

# アヤトリ図形のトポロジカルな性質と結び目多項式による特性化

山 田 雅 之<sup>†</sup> Rahmat Budiarto<sup>†</sup>

世 木 博 久<sup>†</sup> 伊 藤 英 則<sup>†</sup>

本論文では点を含む紐図形のトポロジカルな性質について述べる。点を含む図形の表示方法としてネットハンドル表示を定義し、この表示の任意の変形が制限のあるライデマイスター移動と、新たにここで定義する3種類の変形の組合せで表現できることを示す。この変形をアヤトリ図形の処理に利用し、その効果を示す。また、紐図形の位相不変量(特性量)となる多項式を定義し、これを用いてアヤトリの紐状態の特性量を抽出する。さらに、ある多項式変換がアヤトリ動作に対応することを示す。

## Topology of Cat's Cradle Diagrams and Its Characterization using Knot Polynomials

MASASHI YAMADA,<sup>†</sup> RAHMAT BUDIARTO,<sup>†</sup> HIROHISA SEKI<sup>†</sup>  
and HIDENORI ITOH<sup>†</sup>

In this paper, we describe topological characteristics of string diagrams including some dots. We represent the string diagrams by *Net-Handle* representation and show that any deformation of Net-Handle diagrams is equivalent to a combination of restricted Reidemeister moves and new three deformations which are defined in this paper. We apply these deformations to a cat's cradle diagram processing and show their usefulness. Furthermore, we describe a characterizing method of string states of cat's cradle. This method uses a knot polynomial. Then, it is shown that a polynomial transformation can be corresponded to a cat's cradle action.

### 1. はじめに

身の周りには紐状の物体が多く存在し、それらの形状や振舞いを計算機上で表現処理することはシミュレーションやグラフィックス、アニメーションの分野でしばしば必要となる。本研究は紐状物体を表現する図形(紐図形)の位相的な性質を解明し、計算機上での紐状物体の表現や処理にその性質を利用することを目的とする。特に、本論文では、時間的に位相構造が変化する対象の例としてアヤトリの紐図形を取り上げ、その性質について述べる。アヤトリには、紐の絡みや紐の移動の制約など、紐状物体特有の現象が多く存在することから、アヤトリの紐図形の性質を明らかにすることは、他の紐状物体を取り扱う場合にも有益と考えられる。

これまでに我々はアヤトリにおける紐状態を点を含む図形で表現し、この表現を使ってアヤトリ過程をシ

ミュレーションする方法について報告した<sup>1)~3)</sup>。そこでは絡みや結びなどの紐の位相構造を保存しつつ紐図形を処理する方法を示した。その方法はライデマイスター移動<sup>4)</sup>による変形を利用した。我々の取り扱う紐図形は複数の点を含んでおり、これらの点は3次元空間における直線を表し、この直線に絡まる紐は曲線で表す(図1)。このような紐図形の性質はこれまでまだ詳細に調べられていない。

本論文では、点を含む紐図形の性質と特性化方法について述べる。以下では、点を含む紐図形の表示方法としてネットハンドル表示を定義し、任意の紐図形がネットハンドル表示できることを示す。また、ネットハンドル表示の任意の変形が、制限のあるライデマイスター移動と、ここで新たに定義する3種類の変形の組合せで表現できることを示す。さらに、これらの変形をアヤトリ図形処理に利用し、いくつかのアヤトリ過程シミュレーションに有効であることを示す。また、紐図形の位相不変量(特性量)となる多項式を定義し、これを用いてアヤトリの紐状態の特性量を抽出する。さらに、ある多項式変換がアヤトリ動作に対応するこ

<sup>†</sup> 名古屋工業大学工学部知能情報システム学科

Department of AI and Computer Science, Nagoya Institute of Technology

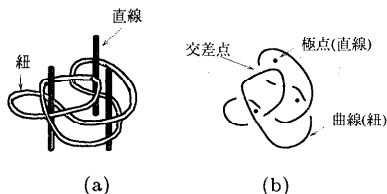


図1 (a) 3次元空間内の線分に絡んだ閉じた紐, (b) 紐図形  
Fig. 1 (a) Closed string tangled with lines in 3-dimensional space, (b) A string diagram.

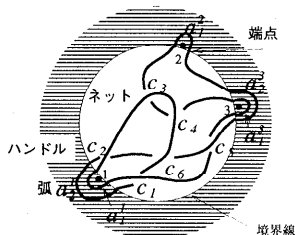


図2 ネットハンドル表示  
Fig. 2 Net-Handle diagram.

とを示す。

## 2. 等価変形

ここでは、紐図形の表示方法としてネットハンドル表示を定義し、その変形に関する性質について述べる。

[定義 1](紐図形) 図1(a)のような3次元空間内の閉じた紐の状態は(b)のような平面図形で表現できる。この平面図形では、3次元空間内で紐が絡んでいる直線は点で表す。一般の点と区別するため、この点を極点と呼ぶことにする。また、極点はその番号  $i$  により表記する。紐が重なった点、すなわち、二重点では下に位置する紐を区切れた曲線で表し、この点を交差点と呼ぶ。以降、このような紐の状態を表す図形を紐図形と呼ぶ。また、紐図形における極点の個数および紐の長さは有限と仮定する。

[定義 2](ネットハンドル表示) 紐図形の1つの表示方法としてネットハンドル表示を定義する。ネットハンドル表示は2つの領域からなる。一方は極点  $i$  とそれを囲む複数の弧  $a_j^i$  を含み、交差点を含まない領域であり、もう一方は交差点  $c_k$  を含む領域である。前者をハンドル領域、後者をネット領域と呼ぶ(図2)。ハンドル領域とネット領域の境界は閉曲線となる。これを境界線と呼ぶ。紐を表す曲線と境界線との交わる点を端点と呼ぶ。

[補題 1] 紐図形はネットハンドル表示できる。

(証明) 紐図形  $L$  に対し  $L$  を囲む閉曲線  $v$  がとれる(図3)。交差点の上を通らないように、かつ、互

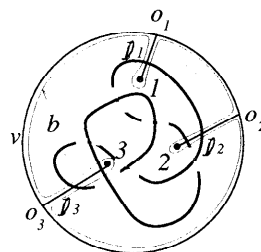


図3 紐図形に対する境界線  $b$  の付加  
Fig. 3 Adding a border line  $b$  to the string diagram.

いに交わらないように、 $v$  上のある点  $o_i$  から極点  $i$  まで線  $l_i$  をひける ( $i$  は極点の番号を表す)。  $v$  の内側に、  $l_i$  と交わらず、かつ、極点  $i$  を外側に持ち、交差点を内側に持つ閉曲線  $b$  がとれる。これを境界線とすることによりネットハンドル表示が得られる。任意の紐図形に対しこのような閉曲線  $b$  がとれることから紐図形はネットハンドル表示できる。 □

1つの紐図形に対し複数のネットハンドル表示が存在する。これは  $v$  上の極点  $o_1, \dots, o_n$  の互いの位置関係、および、  $o_i$  から極点  $i$  までの線  $l_i$  のとり方が1通りではないことによる。  $v$  上の点  $o_1, \dots, o_n$  の位置関係はネットハンドル表示における極点の位置関係に対応する。ネットハンドル表示の境界線を時計回りにたどったときの極点の円順序を単に極点の順序と呼ぶ。図4の  $nh_1(L_1)$  と  $nh_2(L_1)$  は紐図形  $L_1$  のネットハンドル表示である。これらは極点の順序が異なり、前者は2, 1, 3, 後者は1, 2, 3の順である。  $v$  上の点  $o_1, \dots, o_n$  の位置関係が等しい場合でも、  $o_i$  から極点  $i$  までの線  $l_i$  のとり方が異なれば得られるネットハンドル表示も異なることもある。図4の  $nh_2(L_1)$  と  $nh_3(L_1)$  は  $L_1$  のネットハンドル表示である。これらは極点の順序は等しいが  $l_2$  のとり方が異なる。

このように同一の紐図形の異なるネットハンドル表示については以下の補題2と補題3が成り立つ。

[定義 3](同形な紐図形, 同形なネットハンドル表示) 2つの紐図形  $L$  と  $L'$  が以下 a), b), c) を満たすとき、  $L$  と  $L'$  は同形という。

- $L$  と  $L'$  について、交差点を頂点、曲線を辺としたときできる各々の平面的グラフが同形<sup>5)</sup>である。
- 対応する交差点における紐の上下関係が等しい。
- 対応する領域に同一番号の極点が存在する。

2つのネットハンドル表示  $nh(L)$  と  $nh(L')$  が次の a') と上の b), c) を満たすとき、  $nh(L)$  と  $nh(L')$  は同形という。

- $nh(L)$  と  $nh(L')$  について、交差点と端点を頂点とし、境界線と曲線を辺としたときできる各々の

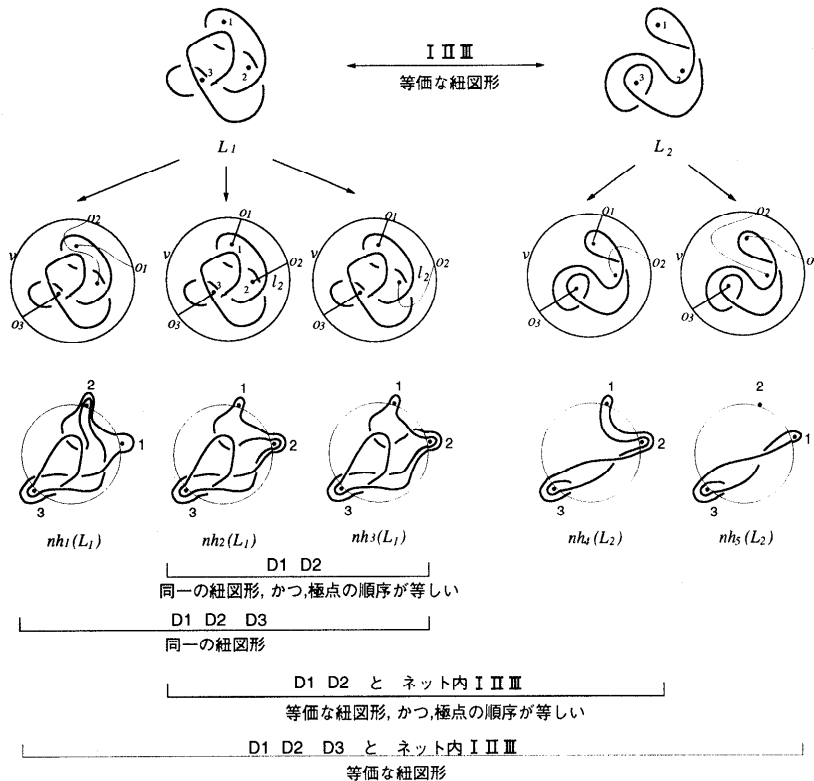


図 4 ネットハンドル表示の等価関係

Fig. 4 Equivalent relation among Net-Handle diagrams.

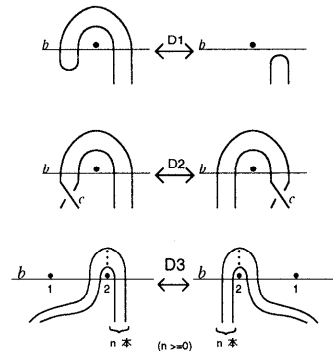
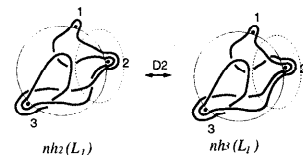
平面的グラフが同形である。

また、紐図形  $L$  を紐図形  $L'$  と同形な紐図形に変形できるとき、 $L$  は  $L'$  に移るといふ。同様に、ネットハンドル表示  $nh(L)$  をネットハンドル表示  $nh(L')$  と同形なネットハンドル表示に変形できるとき、 $nh(L)$  は  $nh(L')$  に移るといふ。

[定義 4] (ネットハンドル表示の変形) ネットハンドル表示の変形 **D1**, **D2**, **D3** を図 5 に定義する (図中  $b$  は境界線を表す)。なお、左から右への変形と右から左への変形を区別するときは、それぞれ **D1r**, **D2r**, **D3r** と **D1l**, **D2l**, **D3l** で表す。

[補題 2]  $nh(L)$  と  $nh(L')$  を紐図形  $L$  の異なる 2 つのネットハンドル表示とする。  $nh(L)$  と  $nh(L')$  の極点の順序が等しいならば、 $nh(L)$  と  $nh(L')$  は変形 **D1** と **D2** の繰返しで一方から他方へ移る (証明は付録参照)。

[例 1] 図 4 の  $nh_2(L_1)$  と  $nh_3(L_1)$  は紐図形  $L_1$  の異なる 2 つのネットハンドル表示で、かつ、極点の順序が同一である。  $nh_2(L_1)$  と  $nh_3(L_1)$  は **D2** で移行合う☆ (図 6)。

図 5 ネットハンドル表示の変形 **D1**, **D2**, **D3**Fig. 5 Net-Handle diagram deformations **D1**, **D2** and **D3**.図 6 破線で囲まれた部分に **D2** を施して  $nh_2(L_1)$  と  $nh_3(L_1)$  は移行合うFig. 6 **D2** deforming  $nh_2(L_1)$  and  $nh_3(L_1)$ .

☆ 簡単のため **D1** の必要ない例を用いた。

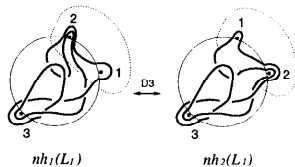
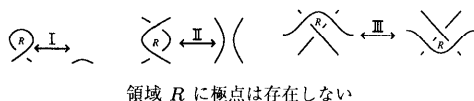


図7 破線で囲まれた部分に  $D3$  を施して  $nh_1(L_1)$  と  $nh_2(L_1)$  は移り合う

Fig. 7  $D3$  deforming  $nh_1(L_1)$  and  $nh_2(L_1)$ .



There is no dot in  $R$ .

図8 ライデマイスター移動 I, II, III  
Fig. 8 Reidemeister moves I, II, III.

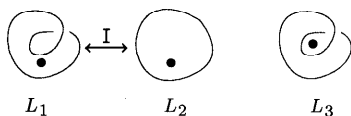


図9 等価な紐図形  $L_1$  と  $L_2$

Fig. 9 Equivalent string diagrams  $L_1$  and  $L_2$ .

[補題 3] 紐図形  $L$  の異なる 2 つのネットハンドルの表示  $nh(L)$  と  $nh(L')$  は変形  $D1$ ,  $D2$  と  $D3$  の繰返しで一方から他方へ移る (証明は付録参照)。

[例 2] 図 4 の  $nh_1(L_1)$  と  $nh_2(L_1)$  は紐図形  $L_1$  の異なる 2 つのネットハンドルの表示で、極点の順序が異なる。  $nh_1(L_1)$  と  $nh_2(L_1)$  は  $D3$  で移り合う (図 7)。また、例 1 より図 4 の  $nh_1(L_1)$  と  $nh_3(L_1)$  は  $D2$  と  $D3$  で移り合う。

[定義 5] (等価な紐図形) 2 つの紐図形  $L$  と  $L'$  が図 8 に示す 3 つのライデマイスター移動 I, II, III を繰返し適用して一方から他方へ移るとき、 $L$  と  $L'$  は等価という。なお、各ライデマイスター移動において領域  $R$  には極点は存在しないとする。また、左から右への変形と右から左への変形を区別するときは、それぞれ  $I_r$ ,  $II_r$ ,  $III_r$  と  $I_l$ ,  $II_l$ ,  $III_l$  で表す。

[例 3] 図 9 の紐図形  $L_1$  と  $L_2$  はライデマイスター移動 I で一方から他方へ移るので等価である。一方、紐図形  $L_3$  はライデマイスター移動の繰返しで  $L_1$  または  $L_2$  へ移ることができない。  $L_3$  は  $L_1$  および  $L_2$  と等価でない。

2 つの等価な紐図形の各々のネットハンドルの表示の間には、以下の補題 4、補題 5 と定理 1 が成り立つ。

[補題 4] 紐図形  $L$  と  $L'$  のネットハンドルの表示をそれぞれ  $nh(L)$  と  $nh(L')$  とする。もし  $L$  と  $L'$  が等価であり、かつ、 $nh(L)$  と  $nh(L')$  の極点の順序が

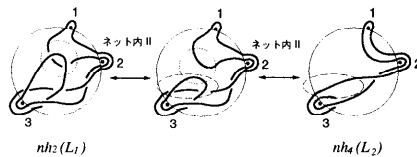


図10 破線で囲まれた部分にネット内 II を施して  $nh_2(L_1)$  と  $nh_4(L_2)$  は移り合う

Fig. 10 II deforming  $nh_2(L_1)$  and  $nh_4(L_2)$  in Net area.

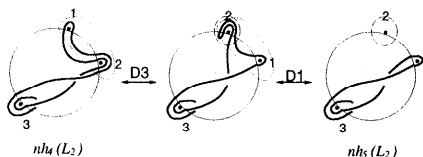


図11 破線で囲まれた部分に  $D1$  と  $D3$  を施して  $nh_4(L_2)$  と  $nh_5(L_2)$  は移り合う

Fig. 11  $D1$  and  $D3$  deforming  $nh_4(L_2)$  and  $nh_5(L_2)$ .

等しいならば、 $nh(L)$  と  $nh(L')$  はネット領域内のライデマイスター移動 I, II, III と変形  $D1$ ,  $D2$  の繰返しで一方から他方へ移る (証明は付録参照)。

[例 4] 図 4 の  $L_1$  と  $L_2$  は等価な紐図形である。  $L_1$  のネットハンドルの表示  $nh_2(L_1)$  と  $L_2$  のネットハンドルの表示  $nh_4(L_2)$  は極点の順序が同一である。  $nh_2(L_1)$  と  $nh_4(L_2)$  はネット領域内での 2 回のライデマイスター移動 II (ネット内 II) で移り合う (図 10)。また、例 1 より図 4 の  $nh_3(L_1)$  と  $nh_4(L_2)$  は 2 回のネット内 II と  $D2$  で移り合う。

[定理 1] 紐図形  $L$  と  $L'$  のネットハンドルの表示をそれぞれ  $nh(L)$  と  $nh(L')$  とする。もし  $L$  と  $L'$  が等価ならば、 $nh(L)$  と  $nh(L')$  はネット内のライデマイスター移動 I, II, III と変形  $D1$ ,  $D2$ ,  $D3$  の繰返しで一方から他方へ移る。

(証明)  $nh(L')$  を極点の順序が  $nh(L)$  と等しい  $L'$  のネットハンドルの表示とする。補題 3 より、 $nh(L')$  と  $nh(L)$  は変形  $D1$ ,  $D2$ ,  $D3$  の繰返しで一方から他方へ移る。また、 $L$  と  $L'$  は等価であり、かつ、 $nh(L)$  と  $nh(L')$  の極点の順序が等しい。よって補題 4 より、 $nh(L)$  と  $nh(L')$  はネット領域内のライデマイスター移動 I, II, III および変形  $D1$ ,  $D2$  の繰返しで一方から他方へ移る。以上より、 $nh(L)$  と  $nh(L')$  はネット内のライデマイスター移動 I, II, III と変形  $D1$ ,  $D2$ ,  $D3$  の繰返しで一方から他方へ移る。 □

[例 5] 図 4 の  $nh_4(L_2)$  と  $nh_5(L_2)$  は紐図形  $L_2$  の異なる 2 つのネットハンドルの表示で、極点の順序が異なる。  $nh_4(L_2)$  と  $nh_5(L_2)$  は  $D1$  と  $D3$  で移り合う (図 11)。また、例 1, 例 2, 例 4 より図 4 の  $nh_1(L_1)$ ,  $nh_2(L_1)$ ,  $nh_3(L_1)$ ,  $nh_4(L_2)$ ,  $nh_5(L_2)$  は

**D1, D2, D3** およびネット内 I, II, III を適用して互いに移り合う。

### 3. 紐図形の多項式

結び目理論は位相構造の異なる結び目を多項式を使って分類する方法を見つけた。ここでは複数個の極点を含む紐図形の多項式を定義する。なお、以下に示す補題は文献 4), 6) 等で紹介される証明方法を用いれば、容易に証明できるので省略する。

Hoste 等は 1 つの極点を含む紐図形の多項式を示した<sup>7)</sup>。ここでは複数の極点を取り扱うため、以下のよう

$$d(L; A, h_\alpha) = (-A^3)^{-w(L)} \langle L \rangle. \quad (1)$$

式 (1) において  $L$  は紐図形を表し,  $A, h_\alpha$  は可換な変数とする。  $w(L)$  は,  $L$  の各交差点の符号の総和である。交差点の符号は, 紐に 1 つの向きを与え,  $\nearrow$  のパターンならば +1,  $\searrow$  ならば -1 である。また,  $\langle L \rangle$  は以下の公理 1~5 で得られる多項式である。

1.  $\langle \alpha \bigcirc \rangle = 1$ ,
2.  $\langle \alpha \bigcirc \beta \rangle = h_\beta, \beta \neq \phi$
3.  $\langle \nearrow \rangle = A \langle \rangle + A^{-1} \langle \searrow \rangle$ ,
4.  $\langle \alpha \bigcirc L \rangle = -(A^2 + A^{-2}) \langle \alpha L \rangle, L \neq \phi$ ,
5.  $\langle \alpha \bigcirc \beta L \rangle = -(A^2 + A^{-2}) h_\beta \langle \alpha \cup \beta L \rangle, L \neq \phi$ ,

ここで,  $\bigcirc$  は交差点のない閉曲線を表し,  $\bigcirc_\beta$  は極点の集合  $\beta$  を含む閉曲線を表す。  $\beta$  の要素が  $i, j, k$  のとき  $h_\beta$  を  $h_{i,j,k}$  と書く。  $\langle \nearrow \rangle, \langle \rangle, \langle \searrow \rangle$  は  $\langle \rangle$  内の部分のみ異なり, それ以外は同一の図形の多項式を表す。

[例 6]  $d(\bigcirc_1 \bigcirc_2)$  を計算する。式 (1) より,

$$d(\bigcirc_1 \bigcirc_2) = (-A^3)^{-w(\bigcirc_1 \bigcirc_2)} \langle \bigcirc_1 \bigcirc_2 \rangle$$

ここで,

$$w(\bigcirc_1 \bigcirc_2) = +1$$

$$\langle \bigcirc_1 \bigcirc_2 \rangle = A \langle \bigcirc_1 \bigcirc_2 \rangle + A^{-1} \langle \bigcirc_1 \bigcirc_2 \rangle$$

← 公理 3 を使用

$$= A(-(A^2 + A^{-2}))h_1 \langle \bigcirc_1 \bigcirc_2 \rangle$$

+  $A^{-1}h_{1,2}$  ← 公理 2,5 を使用

$$= A(-(A^2 + A^{-2}))h_1 h_2 + A^{-1}h_{1,2}$$

← 公理 2 を使用

よって,

$$d(\bigcirc_1 \bigcirc_2) = (-A^3)^{-1} \{ A(-(A^2 + A^{-2}))h_1 h_2$$

$$+ A^{-1}h_{1,2} \} \\ = (1 + A^{-4})h_1 h_2 - A^{-4}h_{1,2}.$$

多項式  $d(L)$  については以下が成り立つ。

[補題 5] 紐図形  $L$  と  $L'$  が等価ならば,  $d(L) = d(L')$  が成り立つ。

[例 7] 図 4 の  $L_1$  と  $L_2$  について,  $d(L_1) = d(L_2)$  が成り立つ。

### 4. アヤトリへの応用

ここでは 2 章で述べたネットハンドル表示の変形をアヤトリ表示処理に応用する。また, 3 章で述べた紐図形の多項式を用いてアヤトリの紐状態の特性量を抽出し, それがアヤトリ動作によりどのように変化するかについて述べる。

#### 4.1 アヤトリ表示システムの拡張

著者等は文献 1)~3) において, アヤトリ表示システムについて述べた。このシステムは, アヤトリの紐状態を紐図形で表現し, 図 12 に示す 3 種類の基本動作, ‘取る’, ‘外す’, ‘入れ換える’の変形により得られるアヤトリのパターンをネットハンドル表示する。

ここでは, 新たに変形 **D1r** をシステムの図形処理に採用し, その効果を実行例を通して示す。

[アヤトリ表示システム] アヤトリ表示システムはユーザからマウス入力された 3 つの基本動作に従いネットハンドル表示を変形し, 動作後のアヤトリのパターンを表示する。システムの処理概念を図 13 に示す。

まず, アヤトリの初期パターンがモニタに表示され

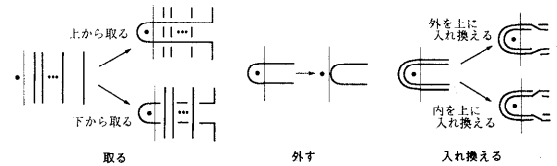


図 12 基本動作：取る，外す，入れ換える

Fig. 12 Basic moves: pull, release, exchange.

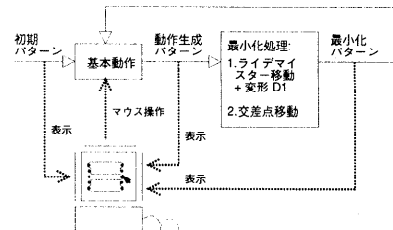
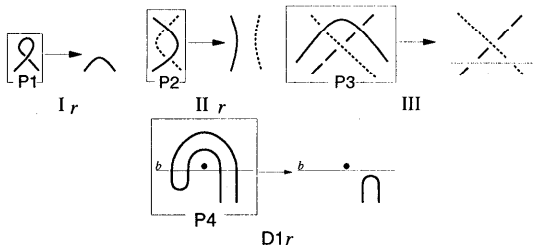


図 13 アヤトリ表示システムの処理概念

Fig. 13 The process scheme of the cat's cradle display system.

図 14 ライデマイスター移動と  $D1r$  による処理Fig. 14 Processing with Reidemeister moves and  $D1r$ .

る。ユーザは、3つの基本動作、‘取る’、‘外す’、‘入れ換える’、をマウス入力で指示する。システムはその動作を表現するパターン（動作生成パターン）を表示し、それに最小化処理を施す。最小化処理は、ネットハンドル表示の交差点数および紐の長さを（曲線の長さ総和）を最小化する。これにより生成したネットハンドル表示を動作後のアヤトリパターンとして表示する。ユーザはこの表示に対し再びアヤトリの動作を指示し、システムは同様の処理を施してアヤトリの動作後のパターンを表示する。以上の対話処理を繰り返し、ユーザは目的のアヤトリのパターンを作り上げる。

〔システムの拡張〕 実際のアヤトリの過程をより忠実に模倣するため、最小化処理に変形  $D1r$  を用いてシステムを拡張した。最小化処理は、紐の長さとの交差点数を最小化し、アヤトリの形を生成する。これまでの最小化処理は、交差点移動とネット内ライデマイスター移動を適用する。交差点移動は、交差点を紐の長さが小さくなる方向に移動する。またネット内  $I_r$ ,  $II_r$ ,  $III$  を用いて紐の長さとの交差点数が小さくなるよう変形する（図 14）。ここではさらに図 14 に示すパターン  $P4$  がネットハンドル表示に存在する場合に、そのパターンに  $D1r$  を施す処理を用いた<sup>☆</sup>。

ここでは拡張したシステムの評価のため、アヤトリを以下の3つのレベルに分類し、各々の例を示す。

レベル 1 従来システムにより変形過程を表現できるアヤトリ

レベル 2 レベル 1 以外で拡張システムにより変形過程を表現できるアヤトリ

レベル 3 拡張システムで変形過程を表現できないアヤトリ

なお、本システムは複数の輪を取り扱える。これを示すため以下では、いくつかあるアヤトリの中で、“輪落し”と呼ばれる種類の手品アヤトリを例にあげる。

図 15(a) と (b) に2つの“輪落し”を示す。両者を区別するため、図 15(a) を“輪落し A”，(b) を“輪落し B”と呼ぶ。

〔例 8〕 レベル 1 のアヤトリ：輪落し A

図 16(a) は“輪落し A”の変形過程をアヤトリ表示システムにより表現した様子である。図 16(a) の左側は右上のパターンに基本動作を施したとき（図中  $\rightarrow$  で表す）にできる動作生成パターンであり、右側のパターンは、左のパターンに最小化処理（図中  $\Rightarrow$  で表す）を施して得られるパターンである。

〔例 9〕 レベル 2 のアヤトリ：輪落し B

図 16(b) は“輪落し B”の変形過程を拡張システムにより表現した様子である。図 17 の (a) と (b) はそれぞれ図 16(b) の  $L_{14}$  と  $L_{15}$  の紐の状態を表す。 $L_{14}$  の指 4 にかかる紐を外すと  $L_{15}$  になる。 $D1r$  を用いない従来の最小化処理を  $L_{15}$  に施すと、 $II_r$  を 3 回繰り返し、(c) の状態で処理が終わる。 $D1r$  を用いた拡張システムの最小化処理は、(c) の指 3 の部分のパターン  $P4$  に対し  $D1r$  を施し、さらに  $II_r$ ,  $I_r$ ,  $II_r$ ,  $II_r$  の順に施し、図 16(b)  $L_{16}$  の状態で処理が終わる。

〔例 10〕 レベル 3 のアヤトリ：一段ばしご

図 18(a) は“一段ばしご”の出来上がり状態である。アヤトリには図 18(b) に示すような手首を 180 度回転する動作は多い。この動作では指の相対位置が変化する（図 18(c)）。指の相対位置を変える動作を表現するためには  $D3$  を含む基本動作を新たに用意する必要がある。しかしながら現時点では、3つの基本動作を複数回行い、紐パターンのみ一致させることとしている。

## 4.2 アヤトリの紐状態の多項式量

ここでは先の多項式  $d(L)$  を用いてアヤトリの紐状態の特性量を抽出する。また、アヤトリ動作による紐状態の変換が多項式にどのように影響するかについて述べる。

最小化処理は交差点を移動する処理とネット内  $I_r$ ,  $II_r$ ,  $III$  および  $D1r$  による変形処理のみを用いていることから、補題 5 より、

〔補題 6〕 紐図形  $L$  と、 $L$  に最小化処理を施して得られる紐図形  $L'$  について、 $d(L) = d(L')$  が成り立つ。

〔例 11〕 図 16(a) の  $L_1 \sim L_8$  について次が成り立つ。 $d(L_1) = d(L_2)$ ,  $d(L_3) = d(L_4)$ ,  $d(L_5) = d(L_6)$ ,  $d(L_7) = d(L_8)$ 。

アヤトリの紐状態が変わるとともにその多項式も変わる。アヤトリの動作がどのような多項式変換に対応するかは興味深い。極点が多項式の変数  $h_\alpha$  における

<sup>☆</sup>  $D2$  と  $D3$  は最小化処理に用いない。 $D2$  を用いないのは、 $D2$  を施すか否かの判定が困難なためである。 $D3$  を用いないのは、最小化の過程では極点の位置は固定とするためである。

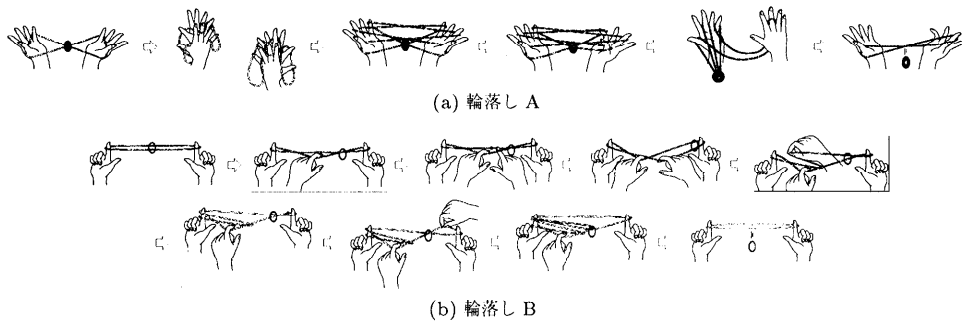


図 15 輪落し：2つの輪が絡んでいる状態からはじめ、紐を取ったり外したりという動作を数回繰り返して、絡みのない状態にして一方の輪を落とす（図は文献8）より

Fig. 15 Waotoshi.

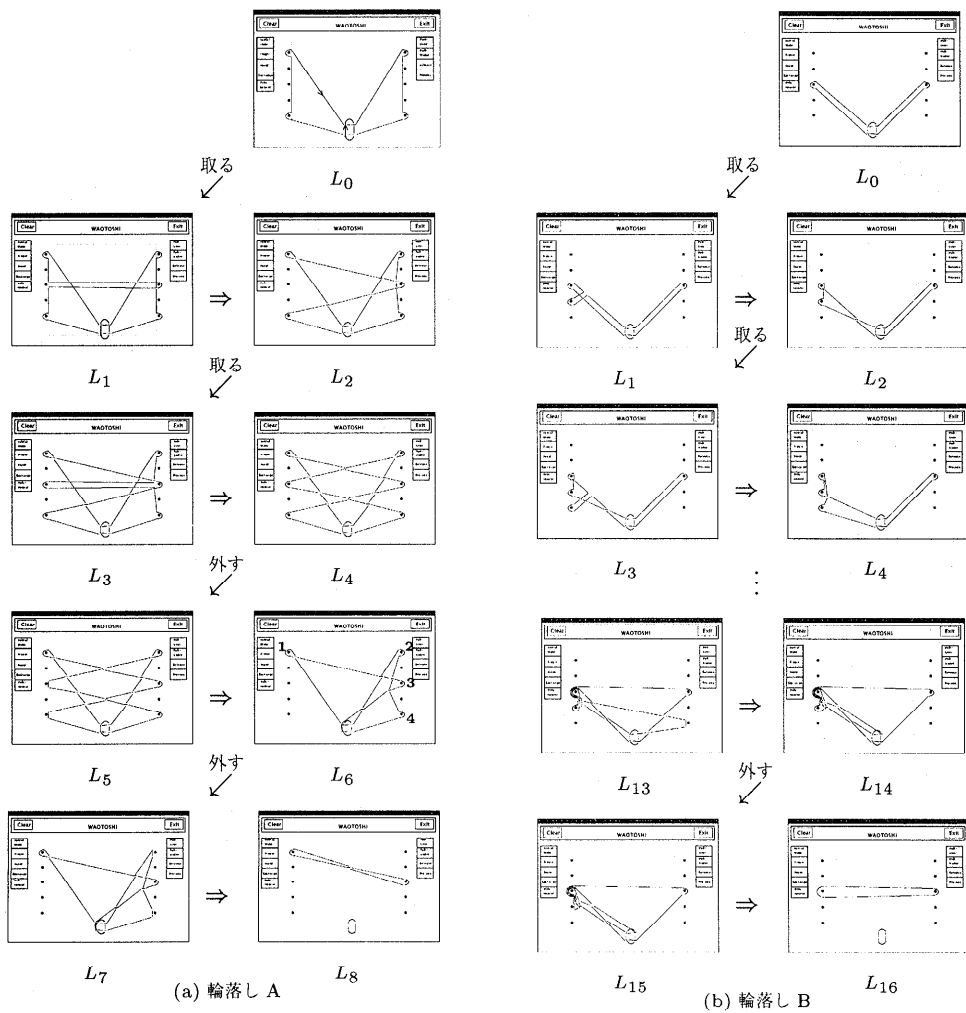


図 16 実行例：輪落し

Fig. 16 An execution example Waotoshi.

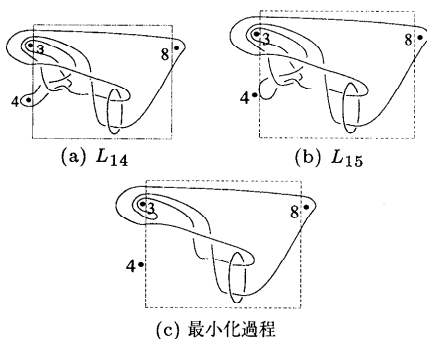


図 17 “輪落し B”  $L_{15}$  への最小化処理の過程  
Fig. 17 Process of Minimizing Processing for Waotoshi (B)  $L_{15}$ .

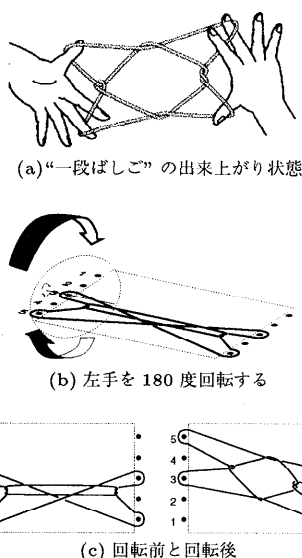


図 18 “一段ばしご” における左手の半転  
Fig. 18 Half rotation of left hand for Ichidan-basigo.

$\alpha$  の要素に対応することより，以下の補題が成り立つ．

[補題 7] あるアヤトリの紐状態  $s$  を表す紐図形を  $L$  とする．状態  $s$  から，指  $i$  に掛かっている紐を外して，新たに得られる紐状態を  $s'$  とし，それを表す紐図形を  $L'$  とする．このとき， $d(L')$  は次の操作を  $d(L)$  に施して得られる．

変数  $h_\alpha$  を  $h_{\alpha - (\alpha \cap \{i\})}$  へ置き換える．ただし  $\alpha - (\alpha \cap \{i\}) = \phi$  なら  $h_\alpha$  を 1 に置き換える．

[例 12]

$$d\left(\begin{array}{c} \text{図 12(a) の紐図形} \\ \text{指 1 に掛かっている紐を外して得られる紐図形} \end{array}\right) = -(A^6 + A^{-6})h_{1,2,3} + (A^6 - A^2 - A^{-2} + A^{-6})h_1h_{2,3} \cdots (*)$$

の指 1 に掛かっている紐を外して得られる紐図形  $\begin{array}{c} \text{図 12(a) の紐図形} \\ \text{指 2 に掛かっている紐を外して得られる紐図形} \end{array}$  の多項式は式 (\*) の  $h_{1,2,3}$  と  $h_1$  をそれぞれ  $h_{2,3}$  と 1 に置き換えて得られる．また補題 6 より， $\begin{array}{c} \text{図 12(a) の紐図形} \\ \text{指 2 に掛かっている紐を外して得られる紐図形} \end{array}$  の多項式は等しい．

$$\begin{aligned} d\left(\begin{array}{c} \text{図 12(a) の紐図形} \\ \text{指 2 に掛かっている紐を外して得られる紐図形} \end{array}\right) &= -(A^6 + A^{-6})h_{2,3} \\ &\quad + (A^6 - A^2 - A^{-2} + A^{-6})h_{2,3} \\ &= -(A^2 + A^{-2})h_{2,3} \\ &= d\left(\begin{array}{c} \text{図 12(a) の紐図形} \\ \text{指 2 に掛かっている紐を外して得られる紐図形} \end{array}\right) \end{aligned}$$

の指 2 に掛かっている紐を外して得られる紐図形  $\begin{array}{c} \text{図 12(a) の紐図形} \\ \text{指 2 に掛かっている紐を外して得られる紐図形} \end{array}$  の多項式は式 (\*) の  $h_{1,2,3}$  と  $h_{2,3}$  をそれぞれ  $h_{1,3}$  と  $h_3$  に置き換えて得られる．また，補題 6 より， $\begin{array}{c} \text{図 12(a) の紐図形} \\ \text{指 2 に掛かっている紐を外して得られる紐図形} \end{array}$  の多項式は等しい．

$$\begin{aligned} d\left(\begin{array}{c} \text{図 12(a) の紐図形} \\ \text{指 2 に掛かっている紐を外して得られる紐図形} \end{array}\right) &= -(A^6 + A^{-6})h_{1,3} \\ &\quad + (A^6 - A^2 - A^{-2} + A^{-6})h_1h_3 \\ &= d\left(\begin{array}{c} \text{図 12(a) の紐図形} \\ \text{指 2 に掛かっている紐を外して得られる紐図形} \end{array}\right) \end{aligned}$$

[例 13] 輪落し A：図 16(a) の  $L_6$  の多項式  $d(L_6)$  は以下となる．なお，紐の向きは図 16(a)  $L_0$  のようにはじめに与えられているものとする．指の番号は図 16(a)  $L_6$  のように与える．

$$\begin{aligned} d(L_6) &= (A^2 + A^{-10} - A^6 - 2A^{-6} - A^{-2})h_{1,2,3,4} \\ &\quad + (A^{10} + 2A^{-2} + A^2)h_{1,2}h_{3,4} \\ &\quad + (-A^2 + A^2 - A^{-10} + A^{-6})h_3h_{1,2,4} \\ &\quad + (-A^{10} - A^2 - 2A^{-6} - A^6 - 3A^{-2})h_3h_{1,2}h_4 \\ &\quad + (A^6 + 2A^{-6} + A^{-2})h_{1,2,3}h_4 \end{aligned}$$



$$\begin{aligned}
& + (-A^2 + A^{-2} - A^{-10} + A^{-6})h_2h_{1,3,4} \\
& + (A^2 + A^{-14} - 2A^{-6})h_2h_3h_{1,4} \\
& + (-A^{-2} + A^{-6} - A^{-14} + A^{-10})h_{1,4}h_{2,3} \\
& + (-2A^{-2} + A^6 + A^{-10})h_1h_2h_{3,4} \\
& + (A^{-2} - A^2 + A^{-6} - A^{-10})h_1h_{2,3,4} \\
& + (3A^{-6} + 4A^{-2} - 2A^6 - A^2 \\
& \quad - 2A^{-2} - A^{-14} - A^{-10})h_1h_2h_3h_4 \\
& + (-2A^{-6} + A^2 + A^{-14})h_1h_4h_{2,3} \\
& + (A^6 + A^{-10} - 2A^{-2})h_1h_3h_{2,4} \cdots (**)
\end{aligned}$$

$L_6$  の指 2 に掛かっている紐と指 4 に掛かっている紐をそれぞれ外して得られる図 16 (a)  $L_7$  の多項式  $d(L_7)$  は式 (\*\*) の変数  $h_\alpha$  を,  $\alpha$  から 2 と 4 を取り除いた変数に置き換えて得られる. たとえば  $h_{1,2,3,4}$  は  $h_{1,3}$ ,  $h_{1,2}$  は  $h_1$ ,  $h_{3,4}$  は  $h_3$  とする. また, 補題 6 より,  $d(L_7)$  は  $d(L_8)$  と等しい.

$$\begin{aligned}
d(L_7) &= (-A^2 - 2A^{-2} - A^{-6})h_1h_3 \\
&\quad + (A^{-2} + A^{-6})h_{1,3} \\
&= -(A^2 + A^{-2}) \\
&\quad \{(1 + A^{-4})h_1h_3 - A^{-4}h_{1,3}\} \\
&= d(L_8)
\end{aligned}$$

## 5. おわりに

本論文ではネットハンドル表示と変形 **D1**, **D2**, **D3** を定義した. そして, 紐図形のネットハンドル表示には任意性があることを述べ, 等価な 2 つの紐図形の各々のネットハンドル表示はネット領域内のライデマイスター移動 I, II, III と変形 **D1**, **D2**, **D3** により移り合うことを示した. これらをアヤトリ図形処理に適用し, いくつかのアヤトリ過程シミュレーションに有効であることを示した. また, 紐図形の多項式を定義し, アヤトリ動作と多項式変換との対応について述べた. システムにより得られる動作後のアヤトリパターンが位相的に正しいか否かの検証にこの多項式を利用できる.

今後は, 紐図形の性質とその特性化方法について, さらに検討する. また, ここでは指から紐を外す動作のみについて, 多項式がどのように変化するか述べたが, 今後は他の動作についても調査する予定である.

## 参 考 文 献

- 1) Budiart, R., 山田雅之, 伊藤英則, 世木博久: An Interactive System Implementation for Constructing Cat's Cradle Diagrams and Its Evaluation, 情報処理学会論文誌, Vol.37, No.2, pp.231-241 (1996).
- 2) 山田雅之, Budiart, R., 伊藤英則, 世木博久: アヤトリにおけるひも図形変形過程の表現とその処

理, 情報処理学会論文誌, Vol.35, No.3, pp.497-504 (1994).

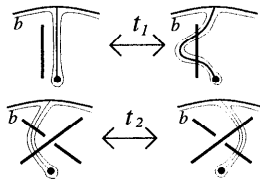
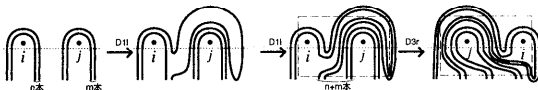
- 3) 山田雅之, 杉山 貴, 世木博久, 伊藤英則: 遺伝的アルゴリズムに基づく紐図形処理—アヤトリ図形生成方法, 情報処理学会論文誌, Vol.35, No.10, pp.2059-2068 (1994).
- 4) Kauffman, L.H.: *On Knots*, Annals of Mathematics Studies, Vol.155, Princeton Univ. Press (1987).
- 5) 日本数学会 (編): 数学辞典第 3 版, 岩波書店 (1987).
- 6) Kauffman, L.H.: State Models and the Jones Polynomial, *Topology*, Vol.26, pp.395-407 (1987).
- 7) Hoste, J. and Przytycki, J.: An Invariant of Dichromatic Links, *Proc. Am. Math. Soc.*, Vol.105, No.4, pp.1003-1007 (1989).
- 8) 池田書店 (編): たのしくあそぶあやとり全集, 池田書店 (1994).

## 付 録

補題 2, 3, 4 の証明のため, いくつか言葉の定義をする. 図 3 のように, 紐図形  $L$  を囲む閉曲線  $v$  と,  $v$  上のある点  $o_i$  から極点  $i$  まで線  $\ell_i$  をひいた図形を  $L$  の中間図形と呼ぶ. また, 線  $\ell_i$  を点引線と呼ぶ. 中間図形とネットハンドル表示は一対一対応することに注意する (図 4).

[補題 2 の証明]  $nh(L)$  と  $nh(L)'$  に対応する中間図形をそれぞれ  $N$  と  $N'$  とすると,  $N$  と  $N'$  は点引線が異なる.  $N$  における点引線  $\ell_i$  を  $N'$  の点引線  $\ell'_i$  と一致するように変形することを考える. 図 19 の変形  $t_1$  は  $\ell_i$  を紐に重ねる変形と  $\ell_i$  と紐の重なりをとる変形である.  $t_2$  は交差点を越えるように  $\ell_i$  を移動する変形である.  $v$  内での  $\ell_i$  の任意の変形は  $t_1$  と  $t_2$  の組合せで表現できる. すなわち,  $t_1$  と  $t_2$  を使って  $\ell_i$  を  $\ell'_i$  に変形できる. よって,  $t_1$  と  $t_2$  を用いて  $N$  を  $N'$  に変形できる. 図 19 のようにネットハンドル表示の境界線  $b$  をとることにより,  $t_1$  と  $t_2$  はそれぞれ **D1** と **D2** に対応する. このことから, 変形 **D1**, **D2** を用いて  $nh(L)$  を  $nh(L)'$  に変形できる. 同様に,  $nh(L)'$  を  $nh(L)$  へ変形することもできるので, 変形 **D1**, **D2** を用いて  $nh(L)$  と  $nh(L)'$  は一方から他方へ移る. □

[補題 3 の証明]  $nh(L)$  と  $nh(L)'$  の極点の順序が等しいならば, 補題 2 より, 補題 3 は成り立つ. また,  $nh(L)$  と  $nh(L)'$  の極点の順序が異なる場合, 一方を他方の極点の順序と等しくできれば, 補題 2 より, 補題 3 が成り立つ. ここでは変形 **D1**, **D3** により極点

図 19 中間図形における変形  $D1$ ,  $D2$ Fig. 19 Deformations  $D1$ ,  $D2$  on the neutral diagram.図 20 変形  $D1$  と  $D3$  により隣り合う 2 つの極点の順序を入れ換えるFig. 20 Two dots which are next to each other are exchanged their order by deformations  $D1$  and  $D3$ .

の順序を等しくできることを示す。

$i, j$  をネットハンドル表示の互いに隣り合う 2 つの極点で、かつ、 $j$  は  $i$  より時計回り側に位置しているとする (図 20)。極点  $i$  を  $n$  個の弧が囲むとき、それらの弧の時計回り方向の端点を通るネット内の  $n$  本の紐に対し、外側から順にそれらの紐が  $j$  を囲むように変形  $D1l$  を  $n$  回適用する。そうすると極点  $i$  と極点  $j$  を入れ換える変形  $D3r$  が適用できる (図 20 の破線で囲まれた部分に  $D3r$  が適用できる)。このことから、変形  $D1$ ,  $D3$  により任意の隣り合う 2 つの極点の順序を入れ換えられる。よって、変形  $D1$ ,  $D3$  により  $nh(L)$  の極点の順序を  $nh(L)'$  と等しくできる。

□

[補題 4 の証明]  $nh(L)$  と  $nh(L')$  に対応する中間図形をそれぞれ  $N$  と  $N'$  とする。 $L$  と  $L'$  が等価であり、かつ、 $N$  と  $N'$  の閉曲線  $v$  上の点  $o_i$  の順序が互いに等しいならば、 $N$  と  $N'$  は  $v$  内のライデマイスター移動 I, II, III により互いに移り合う。 $v$  内の点引線を含まない部分 (パターン) に対するライデマイスター移動と図 19 に示す 2 つの変形  $t_1, t_2$  の組合せで  $v$  内のライデマイスター移動と等価な変形が作れる。 $v$  内の点引線を含まない部分 (パターン) に対するライデマイスター移動は、ネットハンドル表示においては、ネット領域内ライデマイスター移動に対応し、 $t_1$  と  $t_2$  はそれぞれ  $D1$  と  $D2$  に対応する。以上より補題 4 が成り立つ。

□

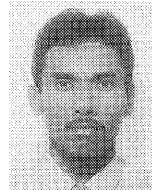
(平成 8 年 7 月 4 日受付)

(平成 9 年 6 月 3 日採録)



山田 雅之 (正会員)

1992 年名古屋工業大学工学部電気情報工学科卒業。1994 年同大学院工学研究科博士前期課程修了。同年同大学工学部知能情報システム学科助手。人工知能学会会員。



Rahmat Budiarto (学生会員)

1986 年 Bandung 工業大学数学科卒業。Bandung 工業専門学校講師, Nusantara Aircraft Industries Ltd. 勤務。現在名古屋工業大学大学院博士後期課程在学中。システム設計, 結び目理論に関する研究に従事。



世木 博久 (正会員)

1979 東京大学工学部計数工学科卒業。1981 年同大学院工学系研究科修士課程修了。同年 4 月より三菱電機 (株) 中央研究所に勤務。1985 ~ 1989 年 (財) 新世代コンピュータ技術開発機構に出向。1992 年 4 月より名古屋工業大学工学部知能情報システム学科助教授。工学博士。論理プログラミング, 演繹データベース等に興味を持つ。電子情報通信学会, 人工知能学会, ACM, IEEE Computer Society 各会員。



伊藤 英則 (正会員)

1974 年名古屋大学大学院工学研究科博士課程電気・電子専攻満了。工学博士号取得。同年日本電信電話公社入社, 横須賀研究所勤務。1985 年 (財) 新世代コンピュータ技術開発機構出向。1989 年より名古屋工業大学教授, 現在知能情報システム学科所属, 人工知能学会理事。これまでに, 数理言語理論とオートマトン, 計算機ネットワーク通信 OS, 知識ベースシステムなどの研究と開発に従事。電子情報通信学会, 人工知能学会, 形の科学学会, ファジィ学会各会員。