

プロダクション・システム OPS 5 を用いた構造解析支援システム

青木 元也*

A consultative system for structural analysis using a production system OPS5

By
Genya AOKI

Abstract

This paper describes a consultative system for the input data format of a general purpose finite element program MARC using a production system OPS5. The knowledge bases about the procedures and data format of MARC are stored in memory as production rules. The MARC input data for a certain analysis are made up through the interaction between a user and the system which consists of a number of rules. An example procedure is demonstrated.

1. ま え が き

構造設計における強度解析にあたっては有限要素法によるプログラムの使用が現在一般的に行われている。そして、計算結果についての信頼性から、世界的に名前の知られた汎用構造解析プログラムが多く用いられている。これらのプログラムはその機能の豊富さ故に、その使用方法が非常に煩雑であり、それらの利用マニュアルも千ページを越える大冊となっている。構造解析あるいは構造設計に従事する技術者にとって、これらのプログラムの入力データの作成法を習得し、それを絶えず身に付けておくことは、有限要素法による構造解析にあたって非常に負担となっている。

前述した負担を軽減させる目的で、10年程前から構造解析を支援する各種のエキスパート・システムの作成が試みられている。ここでそれらの中の代表的なものを概観してみる。Bennett ら^{1),5)}は開発ツールとして EMYCIN を用いて汎用構造解析プログラム MARC の初心者向けの解析方法アドバイス・システムを構築した。Fjellheim²⁾は同じく EMYCIN を用いて、Det

norske Veritas の構造解析プログラム群 Sesam-69 の手法選択支援システムを作成した。Grooms ら³⁾は汎用構造解析プログラム NASTRAN を対象として FORTRAN で記述したマニュアル検索/モデル化支援システムを作った。Chen ら⁴⁾は LISP で記述された開発ツールを用いて構造システムを開発した。これらの構造解析支援のためのエキスパート・システムは、対象プログラム、使用者のレベル、目的等についてかなり限定したものとなっている。これは現状ではエキスパート・システムを成功させるには、対象をできるだけ限定することが望ましいからである。

エキスパート・システムを記述するのに適した言語あるいは開発ツールは多種類ありそれぞれ特質を有している。これらについての詳細な解説は専門書⁶⁾に譲り、ここでは代表例を列挙するだけに止める。まず言語としては、記号処理に優れた LISP、述語論理に基づいた PROLOG、オブジェクト指向の SMALLTALK 等がある。これらの言語を用いてエキスパート・システムを作成することができるが、標準的な推論機能や知識記述機能などを備えたツールがあれば、システムの開発効率が向上する。これがエキスパート・システム開発ツールである。EMYCIN, OPS5 等が古くから知られており、最近では ART, KEE 等の商用ツール

* 構造強度部

原稿受付：昭和63年5月25日

が多く用いられている。

著者らは知識工学の構造解析／設計への適用法についての基礎的な調査研究を行っている。先に、人工知能研究で中心的に使用されているリスト処理言語 LISP の構造解析への適応性を検討したが⁷⁾、今回はプロダクション・システムのうちで代表的な OPS5 の適用法を検討した。知識の断片を IF-THEN 形式で表現したものをプロダクション・ルールというが、このプロダクション・ルールに基づいて知識を処理するシステムをプロダクション・システムと呼んでいる。つぎに適用の対象としては、当研究所において広く用いられている汎用構造解析プログラム MARC の入力データ作成を支援するシステムとした。そこで次に OPS5 の機能および MARC の入力データ構成についてそれぞれの概要を述べる。

2. OPS5 の機能概要

OPS 5^{8),9),10)}は米国カーネギー・メロン大学の C.L. Forgy によって開発されたプロダクション・システムである。その基本構造を Fig.1 に示す。rules は前提—結論または条件—行動の対 (IF-THEN) で表されるルールの集合が格納される記憶領域である。data はルールによって参照あるいは更新されるデータの集合が格納される記憶領域である。interface は利用者との入力機能を有する。patterns matching では各ルールの条件部とデータとの間で照合を行い、条件を満足する全てのルールを見出す。これらを競合集合 conflict set という。conflict resolution では競合集合の中から、予め定めてある選択基準に従って、ひとつのルールを選ぶ。execution ではこの選ばれたルールの結論部を実行し、data の内容を更新する。

data のひとつの要素は次の形式で表される。

```
(name ^attribute-1 value-1
 ^attribute-2 value-2.....)
```

すなわち、要素名称といくつかの属性名と値との組みで表現される。また、ひとつのルールは次の形式で表される。

```
(p rule-name
```

```
(condition pattern) → (action))
```

すなわち、記号 p, ルール名, 条件部, 記号→, 実行部から構成される。条件部, 実行部はそれぞれいくつかの要素から成る。否定を表す条件要素にはパターンの前に負符号をつける。実行部におけるデータ操作とし

ては作成 make, 除去 remove, 変更 modify 等がある。

たとえば、いま data 記憶領域にひとつのデータ

```
(beam ^length 100 ^material concrete
 ^status unused)
```

があるとき、次のルール

```
(p example
```

```
{(beam ^length 100 ^status unused)
```

```
<beam>}
```

```
→
```

```
(modify <beam> ^status used)
```

```
(make beam ^length 50 ^material steel
 ^status unused))
```

を実行すると data 記憶領域のデータは次のようになる。

```
(beam ^length 100 ^material concrete
 ^status used)
```

```
(beam ^length 50 ^material steel
 ^status unused)
```

上の例で<beam>は { } 内に含まれる要素の指示子である。しかし <> が属性名に対応する値として用いられるときは変数を表す。

3. MARC の入力データ構成

MARC¹¹⁾は有限要素法による汎用構造解析プログラムで、線形、大変形、弾塑性、動的応答、破壊、熱伝導などの各種解析が可能であり、多種類の要素を有している。

入力データは3つのグループに分けられる。PARAMETER グループでは作業領域の割り当て、解析機能の選択、使用要素の宣言等が行われる。MODEL DEFINITION グループでは節点座標、要素と節点の組合せ、寸法、材料、荷重、境界条件等の定義を行う。HISTORY グループでは増分計算における荷重あるいは時間の制御を行なう。PARAMETER および MODEL DEFINITION のグループでは各種解析に共通な部分と特有な部分とがあるが、HISTORY グループでは特有な部分だけである。これらをまとめて図示すると Fig.2 のようになる。

それぞれのグループの各データは、キーワードとそれに付随するいくつかの数値から構成され、その入力に際してはマニュアルで指示されている順序に従わなければならない。したがって解析に当たっては、使用要素の選択、入力データの全体構成、各データの作成

等に際して大冊のマニュアルを逐一参照する必要がある。

4. OPS5を用いたMARCの入力データ作成支援システム

4.1 システムの構成

構造解析従事者の負担を軽くし、データの誤りを減らすための入力データ作成支援システムは、これからの構造設計にとって不可欠な要素であると考えられる。そこでここではMARC入力データの作成労力を軽減させるシステムについてOPS5を用いて検討した。

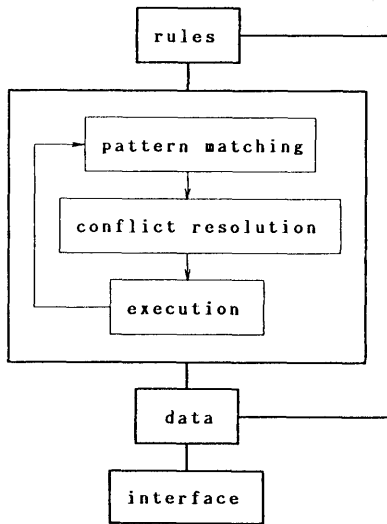


Fig.1 Fundamental architecture of OPS5

Fig.1に示したOPS5システムの基本構造において、rulesという領域にMARCに関する知識が多数のプロダクション・ルールで格納される。利用者およびシステムによって作成されたデータがdataという領域に格納される。プロダクション・ルールのIF部分には、解析種類、使用要素タイプ等が含まれ、THEN部分には各種機能に関する情報、MARC入力データの出力命令、data領域のデータ操作命令等が含まれる。本システムのdata領域にはprocedure、element等の名称を有する要素が用いられている。要素procedureは属性名としてanalysis-typeとdefinitionを有しており、解析種類および入力データ・グループ (Fig.2)の判別に使われる。要素elementはtype、code等の8つの属性名を有しており情報の表示、MARC入力データの作成等に使われる。

対象とする利用者はMARCの使用経験者とし、したがって入出力に際して詳細な注釈は付けていない。得られるMARC入力データは、選択した解析種類に必要なキーワードの全体構成、各キーワードのデータ並び、注釈から成り立っている。システムの操作は対話型のワークステーションによって行われ、システムからの問い合わせは画面に表示され、利用者からの応答はキーボードより行い、そしてMARC入力データはディスク・ファイルに格納される。

システムがスタートすると、まず画面に解析種類一覧が表示され、その中からひとつの利用者が選ぶ。このときの情報が各解析に特有なキーワードの選択に用いられる。次に要素タイプ一覧 (beam, thin-shell等)が表示され、その中から使用するタイプを選ぶ。そして選んだ要素タイプに属するMARCの要素の情報が、Table 1に示す要素データベースを参照して表示される。これに基づいて利用者は解析に使用する要素コードを決定することができる。この後は、Fig.2に示す入

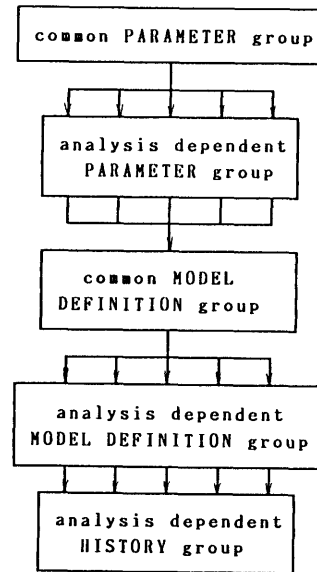


Fig.2 Organization of MARC input data

力データの構成に従って、問い合わせと応答が繰り返されて必要なMARC入力データが作成される。作成された入力データには、選択した要素コード関連の情報、各データの数値決定に参考となる注釈、図番等が付記されている。

4.2 実行例

システムのプログラム・リストおよびその解説は

Table 1 Data base of MARC elements

type	code	nodes	coords	dof	geom	comment
truss	9	2	3	3	a,	straight
beam	13	2	13	8	,sec.no.,	curved open-section
beam	52	2	3	6	a,ix,iy,	straight elastic
beam	77	2-1	6	7-1	,sec.no.,	straight open-section
beam	79	2	6	5	,sec.no.,	straight open-section
flat-plate	49	3	3	5	t,	triangle
flat-plate	50	4	3	6	t,	rectangle
flat-plate	51	4-4	3-0	5-3	t,	quadrilateral
thin-shell	4	4	14	12	t,	curved quadrilateral
thin-shell	8	3	11	9	t,	curved triangle
thin-shell	24	4-4	11	9-3	t,	curved quadrilateral
thin-shell	72	4-4	3-0	3-1	t,	twisted quadrilateral
thick-shll	22	8	6	5	nil	curved quadrilateral
thick-shll	75	4	3	6	t,	twisted quadrilateral
solid	7	8	3	3	nil	hexahedron
solid	21	20	3	3	nil	hexahedron
solid	57	20	3	3	nil	hexahedron reduced-int
axi-shell	1	2	2	3	t,r,	straight
axi-shell	15	2	5	4	t,	curved
axi-solid	2	3	2	2	nil	triangle
axi-solid	10	4	2	2	nil	quadrilateral
axi-solid	28	8	2	2	nil	quadrilateral

Appendix に示したが、ここでは一例についてその応答および得られる MARC 入力データを示す。簡単のため解析種類は固有振動、動的応答、弾性座屈、弾塑性大撓みの 4 通りとする。以下に示すステートメントはシステムからのコメントあるいは問い合わせであるが、? マークの後の文字あるいは数値は利用者の入力である。

* List of analysis types *

eigen-extraction
transient
buckling
large-displacement

Select one of the above analysis types.

? transient

* List of element types *

truss
beam
flat-plate
axi-shell
thin-shell
thick-shell
axi-solid
solid

Please type in one of the above element types.

? beam

Please type in morel or endl according to whether more element types or not.

? morel

Please type in one of the above element types.

? flat-plate

Please type in morel or endl according to whether more element types or not.

? endl

* beam code = 13 2nodes 8dof curved open-section

* beam code = 52 2nodes 6dof straight elastic

* beam code = 77 7-1nodes 7-1 dof straight open

* beam code = 79 2nodes 5dof straight open

* fqat-plate code = 49 3nodes 5dof triangle

* flat-plate code = 50 4nodes 6dof rectangle

* flat-plate code = 51 4-4nodes 5-3dof quad

Please type in the title of your analysis.

? Transient Analysis

Please type in one of element codes for use.

? 13

Please type in more2 or end2 according to whether more element codes or not.

? more2

Please type in one of element codes for use.

? 50

Please type in more2 or end2 according to whether more element codes or not.

? end2

* List of dynamic analysis methods *

1, Modal superposition
2, Newmark-beta
3, Houbolt
4, Central difference

Please type in the number of the above methods.

? 2

以上の問い合わせとそれに対する応答から、次に示す MARC 入力データが得られる。

TITLE, Transient Analysis

SIZING, 1000000,

ELEMENTS, 50, 13,

DYNAMIC, 2,

END

CONNECTIVITY

number of elements,

elem, code, list of nodes,

· * or CONN GENER(Fig.3) available

· * code 13 has 2 nodes

· * code 50 has 4 nodes

COORDINATES

max coord, number of nodes,

node, list of coordinates,

· * or NODE FILL(Fig.3), FXORD available

· * code 50 has 3 coordinates per node

· * code 13 has 13 coordinates per node

GEOMETRY

n,

geom

list of elements,

* repeat n times these two lines

* code 13 has ,sec.no., for geom

* code 50 has t, for geom

BOUNDARY CONDITIONS

n,
 0.,
 list of degree of freedom,
 list of nodes,
 * repeat n times these three lines
 * code 50 has 6 d.o.f. per node
 * code 13 has 8 d.o.f. per node

TYING,,n,

* blank line
 tye-code, tied node,
 retained nodes,
 * repeat n times these two lines

PROPERTY

n,
 youngs, poissions, density,
 list of elements,

* repeat n times these two lines

CONTROL

total time steps,
 0.1, * convergence tolerance

INITIAL CONDITIONS

* or INITIAL VEL, INITIAL DEF available

nd, nv,
 node, dof, disp, * repeat nv times
 node, dof, vel, * repeat nv times

END OPTION

DYNAMIC CHANGE

tstep, total time, nstep,, nstep,
 * Fig.4 for time step size

CONTINUE

大文字で表されているのが MARC 固有のキーワードであり、数字は利用者の応答によって定められた値あるいは既定値である。したがって、小文字で表されているデータの並びに数値を代入することによって入力データは完成する。この小文字で表されているデータは、解析対象の構造物によって大きく異なり、このシステムの中でこれらに対応する数値を定めようとするとき、システムが膨大になり過ぎるので、ここでは文字で表現するに止めた。なお、星印に続く文は注釈である。注釈の中に記してある図番は、該当項目についてより詳細な情報を得たいときに利用される。ここでは、2つの例を Fig.3 および Fig.4 に示す。Fig.3 は要素分割が規則的な場合に、GENER を用いて、節点座標

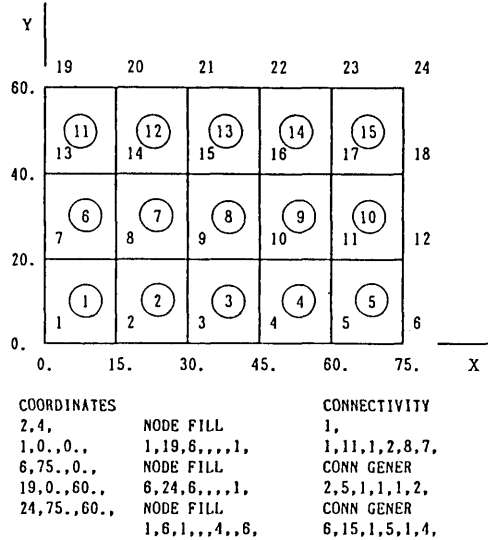


Fig.3 Simplified format for element data

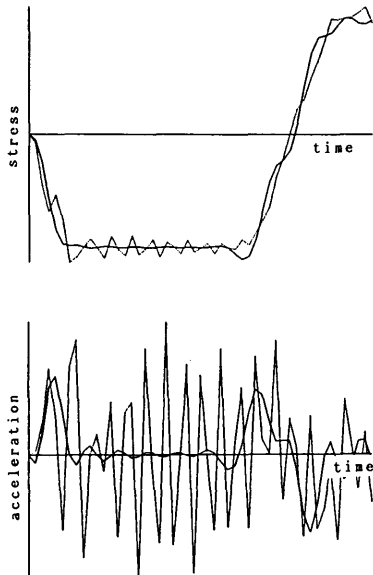
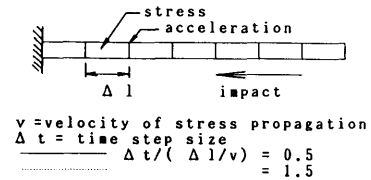


Fig.4 Selection of time step size

および要素・節点関係の入力を簡略化する具体例を示している。Fig.4は動的応答解析における時間刻みを決める際の参考図であり、時間刻みが図に示す $\Delta t/v$ より大きな値の場合には、加速度応答に脈動が生ずることを表している。

5. あとがき

プロダクション・システム OPS5 を用いて汎用構造解析プログラム MARC の入力データ作成を支援するシステムを検討した。データ作成に必要な知識はルールという形で計算機内に保持される。そしてこれらのルールから構成されるシステムと利用者との対話を通じて必要な MARC 入力データが作成される。このシステムの作成経過から得られた知見をまとめると次のようになる。

(1)解析種類、要素コード、キーワード等の間の多様な組合せみ考える場合には、在来の手続き型言語よりも、ここで用いたプロダクション・ルールによるシステムの方が、そのプログラミングに適している。

(2)各ルールの独立性が強いので、システムの変更、追加が容易に行える。

(3)プロダクション・ルールは系統的な知識と断片的な知識との両方を扱えるので適用範囲が広い。

本報告で作成したシステムは機能的にはかなり限定されたものであるが、ここで示したのと同じ手法によって、他の機能を追加すれば MARC の全機能を対象とするシステムを構築することが可能である。

OPS5 による計算およびプログラミングはテクトロニクス TEK 4406 を用いて行った。また、本研究は科学技術庁の科学技術振興調整費による重点基礎研究課題として行なったものである。

参 考 文 献

- 1) J. Bennett, L.Creary, R.Englemore and R.J. Melosh : SACON A knowledge-based consultant for structural analysis,Stanford Univ. Report No.STANCS-78-699,1978.
- 2) R.Fjellheim : An expert system for Sesam-69 program selection, Norweigan Maritime Research No.3,1984.
- 3) H.R.Grooms,W.J.Merriman and P.J.Hinz : An expert/training system for structural analysis, ASME PVP Vol.98 No.5,1989.
- 4) J.L.Chen, P.Hajela : FEMOD A consultative expert system for finite element modeling, AIAA Collect Tech Pap,1987.
- 5) 中村秀治, 寺野隆雄 : 土木構造物エキスパートシステム, オーム社, 1987.
- 6) 大須賀節雄 : 知識ベース入門, オーム社, 1986.
- 7) 青木元也 : LISP による対話型構造解析, 昭和62年度春季船舶技術研究所発表会講演集, 1987.
- 8) C.L.Forgy : OPS5 User's manual,Carnegie-Mellon University,1981.
- 9) L.Brownston,R.Farrell,E.Kant and N.Martin : Programming expert systems in OPS5,Addison Wesley publishing company,1986.
- 10) 小林重信 : 知識工学, 昭晃堂, 1986.
- 11) Manuals of MARC general purpose finite element program, MARC analysis research corporation office, version K.2.

Appendix プログラムリストおよび解説

A-1 プログラムリスト

解析種類として固有振動、動的応答、弾性座屈、弾塑性大撓みの4通りを対象とした支援システムのプログラム例を以下に示す。なお write 文における改行指示子は省略している。

1. (external twrite)
2. (defun twrite (x) (terpr) (print x))
3. (literalize procedure analysis-type definition)
4. (literalize element type code nodes
coords dof geom comment switch)
5. (literalize temporary term)
6. (p initialize)
7. {(start) <start>} -->
8. (write * List of analysis types *
eigen-extraction
transient
buckling
large-displacemet

Select one of the above analysis types.
?)
9. (make procedure ^analysis-type (accept)
^definition parameter)
10. (write * List of element types *
truss)

- ```

beam
flat-plate
axi-shell
thin-shell
thick-shell
axi-solid
solid

11. (remove <start>)
12. (make element ^type truss ^code 9
 ^nodes 2 ^coords 3 ^dof 3 ^geom a,
 ^comment straight ^switch off)
 * data base of elements Table 1
 in the above format *
13. (make morel))
14. (p eleminf (morel) -->)
15. (write Please type in one of the above
 element types.
 ?)
16. (make temporary ^term (accept))
17. (write Please type in morel or endl
 according to whether more element
 types or not.
 ?)
18. (make (accept))
19. (p elem-print (endl))
20. {(temporary ^term <type>) <temp>}
21. {(element ^type <type> ^code <code>
 ^nodes <nodes> ^dof <dof> ^comment
 <com> ^switch off) <elem>}
 -->
22. (write * <type> code = <code> <nodes>
 nodes <dof> d.o.f. <com>)
23. (modify <elem> ^switch on))
24. (p paramcom {(endl) <startpc>} -->)
25. (openfile mdata marcinput out)
26. (default mdata write)
27. (twrite Please type in the title of
 your analysis.)
28. (twrite ?)
29. (write TITLE, (accept)
 SIZING,1000000,
 ELEMENTS,)
30. (remove <startpc>) (make more2))
31. (p paramcom2 (more2) -->)
32. (twrite Please type in one of
 element codes for use.)
33. (twrite ?)
34. (make temporary ^term (accept))
35. (twrite Please type in more2 or endl
 according to whether more element
 codes or not.)
36. (twrite ?)
37. (make (accept))
38. (p paramcom3 (endl))
39. {(temporary ^term <code>) <ttemp>}
40. {(element ^code <code>) <elem>} -->
41. (modify <elem> ^switch on1)
42. (remove <temp>))
43. (p paramcom4)
44. {(element ^code <code> ^switch on1)
 <elem>} -->
45. (write <code> ,)
46. (modify <elem> ^switch on2))
47. (p parameigen)
48. {(procedure ^analysis-type eigen-
 extraction ^definition parameter)
 <proc>}
49. - (element ^switch on1) -->
50. (twrite Please type in the number
 of modes to be found.)
51. (twrite ?)
52. (write DYNAMIC,1, (accept),
 END)
53. (modify <proc> ^definition model)
54. (make startp-m))
54. (p paramtrans)
55. {(procedure ^analysis-type transient
 ^definition parameter) <proc>}
56. - (element ^switch on1) -->
57. (twrite
 * List of dynamic analysis methods *
 1,Modal superposition
 2,Newmark-bata
 3,Houbolt
 4,Central difference

 Please type in the number of

```



- the above methods.
- ?)
58. (write DYNAMIC,(accept),  
END)
59. (modify <proc> ^definition model)
60. (make startp-m))
61. (p parambuck
62. {(procedure ^analysis-type buckling  
^definition parameter) <proc>}
63. - (element ^switch on1) -->
64. (twrite  
Please type in the number of buckling  
modes to be estimated.)  
(twrite ?)
65. (write BUCKLE,(accept),  
END)
66. (modify <proc> ^definition model)
67. (make startp-m))
68. (p paramlarge
69. {(procedure ^analysis-type large-  
displacement ^definition parameter)  
<proc>}
70. - (element ^switch on1) -->
71. (write LARGE DISP  
END)
72. (modify <proc> ^definition model)
73. (make startp-m))
74. (p modcom1 {(startp-m) <startp-m>} -->
75. (write CONNECTIVITY  
number of elements,  
elem,code,list of nodes,  
. \* or CONN GENER available)
76. (remove <startp-m>)
77. (make startmc2))
78. (p modcom2 (startmc2)
79. {(element ^code <code> ^nodes <nodes>  
^switch on2) <elem>} -->
80. (write. \* code <code> has <nodes>  
nodes)
81. (modify <elem> ^switch on3))
82. (p modcom3 {(startmc2) <startmc2>}  
- (element ^switch on2) -->
83. (write COORDINATES  
max coord,number of nodes,  
node,list of coordinates,  
. \* or NODE FILL,FXORD available)
84. (remove <startmc2>)
85. (make startmc4))
86. (p modcom4 (startmc4)
87. {(element ^code <code> ^coords <coords>  
^switch on3) <elem>} -->
88. (write . \* code <code> has <coords>  
coordinates per node)
89. (modify <elem> ^switch on4))
90. (p modcom5 {(startmc4) <startmc4>}
91. - (element ^switch on3) -->
92. (write GEOMETRY  
n,  
geom  
list of elements,  
\* repeat n times these two lines)
93. (remove <startmc4>)
94. (make startmc6))
95. (p modcom6 (startmc6)
96. {(element ^code <code> ^geom <geom>  
^switch on4) <elem>} -->
97. (write \* code <code> has <geom> for geom)
98. (modify <elem> ^switch on5))
99. (p modcom7 {(startmc6) <startmc6>}
100. - (element ^switch on4) -->
101. (write BOUNDARY CONDITIONS  
n.  
0.,  
list of degree of freedom,  
list of nodes,  
\* repeat n times these three lines)
102. (remove <startmc6>)
103. (make startmc8))
104. (p modcom8 (startmc8)
105. {(element ^code <code> ^dof <dof>  
^switch on5) <elem>} -->
106. (write \* code <code> has <dof> degrees  
of freedom per node)
107. (modify <elem> ^switch on6))
108. (p modcom9 {(startmc8) <startmc8>}
109. - (element ^switch on5) -->
110. (write TYING,,n,  
\* blank line



132. (modify <proc> ^definition history))
133. (p histeigen
134. (procedure ^analysis-type eigen-  
extraction ^definition history)  
-->
135. (write MODAL SHAPE  
\* blank line  
CONTINUE)
136. (closefile mdata))
137. (p histtrans
138. (procedure ^analysis-type transient  
^definition history) -->
139. (write DYNAMIC CHANGE  
tstep,total time,nstep,,nstep,  
CONTINUE)
140. (closefile mdata))
141. (p histbuck
142. (procedure ^analysis-type buckling  
^definition history) -->
143. (write BUCKLE  
40,0.0001,  
\* max iterations,convergence tol.  
CONTINUE)
144. (closefile mdata))
145. (p histlarge
146. (procedure ^analysis-type large-  
displacement ^definition history)  
-->
147. (write AUTO LOAD  
number of times,  
\* for this load increment  
PROPORTIONAL INCREMENT  
0,ratio,  
\* min cycles,ratio of next  
to present  
CONTINUE  
\* repeat any times these five  
lines up to collapse)
148. (closefile mdata))

## A-2 プログラムの解説

行1-2では端末画面への出力文を定義している。これはプログラムの途中から、正規の write 文をディスクファイルへの出力文として使用するためである。行3-5では要素名称とそれに含まれる属性名の宣言を行なっている。行6-13はシステム初期化のルールであり、解析種類の決定、要素タイプの表示、および要素データベースの作成が行われる。行14-18では使用要素タイプが決められ、それらについての情報が行19-23で画面に表示される。行24-46で MARC 入力データの PARAMETER グループの共通部分が作成される。行25-26が作成する入力データを格納するファイルの設定である。行47-73で PARAMETER グループにおいて各解析に特有な部分が作成される。解析種類は要素名称 procedure の属性名 analysis-type で区別される。

MODEL DEFINITION グループの共通部分が行74-112で作成される。選択した MARC 要素コードに関する各種の情報が要素名 element を通じて、作成される入力データに注釈として表示される。行113-132で MODEL DEFINITION グループにおいて各解析に特有な部分が作成される。要素名 procedure の属性名 definition がグループを区別する。行133-148で HISTORY グループのデータが作成される。最後に MARC 入力データが作成されたファイルがクローズされる。