

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, この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。

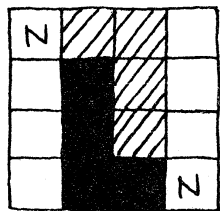
§2 L ゲームのルール

L ゲームのゲーム盤は縦・横4列の基盤目からなる。

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

を占める。このほかに基盤目 1 つを占める中立的な
N コマが 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。



blue コマ

red コマ



N コマ

動かし方: player は交互に, 自分の L コマの位置と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 つの基盤目通りにおかねばどこにいてもよい。ただし相手の L コマや, N コマと重なってはならない。L コ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 コ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 コマを動けないうように追いつめることにある。相手の L コマが位置を変えられなくなれば相手の勝ち。

§3 ゲームの定義

最初に、一般論として、ゲームの定義をその中に關するいくつかの言葉の定義と与える。

Def. 集合 S と、写像 $h: S \rightarrow \mathcal{P}(S)$ ($= 2^S = (S \text{ の部分集合全体。})$), 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 $H = (x, y)$ のことを手という、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 $s, s_1, s_2, \dots$ ができるが、ある局面 s_n と最終局面 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 ($i = 0, 1, 2, \dots$) を次のように帰納的に定める。

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

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

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

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

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

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

$$U = S - (B \cup F), \quad \text{と置く。}$$

$S = B \cup F \cup U$ (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 を開始局面と見て, 後手・後手を選びるとき), 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。

以下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
 T とすると, $S = T \times \{r, b\}$ (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 手, blue 手, N 手, 空白に対応する。)

(2) A は $r \in 4$ つ, $b \in 4$ つ, $n \in 2$ つ, $o \in 6$ つ 含む。

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

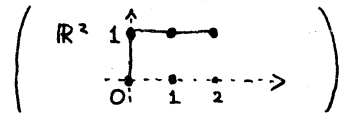
$$L \supset D = \{(i, j); 1 \leq i, j \leq 4\} \text{ とおく。}$$

A に対して i_A を次のように定める。

$$\begin{array}{ccc} i_A: D & \longrightarrow & \{r, b, n, o\} \\ \downarrow & & \downarrow \\ (i, j) & \longmapsto & a_{ij} \end{array}$$

α, β は L の等長変換 α, β が存在して,

$$\begin{aligned} i_A \circ \alpha(p) &= r \\ i_A \circ \beta(p) &= b \end{aligned} \quad (\text{for } p \in \{(0,0), (0,1), (1,1), (2,2)\})$$



Rem. L の等長変換全体は群となるが, その一つの部分群 $\mathcal{A}_d$ を, 後で使用するのて, 定義しておく。

$\exists g \in L$ である $L \subset \mathbb{C}$ とみえる。

$$\mathcal{A}_d = \{ \alpha; \alpha \text{ は } L \text{ の等長変換 s.t. } \exists g \in L, \forall p \in L, \alpha(p) = p + g. \}$$

$$S = T \times \{r, b\} = \{(A, a); A \in T, a \in \{r, b\}\} \text{ とおく。}$$

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

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

$$\Leftrightarrow (1) \quad a' \neq a$$

(2) $A \wedge A'$ は $a \neq 3$ 以下, $a' \in 4$ つ, $n \in 1$ 以上, 含む。さらに, $A \wedge A' = (a_{ij} \wedge a'_{ij})$

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

開始局面 S は,

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

§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,

$f: S \rightarrow S'$ が準同型写像

$$\Leftrightarrow (1) \quad f(s) = s'$$

$$(2) \quad f \circ h = h' \circ f$$

とくに f が単射のとき同型写像という。

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

同型写像の例としてゲーム A について考えてみよう。 $D \subset L$
 は 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 となる。 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 $A' \in T$

$$\begin{array}{ccc}
 D & \xleftarrow{\alpha} & D \\
 i_A \downarrow & & \swarrow i_{A'} \\
 \{r, b, n, o\} & &
 \end{array}$$

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。よい。

$\alpha: S \rightarrow S$ が上の同型写像になるのはあきらかである。

つまり, $(S, h, \alpha(S))$ は (S, h, S) と同型である。

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。

$$\begin{array}{lcl}
 c: S & \longrightarrow & S \\
 \downarrow & & \downarrow \\
 (A, a) & \longmapsto & (A', a') \\
 \text{i.e. color change} & &
 \end{array}
 \left[\begin{array}{l}
 \text{たとえば } A = (a_{ij}), A' = (a'_{ij}) \text{ ならば} \\
 a'_{ij} = r(a_{ij} = b), b(a_{ij} = r), \\
 a_{ij}(a_{ij} = n, o) \text{ また, } a' = r(a = b), \\
 b(a = r).
 \end{array} \right.$$

§ 6 商ゲーム

Def. 一般にゲーム $P = (S, h, s)$ とその自己同型 ($\alpha: S \rightarrow S$ が上の同型写像となるとき $\alpha \in P$ の自己同型と言ふことにする。) による群 G (P の自己同型全体と一致していてもよい。) があるとき, $P' = (S', h', s')$ を次のように定め $P \in G$ で割ったゲームという。そして $P' = P/G$ とかく。

(1) G は S に作用するとみて, $S' = S/G$ 。

(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 $[]: S \rightarrow S' = S/G$ は S への準同型写像になる。

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

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

$\left[\text{つまり, } A \in M \text{ は } A = \left(\begin{array}{|c|ccc|} \hline r & r & r & r \\ \hline r & & & \\ \hline \end{array} \right) \text{ の形をしている。} \right]$

(2) μ は次のように定める。

$A, A' \in M$ 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 M について調べることにした。 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

し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、あ るいはその9向の対応づけとせるホニヤ routine を用意して 計算を行った。

(1) A型表現 Aの局面と対応する。人間との対局に使わ る。

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

(3) l型表現 Mの個教長がわかれば、一対一対応により、 $M = [0, k-1]$ ($[a, b]$ は $\{x \in \mathbb{Z}; a \leq x \leq b\}$ 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の個教を計算した。それを 以下に述べる。

index の対応 $(i, j) \leftrightarrow k = 4(i-1) + (j-1)$ により、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

$$\begin{aligned}
m &= (0, 0, 0, 1, 3, 3, 3, 2, 3, 2, 2, 2, 1, 0, 0, 0) \\
&= 3(0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0) \\
&\quad + 2(0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0) \\
&\quad + 1(0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0)
\end{aligned}$$

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,

$$\begin{aligned}
m &= 3(07200) + 2(0560) + 1(010010) \\
&= 07200r + 0560b + 010010h \quad \text{とかけろ。}
\end{aligned}$$

(0 は はじまる数は octal number である。)

このようにして, 一般に,

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

と書ける。これはよく知られた, 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 は 3 words に格納することはできる。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 + 2 はじまる 6 けたの core に格納される。
 (program は assembler 言語 SHAP II で書かれており、この文法に於ては、+ <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、
 $0 \sim 95$ までの数が対応させられる。 $B = (b_k) (k=0, \dots, 15)$ について
 $l(B) = \min\{k; b_k = 1\}$ とおく。 $l(B)$ は 0 から 11 までの値をとる。次に B に対して $O(B)$ を定める。そのために、

$$P_0 = \{(-2, 0), (-2, 1), (-1, 1), (0, 1)\}$$

$$P_1 = \{(2, 0), (2, 1), (1, 1), (0, 1)\}$$

$$P_2 = \{(-2, 0), (-2, -1), (-1, -1), (0, -1)\}$$

$$P_3 = \{(2, 0), (2, -1), (1, -1), (0, -1)\}$$

$$P_4 = \{(0, 2), (-1, 2), (-1, 1), (-1, 0)\}$$

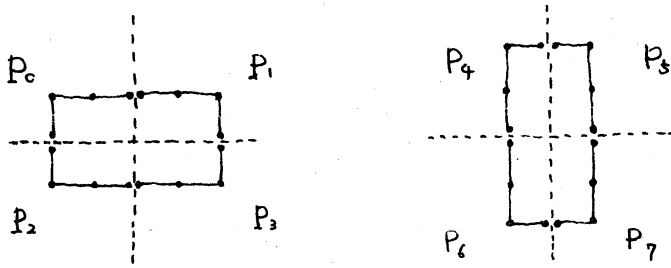
$$P_5 = \{(0, 2), (1, 2), (1, 1), (1, 0)\}$$

$$P_6 = \{(0, -2), (-1, -2), (-1, -1), (-1, 0)\}$$

$$P_7 = \{(0, -2), (1, -2), (1, -1), (1, 0)\}$$

とおく。

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



§ 4 の ad の定義を思いおこせば次が言える。

$\forall B \in \mathcal{B}, \exists 1 P_i, \exists 1 \alpha \in ad \text{ s.t. } i_B \circ \alpha(p) = 1 \text{ (for all } p \in P_i)$
 (ただし $i_B: D \rightarrow \{0, 1\}$, $i_B(i, j) = b_{ij} = b_k$ ($B = (b_{ij}) = (b_k)$))
 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 コマがどのような向きに並んでいる
 かを示している。 $p: \mathcal{B} \rightarrow [0, 95]$ と $p(B) = 8 \times k(B) + O(B)$
 と定める。 p は明らかに 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
 とする) 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
 方法はペンタミノ (pentomino 等の箱詰めパズル) と同じく、
 4×4 のマス目に r コマ, b コマ, n コマを重ねられないように
 置くことを考えなければならない。最初に r コマを置く。 r コマは +
 fred+ から格納してある順にとり出す。したがって最初、
 $R = \text{fred}(0) = 0164000$ である。(たとえば a 番地に +name+
 という名前がついておれば、 $\text{name}(i) = a + i$ 番地の core (の値)
) をみつけることができる。) 次に +form+ から順次 b コマを

とり出して置く。(ただしとり出した $B=0$ なら skip する。) 重なるかどうかは bitwise and をとってみる。このようにして form (91) をいけば (form (92) 以下は 0), γ と ω の置き方をかえて fred (1) とする。以下同様。このようにして, γ と ω と ω と ω と重なる方ないようにみる, 置き方は 82 通りあることがわかった。このとき, 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 $+mt+$ とみただけでは元の列がえられるだけで, いっしょに置かなくては R がわからぬ。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, ti が実行されると, $ti(0)=0, ti(1)=24, \dots, ti(6)=82$ となる。) これは次のことを示す。 $B=mt(i) (i \in [0, 23])$ といっしょに置かなくては $R=fred(0)$ 。 $B=mt(i) (i \in [24, 41])$ といっしょに置かなくては $R=fred(1)$ 。……

γ と ω と ω と ω の置き方は 82 通りあることがわかった, そのおのののに対して γ と ω と ω と ω の置き方は $82=28$ 通り。よって M の個数は $82 \times 28 = 2296$ である。

最後に M の表現を与えよう。い.e. M と $[0, 2295]$ と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

一意にまわりまわす全単射同型 f である。

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

$R = R(A)$, $B = B(A)$ とする。 $R = \text{fred}(\{t_i\})$ ($0 \leq i \leq 5$) とか

ける。また $\exists j$ s.t. $B = \text{mt}(j)$, $t_i(i) \leq j < t_i(i+1)$ 。 $i = 9$ j

より $m(A) = j$ と定める。 $A, A' \in M$ ($A \neq A'$) とするとき,

$m(A) \leq m(A') \Rightarrow A \leq A'$ と定める。 $m(A) = m(A')$ のときは,

$N(A) \leq N(A') \Rightarrow A \leq A'$ 。 $\therefore$ f を f の同型 $g: M \rightarrow [0, 2295]$

とかく。

§ 8 L が $-4M$ の分類

一般の L が $-4P = (S, h, s)$ と考え, その分類 (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 P の不完全分類という。この不

完全分類の特性関数 $\chi: S \rightarrow \{0, 1, 2\}$ 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, χ

の拡大 $(\tilde{C}, \tilde{G}, \tilde{V})$ を次のように定める。

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

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

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

$(\tilde{C}, \tilde{G}, \tilde{V})$ 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 $\varepsilon(\phi, \phi, S)$ とすれば、与えられた B, F が拡大の過程でえられる。この algorithm とそのおまけの program も作るが、次に述べるように、計算時間を短かくする方法がある。

Let μP について、 $\exists f, g: S \rightarrow \mathcal{P}(S)$ 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、 $\phi, \tilde{\chi}: S \rightarrow \{0, 1, 2\}$ を以下のように定める。

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

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

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

Let $\mu M = (M, \mu, m)$ について、 f, g を以下のように定めれば、それは上の条件を満たす。 $A \in M$ に対して、 $f(A) = \{\text{局面 } A \text{ から } n \text{ ステップ高々 } 1 \text{ コ動かして得られる局面}\}$ 、 $g(A) = \{\text{局面 } A \text{ から } m \text{ ステップ動かして得られる局面}\}$ 。 $h(A) = f(g(A))$ とするのには μ の定義から明きらかである。 μ 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、て分類を完了した。 B_i, F_i は 3 のとありとして、 $\overline{B}_i = B_i - B_{i-1}$ ($B_1 = \phi$), $\overline{F}_i = F_i - F_{i-1}$ ($F_1 = \phi$) と定める。これについての $\overline{B}_i, \overline{F}_i$ の個数は次の通りである。($\overline{B}_i = \overline{F}_i = \phi$ ($i \geq 5$))

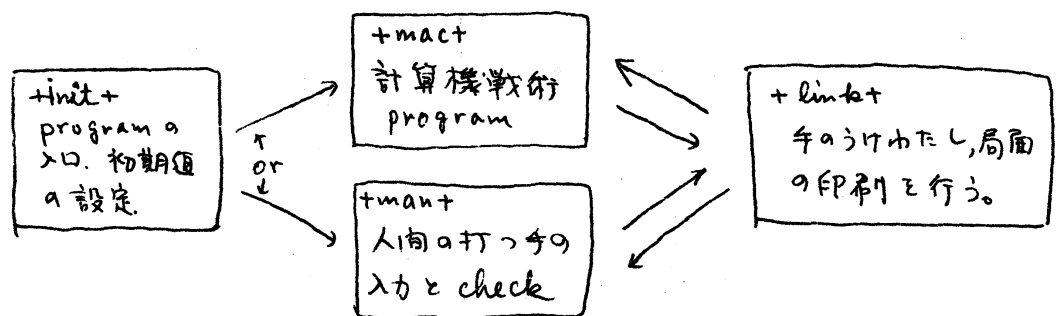
$\overline{B}_0$	1 5	$\overline{F}_0$	7 6 8
$\overline{B}_1$	3	$\overline{F}_1$	2 7
$\overline{B}_2$	3	$\overline{F}_2$	8 1
$\overline{B}_3$	5	$\overline{F}_3$	1 1
$\overline{B}_4$	3	$\overline{F}_4$	1 1 9

B 2 9 F 1 0 0 6 U 1 2 6 1

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 になることが多し。開始局面は U に属するので 2 人とも間違えないうかり、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おくわけではないので、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この 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ゲームを対局するに必要の 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 群が占め、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。これは、subroutine の仕事は大体次の3つのことである。(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 の特性函数 $\chi(A)$ の計算。このように subroutine があるので、計算機の戦術 program は Lゲーム M をやっているとおりであり、人間の方は Lゲーム N をやっているとおりでなければよい。人間も計算機も打った手は +link+ に報告し、+link+ はその手と次の手番の戦術 program に教えてやる。



いたるこは call する subroutine 群

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

+form+ , +fred+ については前に述べた。

こゝで使われている, 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。

+b0+ は B₀ の 15 ケの元を, L 型表現で書いたものである。