U.D.C. 72.011

清水建設研究所報
第5号昭和40年4月

直交図面の記号法と電算処理 (3)

清 水 達 雄

目 次

4. 最小寸法図の自動図示(続)

§ 17. 階数計算

前号で，長方形分割図の，2進記号法からの，その最小寸法図の，電子計算機による自動図示について，正則な場合に限定しつつ，プログラム第1部の，ALGŌL 的記述までを，解説した.

ただしここで，長方形分割図とは，長方形を，その縦横の辺と平行な線分で，いくつかの長方形に，分割した構図をいう. うち，分割線の，十字交差のないものを，正則とよぶ. 一般には，実数の範囲で，計量をもつわけだけれども，とくに寸法が自然数，つまり方眼紙上に描けるもので，最小なもの. といっただけでは，確定しないけれども，そのようなのを，最小寸法図とよぶ. くわしくは，前々号を見られたい. 基本的には，計量を離れた，位相的な構造，に対応する概念といえる.

その，長方形分割図の位相的構造を，区別して記述する，2進記号法に，方式は，なおいろいろとある. この「所報」1～2号にわたってのべた，転置列・接着列の併記によるものも，一法に数えられよう. しかし同3号にのべたものを，ここでは指す. 両記号法の関係については，別に考えてみたい. あとの，新しいほう，これは図の表現に必要な，記号法上のビット数の，節約を指針に，数年来，改良してきての，一応の結論に相当する.

追記 ただし，十字交差の表現については考察不足で

前枝／後枝　でなく　前枝＼後枝

の特殊の場合とみなすほうが，自動図示が簡単になる.

さて，前号の続きとして，プログラム第2部の，解説に進みたい．第1部で設定された，準順序相当の，

$\bar{O}[I,\ J]=0$　または　1

の関係から，線の階数を算出する．一般に，

定義47　線 I の階数を，$R[I]$ で表わす．

まず，当然に，上辺と左辺については

$\left.\begin{array}{l} R[0]=0 \\ R[1]=0 \end{array}\right\}$

そのつぎ，これら前辺に，直接おくれるもの，つまり

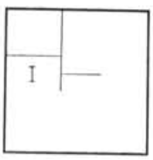
$\bar{O}[$前辺$,\ I]=1$,　前辺$/I$

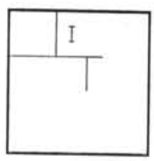
は当然として，

前辺$/X/I$

のような X の存在しないもの．そのような I については

$R[I]=1$





以下，おなじようにして，

$R[H]=r-1$

に直接おくれるもの，つまり

H/I

のような I で，しかし

$H/X/I$

のような X の存在しないもの．そういう I については

$R[I]=r$

このようにして，階数 R がきめられてゆく．

これを，集合論の言葉で，のべかえておこう．

定義48　線全体の集合を $\mathfrak{L}$ とし，そのうちで

階数 r 以下の線の集合を　$\mathfrak{L}^r$

〃　以上　〃　　$\mathfrak{L}_r$

で表わす．当然に

$\{0,\ 1\}=\mathfrak{L}^0\subset\mathfrak{L}^1\subset\mathfrak{L}^2\subset\cdots\cdots$

$\mathfrak{L}=\mathfrak{L}_0\supset\mathfrak{L}_1\supset\mathfrak{L}_2\supset\mathfrak{L}_3\supset\cdots\cdots$

また，$r-1$ 以下でないのが，r 以上だから

$\mathfrak{L}^{r-1}$ の補集合$=\mathfrak{L}_r$

定義49　$\mathfrak{L}$ の部分集合 $\mathfrak{A}$ の元 I で，

X/I

のような，$\mathfrak{A}$ の元 X の存在しないものを，$\mathfrak{A}$ での**局大元**とよぶ．

補題　$\mathfrak{L}_r$ の局大元が，ちょうど，階数 r の線．

この補題にしたがって，階数計算のプログラミングを試みよう．$\mathfrak{L}^{r-1}$ まで定まったとして，この範囲に X が

属していれば　$RR[X]:=0$

属しなければ　〃　$:=1$

こうおいて，$\mathfrak{L}_r$ 内の I に対し，つぎの積を考える．

$RR[X]\times\bar{O}[X,\ I]$　　X は変数

これは，X が $\mathfrak{L}^{r-1}$ に属していれば，RR が0で，0．反対に，X が $\mathfrak{L}_r$ のほうに属していれば，$\bar{O}$ そのままで，

X/I　の場合だけ1，あとは0

だから，どの X についても0の I が，$\mathfrak{L}_r$ の局大元．

そういう I の階数を r とし，さらに局大元のしるしに

$M[I]:=1$,　ほかのには $:=0$

こうして，ひと通り I を調べたあと

$RR[X]:=RR[X]-M[X]$

こうすると，こんどは，階数 r まで，つまり $\mathfrak{L}^r$ に X が

属していれば　$RR[X]=0$

属しなければ　〃　$=1$

となって，手順がひとつ進む．

はじめに準備として

for $I:=0$ step 1 until $N+2$ do

$RR[I]:=1$;

$R:=0$;

ただしここで，N は部分長方形数．それから

R1:　for $I:=0$ step 1 until $N+2$ do

その，begin … end の中味として

if $RR[I]=0$ then go to R2;

for $X:=0$ step 1 until $N+2$ do

if $RR[X]\times\bar{O}[X,\ I]=1$ then

go to R2;

$R[I]:=R$;　$M[I]:=1$;

go to R3;

R2:　$M[I]:=0$;

R3:　空;

それから

for $X:=0$ step 1 until $N+2$ do

$RR[X]:=RR[X]-M[X]$;

for $X:=0$ step 1 until $N+2$ do

if $RR[X]=1$ then

begin $R:=R+1$;　go to R1　end;

これで，RR すべて0のとき，終了する．

§18. 語のビット分割

ところで，上記のプログラムに出てくる

$RR[X]$，$M[X]$，また $\bar{O}[X, I]$

というのは，その値が

0 または 1

に限られており，2進1桁，1ビットの内容しかない．それに，1語を割り当てるのは，もったいない．この語の長さとくらべて，部分長方形数Nは，いく分か小さいものとしよう．くわしくは

数値部の桁数≧線の総数$=N+3$

そうすれば

$X=0,\ 1,\ \cdots,\ N+2$

に対する値を，1語のなかに収めてしまえる．たとえば

$M[X]$ を 語Mの，数値部第$X+1$桁に

入れればよい．そうするとして，たとえば

$M[2]=M[3]=M[5]=1$，あとは$=0$

のようなのは

Mの数値部$=0011010\cdots0$

こういう，入れ方，またそれからの読み出し方を，のべるためには，しかし，ALGŌL に，補足がいる．まず，おなじ1でも，位取り位置がちがうことを，指す方法．数値部第q桁目の1のことを

$1@q$

で表わそう．もっと一般的に

位取りの指示 数値部第q桁目の，すぐ右に，小数点位置が合うように入れた，数Aのことを

$A@q$

で表わす．@0を標準と考えて，くらべれば

$A@q=(A@0)/2\uparrow q$

たとえば，上記の

$M=1101(2進)@6=13(10進)@6$

$=1@3+1@4+1@6$

このMを作るのに，

$M:=0;$

から始めて，Xを増加させながら

$Q:=1@(X+1)$

とおいて，$M[X]:=1$ に相当する場合に

$M:=M+Q$

とすればよい．ただし，@の右に変数が来るのを，この場合は避けられるから，もひと工夫しよう．

関数の定義 以下の記述に，つぎの関数を使用する．

SHIFT $(A,\ B)$	AをB桁，右にずらす．Bが負なら左にずらす．
EXTR $(A,\ B)$	AとBとの，同桁目ごとの積を，その桁におく．ビットごとの積．

この SHIFT を使えば，X増加ごとに

$Q:=$SHIFT $(Q,\ 1)$

とすればよい．EXTR のほう，ある語から，たとえばその第I桁目をとりだすのに，よろしい．つまり

EXTR $(A,\ 1@I)=A$のI桁目．

もっと一般に，第 $I, J, \cdots$桁同時とりだしに使える．

こうした記号を使うと，階数計算の部分は，たとえば

```
      Q:=1@1;  R:=RR:=0;
      for I:=0 step 1 until N+2 do
      begin  RR:=RR+Q;
             Q:=SHIFT (Q, 1)  end;
R1:   Q:=1@1;  M:=0;
      for I:=0 step 1 until N+2 do
      begin
      if EXTR (RR, Q) ≠0 then
      if EXTR (RR, Ō[I])=0 then
      begin  R[I]:=R;
             M:=M+Q  end;
      Q:=SHIFT (Q, 1)  end;
      RR:=RR-M;
      if RR≠0 then
      begin  R:=R+1;  go to R1  end;
```

この最初の4行は，本当は，第1部で実行する．もともと前後関係$\bar{O}$は，第1部で設定した．そこのたとえば

$\bar{O}[I,\ W]:=1;$

が，こんどは，$Q=1@(I+1)$ として

$\bar{O}[W]:=\bar{O}[W]+Q;$

に変わる．このQ作りを追加，RRもそこで算出する．

それから，逆向きの V/I のたぐいの関係設定

$\bar{O}[V[J],\ I]:=1;$

のためには，Vを$1@(V+1)$にする，サブルーチンがいる．便宜上，Vの代りにWとかいて，ビット位置化

```
      integer procedure X(W); value W;
      begin  X:=1@1;  go to B2;
B1:   X :=SHIFT (X, 1);
B2:   W:=W-1;
      if W≥0 then go to B1 end;
```

§19. 線要素の割り付け

さて，階数そのものについても，一般的に

$R[I] \leq N$

ただし，Nは部分長方形の数．というのは，階数が

$0,\ 1,\ 2,\ \cdots,\ R$

の，たとえば縦線があったとすると，それらの間に，部分長方形がR箇はあるはずだから．あるいはまた

線の総数$=N+3$

のうち，縦線・横線とも，周の4辺の分，最少限2本はあるのだから

縦線数，横線数$\leq N+1$

そうして，たとえば

縦線の階数≦縦線数$-1 \leq N$

つぎに，席数について考えると，一般には

$S[I] \leq I$の後枝数

特別な場合として，前辺，$I=0,\ 1$ に対しては

$S[I] \leq \infty$

ただし∞は，充分大きい数の略記，だった．くわしくは

後枝がつくごとに，席数Sを1へらす

算法で，負にならないようにとる．前辺の場合，

左辺は上辺の後枝，上辺は左辺の後枝

の関係は，一般算法とは別に最初に設定するから，

つく後枝数=後枝数$-1 \leq \infty$

にとればよい．そうして当然に

後枝数≦横線数または縦線数$\leq N+1$

だから，

$N \leq \infty$ にとれば充分

ふつうの内部分割線Iの場合は，それが縦線なら

縦線数≥ 3

となって

横線数$\leq N$

そのうち上辺と下辺は，後枝でないから，Iの

後枝数≦横線数$-2 \leq N-2$

したがって

$S[I] \leq N-2$

前辺の場合をふくめていえば

$S[I] \leq \infty$，∞はN以上のある定数

さて，プログラム第1部では，また

$B[I]$，$E[I]$，$V[I]$，$W[I]$

の算出をした．これらは，なにかの線番号で，その

線番号$\leq N+2$

取り扱う範囲　以下，つぎの制限をもうける．

部分長方形数$N \leq 29$

また便宜上，つぎのようにおく．

$\infty = 31 = 2^5 - 1$

この範囲では，

$S[I]$，$B[I]$，$E[I]$，$V[I]$，$W[I]$，

$R[I] \leq 31$

つまり，どれもが2進5桁，5ビットで表わされる．

それが6箇あるから，みなで30ビット．それだけの，語長があれば，1語に割り付けてしまえる．もっとくわしく，25ビットで，以上には足りる．というのは，プログラム第1部末のところで，すべての

$S[I]=0$

に落ち着き，この5ビットの空きが，第2部で使える．

第1部の　$S[I]$

第2部の　$R[I]$

には，おなじ部分を割り当てよう．

その一方，もうひとつ，線方向を示す

$D[I]$

があった．これは1ビットのもの．おなじく1ビットの

$G[I]$，$F[I]$，$U[I]$，$C[I]$

を，以下で導入する．十字交差もある一般の分割図を，扱うためのもの．これらを合わせて，1ビットが5箇．総計，やはり30ビット．それを1語に盛りこもう．いま

数値部の長さ≧30

として，プログラムの便宜も考え，

線要素の割り付け

ビット番号	
1〜5	$S[I]$ のちに $R[I]$
6	$G[I]$
7〜11	$B[I]$ または $H[I]$
12	$F[I]$
13〜17	$E[I]$
18	$U[I]$
19〜23	$V[I]$
24	$D[I]$
25〜29	$W[I]$
30	$C[I]$

以上のように割り付けた1語を，つぎのように表わす．

$BER[I]$

なおここで，$H[I]$ は，I の頭線を示すが，くわしくは，のちにゆずる．

§ 20. 入力記号の長語分析

もうひとつ，検討すべきことがある．そもそものはじめ，入力として与えられたのを

$Q[1]$，$Q[2]$，…

とした．一般的には

$Q[M]$

しかしその内容は，0 または 1．だから，1 語でなくて 1 ビットずつを割り当てればよい．Q という語の数値部

第 M ビットに，$Q[M]$

しかし，この M は，相当に大きくなる．部分長方形数

$N \fallingdotseq$ 語長

とすれば，とても 1 語には，おさめきれるものでない．

入力 $=Q[M]$ のつづけ書き

は，正則用記号として

$=DA\langle I\rangle$ のつづけ書き

$=D\langle I\rangle A\langle I\rangle$ のつづけ書き

ただしここで，

$I=2, \cdots, N$

$D\langle I\rangle$ は，1 ビット

$A\langle I\rangle$ のビット数 $=$ 枝の数 $+1$

ところでこの，枝の数は，分割線 I 上の

分割点（両端を除く）の数

それを I について加えれば，分割図の

内部分割点の数

これに外部分割点を合わせた

全分割点数 $=$ 分割線数 $\times 2=2(N-1)$

だからいま

外部分割点数 $=K$

とすれば

全枝数 $=2(N-1)-K$

これになお

$A\langle I\rangle$ の最終ビット，$D\langle I\rangle$

の 2 ビットずつが加わるから

入力の長さ $=4(N-1)-K$

定理　入力の長さは，部分長方形数を N とすると

正則用のとき　$4(N-1)-K$

一般用のとき　$5(N-1)-K$

ただし K は，外部分割点数で，$N\neq 1$ なら

$K \geq \mathrm{Min}(4, N)$，等号もなりたつ

この不等式の証明は，いまはぶく．

ともかくそれで，数値部の長さが 30 ならば，入力は，$N=29$ で 4 ないし 5 語にわたる．記号を改め，その語を

$Q[1]$，$Q[2]$，…

で表わそう．その

$Q[1]$ の数値部，第 1〜30 ビット

$Q[2]$　〃　〃

………………………………………

に，順に入力ビットを盛りこむことにする．この場合，どこまで続くのかを，別に指示しなければならない．

語の構成　数値部30ビットに先立つ，第 0 ビットが

符号ビット

で，1 のとき負数を，補数表示で表わす．

この符号ビットを利用し，数語にわたる入力の，その

末語の符号ビットに 1

を付加する．その

末語の数値部中の最終の 1 が，**語末**

ただし，特別な場合として

$N=1$，単一長方形そのもの

に対しては，分割線がなく，記号数 0 で，末語は

符号ビットが 1，数値部は 0

この場合は，取り扱い範囲から除外しよう．数としては

$-1@0$

それを，課題終了の記号にする．つまり，数語にわたる入力記号の，さらにいくつかの系列を，順々に処理し，上記の終了記号で停止する，プログラムを頭におく．

入力記号のビット分析の部分としては，はじめに

```
         M: =-1;
A0:      L: =0;  P: =-1;
```

としておいて，分析したいたびごとに，go to の

```
P0:      if  L≥0  then go to  P1;
         if  T の数値部  SHIFT (T, 1)=0
         then go to  L7 後辺特殊段取り;
P1:      P: =P+1;
         if  P<0  then go to  P2;
         P: =-30;  M: =M+1;
         T: =L: =Q[M];
         if  T=-1@0  then go to END;
P2:      T: =SHIFT (T, 1);
```

と先頭ビットを符号ビットに入れて，1 か 0 か，つまり

```
         if  T<0  then  Q: =-1@5
                  else  Q: =1@5;
```

それから，go to $SW[*]$ とでもして，本体へ返す．なお，方向分析以外のときは，P1 からやればよい．

§21. 一般の場合，頭線化

さて，もうこの辺で，一般の場合に，論を進めよう．線の十字交差をゆるし，一般用記号を，入力とする．追加される情報は，1ビットの

後続符 （定義28, 3号49ページ）

そうして，新しく活動する中心的概念は

頭線 （定義31, 同号50ページ）

これを，図形の直観にうったえずに，算法のかたちで，定義し直さなければならない．

もしも，記号法の組み立てに

後続符 でなく 前続符

を採用したのだったら，それで頭線かどうかがきまる．そう定めなかったから，前へ続くかどうかを判定する，算法が，必要になる．第1部の前半，新加入の線 I の

起線分 J

が見つかり，その段階での J の

副待機線 $V[J]=V$

を引き出したところで，場合をわけて考える．その V の

後続符 $C[V]$ （後述）

が0，つまり V が後続していなければ，当然に I は前続をしない．しかし，それが1，つまり V が後続するときも，それに I が前続する，とはかぎらない．というのは

$V=B$, B は J の起線分

の状態のとき，V に後続するのは，I ではない．また

V が J の後枝

のときも，I ではない．I が前続するのは

V が J の前枝

の場合にかぎられる．

こうした場合の区別は，たとえば

$E[V]$ の有無

で判定されよう．しかし V の線要素をいちいち探らず，J のところで決着する方法として，1ビットの

$U[J]$

を設定しよう．それは，はじめ

$V:=B$ では $:=0$

前枝のなかからの V 定め

$V:=$ のとき $:=C[V]$

後枝がついての

$V:=$ のとき $:=0$

そうしておけば，ちょうど

$U[J]=1$ のとき I は前続

なお，このように前続する後枝 I がついたとき，変更

$V:=I$

は行わなくてよいし，いま行わないことにする．ただし

$U:=0$

だけは，やっておく．こういうようにして，I が前続，つまり頭線でない場合と，頭線の場合とを区別する．

定義50 この区別を示すため，1ビットの

$G[I]:=0$, I 自身が頭線

$:=1$, I は頭線でない

定義51 頭線でない I については，それの頭線を

$H[I]:=H(V)$

と，V から引きつぐ．ただし右辺は

```
integer procedure  H(V);  value V;
begin if  G[V]=0  then  H(V):=V
else  H(V):=H[V]  end;
```

定義52 自身が頭線の I について，それの起頭線を，

$B[I]:=H(J)$

ただし J は，I の起線分．

正則な場合には，単に

$B[I]:=J$

としたところを，このように，さらに頭線化する．また

$I\diagdown V[J]$

満席の場合の

$I/W[J]$

のところも，頭線化をして

定義53 自身が頭線の I について

$I\diagdown H(V[I])$

満席の場合，$W[J]\neq J$ ならさらに

$I/H(W[I])$

として，線の準順序関係を定める．

もっとも実際には，ビット位置化しての

$\bar{O}[I]:=X(H(V[J]))$

$\bar{O}[W]:=\bar{O}[W]+(1@I+1)$

をする．ただしここで

$W=H(W[J])$

このように，$\bar{O}$ は

頭線相互間の関係

だけに，しぼって定める．それで第2部でも，階数 R は頭線についてだけ算出し，つぎの判定が加わる．

```
if  G[I]=0  then  …  else  …
```

§22. 尾線での終線変更

一般化に必要な，算法の修正は，もうひとと種類ある．もともと第1部は，ひとつの I について

前半　起線探し　（定義38）

後半　終線与え　（定義39）

からなり，いま考えたのは，その前半に属する．後半の

$E[J]:=I$

まず，これも頭線化して

$E[J]:=H(I)$

この右辺，H で，J が終わる．それが実は十字交差で，H を越えて後続するのかどうか．この区別は，後続符

$C[J]$

が，そのまま示している．これを使って

定義54　J が I で終るとき，一般に

$E[J]:=H(I)$

この J が後続しないとき，改めて

$E[H(J)]:=H(I)$

ただし J 自身が頭線のときは，この書きかえをしない．

というのは，このとき

$H(J)=J$

だから．さらに，この書きかえに関して

定義55　はじめに，1ビットの

$F[K]:=0$

この K の終線の変更をしたときに

$:=1$

この区別を示す F を，第1部の前半で使用する．というのは，満席で待機線 W を変えるところ，N2以下で

$E[K]=J$　で　$S[K]>0$

を探している．一般の場合

$E[K]=H(J)$

になるわけだけれど，そのほかに，書きかえてない

$F[K]=0$

という条件がいる．その判定を追加しないといけない．

なおそれから，第1部後半の，

後枝なら　$V[J]/I$

ここも右辺を頭線化し，$/H(I)$　とする．左辺の V は頭線に限定しないとうまくないが，それは，前続の後枝では V を変更しない，としたからよい．

§23. 機内図示

以上，かけ足になってしまったけれど，一般の場合を扱う算法は，これで材料がととのった．実行の結果は

$BER[I]$，$\bar{O}[I]$

のところに残る．とくに頭線 I に対して，BER 中に

方　　向　$D[I]$

階　　数　$R[I]$

起 頭 線　$B[I]$

尾終頭線　$E[I]$

が出る．これだけから，最小寸法図が，えがかれる．

それをまず，計算機内に図示しよう．記憶装置内に，順に並んでいる語

$VIS[L]$　略して　$V[L]$

を，1語おきに，横線と縦線とに配当する．横線部は

$L=1$　に　上辺

$=3$　に　階数1の線

一般に，階数 R の線を

$L=2\times R+1$

のところにかく．ただしこの

かく＝ビットを1にすること

その1にする範囲は，いま

R_0＝起頭線の階数

R_1＝尾終頭線の階数

とするとき

$@R_0+2$　から　$@R_1+1$　まで

の，合計して

R_1-R_0 ビット

これだけの1が並んだものは，数として

$(1@R_0+1)-(1@R_1+1)$

とひとしい．これを

$V[2R+1]$

に，加えてやる．ただし，はじめにすべての

$V[L]:=0$

つぎに，縦線のほうは，偶数番地を利用する．それの

$@R+1$

のところを，1にしてやってゆく．範囲は

$L=2\times R_0+2$

$=2\times R_0+4$

……………

$=2\times R_1$

の，合計して

R_1-R_0 語

いまの，範囲のきめ方から，左辺の記入位置は

$L=2,\ 4,\ \cdots$　の@1

このように

$L=0$

に当たるところには，何の記入もされないが，はじめに

$V[0]:=0$

をやっておく．

一方，どこまで記入がされるかというと

下辺の階数　RL

右辺の階数　RR

としたとき，左・右辺のが

$L=2,\ \cdots,\ 2\times RL$　の@1

と@$RR+1$

そうして下辺のが

$L=2\times RL+1$　の@2～$RR+1$

そうして，階数の

$RL,\ RR\leq N\leq 29$

だったから

$L\leq 59$　で@は≤ 30

そこでプログラムとしては，はじめに

```
      for L := 0 step 1 until たとえば 63
      do V[L] := 0;
```

としておいて

```
      for I := 0 step 1 until N+2 do
      begin if G[I]=1 then go to V2;
      if D[I]=1 then go to V1;
      X := X(R[I]);
      L := 2×R[B[I]]+2;
V0:   V[L] := V[L]+X;
      L := L+2;
      if L≤2×R[E[I]] then go to V0;
      go to V2;
V1:   X := X(R[E[I]]);
      L := 2×R[I]+1;
      V[L] := V[L]+X(R[B[I]])-X;
V2:   end
```

以上が，第3部の機内図示．

これを実行した直後の

$X=1$ @$RR+1$

$L=2\times RL+1$

これが，図示範囲の右下隅に相当する．そういう範囲にある1を，線として機外に図示するのが，つぎの第4部の仕事になる．そこで必要となるから，上記の値を

$XX,\ LL$

で表わしておこう．

§24. 機外打ち出し

機外図示には，附属電動タイプライターを使用する．それは，図示機としては，あまり適切なものではない．なによりの制約として，

下から上へは戻れない

だから，図の上の方を全部すませてからでなければ，下へ移れない．縦線を1本ごとに引く，方式はとれない．そのために，第3部，機内図示を設定した．その上では

ます目なりに打ち進む

機構が，ぴったりと役に立つ．プログラムは簡単で

```
      UC 上段にして CRLF;
      L := 1;
P0:   X := 1@1; CRLF;
P1:   if EXTR (V[L-1], X)≠0 then
      PRINTS (]♮);
      if EXTR (V[L], X)≠0 then
      PRINTS (_) else
      PRINTS (#);
      X := SHIFT (X, 1);
      if X≥XX then go to P1;
      L := L+2;
      if L≤LL then go to P0;
      go to 次の図の処理;
```

ここで，PRINTS は，かっこ内のものの印刷で

_　アンダーラインは，横線単位

]　角がっこの右側は，縦線単位

それから ALGŌL の記号として

\#　スペースをする

さらに臨時の記号だけれど

♮　バックスペース

以上で，どうやら，最小寸法図の自動図示について，機械語プログラムに進める，準備ができた．その肝心のプログラムに，しかし不注意の直し残しがまだあった．それを，つぎのページで，訂正しておく．重大なのは，第4部の機外打ち出しのところ，3号のでいうと

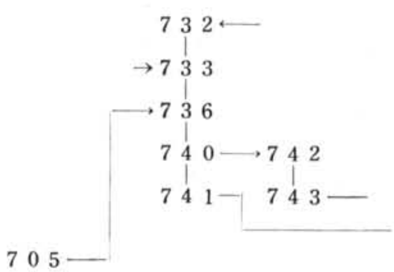
734～7　図完成テスト

命令語の番地部を比較したつもりが，機能部がちがっていた．それと716の，番地部の誤り．ほかは，逐語解の解説中の誤記などで，さしたことはない．

なお，第3部では3語が節約でき，ビット位置化サブルーチンを2語まして，$N\leq 29$が本当に扱えるようにできる．あのままでは，実は≤ 27までとなる．

0211		*j*
0212		1@18
0410		*i*
0509		1@30
0527		C0832＋N＋3
0540		C0801
0662		B0800＋N＋3
0704		B0901
0705	U0736	
0716	S0744	1@30
0718		—
0732		*CRLF*
0734	B0714	
0735	A0730	2@29
0736	Y0714	
0737	S0729	1@29
0738	Y0706	
0739	S0622	B0902＋2×RL
0740	T0742	
0741	U0002	

70ページ



044～57
この f を　　　209～10

53ページ，右下ノ図.
352—741—743

152～6
で通過，314
157～63
足す138

232～7　1行ヲ追加スル.
232，数値として704で使う.

***J* 探し，準サブルーチン**　313～29
〃　通常　　　433

314～9　1行ヲ追加スル.
314，数値として155で使う.

***S* 計算・*W* 定め**　332～55
一般　　　433

61ページ，右下ノ図　434—435

521～8
232～7，434～5

622～31
622，　　　739で

704～5　　　900が

尾部　732～43
入口　部首　736
732～3
741から002へ，
734～8　*I* 進め．714は横線番地，2進める．1引いて縦線番地．
739～41　図完成テスト．622に残留の縦線記入用，最後の番地は，右辺についての

$$900+2\ (R[E[N+1=\text{右辺}]]+1)$$
$$=902+2\times RL$$

742～3　*I* 進め，E積用961を1@1に戻す．

5. 問題の展開

§25. 入力の過誤

ところで，以上に説いてきた，プログラムは，いわば理論上のもので，実用——というより，公開実験のためにも，たりないところがある．計算機の誤動作は別としても，外部から人間が与える，入力の，誤りを考えに入れてない．この点をすこしく補い述べてみよう．

まず，書式上の誤りとして，何語かにわたる記号列の末語の印，負号を欠いた場合．このとき計算機は，記号列の終末を認識できぬまま，それを越えて，無意味な計算を続行する．これは暴走だから，どういうことになるか，いちがいにはいわれない．末語を

+＊…＊0…0

とすれば，空白の0…0の部分で，架空の分割線を設定し，まず方向は横，などとし，また0を数えて区間席数にしたりする．つぎの語が，たまたま

+0…0

なら，その0も算入し，席数に30を加えてしまおうとする．ただし，和が32になると，@5の計算だから，加法のオーバー・フローで，停止する．末語のつぎの語

±＊…＊

しだいでは，もっといろいろな計算をしよう．負の語がでてこないと，どこまで走ってしまうか，わからない．

この暴走を防ぐには，語数に制限をおくとよい．部分長方形数Nをおさえれば，記号列の最長の長さ，そして語数の最大値もきまることは，すでに述べた．それを，突破してしまったら，入力の誤りとするような，命令語群をつけ加えるとよい．

さらに，この種の見当違いの計算に対処して，線番号の過大を，すぐと検知する部分が，ぜひと要る．第1部のはじめのほう，I進めのところに追加する．たとえば

019～21のB，M，Hで1@$I+1$作り

のあとに，つぎの2命令を組み入れる．

1@30のS，適当なところへのT

そうして，後辺段取りの進行状態を吟味する，など．

線番号の過大で，一番困るのは，線がやたらに前辺にくっついて，そこに仮設しておいた席数が，ついに0となったときのこと．そうなると，起線探しの部分で，

$J=I-1$，$I-2$，…，1，0

でも，見当たらないとあって，Jが負にもなる．すると

$I=0$　　800番地　　要素表の先頭

のまえにある

$J=-1$　　763番地　　プログラム末尾

で，プログラム自体にとびこみ，しかも，偶然にも

$J=-2$　　762番地　　左辺要素の初期値

-3　　761番地　　上辺　〃

だから，このどちらかを，起線として発見する．発見するだけでなく，その席数を1へらす．プログラム自体を変改する．さらに被害が，760番地以前に拡がれば，いよいよいけないことになる．交通標識を任意に書きかえたあとみたいなもので，どこへとんでいって，何をしでかすか，わからない．

一方，線番号の過大を防いでおけば，重大なことは，まず起らない．その事情を，プログラムを追いながら，説いてみたかったのだけれども，いまは止めておく．ひとつには，期限超過の関係から．またひとつには，永い断続的な執筆の間に，別の問題や方針がでてきたから．

プログラムを，もすこし拡充して，

かってな2進語群に対し，それが長方形分割図に正しく対応するものなら，長方形数の制限下で，最小寸法図などを打ち出すが，そうでなければ，誤りなどを判定する

ことは，それほどめんどうでない．一般用記号では，

後続符の残り

も生ずるが，正則用に限定すると，主要な誤りは

席数の残り

いいかえれば，

分割線の不足

さらに進んで，

定理　どんな2進有限列も，ある正則分割図の正則用記号の，左方部分として解釈できる．

正則用記号は，0と1の間に，稠密に分布する．

ただしもちろん，可付番だけれども，それをまた順々に枚挙してゆく，算法も，原理的には作れる．ただし，この種の原理的な問題のためには，長方形分割図より，さらに抽象的な概念から，説きおこしたほうがよい．

その一方，はじめに予定していた，建築図面の面積の区分集計などについては，一般用記号などは，かなりに迂遠なところがある．実用方面と，基礎方面と，主題・方法をわけて，しばらく別々に，論じ進めてみたい．