

Poincaré 予想に関する Computer 実験

(反例を求めて)

東工大 理 森川 治

Def. 1 Poincaré 予想 (3次元)

W : closed 3-mfd

$\pi_1(W) = \text{trivial} \implies W = S^3$

この問題に対して、mfd を 2 つの Solid torus L, H に分解する Heegaard 分解を用いて、コンピュータ実験を行なった。

Def. 2 $W = L \cup_f H$: W の Heegaard 分解

$\iff L, H$: Solid torus

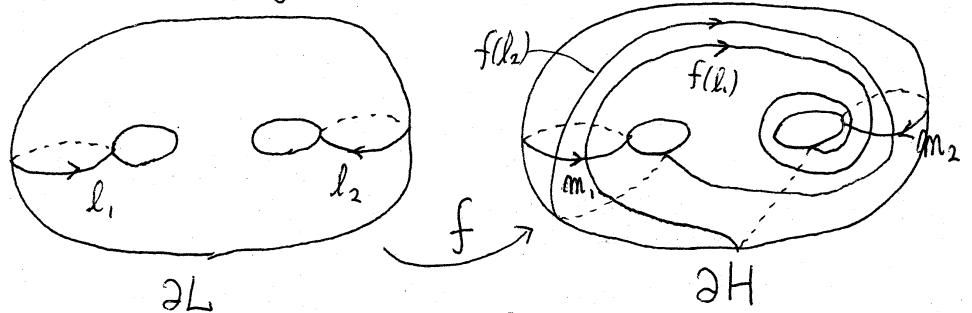
$f: 2L \rightarrow 2H$ homeo

L, H が genus 1 のとき予想は正しいことが知られているので、ここでは genus 2 の場合について論ずる。

Def. 2 より、 W を論じることは、 $f: 2L \rightarrow 2H$ を論じればよい

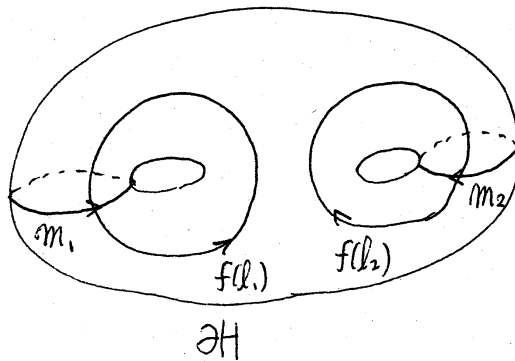
ことがわかる。さらに、 f は ∂H の meridian が、 ∂L の meridian とどのように交わっているかによって決定される。

Def. 3 ∂L の meridian l_i の f による像 $f(l_i) \in \partial H$ を longitude と呼ぶ。 ∂H の meridian, longitude を合せて m - l 系 と呼ぶ。



すなわち、manifold W は、 m - l 系により決定される。

Def. 4 m - l 系の標準型を下図のように定義する。



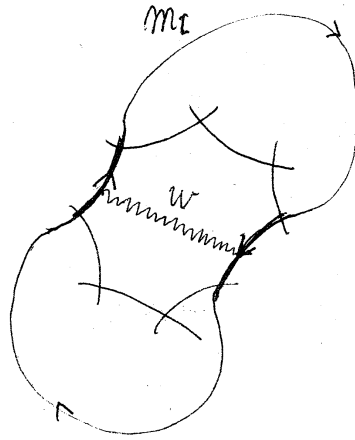
Def. 5 m - l 系が reducible である

$\iff$ m - l 系によって分解された ∂H の面の中に、Wave W をとることが可能な面が含まれる。すなわち、その面の周囲を形作る辺の中に、同じ m_i , 又は $f(l_i)$ が同じ向きに面してい

るペアが存在する。

• wave W による reduction

例えば右図のように、 m_i が W により2つのループ m'_i, m''_i に分解されたとする。このとき、 m_2 に homolog でない方の m'_i (又は m''_i) を m_i の代りにして新しい m -l 系を作る。明らかに、この操作により交点数は減少する。



Th. 1 (本間, 高橋元男, 落合)

S^3 の m -l 系が、標準型でない

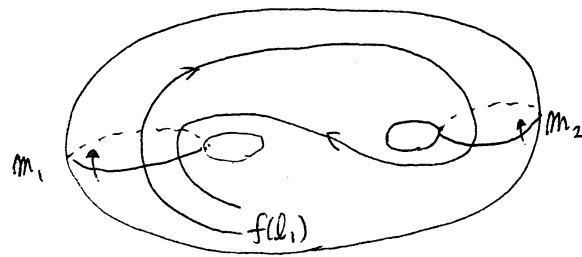
$\implies$ m -l 系は reducible

よって reducible でない、すなわち reduced な m -l 系をもつ 3-mfd W で、 $\pi_1(W) = \text{trivial}$ なものが存在すれば、Poincare 予想の反例となる。

Th. 2 W : closed 3-mfd

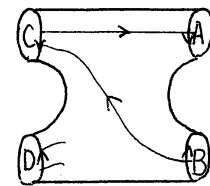
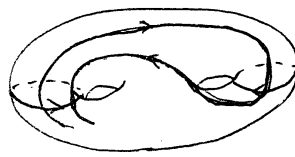
$\pi_1(W) = \{ m_1, m_2 ; \hat{l}_1, \hat{l}_2 \}$ と表わせる。

ただし、 $\hat{l}_i$ は、 fcl_i にそって m_1, m_2 との交わりを表わす word である。



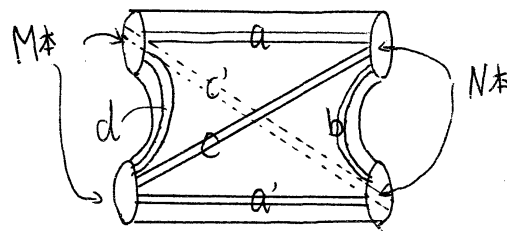
$$\hat{l}_1 = \dots m_1 m_2^{-1} m_1^{-1} \dots$$

Th. 3 reduced な m - l 系は、次の2つの type に分類される。

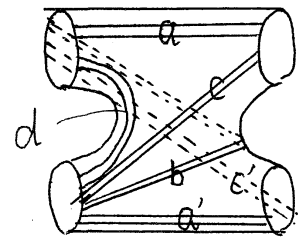


とした場合。

type 1



type 2



このとき、 a, b, c, d は本数を表わす。又、簡単な計算により、 $a = a'$, $c = c'$ が求まる。

すなわち、コンピュータを用いて、

m_1, m_2 との交点数 M, N を指定してやり、その条件下で、可能な type 1, type 2 の m - l 系を作る。その中で、reduced な $\pi_1(W)$ の trivial なものを見つける。もし見つければ、反例である。

Th. 4 $\pi_1(W) : \text{trivial} \implies H_1(W) = \pi_1(W) / [m_1, m_2] : \text{trivial}$
 $H_1(W) : \text{trivial} \iff$ 次のような matrix C が $\det C = \pm 1$
 $C = (C_{ij})$, $C_{ij} : \hat{l}_i$ の中の m_j のベキ数の和

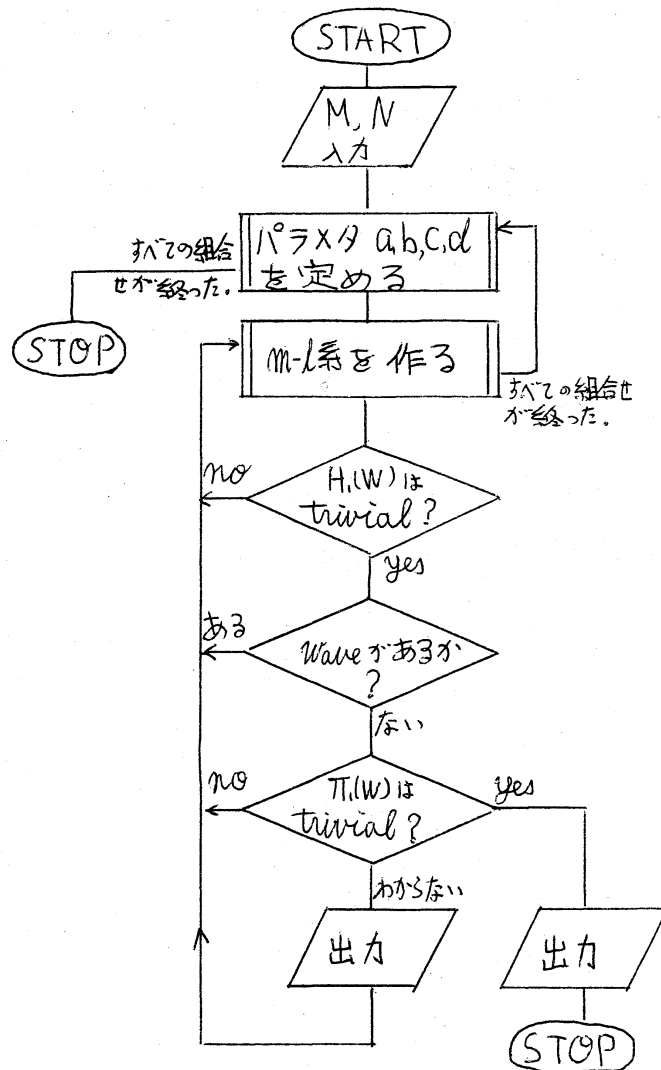
$$\left(\begin{array}{l} \text{例 } \hat{l}_1 = m_1^3 m_2^{-3} m_1^{-1} m_2^4 \\ \hat{l}_2 = m_1 m_2^2 \end{array} \implies C = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \right)$$

$\pi_1(W)$ の triviality は、
 判定が難しいので、
 Th. 4 より、先ず $H_1(W)$ の
 triviality を判定を行な
 った。よって流れ図は
 右図のようになった。

以降プログラムにそ
 って説明を行なってゆ
 くが、その前に私の使
 用した言語 R A T F O R
 (Rational FORTRAN) に
 ついて簡単に述べる。

・RATFOR はベル研究

所で作成された拡張 FORTRAN 言語であり、次の点が改良され
 ている。



(1) 文のブロック化 ----- { } でくる。(カードパンチに { } がなかったのて、私は $\{ \}$ で代用した。)

このため、if のうしろに、2つ以上の文を実行させることが可能になる。又、DO ループのための文番号が不要となる。

(2) "else" が書ける ---- if の条件の成立たない時実行。

さらに、DO ループの他に、while ループ, repeat ループ, for ループを付加。

(3) while(条件) 文 ---- 条件が成立っている間実行。

(4) repeat 文

until (条件) ----- 条件が成立つまで実行。(until は省略可)

(5) for (初期値; 条件; 更新) ---- DO ループとほとんど同じだが、条件が成立たないと1回も実行しない。

(6) break ----- 現在いるループから外へ出る。

next ----- 現在いるループの次の回へ行く。

(7) .GT., .GE., .EQ., .NE., .LE., .LT., .AND., .OR. の代りに、

<, <=, ==, /=, >=, >, &, | が使える。

(8) 書式が Free Format となる。又、"#" のうしろは注釈となる。";" で区切れば、1枚のカードに2つ以上の文が書ける。

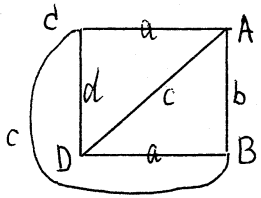
(9) Define (名前, 置換文字列) ---- 以降 "名前" は

“置換文字列”となる。

RATFOR については以上ですが、詳しくは、“Software Tools”
-1, 又、原プログラムは Addison-Wesley 社より購入できます。

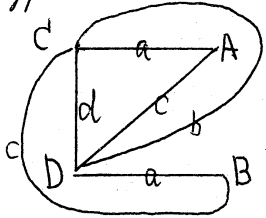
○ パラメータ a, b, c, d を定める。(参. プログラム P1, l36~41, l51~55)

type 1 について。(簡略化して、Whitehead graph で書く)



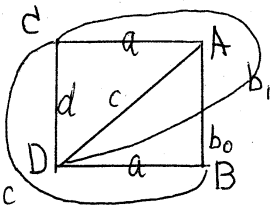
graph の対称性より、 $a \leq c$,
 $b \leq d$ と仮定した。

type 2 について。



type 1 と同様に、 $a \leq c$, $b \leq d$
さらに $a \neq 0$ ($a=0$ ならば、 w が存在する。), $b \neq 0$ ($b=0$ ならば type 1 に含まれる。) と仮定した。

コンピュータに2つの graph を記憶させるのは大変なので、
次の graph を考えた。(P3. l11~22)



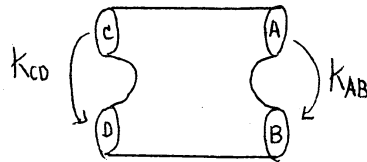
よって、パラメータは実際には、 a, b_0, b_1, c, d の5つを定めることになった。
さらに仮定より $N \leq M$ とした。

type 1 $\Rightarrow b_1 = 0$

type 2 $\Rightarrow b_0 = 0$

② m - l 系を作る。(P3. l30~41)

出来上がった Whitehead Graph の頂点 A と B , C と D を貼合
わせて、頂度 2 の closed loop をつくればよい。



③ $H(W)$ の triviality の判定。(P3. l42~47)

Th4 より、matrix C を作り、 $\det C$ を求めればよい。

④ wave の有無。(P3. l49~52)

1). 10 辺形以上があれば、必ず wave が meridian 系に存在す
る。よって 4, 6, 8 辺形について考えればよい。

2). 1 つの meridian をはさんで wave がある場合

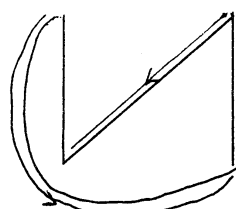


$\hat{m}_i$ の word を作ってみると、annihilation
部分がある。(P9 l5~l10)

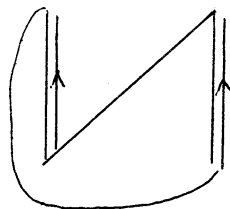
3). 2 つの meridian をはさんで wave がある場合。(8 辺形)

type 1 について。

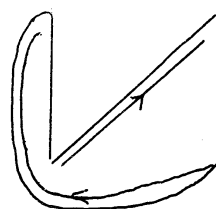
$a=0$ のとき。



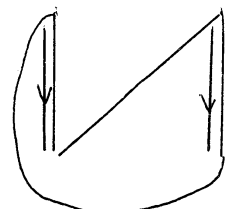
Case 1.1



1.2



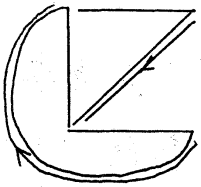
1.3



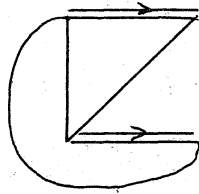
1.4

genus 2 の場合、対称性により case 1.3 と 1.1, 1.4 と 1.2 は同値となる。よって case 1.1, 1.2 を調べる。(Pg. 117~20)

$b=0$ のとき



1.5



1.6

(Pg. 112~15)

type 2 について、8 辺形は出来ない。

◎ $\pi(W)$ の triviality の判定。(P3 154~56)

Th.5 次の条件を満たす有限決定性オートマトンが存在すれば、群は trivial ではない。

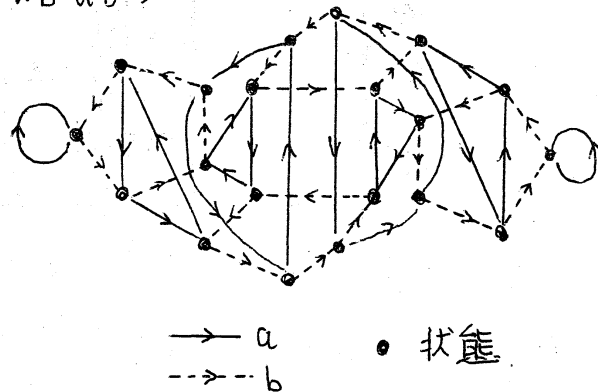
1. 各状態 P に対し、 λ 列として relation V_i を与えると、再び P となる。

$$2. P \xrightarrow{x} Q \iff Q \xrightarrow{x^{-1}} P$$

3. 状態数は、2 以上である。

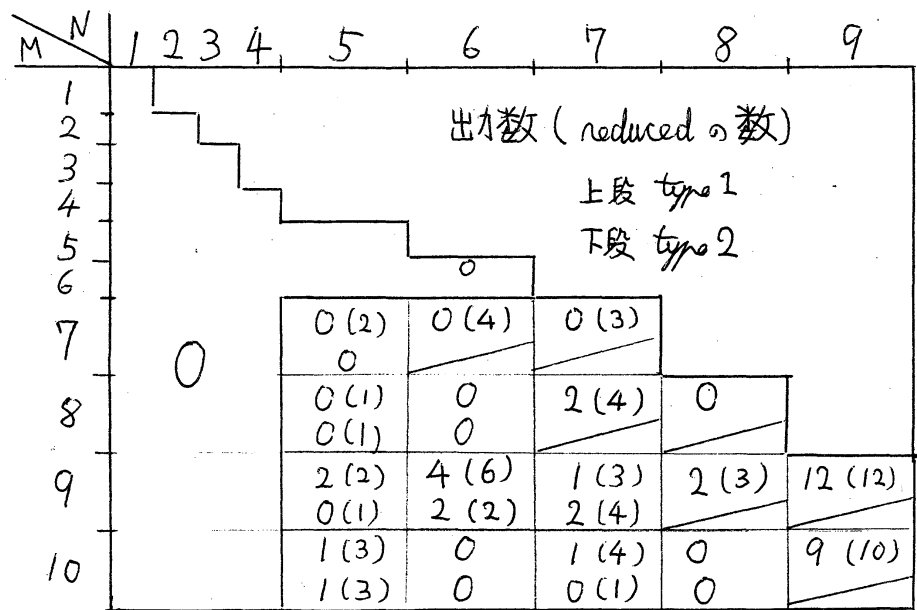
例. 群 $G = \langle a, b; r_1 = a^3 b^5, r_2 = a b^{-1} a b^4 \rangle$

に対し右図のオートマトンが条件 1, 2, 3 を満たしていることは明らかである。よって $G \neq \text{trivial}$ 。



よってこのようなオートマトンが見つければ、群が *trivial* でないことが判断できる。又、このオートマトンを見つけるときに、何の条件も付加せずに、すなわち n の結合以外では、再び状態 P にもどらないようにしてオートマトンを作って、状態数が 1 となったら、群が *trivial* であると判断できる。今回は状態数を 300 ($= \text{MAXK0}, p/10, 19$) までのオートマトンを対象にし、それより大きくなるときは判定を保留することにした。

実験の結果

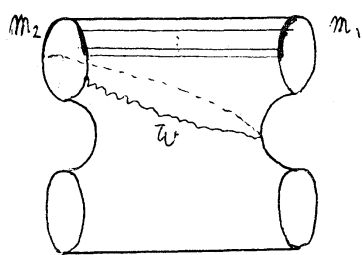


N, M が 8 以下では、ほとんどこのアルゴリズムで、反例のないことが示せた。しかし、 N, M が大きくなると、 $\pi(w)$ の *triviality* 判定ルーチンの力不足である。 $N=12, M=17$ の場合

reducedなものが type 1 で 53ヶ, type 2 で 16ヶ、に対し、出力数はそれぞれ 52ヶ, 15ヶ とほとんど無効であった。

③ wordの reducibility と wave の存在について。

右の図より meridian m_2 に wave
があれば、 m_1, m_2 にそって longitude
 l_1, l_2 を読んだ word $\hat{m}_1, \hat{m}_2$ に対し、
 $\hat{m}_1$ が $\hat{m}_2$ の subword となっている



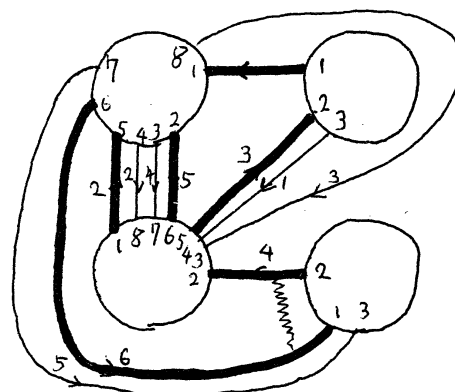
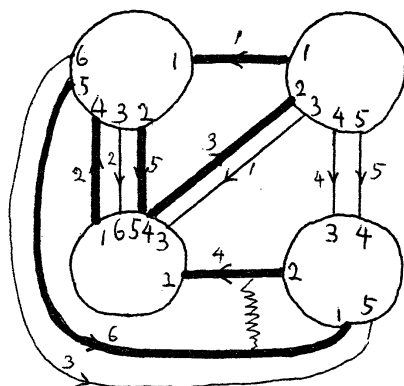
ことが予想される。又当然逆の予想、「 $\hat{m}_1$ が $\hat{m}_2$ の subword なら
 m_2 に wave が存在するであろう」が考えられる。これらの予想
に対し、 $M+N \leq 40$ に対し実験を行なったところ、次の結果を
得た。

「 $M+N \leq 40$, $H_1(W) = \text{trivial}$ ならば、一方の word が他方の
subword となれば wave が存在する。又逆に wave が存在すれば
一方は他方の subword となる。」

さらに落合氏により、type 1 の場合には、 $M+N > 40$ の場合に
も成立つことが示された。

又、 $H_1(W) = \text{trivial}$ の条件をはずすと反例が得られた。

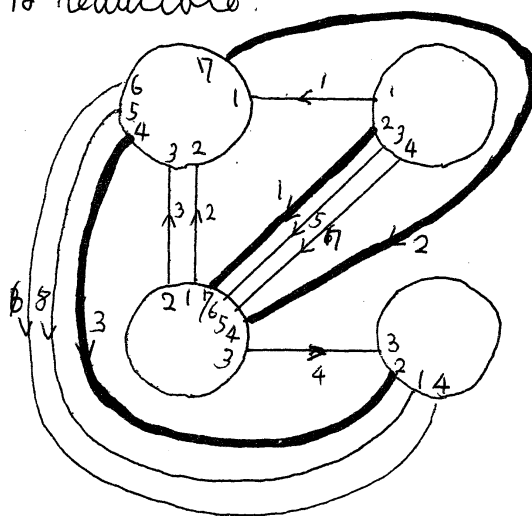
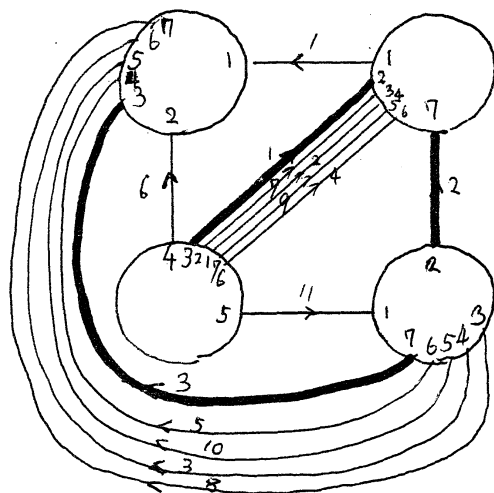
㊦ Waveがあるが、wordは reducibleでない。



$$N=5, M=6, (\hat{a}, \hat{b}, c, d) = (1, 2, 2, 3) \quad N=3, M=8, (a, b, c, d) = (1, 1, 2, 4)$$

$$\begin{cases} \hat{\ell}_1 = ab^{-2}a^{-1}b^2 \\ \hat{\ell}_2 = a^3b^2 \end{cases} \quad \begin{cases} \hat{\ell}_1 = ab^{-2}a^{-1}b^2 \\ \hat{\ell}_2 = ab^4 \end{cases}$$

㊦ Waveがないが、wordは reducible。

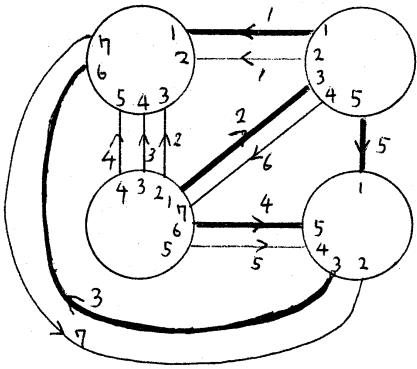


$$N=7, M=7, (1, 1, 5, 1) \quad N=4, M=7, (1, 1, 3, 2)$$

$$\begin{cases} \hat{\ell}_1 = a^2b \\ \hat{\ell}_2 = ab^{-1}a^{-1}b^{-1}a^{-1}b^{-2}a^{-1}b^{-1}b^{-1} \end{cases} \quad \begin{cases} \hat{\ell}_1 = ab^2 \\ \hat{\ell}_2 = ab^{-3}abab \end{cases}$$

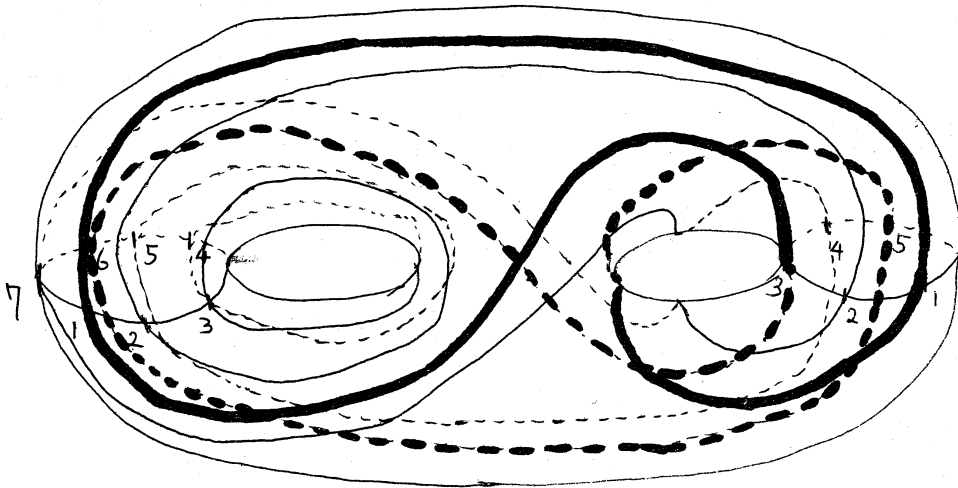
しかし、Waveがあり、Wordが reducible でない例は、closed loop が m - l 系をなしていない、すなわち、homology 0 だったり、一方が他方と homology であるものしか見つからなかった。ですから現在のところ、「Waveがあれば、wordが reducible である」という予想は、 m - l 系においては未だ解決していない。

① 最も交点数の少ない、reduced な m - l 系(標準型以外)



$$N=5, M=7, (2, 1, 2, 3)$$

$$\begin{cases} \hat{l}_1 = a^2 b^{-1} a^{-1} b^{-1} \\ \hat{l}_2 = a b a b^{-4} \end{cases}$$



この Heegaard 分解をもつ 3-mfd W は明らかに S^3 でない。
又、 $\pi_1(W) \neq \text{trivial}$ である。

参考文献

- [1] 本間龍雄 $\text{genus } 2$ の Heegaard 分解をもつ S^3 について
(曲面の位置の理論と関連する話題), 数理解析研講究録
297 (1977)
- [2] J. A. Todd and H. S. M. Coxeter, A practical method for
enumerating cosets of a finite abstract group, Proc.
Edinburgh Math. Soc. (2), 5 (1936), 26~34
- [3] J. Leech, Coset enumeration on digital computers,
Proc. Camb. Phil. Soc. 59 (1963), 257~267
- [4] N. S. Mendelsohn, An algorithmic solution for a word
problem in group theory, Dept. of Math. Publications,
Univ. of Manitoba, 1963
- [5] 落合豊行 On geometric reductions of homology
3-spheres of genus two (多様体に於ける低次元トポロジ
ーの問題), 数理解析研講究録 309 (1977)

```

1 #####
2 # HEEGAARD SPLITTING OF GENUS TWO 1977 6/6 BY MORIKAWA #
3 # 'HGDSP0' (KYOTO 1978 8/29 ) #
4 #####
5 DEFINE(OWARI,900)
6 DEFINE(SIZE,80)
7 DEFINE(TRUE,1); DEFINE(FALSE,0)
8 DEFINE(CHARACTER,INTEGER)
9 DEFINE(COMMONCONSTANT,COMMON /CONST/ M,N,MA,MB,MC,MD,MN,
10 ICNT1,ICNT2,ICNT3,ICNT4)
11 DEFINE(COMMONOUT,COMMON /OUT/ IPAGE,CHAR,IA2,IB2,IC2,ID2;
12 CHARACTER CHAR(SIZE,2))
13 COMMONCONSTANT
14 COMMONOUT
15 COMMON /COND/ ICOND # 1978 7/24
16 #
17 # MA MB MC MD MN
18 # I I I I I
19 # V V V V V
20 # CHAR: AAA.....ABBB.....BBBB.....CCCC.....D
21 # 123.....N123.....N123.....M123.....M
22 # VERTEX: ( MAPPING BY 'KAB' ) ( MAPPING BY 'KCD' )
23 # EDGE: ( MAPPING BY IA0,IB0,IB1,IC0,IDO )
24 #
25 REPEAT *(
26 READ(5,500,END=OWARI) N,M,ICOND # 1978 7/24
27 500 FORMAT(2I4,6X,I1) # 1978 7/24
28 IF( MOD(N,2)+MOD(M,2) == 0) NEXT # 1978 7/17
29 IF(N > M) *( I=N; N=M; M=I; *)
30 MA=0; MB=N; MC=2*N; MD=MC+M; MN=MD+M
31 CALL SETCHA(CHAR)
32 #
33 IPAGE=ICNT1=ICNT2=ICNT3=ICNT4=0
34 WRITE(6,600) N,M
35 600 FORMAT('1N=',I4,5X,'M=',I4,10X,'GRAPH=G1')
36 IAMAX=N/2+1
37 DO IA=1,IAMAX *(
38 ICMAX=N-IA+2
39 DO IC=IA,ICMAX
40 CALL WGRAPH(IA-1,N-IA-IC+2,0,IC-1,M-IA-IC+2)
41 *)
42 WRITE(6,610) ICNT4,ICNT3,ICNT2,ICNT1
43 610 FORMAT(' #### ',I3,' (REDUCED',I4,', HOMOMLOGY',I4,
44 ', 2-LOOPS',I4,') ####')
45 #
46 IF(2 > M-N) NEXT # 1978 8/5
47 IF(N <= 1) NEXT # 1978 9/1
48 IPAGE=ICNT1=ICNT2=ICNT3=ICNT4=0
49 WRITE(6,601) N,M
50 601 FORMAT('1N=',I4,5X,'M=',I4,10X,'GRAPH=G2')
51 IAMAX=N/2; IBMAX=(M-N)/2 # 1978 8/15
52 DO IA=1,IAMAX # 1978 8/15
53 DO IB=1,IBMAX
54 CALL WGRAPH(IA,0,IB,N-IA,M-N-IB)
55 *)
56 WRITE(6,610) ICNT4,ICNT3,ICNT2,ICNT1
57 *)
58 OWARI STOP
59 END

```

```
1 # CHAR:= TRANSLATION TABLE FOR 'OUT' ROUTINE
2 # INTERNAL CODE --> VERTEX ON GRAPH
3 #     MA+1      -->    (A 1)
4 #     MA+2      -->    (A 2)
5 #     :         :
6 #     MD+N      -->    (D N)
7 #
8 SUBROUTINE SETCHA(CHAR)
9 COMMONCONSTANT
10 CHARACTER CHAR(SIZE,2)
11 DO I=1,N *(
12     CHAR(I+MA,1)='A'
13     CHAR(I+MB,1)='B'
14     CHAR(I+MA,2)=CHAR(I+MB,2)=I
15 *)
16 DO I=1,M *(
17     CHAR(I+MC,1)='C'
18     CHAR(I+MD,1)='D'
19     CHAR(I+MC,2)=CHAR(I+MD,2)=I
20 *)
21 RETURN
22 END
```

```

1      DEFINE(TRIVIAL,1)
2      DEFINE(JUST2LOOPS,0)
3      DEFINE(ONLY1LOOP,1)
4      DEFINE(ATLEAST3LOOPS,2)
5      DEFINE(REDUCIBLE,FALSE);   DEFINE(REDUCED,TRUE)
6      DEFINE(COMMONWORK1,COMMON /WORK1/ EDGE,EDGE1,VERTEX,CHECK;
7          INTEGER EDGE(SIZE),EDGE1(SIZE),VERTEX(SIZE),CHECK(SIZE))
8      DEFINE(COMMONWORK2,COMMON /WORK2/ LOOP,LL,WORD,LW,C;
9          INTEGER LOOP(SIZE,2),LL(2),WORD(SIZE,2),LW(2),C(2,2))
10     #
11     # GRAPH:  /--(IB1)-----I
12     #          C---(IA0)---A I          IA0 <= IC0
13     #          /I              / I I      IB0 <= ID0
14     #          I I            / I I      IB1 <= ID0
15     #          I(ID0) (IC0) I I          GRAPH-1 ... IB1=0
16     #          I I          / ---+--/    GRAPH-2 ... IB0=0
17     #          I I / ---/ I              0 < IB1
18     #          I I /---/ (IB0)
19     #          I D---(IA0)---B
20     #          I----- (IC0)---/
21     #
22     SUBROUTINE WGRAPH(IA0,IB0,IB1,IC0,ID0)      # WHITEHEAD-GRAPH
23     COMMONCONSTANT
24     COMMONOUT
25     COMMONWORK1
26     COMMONWORK2
27     IF(IA0+IB0==0 | IA0+IC0==0) RETURN      # REDUCIBLE # 1978 8/10
28     IA2=IA0; IB2=IB0+IB1; IC2=IC0; ID2=ID0
29     CALL SETEDG(EDGE,IA0,IB0,IB1,IC0,ID0)
30     #
31     ##### PARAMETER KAB,KCD WO KIMERU #####
32     #      KAB: ATTACHING MAPPING (A J)--->(B J+KAB)
33     #      KCD: ATTACHING MAPPING (C I)--->(D I+KCD)
34     #
35     DO KCD=1,M *(
36         CALL SETKCD(KCD,NLOOP)
37         IF(NLOOP > 2) NEXT
38         DO KAB=1,N *(
39             CALL SETKAB(KAB,NLOOP,ISW)
40             IF(ISW /= JUST2LOOPS) NEXT
41             ICNT1=ICNT1+1
42         #
43         ##### HOMOLOGY 3-SPHERE ? #####
44         #
45         CALL MAKEC
46         IF(ABS(C(1,1)*C(2,2)-C(1,2)*C(2,1)) /= 1) NEXT
47         ICNT2=ICNT2+1
48         #
49         CALL MAKEWD
50         CALL GEORED(ISW,IA0,IB0,IB1,IC0,ID0)      # 1978 8/30
51         IF(ISW == REDUCIBLE) NEXT
52         ICNT3=ICNT3+1
53         #
54         CALL COSET(WORD,LW,ISW)
55         IF(ISW == FALSE) NEXT
56         ICNT4=ICNT4+1
57         CALL OUT
58         *)
59     *)
60     RETURN

```

```

61         END
62
63     #
64     ### GRAPH WO TSUKURU : EDGE (A K) <--> (C J) ###
65     #
66     SUBROUTINE SETEDG(EDGE,IA0,IB0,IB1,IC0,ID0)
67     COMMON CONSTANT
68     INTEGER EDGE(SIZE)
69     KA=MA; KC=MC; KB=MB; KD=MD
70     FOR(K=1; K <= IA0; K=K+1) *(
71         EDGE(KA+K)=KC+K      # (A K) <--> (C K)
72         EDGE(KC+K)=KA+K
73         EDGE(KB+K)=KD+K      # (B K) <--> (D K)
74         EDGE(KD+K)=KB+K
75     *)
76     KA=IA0; KB=MB+IA0+IB0
77     KC=MC+IA0+IC0+ID0+1; KD=MD+IA0+IB1+IC0+1
78     FOR(K=1; K <= IC0; K=K+1) *(
79         EDGE(KA+K)=KD-K
80         EDGE(KD-K)=KA+K
81         EDGE(KB+K)=KC-K
82         EDGE(KC-K)=KB+K
83     *)
84     KA=IA0+IC0; KB=MB+IA0
85     FOR(K=1; K <= IB0; K=K+1) *(
86         EDGE(KA+K)=KB+K
87         EDGE(KB+K)=KA+K
88     *)
89     KC=MC+IA0+IC0+ID0; KD=MD+IA0
90     FOR(K=1; K <= IB1; K=K+1) *(
91         EDGE(KC+K)=KD+K
92         EDGE(KD+K)=KC+K
93     *)
94     KC=MC+IA0; KD=MD+IA0+IB1+IC0
95     FOR(K=1; K <= ID0; K=K+1) *(
96         EDGE(KC+K)=KD+K
97         EDGE(KD+K)=KC+K
98     *)
99     RETURN
100    END

```

```

1  ### SET THE 'VERTEX' BY THE PARAMETER 'KCD' ###
2  #   NLOOP:= NUMBER OF LOOP(MADE FROM 'C'&'D') -1
3  #   VERTEX:= MAPPING OF VERTEX ON GRAPH TO ONE
4  #
5  SUBROUTINE SETKCD(KCD,NLOOP)
6  COMMONCONSTANT
7  COMMONWORK1
8  COMMONOUT
9  DO I=1,M *(
10     VERTEX(I+MC)=K=MD+1+MOD(I+KCD,M)
11     VERTEX(K)=I+MC
12     CHAR(K,2)=I
13     *)
14  IMAX=2*M
15  DO I=1,IMAX
16     CHECK(I+MC)=0
17  #
18  ### EDGE1 WO EDGE,VERTEX NI-YORI SET ###
19  #
20  #           EDGE      VERTEX  EDGE      VERTEX EDGE
21  #           I         I       I         I       I
22  #           V         V       V         V       V
23  #   EDGE,VERTEX: (A I)-->(C J1)=(D K1)-->(C J2)=.,,-->(B L)
24  #   EDGE1      : (A I)-->(B L)
25  #
26  DO I=1,MC *(
27     J=I
28     REPEAT *(
29         J=EDGE(J)
30         IF(J <= MC) BREAK
31         CHECK(J)=1
32         J=VERTEX(J)
33         CHECK(J)=1
34     *)
35     EDGE1(I)=J
36     *)
37  #
38  ### 'C','D' DAKE-KARA-NARU LOOP WO KAZOERU ###
39  #
40  NLOOP=1
41  DO I=1,IMAX *(
42     IF(CHECK(I+MC) /= 0) NEXT
43     NLOOP=NLOOP+1
44     J=I+MC
45     REPEAT *(
46         CHECK(J)=1
47         J=EDGE(J)
48         CHECK(J)=1
49         J=VERTEX(J)
50     *)
51     UNTIL(CHECK(J) == 1)
52     *)
53  RETURN
54  END

```

TIME 12:58

** PAGE 6 **

```

1  *** SET THE 'VERTEX' BY THE PARAMETER 'KAB' ***
2  #   ISW:= RETURN CONDITION
3  #
4  SUBROUTINE SETKAB(KAB,NLOOP,ISW)
5  COMMONCONSTANT
6  COMMONWORK1
7  COMMONOUT
8  DO I=1,N *(
9      VERTEX(I)=K=N+1+MOD(I+KAB,N)
10     VERTEX(K)=I
11     CHAR(K,2)=I
12     *)
13  #
14  ##### LOOP WO 2-TSU MITSUKERU #####
15  #
16  DO I=1,MC
17     CHECK(I)=0
18  DO I=NLOOP,2 *(
19     J=1
20     REPEAT *(
21         IF(CHECK(J) == 0) BREAK
22         IF(J == MC) *(
23             ISW=ONLY1LOOP
24             RETURN
25             *)
26         J=J+1
27         *)
28     JO=J
29     REPEAT *(
30         CHECK(J)=1
31         J=EDGE1(J)
32         CHECK(J)=1
33         J=VERTEX(J)
34         *)
35     UNTIL(J == JO)
36     *)
37  DO I=1,MC
38     IF(CHECK(I) == 0) *(
39         ISW=ATLEAST3LOOPS
40         RETURN
41         *)
42  ISW=JUST2LOOPS
43  RETURN
44  END

```

```

1  ##### MAKE 'LOOP' & 'C' OF 'VERTEX' & 'EDGE'  ###
2  #
3  #   C:= MATRIX      C(1,*)=#(A)-#(B)
4  #                   C(2,*)=#(C)-#(D)
5  #                   #(A):= NUMBER OF 'A' IN THE LOOP
6  #   LOOP:= SEQUENCE OF INTRENAL CODE (VERTEX ON GRAPH)
7  #   LL := LENGTH OF LOOP
8  #
9  #           EDGE VERTEX EDGE VERTEX
10 #            I      I      I      I
11 #            V      V      V      V
12 #   LOOP= (A I),(B J),(A K),(C L),....;
13 #
14 SUBROUTINE MAKEC
15 COMMON CONSTANT
16 COMMON WORK1
17 COMMON WORK2
18 #
19 DO I=1,MN
20   CHECK(I)=0
21 C(1,1)=C(1,2)=C(2,1)=C(2,2)=0
22 DO J=1,2 *(
23   DO I=1,MN
24     IF(CHECK(I) == 0) BREAK
25     K=1
26     LOOP(1,J)=I; CHECK(I)=J
27     REPEAT *(
28       IF (I <= MB) C(1,J)=C(1,J)+1
29       ELSE IF(I <= MC) C(1,J)=C(1,J)-1
30       ELSE IF(I <= MD) C(2,J)=C(2,J)+1
31       ELSE C(2,J)=C(2,J)-1
32       I=LOOP(K+1,J)=EDGE(I); CHECK(I)=-J
33       I=LOOP(K+2,J)=VERTEX(I); CHECK(I)= J
34       K=K+2
35     *)
36     UNTIL(I == LOOP(1,J))
37     LL(J)=K-1
38   *)
39 RETURN
40 END

```

TIME 12:58

** PAGE 8 **

```

1 ##### MAKE 'WORD' OF 'LOOP' #####
2 #      LOOP      ==> WORD
3 #      ... (A *) = (B *) ... ==> 1
4 #      ... (B *) = (A *) ... ==> -1
5 #      ... (C *) = (D *) ... ==> 2
6 #      ... (D *) = (C *) ... ==> -2
7 #      LW:= LENGTH OF WORD
8 #
9 SUBROUTINE MAKEWD
10 COMMON CONSTANT
11 COMMON WORK2
12 #
13 DO J=1,2 *(
14     I=0; KMAX=LL(J)
15     DO K=1,KMAX,2 *(
16         I=I+1
17         IF (LOOP(K,J) <= MB) WORD(I,J)= 1
18         ELSE IF (LOOP(K,J) <= MC) WORD(I,J)=-1
19         ELSE IF (LOOP(K,J) <= MD) WORD(I,J)= 2
20         ELSE WORD(I,J)=-2
21     *)
22     LW(J)=I
23 *)
24 RETURN
25 END

```

```

1  SUBROUTINE GEORED(ISW,IA0,IB0,IB1,IC0,ID0)
2  COMMONCONSTANT
3  COMMONWORK1
4  ISW=REDUCIBLE
5  IF(CHECK(1)+CHECK(N) == 0) RETURN
6  DO I=2,N
7      IF(CHECK(I-1)+CHECK(I) == 0) RETURN
8  IF(CHECK(MC+1)+CHECK(MC+M) == 0) RETURN
9  DO I=2,M
10     IF(CHECK(MC+I-1)+CHECK(MC+I) == 0) RETURN
11 #
12 IF(IB0+IB1 == 0) *(
13     IF(CHECK(MA+IA0+IC0) == CHECK(MB+IA0+1)) RETURN
14     IF(CHECK(MA+1) == CHECK(MB+IA0)) RETURN
15     *)
16 #
17 ELSE IF(IB1+IA0 == 0) *( # TYPE-1
18     IF(CHECK(MA+1) == CHECK(MC+IC0+ID0)) RETURN
19     IF(CHECK(MA+IB0+IC0) == CHECK(MC+1)) RETURN
20     *)
21 # ELSE IF(IB0+IA0 == 0) RETURN # TYPE-2
22 #
23 ISW=REDUCED
24 RETURN
25 END
26
27 SUBROUTINE OUT
28 COMMONOUT
29 COMMONWORK2
30 CHARACTER MOJI(5)
31 DATA MOJI /'-B','-A',' ','+A','+B'/
32 #
33 IPAGE=IPAGE+1
34 IF(IPAGE > 3) *( WRITE(6,640); 640 FORMAT('1'); IPAGE=1; *)
35 WRITE(6,600) IA2,IB2,IC2,ID2
36 600 FORMAT(' A=',I2,', B=',I2,', C=',I2,', D=',I2,
37           /' ***** LOOP *****')
38 IMAX=LL(1);WRITE(6,601) ((CHAR(LOOP(I,1),J),J=1,2),I=1,IMAX)
39 IMAX=LL(2);WRITE(6,601) ((CHAR(LOOP(I,2),J),J=1,2),I=1,IMAX)
40 601 FORMAT(/(' ',10('(',A1,I2,',',A1,I2,')'))
41 WRITE(6,610)
42 610 FORMAT(/' ***** RELATION (LONGITUDE) *****')
43 IMAX=LW(1); WRITE(6,611) (MOJI(WORD(I,1)+3),I=1,IMAX)
44 IMAX=LW(2); WRITE(6,611) (MOJI(WORD(I,2)+3),I=1,IMAX)
45 611 FORMAT(' ',40A2)
46 WRITE(6,620) C
47 620 FORMAT(/' C11=',I3,' C12=',I3,' C21=',I3,' C22=',I3/)
48 RETURN
49 END

```

TIME 12:58

** PAGE 10 **

```

1  SUBROUTINE COSET(WORD,LW,ISW)
2  COMMON /COND/ ICOND                                     # 1978 7/24
3  INTEGER WORD(SIZE,2),LW(2)
4  #####
5  # COSET ENUMERATION METHOD FOR ORDER OF THE GROUP #
6  #####
7  DEFINE(CHARACTER,INTEGER)
8  DEFINE(ORIGIN,3)
9  DEFINE(MAXK0,300); DEFINE(MAXK01,301)
10 DEFINE(TRUE,1); DEFINE(FALSE,0)
11 COMMON /COSET/WTABLE(SIZE,2), # WORKING TABLE
12                                # RTABLE(5,MAXK0), # RELATION TABLE
13                                # UDEFP(2,2), # UNDEFINED POINTER FOR WTABLE
14                                # LCON(3), # LENGTH OF CONDITION
15                                # MAXK # MAXIMUM INDEX OF COSET
16 INTEGER WTABLE,RTABLE,UDEFP,LCON,SET,SETWT
17 DIMENSION AB(3)
18 #
19 ##### WORKING TABLE INITIALIZE #####
20 K=LCON(1)=1
21 DO N=1,2 *(
22     IMAX=LW(N)
23     DO I=1,IMAX *(
24         WTABLE(K,1)=WORD(I,N)
25         K=K+1
26     *)
27     WTABLE(K,1)=0
28     LCON(N+1)=K=K+1
29 *)

```

```

1      DO L=1,2 *(
2          UDEF P(L,1)=LCON(L)
3          UDEF P(L,2)=LCON(L+1)-1
4          WTABLE(UDEF P(L,1),2)=WTABLE(UDEF P(L,2),2)=1
5          *)
6      #
7      ##### RELATION TABLE INITIALIZE #####
8          MAXK=1
9          DO IS=1,5
10             RTABLE(IS,1)=0
11 ##### MAIN PROGRAM #####
12     REPEAT
13         CALL SCHDEF                # SEARCH & DEFINED RELATION
                                     TABLE
14     UNTIL(SETWT(1) == TRUE)      # SET WORKING TABLE
15         MAXK >= MAXKO)
16     IF (MAXK == 1) *(
17         WRITE(6,600); 600 FORMAT(' THIS GROUP IS TRIVIAL !
                                     !!!')
18         CALL OUT
19         STOP 900
20         *)
21     ELSE IF (MAXK < MAXKO) *(
22         ISW=FALSE
23         RETURN
24         *)
25     ELSE *(
26         AB(1)='PUNC'
27         IF (ICOND > 5) *(
28             DO K=1,MAXKO
29                 RTABLE(ORIGIN,K)=K
30             DO K=1,MAXKO,30 *(
31                 IMAX=K+29
32                 WRITE(6,1) ((RTABLE(J,IS),IS=K,IMAX),J=1,5)
33                 1 FORMAT('  +B',3(10I4,3X),
34                     /'  +A',3(10I4,3X),
35                     /'      ',3(10I4,3X),
36                     /'  -A',3(10I4,3X),
37                     /'  -B',3(10I4,3X)/)
38             *)
39         *)
40     *)
41 #
42     DO L=1,2 *(
43         UDEF P(L,1)=LCON(L)
44         UDEF P(L,2)=LCON(L+1)-1
45         WTABLE(UDEF P(L,1),2)=WTABLE(UDEF P(L,2),2)=1
46         *)
47     MAXK=1
48     RTABLE(1,1)=RTABLE(5,1)=0
49     RTABLE(2,1)=RTABLE(4,1)=1
50     RTABLE(3,1)=0
51     IGOMI=SETWT(1)
52     REPEAT
53         CALL SCHDEF
54     UNTIL(SETWT(1) == TRUE | MAXK >= MAXKO)
55     IF (MAXK == 1)
56         AB(2)='TRIV'
57     ELSE IF (MAXK < MAXKO) *(
58         ISW=FALSE

```

TIME 12:58

** PAGE 12 **

```

59         RETURN
60     *)
61 ELSE *(
62     AB(2)='PUNC'
63     IF (ICOND > 5) *(
64         DO K=1,MAXKO
65             RTABLE(ORIGIN,K)=K
66         DO K=1,MAXKO,30 *(
67             IMAX=K+29
68             WRITE(6,1) ((RTABLE(J,IS),IS=K,IMAX),J=1,5)
69             *)
70         *)
71     *)
72 #
73 DO L=1,2 *(
74     UDEFP(L,1)=LCON(L)
75     UDEFP(L,2)=LCON(L+1)-1
76     WTABLE(UDEFP(L,1),2)=WTABLE(UDEFP(L,2),2)=1
77     *)
78 MAXK=1
79 RTABLE(1,1)=RTABLE(5,1)=1
80 RTABLE(2,1)=RTABLE(4,1)=0
81 RTABLE(3,1)=0
82 IGOMI=SETWT(1)
83 REPEAT
84     CALL SCHDEF
85 UNTIL (SETWT(1) == TRUE | MAXK >= MAXKO)
86 IF (MAXK == 1)
87     AB(3)='TRIV'
88 ELSE IF (MAXK < MAXKO) *(
89     ISW=FALSE
90     RETURN
91     *)
92 ELSE *(
93     AB(3)='PUNC'
94     IF (ICOND > 5) *(
95         DO K=1,MAXKO
96             RTABLE(ORIGIN,K)=K
97         DO K=1,MAXKO,30 *(
98             IMAX=K+29
99             WRITE(6,1) ((RTABLE(J,IS),IS=K,IMAX),J=1,5)
100            *)
101        *)
102    *)
103 #
104 WRITE(6,2) AB
105 2 FORMAT(/' NORMAL--',A4,',', COSET(A)--',A4,
106                                     ', COSET(B)--',A4)
107 ISW=TRUE
108 RETURN
109 END

```

```

1  INTEGER FUNCTION SETWT(IGOMI)  * SET WTABLE AS WELL AS
2                                  POSSIBLE
3                                  * TRUE ... CLOSED UP !
4  DEFINE(COMMONCOSET,
5  COMMON /COSET/WTABLE(SIZE,2),RTABLE(5,MAXK0),
6  UDEFK(2,2),LCON(3),MAXK
7  INTEGER WTABLE,RTABLE,UDEFK,LCON)
8  COMMONCOSET
9  INTEGER FULLSW(10),SET
10 #
11 REPEAT *(
12     ISW=TRUE
13     DO K=1,2
14         REPEAT *(
15             IF(SET(UDEFK(K,1),UDEFK(K,2)) == FALSE) *(
16                 FULLSW(K)=FALSE
17                 BREAK
18             *)
19             I=WTABLE(UDEFK(K,1),2)
20             J=WTABLE(UDEFK(K,2),2)
21             IS=WTABLE(UDEFK(K,1),1)
22             IF(RTABLE(ORIGIN+IS,1) /= J)
23                 IF(RTABLE(ORIGIN+IS,1) /= 0) *(
24                     CALL COLLAP(RTABLE(ORIGIN+IS,1),J)
25                     ISW=FALSE
26                     BREAK
27                 *)
28             ELSE IF(RTABLE(ORIGIN-IS,J) /= 0) *(
29                 CALL COLLAP(RTABLE(ORIGIN-IS,J),1)
30                 ISW=FALSE
31                 BREAK
32             *)
33             ELSE *(
34                 ISW=FALSE ***** NEW RELATION SET
35                 RTABLE(ORIGIN+IS,1)=J
36                 RTABLE(ORIGIN-IS,J)=I
37                 *)
38             I=WTABLE(LCON(K),2)
39             IF(I >= MAXK) *( ***** THIS TABLE IS CLOSED
40                                     UP !
41                 FULLSW(K)=TRUE
42                 BREAK
43             *)
44             UDEFK(K,1)=LCON(K)
45             UDEFK(K,2)=LCON(K+1)-1
46             WTABLE(UDEFK(K,1),2)=WTABLE(UDEFK(K,2),2)=I+1
47             *)
48         *)
49     UNTIL(ISW == TRUE)
50     #### ANSWER --> SETWT ####
51     SETWT=TRUE
52     DO K=1,2
53         SETWT=SETWT*FULLSW(K)
54     RETURN
55 END

```

TIME 16:45

** PAGE 14 **

```

1      INTEGER FUNCTION SET(IP,JP) # TRUE ... OK, THIS TABLE IS
                                           FILLED UP
2      COMMONCOSET
3      WHILE(IP+1 < JP) *(
4          I=WTABLE(IP,2)
5          IS=WTABLE(IP,1)
6          IF(RTABLE(ORIGIN+IS,I) /= 0) *(
7              IP=IP+1
8              WTABLE(IP,2)=RTABLE(ORIGIN+IS,I)
9              NEXT
10         *)
11         J =WTABLE(JP,2)
12         JS=WTABLE(JP-1,1)
13         IF(RTABLE(ORIGIN-JS,J) /= 0) *(
14             JP=JP-1
15             WTABLE(JP,2)=RTABLE(ORIGIN-JS,J)
16             NEXT
17         *)
18         SET=FALSE
19         RETURN
20     *)
21     SET=TRUE
22     RETURN
23     END
24
25
26
27
28
29     SUBROUTINE SCHDEF          # SEARCH & DEFINED
30     COMMONCOSET
31     #### SEARCH ####
32     MINI=MAXK+2
33     DO I=1,2
34         IF(UDEFPP(I,1)+1 /= UDEFPP(I,2))
35             IF(MINI > WTABLE(LCON(I),2)) *(
36                 L=I
37                 MINI=WTABLE(LCON(I),2)
38             *)
39     #### DEFINED ####
40     MAXK=MAXK+1
41     DO IS=1,5
42         RTABLE(IS,MAXK)=0
43         I=WTABLE(UDEFPP(L,1),2)
44         IS=WTABLE(UDEFPP(L,1),1)
45         RTABLE(ORIGIN+IS,I)=MAXK
46         RTABLE(ORIGIN-IS,MAXK)=I
47     RETURN
48     END

```

FACOM 230-45S RATFOR

DATE 78-09-16 ** PAGE 15 **

```

1  SUBROUTINE COLLAP(I0,J0) # COLLAPSING I0<=>J0
2  COMMON COSET
3  DIMENSION ISTACK(MAXK,2)
4  I=I0; J=J0
5  IF(I < J) #( I0=I; I=J; J=I0; *)
6  CALL RNUM(I,J)
7  # WRITE(6,600) I,J; 600 FORMAT(' COLLAPSING',I3,' = ',I3)
8  #
9  ISTN=0
10 REPEAT #(
11     DO IS=1,5
12         IF(RTABLE(IS,I) /= RTABLE(IS,J)
13            & RTABLE(IS,I) /= 0)
14             IF(RTABLE(IS,J) == 0)
15                 RTABLE(IS,J)=RTABLE(IS,I)
16             ELSE #(
17                 I1=RTABLE(IS,I)
18                 J1=RTABLE(IS,J)
19                 IF(I1 < J1) #( I0=I1; I1=J1; J1=I0; *)
20                 CALL RNUM(I1,J1)
21                 # WRITE(6,600) I1,J1
22                 ISTN=ISTN+1
23                 ISTACK(ISTN,1)=I1
24                 ISTACK(ISTN,2)=J1
25                 *)
26         RTABLE(ORIGIN,I)=J
27         IF(ISTN == 0) BREAK
28         I=ISTACK(ISTN,1)
29         J=ISTACK(ISTN,2)
30         ISTN=ISTN-1
31     #)
32 I=MAXK
33 CALL GBGCOL # GERBAGE COLLECTOR
34 # WRITE(6,601) I,MAXK
35 # 601 FORMAT(' GERBAGE COLLECTOR ! MAXK:',I3,'->',I3)
36 RETURN
37 END

```

TIME 12:58

** PAGE 16 **

```

1      SUBROUTINE GBGCOL      # GERBAGE COLLECTOR
2      COMMONCOSET
3      DIMENSION NEWNUM(MAXK01) # NEW NUMBERING
4      #
5      ##### 'NEWNUM' SET #####
6      J=NEWNUM(1)=0
7      DO I=1,MAXK *(
8          IF(RTABLE(ORIGIN,I) == 0) J=J+1
9          NEWNUM(I+1)=J
10         *)
11     #
12     ##### RTABLE REWRITE #####
13     J=1
14     DO I=1,MAXK *(
15         IF(RTABLE(ORIGIN,I) /= 0) NEXT
16         DO IS=1,5
17             RTABLE(IS,J)=NEWNUM(RTABLE(IS,I)+1)
18             J=J+1
19         *)
20     MAXK=J-1
21     #
22     ##### WTABLE REWRITE #####
23     DO K=1,2 *(
24         UDEFK(K,1)=LCON(K)
25         UDEFK(K,2)=LCON(K+1)-1
26         WTABLE(UDEFK(K,1),2)=WTABLE(UDEFK(K,2),2)=_
27             NEWNUM(WTABLE(UDEFK(K,1),2)+1)
28     *)
29     RETURN
30     END
31
32
33     SUBROUTINE RNUM(IO,J0) # RENUMBERING IO->J0
34     COMMONCOSET
35     DO I=1,MAXK
36         DO IS=1,5
37             IF(RTABLE(IS,I) == IO) RTABLE(IS,I)=J0
38     RETURN
39     END

```