受け辞書の係りの語の漢字コードが一致するものが探される。一致するものがあれば、辞書中の受けの語の「読み」に一致する語が入力テーブル中にあるか否かが文の範囲内で探索される。

ただし、接続助詞を含む文節が文の中に現れた場合は、その文節までを1文とみなして、その範囲内で係り受けの検定を行っている。

係り受けの句のうちNP型の句(標準の格助詞がついた名詞とそれを受ける用言のつながり:名詞-用言型)は、倒置して用いられることもあるため、受けの語の探索は係りの語の前後に対して行われる。

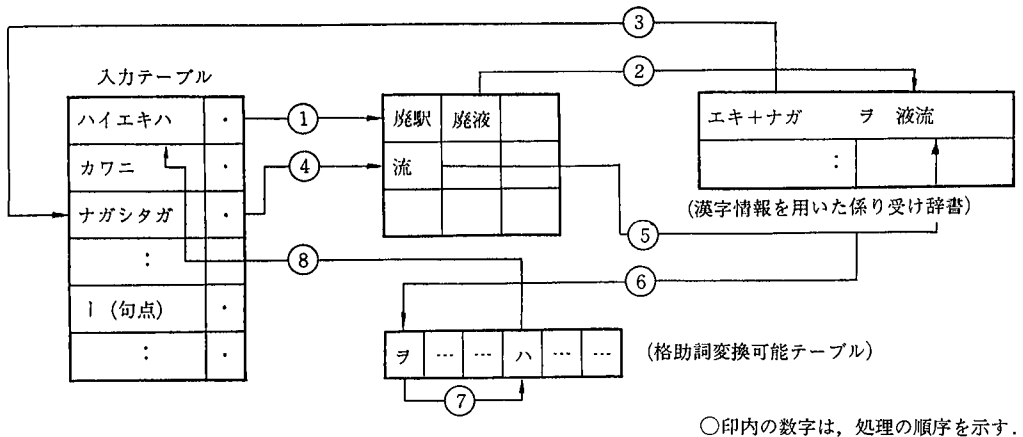


図4 漢字情報を用いた係り受けの検定

Fig. 4 Processing of Kakari-uke relation

4.2 文法情報の検定

文法情報の検定は、入力テーブル中にある受けの語の文法情報と、漢字情報を用いた係り受け辞書に登録された受けの語の文法情報が一致するか否かによって行われる。

[†]「名詞の分類を用いた係り受け辞書」とは、「人間」、「動物」、「植物」、などの名詞の分類に対して、たとえば「[動物]は走(る)」という係り受けの句を登録したものである。動物の個所に動物に属する名詞が代入可能であることによって、用例の数の大幅な減少が図られる。

各名詞については、固有名詞は7種類、普通名詞の基本語/複合語/対義語は35種類に分類して辞書に登録している。

検定の対象となる文法情報には次のものがある。

- 1) 動詞の場合……12 種類の活用の型（五段，上一段，下一段など），さらに五段活用動詞の場合には活用行
- 2) 形容詞の場合……活用の型（ク活用／シク活用，口語形容詞／文語形容詞）
- 3) 形容動詞の場合……活用の型（文語形容動詞のタリ活用／ナリ活用，口語形容動詞など）

4.3 格助詞の変換

文法情報が一致した場合，さらに標準格助詞が使用されているかどうか判定される。もし使用されていない場合は，格助詞変換可能テーブルによって，標準格助詞を変換可能な助詞（列）に変換した後，使用されている助詞とのマッチングがとられる。

標準格助詞と変換可能性のある助詞（列）との対応表（変換可能テーブル）を表 1 に示す。

表 1 標準格助詞と変換可能助詞（列）

Table 1 Standard case particles and convertible particles (line-up)

標準格助詞	変換可能な助詞（列）
ノ	なし
ニ	ヘ ト
ト	ニ
デ	なし (表 2 の探索)
ガ	ノ ニ デモ デサエ デサエモ デサエモガ カラ
カラ	ヨリ ヲ (バ)
ヲ	ヨリ ノ ガ カラ

表 2 副助詞と係助詞

Table 2 Sub-particles and Kakari-particles

ヤラ	モ	マデ	ホド	バカリ	ハ	ノミ	ナリ	ナンゾ	ナンカ	ナド
ナゾ	デモ	デスラ	デサエ	ダノ	ダッテ	ダケ	スラ	シカ	サエ	
コソ	グライ	クライ								

表 3 格助詞

Table 3 Case particles

ヨリ (ノ)	ヘ (ノ)	ノ	ニト (ノ)	デ	ガ	カラ (ノ)	ヲ
--------	-------	---	--------	---	---	--------	---

入力された助詞（列）が，標準格助詞あるいは変換可能な助詞（列）と一致しない場合や，未一致の残余部がある場合は，表 2（副助詞と係助詞のテーブル）および表 3（格助詞テーブル）を交互に繰り返し使用することによって，入力された助詞列を分解し，マッチングがとられる。

なお，この格助詞変換可能テーブルは，変換可能助詞（列）の追加や，副助詞／係助詞と格助詞の接続関係の登録などについて現在見直しを行っている。

4.4 係り受け関係の成立

以上の検定がすべて成立したとき，係り受け関係が成立した係りと受けの単語を，同形異義語群または接尾語群の中の第一義のものと認定する。

この検定方法では、一つの係りの語に対して複数の受けの語との間で係り受け関係が成立する場合がある。この場合は、係りと受けの距離（係りの語を含む文節と受けの語を含む文節の距離）と、辞書に登録されている係りの強さ（3段階）から評価関数の値を求めて、複数の受けの語から最適と思われるものを一つ選択する。

5. 実験

ここでは、第3章で述べた係り受け辞書と、第4章で述べた係り受けの検定方法による「漢字情報を用いた係り受けの実験とその結果」について述べる。

実験プログラムの使用言語は COBOL であり、芸工大情報処理センターに設置してある汎用計算機の TSS のもとで実行し、入出力はカナ鍵盤付きの漢字端末装置によって行った。

5.1 文の入力

文は、文節単位に分かち書き入力され、ファイルに格納される（図5）。

ファイルに格納された文章は、カナ漢字変換プログラムや付属語接続検定プログラム等によって文節単位に解析され、自立語、付属語（列）、接尾語（列）がファイル上に書き出される。

1. 000 カフ センリョクノ スイイハ、 ヒカシ カワカ ユウニ タツチイルコトニ トモナツテ、 ニシ カワチノ ヲ シツヲ モツテ
2. 000 カズヲ オキナウトイウ コレマデノ キシユツヂキ ユウイサエ ウシナツチイル。
3. 000 サクセンノ セイカノ コロ（ ナツノ サカリニ ）, アノ カイシャノ ハイエキヲ タイリョウニ カワニ ナカシタノデ、
4. 000 シンファンタネニ ナリマシタ。
5. 000 カワノ サイト ヨコレ フアイト ソクタイシマシタ。

図5 入力文章

Fig. 5 Input sentences

図6に示したのは、上記の処理を行って作り出された入力テーブルであり、カナ見出し、接続する助詞の有無、ファイル中の自立語／付属語（列）／接尾語（列）の相対ブロック番号などが格納されている。

カナ見出し中の％は読点、|は句点、＃はカナの“ヲ”を表している。

5.2 カナ漢字変換結果

入力文章（図7）をカナ漢字変換した結果を、図7に示す。

ここでは、まだ係り受けの検定を行っておらず、“核戦力の水位”、“成果の頃”、“廃駅を…流した”などと正しくない変換が行われたままである。*印のついた語は同形異義語が存在することを表し、同形異義語はその読みと共に図7の下部に一括して示している。

たとえば、“核戦力の水位”が“核戦力の推移”と変換された経過は次のとおりである。

3番目の文節（スイイハ：水位は）は、9番目の文節（トモナツテ：伴って）と20番目の文節（ウシナツチイル：失っている）と係り受けの関係が成立する（漢字情報を用いた係り受け辞書にこれらの係り受け関係が登録されており、かつ文法情報のチェック、格助詞の変換も正しく行われた）。

- ① 移+伴う（係り受けの距離は6）
- ② 位+失う（係り受けの距離は17）

（辞書にも登録されている“係り受けの強さ”は、いずれも2である）

この場合、評価関数（係り受けの距離×係り受けの強さ）の値が小さい①を優先して（水位→推移）の変換が行われている。

1 - カフ	2 SN-+ 31 KS- KP-	KM- KB- 28 UB- 1
2 - センリヨフノ	6 SN-* 23 KS-1 KP-04000000	KM- KB- 14 UB- 5
3 - スイイハ	4 SN-+ 41 KS- KP-	KM- KB- 16 UB- 3
4 - %	1 SN- KS- KP-	KM-% KB- UB-
5 - ヒカシ	4 SN-+ 32 KS- KP-	KM- KB- 4 UB- 4
6 - カフカ	5 SN-* 40 KS- KP-	KM- KB- 25 UB- 3
7 - ユウニ	4 SN-+ 19 KS- KP-	KM- KB- 2 UB- 3
8 - タツマイルコトニ	8 SN-+ 33 KS-1 KP-02000000	KM- KB- 12 UB- 1
9 - トモナツテ	5 SN-* 8 KS-1 KP-04000000	KM- KB- 10 UB- 3
10 - %	1 SN- KS- KP-	KM-% KB- UB-
11 - ニシ	2 SN-+ 38 KS- KP-	KM- KB- 6 UB- 2
12 - カフチハ	6 SN-* 6 KS- KP-	KM- KB- 25 UB- 3
13 - シツキ	3 SN-+ 22 KS- KP-	KM- KB- 18 UB- 2
14 - モツテ	3 SN-+ 5 KS-1 KP-02000000	KM- KB- 3 UB- 1
15 - カス*	4 SN-* 1 KS- KP-	KM- KB- 27 UB- 3
16 - オキナウトイウ	8 SN-* 20 KS- KP-	KM- KB- 30 UB- 4
17 - コレマナノ	6 SN-* 43 KS- KP-	KM- KB- 22 UB- 5
18 - キシユツキ	8 SN-* 12 KS-1 KP-05000000	KM- KB- 24 UB- 8
19 - ユウサイ	5 SN-+ 14 KS- KP-	KM- KB- 2 UB- 3
20 - ウシナツマイル	7 SN-* 30 KS-1 KP-04000000	KM- KB- 31 UB- 3
21 - l	1 SN- KS- KP-	KM-l KB- UB-
22 - サクセンノ	5 SN-* 27 KS- KP-	KM- KB- 19 UB- 4
23 - セイカノ	4 SN-+ 3 KS- KP-	KM- KB- 15 UB- 2
24 - コロ	2 SN-+ 17 KS- KP-	KM- KB- 21 UB- 2
25 - (2 SN- KS- KP-	KM-S KB- UB-
26 - ナツノ	3 SN-* 35 KS- KP-	KM- KB- 8 UB- 2
27 - サカリニ	4 SN-* 24 KS- KP-	KM- KB- 20 UB- 3
28 -)	2 SN- KS- KP-	KM-S KB- UB-
29 - %	1 SN- KS- KP-	KM-% KB- UB-
30 - アノ	2 SN-* KS- KP-	KM- KB- 32 UB- 1
31 - カイシヤハ	5 SN-+ 16 KS-1 KP-04000000	KM- KB- 29 UB- 2
32 - ハイエキ*	5 SN-+ 11 KS- KP-	KM- KB- 5 UB- 4
33 - タイリヨウニ	6 SN-+ 44 KS-1 KP-04000000	KM- KB- 11 UB- 5
34 - カワニ	3 SN-+ 15 KS- KP-	KM- KB- 26 UB- 2
35 - ナカシタノ*	8 SN-* 9 KS- KP-	KM- KB- 9 UB- 1
36 - %	1 SN- KS- KP-	KM-% KB- UB-
37 - シンフンダネニ	9 SN-* 13 KS- KP-	KM- KB- 17 UB- 5
38 - ナリマシタ	5 SN-+ 26 KS- KP-	KM- KB- 7 UB- 1
39 - l	1 SN- KS- KP-	KM-l KB- UB-
40 - カワノ	3 SN-+ 34 KS- KP-	KM- KB- 26 UB- 2
41 - スイト	4 SN-+ 37 KS- KP-	KM- KB- 33 UB- 3
42 - ヨコレ	4 SN-* 7 KS- KP-	KM- KB- 1 UB- 4
43 - ツアイ*	5 SN-* 18 KS- KP-	KM- KB- 23 UB- 4
44 - ソクテイシマシタ	8 SN-* 2 KS- KP-	KM- KB- 13 UB- 4
45 - l	1 SN- KS- KP-	KM-l KB- UB-

図 6 入力テーブル

Fig. 6 Input sentences after analyzing bunsetsu

41 番目の文節（スイイト：水位と）は、45 番目の文節（ソクテイシマシタ：測定しました）と係り受け関係が成立（位+測定）し、変換結果は“水位”のままである。

5.3 係り受けの検定結果

図 8 では上記の文章に対して、漢字情報を用いた係り受けの検定を行った結果を示している。

ここでは“核戦力の推移”，“盛夏の頃”，“廃液を…流した”と、正しく変換されている。

*核戦力の*水位は、*東側が*優位に*立っていることに伴って、*西側では*質を*持って数を補うという此れ迄の技術的*優位さえ失っている。
 昨年の*成果の*頃(夏の盛りに)、あの*会社は*廃駅を*大量に*川に流したので、新聞種に*なりました。
 *川の*水位と汚れ具合を測定しました。

【カク】	格, 斯く, 佳句, 角, 書; く, 欠; く, 掻; く
【スイイ】	推移
【ヒガシ】	干菓子
【ユウイ】	有為, 有意, 雄偉
【タ】	断, 経, 足
【ニシ】	螺
【シツ】	室
【モ】	漏, 盛, 守
【ユウイ】	有為, 有意, 雄偉
【セイカ】	青果, 声価, 正価, 正貨, 正課, 生花, 生家, 盛衰, 精華, 聖火, 聖歌, 製菓, 製靴, 請暇, 所為; か, 膏; か
【コロ】	ころ
【カイシャ】	かいしゃ
【ハイエキ】	廃液
【タイリョウ】	大漁
【カワ】	皮, 革, 側, 佳話, 歌話
【ナ】	鳴, 為, 生, 成
【カワ】	皮, 革, 側, 佳話, 歌話
【スイイ】	推移

図 7 カナ漢字変換結果

Fig. 7 Experimental results of the conversion
(without using the Kakari-uke dictionary)

*核戦力の*推移は、*東側が*優位に*立っていることに伴って、*西側では*質を*持って数を補うという此れ迄の技術的*優位さえ失っている。
 昨年の*盛衰の*頃(夏の盛りに)、あの*会社は*廃液を*大量に*川に流したので、新聞種に*なりました。
 *川の*水位と汚れ具合を測定しました。

【カク】	格, 斯く, 佳句, 角, 書; く, 欠; く, 掻; く
【スイイ】	水位
【ヒガシ】	干菓子
【ユウイ】	有為, 有意, 雄偉
【タ】	断, 経, 足
【ニシ】	螺
【シツ】	室
【モ】	漏, 盛, 守
【ユウイ】	有為, 有意, 雄偉
【セイカ】	青果, 声価, 正価, 正貨, 正課, 生花, 生家, 成果, 精華, 聖火, 聖歌, 製菓, 製靴, 請暇, 所為; か, 膏; か
【コロ】	ころ
【カイシャ】	かいしゃ
【ハイエキ】	廃駅
【タイリョウ】	大漁
【カワ】	皮, 革, 側, 佳話, 歌話
【ナ】	鳴, 為, 生, 成
【カワ】	皮, 革, 側, 佳話, 歌話
【スイイ】	推移

図 8 漢字情報を用いた係り受けの解析結果

Fig. 8 Experimental results of the conversion
(after using the Kakari-uke dictionary)

5.4 検 討

以上で述べた係り受け処理は有効な方法ではあるが、たまたま係り受け辞書を用いて係り受け関係が成立した単語を優先するために誤変換を引き起こすことがある。

また、標準格助詞が他の助詞列に変換されるとき、変換規則が文脈に依存する場合もあるため、条件の設定が必要である。詳細な条件については、現在検討中である。

これらは、辞書の不備とともに文の構造を解析する手法が十分でないことが原因であり、その解決は今後に残された課題である。

さらに現行のシステムでは、代表名詞漢字数部が1または6のもの（末尾の漢字が意味の主体を成すと考えられるもの）すべてに対して“受けの語”を探し、辞書とのマッチングがとれたものを“正しい係り受け関係”にあるものと仮定して処理している。

したがって、係り受けの非交差性(文節間の係り受け関係は互いに交差してはならない)については何ら保証されているものではなく、係り受けが交差した文は“文”そのものを誤りとして処理するか、あるいは各々の係り受け関係の取捨選択を何らかの方法で決定するなどの検討が問題として残されている。

また、本稿で述べた係り受け辞書の無制限な増大を抑えるため、各名詞に付与した分類記号(32種)を利用する係り受け辞書の作成および作成済みの対義語辞書のシステムへの組み込みも検討している。

6. おわりに

昨今のパーソナル・コンピュータやワード・プロセッサの発達は目ざましく、カナ漢字変換は今やこれらの小規模システムで行われるものが大部分となった。

これらは入力文の先頭から末尾に向かっての文節処理が基本にあり、人間の介入を前提として既出の単語の頻度情報や直前の使用語等の情報をいかに有効に用いるかということで同形異義語の処理を行おうとしている。

本稿で述べた方法は、まとまりのある文章を一括入力し、文章中に散在する係り受け関係を、辞書に登録してある係り受けの句を適用して係り受けの検定を行い、係り受けの関係が成立した同形異義語群中の単語を他に優先して選択しようとするものである。また、同時に文の構造解析いわゆるパーズングにも役立てようとするのが基本にある。

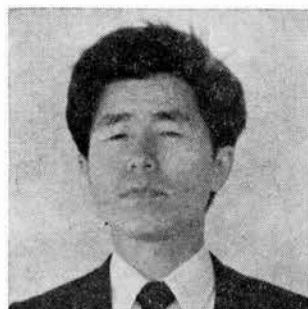
現状ではいわゆる完全な自動カナ漢字変換システムの構築は望み得ないが、本稿で述べた文節間の関係を辞書中に有効に組み込む方法は、構文解析時における曖昧さの減少(係り受け関係が成立したものを優先するなどの方法によって構文解析木の数を減らす)、あるいは手書き文字入力や音声入力等において、認識された文字列から意味のある日本語を抽出するなどへの応用が考えられる。

本研究を行うに当たり、ご指導ご鞭撻いただいた吉田将教授(九州大学工学部)に深甚の謝意を表す。

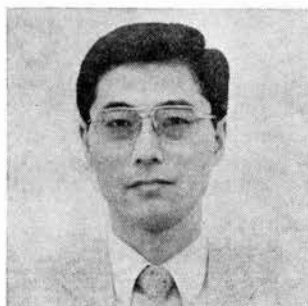
-
- 参考文献 [1] 田中康仁, 吉田 将, “自然言語の分析による知識データの収集”, 自然言語処理技術シンポジウム論文集, 1984 (11月).
- [2] 稲永紘之, 小山憲一, “文節構造解析のための接続辞書の構成と付属語の接続検定”, 九州大学工学集報, Vol. 54, 1981 (8月).
- [3] 稲永紘之, 小西彬允, “漢字情報による名詞の分類と係り受け辞書”, 昭和50年度電子通信学会講演論文集, 1975.
- [4] 稲永紘之, 小西彬允, “カナ文字文の機械処理のための辞書について”, 電子通信学会オートマトンと言語研究会資料, 1981 (10月).
- [5] 稲永紘之, 吉田 将, “日本語処理のための機械辞書”, 情報処理, Vol. 23, No. 2, 1982.
- [6] 稲永紘之, 橋本和博, 吉田 将, “漢字情報を用いた係り受けの実験について”, 昭和59年度電気関係学会九州支部連合大会講演論文集, 1984.
- [7] 稲永紘之, 橋本和博, 吉田 将, “係り受け辞書のカナ漢字変換への応用”, 昭和60年前期情報処理学会全国大会講演論文集, 1985.

執筆者紹介 稲永 紘之 (Hiroyuki Inanaga)

1965年九州大学工学部電子工学科卒業。1970年同大学工学研究科博士課程退学。同年九州芸術工科大学講師となる。1967年より仮名漢字変換を中心とする日本語文の機械処理に関する研究に従事。1975年電子通信学会米沢賞受賞。電子通信学会および情報処理学会会員。

**橋本 和博 (Kazuhiro Hashimoto)**

1969年大阪工業大学電子工学科卒業。同年日本ユニバック(株)入社。客先担当 SE, 教育担当, テクニカル・パブリケーション担当を経て、現在知識システム開発部に所属。エキスパート・システムの開発および導入に従事。



論文 知的システムで使う問題解決戦略

Problem Solving Strategies Used by Intelligent Systems

M. D. Brouwer-Janse

要 約 本稿では、まず人間および AI システムの双方に頻繁に利用される問題解決戦略についての概要を述べる。つづいて、これらの戦略を取り入れた問題解決モデルを検討する。さらに人間の問題解決がどのように計測されるかを図示し、専門家および非専門家の問題解決の実行についての比較研究を述べる。

Abstract The purpose of this paper is to give an overview of problem solving strategies that are frequently used by both, humans and AI systems. Problem-solving models that account for these strategies are discussed. As illustration of how human problem solving can be measured, a study is presented in which problem-solving performance of experts and non-experts is compared.

1. はじめに

本稿では、問題解決の研究分野について概説する。まず、第2章で、AI と認知心理学における問題解決の関係を考察する。第3章では、AI 技法から生まれ、人間システムに適用される問題解決戦略を紹介する。なお、知的システムを作る場合には、使用する諸戦略やそれらの相互関係を定める問題解決モデルが重要である。これらの問題解決モデルについては第4章で述べる。最後の章では、問題解決の研究における方法論的問題を述べ、人間の専門家 (expert) と非専門家が使う問題解決の戦略についての研究とその分析データを述べる。

2. 問題解決, 人工知能および認知心理学

応用人工知能の主な目標は、ある仕事について人間の行動に似た行動を示す問題解決システムを構築することである。たとえば、エキスパート・システムの目標となるような人間の専門家の問題解決においては、専門領域の知識、問題を分類する専門知識、および一般的な問題解決知識の少なくとも三つの異質な問題解決知識に分類できる。

AI や人間の問題解決における大半の研究ではこれまで、専門領域の知識に焦点を当ててきた。特定の領域における問題解決では、概念、原理、対象およびそれらの関係に関するすべての知識が必要となる。これらの知識は、多くの場合、テキストや訓練や経験などから修得される。また問題を分類する専門知識では、問題のタイプや複雑さの程度を分類する分析能力を使う。この専門知識は、専門領域から独立したものであり、診断、分析、総合、比例、評価、確率などの一般的な作業特性に関係する。また一般的な問題解決の知識の場合には、効果的に適当な戦略を使い、利用できる発見的方法 (heuristics) を自由に駆使する必要がある。

人間の専門家と人工のシステムの間知識の移転を仕事とする知識工学エンジニアもまた、これらの様々な知識を持たなければならない。ただし、知識工学エンジニアの領域は人工知能である。これら各種の知識は相互に作用しあうため、たいいていの人工知能プログラムでは、同様の環境で人間がとる行動様式についてのプログラマの直感 (暗黙の理論)

に依存する。これらの暗黙の理論を明示化する作業は、認知心理学と人工知能の研究双方の中心課題である。

認知心理学の研究の主要な対象は、人間の問題解決、つまり思考である。また人工知能研究の主要な対象は、「人間が知能を使って行うようなことを機械に行わせること」である^[1]。人工知能と認知心理学の間の著しい違いは、前者が問題解決システムを総合するのに対し、後者は問題解決システムを分析する点にある。Boden^[1]はこの二つの分野間の相互的依存をとりあげて、「人工知能とはコンピュータを研究するのではなく、思考や行動に現れる知能を扱う研究である。コンピュータはその道具にすぎない」と明確に述べている。

手短かにいうと、人工知能と認知心理学における研究は次の(a)~(d)において、互いに補完しあう。(a)コンピュータの事例は思考を助ける機能を持つ。(b)コンピュータ・モデルは厳密である。(c)コンピュータによる処理は、作業条件を量的に表す。(d)コンピュータ・モデルでは、各処理を切り離すことができる。

各章では、AI と認知心理学の両分野に応用される問題解決の戦略をとりあげる。

3. 問題解決戦略

問題解決戦略とは、人間と AI システムの両方で特定な問題や問題の一部を解くのに利用される戦略である。与えられた問題の複雑さや性質に応じて、問題を解決するのに適当な戦略あるいは一連の戦略を選ぶ。ここでは次の五つの戦略を検討する。

- 1) アルゴリズムと発見的方法
- 2) AND/OR 木
- 3) 後ろ向きの推論
- 4) 類推による推論
- 5) 手段——目標分析

3.1 アルゴリズムと発見的方法

問題解決の研究では、計画ないし作用素を、アルゴリズムと発見的方法の二つのタイプに区別する。アルゴリズムは、数学では「任意の機械的あるいは帰納的計算手続き」と定義され、問題解決では「それに忠実に従えば、自動的に正しい解を生成するような規則の集合」と定義される^[2]。

情報処理システムでの発見的方法の定義はややあいまいである。たとえば、Lindsay および Normen^[2]は、発見的方法を「比較的利用しやすく、また前の問題解決でしばしば有効であった解の探索手続きあるいはその概要」と定義している。アルゴリズムとは異なり、発見的方法では、成功は保証されない。Barr および Feigenbaum^[3]は、人工知能分野で開発された少なくとも六つの異なる発見的方法の定義を区別している。発見的方法の役割については、Newell および Shaw^[5]の説明が最も簡潔である。すなわち、「一般に解の探索の手間を小さくする原理や工夫」である。

3.2 AND/OR 木, ゴール木, 家族木

木構造は、基本的に分類戦略をある問題に適用した結果得られる。分類戦略を最初に研究したのは、ゲシュタルトの心理学者 Duncker^[5]である。彼は実験で問題を以下のように設定した。

「ある人が手術不能な胃かいようを患っている。この人は現在、有機組織に対してかなり強烈な破壊力をもつ放射線を照射されているものとする。さて、どんな手順を取ればこ

これらの放射線によって腫瘍を取りさり、同時に周囲の健康な組織を破壊しないで済むか。」

そして Duncker は被験者がこの問題を解決するために発言した内容を記録した。その問題解決のプロトコルを分析すると、解法がいくつかのグループに整理されることがわかった。その結果から、これらの解法は系図のような形式で表せた。ごく典型的な被験者の場合には、木のある枝を降りる特徴を示した。この解法では、問題のある部分問題に分割し、それを解く過程を反復する。そして結局、ある時点で、問題が解けなくなるか、あるいは、ある実行可能な解に到達する。

人工知能では、この問題解決戦略を目標の還元ないし問題の還元と呼び、この木を目標木と呼ぶ。目標木は、AND ノードと OR ノードからなり、AND/OR 木と呼ばれる。

3.3 後ろ向きの推論

目標状態からその仮定に向かって後ろ向きに解くのは、きわめて有効な戦略である。この戦略は、しばしば Plato に始まるとされる。たとえば、以下のような問題を考えてみよう。

「4 クォーツと9 クォーツの容器だけを使って6 クォーツの水を量るにはどのようにすればよいか。」

まず、9 クォーツの容器を満たしてからそれを4 クォーツの容器に注ぎ、大きな方に5 クォーツ分だけ残すかもしれない。では次にはどうするか。後ろ向きの推論を用いて別の戦略でやり直す方法もある。ところで、問題が解かれているとしよう。このとき、9 クォーツの容器に6 クォーツの水が入っている。この目標状態の直前の状態はどうだったのだろうか。4 クォーツの容器に3 クォーツの水を注いだに違いない。では、その前の状態はどうだろうか。4 クォーツの容器にはすでに1 クォーツの水があったはずである。ではその前の状態はどうだろう。明らかに、9 クォーツの容器に水を入れ、4 クォーツの容器に2 回水を注ぎ込んだはずである。などと続く。この問題の解法は逆の順序で操作の列を実行しているにすぎない。

目標状態から解いていく後ろ向きの探索は、人工知能プログラムでよく応用される技法である。

3.4 類推による推論

問題解決のある非常に重要な戦略として、当面の問題と解がわかっている問題の間の類似点を探す方法がある。この戦略では明らかな相違点にもかかわらず、類似点を認識する技術が必要である。

次のような問題が与えられているとする^[6]。

「New York 市にはピアノの調律師が何人いると思うか。」

ある解答者はこの問題を次のように解くだろう。New Haven 市にはピアノの調律師が3~4人いる。一方、市の人口は約30万人である。だから、その比はおよそ1対100,000だ。New York 市の人口は7,000,000人なので、約70人ということになる。したがって、50人から60人ほどだと思う。

この解法では、まず解答者が自分が知っている事実を利用し、その事実を決定要因として仮定して、一般的法則を作成し、その規則を適用して、他の変数に合うようにその結果を修正して、最終的な答えを出す。類推の構造は、解答者には与えられておらず、たとえば、New Haven:3人のピアノ調律師:New York:?のように、自分で構成しなければならない。

構造化された類推問題では、解答者は、「A 対 B は、C 対 P, Q, R, S のいずれかに

等しい」のような形で与えられた問題を解く。この類推では、(a) AをBに変える変換を発見し、(b)この変換をCに適用して理想的な類推を求め、そして(c)各選択肢 P, Q, R, S とこの理想的な類推を比較して最も近いものを選ぶことによって解を求める。

さらに複雑な類推の場合、変換過程は簡単ではない。つまり対象の変換だけでなく対象間の変換も必要となるからである。

類推による推論に対して、AI プログラムで利用されるモデルでは同じ推論パターンを用いる。Winston^[7]によれば、類推の問題を解く規則は、(a)もとの図式をその部分図式の相互関係によって記述し、(b)目標の図式をその部分図式の相互関係によって記述し、そして、(c)もとの図式の部分図式がどのように目標図式の部分図式に変換されるかを記述する、ことである。AI での類推による問題解決の基本戦略は、記述と照合 (describe-and-match) である。

3.5 手段-目標分析

問題によっては、現在の状態と望ましい状態との関係を考えるのが自然な場合がある。現在の状態とは、問題を含む現在の事実の状態であり、望ましい状態とは目標の状態である。このような問題は、手段-目標分析で解くことができる。手段-目標分析では、現在の状態と目標の状態の間にある相違を縮小する程度によって、処理手順を選び、結局、問題を一連の部分問題の列に分割する。

さて手段-目標分析をささえる直感的な推論とは、次のように記述できる。

「私は息子を幼稚園に入れたい。現在ののと希望のとの違いは何か。距離である。どうすれば距離は変わるか。車を使えばよい。私の車は動かない。それを動かすには何が必要か。バッテリーの交換である。新しいバッテリーはどこにあるか。修理工場にある。私は修理工場で新しいバッテリーを付け替えて貰いたいが、修理工場は私の要求を知らない。何が障害なのか。コミュニケーションの障害だ。何を使えば意志を伝達できるか。電話でできる。……」^[8]

手段-目標分析は、General Problem Solver (GPS) に組み込まれた基本的戦略である。GPS は、定理を証明したり、論理的数学的問題を解くために Newell および Simon^[8]が設計したコンピュータ・プログラムである。

4. 一般的な問題解決モデル

人間の問題解決についての研究の多くは、特定の問題のためのひとつの戦略の適用に焦点を置いてきた。だから、一般的問題解決モデル、つまり様々な戦略が組み込まれ、それらの戦略を有機的に組み合わせて使うモデルの開発例はまだほんの数例にすぎない。本稿では、これらのうち三つのモデルをとりあげる。最初のモデルは、Polya による発見的問題解決法である。このモデルでは、特定の専門分野に限らず一般の問題に応用できる、一般的な処方箋を与えてくれる。第2のモデルは GPS である。これは、よく構造化された問題に適用され、人間の問題解決行動について経験的に的確な実行モデルである。第3のモデルは、Piaget の形式的操作のモデルである。この認識論的モデルによって、人間の問題解決についての特定分野に限定されない認識構造の理論的枠組みが与えられる。

4.1 Polya の発見的問題解決法

ある問題を解こうとする際に、採用すべき方法を選択するアルゴリズムは存在しない。Polya^[9]の発見的問題解決法では、後ろ向きの推論や類推による推論、手段-目標分析など、いくつかの手法を組み合わせる。Polya の見解では、問題解決は、以下の四つの局面

から構成される。(a)問題の理解:すなわち目標、所定のデータ、制約条件などを判定する。(b)プランの策定:すなわち、与えられたデータと目標を結び付ける方法を発見して問題を再構成する。そして、問題の部分的解決を試みる。(c)プランの実行:すなわち、各ステップを実行し、各ステップをチェックする。(d)見直し作業:すなわち、問題を解いた後で、その結果を検証し、方法を再検討する。

Polya の発見的方法によって、問題を解決するための、非常に強力なツールが与えられる。Polya の意図は学生の問題解決能力を改善することであり、人間の問題解決の分析やシミュレーションのためではなかったけれども、彼の研究が問題解決研究に及ぼした影響は多大であった。

4.2 汎用問題解決プログラム: GPS

GPS は、手段-目標分析、縦型探索、問題還元などのいくつかの強力な問題解決戦略を織り込んでいる。GPS は AI に計画の概念を導入した最初のプログラムである。また表現の手段としてプロダクション・ルールを導入した。GPS はさらに人間の問題解決の最初の詳細なモデルでもあった。GPS では、人間の被験者とまったく同様に、一連の知識の状態の系列をたどり、目標を設定し、試行し、失敗すれば前の状態へ戻って、再び進んでいく。

Newell と Simon による問題解決へのアプローチは、問題の記述から最終解に至る事象の列を図式化することであった。被験者の問題解決行動を観察するのに、被験者が到達した解を書かせて、問題を解決したときには、“考えを述べる”ように依頼した。そして、被験者の発言を記録して、分析した。この分析のために、Newell と Simon は“問題空間”の概念を導入した。問題空間とは、その問題についての知識の可能な状態の集合とこれらの状態から新しい知識の状態を生成する操作の集合からなる。初期知識の状態を与え、問題の解を含む最終の知識の状態への経路の発見を求めることによって、実験者に問題を提起する。

AI の応用に関して言えば、GPS は、手段-目標分析の上に組み立てられる制御の戦略である。GPS は現在の状態から目標の状態まで解き進んでいく。その機構は階層的であり帰納的であり、相違を縮小する手続きを発見することによって目標を達する。GPS 自体は手続きを発見する手続きである。けれども、それはできないことを決定することはできない。また手続き間での資源の割当てやコミュニケーションはできない。

4.3 Piaget の形式的推論のモデル

Piaget の形式的推論の理論は、人間の問題解決能力を開発するためのただ一つの理論的枠組みである。Piaget の理論では、形式的操作者は命題論理によってモデル化できる論理的能力を有するものと想定する。認識の展開は、多数の明確に定義された不変の段階を通して現れる。

“形式的操作”の最終段階では、命題論理に対応する枠組みの内での問題の変数の組合せ論的分析を使って、若者でも問題を解くことができる。すなわち、組織的に変数の組合せを考えて彼は“真”の解は多くの可能性のうちただ一つの場合となるようにできる。問題解決行動についての Piaget の観察や解説は、半世紀以上もの間持ちこたえてきた。けれども、彼の方法論や、基本的認識構造を表現する手段として、形式論理を利用したことについて重大な批判を受けている。

5. 問題解決研究

本章では、これまでに述べた三つのモデルの概念を織り込み、人間の問題解決行動を分

析するための道具となるような問題解決研究の真の論点は、「人はどのように問題を解くか」であって、ある問題に対する人の答が正しいか否かを問うことではない。すなわち、問題解決の結果ではなく、過程を把握したいのである。ここで利用できる唯一のデータは、問題解決の言語表現である。けれども、これらの資料を収集する際の方法論的問題は、想像以上に大きい。そこで最初に方法論的問題をいくつかとりあげ、つづいて実際の問題解決研究を述べる。

5.1 方法論的問題

人間の問題解決に関する Newell と Simon の仕事では、実験心理学における言語プロトコルの利用を正式に採用した。それ以前は、心理学の厳密な方法に基づいていたため、実験者が直接管理できる枠外の被験者により生成されたデータなどを許容しなかった。たいていの推論過程は、実はこうした範ちゅうに属する。上記の Newell と Simon が開発した“考えを述べる”方法論では、被験者に適切な問題解決のタスクを選択させ、問題解答中に被験者に対して「考えを述べる」ように要請し、さらにテープに記録した言語に表現された命題（プロトコル）を理論モデルに応じて分析する方法で構成されている。問題解決過程では、開始指示以外には、実験者と被験者間の対話は許されていない。分析道具は GPS である。Newell と Simon の方法では、問題解決の実行がとらえられる。

一方、Piaget の方法論では、被験者は未知の問題を解明し、それらを解く方法を発明する科学者の立場に設定される。言語プロトコルは、実験者と被験者間の対話の結果であり、つまり臨床インタビュー方式である。分析道具は形式的論理操作モデルである。Piaget の方法では、問題解決の適性がとらえられる。

最大の効率を引き出すには、問題解決タスクが被験者の適性レベルに見合うものでなければならない。やさしすぎる問題の場合には、被験者が持っている戦略や発見的方法の全体を生成することはできない。また問題がむずかしすぎる場合には、被験者は推論の堂々巡りや中断に陥ってしまう。十分な協力や意欲を引き出すため、問題解決タスクは、被験者のだれにも興味深く、刺激的でなければならない。異なった問題解決の適性レベルにある被験者には複雑さのレベルが順に増大するような問題の列を解いてもらうようにすれば、これらの要請は満たされる。

次節で、専門家および非専門家による問題解決戦略の利用を比較する問題解決研究をとりあげる。Pitt と Brouwer-Janse らが開発した汎用問題解決モデルは、Newell と Simon の方法および Piaget 派のパラダイムを補完するものとされているが、これを分析道具として使用する。

5.2 問題解決タスク

ここで報告する研究では、静的あるいは動的平衡の概念に関する問題を解くときに用いる戦略や発見的方法について、大学生（非専門家）と理科系の大学教授（専門家）との間の相違を扱う。化学で出てくる標準的平衡問題から問題をとった。しかし、問題を解くには化学の事実や用語あるいは技法などの知識は不要である。

ある架空の国の政治会議における実験的な投票システムについての物語で、二つの化学反応物の間に存在する動的平衡を表現した。同会議の二つの会派である、イエロー党およびパープル党は以下のような規則の新しい投票システムを試みる。すなわち、紙の投票用紙の代わりに、無色透明の液体を用いて投票する。1滴の液体が1票を構成する。代表者達はイエロー党かパープル党のいずれかのメンバーであるものとし、自分の党派に投票するか棄権するかをどちらかを強制される。イエローおよびパープルの得票は黄色と紫色で

現れ、得票差を決定するには、色に変化するまで相手の党の投票を加えることにする。化学用語でいうと、これは中性点滴定の平衡である。黄色か紫色のいずれかを示す表示器によって、どちらが先行しているかを報告する。さて問題となるのは、投票箱の中味についての“パズル”である。被験者には次のものが与えられる。黄色の投票、紫色の投票および表示器からなる三つのビン。滴定用ピペット、および投票箱として用いる 100cc のビーカーである。図 1 に示す画面は、セッション全体を通じて問題解決者が利用できる。

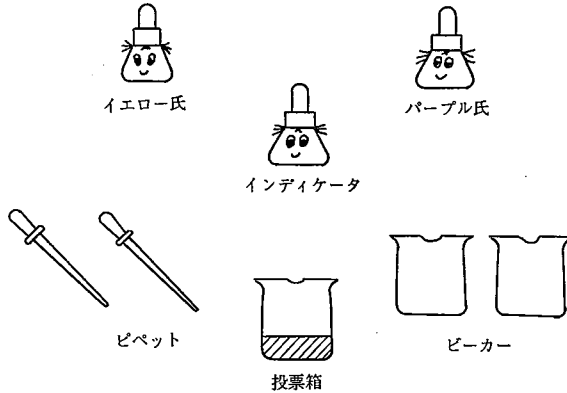


図 1 表示画面

Fig. 1 Display

表 1 問題

Table 1 Problems

問題 1 案件は公園におけるスノウモービルである。公園管理者であるイエロー氏は、公園でのスノウモービルの利用を禁止したいと考えている。一方、スノウモービルを自分でも利用するパープル氏は、その禁止は反対である。

パープル側とイエロー側は各々、同数の代議員をもっている。さてここに投票結果を入れる投票ビーカーがある。イエロー派はすべて投票したが、パープル派は何人かが投票をしなかった。ではパープル派の何人が棄権したか。

問題 2 この投票は、ガソリン不足に関するものである。イエロー氏は日曜日のドライブを禁止したいと思っている。パープル氏は禁止に反対している。誰もがこれについては強硬な意見を抱いている。そこで全員が投票する。イエロー派とパープル派はまったく同数である。

投票の途中で、イエロー派が先行したが、まだ投票していない代表が 50 人残っている時に、選挙管理人がこの問題に介入して、ビーカーをたたいて破壊した。代議員はすぐにこの事態に気付いたが、誰も彼らがビーカーを壊したのを目撃していなかった。急いで投票用ビーカーを取り換え、何事も起きなかったかのように投票を進行させた。

管理人はビーカーが壊れた時にイエロー派の半数が投票を済ませたことは覚えていたが、その騒ぎの中で、パープル派の何人が投票を済ませたかを記録し忘れた。彼らはそれを計算できると思っているのであまり気に掛けていない。彼らは、問題が白熱化しているので、誰もが投票権を行使し棄権票を投じる者がいないと見なししている。

さて、あなたが選挙管理人の一員であり、また投票の残りが入った新しいビーカーがここに置かれているものとする。ビーカーが壊れる前に何人のパープル派が投票したかを計算で求めよ。

被験者は、口に出して考えるよう要求され、かれらの自発的な口答は、テープに収録された。時間の制限はなかった。被験者が問題を終わったことを表示すると、実験者はその解が正しいか否かをフィードバックせずに次の問題に移った。すべてのセッションは、個別に行った。各参加者は5問を解いた。そのうちの2問を表1に示した。

各参加者の言語プロトコルは、24個のサブルーチンの識別ができるプロセス・モデルを使ってコード化される。

問題解決モデルでは、次の三つの異なったレベルで、“口に出して考える”プロトコルを記述し分析する。

- 1) 基本的要素プロセスないしサブルーチン (SR).
- 2) 全体のプロセス列を定める発見的ガイドラインとして、サブルーチンを機能に応じてグループ分けする。
- 3) そして、情報の処理と仮説の生成と検証の戦略として協調して働くサブルーチンをグループ分けする。

モデルで表現された基本的プロセスは、与えられた情報をリストしたり、可能な質問をリストしたり、関係のない情報を削除したりする24個のサブルーチンである。表2に、24個のSRをすべて示す。

表2 三つの発見的部分プロセスにグループ分けした問題解決モデルの構成要素 (SR)
Table 2 Components of the problem solving model (SR) grouped into 3 heuristic sub processes

問題の定義			
SR 1	情報のリスティング	SR 2	仮定のリスティング
SR 3	可能な質問のリスティング	SR 4	評価基準の選択
SR 5	優先順位の割当	SR 6	関係する情報のリスティング/無関係な情報の削除
SR 7	仮説の定式化	SR 9	関連質問の選択
SR 8	予測の定義		
データの獲得			
SR 10	初期状態の定義	SR 11	目標状態の定義
SR 12	必要なデータの決定	SR 13	利用可能なアルゴリズムの決定
SR 14	規則/方法/手続き/アルゴリズムなどの選択	SR 15	アルゴリズムの編集
		SR 16	プログラムの実行
SR 17	フィードバックの決定	SR 18	新しい情報のタグ付け
解釈			
SR 19	データの編集とコンパイル	SR 20	予測とデータの照合
SR 21	予測の真理値の決定	SR 22	データからのパターンの抽出
SR 23	関連するパターンの要約	SR 24	結論の出力

発見的部分プロセスとしてのサブルーチンのグループ化は、Polyaの発見的問題解決法を修正して適用したものである。問題解決プロセスは、以下の三つの主要な部分プロセスで構成されるものと仮定する(表2)。

- 1) 問題の定義: 予備的な限定局面
- 2) データの獲得: 必要なデータの決定とその獲得
- 3) 解釈: 情報の集積と解釈および結論

基本サブルーチンは、手段-目標アルゴリズムを使ったデータの獲得、蓄積データからのパターンや規則性の抽出、仮説の生成と検証、体系的評価、関連情報や質問や手続きやデータなどの選択など、八つの戦略に分類できる。この八つの戦略を表3に示す。

表 3 戦略：構成サブルーチンとその説明
Table 3 Strategies: Constituent subroutines and description

戦 略	サ ブ ル ー チ ン	説 明
選 択	SR6—関連情報の リスティング/無関係な情報の削除. SR9—関連する質問の選択. SR13—利用可能なアルゴリズムの決定. SR19—データの編成.	最適情報を選択し, 出す重要質問を指定し, アルゴリズムを入念に選択し, データを選択的に再編成し, 編集するプロセス.
評 価	SR2—仮定のリスティング. SR4—評価基準の選択. SR5—優先順位の割当. SR15—アルゴリズムの編集.	全体の問題解決プロセスならびにその構成要素などをモニタして再評価する連続的評価プロセス.
仮説の設定	SR7—仮説の定式化. SR8—予測の定義. SR20—データと予測の照合. SR21—予測の真値の決定.	IF-THEN 仮説とその予測の生成, および予測の確認をする.
GPS	SR10—初期状態の定義. SR11—目標状態の定義. SR12—必要なデータの確認.	手段-目標アルゴリズムを使って一連の目標を達成する.
パターン抽出	SR22—データからのパターンの抽出. SR23—関連パターンの要約.	集積したデータから関係するパターン, 対称性, 規則性, 類推性を識別する.
条 件	SR1—与えられた情報リスティング. SR3—可能な質問のリスティング. SR14—手続きの選択.	プログラム実行に必要な条件のコンパイル. たとえば, 直接利用可能な情報, 出す質問, 適用する規則など.
行 為	SR16—プログラムの実行. SR17—フィードバックの決定. SR18—新しい情報のタグ付け.	プログラムの実行と結果の記録.
結 論	SR24—結論の出力.	熟考の末に得られた, またしばしば先行プロセスの要約を含む, 問題解決プロセスの結果あるいは成果.

表 2 に示した 24 個のサブルーチンは, 各プロトコルの中の問題に関連した命題をコード化するのに使われる. 全体の処理は, 少なくとも一度は使われたサブルーチン全体の集合で計られる. 発見的処理は, 三つのそれぞれの発見的部分プロセスの中で使われたサブルーチンの集合で計られる. また戦略の処理は, 八つの各戦略内で使われたサブルーチンの集合で計られる.

5.3 結 果

ほぼすべての被験者が, 問題 1 を解くことができた. 彼らは, ほぼ同数のサブルーチンを使用した (30%, および 33%). また命題の個数で計ったプロトコルの長さも同様であった (表 4 参照). この問題は明らかに実質的な処理を必要としなかった. 問題 2 を解く場合を比較してみると, サブルーチンの 47% および 69% がそれぞれ大学生および教授によって使用された. 教授の 75% および大学生の 44% だけが問題 2 を正しく解いた. どの被験者も最初のタスクよりも第 2 のタスクの方が余計に時間を要した. また問題 2 を解く場合に教授は学生よりも多くの時間を掛けた.

問題 2 を解くのに, 教授達は学生達よりもはるかに多く三つの発見的部分プロセスをすべて使用している (表 5). すなわち, 仕事の複雑さが増大して, 必要な場面では, 教授達は問題をより良く定義し, データをあますところなく獲得し, また, 彼らの結果の解釈は完全であった.

戦略の利用における差は複雑な問題 2 の場合だけに現れている (表 6). そして選択, 評価の戦略, GPS, 仮説の設定の戦略の場合に差が顕著である.

教授達は, 学生よりもはるかに多くこれらの戦略を利用した. さらに, 比較的簡単な問

表 4 プロトコル・データ

Table 4 Protocol data

	問 題 1		問 題 2	
	大学生	教 授	大学生	教 授
平均年齢	20	39	20	39
被験者数	16	16	16	16
正解率	90	96	44	75
消費時間	17.4	17.8	57.2	79.2
サブルーチン使用率	30	33	47	69

表 5 発見的部分プロセスとしてのサブルーチンの使用率

Table 5 Heuristic processing—Proportional use of SR's within subprocess

	問 題 1		問 題 2	
	大学生	教 授	大学生	教 授
問題の定義 (SR 1~9)	0.19	0.26	0.43	0.62
データの獲得 (SR 10~18)	0.38	0.41	0.47	0.77
解釈 (SR 19~24)	0.22	0.18	0.48	0.66

表 6 戦略の使用——戦略としてのサブルーチンの使用率

Table 6 Use of strategies—Proportional use of SR's within strategies

戦 略	問 題 1		問 題 2	
	大学生	教 授	大学生	教 授
選 択	0.29	0.31	0.61	0.86
評 価	0.13	0.18	0.26	0.61
仮説の設定	0.00	0.02	0.06	0.34
GPS	0.03	0.06	0.21	0.75
パターン抽出	0.08	0.00	0.75	0.69
条 件	0.43	0.53	0.66	0.68
行 為	0.84	0.82	0.86	1.00
結 論	0.89	0.86	0.69	1.00

問題1ではもっぱら条件および行為を用いて解き、結論を出した。これらの戦略を使うだけでは、問題2を解くのに、明らかに不十分である。選択の戦略の利用が増えたこと以外に、たいていの学生は、問題を解くのに、条件、行為、結論、パターン抽出の戦略を使っている。

6. お わ り に

行為の戦略は、すべての問題においてすべての被験者によって最もよく使用された。条件および行為の戦略と一緒に使う人間の戦略は、AI システムにおけるプロダクション・ルールの利用に対応させることができる。プロダクション・ルールまたは条件/行為戦略の利用は、簡単な問題を解くには十分であり、非専門家の解答者でもこれらの手法は容易に利用できたのは、まったく自然のことである。選択、評価、GPS、仮説の設定などの戦略は、問題解決プロセス内では制御の機構として働く。つまり、これらの戦略は問題解決プロセスの時、場所、方法、理由などを決定する。これらは人間の専門家が複雑な問題を解く場合に広く利用されるが、非専門家の問題解決者には容易には利用されない。

データの解釈では、二つの論点を考える必要がある。第1は、これらの結果のうちどれだけが認識の展開の差によるかであり、第2は、これらの結果のうちどれだけが学習/経験の差によるか、である。ところで、補足的な研究で、化学および心理学の、二つの異なる専門領域の平均年齢24歳の大学院生に、(基礎化学から取った)同じ一連の問題を解かせてみた。そのデータの示すところでは、大学院生は、選択と評価の戦略の使用については大学生の方に近く、GPSの戦略を利用する点については教授と大学生の中間に位置している。化学系大学院生は、心理学系大学院生よりもかなり広範にGPSの戦略を活用した。後者は、パターン抽出の戦略の方をより利用した。化学系学生の50%および心理学系学生の44%だけが、問題2を正しく解いた。認識の展開因子は、少なくともこのタスクの場合には、領域訓練因子よりも大きな効果を持っているようである。

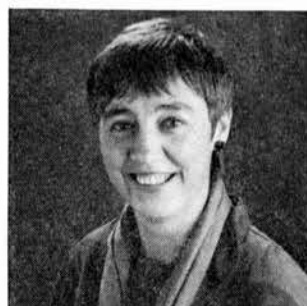
人間が一定の問題解決戦略を利用する方法については経験的に判断できるが、一般的问题解決の知識と専門分野の知識の切り離しについては、問題が残されている。というのも、いかなる種類のものであれ、知識は静的な対象ではないからである。知識工学エンジニアに荷せられた実際の問題解決タスクは、適正な時期に適正な条件の下で動的な人間の問題解決システムを開拓することであり、またその成長するシステムの有益な部分をコンピュータ・プログラムに取り込むことである。この問題解決タスクに取り組む場合には、知識工学エンジニアは独自の問題解決戦略を広く活用する必要がある。選択、評価、GPSなどの戦略の明らかな知識は、この過程を確実に促進することになる。

(技術研究部 山田真市 訳)

- 参考文献 [1] M. A. Boden, *Artificial Intelligence and Natural Man*, Basic Books, New York, 1977.
 [2] P. H. Lindsay, D. A. Norman, *Human Information Processing, An Introduction to Psychology*, Academic Press, New York, 1977.
 [3] A. Barr, E. A. Feigenbaum (Eds.), *The Handbook of Artificial Intelligence*, Vol. I to III, William Kaufman, Inc., Los Altos, Cal., 1981.
 [4] A. Newell, J. C. Shaw, "The Process of Creative Thinking", In: H. A. Simon (Ed.), *Models of Thought*, Yale University Press, New Haven, 1979.
 [5] P. N. Johnson-Laird, P. C. Wason (Eds.), *Thinking, Readings in Cognitive Science*, Cambridge University Press, Cambridge, 1977.
 [6] D. E. Rumelhart, *Introduction to Human Information Processing*, John Wiley & Sons, Inc., New York, 1977.
 [7] P. H. Winston, *Artificial Intelligence*, Addison-Wesley Publ., Reading, Mass., 1984.
 [8] A. Newell, H. A. Simon, *Human Problem Solving*, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1972.
 [9] G. Polya, *How to Solve It*, Doubleday-Anchor, Garden City, L. I., 1957.

執筆者紹介 Maddy D. Brouwer-Janse

1967年にオランダの Wageningen 農芸大学から食品科学・工学の博士号を修得。同年より1977年まで、Amsterdamの中等学校の化学科の筆頭教師を勤める。1977年より1983年の間、MinneapolisのMinnesota大学において研究および教育の助手を勤めるとともに、同大学のHuman Learning研究センターの一員となる。1983年Minnesota大学から認知心理学のPh. Dを得る。1984年Sperry社に入社、現在、システム・プロダクト部門のアドバンスド・アプリケーションの研究員として、問題解決の分野の研究に従事。AAAI, ACL, CSS, Sigma Xiに所属。



論文

変分原理による軸対称殻弾性方程式の導入

Introduction of Axisymmetric Shell Elastic Equation
by the Variational Principle

藤野 勉

要 約 任意形状の弾性体について、その荷重による変形を支配する弾性方程式を導入するためには、一般に (a) ベクトル(微分方程式)法と (b) エネルギー(変分原理)法とがあるが、形状の複雑化と共に後者のほうが容易である。本稿は任意形状の軸対称殻について変分原理によって支配方程式の導入を行ったもので、さらに差分法によるその数値解析法について述べ、その一つの例として円環殻をとりあげた。

Abstract For the introduction of elastic equation about arbitrary shaped elastic body, there are two methods, namely (a) vector (differential equation) method and (b) energy (variational principle) method. In general, the later is easier than the former with complexity of elastic body shape. This paper is concerned with the introduction of governing equation about arbitrary shaped axisymmetric shell by the variational principle and numerical analysis method by the difference equation method is also presented. For one example, circular ring shell is treated.

1. はじめに

一般に特定の形状をもつ弾性体の変形に関する支配方程式は、従来では力、モーメントのベクトル的な平衡条件によって求められているが、ここでは変分原理による導入を試み、その対象を軸対称殻とした。これら二つの方法を比較するとき、形状が複雑となると共に前者よりも後者のほうが求めやすくなる。とくに高い精度の弾性方程式を必要とするとき力およびモーメント以上の力学的概念を必要とすることとなり、前者による導入は困難となる。

いま直角座標系 x, y, z 空間内に存在する弾性体において、その x, y, z 方向変位を u, v, w とするとき、その歪は

$$\varepsilon_{xy} = u_x, \varepsilon_{yy} = v_y, \varepsilon_{zz} = w_z, \varepsilon_{xy} = u_y + v_x, \varepsilon_{xx} = u_x + w_x, \varepsilon_{yz} = v_z + w_y \quad (1-1)$$

応力は

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \end{Bmatrix} = E^* \begin{bmatrix} 1-\nu & \nu & \nu \\ \nu & 1-\nu & \nu \\ \nu & \nu & 1-\nu \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \end{Bmatrix}, \quad \begin{Bmatrix} \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yz} \end{Bmatrix} = G \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \begin{Bmatrix} \varepsilon_{xy} \\ \varepsilon_{xz} \\ \varepsilon_{yz} \end{Bmatrix} \quad (1-2)$$

によって定義される。ただし、 E =Young 率、 G =剛性率、 ν =Poisson 比として $E^* = E/(1+\nu)(1-2\nu)$ 、 $G = E/2(1+\nu)$ である。

いま x, y, z 方向の力を X, Y, Z とするとき、弾性体の内部では

$$\left. \begin{aligned} L_x + X &\equiv \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + X = 0 \\ L_y + Y &\equiv \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} + Y = 0 \\ L_z + Z &\equiv \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + Z = 0 \end{aligned} \right\} \quad (1-3)$$

固定(幾何学的)境界では

$$u = \bar{u}, \quad v = \bar{v}, \quad w = \bar{w} \tag{1-4}$$

自由(力学的)境界では

$$\left. \begin{aligned} \sigma_x &\equiv l_x \sigma_{xx} + l_y \sigma_{xy} + l_z \sigma_{xz} = \bar{\sigma}_x \\ \sigma_y &\equiv l_x \sigma_{xy} + l_y \sigma_{yy} + l_z \sigma_{yz} = \bar{\sigma}_y \\ \sigma_z &\equiv l_x \sigma_{xz} + l_y \sigma_{yz} + l_z \sigma_{zz} = \bar{\sigma}_z \end{aligned} \right\} \tag{1-5}$$

が満たされる。ここに l_x, l_y, l_z は表面における外方法線の方向余弦である。
ここで下記 Green 積分を行う。

$$\begin{aligned} &\iiint_V \{ (L_x + X)\delta u + (L_y + Y)\delta v + (L_z + Z)\delta w \} dx dy dz \\ &- \iint_S \{ (\sigma_x - \bar{\sigma}_x)\delta u + (\sigma_y - \bar{\sigma}_y)\delta v + (\sigma_z - \bar{\sigma}_z)\delta w \} dS = \delta(U - W) = 0 \end{aligned} \tag{1-6}$$

$$\delta U = \iiint_V (\sigma_{xx} \delta \varepsilon_{xx} + \sigma_{yy} \delta \varepsilon_{yy} + \sigma_{zz} \delta \varepsilon_{zz} + \sigma_{xy} \delta \varepsilon_{xy} + \sigma_{xz} \delta \varepsilon_{xz} + \sigma_{yz} \delta \varepsilon_{yz}) dx dy dz \tag{1-7}$$

$$W = \iiint_V (Xu + Yv + Zw) dx dy dz + \iint_S (\bar{\sigma}_x u + \bar{\sigma}_y v + \bar{\sigma}_z w) dS \tag{1-8}$$

ここに, U は内部エネルギー, W は外部エネルギーを表す。式(1-6)は変分原理を表し, 式(1-7)は完全変分で次のように表される。

$$\begin{aligned} U = &\frac{1}{2} \iiint_V E^* \left[(1-\nu) \left\{ \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2 \right\} \right. \\ &\left. + 2\nu \left(\frac{\partial u}{\partial x} \frac{\partial v}{\partial y} + \frac{\partial u}{\partial x} \frac{\partial w}{\partial z} + \frac{\partial v}{\partial y} \frac{\partial w}{\partial z} \right) \right] dx dy dz \\ &+ \frac{1}{2} \iiint_V G \left\{ \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)^2 \right\} dx dy dz \end{aligned} \tag{1-9}$$

以上述べたことは可逆的拘束を与えた弾性体(保存系)についても, 一般に成立する原則で変分原理による支配方程式の導入の基礎原理となっている。

2. 軸対称殻の弾性方程式

ここでは対称軸を x とし, これに直交して y, z 軸を定め, 軸対称殻中性面の母線に沿って曲線座標 s , さらに s と x の挟む角を φ , 母線 s の曲率半径を ρ とする。

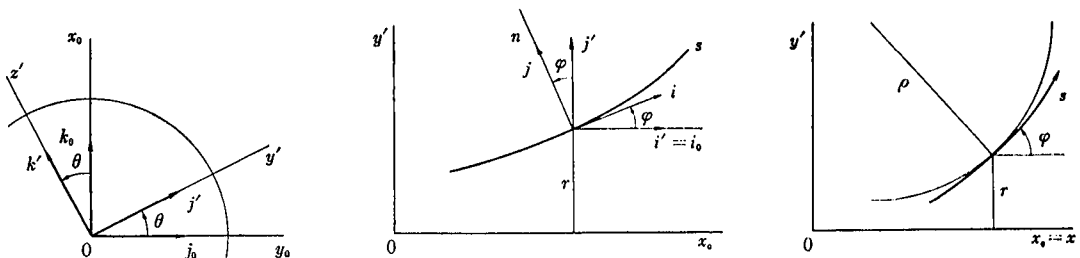


図 1 軸対称殻一般図
Fig. 1 General view of axisymmetric shell

2.1 軸対称殻の微分幾何

つぎに, 歪を定義するための標準微長を求める。その準備として単位ベクトルの変換式ならびにその微分式を示す。

$$\begin{cases} i' \\ j' \\ k' \end{cases} = \begin{bmatrix} 1 & & \\ & \cos \theta & \sin \theta \\ & -\sin \theta & \cos \theta \end{bmatrix} \begin{cases} i_0 \\ j_0 \\ k_0 \end{cases} \quad (2-1)$$

$$\begin{cases} i \\ j \\ k \end{cases} = \begin{bmatrix} \cos \varphi & \sin \varphi & \\ -\sin \varphi & \cos \varphi & \\ & & 1 \end{bmatrix} \begin{cases} i' \\ j' \\ k' \end{cases} = \begin{bmatrix} \cos \varphi & \cos \theta \sin \varphi & \sin \theta \sin \varphi \\ -\sin \varphi & \cos \theta \cos \varphi & \sin \theta \cos \varphi \\ & -\sin \theta & \cos \theta \end{bmatrix} \begin{cases} i_0 \\ j_0 \\ k_0 \end{cases} \quad (2-2)$$

$$\begin{cases} i' \\ j' \\ k' \end{cases} = \begin{bmatrix} \cos \varphi & -\sin \varphi & \\ \sin \varphi & \cos \varphi & \\ & & 1 \end{bmatrix} \begin{cases} i \\ j \\ k \end{cases} \quad (2-3)$$

$$di' = 0, \quad dj' = k'd\theta = kd\theta, \quad dk' = -j'd\theta = -(i\sin\varphi + j\cos\varphi)d\theta \quad (2-4)$$

$$di = jd\varphi + k\sin\varphi d\theta, \quad dj = -id\varphi + k\cos\varphi d\theta, \quad dk = -(i\sin\varphi + j\cos\varphi)d\theta \quad (2-5)$$

2.2 位置ベクトルならびにその微分

0 を原点とし, i_0, j' を含む平面内にある殻上の 1 点 A を端点とする位置ベクトルを

$$\mathbf{r} = i_0 x + j' r + j n \quad (2-6)$$

によって定義するとき, その微分は

$$\begin{aligned} d\mathbf{r} &= i_0 dx + j' dr + k r d\theta + j dn + n(-id\varphi + k\cos\varphi d\theta) \\ &= i \left(1 - \frac{n}{\rho}\right) ds + j dn + k \left(1 + \cos\varphi \frac{n}{r}\right) dt \doteq ids + j dn + k dt \end{aligned} \quad (2-7)$$

である. ここに $dt = r d\theta$ は円周方向微長を表す. なお曲率半径は,

$$\begin{aligned} \rho d\varphi &= ds, \quad \cos\varphi = dx/ds, \quad \sin\varphi = dr/ds \\ \tan\varphi &= \frac{dr}{dx}, \quad \sec^2\varphi d\varphi = \frac{d^2 r}{dx^2} \frac{dx}{ds} ds, \quad \rho = \frac{(ds/dx)^3}{d^2 r/dx^2} \end{aligned} \quad (2-8)$$

によって求められる.

なお x, r が変数 ϕ の関数として表されているときは

$$dr/dx = r'/x', \quad d^2 r/dx^2 = (r''x' - r'x'')/(x')^3, \quad \rho = \{(x')^2 + (r')^2\}^{3/2}/(r''x' - r'x'') \quad (2-9)$$

$$ds = g d\phi, \quad g = \sqrt{(x')^2 + (r')^2} \quad (2-10)$$

によって与えられる. ここに g は測度係数を表す. なお $'$ は ϕ による微分を表す.

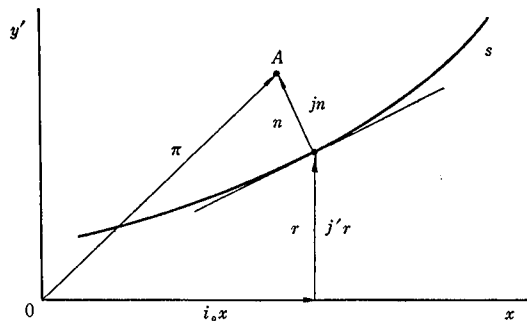


図 2 位置ベクトル
Fig. 2 Position vector

2.3 変位ベクトルならびにその微分

殻上の 1 点 A における i, j, n 方向変位を u, v, w とし, 変位ベクトルを

$$\mathbf{u}(\phi, n, \theta) = iu(\phi, n, \theta) + jv(\phi, n, \theta) + kw(\phi, n, \theta) \quad (2-11)$$

によって定義する. これら変位成分は

$$u(\phi, n, \theta) = u(\phi, \theta) + nu'(\phi, \theta) + n^2u''(\phi, \theta) + \dots \quad (2-12)$$

により n のべき級数によって展開されるものとする. v, w についても同様である.

$$\begin{aligned} du &= i\{u_\phi d\phi + u' dn + u_\theta d\theta + n(u_\phi' d\phi + 2u'' dn + u_\theta' d\theta) + \dots\} \\ &\quad + (j d\phi + k \sin \varphi d\theta)(u + nu' + \dots) \\ &\quad + j\{v_\phi d\phi + v' dn + v_\theta d\theta + n(v_\phi' d\phi + 2v'' dn + v_\theta' d\theta) + \dots\} \\ &\quad + (-i d\phi + k \cos \varphi d\theta)(v + nv' + \dots) \\ &\quad + k\{w_\phi d\phi + w' dn + w_\theta d\theta + n(w_\phi' d\phi + 2w'' dn + w_\theta' d\theta) + \dots\} \\ &\quad - (i \sin \varphi + j \cos \varphi) d\theta(w + nw' + \dots) \\ &= i\{e_{ss} ds + e_{sn} dn + e_{st} dt + n(e_{ss}' ds + e_{sn}' dn + e_{st}' dt) + \dots\} \\ &\quad + j\{e_{ns} ds + e_{un} dn + e_{nt} dt + n(e_{ns}' ds + e_{un}' dn + e_{nt}' dt) + \dots\} \\ &\quad + k\{e_{ts} ds + e_{tn} dn + e_{tt} dt + n(e_{ts}' ds + e_{tn}' dn + e_{tt}' dt) + \dots\} \end{aligned} \quad (2-13)$$

上式より, 下記歪テンソルが導かれる.

$$\left. \begin{aligned} \epsilon_{ss} &= e_{ss} = \frac{\partial u}{g \partial \phi} - \frac{v}{\rho} & \epsilon_{ss}' &= e_{ss}' = \frac{\partial u'}{g \partial \phi} - \frac{v'}{\rho} \\ \epsilon_{nn} &= e_{nn} = v' & \epsilon_{nn}' &= e_{nn}' = 2v'' \\ \epsilon_{tt} &= e_{tt} = \frac{1}{r} \left(\frac{\partial w}{\partial \theta} + \sin \varphi u + \cos \varphi v \right) & \epsilon_{tt}' &= e_{tt}' = \frac{1}{r} \left(\frac{\partial w'}{\partial \theta} + \sin \varphi n' + \cos \varphi v' \right) \\ \epsilon_{sn} &= e_{sn} + e_{ns} = \frac{\partial v}{g \partial \phi} + \frac{u}{\rho} + u' & \epsilon_{sn}' &= e_{sn}' + e_{ns}' = \frac{\partial v'}{g \partial \phi} + \frac{u'}{\rho} + 2u'' \\ \epsilon_{st} &= e_{st} + e_{ts} = \frac{1}{r} \left(\frac{\partial u}{\partial \theta} - \sin \varphi w \right) + \frac{\partial w}{g \partial \phi} & \epsilon_{st}' &= e_{st}' + e_{ts}' = \frac{1}{r} \left(\frac{\partial u'}{\partial \theta} - \sin \varphi w' \right) + \frac{\partial w'}{g \partial \phi} \\ \epsilon_{nt} &= e_{nt} + e_{tn} = \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \varphi w \right) + w' & \epsilon_{nt}' &= e_{nt}' + e_{tn}' = \frac{1}{r} \left(\frac{\partial v'}{\partial \theta} - \cos \varphi w' \right) + 2w'' \end{aligned} \right\} \quad (2-14)$$

応力テンソルは構成方程式(1-2)によって与えられる. 微視的には, s, n, t は x, y, z と同様に直交直線座標系とみなされるので, 同じ構成方程式が満たされる.

2.4 弾性方程式の導入

以上によって歪, 応力テンソルが求められたので, 前述の変分原理によって弾性方程式, 境界式が導かれる.

$$\begin{aligned} \delta U &= \iint_{-h/2}^{h/2} (\sigma_{mn} + n\sigma_{mn}' + \dots)(\delta \epsilon_{mn} + n\delta \epsilon_{mn}' + \dots) ds dn dt \\ &= \iint \left\{ N_{ss} \left(\frac{\partial \delta u}{g \partial \phi} - \frac{\delta v}{\rho} \right) + N_{nn} \delta v' + \frac{N_{tt}}{r} \left(\frac{\partial \delta w}{\partial \theta} + \sin \varphi \delta u + \cos \varphi \delta v \right) \right. \\ &\quad + N_{sn} \left(\frac{\partial \delta v}{g \partial \phi} + \frac{\delta u}{\rho} + \delta u' \right) + N_{st} \left(\frac{\partial \delta u}{r \partial \theta} - \sin \varphi \frac{\delta w}{r} + \frac{\partial \delta w}{g \partial \phi} \right) \\ &\quad + N_{tn} \left(\frac{\partial \delta v}{r \partial \theta} - \cos \varphi \frac{\delta w}{r} + \delta w' \right) + M_{ss} \left(\frac{\partial \delta u'}{g \partial \phi} - \frac{\delta u'}{\rho} \right) + 2M_{nm} \delta v'' \\ &\quad + \frac{M_{tt}}{r} \left(\frac{\partial \delta w'}{\partial \theta} + \sin \varphi \delta u' + \cos \varphi \delta v' \right) + M_{sn} \left(\frac{\partial \delta v'}{g \partial \phi} + \frac{\delta u'}{\rho} + 2\delta u'' \right) \\ &\quad \left. + M_{st} \left(\frac{\partial \delta u'}{r \partial \theta} - \sin \varphi \frac{\delta w'}{r} + \frac{\partial \delta w'}{g \partial \phi} \right) + M_{tn} \left(\frac{\partial \delta v'}{r \partial \theta} - \cos \varphi \frac{\delta w'}{r} + 2\delta w'' \right) + \dots \right\} g r d\phi d\theta \\ &= \int (N_s \delta u + N_n \delta v + N_t \delta w + M_s \delta u' + M_n \delta v' + M_t \delta w' + \dots) dl \\ &\quad - \iint (L_s \delta u + L_n \delta v + L_t \delta w + L_s' \delta u' + L_n' \delta v' + L_t' \delta w' + \dots) ds dt \end{aligned} \quad (2-15)$$

ただし, $k^* = E^*h$, $k' = Gh$, $D^* = E^*h^3/12$, $D' = Gh^3/12$, $h =$ 殻の板厚, であり

$$\left. \begin{aligned} N_{ss} &= k^* \{ (1-\nu)\varepsilon_{ss} + \nu\varepsilon_{nn} + \nu\varepsilon_{tt} \}, & N_{nn} &= k^* \{ \nu\varepsilon_{ss} + (1-\nu)\varepsilon_{nn} + \nu\varepsilon_{tt} \}, \\ N_{tt} &= k^* \{ \nu\varepsilon_{ss} + \nu\varepsilon_{nn} + (1-\nu)\varepsilon_{tt} \} \\ N_{sn} &= k' \varepsilon_{sn}, & N_{st} &= k' \varepsilon_{st}, & N_{nt} &= k' \varepsilon_{nt} \\ M_{ss} &= D^* \{ (1-\nu)\varepsilon'_{ss} + \nu\varepsilon'_{nn} + \nu\varepsilon'_{tt} \}, & M_{nn} &= D^* \{ \nu\varepsilon'_{ss} + (1-\nu)\varepsilon'_{nn} + \nu\varepsilon'_{tt} \}, \\ M_{tt} &= D^* \{ \nu\varepsilon'_{ss} + \nu\varepsilon'_{nn} + (1-\nu)\varepsilon'_{tt} \} \\ M_{sn} &= D' \varepsilon'_{sn} & M_{st} &= D' \varepsilon'_{st} & M_{nt} &= D' \varepsilon'_{nt} \end{aligned} \right\} \quad (2-16)$$

である。

2.5 境界関数

s, t 展開面における境界の外方方向余弦を l_s, l_t として, 境界関数は次のように表す。

$$\left. \begin{aligned} N_s &= l_s N_{ss} + l_t N_{st}, & N_n &= l_s n_{sn} + l_t N_{tn}, & N_t &= l_s N_{st} + l_t N_{tt} \\ M_s &= l_s M_{ss} + l_t M_{st}, & M_n &= l_s M_{sn} + l_t M_{tn}, & M_t &= l_s M_{st} + l_t M_{tt} \end{aligned} \right\} \quad (2-17)$$

2.6 微分式

$$L_s = \frac{\partial N_{ss}}{\partial \phi} + \frac{\partial N_{st}}{r \partial \theta} + \frac{\sin \varphi}{r} (N_{ss} - N_{tt}) - \frac{N_{sn}}{\rho} \quad (2-18)$$

$$L_n = \frac{\partial N_{sn}}{\partial \phi} + \frac{\partial N_{tn}}{r \partial \theta} + \frac{1}{r} (\sin \varphi N_{sn} - \cos \varphi N_{tn}) + \frac{N_{ss}}{\rho} \quad (2-19)$$

$$L_t = \frac{\partial N_{st}}{\partial \phi} + \frac{\partial N_{tt}}{r \partial \theta} + \frac{1}{r} (2 \sin \varphi N_{st} + \cos \varphi N_{tn}) \quad (2-20)$$

$$L'_s = \frac{\partial M_{ss}}{\partial \phi} + \frac{\partial M_{st}}{r \partial \theta} + \left(\frac{\sin \varphi}{r} \right) (M_{ss} - M_{tt}) - \frac{M_{sn}}{\rho} - N_{sn} \quad (2-21)$$

$$L'_n = \frac{\partial M_{sn}}{\partial \phi} + \frac{\partial M_{tn}}{r \partial \theta} + \frac{1}{r} (\sin \varphi M_{sn} - \cos \varphi M_{tn}) + \frac{M_{ss}}{\rho} - N_{nn} \quad (2-22)$$

$$L'_t = \frac{\partial M_{st}}{\partial \phi} + \frac{\partial M_{tt}}{r \partial \theta} + \frac{1}{r} (2 \sin \varphi M_{st} + \cos \varphi M_{tn}) - N_{tn} \quad (2-23)$$

つきに変分原理によって平衡方程式, 境界条件式を導くために外力を次のように定める。

$$\left. \begin{aligned} \text{境界力, モーメント} & \quad i\bar{N}_s + j\bar{N}_n + k\bar{N}_t, \quad i\bar{M}_s + j\bar{M}_n + k\bar{M}_t \\ \text{殻内力} & \quad iS + jN + kT \\ \text{表面力} & \quad \text{外面: } ip_s^0 + jp_n^0 + kp_t^0, \quad \text{内面: } ip_s^i + jp_n^i + kp_t^i \end{aligned} \right\} \quad (2-24)$$

これら外力による外部エネルギーは

$$\begin{aligned} W &= \int_l (\bar{N}_s u + \bar{N}_n v + \bar{N}_t w + \bar{M}_s u' + \bar{M}_n v' + \bar{M}_t w') dl \\ &+ \iint_s \{ Su + Nv + Tw + p_s^0(u + hu'/2) + p_s^i(u - hu'/2) \\ &+ p_n^0(v + hv'/2) + p_n^i(v - hv'/2) + p_t^0(w + hw'/2) \\ &+ p_t^i(w - hw'/2) \} ds dt \end{aligned} \quad (2-25)$$

によって与えられ, 変分原理(1-6)によって

$$\begin{aligned} \delta U - \delta W &= \int_l \{ (N_s - \bar{N}_s) \delta u + (N_n - \bar{N}_n) \delta v + (N_t - \bar{N}_t) \delta w \\ &+ (M_s - \bar{M}_s) \delta u' + (M_n - \bar{M}_n) \delta v' + (M_t - \bar{M}_t) \delta w' \} dl \\ &- \iint_s \{ (L_s + f_s) \delta u + (L_n + f_n) \delta v + (L_t + f_t) \delta w \\ &+ (L'_s + f'_s) \delta u' + (L'_n + f'_n) \delta v' + (L'_t + f'_t) \delta w' \} ds dt = 0 \end{aligned} \quad (2-26)$$

が導れ、さらに下記境界条件式、平衡方程式が求められる。

$$\text{境界条件式} \quad \begin{cases} \text{固定} & u=\bar{u}, \quad v=\bar{v}, \quad w=\bar{w}, \quad u'=\bar{u}', \quad v'=\bar{v}', \quad w'=\bar{w}' \\ \text{自由} & N_s=\bar{N}_s, \quad N_n=\bar{N}_n, \quad N_t=\bar{N}_t, \quad M_s=\bar{M}_s, \quad M_n=\bar{M}_n, \quad M_t=\bar{M}_t \end{cases} \quad (2-27)$$

$$\text{平衡方程式} \quad \begin{aligned} L_s+f_s=0, \quad L_n+f_n=0, \quad L_t+f_t=0, \quad L_s'+f_s'=0, \\ L_n'+f_n'=0, \quad L_t'+f_t'=0 \end{aligned} \quad (2-28)$$

ただし

$$\begin{aligned} f_s=S+p_s^0+p_s^i, \quad f_n=N+p_n^0+p_n^i, \quad f_t=T+p_t^0+p_t^i, \\ f_s'=(h/2)(p_s^0-p_s^i), \quad f_n'=(h/2)(p_n^0-p_n^i), \quad f_t'=(h/2)(p_t^0-p_t^i) \end{aligned} \quad (2-29)$$

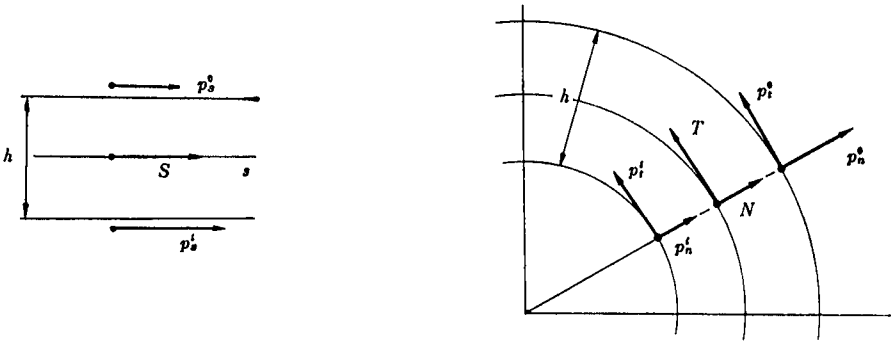


図 3 殻内力、表面力
Fig. 3 Internal force, surface force

以上によって、平衡方程式は変数を u, v, w, u', v', w' の 6 個として、次のように表される。

$$\begin{aligned} L_s+f_s &= \frac{\partial}{\partial \phi} k^* \left\{ (1-\nu) \left(\frac{\partial u}{\partial \phi} - \frac{v}{\rho} \right) + \nu v' + \frac{\nu}{r} \left(\frac{\partial w}{\partial \theta} + \sin \varphi u + \cos \varphi v \right) \right\} \\ &+ \frac{\partial}{r \partial \theta} k' \left\{ \frac{1}{r} \left(\frac{\partial u}{\partial \theta} - \sin \varphi w \right) + \frac{\partial w}{\partial \phi} \right\} + \sin \varphi (1-2\nu) \frac{k^*}{r} \left\{ \frac{\partial u}{\partial \phi} - \frac{v}{\rho} \right. \\ &\left. - \frac{1}{r} \left(\frac{\partial w}{\partial \theta} + \sin \varphi u + \cos \varphi v \right) \right\} - \frac{k'}{\rho} \left(\frac{\partial v}{\partial \phi} + \frac{u}{\rho} + u' \right) + f_s = 0 \end{aligned} \quad (2-30)$$

$$\begin{aligned} L_n+f_n &= \frac{\partial}{\partial \phi} k' \left(\frac{\partial v}{\partial \phi} + \frac{u}{\rho} = u' \right) + \frac{\partial}{r \partial \theta} k' \left\{ \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \varphi w \right) + w' \right\} \\ &+ \frac{1}{r} \left[\sin \varphi k' \left(\frac{\partial v}{\partial \phi} + \frac{u}{\rho} + u' \right) - \cos \varphi k^* \left\{ \nu \left(\frac{\partial u}{\partial \phi} - \frac{v}{\rho} \right) + \nu v' \right. \right. \\ &\left. \left. + \frac{1-\nu}{r} \left(\frac{\partial w}{\partial \theta} + \sin \varphi u + \cos \varphi v \right) \right\} \right] + \frac{k^*}{\rho} \left\{ (1-\nu) \left(\frac{\partial u}{\partial \phi} - \frac{v}{\rho} \right) \right. \\ &\left. + \nu v' + \frac{\nu}{r} \left(\frac{\partial w}{\partial \theta} + \sin \varphi u + \cos \varphi v \right) \right\} + f_n = 0 \end{aligned} \quad (2-31)$$

$$\begin{aligned} L_t+f_t &= \frac{\partial}{\partial \phi} k' \left\{ \frac{1}{r} \left(\frac{\partial u}{\partial \theta} - \sin \varphi w \right) + \frac{\partial w}{\partial \phi} \right\} + \frac{\partial}{r \partial \theta} k^* \left\{ \nu \left(\frac{\partial u}{\partial \phi} - \frac{v}{\rho} \right) + \nu v' \right. \\ &\left. + \frac{1-\nu}{r} \left(\frac{\partial w}{\partial \theta} + \sin \varphi u + \cos \varphi v \right) \right\} + \frac{1}{r} \left[2 \sin \varphi k' \left\{ \frac{1}{r} \left(\frac{\partial u}{\partial \theta} - \sin \varphi w \right) + \frac{\partial w}{\partial \phi} \right\} \right. \\ &\left. + \cos \varphi k' \left\{ \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \varphi w \right) + w' \right\} \right] + f_t = 0 \end{aligned} \quad (2-32)$$

$$\begin{aligned}
L_s' + f_s' &= \frac{\partial}{g\partial\phi} D^* \left\{ (1-\nu) \left(\frac{\partial u'}{g\partial\phi} - \frac{v'}{\rho} \right) + \frac{\nu}{r} \left(\frac{\partial w'}{\partial\theta} + \sin\phi u' + \cos\phi v' \right) \right\} \\
&\quad + \frac{\partial}{r\partial\theta} D' \left\{ \frac{1}{r} \left(\frac{\partial u'}{\partial\theta} - \sin\phi w' \right) + \frac{\partial w'}{g\partial\phi} \right\} + \sin\phi (1-2\nu) \frac{D^* \left\{ \frac{\partial u'}{g\partial\phi} - \frac{v'}{\rho} \right\}}{r} \\
&\quad - \frac{1}{r} \left(\frac{\partial w'}{\partial\theta} + \sin\phi u' + \cos\phi v' \right) \left\} - \frac{D'}{\rho} \left(\frac{\partial v'}{g\partial\phi} + \frac{u'}{\rho} \right) \\
&\quad - k' \left(\frac{\partial v}{g\partial\phi} + \frac{u}{\rho} + u' \right) + f_s' = 0
\end{aligned} \tag{2-33}$$

$$\begin{aligned}
L_n' + f_n' &= -\frac{\partial}{g\partial\phi} D' \left(\frac{\partial v'}{g\partial\phi} + \frac{u'}{\rho} \right) + \frac{\partial}{r\partial\theta} \frac{D'}{r} \left(\frac{\partial v'}{\partial\theta} - \cos\phi w' \right) + \frac{1}{r} \left[\sin\phi D' \left(\frac{\partial v'}{g\partial\phi} + \frac{u'}{\rho} \right) \right. \\
&\quad \left. - \cos\phi D^* \left\{ \nu \left(\frac{\partial u'}{g\partial\phi} - \frac{v'}{\rho} \right) + \frac{1-\nu}{r} \left(\frac{\partial w'}{\partial\theta} + \sin\phi u' + \cos\phi v' \right) \right\} \right] \\
&\quad + \frac{D^*}{\rho} \left\{ (1-\nu) \left(\frac{\partial u'}{g\partial\phi} - \frac{v'}{\rho} \right) + \frac{\nu}{r} \left(\frac{\partial w'}{\partial\theta} + \sin\phi u' + \cos\phi v' \right) \right\} \\
&\quad - k^* \left\{ \nu \left(\frac{\partial u}{g\partial\phi} - \frac{v}{\rho} \right) + (1-\nu)v' + \frac{\nu}{r} \left(\frac{\partial w}{\partial\theta} + \sin\phi u + \cos\phi v \right) \right\} + f_n' = 0
\end{aligned} \tag{2-34}$$

$$\begin{aligned}
L_t' + f_t' &= \frac{\partial}{g\partial\phi} D' \left\{ \frac{1}{r} \left(\frac{\partial u'}{\partial\theta} - \sin\phi w' \right) + \frac{\partial w'}{g\partial\phi} \right\} + \frac{\partial}{r\partial\theta} D^* \left\{ \nu \left(\frac{\partial u'}{g\partial\phi} - \frac{v'}{\rho} \right) \right. \\
&\quad \left. + \frac{1-\nu}{r} \left(\frac{\partial w'}{\partial\theta} + \sin\phi u' + \cos\phi v' \right) \right\} + \frac{1}{r} \left[2\sin\phi D' \left\{ \frac{1}{r} \left(\frac{\partial u'}{\partial\theta} - \sin\phi w' \right) \right. \right. \\
&\quad \left. \left. + \frac{\partial w'}{g\partial\phi} \right\} + \cos\phi \frac{D'}{r} \left(\frac{\partial v'}{\partial\theta} - \cos\phi w' \right) \right] - k' \left\{ \frac{1}{r} \left(\frac{\partial v}{\partial\theta} - \cos\phi w \right) + w' \right\} + f_t' = 0
\end{aligned} \tag{2-35}$$

とくに利用度の高い軸対称変形では $w=0$, $w'=0$ さらに θ による微分項を落とし、下記4個の平衡方程式となる。

$$\begin{aligned}
L_s + f_s &= \frac{d}{gd\phi} k^* \left\{ (1-\nu) \left(\frac{du}{gd\phi} - \frac{v}{\rho} \right) + \nu v' + \frac{\nu}{r} (\sin\phi u + \cos\phi v) \right\} \\
&\quad + \sin\phi (1-2\nu) \frac{k^*}{r} \left\{ \frac{du}{gd\phi} - \frac{v}{\rho} - \frac{1}{r} (\sin\phi u + \cos\phi v) \right\} - \frac{k'}{\rho} \left(\frac{dv}{gd\phi} + \frac{u}{\rho} + u' \right) + f_s = 0
\end{aligned} \tag{2-36}$$

$$\begin{aligned}
L_n + f_n &= \frac{d}{gd\phi} k' \left(\frac{dv}{gd\phi} + \frac{u}{\rho} + u' \right) + \frac{1}{r} \left[\sin\phi k' \left(\frac{dv}{gd\phi} + \frac{u}{\rho} + u' \right) \right. \\
&\quad \left. - \cos\phi k^* \left\{ \nu \left(\frac{du}{gd\phi} - \frac{v}{\rho} \right) + \nu v' + \frac{1-\nu}{r} (\sin\phi u + \cos\phi v) \right\} \right] \\
&\quad + \frac{k^*}{\rho} \left\{ (1-\nu) \left(\frac{du}{gd\phi} - \frac{v}{\rho} \right) + \nu v' + \frac{1-\nu}{r} (\sin\phi u + \cos\phi v) \right\} + f_n = 0
\end{aligned} \tag{2-37}$$

$$\begin{aligned}
L_s' + f_s' &= \frac{d}{gd\phi} D^* \left\{ (1-\nu) \left(\frac{du'}{gd\phi} - \frac{v'}{\rho} \right) + \frac{\nu}{r} (\sin\phi u' + \cos\phi v') \right\} \\
&\quad + \sin\phi (1-2\nu) \frac{D^*}{r} \left\{ \frac{du'}{gd\phi} - \frac{v'}{\rho} - \frac{1}{r} (\sin\phi u' + \cos\phi v') \right\} \\
&\quad - \frac{D'}{\rho} \left(\frac{dv'}{gd\phi} + \frac{u'}{\rho} \right) - k' \left(\frac{dv}{gd\phi} + \frac{u}{\rho} + u' \right) + f_s' = 0
\end{aligned} \tag{2-38}$$

$$\begin{aligned}
L_n' + f_n' &= \frac{d}{gd\phi} D' \left(\frac{dv'}{gd\phi} + \frac{u'}{\rho} \right) + \frac{1}{r} \left[\sin\phi D' \left(\frac{dv'}{gd\phi} + \frac{u'}{\rho} \right) - \cos\phi D^* \left\{ \nu \left(\frac{du'}{gd\phi} - \frac{v'}{\rho} \right) \right. \right. \\
&\quad \left. \left. + \frac{1-\nu}{r} (\sin\phi u' + \cos\phi v') \right\} \right] + \frac{D^*}{\rho} \left\{ (1-\nu) \left(\frac{du'}{gd\phi} - \frac{v'}{\rho} \right) + \frac{\nu}{r} (\sin\phi u' + \cos\phi v') \right\}
\end{aligned}$$

$$-k^* \left\{ \nu \left(\frac{du}{g d\phi} - \frac{v}{\rho} \right) + (1-\nu)v' + \frac{\nu}{r} (\sin \phi u + \cos \phi v) \right\} + f_n' = 0 \quad (2-39)$$

非常に薄い殻では、式(2-24)、(2-30)によって近似的に

$$L_n' + f_n' \doteq N_{nn} + f_n' = 0 \quad (2-40)$$

が許される。したがって

$$N_{ss} = k(\varepsilon_{ss} + \nu \varepsilon_{tt}) + a f_n', \quad N_{nn} = f_n', \quad N_{tt} = k(\nu \varepsilon_{ss} + \varepsilon_{tt}) + a f_n' \\ (k = Eh/(1-\nu^2), \quad a = \nu/(1-\nu)) \quad (2-41)$$

に変形される。ここで便宜上改めて

$$N_{ss} = k(\varepsilon_{ss} + \nu \varepsilon_{tt}), \quad N_{tt} = k(\nu \varepsilon_{ss} + \varepsilon_{tt}) \quad (2-42)$$

とし、さらに

$$f_s = S + p_s^0 + p_s^i + a \frac{\partial f_n'}{g \partial \phi}, \quad f_n = N + p_n^0 + p_n^i + \left(-\frac{\cos \phi}{r} + \frac{1}{\rho} \right) a f_n', \\ f_t = T + p_t^0 + p_t^i + a \frac{\partial f_n'}{r \partial \theta} \quad (2-43)$$

と置けば平衡方程式は

$$L_s + f_s = 0, \quad L_n + f_n = 0, \quad L_t + f_t = 0, \quad L_s' + f_s' = 0, \quad L_t' + f_t' = 0 \quad (2-44)$$

によって置き換えられる。ここで変数は u, v, w, u', w' の5個である。

$$L_s + f_s = \frac{\partial}{g \partial \phi} k \left\{ \frac{\partial u}{g \partial \phi} - \frac{v}{\rho} + \frac{\nu}{r} \left(\frac{\partial w}{\partial \theta} + \sin \phi u + \cos \phi w \right) \right\} \\ + \frac{\partial}{r \partial \theta} k' \left\{ \frac{1}{r} \left(\frac{\partial u}{\partial \theta} - \sin \phi w \right) + \frac{\partial w}{g \partial \phi} \right\} + \frac{\sin \phi}{r} k(1-\nu) \left\{ \frac{\partial u}{g \partial \phi} - \frac{v}{\rho} \right. \\ \left. - \frac{1}{r} \left(\frac{\partial w}{\partial \theta} + \sin \phi u + \cos \phi w \right) \right\} - \frac{k'}{\rho} \left(\frac{\partial v}{g \partial \phi} + \frac{u}{\rho} + u' \right) + f_s = 0 \quad (2-45)$$

$$L_n + f_n = \frac{\partial}{g \partial \phi} k' \left\{ \frac{\partial v}{g \partial \phi} + \frac{u}{\rho} + u' \right\} + \frac{\partial}{r \partial \theta} k' \left\{ \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \phi w \right) + w' \right\} + \frac{1}{r} \left\{ \sin \phi k' \left(\frac{\partial v}{g \partial \phi} + \frac{u}{\rho} \right. \right. \\ \left. \left. + u' \right) - \cos \phi f_n' \right\} + \frac{k}{\rho} \left\{ \frac{\partial u}{g \partial \phi} - \frac{v}{\rho} + \frac{\nu}{r} \left(\frac{\partial w}{\partial \theta} + \sin \phi u + \cos \phi w \right) \right\} + f_n = 0 \quad (2-46)$$

$$L_t + f_t = \frac{\partial}{g \partial \phi} k' \left\{ \frac{1}{r} \left(\frac{\partial u}{\partial \theta} - \sin \phi w \right) + \frac{\partial w}{g \partial \phi} \right\} + \frac{\partial}{r \partial \theta} k \left\{ \nu \left(\frac{\partial u}{g \partial \phi} - \frac{v}{\rho} \right) \right. \\ \left. + \frac{1}{r} \left(\frac{\partial w}{\partial \theta} + \sin \phi u + \cos \phi w \right) \right\} + \frac{k'}{r} \left[2 \sin \phi \left\{ \frac{1}{r} \left(\frac{\partial u}{\partial \theta} - \sin \phi w \right) + \frac{\partial w}{g \partial \phi} \right\} \right. \\ \left. + \cos \phi \left\{ \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \phi w \right) + w' \right\} \right] + f_t = 0 \quad (2-47)$$

$$L_s' + f_s' = \frac{\partial}{g \partial \phi} D \left\{ \frac{\partial u'}{g \partial \phi} - \frac{v'}{\rho} + \frac{\nu}{r} \left(\frac{\partial w'}{\partial \theta} + \sin \phi u' + \cos \phi w' \right) \right\} \\ + \frac{\partial}{r \partial \theta} D' \left\{ \frac{1}{r} \left(\frac{\partial u'}{\partial \theta} - \sin \phi w' \right) + \frac{\partial w'}{g \partial \phi} \right\} + \frac{\sin \phi}{r} D(1-\nu) \left\{ \frac{\partial u'}{g \partial \phi} - \frac{v'}{\rho} \right. \\ \left. - \frac{1}{r} \left(\frac{\partial w'}{\partial \theta} + \sin \phi u' + \cos \phi w' \right) \right\} - \frac{D'}{\rho} \left(\frac{\partial v'}{g \partial \phi} + \frac{u'}{\rho} \right) + f_s' = 0 \quad (2-48)$$

$$L_t' + f_t' = \frac{\partial}{g \partial \phi} D' \left\{ \frac{1}{r} \left(\frac{\partial u'}{\partial \theta} - \sin \phi w' \right) + \frac{\partial w'}{g \partial \phi} \right\} + \frac{\partial}{r \partial \theta} D \left\{ \nu \left(\frac{\partial u'}{g \partial \phi} - \frac{v'}{\rho} \right) \right. \\ \left. + \frac{1}{r} \left(\frac{\partial w'}{\partial \theta} + \sin \phi u' + \cos \phi w' \right) \right\} + \frac{D'}{r} \left[2 \sin \phi \left\{ \frac{1}{r} \left(\frac{\partial u'}{\partial \theta} - \sin \phi w' \right) \right. \right. \\ \left. \left. + \frac{\partial w'}{g \partial \phi} \right\} + \cos \phi \left\{ \frac{1}{r} \left(\frac{\partial v'}{\partial \theta} - \cos \phi w' \right) + f_t' \right\} \right] + f_t' = 0 \quad (2-49)$$

ただし

$$v' = \frac{1}{1-\nu} \left[\frac{f_n'}{k^*} - \nu \left\{ \frac{\partial u}{g \partial \phi} - \frac{v}{\rho} + \frac{1}{r} \left(\frac{\partial w}{\partial \theta} + \sin \varphi u + \cos \varphi w \right) \right\} \right] \quad (2-50)$$

このように、変数は1個低減し、5個の式となる。つぎに、さらに u', w' を消去し、変数 u, v, w の3個とする。実際には表面荷重 $p_s^0 = p_s' = p_t^0 = p_t' = 0$ の場合が多いので、ここでは $f_s' = f_t' = 0$ とする。したがって、近似的に

$$N_{sn} = k' \varepsilon_{sn} = 0, \quad N_{tn} = k' \varepsilon_{tn} = 0$$

によって剪断歪が消失する。上式より

$$u' = - \left(\frac{\partial v}{g \partial \phi} + \frac{u}{\rho} \right), \quad w' = - \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \varphi w \right) \quad (2-51)$$

をうる。実際には剪断力は完全に消失するのではなく

$$N_{sn} = \frac{\partial M_{ss}}{g \partial \phi} + \frac{\partial M_{st}}{r \partial \theta} + \frac{\sin \varphi}{r} (M_{ss} - M_{tt}) \quad (2-52)$$

$$N_{tn} = \frac{\partial M_{st}}{g \partial \phi} + \frac{\partial M_{tt}}{r \partial \theta} + 2 \sin \varphi \frac{M_{st}}{r} \quad (2-53)$$

の程度となる。 v' は省略して

$$\varepsilon_{ss}' = \frac{\partial u'}{g \partial \phi} = - \frac{\partial}{g \partial \phi} \left(\frac{\partial v}{g \partial \phi} + \frac{u}{\rho} \right), \quad (2-54)$$

$$\begin{aligned} \varepsilon_{st}' &= \frac{1}{r} \left(\frac{\partial u'}{\partial \theta} - \sin \varphi w' \right) + \frac{\partial w'}{g \partial \phi} \\ &= - \left[\frac{1}{r} \left\{ \frac{\partial}{\partial \theta} \left(\frac{\partial v}{g \partial \phi} + \frac{u}{\rho} \right) - \sin \varphi \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \varphi w \right) \right\} + \frac{\partial}{g \partial \phi} \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \varphi w \right) \right] \end{aligned} \quad (2-55)$$

$$\varepsilon_{tt}' = \frac{1}{r} \left(\frac{\partial w'}{\partial \theta} + \sin \varphi u' \right) = - \frac{1}{r} \left\{ \frac{\partial}{\partial \theta} \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \varphi w \right) + \sin \varphi \left(\frac{\partial v}{g \partial \phi} + \frac{u}{\rho} \right) \right\} \quad (2-56)$$

$$M_{ss} = D(\varepsilon_{ss}' + \nu \varepsilon_{st}'), \quad M_{st} = D' \varepsilon_{st}', \quad M_{tt} = D(\nu \varepsilon_{ss}' + \varepsilon_{tt}') \quad (2-57)$$

さらに式(2-52), (2-53)により N_{sn}, N_{tn} が u, v, w の関数として表され、最終的に式(2-18), (2-19), (2-20)により L_s, L_n, L_t が u, v, w に関する3階, 4階の微分式として表される。このように変数は3個に低減されるが微分方程式の階数は高くなる。

実際に利用される軸対称殻には次の種類がある。

- (a) 円筒殻 $x = \phi = s, r = a = \text{一定}, \varphi = 0, g = 1, \rho = \infty$
- (b) 円錐殻 $\phi = s, x = s \cos \varphi, r = s \sin \varphi, \varphi = \text{一定}, g = 1, \rho = \infty$
- (c) 球殻 $x = a \cos \phi, r = a \sin \phi, \varphi = \phi + \pi/2, g = a, \rho = a$
- (d) 楕円殻 $x = a \cos \phi, r = b \sin \phi, \varphi = \tan^{-1}(-(b/a) \cot \phi),$
 $g = \sqrt{a^2 \sin^2 \phi + b^2 \cos^2 \phi}, \rho = g^3/ab$
- (e) 放物線殻 $x = \phi^2/a, r = \phi, \varphi = \tan^{-1}(a/2\phi), g = \sqrt{1 + (2\phi/a)^2}, \rho = -ag^3/2$
- (f) 双曲線殻 $x = a \cosh \phi, r = b \sinh \phi, \varphi = \tan^{-1}((b/a) \coth \phi),$
 $g = \sqrt{a^2 \sinh^2 \phi + b^2 \cosh^2 \phi}, \rho = -g^3/ab$

2.7 差分法による数値解析

たとえば平衡方程式(2-30)~(2-35)は6個の変数 u, v, w, u', v', w' に関し、独立変数を ϕ, θ とする6個の偏微分方程式で構成されている。したがって、解析平面を ϕ - θ 面とすると解析領域は、同平面内に閉じた境界をもつ領域となる。とくに、 θ に関し閉じた軸対称線殻では $\theta = 0, 2\pi$, 閉じていない場合でも $\theta = \text{一定}$ を θ 境界とし、さらに $\phi = \text{一定}$ を ϕ 境界とする軸対称殻では解析平面で矩形の領域に写像される。したがって ϕ, θ

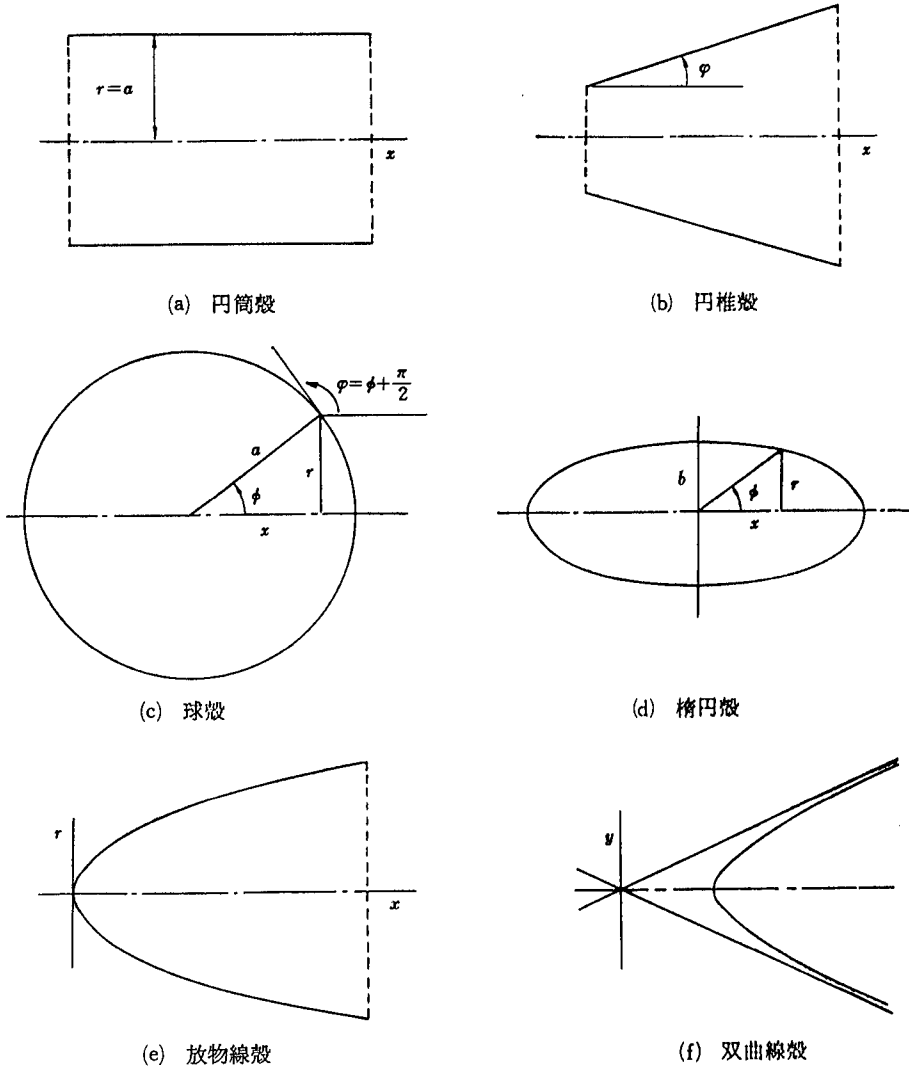


図 4 軸対称殻の種類
Fig. 4 A few example of axisymmetric shell

に関して、それぞれ一定の格子定数をもつ直交格子系を張ることができるので、上記微分方程式の差分化が容易に行れる。なおこれらの微分方程式は、たとえば

$$\frac{\partial}{\partial \phi} f(\phi) \frac{\partial u}{\partial \phi}$$

の形となっているので、そのままでは差分化が困難で、次の形

$$f(\phi) \frac{\partial^2 u}{\partial \phi^2} + f'(\phi) \frac{\partial u}{\partial \phi} \quad \left(f'(\phi) = \frac{df(\phi)}{d\phi} \right)$$

$$\left(\frac{\partial}{\partial \phi} f(\phi) \frac{\partial u}{\partial \phi} \right)_i = \{ [f_i + (f_{i-1} - f_{i+1})/4] u_{i-1} - 2f_i u_i + [f_i + (-f_{i-1} + f_{i+1})/4] u_{i+1} \} / l^2$$

または図 5 に示す節点配置によって

$$\left\{ \frac{\partial}{\partial \phi} f(\phi) \frac{\partial u}{\partial \phi} \right\}_i = \{ f_{i-1/2} u_{i-1} - (f_{i-1/2} + f_{i+1/2}) u_i + f_{i+1/2} u_{i+1} \} / l^2$$

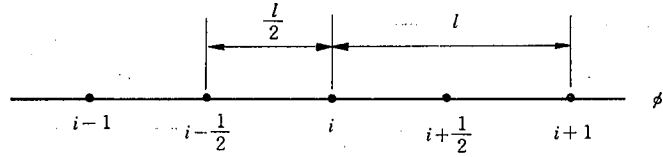
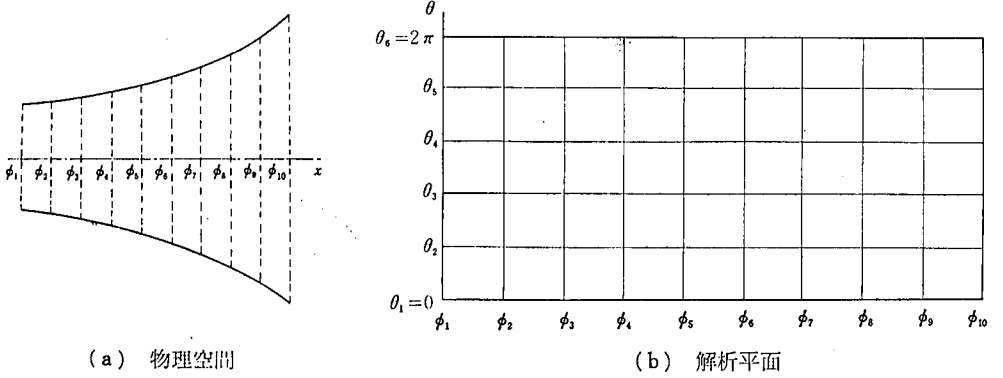


図 5 ϕ 格子ノード配置

Fig. 5 Node distribution on ϕ mesh



(a) 物理空間

(b) 解析平面

図 6 物理空間より解析平面への実像

Fig. 6 Mapping of mesh from physical space to analytical plane

に置き換えることが必要である。 θ に関する微分では、係数が θ の関数となることはない
ので、 θ 格子のように中間に補助ノードを設ける必要はない。

3. 円環殻

軸対称殻の一つの例として円環殻を取りあげる。この殻はマイタ管と同様に管系の接続
部に利用され、またブルドン管にも用いられている。

図 7 により、円環殻の形ならびに諸設定数は次のとおり。

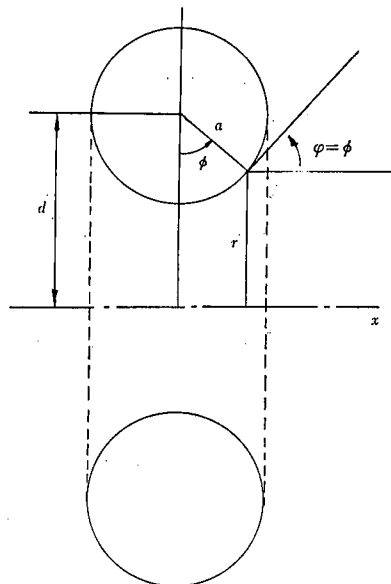


図 7 円環殻

Fig. 7 Circular ring shell

$$\left. \begin{aligned} x &= a \sin \phi, & x' &= a \cos \phi, & x'' &= -a \sin \phi \\ r &= d - a \cos \phi, & r' &= a \sin \phi, & r'' &= a \cos \phi \\ g &= \sqrt{(x')^2 + (r')^2} = a, & \rho &= g^3 / (r'' x' - r' x'') = a, & \varphi &= \phi \end{aligned} \right\} \quad (3-1)$$

3.1 平衡方程式

式(2-30)～(2-35)によって、次のように与えられる。

$$\begin{aligned} L_s + f_s &= \frac{\partial}{a \partial \phi} k^* \left\{ \frac{1-\nu}{a} \left(\frac{\partial u}{\partial \phi} - v \right) + \nu v' + \frac{\nu}{r} \left(\frac{\partial w}{\partial \theta} + \sin \phi u + \cos \phi v \right) \right\} \\ &\quad + \frac{\partial}{r \partial \theta} k' \left\{ \frac{1}{r} \left(\frac{\partial u}{\partial \theta} - \sin \phi w \right) + \frac{\partial w}{a \partial \phi} \right\} + \sin \phi (1-2\nu) \frac{k^*}{r} \left\{ \frac{1}{a} \left(\frac{\partial u}{\partial \phi} - v \right) \right. \\ &\quad \left. - \frac{1}{r} \left(\frac{\partial w}{\partial \theta} + \sin \phi u + \cos \phi v \right) \right\} - \frac{k'}{a} \left\{ \frac{1}{a} \left(\frac{\partial v}{\partial \phi} + u \right) + u' \right\} + f_s = 0 \end{aligned} \quad (3-2)$$

$$\begin{aligned} L_n + f_n &= \frac{\partial}{a \partial \phi} k' \left\{ \frac{1}{a} \left(\frac{\partial v}{\partial \phi} + u \right) + u' \right\} + \frac{\partial}{r \partial \theta} k' \left\{ \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \phi w \right) + w' \right\} \\ &\quad + \frac{1}{r} \left[\sin \phi k' \left\{ \frac{1}{a} \left(\frac{\partial v}{\partial \phi} + u \right) + u' \right\} - \cos \phi k^* \left\{ \frac{\nu}{a} \left(\frac{\partial u}{\partial \phi} - v \right) + \nu v' \right. \right. \\ &\quad \left. \left. + \frac{1-\nu}{r} \left(\frac{\partial w}{\partial \theta} + \sin \phi u + \cos \phi v \right) \right\} \right] + \frac{k^*}{a} \left\{ \frac{1-\nu}{a} \left(\frac{\partial u}{\partial \phi} - v \right) + \nu v' \right. \\ &\quad \left. + \frac{\nu}{r} \left(\frac{\partial w}{\partial \theta} + \sin \phi u + \cos \phi v \right) \right\} + f_n = 0 \end{aligned} \quad (3-3)$$

$$\begin{aligned} L_t + f_t &= \frac{\partial}{a \partial \phi} k' \left\{ \frac{1}{r} \left(\frac{\partial u}{\partial \theta} - \sin \phi w \right) + \frac{\partial w}{a \partial \phi} \right\} + \frac{\partial}{r \partial \theta} k^* \left\{ \frac{\nu}{a} \left(\frac{\partial u}{\partial \phi} - v \right) + \nu v' \right. \\ &\quad \left. + \frac{1-\nu}{r} \left(\frac{\partial w}{\partial \theta} + \sin \phi u + \cos \phi v \right) \right\} + \frac{1}{r} \left[2 \sin \phi k' \left\{ \frac{1}{r} \left(\frac{\partial u}{\partial \theta} - \sin \phi w \right) + \frac{\partial w}{a \partial \phi} \right\} \right. \\ &\quad \left. + \cos \phi k' \left\{ \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \phi w \right) + w' \right\} \right] + f_t = 0 \end{aligned} \quad (3-4)$$

$$\begin{aligned} L_{s'} + f_{s'} &= \frac{\partial}{a \partial \phi} D^* \left\{ \frac{1-\nu}{a} \left(\frac{\partial u'}{\partial \phi} - v' \right) + \frac{\nu}{r} \left(\frac{\partial w'}{\partial \theta} + \sin \phi u' + \cos \phi v' \right) \right\} \\ &\quad + \frac{\partial}{r \partial \theta} D' \left\{ \frac{1}{r} \left(\frac{\partial u'}{\partial \theta} - \sin \phi w' \right) + \frac{\partial w'}{a \partial \phi} \right\} + \sin \phi (1-2\nu) \frac{D^*}{r} \left\{ \frac{1}{a} \left(\frac{\partial u'}{\partial \phi} - v' \right) \right. \\ &\quad \left. - \frac{1}{r} \left(\frac{\partial w'}{\partial \theta} + \sin \phi u' + \cos \phi v' \right) \right\} - \frac{D'}{a^2} \left(\frac{\partial v'}{\partial \phi} + u' \right) - k' \left\{ \frac{1}{a} \left(\frac{\partial v}{\partial \phi} + u \right) + u' \right\} + f_{s'} = 0 \end{aligned} \quad (3-5)$$

$$\begin{aligned} L_{n'} + f_{n'} &= \frac{\partial}{a \partial \phi} \frac{D'}{a} \left(\frac{\partial v'}{\partial \phi} + u' \right) + \frac{\partial}{r \partial \theta} \frac{D'}{r} \left(\frac{\partial v'}{\partial \theta} - \cos \phi w' \right) + \frac{1}{r} \left[\sin \phi \frac{D'}{a} \left(\frac{\partial v'}{\partial \phi} + u' \right) \right. \\ &\quad \left. - \cos \phi D^* \left\{ \frac{\nu}{a} \left(\frac{\partial u'}{\partial \phi} - v' \right) + \frac{1-\nu}{r} \left(\frac{\partial w'}{\partial \theta} + \sin \phi u' + \cos \phi v' \right) \right\} \right] \\ &\quad + \frac{D^*}{a} \left\{ \frac{1-\nu}{a} \left(\frac{\partial u'}{\partial \phi} - v' \right) + \frac{\nu}{r} \left(\frac{\partial w'}{\partial \theta} + \sin \phi u' + \cos \phi v' \right) \right\} \\ &\quad - k^* \left\{ \frac{\nu}{a} \left(\frac{\partial u}{\partial \phi} - v \right) + (1-\nu)v' + \frac{\nu}{r} \left(\frac{\partial w}{\partial \theta} + \sin \phi u + \cos \phi v \right) \right\} + f_{n'} = 0 \end{aligned} \quad (3-6)$$

$$\begin{aligned} L_{t'} + f_{t'} &= \frac{\partial}{a \partial \phi} D' \left\{ \frac{1}{r} \left(\frac{\partial u'}{\partial \theta} - \sin \phi w' \right) + \frac{\partial w'}{a \partial \phi} \right\} + \frac{\partial}{r \partial \theta} D^* \left\{ \frac{\nu}{a} \left(\frac{\partial u'}{\partial \phi} - v' \right) \right. \\ &\quad \left. + \frac{1-\nu}{r} \left(\frac{\partial w'}{\partial \theta} + \sin \phi u' + \cos \phi v' \right) \right\} + \frac{1}{r} \left[2 \sin \phi D' \left\{ \frac{1}{r} \left(\frac{\partial u'}{\partial \theta} - \sin \phi w' \right) + \frac{\partial w'}{a \partial \phi} \right\} \right. \\ &\quad \left. + \cos \phi \frac{D'}{r} \left(\frac{\partial v'}{\partial \theta} - \cos \phi w' \right) \right] - k' \left\{ \frac{1}{r} \left(\frac{\partial v}{\partial \theta} - \cos \phi w \right) + w' \right\} + f_{t'} = 0 \end{aligned} \quad (3-7)$$

以上のように6個の変数 u, v, w, u', v', w' に対し6個の式が対応する。これらの式は非常に複雑な式であるから解析的な解を求めることは困難で、前述の解析平面における直交直線格子系による差分法が有効である。

4. おわりに

以上述べたところによって、変分原理による一般軸対称殻ならびに円環殻についてその弾性方程式を導入したが、これらは変数6個を含む6個の2階微分方程式で構成され、かなり複雑である。実用的な式として変数 u, v, w のみによる膜理論による弾性方程式も有効と思われるが、これらは次の研究課題としたい。また数値計算による式の精度の検討も必要と思われる。

参考文献 [1] K. Washizu, Variational Methods in Elasticity and Plasticity, Pergamon Press Ltd. 1968, pp.182-204.

執筆者紹介 藤野 勉 (Tsutomu Fujino)

明治45年生、昭和11年東京帝国大学理学部物理科卒業、同年三菱重工(株)入社、主に応力・振動・流体力学等の解析法(有限要素法を含む)の研究に従事、32年、工学博士号を取得、47年、同社技術本部顧問となる。また、48年より東海大学工学部教授、52年特任教授を歴任。51年より日本ユニバック(株)の技術顧問となり、現在に至る。「コンピュータによる構造工学講座II-4-B-熱伝導と熱応力」(培風館、1972)等の著書がある。



CAD/CAM 技術に欠けているもの

A Fault in Present CAD/CAM Application Technique

川合平夫

1. はじめに

現在までの CAD/CAM と称されてきている一連の技術を、実際に利用する側から見ると、表面は急速、着実に進歩してきているように見えるものの、実際の適用という面ではまだ思うようには進んでいない、というのが正直なところである。

コンピュータに直結した技術のみが先行しすぎて、それらを実用化し、効果を得るのに必要な、いわば周辺技術の整備が追いつけない、というのが一つの大きな理由ではなからうか。

2. 基本的な問題

CAD/CAM, さらには今後における広義の CAE システムというものを考えたとき、これらを有効に支援するのに必要となる基本の3本柱は、

- ① CAドラフティング・システム
- ② シミュレーション・システム
- ③ インタラクティブ・システム

であり、さらにこれらを全面的にサポートするものとして、

- ④ データ（ベース）管理システム

が不可欠となる（図1）。

しかし、シミュレーション、インタラクティブ、それに最も重要と思われるエンジニアリング・データ・マネジメントについては、実用化を進める上で多くの、そしてかなり基本的な問題（技術の欠

落）をかかえているというのが現状である。

つまりは、現在商用化されている CAD/CAM システムとは、図1の CAドラフティング・システムのレベルで、図形処理、図面処理という域を出ていないのである。

ここでは、とくに実用化が進んでいる CAドラフティング・システムに関連する問題のうちの一つをとりあげて、以下検討することにした。

2.1 図形読取り能力の弱さ

さて、コンピュータ・グラフィックスはかなり進歩し、ほぼ完成に近づいたように見られている。つまり、1960年頃から実用されたグラフィックスは、10数年を経てほぼ完成に近くなっており、形状モデリングもソリッド・モデルの実用化をピークとして、成熟期に入りつつある。

ところが、ここに依然として未開拓の一分野（機能）が残っている。これは利用側としてはかなり重要な機能なのだが、このことにメーカーの人々が気がついていないかどうか。

それは、つまり図形入力機能なのである。

いまでも、キーボード、タブレット、ライトペンなどによって、人間自身がこと細かに図形構成要素を、一つ一つ入力していくしか手がない。しかも、かなりの時間をかけてである。

ひとたび入力してしまった図形や図面に対しては、いろいろな変換や修正処理そして視覚化がかなり自由に、短時間に容易に行えるのに、入力は依然として人力のみに頼っている。これは、まさに片手落ちといえよう。

2.2 CAドラフティングの効用

この表1の数字はドラフティング・システムの効用は、「新しい図面を作成することにはなくて、すでに持っている既成図面を扱う時にこそある」ことを、端的に表している。

表1 製図主体のCADの効果例⁽¹⁾ (ウェスティングハウス社)
Table 1 An example of effects on computer aided drafting⁽¹⁾ (Westinghouse)

新図面作成の場合	工数低減比	0.1~1.1倍
修正、編集の場合	工数低減比	10~20倍

この例では、必要な、つまり「生きている」図面はすでにファイルに100パーセント入ってしまっているとすると、このような実績が出て当然なのである。

CAドラフティングに効果が期待できるのは、必要な図面（既成のものも含む）は、すべてファイル

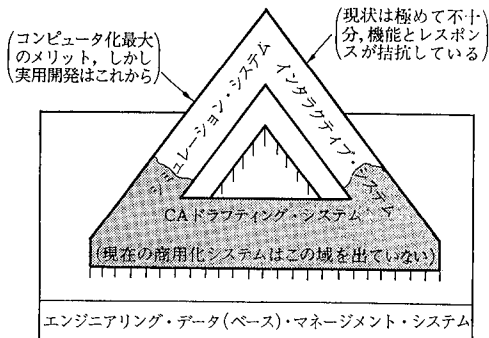


図1 CAD/CAM/CAEに必要な3大機能

Fig. 1 Indispensable basic functions on CAD/CAM/CAE systems

されているという大前提があつてのことである。

ところが、現在のシステムでは、その効用を現実化するのに、残念ながら、かなりの時間をユーザに強要しているといえよう。

3. コンピュータ・ビジョンへの動き

さて、片手落ちときめつけた“入力機能”であるが、図面の自動読取りとは、図面の内容を自動的に“認識”することであつて、単にその形をそのままコピーすることではない。(単純にコピーする装置はすでにいくつか市販されている)

図形はベクトル的に分類するほか、記号や文字、そしてそれらの接続関係などを、自動的に認識する機能が必要なのである。

これらの機能については、ようやく基礎的な面での研究が始まり^{[2]-[6]}、また、制約や不十分な点があるものの、実際に利用可能になりつつある機器やソフトが出始めている(図2)^[7]。

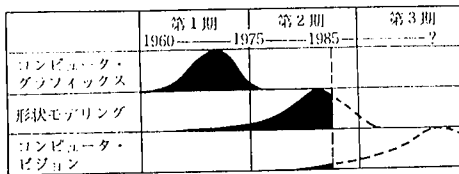


図2 図形処理の発展^[7]
Fig. 2 Development of processing techniques on graphics^[7]

日本ユニバック(株)においても、東京大学生産技術研究所の協力を得て、図面自動入力システムとしてAI-MUDAMSを昭和60年夏より提供しつつある。

4. おわりに

自動入力については、なお今後の研究成果に期待したいし、一日も早く実用でき、実効の上がるハード、ソフトの実現を切望してやまないものである。

参考文献

[1] 日本能率協会, 第2回CAD/CAM センタ訪米視察チーム報告, 1983.
 [2] 山口富士夫, “図形・形状処理入門”, 機械の研究, Vol. 37, No. 1, 1985, p. 81.
 [3] 白井良明, “CAD/CAMと画像処理・理解”, 機械の研究, Vol. 37, No. 1, 1985, p. 149.
 [4] 大沢 裕, 坂内正夫, “良好な動特性をもつ多次元点データ管理構造の一提案”, 電子通信学会論文誌, Vol. J 66-D, No. 10, 1983.
 [5] 棟上昭男, 佐藤孝紀, “図面の自動認識と理解”, 情報処理, Vol. 24, No. 9, 1983, p. 1086.
 [6] 吉田真澄, ほか, “形状認識技術への応用・Ⅲ 手書き図面の自動入力/処理装置”, 情報処理,

Vol. 22, No. 4, 1981, p. 300.

[7] 国内では、たとえば、AI-MUDAMS(日本ユニバック), TOSGRAPH(東芝), CADIX((株)キャディックス)のドキュメント参照(技術顧問)

機械翻訳の現状と将来

Machine Translation—Current Status and Future Prospects

桑野 龍夫

1. はじめに

英語から日本語への翻訳作業を紙と鉛筆で行った経験があれば誰しも感ずることは、それが非常にわずらわしく根気のいる作業であるということではなからうか。コンピュータの登場以来、コンピュータはわずらわしく根気のいる作業を人間に代わって行いその威力を発揮してきた。したがって、自然言語の翻訳をコンピュータに肩代りさせるという考えは極く自然な発想であろう。しかし一つの原文が人により、また同じ人でもその時の気分などにより、幾通りにも翻訳されるところを見ると、四則演算などとは異なり、翻訳には決まった答が存在するわけではないことに気付く。また人間が翻訳をするとき、そこには原文を理解するという段階が入ると思われるが、理解するという内容は何なのか定説があるわけではない。翻訳者は必ずしも原文の内容を、その分野の専門家のように理解して訳しているわけではなく、文の構造に関する知識など補助的手段も活用して翻訳することもあるはずである。しかし、このような場合にもある種の理解があると考えられ、理解ということの幅広さを示唆している。理解の形式化や、そのアルゴリズム開発は機械翻訳研究の諸分野の中でも注目されている分野である。

機械を用いて言葉を翻訳しようとする試みには約40年の歴史がある。しかし、これは順調な発展の連続といったものではない。とくに米国では機械翻訳を正面からとりあげる研究が1966年以降姿を消した。これは当時、近い将来の機械翻訳の有用性に悲観的な見解を示したALPAC報告(1966年)が出されたためである。現在では欧州共同体本部の機械翻訳サービス、カナダの天気予報文の翻訳など実際に使われている機械翻訳システムがある。また隣接分野のシステムとして電子化された専門用語データバンクのサービスが欧州共同体本部、西ドイツ政府言語局、カナダ政府翻訳局で行われている。翻訳システムの使用の面では、日本はここまで達していない

いが科学技術庁のプロジェクト、新世代コンピュータ技術開発機構の中期計画に研究項目として挙げられていること（知的インタフェース・ソフトウェア）、研究開発を行っている民間企業が多いことなど、近年盛上りが見られる。米国では機械翻訳の研究は行われていないこと、欧州でも研究ということでは西ドイツとフランスくらいしかない上、採算重視の傾向が強いことと比べ、日本は研究意欲が強いといえる。

2. 機械翻訳の仕組み

機械翻訳システムは原言語の文を受け取り、これに対応する目的言語の文を出力することを目的とする。機械とは今コンピュータを指す。コンピュータがこの入力文に対して行う処理は一般にトランファ方式といわれる方法で処理内容は大別して三段階に分けられる。すなわち、解析、変換、生成の三つの処理である。この処理を行うとき言語の知識が必要で、この知識の集りを辞書という。処理の第1段階の解析は、さらに形態素解析、構文解析、意味解析、文脈解析の四つに分けて考えられることが多い。

入力文に対し最初に形態素解析が行われる。形態素は意味を有する最小形態の言語単位（意義素）および意義素と共に語を構成する語尾・接尾語の類で、日本語のような膠着語では英語などに比べ語の認定、切分けを行わねばならない点が大きく異なる。the-state-of-the-arts、東京都港区赤坂は語なのかどうかなど語の定義にも困難がある。

つぎに構文の解析を行う。構文解析においては、意味には触れずに主として品詞など構文情報を参照して語や句の修飾関係、結合関係を明らかにしようとする。このとき解析の指針は文法によって与えられる。

語には多品詞語や多義語がある。また文には構文上の曖昧さがある。このため文を構文解析すると一般に複数の結果が得られる。これを一つに絞り込むには意味を手掛りとする解析が必要とされている。意味の考え方は理論により異なるが、①動作主、経験者、道具、対象、源泉、目標、場所、時間、といった動詞のとりうる格、②語の結合が意味的に整合するよう選択制限を行う手掛りとして使われる物、抽象、現象、測定、時空、部分、全体、といった意味素性、③記号としての言葉とその指し示す実体、④内包的意味と外延の意味、などが部品として使われ、文の意味が構成される。しかし、意味の形式化は総じて未熟であり、全体像もはっきりしていない。

文脈解析では文章、つまり一まとまりの複数の文

が構成する意味を捉えようとする。一つ一つの文は主題に付きいろいろのことを部分的に表現している。したがって、指示代名詞の示すものや省略されたものは一般に複数の文の意味を分析しないとわからない。文章を要約しようとしたり、文章の内容についての質問に答えようとする場合も同様である。このような分析は理解や学習および知識と深い係りがあると思われる。理解するためには、ある一つのパラダイムを共有している中で言葉を使用することが必要であるとか、先験的知識との融合が必要であるといわれる。またメンタル・モデルに置き換えることを（無意識に）行うことが理解だともいわれる。時間・空間概念のモデル化、行為の意味の定義、意図・知識・信念のモデル化の必要性も唱えられている。理解とは何かというのはコンピュータ科学、心理学、教育学、哲学、最近では認知科学の名でも論じられる学際的な問題であり、まだわからないことが多い。解明のため今後一層の研究努力が必要である。

したがって、現発展段階での機械翻訳は形態素、構文については相当に徹底した解析を行い、これにある程度の意味解析を行っているものといえよう。このような解析の結果得られた情報は入力文ごとに中間的表現としての情報実体にまとめられる。この中間的表現は、入力された原言語の属性を持つものでこれに変換の処理を施し、目的言語の属性を持つ中間的表現に変換する。解析が理想的に完璧に行えると原言語や目的言語と独立な中間的表現が考えられ、変換の過程なしで直接目的言語の生成が可能という構図がありえる（ピボット方式）。これは多言語間翻訳に親和的であるが将来の課題である。目的言語の中間的表現からは目的言語の形態素、構文の規則に基づき目的言語の文が生成される。

3. 文法理論

機械翻訳で用いられる文法は1950年代の半ばから N. Chomsky が提唱した生成文法という方法論に基礎を置いている。生成文法では、言語学を経験科学として捉える。すなわち言語学は事実の収集に終始するのではなく事実の背後に控える普遍的真理を発見し、それによって逆に事実を説明することを必要とする。生成文法は文の生成の規則を定式化しようとするもので、理想状態における話者・聴者の内在的言語能力を明示的に記述することを目標とする。

現在の N. Chomsky の理論は GB 理論といい空範疇を適正に扱うための統率 (Government) およ

び、代名詞などが先行詞を持つ条件、持たない条件を定める束縛 (Binding) という二つの概念を軸に組み立てられている。子供が生育環境の言語をどんな言語でも習得することから、言語習得の初期状態としては個別言語の差を捨象したものを考えねばならない。また言語の習得を終えた成人がいろいろな判断ができることは、すでに文法知識を獲得しているということであるが、文法は個別言語によって異なる。これらのことから初期状態で子供にはすでに内在する超文法というべき文法があり (普遍文法)、これに媒介変数が含まれているという枠組みと言語習得とは言語経験によって普遍文法の媒介変数の値を決めることだという仕組みが考えられた。

Lexical Functional Grammar (LFG) は1980年頃 J. Bresnan, R. Kaplan らが提唱した文法理論で、文脈自由文法に基づく。変形操作は行わない。文を表現するためにはC構造 (Constituent Structure) とF構造 (Functional Structure) という二つの構造を用いる。C構造は構文解析木を表し、F構造は表層の主語・目的語といった役割を属性と属性値との対による階層的データ構造として表す。英語の動詞の能動形と受動形とのかかわりには相当の規則性もあるが例外も多い。したがって、この種の問題を主として文法として扱う立場と、辞書 (Lexicon) を充実して対応する立場の二つがある。Lexical Functional Grammar では原則的にはこれを辞書に収める。この結果、辞書の項目は総じて細分化特定である。ただし冗長規則 (redundancy rule) を用い、この傾向を弱めることも行っている。

Generalized Phrase Structure Grammar (GPSG) は1980年頃 G. Gazdar らが提唱した。この文法理論も文脈自由文法に基づくことと、変形操作を行わない点はLFG同様である。英語には主語と述語との間の数の一致を保つ規則性があるが、この種の文法素性はまとめて範疇名に付加して扱うことにより、文法規則の冗長さを減少できる。また、句を展開する規則において親の素性を受けつぐ子の範疇を頭 (Head) という概念でとらえると、文法規則はさらに簡潔になる。(Head Feature Convention) このように文法規則の冗長性を除こうとして、Generalized Phrase Structure Grammar は文法規則より抽象度が高く、文法規則の規則を定める超規則を用いる。意味の表現には R. Montague の論理式を用いる方法 (Truth-conditional Model-theoretic Semantics) を採用している。

Definite Clause Grammar は、1975年頃から A. Colmerauer と R. Kowalski との提唱になる形式系

であり文脈自由文法を一階述語論理で表すこと、一階述語論理を Horn 節 (Definite Clause) に限定して記述することを主内容とする。これにより構文解析をすることは Definite Clause によって構成される公理系から定理を証明することとなる。ほぼ同じ頃に、Definite Clause の集りはプログラムと考えられるということが A. Colmerauer と R. Kowalski によって唱えられ、論理式でプログラムを書く (Programming in Logic) という概念がプログラム言語 Prolog として実現されている。

格文法は1966年頃 C. J. Fillmore の始めた理論で文を捉えるとき動詞を中心に考え、動詞にどんな格範疇がかかわるかを見る。格はすべての言語に共通して存在する概念で、主語や目的語といった構文の要素ではなく意味的な元素であると考えられる。たとえば動作主格、経験者格、道具格、対象格、源泉格、目標格、場所格、時間格などである。ある名詞句がどの格に属するかを決めるとき、意味を主な手掛りとするが、この他に一文一格の原理、対立の原理、相補性の原理を援用する。格文法に対し格の種類の設定に客観的根拠が与えられていないではないかという批判がある。しかし、日本語など語順の自由度の大きな言語の解析の枠組みとして有望視されている。

Montague 文法は、1970年頃に R. Montague が提唱した理論でモンターギュー意味論ともいわれる。この方法は可能世界の中にモデルを設定し、真理条件によって意味を捉えようとする。文の意味は、単に現実の世界に依存して決まるのではなくありうべき世界、あるいはありうべかりし世界、言いかえると可能世界に依存して決まる。(Possible Worlds Semantics) その際モデルが仮定され、これは世界に何が存在するかを決めることから始めて対象言語の解釈を指定する。(Model-theoretic Semantics) 文の意味を知るといことは、その文が真となるためにこの世界がどのようであればならないかを知ることであり(真理条件)、このための道具立てとしてこの世界には多種類の対象物…実体…が存在し、実体は属性を持ち、実体と実体との間にはかかわりがあると仮定する。また文は無限に考えられるので、これに応じて文の意味も無限に決められねばならないが、このため文を構成する要素の意味から文の意味を再帰的に定めるフレーゲの原理 (または Principle of Compositionality) を用いる。

状況意味論は、1981年頃から J. Barwise と J. Perry の提唱した理論で、文の意味は状況によって異なるという立場をとる。状況とは、ある時空領域

における個体などのオブジェクト間の関係の成立・不成立のことで、この世界は多数の状況の同時並存であると考えられる。文の意味が文脈に依存して決まるということは従来も指摘されていたことであるが、状況意味論は文脈依存性を例外的なものとする暗黙の前提を排し、文脈依存性がその意味の本質であるという見直しをした点が新しい。

4. 辞書

翻訳を人が行う場合、辞書は知らないことや正確なことを調べるために用いる。機械翻訳ではシステムは、ほとんどすべての言語の情報をその都度辞書を引いて得る。必要な情報は解析・変換・生成のそれぞれの処理で異なるが、辞書はこのすべての情報の要求に対応できなければならない。機械翻訳では一般に複数の辞書が用いられる。その目的は辞書を使用する効率を向上することや辞書を手際よく管理することなどである。とくに専門用語は分野別辞書とし翻訳対象に応じて入れ替えることが実際的である。

辞書に記述しておく情報を大別すると、表記情報・形態素情報・構文情報・意味情報・語彙情報がある。こういうとそれぞれ確かな定義があるようにひびくが、前述のように単語とは何かを定義することにもむずかしさがあるように、定義が一般にきちんと与えにくいことが大きな問題である。これは辞書の記述内容に揺れを生ずる原因となる。辞書は基本用語で 5 万~10 万語、専門用語は分野ごとに 25 万語といわれる大きさであるため、その作成・維持は普通大勢で分担して行わざるえない。このため人によって記述内容の定義に対する解釈が異なるという事情も加わり、これも辞書の記述内容に揺れを生ずる原因となる。辞書のできばえは、機械翻訳システムの働きを決定的に左右するので、辞書の作成・維持管理・使用など処理の側面の工夫を重ねると共に、言語学的知識の収集と整理を続けていくことが重視されなければならない。

5. 翻訳文の品質

翻訳結果の品質は翻訳システムの良否を左右する主要因となるため、翻訳文の品質の評価はおろそかにできないが、これまでこの分野の報告は少ない。海外では 1965 年の ALPAC 報告書および 1978 年欧州共同体の評価が、国内では科学技術庁機械翻訳プロジェクトの活動に関連して京都大学からの報告が知られている。品質を評価する観点は理解容易性、忠実度、を始めとして、了解度、一貫性、有用

表 1 理解容易性
Table 1 Intelligibility

- 1) 文意は明瞭で疑う余地はない。文法、用語用法、文体ともに適切で、なんら修正の必要はない。
- 2) 文意は明瞭で理解できるが、文法、用語用法、文体上に多少の問題が見られ、1)には至らない。ただし、こうした欠陥を改善するに当たっては出力文だけを見て外人評価者単独で修正を施すことができ、日本人評価者の援助を必要としない。
- 3) 全体的文意は把握できるが、文法、用語用法上の問題のために細部理解に自信が持てない。こうした問題を取り除き、さらに文体面の改善を施そうとすると、修正方法が一つに定まらず、入力文の意味するところを日本人評価者に確認しなければならない。
- 4) 文法、用語用法上の問題が多いので、相当な思考の後、わずかに文意を仮定できるか、もしくはほとんど把握できない。こうした問題を解決して文体を改良するよう手を加えるより、新たに翻訳をやり直した方が時間も速く、質も向上すると思われる。
- 5) まったく理解できない。どんなに熟考、検討しても文意をとることは不可能である。

表 2 忠実度
Table 2 Accuracy

- 1) 入力文の構造は出力文に忠実に再現されていて、外人が見ても明瞭で、修正する必要はない。
- 2) 入力文の構造は出力文に忠実に再現されていて、また外人にも明瞭に理解できるが、多少の修正は必要である。ただし、その修正は与えられた出力文だけにとづき、外人が単独で行える。
- 3) 入力文の構造は出力文に忠実に再現されているが、語の置き換えを必要とする。
- 4) 入力文の構造は出力文にほぼ忠実に再現されているが、句と句の関係、時制、数、態、副詞の位置などに誤りがある。また、単語が重複して出力されているなどの誤りがある。
- 5) 入力文の構造は出力文にあまり良く再現されていない。句の要素の欠落、節と節、節と句、あるいは節内の関係に誤りがある。
- 6) 入力文の構造は出力文に良く再現されていない。節や句の欠落があるが、一応文としては成立している。
- 7) 入力文の構造はまったく出力文に再現されていない。主語や述部の欠落があるため、文として成立していない。ただし、表題のような名詞にあては、その主名詞（日本語で最後に位置する名詞）、またはこれを直接修飾する節・句の動詞的要素が欠落している。

性などいろいろ考えられる。どんな評価項目を設定するかということと同時に、各項目の内容をどのように決めるかも評価機関によって異なるので、報告は慎重に吟味しなければならない。表 1、表 2 に京都大学が提唱している理解容易性と忠実度の内容を示す。（ここでは原文を日本語と仮定している。）

機械翻訳システムは、どんな文が入力されても正確で美しい翻訳文を出力できればよいが、少なくとも現状および近い将来において、この域に達することは考えられない。したがって、機械翻訳システムの性能を高めようという努力と並行して入力文を制限しようという提案がある。その目的は実用ということにある。制限としては使う単語の限定、単語の意味や用法を一意に決めること、文体の制限などが考えられる。また入力文を作る人を養成することも必要となる。

6. おわりに

機械翻訳という魅力的な目標に向かってすでに約40年経過した。この間の知見の集積は理論と実際の両面にわたって膨大である。しかし機械翻訳システムを実際に使おうとすると、①辞書の構築に時間がかかる、②辞書更新の体制を整える必要がある、③出力される文が不測の誤りを含むかもしれないので、それを承知で使うかあるいは人手で事後処理をして使う必要がある、など大きな制約がある。したがって現在は誰でもが気軽に使えるようなものではない。とはいえ機械翻訳は少なくとも大量翻訳の鍵となる技術であり、今後の研究の進展に伴い各種の制約が軽減されたり、応用の新局面が開拓されて実用に耐える域に達することが期待される。

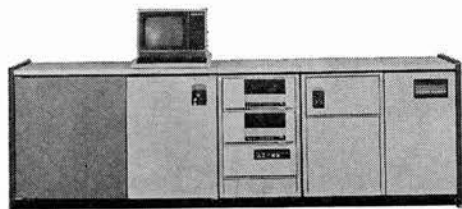
参考文献

- [1] 田町常夫, “機械翻訳の概要と歴史”, 情報処理, Vol. 26, No. 10, 1985, pp. 1140~1147.
- [2] 野村浩郷, “海外における機械翻訳の研究・開発動向”, 情報処理 Vol. 26, No. 10, 1985, pp. 1226~1236.
- [3] J. Slocum, “A Survey of Machine Translation: its History, Current Status, and Future Prospects”, *Computational Linguistics*, Vol. 11, No. 1, 1985, pp. 1~17.
- [4] 長尾 真, 言語工学, 昭晃堂, 1983.
- [5] N. Chomsky 著, 安井稔訳, 文法理論の諸相, 研究社, 1980)
- [6] 日本電子工業振興協会編, 機械翻訳システムの調査研究, 59-C-486, 1984.
- [7] 郡司隆男, “機械翻訳という観点から見た最近の言語理論”, 情報処理, Vol. 26, No. 10, 1985, pp. 1148~1156.
- [8] N. Chomsky, M. Halle 著, 橋本薫太郎, 原田信一訳, 現代言語学の基礎, 大修館, 1972.
- [9] 吉田 将”, 日高 達, “機械翻訳のための構文解析手法”, 情報処理, Vol. 26, No. 10, 1985, pp. 1157~1164.
- [10] T. Winograd, *Language as a Cognitive Process*, Vol. 1, Syntax, Addison-Wesley, 1983.
- [11] 工藤育男, 野村浩郷, 成田成之助, LFG のスキーマを用いた機械翻訳, 自然言語処理研究会資料 48-8, 1985.
- [12] F. Pereira, D. Warren, “Definite ‘Clause Grammars for Language Analysis—A Survey of the Formalism and a Comparison with Augmented Transition Networks’, *Artificial Intelligence*, Vol. 13, No. 3, 1980, pp. 231~278.
- [13] 山田真市, “DCG トランスレータの製作と言語解析への応用”, 技報, 第6号, 日本ユニパック(株), 1984, pp. 49~64.
- [14] C. J. Fillmore 著, 田中春美, 船城道雄訳, 格文法の原理, 三省堂, 1979.
- [15] D. R. Dowty, R. E. Wall, S. Peters, *Introduction to Montague Semantics*, D. Reidel, 1981.
- [16] J. Barwise, J. Perry, *Situations and Attitudes*, The MIT Press, 1983.
- [17] 土屋 俊, “状況意味論の背景と基礎概念”, 論理文法研究会資料 (未公刊).
- [18] 鈴木浩之, “状況の理論”, 論理文法研究会資料 (未公刊)
- [19] 佐伯胖編, 理解とは何か, 認知科学選書4, 東京大学出版会, 1985.
- [20] 三吉秀夫, 安川秀樹, 平川秀樹, 向井国昭, 田中祐一, 横井俊夫, “状況意味論に基く談話理解システム DUALS”, 自然言語処理研究会資料 50-7, 1985.
- [21] 石崎 俊, 井佐原均, “文脈情報学習 (同化) のシステム化について”, 自然言語処理研究会資料 52-7, 1985.
- [22] 辻井潤一, “辞書の構成と機械翻訳”, 情報処理, Vol. 26, No. 10, 1985, pp. 1174~1183.
- [23] 長尾 真, “機械翻訳文の質の評価と言語の制限”, 情報処理, Vol. 26, No. 10, 1985, pp. 1197~1202.

(技術情報サービス部)

今年発表の製品の中から主なものを選んで紹介します。各製品の詳細についてはマニュアル等をご参照ください。

●CHAPARRAL II



CHAPARRAL II は、UNIVAC 1100/90 シリーズで確立されたスーパー・ファシリティをそのままオフィス・サイズに凝縮した新しい汎用コンピュータで、先に発表した CHAPARRAL ファミリーよりも価格/性能比が一段と向上している。主な機能拡張は次のとおり。256 Kbit チップの高集積度 IC をメモリに採用し、基本キャビネット内最大 2 IP, 2 MW の構成が可能となる。基本キャビネット内で ICP と DCP/10 が接続できる。ICP および DCP/10 のメモリが 2 MB まで拡張可能である。IPCC によって、CHAPARRAL 同士、CHAPARRAL と他の UNIVAC シリーズの 1100 のチャンネル結合を実現し、トランザクション転送、ファイル転送およびジョブ転送を高速で行う。DCA ネットワークに接続された SNA ホスト、UNIVAC シリーズ 1100 ホスト、SNA 端末、UTS 端末間の通信が可能となった。

(資料コード：081221003)

(資料コード：481223001)

(資料コード：481223002)

(資料コード：481223003)

●8486 型ディスク装置 (5 GB)

8486 型ディスク装置は、1100/90 シリーズにおける超大容量・低コストの磁気ディスク・サブシステム。データ・ファイルの急激な増大に応えられるよう、1 キャビネット当たり 5.0 GB の記憶容量を持ち、5076 型磁気ディスク制御装置当たり最高 8 キャビネット、約 40 GB の記憶容量を実現する。また、8480-II 型磁気ディスク装置と混合利用することによって、要求されるデータベースにフレキシブルな対応が可能。

●MAPPER Kit 1100

MAPPER Kit 1100 は、コンピュータの専門知識がなくても、望みどおりのシステムが MAPPER

によって容易に作成できるアプリケーション開発支援システムである。

このソフトウェアは、基本となるシステム開発技法 (SUM; Sperry Universal Model)、設計・開発サポート用のソフトウェア・ツール (KIT Build) とシステム保守をサポートするソフトウェア・ツール (KIT Aid) から構成され、次の機能を持つ。

設計サポート機能 (仕様書の作成、検索、表示、印書、削除、更新および管理。仕様書に基づく各種ダイアグラムの作成)。開発サポート機能 (データ辞書の作成、検査、検索。入出力画面、MAPPER レポート、MAPPER ラン(骨組)の自動生成。ソフトウェア部品の作成、引用。デバッグ支援)。使用サポート機能 (利用者自身の実際の業務内容にそった作業メニューの作成。アプリケーション・システム標準メニューの提供。アプリケーション・システムに内蔵された操作解説書による使い方説明機能の提供)。保守サポート機能(データの削除、属性に伴う入出力画面、MAPPER レポート、MAPPER ランの自動変更。アプリケーション・システムに内蔵された仕様書によるシステム説明機能の提供)。

(資料コード：484205011-0)

(資料コード：484205012-0)

●UNIVAC DS シリーズ・ソフトウェア

(KEY FRIEND, U-Micro DPS, U-IMAGE)

多機能ディスクステーション DS7 は、Concurrent CP/M™ の下に統合ソフトウェア U-MENU を配置することによって、多彩なユーティリティ・プログラム群を効率的に統合・管理する統一的操作環境の提供を可能とした。

KEY FRIEND は、楽しみながら短時間でタイピングがマスタできるタイプ・トレーニング・ソフトである。プログラムは初級・中級・上級に分かれ、指の動かし方から始めて、ビジネス文の長文による実践的トレーニングまで、能力に合わせて段階的な訓練ができる。

U-Micro DPS は、画面ファイル受取りプログラム、画面ファイル表示プログラム、画面ファイル保守プログラムによって、あらかじめ画面ファイルをホストから伝送しておき必要に際して表示することができる。これによって、画面ファイルを全部その

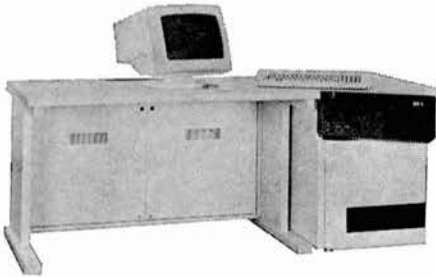
度度伝送する場合に比較して、CPU 負荷や回線使用負荷が大幅に軽減し、パフォーマンス・アップを図ることができる。

U-IMAGE は、マウスを使って自由に作画したり、イメージ・スキャナから読み込まれたイメージ情報をメニュー表示に従って思いどおりに編集できるイメージ処理ソフトウェアである。主な機能は、スキャナ読み取り、直線・円の作図、拡大・縮小機能、図形のコピー、図形の回転・反転、ペイント機能、テキスト入力、図形のセーブ/ロード、データの圧縮・伸張などがある。

(資料コード：481845427)

(資料コード：481845428)

●SS-5



32ビット超高性能ワークステーション SS-5 は、数々の優れた機能と装備の充実によって CAD/CAM 分野のみならず、科学技術分野や EOA 分野などにおいても有効に機能する高速エンジニアリング・ワークステーションである。

SS-5 は SS シリーズの共通な設計思想（パフォーマンスの追求、拡張性の重視、エンジニアリング向き入出力装置・ソフトウェアの拡充、業界標準の準拠）を継承している。

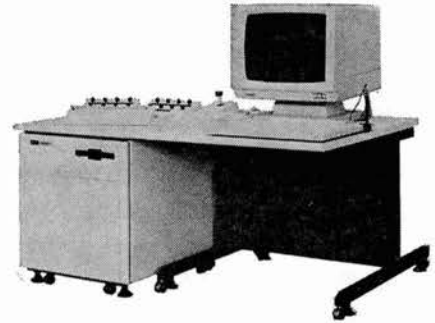
まず、高速処理を実現するため、メインプロセッサに 32 ビットの MC 68020 (16 MHz) を採用した。さらに、浮動小数点演算プロセッサ(MC 68881, 16 MHz) をプロセッサ・ボードに内蔵し、高速キャッシュ・メモリ (16 KB) を標準搭載している。また、32 ビットのバーサ・バスと 64 ビットのメモリ・バスとを分離するなどのアーキテクチャを採用している。

高速大容量を実現し、優れた拡張性を確保するため、32 MB まで拡張可能なメモリを搭載するとともに、最大 680 MB 高速・大容量ディスクが搭載できる。また、カラービットマップ・ディスプレイをはじめ、豊富な周辺装置やインターフェースが使用可能なように、18 枚までの各種ボードが収納できるマザーボードを採用している。

オペレーティング・システムには、UNIX System V を採用している。

高速バスは VERSA BUS (32 ビット) を採用し、LAN はインサネット、TCP/IP プロトコルを用いている。(資料コード：481843122-1)

●AGS 3411/3412 クラフィック・ディスプレイ



従来の AGS 3000 シリーズ・グラフィック・ディスプレイの新機種である。AGS 3411/3412 の特徴は、まず、ターミナル内にモデルを構築し、自由な大きさ・向きで見ることができるローカルビューイング機能や、論理化された入力機能など、高度なターミナル機能を装備していること。思いのままにモデリングが行える 32 ビットのモデル空間、そして大容量セグメント・バッファ、さらに構築されたモデルを正確に像変換し、表示精度を高める高精度演算機能があること。ダブル・バッファ方式のフレーム・バッファと高速処理が可能なハードウェアが、スムーズな動的表示を実現していること。また、隠面処理機能とシェーディング機構でリアリステックな表現が行えること。4 種のホスト接続機構によって、ほとんどすべてのコンピュータとの接続が可能なこと。また、16 ビット・パラレルインタフェースで、ホストとのやりとりが高速で行えること、などである。(資料コード：081841029-0)

●AGS 1500/1500 V グラフィック・ディスプレイ

従来の AGS 1000 シリーズに比べて処理速度を一段と向上させるとともに、AGS 2000 シリーズの機能を低価格で提供するものである。たとえば、最大 528 KB のセグメント・バッファ、2 次元座標変換機能、ウィンドウ・ビューポート変換機能、インタラクティブ機能等。この他、24×24 ドットの漢字フォントとストロークタイプの漢字の使い分け、各種エミュレータの装備、14 インチで 1024×780 ピクセルを実現した高密度表示、タブレットやハードコピーがフルラインアップ、グラフィック・サブルーチン・パッケージ GCSP-II E の提供等が

ある。(資料コード: 481843119-1)

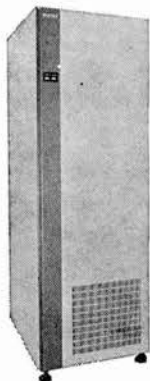
●J0764 型カラー・ハードコピー装置

J0764 型カラー・ハードコピー装置は AGS シリーズおよび IBM 5081 グラフィック・ディスプレイとビデオ・インタフェースで接続する熱転写方式マルチカラー・ハードコピーである。

特徴としては、20 インチの大型グラフィック・ディスプレイの画面をそのままの大きさに A3 サイズの用紙に出力できる。特殊ヘッドによる 8 ドット/mm の高解度と最高 4912 色の多色表現が可能で、CAD から画像処理まで広範囲に適用できる。画像メモリへのデータ取込み(約 0.5~48 秒)後、グラフィック・ディスプレイを解放し約 120 秒で高速出力が可能。ローカルで最大 99 枚まで連続出力可能。モノクロ出力も可能。ハードコピー・マルチプレクサによって、4 台までのグラフィック・ディスプレイの接続が可能。などの特徴がある。

(資料コード: 081841068-0)

●DX 8000 シリーズ (DX 8000 マルチメディア多重化装置)



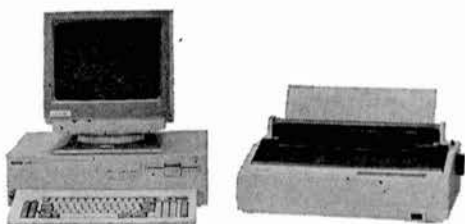
DX 8000 シリーズは、NTT がサービスしている 64 Kbps~6 Mbps の高速デジタル伝送サービスを利用し、データ・音声・ファクシミリ・画像などあらゆるメディアをデジタル多重化して一括伝送する装置である。これによって、高速、大容量、高品質の伝送が可能となり、同時にデータ系および音声系などの通信回線の重複投資の抑制と回線費用の削減が図れる。

今回発表された機種は、NTT の 6 Mbps 対応の多重化装置である“DX 8150”; 同 192 Kbps, 384 Kbps, 1.5 Mbps 対応の“DX 8130”; 同 64 Kbps 対応の“DX 8110”; 低速データ (200 bps~19.2 Kbps) を最大 24 回線集約し、48 Kbps, 56 Kbps, 64 Kbps に多重化する卓上型装置の“TM 42R”;

NTT の高速デジタル回線インタフェースと V 35 インタフェースとを交換する卓上型変換装置の“DLA 64”の五つの装置がある。

DX 8000 シリーズの特徴は、①マルチメディア情報の伝送が可能なこと、②分散している端末を集約し高速デジタル回線を介して DX 8150/8130 に接続するなど、弾力性に富んだネットワーク構築が可能なこと、③ポイント・ツー・ポイント、マルチポート、ドロップ/インサート機能により、用途・目的にあったネットワークを柔軟に効率的に構築できること、④電話網、DDX 網との接続が可能なこと、⑤32 Kbps での音声伝送が行えること、⑥回線バックアップが可能なこと、⑦主要部分が二重化され、高信頼度が実現していること、⑧回線料金が大幅に低減できること。

●シリーズ 8 システム V



シリーズ 8 の最下位機種であるシステム V (ファイブ) は、コンパクトなデスクトップ型でありながら、本格的なオフコン処理用のプロセッサと OA 処理用のプロセッサを一体化して搭載し、あらゆるオフィスワークを 1 台のワークステーションで充足することを狙ったエントリー・モデルの多機能オフコンである。

オフコン処理用のプロセッサにはシリーズ 8 で定評のある DPS IV を搭載しており、オフコンとしてこの多重処理はもちろん、上位シリーズ 8・DPS IV 機とオブジェクト・レベルで完全に互換性を持つ。

一方、OA 処理用のプロセッサには、日本語 MP/M-86 を搭載し、従来の OA ワークステーションのソフトウェアとオブジェクト・レベルで完全な互換性を持つ。これら二つのプロセッサはそれぞれ独立して動作するため、オフコン処理のパフォーマンスを満足させながら、OA 処理が同時に行える。また、システム V はスタンド・アロン・ユース以外にもシリーズ 8 のインライン・ワークステーションやリモート・ワークステーションとして活用できる。

基本部の主な仕様は次のとおり。主記憶容量: 512 KB (メインプロセッサ), 896 KB (OA サブプロセッサ)。ディスプレイ装置: 14 インチ、カラー

／モノクロ。内蔵固定ディスク装置：22 MB。内蔵ディスク装置：5インチ，1 MB。グラフィック制御装置。通信制御装置（1回線）。

（資料コード：081751099-0）

●汎用情報検索システム（CIR-8）

CIR-8 (Conversational Information Retrieval System for Series 8) は、シリーズ8のもとに実現した汎用情報検索システムで、文書などから抽出したデータを蓄積し、そのデータからさまざまな文書（マイクロフィルム、光電子ファイル、など）を手軽に検索できるように作られている。

特徴としては、部分一致検索、各種論理検索、ソーラス検索など豊富な検索機能を持つ。入力方式はメニュー方式とコマンド方式のいずれもが使用できる。インバーテッド・ファイル方式を採用し、高速検索を実現している。その他、国際標準（案）準

拠の検索命令語の採用、機密保護、ログ機能などがある。とくに光電子ファイルと有機的に結合でき、イメージ情報検索システムが容易に構築できる。

（資料コード：084751914-0）

●UP 10 E モデル 55

UP 10E モデル 55 は、モデル 70 の下位機種であり、モデル 50 の後継機である。

特徴としては、「CP/M-86」と「MS-DOS」とを標準搭載。「日本語ワープロ」および「ビジネスグラフィソフト」の2種類のOAソフトウェアと「Micro REPO」との組合せにより統合化OAソフトウェア環境を提供。640×400ドットの高解像表示カラー2画面・モノクロ6画面までの大容量VRAM、上下方向のドット単位スクロール機能、カラー16色中8色表示、モノクロ8階調表示、などの提供がある。

ユニバック技報 目次一覧

次のバックナンバーのうち、★印のあるものは多少残部がございます。
ご希望の方は、弊社テクニカル・パブリケーション室までご連絡下さい。

第 0 号 (1981 年 2 月)★

論 説	言語処理系の生産技術 UCS.....	H. C. Gyllstrom, 他
	データベース・システムの最近の傾向	G. A. Champine
論 文	記憶域階層構成の解析	T. A. Welch
	新しい差分法の提案と数値実験	藤野 勉, 渡部義維
報 告	システム記述言語 PLUS.....	F. W. Stodola
論 説	ソフトウェア開発言語 RDL.....	H. C. Heacox
	マイクロプロセッサ技術によるプロセッサ設計の考察.....	G. S. Tjaden, 他
	回路解析プログラム CIRCUIT の特徴と数値計算技法.....	長島 毅
技術動向		
	米国防総省開発言語 Ada の応用分野	真田正二
	DDX について	山浦史雄
	FTCS について.....	伊東 玄

第 1 号 (1981 年 8 月)★

論 説	3次元磁場解析への有限要素法の適用	
	——ベクトル・ポテンシャルの取扱い	渡部義維
	Kolmogorov-Chaitin の計算論的情報量と	
	Lempel-Ziv 万能データ圧縮算法	山田真市
論 文	VLSI で実現した汎用コンピュータの耐故障性	R. M. Sedmak, 他
	OA のためのワークフロー・コントロール・モデル.....	L. S. Baumann, 他
報 告	MACRO: プログラム言語	S. R. Greenwood
	高水準会話型デバッグ・システム AIDS	J. J. Hart
	汎用構文解析プログラム GSA	T. N. Turba
TECHNOLOGY TREND		
	データベース・マシン	原 潔
	ローカル・ネットワーク	守田洋一
	プロトコルの形式的記述技法	鈴木 直

第 2 号 (1982 年 2 月)

論 説	米国における商用暗号化システムの標準化	J. Nelson
	プログラム・テストにおける神託の仮説について.....	E. J. Weyuker
論 文	コンピュータ・システムの構造的モデル化技法と性能評価.....	B. S. H. Wang
	プログラム・テストの理論と等質な部分領域の応用	E. J. Weyuker, 他

- 報告 マイクロプロセッサに支援されたデータベース管理 J. R. Jordan, 他
 オンライン援助への統一アプローチ N. Relles, 他
 移植性——高水準言語の作成事例 F. P. Mehrlich, 他

TECHNOLOGY TREND

- コンピュータ利用の教育 細井 正
 プログラムレス・ツール 山岸史明

第 3 号 (1982 年 8 月)*

- 論 説 ライフ・サイクルの管理 C. Hammer
 自動設計問題の計算量 S. Sahni, 他
 論文 有限個の先読みによる正則文法の構文解析 T. J. Ostrand, 他
 報告 分散データベースの更新に関する新手法 CMT R. J. Greene
 LSI パッケージの端子と冷却 T. S. Steele
 基本変数法による粘性流解析 村上紀佐
 文節構文解析のための接続辞書の構成と付属語の接続検定 稲永紘之, 他

TECHNOLOGY TREND

- Pascal 標準化の動向 山田 勲
 MRPS について 金子 功

第 4 号 (1983 年 2 月)*

- 論 説 オペレーティング・システムにおけるセキュリティ
 ——権利の役割分離 E. J. Weyuker
 論文 任意節点配置差分法による連続体解析 藤野 勉
 精密ロック J. R. Jordan, 他
 報告 UNIVAC 1100/90 シリーズの新アーキテクチャ 古沢純一, 他
 MAPPER を利用した製造管理システム “TACTICS” L. F. Rogney
 ASHLAR における自動レイアウト——VLSI の “ジュネラル・
 セル” レイアウトの諸問題へのアプローチ J. E. Hassett
 DSS 概念の実現 小坂 武

TECHNOLOGY TREND

- ソフトウェアの品質管理と品質コスト 西島政信
 データ通信回線の第 2 次開放 守田洋一

第 5 号 (1983 年 8 月)

- 論文 テスタビリティを目的とした VLSI 設計論 C. H. Chen
 配置改良アルゴリズムにおける配線可能性の測度 A. M. Patel
 報告 分散システムにおける待ち行列モデルのネットワーク R. C. Heinselman
 ネットワーク・モデル上の関係型ビューの問い合わせ A. Rosenthal, 他
 デザインとコードにおけるウォークスルー効果 J. J. Hart
 高水準言語の下方拡張機能 T. N. Turba
 ソフトウェア開発環境 CASE 斎藤哲郎
 座標変換差分法による連続体解析 藤野 勉

TECHNOLOGY TREND

- 分散処理システムのモデリングと分析松井節男, 他
 特定のコンパイラに対するテストを仕立てる——体験談K. H. Seyfer

第 6 号 (1984 年 2 月)

- 論文 CAD のための data metamodel柳生孝昭
 浅い曲面板の弾性力学とその解析法藤野 勉
 報告 垂直磁気記録と長手磁気記録の実験による比較C. S. Chi, 他
 仕様記述言語 P3長谷川邦夫
 DCG トランスレータの製作と言語解析への応用山田真市
 陰線消去プログラム HLE平林 繁
 CHAPARRAL ファミリのハードウェア間宮健二, 他

TECHNOLOGY TPEND

- 関係問い合わせの最適化原 潔
 コマンド言語の設計における人間工学的配慮M. L. Schneider
 機械翻訳高橋 肇

第 7 号 (1984 年 8 月)*

- 論文 高階微分方程式の解析法藤野 勉
 報告 ボトル設計システム佐藤芳雄, 他
 MOS-VLSI 欠陥の分類, テスト, および除去Y. M. El-ziq
 シリーズ 1100 カナ漢字変換システムの開発三ツ矢裕一, 他
 プログラム言語の日本語化実験——日本語 PL/I の作成真田正二
 データ・セキュリティ
 ——障害に耐える閉じた環境としての 1100/90 シリーズS. J. Rawlins
 二相流解析——その背景とモデリング石丸 潤

TECHNOLOGY TREND

- 情報処理教育のカリキュラム朝倉文敏
 言語 Occam の概要若島陸夫
 無人化運転システム今西秀文

第 8 号 (1985 年 2 月)*

- 論文 還元規則を動作原理とする多重プロセッサの
 性能シミュレーションR. M. Keller, 他
 非直交曲線格子系による座標変換差分法藤野 勉
 報告 3次元家モデルに基づく住宅設計一貫システム篠田博水
 ラスタ図形処理ワークステーションR. G. Bond, 他
 経路述語によるテスト・データの作成長谷川光邦
 LISP を用いた UNIVAC シリーズ 1100 上に作った
 4 種の PROLOG 処理系桑野龍夫
 セミカスタム・セル・ライブラリを作った
 32 ビット CMOS マイクロプロセッサK. LeClair, 他

TECHNOLOGY TREND

- ビデオテックス 富麻悦三
 ANTICS (コンピュータ・アニメーション・システム) 福士祐治郎

第 9 号 (1985 年 8 月)*

- 論文 流体力学解析における偽境界に関する一考察 藤野 勉
 報告 線描きグラフィック・システムにおける
 基本データ型の抽象化の試み R. Lynn, 他
 狭間隙状態における静電気放電の電磁妨害特性
 ——ARP が EMI を決定する 本田昌実, 他
 ポスト・ストア・モードにおける
 ディスク・キャッシュの飽和現象 長尾秀実, 他
 FORTRAN コンバージョンにおける MACRO の適用 小林五郎
 ソフトウェア自動生成の実際 市丸信子

TECHNOLOGY TREND

- マルチリンク手順の概略と標準化動向 庭山正幸
 実施理論の研究について 佐口 功

▶テクニカル・コーディネータ

伊東 玄 (アドバンストプロダクト企画部), 高橋 肇 (知識システム開発部 業務課長), 高山龍雄 (システム本部 副本部長), 外山晴夫 (アドバンストプロダクト 企画部長), 中村 脩 (営業推本部 副本部長), 山田真市 (技術研究部), 米口 肇 (技術研究部長), 渡部義維 (応用ソフトウェア 二部長)

▶エディトリアル・スタッフ

永田利地 (技術情報サービス部長)

●テクニカル・パブリケーション室

村井啓一 (室長), 下田宏一 (主任研究員), 笛木祥平, 青柳幸久, 丹野敬子

●Technical Coordinators

K. Ito, O. Nakamura, H. Takahashi, T. Takayama, H. Toyama, Y. Watanabe, S. Yamada, H. Yoneguchi

●Editorial Staff

T. Nagata, K. Murai, K. Shimoda, S. Fueki, Y. Aoyagi, K. Tanno

ISSN 0289-6257

技 報

UNIVAC TECHNOLOGY REVIEW

No. 10

発行日	昭和61年3月31日
発行人兼編集人	富田和夫
発行所	日本ユニバック株式会社 東京都港区赤坂 2-17-51 〒107 TEL (03) 585-4111 (大代表)
頒布価格	1,500円
印刷所	三美印刷株式会社

禁無断複製転載

最新刊

考えるコンピュータ

B.ラファエル著 溝口文雄・内田ユリ子・岩松 聡訳 A5判・368頁・定価3200円

人類の未来にとって輝かしい夢を与える「賢いコンピュータ」の成功例を紹介すると共に、既に研究室レベルから実社会の重要な部門へと進出している様子を浮き彫りにする。人工知能入門テキストとしても最適である。

系統的プログラミング/入門(第2版) 補訂

N.ヴィルト著 野下浩平・寛捷彦・武市正人訳 A5判・208頁・定価1900円

Pascal による構造的プログラミングの概念をわかり易く解説した名著。良いプログラムを設計し作成する系統的な方法を順序立てて紹介する。このたびの補訂に際し、Pascal の新しい ISO 国際規格に関連する事項を補った。

ソフトウェア工学ライブラリ13

翻訳系構成法序論

N.ヴィルト著 寛捷彦訳 A5判・144頁・定価1500円

形式的言語とその文章分析の理論、および翻訳系(コンパイラ)作成技法に関する入門的教科書。初学者のために必要最小限の知識をコンパクトにまとめ、また各算法をプログラム化する際に Modula-2 を用いた。

好評発売中

PROLOG データベース・システム

D.リー著 安部憲広訳 A5判・240頁・定価2600円

関係データベースと密接に関連する述語論理型言語 Prolog を用いた知識情報処理の考え方とその実現技法について具体的に分かり易く解説したユニークな書。付録として Prolog プログラミング入門コースを設け、初学者のための便をはかった。

コンピュータサイエンス大学講座1.

Lisp入門 —システムとプログラミング(第3版)

中西正和著 A5判・240頁・定価2100円

既にLispの入門書として好評を博している書の第3版。改訂の主な目的としては、Common LISP をやや意識して、MACLISP 系の記述を充実させたことである。勿論、本来 Lisp が持つ単純さが故の美しさをできる限り失わないよう配慮している。

ローカルエリアネットワーク入門

D.C.フrint著 松下 温訳 A5判・400頁・定価3900円

LAN に関する基本的な技術を解説すると共に、実際の製品に深く言及し、その利害得失を注意深く論じている。特に注目すべきは、単一の LAN ばかりでなく複数のシステムを詳細に論じシステム化に際しての注意すべき点を明確に示している。

〒162 東京都新宿区市ヶ谷田町2-10

TEL. 03-260-6163 振替東京6-7625

近代科学社