

# ボードゲーム BAO における周期的動作の解析

稲畑 康博<sup>†</sup>, 高橋 和子<sup>†</sup>

本論文の目的は、ボードゲーム BAO の動作を解析しその特徴を明らかにすることである。BAO は 2 列に並んだ穴の中の 1 つから石を取り出し、一定規則のもとに順に石を動かしていくゲームである。この動作は、穴をプロセス、石をプロセス間で授受されるメッセージととらえると自然なモデル化が可能である。また、BAO は他の多くのボードゲームとは異なり、1 手の間に盤面の状態が複数回変化しそれが停止しない場合があるという特徴を持つ。本論文ではまず、プロセス代数 CCS を使って BAO の動きを記述し、ある開始状態と着手が与えられたとき、その手が停止するかどうか判定可能であることを示す。次に、停止しない全局面の特徴の一般化から、ルールとは逆向きの動作を使った解析からの 2 通りの方法で 1 手の停止条件を求め、それらの間の関係を示す。

## An Analysis of Cyclic Behaviors of Board Game BAO

YASUHIRO INAHATA<sup>†</sup>, and KAZUKO TAKAHASHI<sup>†</sup>

This paper aims at finding the characteristics of board game BAO by its behavioral analysis. In BAO, two players are taking seeds in turn by moving them from hole to hole, and the movement may not terminate. We describe the behavior of BAO using a process algebra CCS by regarding a hole as a process, and a seed as a message passed between processes. We investigate the condition for the termination by two approaches: extracting the condition by generalizing all the patterns that do not terminate, and analysing the behavior by rolling it back. We also discuss the relation between these conditions.

### 1. はじめに

ボードゲームの研究は人工知能の初期の段階からさかんに行われており、これまでに探索手法や評価方法をはじめとする様々な新しい発見があった。探索や評価を考えるには、ゲームをモデル化しその動きを解析する必要がある。本論文では、Mancala ゲームの 1 つである BAO をプロセスを用いてモデル化し、特に循環性に焦点をあてて動作の特性を明らかにする。

Mancala ゲームはアフリカを中心に古くから行われており、地方により呼称も異なりヴァリエーションも豊富である。その動作規則は複数の穴にはいった石を動かしながら取り合うというもので、石の種類は単一である。一般のボードゲームでは、1 手の間における盤面の状態変化は 1 回だけだが、BAO では複数回あり 1 手が停止しない可能性がある。これは、1 着手が穴の間でバケツリレー的に状態変化を引き起こし、その影響が再び返ってくることにより 1 手の間に同一

位置の状態が何度も変化するためである。通常ボードゲームでは盤面全体を状態、着手をオペレーションとする状態遷移システムとしてモデル化することが多いが、BAO はこのような特徴を持つため穴どうしの相互作用を組み込みながら個々の穴の状態変化を表現しないと動きの特徴は見えてこない。そこで、穴をプロセス、石をプロセス間で授受されるメッセージとしてプロセス通信に基づくモデル化を行う。また、穴どうしをつないだものをボードという 1 つのシステム、それに対して手番の交代や着手の選択などプレイヤーの動作をつかさどるものをマネージャというもう 1 つのシステムと見なすと、ゲームの終了条件や 1 手の停止条件はボードとマネージャ間のメッセージの満たす条件として統一的に表現できる。

本論文ではまず、プロセス代数 CCS を使って BAO の動きを記述し、ある盤面と着手が与えられたとき、その手が停止するかどうか判定可能であることを示す。CCS を使うと局所的な動きや穴どうしの相互作用が自然に表現できるが、逆に停止するための盤面と着手の満たす条件を求めるのは難しい。そこで、以下の 2 種類の方法で停止条件を求めた。

1 つは、初期状態から到達可能かつそこから開始す

<sup>†</sup> 関西学院大学

Kwansei Gakuin University

現在、株式会社 DNP 情報システム

Presently with DNP Information Systems

る1手が停止しないような局面をすべて求め、得られた結果を一般化することによってその特徴を抽出する方法である。局面の探索にあたっては、計算量的問題を避けるため、単純な全探索をせずモデルチェッカ SPIN を利用する。SPIN では「必ず停止する」という仕様の検証を行うと停止しないモデルが反例として出力される。その結果、停止しない場合に見られる周期的な動きは2通りに分類でき、各々は特定の条件を満たしていることが分かった。

もう1つの方法は、巻き戻しによる解析である。任意の盤面から任意の穴を選択してゲームのルールとは逆向きの動作を行い、それが停止するための条件を求める。この条件は順方向の動作が停止するための条件にもなる。

さらに、これら2つの方法によって求めた条件が矛盾しないことを示す。

本論文の構成は以下のとおりである。2章で BAO のルールを説明する。3章ではプロセス代数 CCS による BAO の記述について述べる。4章では1手の停止条件の求め方について議論する。5章では周期性を持つ局面の一般化による解析について述べる。6章では逆向きの動作を使った解析について述べ、一般化による解析結果との関係を示す。7章で議論を行い、8章で結論と今後の課題について述べる。

## 2. ボードゲーム BAO

### 2.1 ルール

BAO は2人のプレイヤーによって行われるゲームである。盤上には  $8 \times 2$  個の穴があいており、それぞれ上側 (a~h) が先手の陣地、下側 (i~p) が後手の陣地となっている。初期配置では図1のように1穴おきに石が4個ずつ置かれている。2人のプレイヤーは交互に自分の陣地にある穴から石を取って動かすことでゲームを進めていく。石の動かし方は、以下のルールに従う。

- (1) 自分の陣地の中から石のある穴を選択する。
- (2) その穴にある石をすべて取り出し、時計回りに1つずつ石を置いていく(取り出した石をすべて置くことを1配りとする)。最後の1つを置く穴の状態によってその後の動きが異なる。
- (3) 最後の1個を置く穴に石が1つ以上あれば、2を繰り返す。
- (4) 最後の1個を置く穴に石がなければその穴で終了し、その向かい側の穴にある石を自分の取り

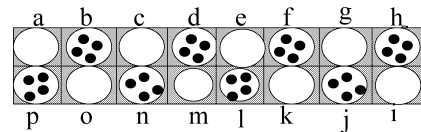


図1 初期配置  
Fig. 1 Initial state.

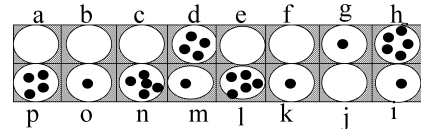


図2 1手移動 (f を選択)  
Fig. 2 First move (select f).

分としてボードから取り除く(これで1手が終了する)。

手の中の石を最後に置く穴に石が存在する限り1配りを続ける。1手は1配り以上で構成される。1手が終了した段階で、どちらかの陣地に石がすべてなくなった時点でゲームは終了する。残っている石はすべてその陣地のプレイヤーの取り分となる。そして最終的に取った石の数が多いほうが勝ちである。

たとえば、初期配置から先手が  $f$  を選択すると、 $f$  から石を取り  $g \rightarrow h \rightarrow i \rightarrow j$  と石を1つずつ置いていく。これで1配りである。 $j$  には石が初めからあるので、次に  $j$  の石をすべて取り出し(5個)、 $k \rightarrow l \rightarrow m \rightarrow n \rightarrow o$  と続けて石を置いていく。 $o$  には石がなかったので、移動はそこで終了し、反対側 ( $b$ ) にある4個の石を先手の取り分とし、後手に移る(図2)。

もしここで  $o$  に石があれば再び  $o$  から石を取り出し、時計回りに石を置いていく動作を繰り返すことになる。したがって、石を置く動作は何周も繰り返される場合があり、1つの穴の石の数は1手の間に複数回変化する可能性がある。

### 2.2 特徴

BAO はチェッカやオセロなどと同様に、2人零和有限確定完全情報ゲームである。その探索空間は比較的小さく、状態数は約  $10^{12}$  (チェッカは  $10^{18}$ , オセロは  $10^{28}$ )、ゲーム木の大きさは約  $10^{16}$  である (チェッカは  $10^{31}$ , オセロは  $10^{58}$  である<sup>10)</sup>)。そのためコンピュータに高い処理能力があればゲームの結果は早い段階で分かってしまう。

本論文では単純なルールを採用した。

このルールだと、1手が停止しない場合があるが、その場合ゲームでは途中で打ち切り、引分けとする。

また、チェスや将棋に比べ可能手は最大 8 と少ないが、1 手で局面が大きく変わり、その影響が盤全体にまで及ぶという特徴がある。このような特徴は他の同種のボードゲームには見られない。1 手の間に盤面が大局的に変化してしまい、しかも 1 つの穴の状態が何度も変化するので、先の局面を人間の手に読むことは困難である。特に石の数が多し序盤戦ではこの傾向は顕著であり、人間同士の勝負では中盤戦までは混沌としたゲーム運びになることが多く、序盤から中盤戦での戦略が重要になる。この特徴が BAO のコンピュータ上での実装をやすくしているのと同時に人間同士の対戦をおもしろくする要因の 1 つになっている。

BAO を実装して動きを解析した結果、特定の局面のときおよびそのときに選択される穴によって、1 手の間に盤面の変化は停止せず周期的な動きをすることが分かった<sup>3),4)</sup>。これは 1 配りの最後に置く穴に石が 1 つでも存在する限り石を配る動作が続くために起こる BAO の特徴である。このときの周期は局面によって異なり、石の配置がずれた盤面がその一連の流れの中で出現する。この種の周期的な動きは、約 1 万局面に対して 1 回の割合でしか起こらないため、実際のゲームをするうえでの妨げにはならない。もう一つの周期性として、手番交代しながらも盤上の石の数が減らずゲーム全体の終了に至らない場合、いわゆる千日手が考えられる。いずれの場合も初期配置から人間の手に周期性が出現する手筋を予想することは困難である。

2.3 準備

以後は、説明の簡単化のため盤の大きさを 4 × 2 に制限する。各穴に対して、先手側の端から時計回りに 0, 1, 2, ..., 7 という名前をつけ、そこに存在する石の数を  $\alpha_0, \dots, \alpha_7$  とする。このとき盤面  $S$  とその盤面で次に石を取る穴を  $x$  を、

$$S = (\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7) \quad x$$

( $\alpha_0 \dots \alpha_7$  は 0 または正の整数)

で表す。先手、もしくは後手が手番を開始する穴の位置を表す場合は  $[x]$  と記述する。たとえば図 3 の初期配置  $S_0$  で穴 3 を選択するとき、

$$S_0 = (0, 4, 0, 4, 0, 4, 0, 4) \quad [3]$$

と表せる。

このとき盤面は以下のように  $S_0 \rightarrow S_0_2 \rightarrow \dots \rightarrow S_0_7$  と変化し、6 配りで 1 手が終了する (図 4)。

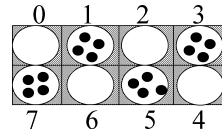


図 3 4 × 2 初期配置  $S_0$   
Fig. 3 4 × 2 initial state  $S_0$ .

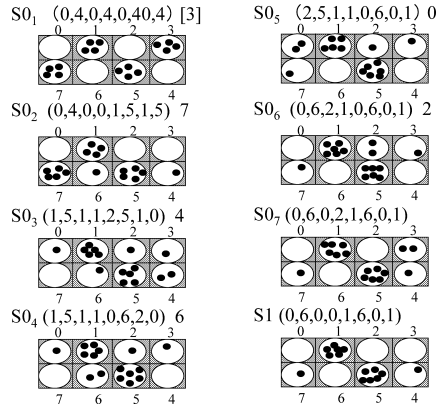


図 4 1 手移動 (4 × 2)  
Fig. 4 First move (4 × 2).

- $S_{0_1} = (0, 4, 0, 4, 0, 4, 0, 4) \quad [3]$
- $S_{0_2} = (0, 4, 0, 0, 1, 5, 1, 5) \quad 7$
- $S_{0_3} = (1, 5, 1, 1, 2, 5, 1, 0) \quad 4$
- $S_{0_4} = (1, 5, 1, 1, 0, 6, 2, 0) \quad 6$
- $S_{0_5} = (2, 5, 1, 1, 0, 6, 0, 1) \quad 0$
- $S_{0_6} = (0, 6, 2, 1, 0, 6, 0, 1) \quad 2$
- $S_{0_7} = (0, 6, 0, 2, 1, 6, 0, 1) \quad (4 \text{ で停止})$
- $S_1 = (0, 6, 0, 0, 1, 6, 0, 1) \quad \text{後手の手番}$

$S_1 = S$  とし、盤面  $S_i$  ( $i = 1, 2, \dots$ ) から 1 配りの後を盤面を  $S_{i+1}$  とすると、1 配りごとの盤面の変化を表す列  $S_1 \rightarrow S_2 \rightarrow \dots$  が一意に決まる。石が 0 個の穴で配り終わった盤面が  $S_n$  のとき、反対側の石を自分の取り分として 1 手が終了して手番が交代する。このときの 1 手は  $n - 1$  配りで構成される。石が 0 個の穴で配り終わらなければ手番の交代は行われず、列  $S_1 \rightarrow S_2 \rightarrow \dots$  は無限列になる。

$S, S'$  が以下の関係にあるとき、 $S'$  を  $S$  から  $t$  ずれた盤面と呼ぶ。

盤面  $S, S'$  をそれぞれ

$$S = (\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7)$$

$$S' = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7)$$

( $\alpha_a, \beta_b$  ( $0 \leq a, b \leq 7$ ) は 0 以上の整数)

とする。

$$\forall i; 0 \leq i \leq 7 \text{ に対し } \exists t; 0 \leq t \leq 7,$$

本論文では、盤面の状態を表す用語として、1 手終了することに更新される場合は「局面」、1 手の間に変化する場合は「盤面」と記述する。

$$\alpha_i = \begin{cases} \beta_{i+t} & (\text{if } i+t \leq 7) \\ \beta_{i+t-8} & (\text{if } i+t > 7) \end{cases}$$

局面  $S$  で穴  $x$  ( $x = 0, 1, \dots, 7$ ) を選択することで 1 配り目が始まり, 1 配りごとに  $S_1(=S) \rightarrow S_2 \rightarrow \dots$  と盤面が変化するとき,  $S_i$  ( $i \geq 2$ ) で初めて  $S$  からいくつかずれた盤面が現れたとする. このとき  $S_1(=S) \rightarrow S_2 \rightarrow \dots \rightarrow S_i$  をループと呼ぶ. また  $S$  から  $S_i$  までの間に穴  $x$  に石を配る回数をこのループの周期,  $S_1$  をループの開始盤面, そのとき選択する穴をループの開始位置と呼ぶ.

### 3. CCS による記述

#### 3.1 プロセス代数 CCS

プロセス代数は, 様々な現象を代数の形でモデル化し, その動作を代数の計算過程として表現した体系である. 互いに通信しながら並行動作する動作主体をプロセスという基本構成要素としてシステムを記述できるため, 並行動作に対する理論的解析を行うのに適している.

CCS (Calculus of Communicating Systems) は Milner によって提案されたプロセス代数の 1 つである<sup>12)</sup>. プロセス  $P$  は以下のように定義される.

$P ::= \emptyset$	停止
$a.P$	入力
$\bar{a}.P$	出力
$\tau.P$	内部計算
$P+Q$	選択動作
$P Q$	並行動作
$P \setminus L$	制限

$\emptyset$  はプロセスの停止を表す.  $a.P$ ,  $\bar{a}.P$  はそれぞれメッセージ  $a$  の入力と出力を表す.  $\tau$  は外側から観測されない動作である.  $P+Q$  は選択動作で  $P$  もしくは  $Q$  が非決定的に選択される.  $P|Q$  は並行動作である.  $P \setminus L$  は制限であり,  $L$  は外部からは観測できないメッセージの集合を表す.

また CCS の遷移の規則は以下のように定義される.

$$\begin{aligned} Act & \frac{}{\alpha.E \xrightarrow{\alpha} E} & Sum_j & \frac{E_j \xrightarrow{\alpha} E'_j}{\sum_{i \in I} E_i \xrightarrow{\alpha} E'_j} (j \in I) \\ Com_1 & \frac{E \xrightarrow{\alpha} E'}{E|F \xrightarrow{\alpha} E'|F} & Com_2 & \frac{F \xrightarrow{\alpha} F'}{E|F \xrightarrow{\alpha} E|F'} \\ Com_3 & \frac{E \xrightarrow{l} E' \quad F \xrightarrow{\bar{l}} F'}{E|F \xrightarrow{\tau} E'|F'} \end{aligned}$$

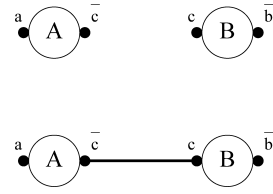


図 5 遷移の例

Fig. 5 Example of transition.

$$Res \frac{E \xrightarrow{\alpha} E'}{E \setminus L \xrightarrow{\alpha} E' \setminus L} (\alpha, \bar{\alpha} \notin L)$$

$$Rel \frac{E \xrightarrow{\alpha} E'}{E[f] \xrightarrow{f(\alpha)} E'[f]} \quad Con \frac{P \xrightarrow{\alpha} P'}{A \xrightarrow{\alpha} P'} (A \stackrel{def}{=} P)$$

$\alpha$  はラベルと呼ばれ入出力メッセージに相当する.  $L$  はラベルの集合,  $f$  はラベルを変更する関数である. ここで

$$\alpha.P \xrightarrow{\alpha} P$$

は  $\alpha.P$  が  $\alpha$  を受け取って  $P$  に変化することを表す. 遷移の簡単な例を示す. 以下のように定義される  $A$ ,  $B$  2 つのプロセスを考える.

$$\begin{aligned} A & \stackrel{def}{=} a.A' & B & \stackrel{def}{=} c.B' \\ A' & \stackrel{def}{=} \bar{c}.A & B' & \stackrel{def}{=} \bar{b}.B \end{aligned}$$

プロセス  $A$  は  $a$  を受け取り,  $A'$  に変わる.  $A'$  は  $c$  を出力し,  $A$  にもどる. 一方  $B$  は  $c$  を受け取り,  $B'$  に変化した後,  $b$  を出力する. この 2 つのプロセスの並行動作を考える.

プロセスの合成は  $A|B$  で表される (図 5).

$A$  から出力された  $c$  が  $B$  の入力になるため, 2 つのプロセスの遷移が同時におこる. これは内部計算  $\tau$  で記述され, 外部からは観測できない.

$$A' \xrightarrow{\bar{c}} A \quad B \xrightarrow{c} B'$$

に対し,  $(Com_3)$  を適用すると,

$$A'|B \xrightarrow{\tau} A|B'$$

が推論される. ここで,  $c$ ,  $\bar{c}$  は内部計算のみに使われ, 新たなプロセスとの通信が生じてもこのポートを通じた通信はできない. これを制限を使って

$$(A|B) \setminus \{c, \bar{c}\}$$

と表す.

#### 3.2 BAO の記述

BAO を CCS を用いて記述する. BAO の動きには並行性はないが, 各穴それぞれを独立したプロセス, 石をプロセス間のメッセージと見なすと, 穴の間にお

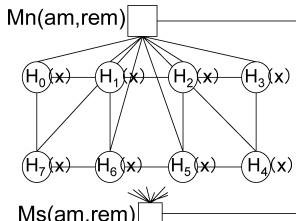


図 6 通信経路

Fig. 6 Communication path.

ける石の動きはプロセス通信における入出力と非常に似ている．そのため，CCS を使うと動作を反映した自然な記述ができる．また，マネージャプロセスとして先手用と後手用の 2 つを用意する．マネージャは，選択する穴を決めるメッセージやゲームが終了していないかどうか調べるメッセージを出す．

- $H_i(x)$  ( $i = 0, \dots, 7$ ) 盤上の各穴
- $Mn(am, rem)$  先手用マネージャ
- $Ms(am, rem)$  後手用マネージャ

ただし， $x$  は対応する穴に存在する石の数を表している．また， $am$  はそれぞれ獲得した石の合計， $rem$  は 1 手が終了したとき盤上の石を獲得した側に存在する石の合計で，ゲームの終了判定に使う (図 6)．

各プロセスの動作規則を説明する．

まず先手用マネージャ  $Mn(am, rem)$  が穴を選択する． $\overline{nselect}_i$  は選択する穴の位置を決めるメッセージである ( $i = 0, \dots, 3$ )．

$$Mn(am, rem) = \sum \overline{nselect}_i.Mn(am, rem)$$

選択された穴は，石をすべて隣の穴に送る．上式における  $\overline{nsend}_{i+1}(x)$  は隣の穴に石の数  $x$  を送るメッセージである．

$$H_i(x) = \overline{nselect}_i.\overline{nsend}_{i+1}(x).H_i(0)$$

各穴は送られてきた石を 1 つ受け取り，隣の穴に送る．

$$H_i(x) = \overline{nsend}_i(y).\overline{nsend}_{i+1}(y-1).H_i(x+1)$$

もし残りの石が 1 つのとき，すなわち  $\overline{nsend}_i(1)$  ならば，石を穴に置き，1 配りの終了をマネージャに知らせる．

$$H_i(x) = \overline{nsend}_i(1).\overline{nput}_i.H_i(x+1)$$

$$Mn(am, rem) = \overline{nput}_i.\overline{nreturn}_i.Mn(am, rem)$$

マネージャから同じ穴にメッセージを返す．1 配り分移動した後は，配り終わった穴の状態によって 2 通りの動きをする． $H_i(x)$  ( $x \neq 1$ ) のとき，再び石を取り出し移動を再開する．

$$H_i(x) = \overline{nreturn}_i.\overline{nsend}_{i+1}(x).H_i(0)$$

$H_i(1)$  のとき，移動はその穴で停止して，反対側の石

を取る．

$$H_i(1) = \overline{nreturn}_i.\overline{nsstop}_{7-i}.H_i(1)$$

$$H_i(x) = \overline{nsstop}_i.\overline{nget}_i(x).H_i(0)$$

これで 1 手の盤上での動きが終了する．先手側マネージャに石を送り，ゲームが終了していないか調べる．石が先手側からマネージャに送られてきたら，先手用マネージャは  $\overline{ncheck}(0)$  をプロセス  $H_0(s_0)$  に送り，順に先手側の穴が空かどうか調べた後，プロセス  $H_3(s_3)$  から  $\overline{nchecked}(y)$  を受け取る．このとき  $y$  は先手側の穴に残っている石の数である．後手側からマネージャに送られてきた場合は， $\overline{scheck}(0)$  をプロセス  $H_4(s_4)$  に送り，後手側を順に調べた後，プロセス  $H_7(s_7)$  から  $\overline{schecked}(y)$  を受け取る．

$$Mn(am, rem) = \overline{nget}_i(x).\overline{ncheck}(0).$$

$$\overline{nchecked}(y).Mn'(am+x, y) \\ (i = 0, \dots, 3)$$

$$Mn(am, rem) = \overline{nget}_i(x).\overline{scheck}(0).$$

$$\overline{schecked}(y).Mn'(am+x, y) \\ (i = 4, \dots, 7)$$

石を取り出した側の盤上に石が残っているかを調べる．

先手側

$$H_0(x) = \overline{ncheck}(0).\overline{check}_1(x).H_0(x)$$

$$H_1(x) = \overline{check}_1(y).\overline{check}_2(y+x).H_1(x)$$

$$H_2(x) = \overline{check}_2(y).\overline{check}_3(y+x).H_2(x)$$

$$H_3(x) = \overline{check}_3(y).\overline{nchecked}(y+x).H_3(x)$$

後手側

$$H_4(x) = \overline{scheck}(0).\overline{check}_5(x).H_4(x)$$

$$H_5(x) = \overline{check}_5(y).\overline{check}_6(y+x).H_5(x)$$

$$H_6(x) = \overline{check}_6(y).\overline{check}_7(y+x).H_6(x)$$

$$H_7(x) = \overline{check}_7(y).\overline{schecked}(y+x).H_7(x)$$

$Mn'(am, rem)$  はゲームが終了するかどうかを確認する状態である． $rem$  の値によって 2 通りの動きをする．もし石がなければ (すなわち  $rem = 0$  ならば) ゲームを終了し，石があれば (すなわち  $rem \neq 0$  ならば) 手番を交代する．

$$Mn'(am, 0) = \overline{end}.\overline{finish}_n.\emptyset \text{ 終了}$$

$$Mn'(am, x) = \overline{change}_n.Mn(am, x) \text{ 手番の交代}$$

$$Ms(am, rem) =$$

$$\sum_{i=4, \dots, 7} \overline{change}_n.\overline{sselect}_i.Ms(am, rem)$$

後手側マネージャの動きも同様に記述できる．

### 3.3 状態遷移の例

初期状態は図 3 のように表される．これを CCS で表現すると，

$$H_0(0)|H_1(4)|H_2(0)|H_3(4)|H_4(0)|H_5(4)$$

$$|H_6(0)|H_7(4)|\sum \overline{nselect_i}.Mn(0,8)|Ms(0,8)$$

となる .

ここで  $H_3(4)$  を選択したとして、動きを調べる .

$$\begin{aligned} Mn(0,8) &= \overline{nselect_3}.Mn(0,8) \\ H_3(4) &= \overline{nselect_3}.\overline{nsend_4}(4).H_3(0) \\ H_4(0) &= \overline{nsend_4}(4).\overline{nsend_5}(3).H_4(1) \\ H_5(4) &= \overline{nsend_5}(3).\overline{nsend_6}(2).H_5(5) \\ H_6(0) &= \overline{nsend_6}(2).\overline{nsend_7}(1).H_6(1) \\ H_7(4) &= \overline{nsend_7}(1).\overline{nput_7}.H_7(5) \\ Mn(0,8) &= \overline{nput_7}.\overline{nreturn_7}.Mn(0,8) \\ H_7(5) &= \overline{nreturn_7}.\overline{nsend_0}(5).H_7(0) \\ H_0(0) &= \overline{nsend_0}(5).\overline{nsend_1}(4).H_0(1) \end{aligned}$$

以下同様に移動を続けていく . すると  $H_4(0)$  へ戻ってきた段階で、一連の動きが停止し、石を獲得する処理を行う .

$$\begin{aligned} H_3(1) &= \overline{nsend_3}(2).\overline{nsend_4}(1).H_3(2) \\ H_4(0) &= \overline{nsend_4}(1).\overline{nput_4}.H_4(1) \\ Mn(0,8) &= \overline{nput_4}.\overline{nreturn_4}.Mn(0,8) \\ H_4(1) &= \overline{nreturn_4}.\overline{nstop_3}.H_4(1) \\ H_3(2) &= \overline{nstop_3}.\overline{nget_3}(2).H_3(0) \end{aligned}$$

ここで石を取り出す穴  $H_3(0)$  と先手用マネージャ  $Mn(0,8)$  間で通信が行われ、獲得した石を渡す . その後盤面が終了していないか確認を行う .

$$\begin{aligned} Mn(0,8) &= \\ & \overline{nget_3}.\overline{ncheck}(0).\overline{nchecked}(7).Mn'(1,7) \\ H_0(0) &= \overline{check_n}.\overline{check_1}(0).H_0(0) \\ H_1(7) &= \overline{check_1}.\overline{check_2}(7).H_1(7) \\ H_2(0) &= \overline{check_2}.\overline{check_3}(7).H_2(0) \\ H_3(0) &= \overline{check_3}.\overline{nchecked}(7).H_3(0) \\ Mn'(1,7) &= \overline{change_n}.Mn(1,7) \\ Ms(0,8) &= \sum \overline{change_n}.\overline{sselect_i}.Ms(0,8) \end{aligned}$$

これで 1 手が終了し、後手の手番となる . 全体は

$$|H_0(0)|H_1(7)|H_2(0)|H_3(0)|H_4(1)|H_5(7) \\ |H_6(0)|H_7(1)|Mn(0,7)|\overline{sselect_i}.Ms(0,8)$$

となっている . この動きをゲーム終了まで行うことで、BAO の動きは記述される .

先手が終了局面に至った場合を考える . 終了局面の 1 手前の局面を

$$|H_0(0)|H_1(0)|H_2(0)|H_3(1)|H_4(0)|H_5(0) \\ |H_6(1)|H_7(1)|\overline{nselect_3}.Mn(8,2)|Ms(6,1)$$

とする .

先手用マネージャが 3 を選択する . 石を動かし、先手用マネージャ  $Mn(8,2)$  に 1 手の終了を伝える .

$$H_3(1) = \overline{nselect_3}.\overline{nsend_4}(1).H_3(0)$$

$$H_4(0) = \overline{nsend_4}(1).\overline{nput_4}.H_4(1)$$

$$Mn(8,2) = \overline{nput_4}.\overline{nreturn_4}.Mn(8,2)$$

$$H_4(1) = \overline{nreturn_4}.\overline{nstop_3}.H_4(1)$$

$$H_3(0) = \overline{nstop_3}.\overline{nget_3}(0).H_3(0)$$

先手側の盤面を調べ、終了局面かどうか確認する .

$$\begin{aligned} Mn(8,2) &= \\ & \overline{nget_3}.\overline{ncheck}(0).\overline{nchecked}().Mn'(8,0) \\ H_0(0) &= \overline{check_n}.\overline{check_1}(0).H_0(0) \\ H_1(0) &= \overline{check_1}.\overline{check_2}(0).H_1(0) \\ H_2(0) &= \overline{check_2}.\overline{check_3}(0).H_2(0) \\ H_3(0) &= \overline{check_3}.\overline{nchecked}(0).H_3(0) \end{aligned}$$

$Mn'(8,0)$  となっているので終了局面である . そこで、先手用マネージャから後手用マネージャとプロセス  $H_0(0)$  へプロセスを停止させるメッセージを送る . さらに  $H_0(0)$  から順にプロセスを停止させる .

$$\begin{aligned} Mn'(8,0) &= \overline{end}.\overline{finish_n}.\emptyset \\ H_0(0) &= \overline{end}.\overline{end_1}.\emptyset \\ H_1(0) &= \overline{end_1}.\overline{end_2}.\emptyset \\ H_2(0) &= \overline{end_2}.\overline{end_3}.\emptyset \\ H_3(0) &= \overline{end_3}.\overline{end_4}.\emptyset \\ H_4(1) &= \overline{end_4}.\overline{end_5}.\emptyset \\ H_5(0) &= \overline{end_5}.\overline{end_6}.\emptyset \\ H_6(1) &= \overline{end_6}.\overline{end_7}.\emptyset \\ H_7(1) &= \overline{end_7}.\emptyset \\ Ms(6,1) &= \overline{finish_n}.\emptyset \end{aligned}$$

終了局面では終了を判定したマネージャからプロセス  $H_0(x)$  と相手側のプロセスを停止させる命令を送る . プロセス  $H_0(x)$  から順番に穴のプロセスを停止させる命令を流すことで全体を終了させている .

マネージャは非決定的に穴を選択するが、その後システムはマネージャプロセスが手番の交代を行うまで決定的に動く .

### 3.4 周期的振舞い

図 7 は 1 手が停止しないループの開始盤面の一例である . 図 7 ではその開始位置は  $H_0(3)$  である .

この局面を CCS で表す .

$$S_0 = H_0(3)|H_1(1)|H_2(0)|H_3(1)|H_4(0)|H_5(1) \\ |H_6(0)|H_7(1)|\overline{nselect_0}.Mn(am,rem) \\ |Ms(am,rem)$$

この局面に先ほど記述したルールを適用する . すると  $S_0$  は以下のように遷移する .

$$S_1 = H_0(1)|H_1(3)|H_2(1)|H_3(0)|H_4(1)|H_5(0) \\ |H_6(1)|H_7(0)|\overline{nreturn_1}.Mn(am,rem) \\ |Ms(am,rem)$$

$S_1$  は  $\overline{nsend_1}(3)$  から始まった動きが、盤上を 1 周し  $H_1(3)$  まできた段階である . すなわち  $S_0$  から 4

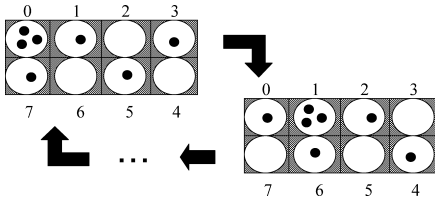


図 7 1 手が終了しないループの例  
Fig. 7 Loop of non-terminated move.

配り後,  $S_1$  の状態に遷移する.  $S_1$  は  $S_0$  から 1 ずれた盤面に相当する. したがってこの作業を有限回繰り返すことで石の配置が元の状態  $S_0$  に戻る.

$$S_8 = H_0(3)|H_1(1)|H_2(0)|H_3(1)|H_4(0)|H_5(1)|H_6(0)|H_7(1)|\overline{nreturn_0}.Mn(am, rem)|Ms(am, rem)$$

さらに  $S_8$  から 4 配り進めた状態  $S_9$  は状態  $S_1$  と一致する. したがってこの状態はループに陥り, その石の配置は周期的に変化していることが分かる.

他のループでも同様に記述することが可能である.

#### 4. 1 手の停止条件の解析方法

##### 4.1 モデル

ここで, 穴どうしをつないだものをボード ( $B$ ) という 1 つのシステム, 先手用, 後手用をあわせたマネージャ ( $M$ ) をもう 1 つのシステムとすると, 各穴の間で授受される  $nsend_i(y)$   $check_i(y)$  などのメッセージは内部計算としてとらえられる.

$$B \stackrel{\text{def}}{=} (H_0(0)|H_1(4)|H_2(0)|H_3(4)|H_4(0)|H_5(4)|H_6(0)|H_7(4)) \setminus L_B$$

$$M \stackrel{\text{def}}{=} (Mn(0, 8)|Ms(0, 8)) \setminus L_M$$

$L_B$  は  $\{nsend_i, ssend_i, nstop_i, sstop_i, ncheck_i, scheck_i, end_i\}$ ,  $L_M$  は  $\{change_n, change_s, finish_n, finish_s\}$  であり, それぞれのシステムにおける内部計算を表す. 図 8 はこのモデルを図に表したものである.

したがって, 状態遷移規則に従うと, 外部から観測される動作は以下ようになる.

$$B|M \xrightarrow{nselect_i, \tau^*, \overline{nput_j}} B'|M$$

$$\xrightarrow{\overline{nreturn_j}, nget_{7-j}} B'|M'$$

$$\vdots$$

$M$  は選択する穴の位置の情報 ( $\overline{nselect_i}$ ) を送信し,  $B$  は受け取った情報 ( $nselect_i$ ) からルールに従って穴に存在する石の数を 1 配り分変化させた後,  $M$  に

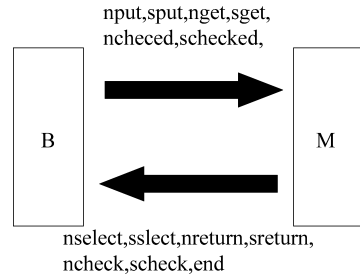


図 8 ボード, マネージャ間の通信モデル  
Fig. 8 Communication model between a board and a manager.

メッセージ ( $\overline{nput_i}$ ) を送信する.  $M$  からメッセージを返した後 ( $\overline{nreturn_i}$ ), 1 手が終了していなければ石の移動を再開する. 1 手が終了していれば,  $B$  から 1 手の終了を  $M$  に伝える ( $\overline{nget_i}$ ). その後  $M$  は盤の状態を調べるメッセージ ( $\overline{ncheck}$ ) を送信する. 局面が終了状態でない場合, 手番を交代して選択する穴の位置 ( $\overline{sselect_i}$ ) を  $B$  に送信する. いずれかの陣地のすべての穴に石がない終了状態であるならば,  $M$  は終了メッセージ ( $\overline{end}$ ) を出す. 終了メッセージが出た後はすべてのプロセスが終了し, メッセージは送られなくなる.

このモデルのメッセージを調べることでゲームの停止性の記述・検証が可能である. ゲーム全体が終了するための条件はある時点で  $\overline{end}$  メッセージがマネージャからボードへ送られることである. したがって, ゲーム全体が終了しない場合は,  $\overline{end}$  メッセージが永遠に送られないことに相当するが, この場合, 1 手が終了しない場合と, ある盤面から手番の交代が永遠に終わらないいわゆる千日手の場合がある. 前者は, 1 手の終了をつげるメッセージ  $\overline{nget_i}$  と  $\overline{sget_i}$  がともある時点以降ずっと観測されないことで分かり, 後者は, 逆に  $\overline{nget_i}$  と  $\overline{sget_i}$  がともに無限に頻りに観測されることで分かる.

##### 4.2 1 手の停止条件

盤面と着手が与えられるとそのときの手番が終了するかどうかは CCS の遷移規則によって検証可能だが, 逆に, 手番が終了する盤面を見つけたり, 千日手が存在するかどうかを検出するには CCS では難しい. そこで, 以下の 2 種類の方法で 1 手の停止条件を求める.

1 つは, 初期状態から到達可能かつそこから開始する 1 手が停止しないような局面をすべて求め, 得られた結果を一般化することによってその特徴を抽出する方法である. 局面の探索にあたっては, 計算量的問題

この処理は 1 配りごとの盤面を見るために行う.

を避けるため、単純な全探索をせずモデルチェッカ SPIN を利用する。SPIN では「必ず停止する」という仕様の検証を行うと停止しないモデルが反例として出力される。

もう 1 つの方法は、巻き戻しによる解析である。任意の盤面から任意の穴を選択してゲームのルールとは逆向きの動作を行い、それが停止するための条件を求める。

この後の 2 つの章では、これら 2 つの方法による解析結果を示し、得られた条件を比較する。

### 5. 一般化による解析

#### 5.1 モデルチェッカによる検証

モデル検査では一般に単一あるいは複数のプロセスの動きを状態の遷移としてモデル化し、有限の状態空間を対象に網羅的、機械的に探索することで、与えられた性質が成り立つか確認する。SPIN はモデル検査を行うツールの 1 つで、与えられた仕様が誤りを含む場合には反例を表示する機能を持つ。仕様は線形時相論理で記述する。線形時相論理は状態の遷移や時間の経過の観点からシステムの性質を記述するための論理体系であり、「いつかは必ず  $P$  が成り立つ」「ある時点以降ずっと  $P$  が成り立つ」などを様相オペレータを用いて表す。

#### 5.2 BAO の仕様記述と実験結果

仕様 1. 「ゲームは必ず終了する」この場合、1 手が終了しない場合と千日手の場合の両方が反例として得られる。

仕様 2. 「ある時点以降ずっと交代メッセージは観測されない」この場合、千日手が反例として得られる。

盤上の穴の数が  $4 \times 2$  について仕様 1, 2 を与えて SPIN で検証を行った結果、それぞれ 17 種類、5 種類の反例が得られた。表 1 に 1 手が終了しない場合を、表 2 に千日手になる場合を、それぞれ示す。

#### 5.3 1 手が停止しないループの解析

仕様 1 の反例すなわち初期状態から到達可能で手番が交代しないループ (表 1) の解析を行う。これらは周期 0 のもの ( $L1$ ) と周期 1 以上のもの ( $L2 \sim L12$ ) で異なる性質を持つ。

##### 5.3.1 周期 0 のループ

周期 0 のループでは、盤面  $S_1$  から 1 配りで達した盤面  $S_2$  が  $S_1$  からいくつかずれた盤面になる。

図 9 では盤面  $L1_1$  から 0 を選択することで 1 配り目が始まる。0 から石を 3 つすべて取り出し、 $1 \rightarrow 2 \rightarrow 3$  と石を置いていくと、3 に石を置いた段階で手の中の石はなくなるので、これで 1 配りが終了し、盤面は

表 1 手番が交代しないループ (左端を開始位置とする)

Table 1 Non-terminated move (starting from the leftmost point).

	盤面例	周期	ずれ
L1	(3,1,0,2,1,0,2,1)	0	3
L2	(3,1,0,1,0,1,0,1)	1	1
L3	(3,1,0,1,0,3,0,1)	2	2
L4	(5,1,0,1,0,1,0,1)	3	3
L5	(3,1,0,3,0,1,0,1)	2	2
L6	(3,1,0,3,0,3,0,1)	4	4
L7	(5,1,0,1,0,3,0,1)	6	6
L8	(5,1,0,3,0,1,0,1)	10	1
L9	(7,1,0,1,0,3,0,1)	30	3
L10	(3,1,0,1,0,1,0,4)	2	2
L11	(3,1,0,3,0,1,0,4)	2	2
L12	(3,3,0,1,0,1,0,4)	9	0

表 2 千日手

Table 2 Infinite turn-taking.

	盤面例	手筋
R1	(0,1,1,1,0,0,0,1)	2, 4, 2, 5, 1, 6, 0, 6, 1, 5
R2	(0,1,0,1,0,0,1,1)	7, 3, 4, 0, 6, 3, 7, 0, 4, 2
R3	(0,1,1,0,1,0,0,1)	1, 5, 2, 5, 1, 7, 3, 5
R4	(0,1,1,1,2,0,2,1)	2, 4, 3, 6, 1, 6, 3, 4, 2, 7
R5	(1,0,2,1,0,3,1,2)	6, 2

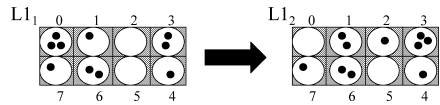


図 9 周期 0 のループ

Fig. 9 A loop with a cycle 0.

$L1_2$  になる。このとき、 $L1_2$  は  $L1_1$  から 3 つずれた盤面であり、周期は 0 である。

$$L1_1 = (3, 1, 0, 2, 1, 0, 2, 1) \quad [0]$$

$$L1_2 = (0, 2, 1, 3, 1, 0, 2, 1) \quad 3$$

⋮

一般に穴の数が  $m$  個、ループの開始位置に存在する石の数が  $n$  個のときに 1 配りでループになる開始盤面を考えると以下の 2 通りのパターンがあり、いずれも 1 配りで  $n$  個ずれた盤面になる。

##### (1) パターン 1

ループの開始盤面は  $(n, 0, 1, \dots, n-1)$ ,  $m = n+1$  となる。この盤面を  $S = (\alpha_0, \dots, \alpha_{m-1})$  として形式的に記述すると  $\alpha_i$  ( $0 \leq i \leq m-1$ ) は以下のようになる。

$$\alpha_i = \begin{cases} n & (i = 0) \\ (i - 1) \pmod{n} & (i \geq 1) \end{cases}$$

穴数が  $4 \times 2$  のとき、このパターンの開始盤面は



表 3 周期 0 のループ (パターン 1)  
Table 3 A loop with a cycle 0 (pattern 1).

開始位置の石の数	$k = 2$ のときの盤面	穴の数
2	(2,0,1)	3, 5, 7, 9, ...
3	(3,0,1,2)	4, 7, 10, ...
4	(4,0,1,2,3)	5, 9, 13, ...
5	(5,0,1,2,3,4)	6, 11, 16, ...

$$S = (7, 0, 1, 2, 3, 4, 5, 6) \quad [0]$$

となる。しかし盤面  $S$  に存在する石の総数は初期配置における石の総数を超えているので、初期配置からこの盤面に至ることはない。

表 3 にパターン 1 を満たす盤面をいくつか示す。

(2) パターン 2

ループの開始盤面は  $(n, n - 2, n - 3, \dots, 1, 0, n - 1, n - 2, \dots, 1, 0, \dots, n - 1, n - 2, \dots, 1)$ ,  $m = kn - 1$  ( $k \geq 2$ ) となる。この盤面を  $S = (\alpha_0, \dots, \alpha_{m-1})$  として形式的に記述すると  $\alpha_i$  ( $0 \leq i \leq m - 1$ ) は以下ようになる。ただし,  $p \geq 1$ 。

条件 (LOOP1)

盤面を  $S = (\alpha_0, \dots, \alpha_{m-1})$  とする。

$$\alpha_i = \begin{cases} n & (i = 0) \\ n - 1 & (i = pn) \\ n - 2 & (i = pn - n + 1) \\ \vdots & \\ 2 & (i = pn - 3) \\ 1 & (i = pn - 2) \\ 0 & (i = pn - 1) \end{cases}$$

パターン 2 の開始盤面

$$S_1 = (n, n - 2, n - 3, \dots, 1, 0, n - 1, n - 2, \dots, 1, 0, n - 1, \dots, 1)$$

は最も左端を選択して石を 1 配りすると盤面

$$S_2 = (0, n - 1, n - 2, \dots, 2, 1, n, n - 2, \dots, 1, 0, n - 1, \dots, 1)$$

となる。 $S_2$  は  $S_1$  から  $n$  個ずれた盤面である。図 10 は  $n = 3$  の場合のパターン 2 の盤面例である。表 4 にパターン 2 を満たす盤面をいくつか示す。この表から  $4 \times 2$  ではパターン 2 のループは 1 種類しか現れないことが分かる。実験結果から得られた 12 種類のループの中でもこのパターン 2 の盤面のみが周期 0 のものとして出現しており、初期配置から到達可能なことが分かる。

5.3.2 周期 1 以上のループ

周期が 1 以上のループの解析を行う。 $L_{2_1}$  のループは、4 配りの後  $L_{2_1}$  から 1 ずれた盤面が現れる。右

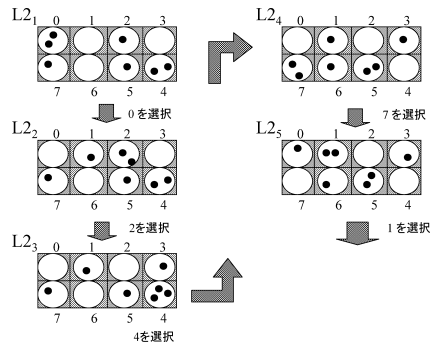


図 10 周期 1 のループ  
Fig. 10 A loop with a cycle 1.

表 4 周期 0 のループ (パターン 2)  
Table 4 A loop with a cycle 0 (pattern 2).

開始位置の石の数	$k = 2$ のときの盤面	穴の数
2	(2,0,1)	3, 5, 7, 9, ...
3	(3,1,0,2,1)	5, 8, 11, ...
4	(4,2,1,0,3,2,1)	7, 11, 15, ...
5	(5,3,2,1,0,4,3,2,1)	9, 14, 19, ...

側の文字は配りはじめの穴を示す。

$$\begin{aligned} L_{2_1} &= (2, 0, 1, 0, 2, 1, 0, 1) \quad [0] \\ L_{2_2} &= (0, 1, 2, 0, 2, 1, 0, 1) \quad 2 \\ L_{2_3} &= (0, 1, 0, 1, 3, 1, 0, 1) \quad 4 \\ L_{2_4} &= (0, 1, 0, 1, 0, 2, 1, 2) \quad 7 \\ L_{2_5} &= (1, 2, 0, 1, 0, 2, 1, 0) \quad 1 \\ &\vdots \end{aligned}$$

盤上の穴の数が  $2n$  とする。開始盤面が  $L_1$  である周期 1 以上のループ  $L_1 \rightarrow L_2 \rightarrow \dots \rightarrow L_k$  において、各配りの配りはじめの穴を最も左側になるようにずらした列  $L'_1 \rightarrow L'_2 \rightarrow \dots \rightarrow L'_k$  を考える。このとき以下の条件を満たす  $L'_i$  ( $1 \leq i \leq k$ ) が存在する。

条件 (LOOP2)

盤面  $S$  を  $S = (\alpha_1, \alpha_2, \dots, \alpha_{2n})$  とする。

- $\alpha_1$  は 3 以上の奇数である
- $\alpha_2, \alpha_4, \dots, \alpha_{2n-2}$  はすべて奇数である
- $\alpha_3, \alpha_5, \dots, \alpha_{2n-1}$  はすべて偶数である
- $\alpha_{2n}$  は盤上の石の総数が奇数のときは  $\alpha_1$  未満の奇数であり、偶数のときは  $\alpha_1$  以上の偶数である

盤上の穴が  $4 \times 2$  のとき、石の総数が奇数のループ

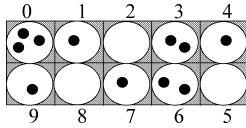


図 11 条件 (LOOP2) を満たさない開始盤面

Fig. 11 The initial state not satisfying (LOOP2).

は 8 通り, 偶数のものは 3 通りある .

$L_{2_1} \rightarrow \dots \rightarrow L_{2_5}$  においては  $L_{2_3} = (3, 1, 0, 1, 0, 1, 0, 1)$  がこの条件を満たす盤面になる . これは  $4 \times 2$  のときのループで石の総数が最少の場合である .

盤上の穴の数が  $2n$  とする .  $n < 4$  ならば条件 (LOOP2) はループを形成するための必要十分条件になるが,  $n \geq 5$  ならば必要条件であるが十分条件ではない . すなわち条件 (LOOP2) を満たさないループが存在する .

図 11 にそのような盤面を示す . この盤面を  $L_1$  とし, 0 を選択して配りはじめると以下のように盤面が変化し,  $L_5$  で  $L_1$  から 1 つずれた盤面が現れる .

- $L_1 = (3, 1, 0, 2, 1, 0, 2, 1, 0, 1)$  [0]
- $L_2 = (0, 2, 1, 3, 1, 0, 2, 1, 0, 1)$  3
- $L_3 = (0, 2, 1, 0, 2, 1, 3, 1, 0, 1)$  6
- $L_4 = (0, 2, 1, 0, 2, 1, 0, 2, 1, 2)$  9
- $L_5 = (1, 3, 1, 0, 2, 1, 0, 2, 1, 0)$  1
- ⋮

このループは条件 (LOOP2) を満たさない .

$n \geq 5$  の場合はこのような盤面も存在するため初期配置から到達可能なループの数は飛躍的に増大する .

### 5.3.3 周期 1 以上のループにおける周期とずれの関係

盤上の穴が  $4 \times 2$  で周期 1 以上でループになる場合, そのループの周期とずれの間に関係深い関係が見られる .

盤面  $L_{4_1} = (5, 1, 0, 1, 0, 1, 0, 1)$  を開始盤面とするループは以下のようになり, 周期 3 で 3 個ずれる .

- $L_{4_1} = (5, 1, 0, 1, 0, 1, 0, 1)$  [0]
- $L_{4_2} = (0, 2, 1, 2, 1, 2, 0, 1)$  5
- ⋮
- $L_{4_{11}} = (1, 2, 0, 4, 1, 0, 1, 0)$  1
- $L_{4_{12}} = (1, 0, 1, 5, 1, 0, 1, 0)$  3
- ⋮

盤面  $L_{9_1} = (3, 1, 2, 3, 0, 1, 0, 1)$  を開始盤面とするループは以下のようになり, 周期 10 で 1 個ずれる .

- $L_{9_1} = (3, 1, 2, 3, 0, 1, 0, 1)$  [0]
- $L_{9_2} = (0, 2, 3, 4, 0, 1, 0, 1)$  3

- ⋮
- $L_{9_{30}} = (0, 2, 1, 2, 3, 0, 1, 2)$  7
- $L_{9_{31}} = (1, 3, 1, 2, 3, 0, 1, 0)$  2

実験結果の解析により, どのループにおいてもそのずれが  $n$  のとき周期は  $n + 9 \times k$  となる ( $k = 0, 1, \dots$ ) ことが判明した . したがって周期と穴数が互いに素であれば, (周期  $\times$  穴数) だけ回って初めてループの開始盤面と完全に一致するものが現れる . 同様に調べると, 一般に盤上の穴の数  $m$  のとき, 周期 1 以上のループの周期は, そのずれが  $n$  ならば  $n + (m + 1) \times k$  になる ( $k = 0, 1, \dots$ ) .

## 6. 巻き戻しによる解析

### 6.1 ロールバック

BAO の 1 手の動きの巻き戻し (ロールバック) を使って 1 手の停止条件を求める . なお, 穴は環状につながっていると見なしてよいので, 以後, 盤面  $(\alpha_0, \dots, \alpha_{m-1})$  において  $\alpha_i$  と書いたときには  $\alpha_j$  ( $j = i \text{ mod } m$ ) を意味するものとする .

ロールバックの動きは以下のルールに従う .

- (1) ロールバックを始める穴を 1 つ決め, その穴から石を 1 つ取る .
- (2) 反時計回りに動き, 石がない穴に到達するまで穴から石を 1 つずつ取り, 隣の穴に移るという作業を繰り返す .
- (3) 穴に石がなければ, 取り出した石をすべてその穴に置く (これが順方向の 1 配りに相当する) .
- (4) 置いた穴に石が 2 つ以上あれば石を 1 つ取り出し (2) と (3) を行う . 石が 1 つしかない場合, ロールバックを停止する .

1 手のロールバックを考える場合, その終了状態からロールバックを始める . このときその 1 手の開始状態が与えられていないと, 戻し終わる位置を特定することができない . なぜなら (3) の段階で石を置いた穴に最後においたものも含めて石が 2 つ以上ある場合, 順方向における開始状態がこの状態なのかさらに巻き戻す必要があるかは判定できないからである . そのため, ロールバックの操作としてはここで停止せずに (4) を行う . したがって, 戻し終わる位置が与えられていない場合, ロールバックが停止するのは空の穴に石を 1 つおいたときのみである . この状態をロールバックの停止状態  $\hat{T}$  と呼ぶ .

実際の配列  $RB1$  を例として考える (図 12) .

$$RB1 = (1, 2, 0, 1, 2, 0, 0, 3) [1]$$

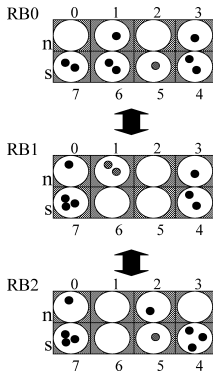


図 12 ロールバック例  
Fig. 12 Example of rollback.

$\alpha_1 = 2$  から順方向の移動を開始する．すると盤面は 2 配りで RB2 の配置になる．

$$RB2 = (1, 0, 1, 0, 3, 1, 0, 3)$$

ここで RB2 からのロールバックを考える． $\alpha_4 = 3$  からロールバックを始めると，RB1 と同じ盤面の状態まで戻すことができる．しかしまだロールバックを行うことは可能である．さらにロールバックを行うと，逆方向に数配り後 RB0 の状態でロールバックは停止する．

$$RB0 = (0, 1, 0, 1, 2, 1, 2, 2)$$

### 6.2 ロールバックの停止条件

一般に  $m$  個の穴を持つ盤面に対し，以下の命題が成り立つ．

#### 命題 1

盤面  $(\alpha_0, \dots, \alpha_{m-1})$  において  $\alpha_i = \alpha_{i+1} = 0$  となるような  $i$  が存在する場合，そこからロールバックを行った結果，有限回の遷移で必ず  $\hat{T}$  に至る．

#### 証明)

盤面が 2 つ連続する空の穴を含むためには，ロールバックの開始時に取りはじめとして指定された穴の (反時計回りの向きで) すぐ隣が空の場合と，途中で空の穴が 2 つ連続して現れる場合の 2 通りがある．前者の場合は明らかにそこで停止する．後者の場合，まずルール (3) で取り出していた石を  $i+1$  番目の穴にすべて置く．石が 1 つしかなければそこで  $\hat{T}$  になる．2 つ以上あれば (4) で石を 1 つ取り出し，空である隣の穴に石を置く．ここで  $\hat{T}$  に至る． ■

ロールバックでは，穴に石がある場合，その石の数が 1 つ減り，穴に石がないときのみ，穴に存在する石の数が増える．そのためロールバックの流れの中で空の穴が連続するためには，同じ石の数を含む穴が 2 つ連続して現れればよい．また， $\alpha_i = \alpha_{i+1}$  で  $\alpha_{i-1} \neq \alpha_i$

かつ  $\alpha_{i+2} \neq \alpha_i$  の場合は  $\alpha_{i+1}$  の穴から石を取りはじめる必要がある．したがって，命題 1 を一般化した命題 2 が成り立つ．

#### 命題 2

盤面  $(\alpha_0, \dots, \alpha_{m-1})$  において  $\alpha_i = \alpha_{i+1}$  となるような  $i$  が存在すれば，そこからロールバックを行った結果，有限回の遷移で必ず  $\hat{T}$  に至る．ただし， $\alpha_{i-1} \neq \alpha_i$  かつ  $\alpha_{i+2} \neq \alpha_i$  の場合は  $\alpha_{i+1}$  の穴から石を取りはじめる必要がある．

さらに，命題 2 の条件を満たす盤面を  $\hat{R}$  とおくと，以下の命題が成り立つ．

#### 命題 3

盤面  $(\alpha_0, \dots, \alpha_{m-1})$  が以下の停止パターン (PT1), (PT2) のいずれかを満たせば，そこからロールバックを行うとき，有限回の遷移で  $\hat{R}$  に至る．

以下で， $x_t$  は 1 以上の任意の整数 ( $t \geq 0$ )， $i$  は  $0 \leq i \leq m-1$  を表す．

#### (1) 停止パターン 1 (PT1)

(1.1)  $n \geq 3$  のとき

$$\begin{aligned} & \vdots \\ & \alpha_i = x_0 \\ & \alpha_{i+1} = 0 \\ & \alpha_{i+2} = n \\ & \alpha_{i+3} = x_1 \\ & \vdots \\ & \alpha_{i+n} = x_{n-2} \\ & \alpha_{i+n+1} = 0 \\ & \alpha_{i+n+2} = x_{n-3} \\ & \vdots \end{aligned}$$

ただし， $i \leq s \leq i+n+2$  を満たす  $s$  は取りはじめの位置にはならない．

(1.2)  $n = 2$  のとき

$$\begin{aligned} & \vdots \\ & \alpha_i = x_0 \\ & \alpha_{i+1} = 0 \\ & \alpha_{i+2} = 2 \\ & \alpha_{i+3} = 0 \\ & \alpha_{i+4} = x_1 \\ & \vdots \end{aligned}$$

ただし， $i \leq s \leq i+4$  を満たす  $s$  は取りはじめの位置にはならない．

## (2) 停止パターン 2 (PT2)

 $n \geq 2$  のとき

$$\begin{aligned} & \vdots \\ \alpha_i &= n \\ \alpha_{i+1} &= 0 \\ \alpha_{i+2} &= x_1 \\ & \vdots \\ \alpha_{i+n} &= x_{n-1} \\ \alpha_{i+n+1} &= 0 \\ \alpha_{i+n+2} &= x_n \\ & \vdots \end{aligned}$$

ただし、 $i \leq s \leq i+n+2$  を満たす  $s$  は取りはじめの位置にはならない。

証明)

## (1) 停止パターン 1 の場合

ロールバックを開始すると以下のように盤面は変化する。ロールバックの開始盤面

$$(\dots, x_0, 0, n, x_1, \dots, x_{n-2}, 0, x_{n-3} \dots)$$

から有限回の遷移の後、 $\alpha_{i+n+1}$  の穴 (石の数は 0 個) に石を  $x$  個 ( $x \geq 1$ ) 置く。

$$(\dots, x_0, 0, n, x_1, \dots, x_{n-2}, x, x_{n-3} - 1 \dots)$$

ロールバックを続行し、 $\alpha_{i+1}$  の穴 (石の数は 0 個) に石を  $n$  個置く。

$$(\dots, x_0, n, n-1, x_1-1, \dots, x_{n-2}-1, x-1, x_{n-3}-1, \dots)$$

さらにロールバックを続行すると、盤上に石が  $n-1$  個入った穴が並ぶ。

$$(\dots, x_0-1, \underline{n-1}, \underline{n-1}, x_1-1, \dots, x_{n-2}-1, x-1, x_{n-3}-1, \dots)$$

## (2) 停止パターン 2 の場合

ロールバックを開始すると以下のように盤面は変化する。ロールバックの開始盤面

$$(\dots, n, 0, x_1, \dots, x_{n-1}, 0, x_n, \dots)$$

から有限回の遷移の後、 $\alpha_{i+n+1}$  の穴に石を  $x$  個 ( $x \geq 1$ ) 置く。

$$(\dots, n, 0, x_1, \dots, x_{n-1}, x, x_n - 1 \dots)$$

ロールバックを続行すると、 $\alpha_{i+1}$  の穴に石を  $n$  個置く。このとき、盤上に石が  $n$  個入った穴が並ぶ。

$$(\dots, \underline{n}, \underline{n}, x_1-1, \dots, x_{n-1}-1, x-1, x_n-1 \dots)$$

したがって (PT1), (PT2) からロールバックを行うと、有限回の遷移で  $\hat{R}$  に至る。 ■

以上により、ロールバックの停止条件として次の命

題が成り立つ。

## 命題 4

任意の面  $S$  に対して、与えられた位置からロールバックを開始したとき、有限回の遷移で (PT1) または (PT2) に至るならば、 $S$  からのロールバックは有限回の遷移で  $\hat{T}$  に至る。

また、ロールバックがループになることと順方向で動かしたときのループになることは同一なので、以下の命題が成り立つ。

## 命題 5

任意の盤面  $S$  に対して、与えられた位置からロールバックを開始したとき、有限回の遷移で (PT1) または (PT2) に至るならば、 $S$  から順方向で移動するとその 1 手は必ず停止する。

さらに、ロールバックを行うことなく、盤面そのものから 1 手の停止条件が判定できる場合もある。

## 命題 6

任意の盤面  $S$  に対して、それが (PT1), (PT2) または  $\hat{R}$  のいずれかを満たすならば  $S$  から順方向で移動するとき、その 1 手は必ず停止する。

## 6.3 一般化による解析結果との関係

前章では実験結果を一般化することによって 1 手が終了しないための条件 (LOOP1) および (LOOP2) を求めた。この 2 つの条件を合わせて (LOOP) とおく。また、命題 6 で示した条件を (TERM) とおくと、(LOOP) と (TERM) は矛盾しないことを示す。

## 命題 7

任意の盤面  $S$  に対して、 $S$  が (LOOP) と (TERM) を同時に満たすことはない。

証明)

$S$  が (LOOP) と  $\hat{R}$  を同時に満たさないのは明らか。 $S$  が (LOOP2) と (PT1) を同時に満たすとすると、盤面を  $S = (\alpha_1, \dots, \alpha_{2n'})$  とする。(PT1) を満たすことから命題 3 の証明で述べたように、 $S$  からロールバックを行うと有限回の遷移で

$$S' = (\dots, x_0, 0, n, x_1, \dots, x_{n-2}, x, x_{n-3}-1, \dots)$$

に至るが、このとき  $(n, x_1, \dots, x_{n-2})$  部分については  $S$  と  $S'$  で変化はない。したがって、この部分について調べれば十分である。(LOOP2) を満たすことから、1 番目の穴がロールバックの開始位置になるように  $S'$  を書き直す。

$$S'' = (x, x_{n-3}-1, \dots, 0, n, x_1, \dots, x_{n-2})$$

$S''$  における  $n$  の位置を考えると  $\alpha_{2n'-(n-2)} = n$  が

成り立つ．ここで， $n$  が奇数ならば  $2n' - (n-2)$  は奇数になるが，(LOOP2) から  $\alpha_{2n'-(n-2)}$  は偶数になるので矛盾する．同様に， $n$  が偶数ならば  $2n' - (n-2)$  は偶数になり，(LOOP2) から  $\alpha_{2n'-(n-2)}$  は奇数になるので．矛盾する．したがって (LOOP2) と (PT1) を同時に満たすことはない．(LOOP2) と (PT2) についても同様の議論ができる．

$S$  が (LOOP1) と (PT1) を同時に満たすとする．(LOOP2) の場合と同様に

$$S'' = (x, x_{n-3} - 1, \dots, 0, n, x_1, \dots, x_{n-2})$$

における  $(n, x_1, \dots, x_{n-2})$  部分について調べれば十分である． $S$  は (LOOP1) を満たすことから，

$$S = (n', \dots, 1, 0, n' - 1, n' - 2, \dots, 2, 1)$$

となる． $S''$  で最も右側に現れている 0 の位置から右端までには  $n-1$  個の穴があり， $S$  で最も右側に現れている 0 の位置から右端までには  $n'-1$  個の穴がある．したがってこれらが一致するためには  $n = n'$  でなければならない．一方，0 の位置の右隣を比べると  $n = n' - 1$  でなければならないので矛盾する．したがって， $S$  が (LOOP1) と (PT1) を同時に満たすことはない．(LOOP1) と (PT2) を同時に満たすとすると， $n = 1$  となるため明らかに矛盾する． ■

以上から，次の命題が成り立つ．

#### 命題 8

任意の盤面と着手に対して，(LOOP) を満たせばそこから始まる 1 手はループし，(TERM) を満たせば停止する．どちらの条件も満たさない場合は有限回の遷移によっていずれかの条件を満たす盤面が得られる．

## 7. 議 論

Mancala 一般に関する研究は主にゲームプログラミングの分野で行われているが，それらの主題は戦略や序盤戦や終盤戦における表の利用についてであり<sup>13),14)</sup>，動きに関する理論的解析を行ったものはほとんどない．Mancala の 1 つ Awari で終盤戦のデータベースを作るため，逆向きに解析したもの<sup>14)</sup> や，より複雑な Zanzibar BAO に関して，そのゲームの長さや平均分岐因子などを調べたもの<sup>7)</sup> があるが，動きそのものの解析は行われていない．本研究では特に周期性に焦点をあてて BAO の性質の理論的解析を行った．

本論文で述べたプロセス通信に基づくモデルを他のボードゲームに応用することは可能である．たとえば，オセロでは各セルを 1 つのプロセスとし，メッシュ型につながったネットワーク上でのメッセージ伝達を考

えれば，やはりバケツリレー型にメッセージが流れていくようなモデルができる．しかし，両者の根本的な違いは，オセロでは 1 つのセルがメッセージを受け取るのは 1 手に 1 回のみなのに対して，BAO では複数回あり自分の出したメッセージが自分自身に影響を及ぼす場合があるという点である．

## 8. ま と め

本論文では，プロセス通信に基づいて BAO のモデル化を行い，プロセス代数 CCS を使って動きを記述した．そして，ある開始状態と着手が与えられたとき，その手が停止するかどうか判定可能であることを示した．

プロセス通信に基づくモデル化は，以下のようなメリットを持つ．

- 局所的な変化の特性や隣接位置での相互作用が記述可能
  - 穴の個数に関する一般化が可能
- また，動作を解析し 1 手が停止するために盤面の満たす条件について議論した．
- 1 手が停止せず周期的に振る舞う局面をすべて抽出し，それらを 2 通りに分類しそれぞれの満たす条件を求めた．
  - 巻き戻しを使った解析により 1 手が停止するための条件を求めた．
  - これらの条件は矛盾しないことを示した．

BAO は開始局面におけるわずかな差異でも動作結果はまったく異なるというカオスの性質を持つ．そのため人間にとって先読みのしにくい興味深いゲームとなっている．

今回は盤の大きさを  $4 \times 2$  に制限して停止条件を示したが， $8 \times 2$  のときの停止条件としては不十分であり，盤面のサイズによって条件がどのように変化するかも検討したい．

## 参 考 文 献

- 1) Clarke, Jr., E.M., Grumberg, O. and Peled, D.A.: *Model Checking*, MIT Press (1999).
- 2) Holzmann, G.J.: *The SPIN Model Checker Primer and Reference Manual*, Addison Wesley (2004).
- 3) 稲畑康博, 高橋和子: ボードゲーム BAO の対戦戦略の評価と解析, 情報科学技術フォーラム (FIT), pp.271-273 (2003).
- 4) 稲畑康博, 高橋和子: ボードゲーム BAO の CCS による記述と解析, 第 48 回プログラミング研究会資料 (PRO-2003-5) (2004).

- 5) 稲畑康博, 高橋和子: ボードゲーム BAO における周期性の解析とモデルチェッカによる検証, 第 9 回ゲームプログラミングワークショップ 2004, pp.151-158 (2004).
- 6) <http://ist.ksc.kwansei.ac.jp/~ktaka/LABO/BAO/>
- 7) Donker, J. and Uiterwijk, J.: Programming Bao, *7th Computer Olympiad: Computer Games Workshop Proceedings*, Uiterwijk, J. (Ed.) (2002).
- 8) Russ, L.: *The Complete Mancala Games Book*, Marlow & Company (1995).
- 9) 松原 仁, 竹内郁雄 (編): *ゲームプログラミング*, 共立出版 (1998).
- 10) 松原 仁: 最近のゲームプログラミング研究の動向, *人工知能学会誌*, Vol.10, No.6, pp.835-845 (1995).
- 11) Huth, M.R.A. and Ryan, M.D.: *Logic in Computer Science*, Cambridge University Press (2000).
- 12) Milner, R.: *Communication and Concurrency*, Prentice Hall (1989).
- 13) Lincke, T.R. and Marzetta, A.: Large Endgame Database with Limited Memory Space, *ICGA Journal*, Vol.23, No.3, pp.131-138 (2000).
- 14) van der Goot, R.: Awari Retrograde Analysis, *Proc. CG2000*, pp.87-95 (2001).
- 15) van den Herik, H.J. and Iida, H. (Eds.):

*Games in AI Research*, Universiteit Maastricht (2000).

(平成 17 年 2 月 10 日受付)

(平成 17 年 3 月 23 日再受付)

(平成 17 年 4 月 20 日採録)



稲畑 康博

2005 年関西学院大学大学院理学研究科修了。同年株式会社 DNP 情報システム入社。知識表現とゲームプログラミングに興味を持つ。



高橋 和子 (正会員)

1982 年京都大学理学部卒業。同年三菱電機株式会社入社。同社中央研究所, 先端技術総合研究所, 産業システム研究所を経て 2000 年から関西学院大学理学部助教授。現在同大学理工学部情報科学科助教授。博士 (工学)。1997 年から 1999 年 ATR 音声翻訳通信研究所研究員。知識表現と推論機構, 特に時空間の扱いについて興味を持つ。電子情報通信学会, 日本ソフトウェア科学会, 人工知能学会各会員。