数理解析研究所講究録
第98巻1970年12-29

Lゲームの計算機による分類

東大　理　佐藤　雅彦

§1　序

Lゲームは水平思考で有名なEdward de Bono博士によって考案されたゲームです。数学教室計算機室のTOSBAC-3300を使って，Lゲームの“分類”（後述）を行うことができました。またこの分類結果を用いて，人間と対局するprogramを作ったので，これらについて述べてみたいと思います。

Lゲームはデパート等で手に入れることができます。またde Bono氏の“水平思考5日間コース”という本にも解説があります。それによればルールは次のようである。

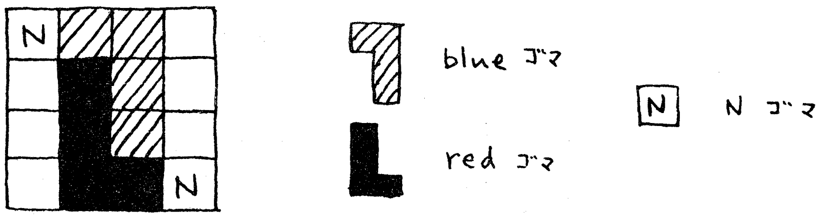
§2　Lゲームのルール

Lゲームのゲーム盤は縦・横4列の碁盤目からなる。

コマ：　playerはそれぞれ1個のL字型コマ（Lゴマ。赤色と青色がある。）をもつ。Lゴマは碁盤目4つ

を占める。このほかに碁盤目1つを占める中立的なNゴマが2つある（黄色）。

出発位置：　図のようなコマの位置から始める。



動かし方：　playerは交互に，自分のLゴマの位置を変えてゆかねばならない。この場合，コマをウラ返したり，回転したりしてもよく，4つの碁盤目通りにおかれればどこにおいてもよい。ただし相手のLゴマや，Nゴマと重なってはならない。Lゴマを動かした後からならplayerは，もしそうしたいなら，2つあるNゴマのどちらか一方を盤上の空いているところに動かしてもよい。

ゲームの勝敗：　ゲームの目的は相手のLゴマを動けないように追いつめることにある。相手のLゴマが位置を変えられなくなればアナタの勝ち。

§3　ゲームの定義

最初に，一般論として，ゲームおよびそれに関するいくつかの言葉の定義を与える。

Def. 集合Sと，写像$h: S \longrightarrow \mathcal{P}(S)$ ($= 2^S =$ (Sの部分集合全体。))，そして$s \in S$が与えられたとき，組(S, h, s)をゲームという。Sの元xを局面という。とくにsを開始局面という。$h(x) = \phi$のときxを最終局面という。また$y \in h(x)$のとき$H = (x, y)$のことを手といい，局面xが手Hによって局面yになったという。

多くの2人で競技するゲームは，適当な解釈により，いま定義した意味でのゲームとみなすことができる。

ゲーム$\Gamma = (S, h, s)$のruleは次のようである。

先手は$s_1 \in h(s)$をえらんで，手(s, s_1)を打ち，局面をs_1に変えて後手にわたす。後手は$s_2 \in h(s_1)$をえらんで局面をs_2にして先手にわたす。このようにして，局面の列$s, s_1, s_2, \cdots$ができるが，ある局面s_nを最終局面s_{n+1}に変えて相手にわたすことのできた方が勝ちである。

集合B_i, F_i ($i = 0, 1, 2, \cdots$)を次のように帰納的に定める。

$$B_0 = \{ x \in S ; h(x) = \phi \}$$

$$F_i = \{ x \in S ; h(x) \cap B_i \neq \phi \}$$

$$B_{i+1} = \{ x \in S ; h(x) \subset F_i \}$$

$B_0 \subset B_1 \subset B_2 \subset \cdots$，$F_0 \subset F_1 \subset F_2 \subset \cdots$，である。

$$B = \bigcup_{i=0}^{\infty} B_i ,$$

$$F = \bigcup_{i=0}^{\infty} F_i ,$$

$$U = S - (B \cup F) ,$$ とおく。

$S = B \cup F \cup U$ (disjoint) である。そこで,

Def. $x \in S$ が後手必勝形 $\Leftrightarrow$ $x \in B$,

$x \in S$ が先手必勝形 $\Leftrightarrow$ $x \in F$,

$x \in S$ が 不定形 $\Leftrightarrow$ $x \in U$。

この意味はあきらかであろう。たとえば,$x \in B$ ならば,(x を開始局面と思って先手・後手をきめるとき),先手がどのような手を打っても,後手がそれに対して適当な手を打っていくことにより後手が必ず勝つことができる。

ゲームが与えられたときに,B, F, U, を具体的に求めることをゲームの分類の問題ということにすれば,目標はLゲームを分類することである。

次に上の定義にあわせてLゲームを定義しよう。

§4 Lゲーム $\Lambda = (S, h, s)$ の定義

次の条件(1),(2),(3),をみたす 4×4 matrix $A = (a_{ij})$ の全体をTとするとき,$S = T \times \{r, b\}$ (直積)である。

(1) $a_{ij} \in \{r, b, n, o\}$ (r, b, n, o は単なる文字で red ゴマ, blue ゴマ,Nゴマ,空白に対応する。)

(2)　A は r を4つ，b を4つ，n を2つ，0 を6つ含む。

(3)　L を $\mathbb{R}^2$ の格子点全体とし，$\mathbb{R}^2$ の部分距離空間と考える。

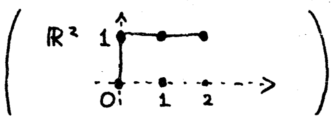
$L \supset D = \{(i,j);\ 1 \leqq i,j \leqq 4\}$ とおく。

A に対して i_A を次のように定める。

$$\begin{array}{cccc} i_A : & D & \longrightarrow & \{r, b, n, 0\} \\ & \rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} \\ & (i,j) & \longmapsto & a_{ij} \end{array}$$

このとき L の等長変換 α, β が存在して，

$$i_A \circ \alpha\,(p) = r \qquad (\text{for } p \in \{(0,0), (0,1), (1,1), (2,2)\})$$
$$i_A \circ \beta\,(p) = b$$



Rem.　L の等長変換全体は群をなすが，その一つの部分群 ad を，後で使用するので，定義しておく。

そのため $L \subset \mathbb{C}$ とみなす。

$$ad = \{\alpha\ ;\ \alpha \text{ は } L \text{ の等長変換 s.t. } \exists q \in L, \forall p \in L,\ \alpha(p) = p + q。\}$$

$S = T \times \{r, b\} = \{(A, a);\ A \in T,\ a \in \{r, b\}\}$ である。

次に　$h : S \longrightarrow \mathcal{P}(S)$ を定める。

$(A, a), (A', a') \in S$ に対して，$(A', a') \in h(A, a)$

$\Longleftrightarrow$ (1)　$a' \neq a$

(2)　$A \wedge A'$ は a を3つ以下，a' を4つ，n を1つ以上，含む。ただし，$A \wedge A' = (a_{ij} \wedge a'_{ij})$

$$a_{ij} \wedge a'_{ij} = \begin{cases} a_{ij} & (a_{ij} = a'_{ij}) \\ 0 & (a_{ij} \neq a'_{ij}) \end{cases}$$

開始局面 s は，

$$s = \left(\begin{pmatrix} n & b & b & 0 \\ 0 & r & b & 0 \\ 0 & r & b & 0 \\ 0 & r & r & n \end{pmatrix}, \quad ȴ \right) \quad である。$$

§5　（準）同型

Def.　一般に 2 つのゲーム (S, h, s) と (S', h', s') に対して，

$f : S \to S'$ が準同型写像

$\Leftrightarrow$ (1)　$f(s) = s'$

(2)　$f \circ h = h' \circ f$

とくに f が単射のとき同型写像という。

$f : S \to S'$ が上への準同型であるとき，§3 のように，$S = B \cup F \cup U$，$S' = B' \cup F' \cup U'$ と分解するとき，$f(B) = B'$，$f(F) = F'$，$f(U) = U'$ となる。

同型写像の例を L ゲーム Λ について考えてみよう。$D \subset L$ を L の部分距離空間とみるとき，D の等長変換全体は dihedral group D_4 をなす。D_4 は次のようにして S に act する。

$\alpha \in D_4$，$A \in T$ に対して，次の図式が可換になるような $A' \in T$

$$\begin{tikzcd} D \arrow[d, "i_A"'] & D \arrow[l, "\alpha"'] \arrow[dl, "i_{A'}"] \\ \{r, b, n, o\} & \end{tikzcd}$$

がただ一つ存在する。

$(A, a) \in S$ に対して,

$\alpha(A, a) = (A', a)$ と定めるとよい。

$\alpha : S \to S$ が上への同型写像になるのはあきらかである。つまり, $(S, k, \alpha(s))$ は (S, k, s) と同型である。

Lゲーム Λ についてもう一つの trivial な同型は次のようなものである。

$$\begin{array}{rccc} c : & S & \longrightarrow & S \\ & \Psi & & \Psi \\ & (A, a) & \longmapsto & (A', a') \end{array}$$

i.e. color change

ただし $A = (a_{ij})$, $A' = (a'_{ij})$ において, $a'_{ij} = \begin{cases} r & (a_{ij} = b), \\ b & (a_{ij} = r), \\ a_{ij} & (a_{ij} = n, o) \end{cases}$ また, $a' = \begin{cases} r & (a = b), \\ b & (a = r) \end{cases}$。

§6 商ゲーム

Def. 一般にゲーム $\Gamma = (S, k, s)$ とその自己同型($\alpha : S \to S$ が上への同型写像となるとき α を Γ の自己同型と言うことにする。)よりなる群 G (Γ の自己同型全体と一致しなくてもよい。)があるとき, $\Gamma' = (S', k', s')$ を次のように定め Γ を G で割ったゲームという。そして $\Gamma' = \Gamma / G$ とかく。

(1) G は S に act するとみて, $S' = S/G$。

(2) $s' = [s]$ (s の G による同値類)

(3) $x'=[x]\in S'$に対して，$h'(x')=[h(x)]$と定める。

このように定義したとき$[\ ]: S\longrightarrow S'=S/G$は上への準同型写像になる。

LゲームΛについて，$G=$(D_4とcで生成される群)として，$\Lambda'=\Lambda/G$とおく。Λ'は次のゲーム$\mathcal{M}=(M,\mu,m)$と同型である。

(1) $M=\{A\in T$；§4のTの定義の中のαとして，$\alpha\in ad$がとれる。$\}$

$$\left[\text{つまり，} A\in M \text{ は } \quad A=\begin{pmatrix} \boxed{r\ r\ r} \\ \boxed{r} \end{pmatrix} \text{ の形をしている。}\right]$$

(2) μは次のようにきめる。

A，$A'\in M$に対して，

$$A'\in\mu(A) \iff [(A',r)]\in h'[(A,r)]$$

(3) 開始局面

$$m=\begin{pmatrix} o & o & o & n \\ r & r & r & b \\ r & b & b & b \\ n & o & o & o \end{pmatrix}$$

Λを直接調べるかわりに，計算機では$\mathcal{M}$について調べることにした。$\mathcal{M}$のこともLゲームということにする。

§7 計算機内部での局面の表現

Lゲームの分類の問題を解き，また人間と対局するprogramを作るためには，計算機内部で局面をいくつかの方法で表現することが必要である。実際には大体次の3通りの表現，およびそれらの間の対応づけをするホンヤク routine を用意して計算を行った。

(1) Λ型表現　Λの局面と対応する。人間との対局に使われる。

(2) M型表現　Mの局面と対応する。局面xに対して打つことのできる手を計算するときにはこの表現を用いる。

(3) ℓ型表現　Mの個数kがわかれば，一対一対応により，$M=[0,k-1]$　（$[\alpha,\beta]$は$\{x\in\mathbb{Z};\alpha\leqq x\leqq\beta\}$を表はす。）とみなしてよい。このような自然数としての表現をℓ型表現という。局面についての情報をcoreに貯えるときは，局面をℓ型表現で表はす。

ℓ型表現を実現するためにまずMの個数を計算した。それを以下に述べる。

index の対応$(i,j)\longleftrightarrow k=4(i-1)+(j-1)$によって局面$A=(a_{ij})\in M$を横 vector (a_k)とみることができる。

$0\leftrightarrow 0$，$n\leftrightarrow 1$，$b\leftrightarrow 2$，$r\leftrightarrow 3$と対応づければ，形式的に，$A=(a_k)=3(r_k)+2(b_k)+1(n_k)$　（r_k, b_k, n_kは0 or 1）とかける。たとえばMの開始局面は

$$m = (0,0,0,1,3,3,3,2,3,2,2,2,1,0,0,0)$$
$$= 3(0,0,0,0,1,1,1,0,1,0,0,0,0,0,0,0)$$
$$+2(0,0,0,0,0,0,0,1,0,1,1,1,0,0,0,0)$$
$$+1(0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0)\text{。}$$

さらに vector (v_k) $(v_k = 0 \text{ or } 1)$ を2進数とみて，$V = \sum_{k=0}^{15} 2^{15-k} v_k$ と同一視すれば，

$$m = 3(07200) + 2(0560) + 1(010010)$$
$$= 07200r + 0560b + 010010n$$ とかける。

（0ではじまる数は octal number である。）

このようにして，一般に，

$$M \ni A = R(A)r + B(A)b + N(A)n$$
$$= Rr + Bb + Nn \qquad (0 < R, B, N < 2^{16})$$

と書ける。これらをそれぞれ，AのR成分，B成分，N成分という。T-3300 の core は 1 word 24 bits であるから，R, B, N を 3 words に格納することができる。この 3 words は局面 A の M 型表現のひとつである。

R については

$$\{R\ ;\ R = R(A),\ A \in M\}$$

$$= \left\{ \begin{pmatrix} 1&1&1&0\\1&0&0&0\\0&0&0&0\\0&0&0&0 \end{pmatrix}, \begin{pmatrix} 0&1&1&1\\0&1&0&0\\0&0&0&0\\0&0&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&0&0\\1&1&1&0\\1&0&0&0\\0&0&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&0&0\\0&1&1&1\\0&1&0&0\\0&0&0&0 \end{pmatrix}, \begin{pmatrix} 0&0&0&0\\0&0&0&0\\1&1&1&0\\1&0&0&0 \end{pmatrix}, \right.$$

$$\left.\begin{pmatrix} 0&0&0&0\\ 0&0&0&0\\ 0&1&1&1\\ 0&1&0&0 \end{pmatrix}\right\} = \{0164000, 072000, 07200, 03500, 0350, 0164\}$$

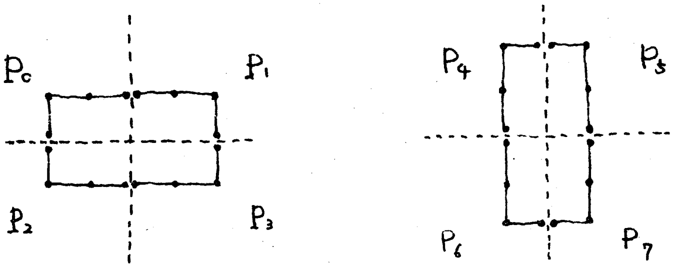
これは name +fred+ ではじまる6ケの coreに格納されている。（program は assembler 言語 SMAPⅢ で書かれており，その文法によれば，+<string>+ により address に name をつけることができる。）

$\mathcal{B} = \{B\ ;\ B = B(A),\ A \in M\}$ については，次のようにして，0〜95までの数が対応させられる。$B = (b_k)\ (k=0,\cdots,15)$ について $k(B) = \min\{k\ ;\ b_k = 1\}$ とおく。$k(B)$ は 0 から 11 までの値をとりうる。次に B に対して $O(B)$ を定める。そのために，

$$P_0 = \{(-2,0),\ (-2,1),\ (-1,1),\ (0,1)\}$$
$$P_1 = \{(2,0),\ (2,1),\ (1,1),\ (0,1)\}$$
$$P_2 = \{(-2,0),\ (-2,-1),\ (-1,-1),\ (0,-1)\}$$
$$P_3 = \{(2,0),\ (2,-1),\ (1,-1),\ (0,-1)\}$$
$$P_4 = \{(0,2),\ (-1,2),\ (-1,1),\ (-1,0)\}$$
$$P_5 = \{(0,2),\ (1,2),\ (1,1),\ (1,0)\}$$
$$P_6 = \{(0,-2),\ (-1,-2),\ (-1,-1),\ (-1,0)\}$$
$$P_7 = \{(0,-2),\ (1,-2),\ (1,-1),\ (1,0)\}$$

とおく。

$P_i \subset L\ (i = 0,1,\cdots,7)$ である。



§4 の $\mathfrak{A}_\alpha$ の定義を思いだせば次が言える。

$\forall B \in \mathcal{B}$, $\exists 1\, P_i$, $\exists 1\, \alpha \in \mathfrak{A}_\alpha$ s.t. $i_B \circ \alpha(p) = 1$ (for all $p \in P_i$)

(ただし $i_B : D \to \{0, 1\}$, $i_B(i, j) = b_{ij} = b_k$ $(B = (b_{ij}) = (b_k))$)

このようにしてBに対して定まる P_i の index $i = O(B)$ と定める。つまり $O(B)$ は b ゴマがどのような向きにおかれているかを示している。$p : \mathcal{B} \to [0, 95]$ を $p(B) = 8 \times k(B) + O(B)$ と定める。p はあきらかに injection である。(surjection ではない。) p^{-1} の値(ただし $p^{-1}(i)$ が定義されないときは仮に0として)が +form+ という name で始まる96個の core に格納されている。

以上の表現によってMの個数を数えることができる。その方法はプラパズル(pentomino 等の箱詰めパズル)と同じく、4×4のます目にrゴマ, bゴマ, nゴマを重ならないように置くことを考えればよい。最初にrゴマをおく。rゴマは +fred+ から格納してある順にとり出す。したがって最初は、R = fred(0) = 0164000 である。(たとえば a 番地に +name+ という名前がついておれば、name(i) = a+i 番地の core (の値)をいみすることとする。) 次に +form+ から順次 bゴマを

とり出して置く。(ただしとりだしたB=0 ならskipする。)重なるかどうかはbitwise andをとってみればよい。このようにしてform(91)までいけば(form(92)以下は0), rゴマの置き方をかえてfred(1)とする。以下同様。このようにして, rゴマとbゴマを重ならないようにおく, 置き方は82通りあることがわかった。このさい, 新しい置き方が得られるごとにそのときのBを+mt+というnameで始まるblockに順次格納した。しかし+mt+をみただけではBの元の列がえられるだけで, いっしょに置かれているRがわからない。そのために, +ti+というnameで始まるblockを次のように定めた。+ti+ fix. 0, 24, 42, 52, 58, 71, 82 (これはSMAPのmacro命令で, これが実行されると, $ti(0)=0, ti(1)=24, \cdots,$ $ti(6)=82$ となる。)これは次のことを示す。$B=mt(i)$ $(i\in[0,23])$ といっしょに置かれている$R=fred(0)$。$B=mt(i)$ $(i\in[24,41])$ といっしょに置かれている$R=fred(1)$。….

rゴマとbゴマの置き方は82通りあることがわかった, そのおのおのに対してnゴマ2ケのおき方は${}_8C_2=28$通り。よってMの個数は$82\times 28=2296$である。

最後にλ型表現を与えよう。i.e. Mと$[0,2295]$を同一視するための特定の全単射同型を与える。そのためにMに次のような全順序を入れる。そうすればMと$[0,2295]$の順序同型が

一意にきまりそれは全単射同型をあたえる。

$A \in M$に対して$m : M \to [0, 81]$を次のように定める。

$R = R(A)$, $B = B(A)$とする。$R = \mathrm{fred}(\exists 1 i)$ $(0 \leqq i \leqq 5)$とかける。また$\exists 1 j$ s.t. $B = mt(j)$, $t_i(i) \leqq j < t_i(i+1)$。このjをとり$m(A) = j$と定める。$A$, $A' \in M$ $(A \neq A')$をとるとき，$m(A) \lessgtr m(A') \Rightarrow A \lessgtr A'$と定める。$m(A) = m(A')$のときは，$N(A) \lessgtr N(A') \Rightarrow A \lessgtr A'$。こうしてえられた同型を$\ell : M \to [0, 2295]$とかく。

§8　LゲームMの分類

一般のゲーム$\Gamma = (S, h, s)$を考え，その分類を(B, F, U)とする。このとき集合C, G, Vが$C \subset B$, $G \subset F$, $V = S - (C \cup G)$をみたすとき，(C, G, V)をΓの不完全分類という。この不完全分類の特性関数$\chi : S \to \{0, 1, 2\}$を$\chi(x) = 0$ $(x \in V)$, 1 $(x \in G)$, 2 $(x \in C)$と定める。不完全分類(C, G, V)があるとき，その拡大$(\tilde{C}, \tilde{G}, \tilde{V})$を次のように定める。

$$x \in \tilde{C} \iff h(x) \subset G$$

$$x \in \tilde{G} \iff h(x) \cap C \neq \phi$$

$$\tilde{V} = S - (\tilde{C} \cup \tilde{G})$$

$(\tilde{C}, \tilde{G}, \tilde{V})$も不完全分類であり，$C \subset \tilde{C}$, $G \subset \tilde{G}$, $V \supset \tilde{V}$である。Sが有限集合のときは，任意の不完全分類をとってそれ

の拡大を重ねれば，有限回でSの分類に達する。最初の不完全分類を(ϕ, ϕ, S)とすれば，§3でのべたB_i, F_iが拡大の過程でえられる。このalgorithmをそのまま使うprogramも作れるが，次に述べるように，計算時間を短かくする方法がある。

ゲームPについて，$\exists f, g : S \to \mathcal{P}(S)$ s.t. $\forall A \in S$, $h(A) = f(g(A))$と仮定する。Pの不完全分類(C, G, V)とその特性関数χが与えられたとして，$\varphi, \tilde{\chi} : S \to \{0, 1, 2\}$を次のように定める。

$$\varphi(x) = \begin{cases} 1 & (\text{if } \exists y \in f(x),\ \chi(y) = 2) \\ 2 & (\text{if } \forall y \in f(x),\ \chi(y) = 1) \\ 0 & (\text{その他の場合}) \end{cases}$$

$$\tilde{\chi}(x) = \begin{cases} 1 & (\text{if } \exists y \in g(x),\ \varphi(y) = 1) \\ 2 & (\text{if } \forall y \in g(x),\ \varphi(y) = 2) \\ 0 & (\text{その他の場合}) \end{cases}$$

このとき$\tilde{\chi}$は，拡大$(\tilde{C}, \tilde{G}, \tilde{V})$の特性関数になる。

Lゲーム$M = (M, \mu, m)$について，f, gを次のように定めれば，それは上の条件をみたす。$A \in M$に対して，$f(A) =$ {局面Aからnゴマを高々1コ動かして得られる局面}，$g(A) =$ {局面AからLゴマを動かして得られる局面}。$h(A) = f(g(A))$となるのはゲームの定義から明らかであろう。

この方法で計算時間は$\frac{1}{3}$ほどになったはずである。

§9　分類の結果

上のような方法で，一回の拡大に約20〜30分かかって分類を完了した。B_i, F_iは§3のとおりとして，$\overline{B_i} = B_i - B_{i-1}$ ($B_{-1} = \phi$)，$\overline{F_i} = F_i - F_{i-1}$ ($F_{-1} = \phi$) と定める。Mについての$\overline{B_i}$, $\overline{F_i}$の個数は次の通りである。($\overline{B_i} = \overline{F_i} = \phi$ ($i \geq 5$))

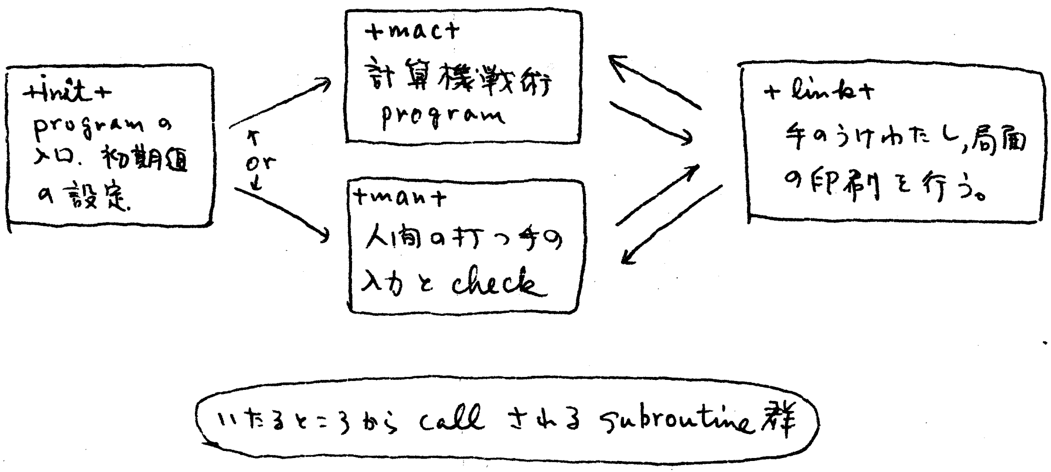
$\overline{B_0}$	15	$\overline{F_0}$	768
$\overline{B_1}$	3	$\overline{F_1}$	27
$\overline{B_2}$	3	$\overline{F_2}$	81
$\overline{B_3}$	5	$\overline{F_3}$	11
$\overline{B_4}$	3	$\overline{F_4}$	119

B　29　　F　1006　　U　1261

このようにUが非常に大きいので，実際にゲームをやってみるとloopになることが多い。開始局面はUに属するので2人とも間違わないかぎり，必ずloopになる。人間どうしの対局では，一般に，対局中の局面を全部憶えているわけではないので，loopに気づかず，結局どちらかが手を誤って勝負がついてしまうようである。しかし計算機との対局では，局面を毎回出力することにしているのでloopにすぐ気がつくわけである。

§10 Lゲーム対局program

LGMは人間/計算機対人間/計算機でLゲームを対局することのできるprogramである。その構造は下記のようになっている。programの大半はsubroutine群からなり、その大部分は、Lゲーム分類programで使用したものをそのまま使っている。これらのsubroutineの仕事は大体次の3つのどれかである。(1) 局面の各種表現の間の翻訳。(2) 手のsearch。i.e. 局面Aに対する$h(A)$の計算。(3) 局面の評価。i.e. 局面Aの特性関数$\chi(A)$の計算。 このようなsubroutineがあるので、計算機の戦術programはLゲームMをやっているつもりでおり、人間の方はLゲームAをやっているつもりですればよい。人間も計算機も打った手は+link+に報告し、+link+はその手を次の手番の戦術programに教えてやる。



§ 11　付録

```
+form+        oct.164000,161000,107000,0,144200,142100,104300,0
              oct.72000,70400,43400,0,62100,61040,42140,42300
              oct.0,0,0,27000,31040,30420,21060,21140
              oct.0,0,0,13400,0,0,0,10460
              oct.7200,7040,4340,0,6210,6104,4214,0
              oct.3500,3420,2160,0,3104,3042,2106,2114
              oct.0,0,0,1340,1442,1421,1043,1046
              oct.0,0,0,560,0,0,0,423
              oct.350,342,216,0,0,0,0,0
              oct.164,161,107,0,0,0,0,0
              oct.0,0,0,56,0,0,0,0
              oct.0,0,0,27,0,0,0,0

+fred+        oct.164000,72000,7200,3500,350,164

+f0+          oct.110,111,144,145,240,274,302,421
 b            oct.422,425,454,766,1443,3402,3530
```

+form+，+fred+ については前に述べた。

ここで使われている，oct. は fix. 同様 SMAP の macro 命令であって，以下に comma で区切って書かれている数字を 8 進数として読んで，location counter の示す番地以降に順次格納する。

+bo+ は B_0 の 15 ケの元を，ℓ 型表現で書いたものである。