

## 折り紙過程の知識表現とその処理プログラムの作成†

内 田 忠\*\* 伊 藤 英 則\*\*

折り紙は、折り目と平面の構成によって様々な形を作り出すことができる。本論文では、折り紙の展開図を構成する幾何学的要素から出来上がり図を推論する一方法について述べる。まず、折り紙の変形過程を表現するタワーを提案し、その生成方法と性質について述べる。また、展開図を構成する幾何学的要素から、真理性維持システムを用いてタワー構築する方法を述べる。最後にタワーを構成するフロアの性質を利用してタワーから出来上がり図を決定するアルゴリズムを述べる。

## 1. はじめに

折り紙の歴史は古く、また毎年新しい作品が発表されている<sup>1)</sup>。折り紙は付録1に例示するように折り目と平面の構成によって様々な形を作り出すことができる。このような折り紙の変形過程を知識表現し、機械的に処理することは大変意味深い、これまで知識表現手法を与えその処理プログラムの作成についての研究は報告されていない。

本論文では、折り紙の展開図を構成する幾何学的要素から出来上がり図を推論する一方法について述べる<sup>2)</sup>。ただし、ここでは出来上がり図とは二次元平面の折り目による重ね合わせである制限を設ける。

以下、2章で折り紙の操作過程に現れる折り目と平面の関係の変化過程を表現するタワーを定義し、3章でその生成方法と性質について述べる。4章で展開図を構成する幾何学的要素から、タワーを構築する方法について述べ、5章でタワーから出来上がり図を決定するアルゴリズムを示す。

## 2. 諸定義と変形過程のタワー表現

まず、展開図を構成する幾何学的要素である折り目と平面素を以下に定義する。

【定義1】 折り目は平面と平面の境界である。折り目属性を  $(M-V, O-F)$  とする。ここに、 $M-V \in \{\text{山, 谷}\}$ 、 $O-F \in \{\text{開き, 折り}\}$  とし、'山' は *mountain*、'谷' は *valley*、'開き' は *open*、'折り' は *fold* と表すこともある。また、それ以上分割されることのない折り目を原子折り目とする。

† Knowledge Representation of Origami and Its Implementation by TADASHI UCHIDA and HIDENORI ITOH (Department of Intelligence and Computer Science, Nagoya Institute of Technology).

\*\* 名古屋工業大学知能情報システム学科

\* 現在 中部電力(株)

【定義2】 折り目または折り目と縁によって定まる二次元の  $n$  角形を平面素 (plane) とし、その属性を (裏表, 頂点1, ..., 頂点  $n$ ) とする。また、折り目によってそれ以上分割されることのない平面素を原子平面素とする。なお、平面素には一意に名前を付ける。以降は、平面素を単に平面と記すこともある。

つぎに、折り紙の変形過程に現れる折り目と平面の関係を表現するためにタワーを定義する。

【定義3】 節点 ( $v$ ) の集合を  $V$ 、枝 ( $e$ ) の集合を  $E$  とし、 $T=(V, E)$  を二分木とする。 $v^0, v^1, \dots, v^k \in V$  で  $0 \leq i \leq k$  に対して  $v^i$  を  $v^{i+1}$  の親節点  $v^{i+1}$  を  $v^i$  の子節点とし親節点と子節点間は枝  $e_i$  ( $e_i \in E$ ) で結ばれる。 $v^k$  を深さ  $k$  の節点とする。さらに、最も深い節点の深さが  $n$  である  $T$  の高さを  $n$  とし、 $T^n$  と記す。

【定義4】 平面素を節点  $v$  に対応させる。平面素  $p1$  が、ある折り目によって平面素  $p11$  と平面素  $p12$  に二分割される時、親節点  $v1$  は二つの子節点  $v11, v12$  を持つ。ここに、親節点  $v1$  は平面素  $p1$  に、二つの子節点  $v11, v12$  はそれぞれ平面素  $p11, p12$  に対応する。また、特別に或る折り目が平面素  $pi$  以外に作用し、 $pi$  には作用しなかったとき、親節点  $vi$  はその折り目の操作によって、自分自身を子節点として持つことにする。ここに、親節点  $vi$  と子節点  $vi$  は同一平面素  $pi$  に対応する。

【定義5】 二分木  $T=(V, E)$  の深さが  $k$  である節点の集合を  $V^k$  とする。 $V^k$  の任意の二つの節点  $v1, v2$  を接続している折り目を深さ  $k$  の弦  $r^k$  と呼び、 $r^k(v1, v2)$  と記し、また、 $r^k(v1, v2)$  の折り目属性を  $attrib(r^k(v1, v2))$  と記す。深さ  $k$  の弦の集合を  $R^k = \{r^k(v1, v2) \mid v1, v2 \in V^k, attrib(r^k(v1, v2)) = (M-V \text{ 属性}, O-F \text{ 属性})\}$  とする。とくに、 $V^k$  の任意の節点  $v$  を  $v^k$  と記すこともある。

【定義6】  $To^k = TU \left\{ \sum_{n=0}^k R^n \right\}$  を高さ  $k$  のタワーと

する。

【定義7】  $F^k=(V^k, R^k)$  を深さ  $k$  のフロアとする。

【定義8】 弦  $r^k(v_1, v_2)$  で、 $v_1, v_2$  が同一の親節点を持つとき、 $v_1, v_2$  が bro (兄弟) 関係であるとし、 $rel(r^k(v_1, v_2))=bro$  と記す。また、その集合を  $R_b^k=\{r^k(v_1, v_2) | v_1, v_2 \in V^k, rel(r^k(v_1, v_2))=bro\}$  とする。

【定義9】 弦  $r^k(v_1, v_2)$  で、 $v_1, v_2$  が異なる親節点を持つとき、 $v_1, v_2$  が cou (従兄) 関係であるとし、 $rel(r^k(v_1, v_2))=cou$  と記す。また、その集合を  $R_c^k=\{r^k(v_1, v_2) | v_1, v_2 \in V^k, rel(r^k(v_1, v_2))=cou\}$  とする。

定義8, 9 より、 $rel(r^k(v_1, v_2))$  は bro, cou のいずれかである。

なお、タワーの弦と展開図の折り目属性は図1の表記とする。

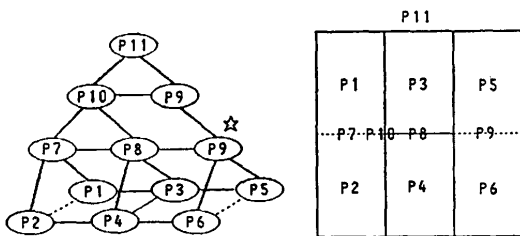
弦/折り目	M-V属性	O-F属性
.....	山	開き
.....	山	折り
————	谷	開き
————	谷	折り

図1 弦と折り目の属性  
Fig. 1 Attributes of chord and fold.

以上の定義より、 $k$  番目の動作による平面素  $V^k$  と  $To^{k-1}$  の  $V^{k-1}$  との親子関係をもつ節点を枝で結び、 $To^{k-1}$  に  $F^k$  を加えたものが  $To^k$  に対応する。また  $R^k$  は以下の性質が成り立つ。

1.  $R_b^k$ :  $k$  番目の動作で新しくできる折り目の集合。
2.  $R_c^k$ :  $k$  番目の動作以前に存在していた折り目またはそれが分割された折り目の集合。

これらを用いて高さ3のタワー  $To^3$  を図2に例示



$$V^0=\{P11\} V^1=\{P10, P9\} V^2=\{P7, P8, P6\} V^3=\{P1, P2, P3, P4, P5, P6\}$$

$$R_b^1=\{r^1(P10, P9)\} R_b^2=\{r^2(P7, P8)\} R_b^3=\{r^3(P1, P2), r^3(P3, P4), r^3(P5, P6)\}$$

$$R_c^1=\{r^1(P8, P9)\} R_c^2=\{r^2(P1, P3), r^2(P3, P5), r^2(P2, P4), r^2(P4, P6)\}$$

図2 高さ3のタワー  $To^3$  とその展開図  
Fig. 2 Tower of height three and corresponding crease pattern.

\* 定義4で述べた、自分自身を子節点としている。

する。

### 3. タワーとフロアの性質

動作に対応するタワーとフロアの性質について述べる。

#### 3.1 動作

ここでは折る動作を以下のように定義する<sup>1)</sup>。ここで対象とする折り紙はこれらの動作の繰り返しによるものとし、また、折り紙の変形過程はつねに二次元平面の折り目による重ね合わせとする。

基本動作: '折り畳む', '折り込む', '開く' の三種類である。とくに折り始めの '折り畳む' は原始動作と呼ぶ。

複合動作: 二つの基本動作が同時に行われる動作, '折り畳み込む' (= '折り畳む' と '折り込む' の複合動作), '折り畳み開く' (= '折り畳む' と '開く' の複合動作) の二種類とする。

#### 3.2 タワーの生成方法

1) 折り紙の初期平面を根 (root) 節点で表す (図3参照)。



図3 初期平面のフロアと展開図  
Fig. 3 Floor of a first stage plane and crease pattern.

2) 原始動作

原始動作\*\*によるフロア  $F^1$  を原始フロア (primitive-floor) とする (図4参照)。

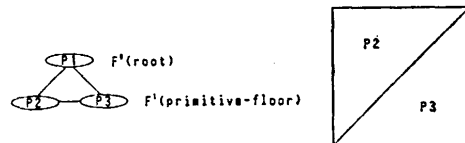


図4 原始フロアと展開図  
Fig. 4 A primitive-floor and crease pattern.

3) 基本動作

基本動作によるタワーとフロアを示す。

#### 3.1) 折り畳む

'折り畳む' によるフロア  $F^i$  ( $i \geq 2$ ) を折り畳みフロア (fold-floor) とする (図5参照)。

\*\* タワー構造は同一であるが、平面 P2, P3 の属性の異なるものが存在する。以下、他の動作においても同様である。

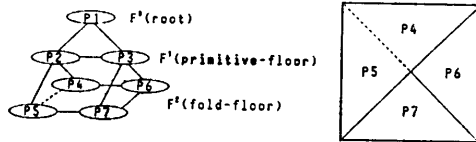


図 5 折り畳みフロアと展開図

Fig. 5 A fold-floor and crease pattern.

3.2) 折り込む

‘折り込む’によるフロア  $F^i$  ( $i \geq 2$ ) を折り込みフロア (tuck-floor) とする (図 6 参照).

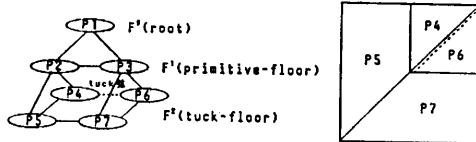


図 6 折り込みフロアと展開図

Fig. 6 A tuck-floor and crease pattern.

【定義 10】 折り込み弦 (tuck 弦): 弦  $r^k(v_1, v_2)$  の M-V 属性が、節点  $v_1, v_2$  のそれぞれの親節点間に存在する弦の M-V 属性と排反であるとき、弦  $r^k(v_1, v_2)$  を折り込み弦とする。

3.3) 開く

‘開く’によるフロア  $F^i$  ( $i \geq 2$ ) を開きフロア (open-floor) とする (図 7 参照).

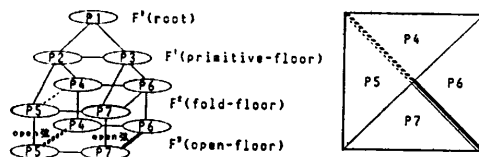


図 7 開きフロアと展開図

Fig. 7 An open-floor and crease pattern.

【定義 11】 開き弦 (open 弦): 弦  $r^k(v_1, v_2)$  の O-F 属性が ‘open’ で、節点  $v_1, v_2$  のそれぞれの親節点間に存在する弦の O-F 属性が ‘fold’ のとき、弦  $r^k(v_1, v_2)$  を開き弦とする。

4) 複合動作

4.1) 折り畳み込む

‘折り畳み込む’によるフロア  $F^i$  ( $i \geq 3$ ) を折り畳み込みフロア (fold-tuck-floor) とする (図 8 参照).

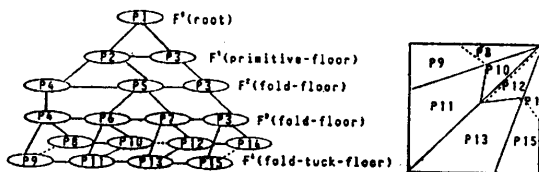


図 8 折り畳み込みフロアと展開図

Fig. 8 A fold-tuck-floor and crease pattern.

4.2) 折り畳み開く

‘折り畳み開く’によるフロア  $F^i$  ( $i \geq 3$ ) を折り畳み開きフロア (fold-open-floor) とする (図 9 参照).

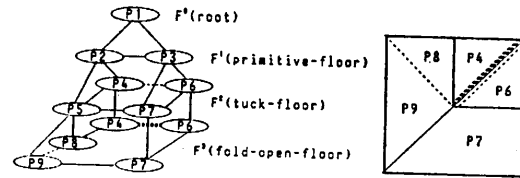


図 9 折り畳み開きフロアと展開図

Fig. 9 A fold-open-floor and crease pattern.

3.3 フロアの性質

3.2 節で定義した各々の動作から生成されるフロアの性質は以下となる。

(a) 折り畳みフロアの性質

1.  $r^k(v_1, v_2)$  の M-V 属性と O-F 属性は、節点  $v_1, v_2$  のそれぞれの親節点間に存在する  $r^{k-1}(v_{11}, v_{21})$  の M-V 属性と O-F 属性をそのまま継承する。
2.  $rel(r^k(v_1, v_2))=cou$  に隣合う左右の  $r^k$  が bro 関係であり、これらの  $r^k$  の O-F 属性がすべて fold であれば、その二つの bro 関係の  $r^k$  の M-V 属性は排反である。
3. bro 関係と cou 関係が交互に現れている四つの  $r^k$  から構成される閉路で、これらの  $r^k$  の O-F 属性がすべて fold であれば、その向かい合う二つの cou 関係の  $r^k$  の M-V 属性は同値である。

(b) 折り込みフロアの性質

1.  $r^k(v_1, v_2)$  の O-F 属性は、節点  $v_1, v_2$  のそれぞれの親節点間に存在する  $r^{k-1}(v_{11}, v_{21})$  の O-F 属性をそのまま継承する。
2.  $rel(r^k(v_1, v_2))=cou$  に隣合う左右の  $r^k$  が bro 関係であり、これらの  $r^k$  の O-F 属性がすべて fold であれば、その二つの bro 関係の  $r^k$  の M-V 属性は同値である。
3. bro 関係と cou 関係が交互に現れている四つの  $r^k$  から構成される閉路で、これらの  $r^k$  の O-F 属性がすべて fold であれば、その向かい合う二つの cou 関係の  $r^k$  の M-V 属性は排反である。

(c) 開きフロアの性質

1.  $r^k(v_1, v_2)$  の M-V 属性は、節点  $v_1, v_2$  のそれぞれの親節点間に存在する  $r^{k-1}(v_{11}, v_{21})$  の M-V 属性をそのまま継承する。

- 2. O-F 属性が fold の  $r^k$  に隣合う左右の  $r^k$  が open 弦であれば、その二つの open 弦の M-V 属性は排反である。
  - 3. O-F 属性が fold の  $r^k$  と open 弦が交互に現れている四つの  $r^k$  から構成される閉路であれば、その向かい合う二つの O-F 属性が fold の  $r^k$  の M-V 属性は同値である。
  - 4. 開きフロアに存在する  $r^k$  はすべて cou 関係である。
- (d) 折り畳み込み, 折り畳み開きフロアの性質
- 1. 基本動作の複合である折り畳み込み, 折り畳み開きによるフロアに存在する  $r^k$  の性質は, それぞれの基本動作の性質を引き継ぐ。

4. 展開図からタワーの構築

ここでは, 推論システムと真理性維持システム<sup>3)</sup> (TMS: Truth Maintenance System) を用いて展開図からタワー構築する方法について述べる。

4.1 タワー構築方法

折り紙の展開図を構成する原子折り目と原子平面の情報からタワーを構築するには, 折り紙の変形過程を推論すればよい。なお, ここでは折り紙の変形過程を, 最後に折られるものから順次推論していく方法をとる。

タワーの構築過程には, その折り目と平面の組合せに多くの選択候補があるため, 試行錯誤が必要である。この場合, 展開図を構成する原子平面または原子折り目について仮説を立て, 仮説に基づいた信念(知識)を用いて推論していく。推論過程で, 信念の状態がタワー構築の制約条件に矛盾すると判明した場合には, その仮説と仮説から導き出された信念を棄却破棄する。

ここでは, このような信念の翻意を行うときに知識の無矛盾性を管理する真理性維持システムを用いる。仮説に基づいた推論を行うシステムを推論システムとし, これらのシステムの間関係を図 10 に示す。

4.2 タワー構築の制約条件

以上よりタワー構築時の制約条件を挙げる。

- 1. 3.1 節より折り紙の変形過程はつねに二次元平面の折り目による重ね合わせである。
- 2. 高さ  $n-1$  のタワーは  $n$  個のフロアから構成され, これら  $n$  個のフロアは 3.3 節で述べて各フロアの性質のいずれかを満たす。
- 3. 二分木の性質より任意の節点は親節点を二つ以上持たない。
- 4. 折り紙の展開図を構成する原子折り目または原子平面は過不足なくタワーに存在する。
- 5. 2 節で述べた  $R_1^k$  と  $R_2^k$  の性質より深さ  $k$  の兄弟関係の弦は深さ  $k$  のフロアのみが存在する。

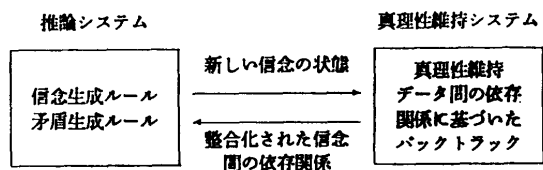


図 10 推論システムと真理性維持システム  
Fig. 10 Inference system and truth maintenance system.

- 4.3 推論システム
- 推論システムは信念生成ルールと矛盾生成ルールを保持する。また, 推論システムは, 信念生成ルールまたは矛盾生成ルールの条件部を構成する条件がすべて満たされた場合, 結論部の知識を TMS に理由付けとして与える。
- ここで, 2.1 節で定義した, 平面, 折り目, 兄弟関係, 従兄関係, フロアをそれぞれ以下の述語を用いて記述する。また, 信念生成ルールと矛盾生成ルールを以下の書式で表す。

4.3 推論システム

推論システムは信念生成ルールと矛盾生成ルールを保持する。また, 推論システムは, 信念生成ルールまたは矛盾生成ルールの条件部を構成する条件がすべて満たされた場合, 結論部の知識を TMS に理由付けとして与える。

ここで, 2.1 節で定義した, 平面, 折り目, 兄弟関係, 従兄関係, フロアをそれぞれ以下の述語を用いて記述する。また, 信念生成ルールと矛盾生成ルールを以下の書式で表す。

平面: plane (平面名, 裏表, ver ([頂点 1, ..., 頂点  $n$ ]), children (子平面名))。

折り目: fold (seg ([頂点 1, 頂点 2]), attribute (折り目の属性))。

兄弟関係: brother (planes ([平面名, 平面名]), seg ([頂点 1, 頂点 2]), attribute (折り目の属性))。

従兄関係: cousin (planes ([平面名, 平面名]), seg

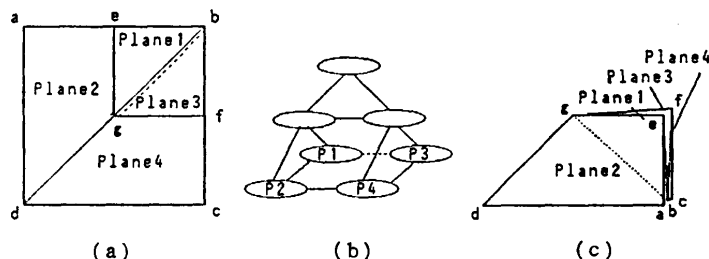


図 11 (a) 展開図, (b) タワー, (c) 出来上がり図  
Fig. 11 (a) Crease pattern, (b) Tower, (c) Resulting shape.

([頂点1, 頂点2], attribute (折り目の属性)).

裏表: {bottom, top}

頂点: 平面のかど

子平面名: {[ ], [平面名], [平面名, 平面名]}

折り目の属性: {[mountain, open], [mountain, fold], [valley, open], [valley, fold]}

rule (ルール名, [条件部], アクション部).

ルール名: ルール番号.

条件部: 述語, n (述語), m (述語)

述語: 述語が in である場合成功しその理由付けは「ground」とする.

n (述語): 述語が in である場合成功しその理由付けはなしとする.

m (述語): not (述語) が out である場合成功しその理由付けは「assumption」とする.

アクション部: 述語, not (述語)

述語, not (述語) を新しい信念として追加する.

#### 4.4 TMS

TMS は推論システムから与えられた理由付けを記録し, これらの理由づけから信念の状態を決定し推論システムにそれを返す.

なお, TMS では述語を以下のノード形式で表す.

node (ノード番号, 事実, 理由づけ, 真偽).

ノード番号: 自然数

事実: 述語, not (述語)

理由づけ

「前提」: 'premise'

「仮定」:

[assumption, インノード, アウトノード]

インノード: ノード番号を要素とするリスト

アウトノード: ノード番号を要素とするリスト

「根拠」: [ground, インノード]

インノード: ノード番号を要素とするリスト

「cp 型証明」: [cp, 結論部, 前提部]

結論部: 矛盾ノードのノード番号

前提部: ノード番号を要素とするリスト

真偽: {in, out}

ここで, 事実と真偽の組み合わせによって以下の四つの状態が存在する.

(述語, in): 「述語が表している信念を信じている」

状態

(述語, out): 「述語が表している信念を信じてない」状態

(not (述語), in): 「述語が表している信念の否定を信じている」状態

(not (述語), out): 「述語が表している信念の否定を信じてない」状態

#### 4.5 実行例

図 11(a)の展開図を構成している原子折り目と原子平面から図 11(b)のタワーを構築する実行過程を以下に図示する (図 12).

#### 5. タワーから出来上がり図の決定アルゴリズム

タワーから折り目の折られた順序と平面の裏表の変化過程を決定した出来上がり図を作成するアルゴリズムについて述べる.

このアルゴリズムはタワーの上から逐次下に向かって手続き「フロアチェック」により, 3.3 節で与えたフロアの性質を照合することよりフロアの種別を決定する. 次に以下に示す各々のフロアについての処理を行う. なお, これらの各処理では平面の裏表を決定するためにタワーが持つ節点を以下のように定義する.

fre 節点: 手続き「ホールドケース」, 「タックケース」, 「オープンケース」の初期値としてその時点に与えられた平面

inf 節点: 動作によって裏表が変わる平面

ind 節点: 動作によって裏表が不変の平面

また, 手続き「ホールドケース」, 「タックケース」, 「オープンケース」は与えられた fre 節点から折り目の折られた順序と inf 節点と ind 節点を決定する.

BEGIN

$k=0$

タワーが持つすべての節点の裏表属性を裏とする.

WHILE ( $k \leq n$ ) DO

$k=k+1$

フロアチェック ( $F^k$ )

WHILE ( $F^k$  のすべての節点に対して) DO

その節点を fre 節点とする.

END

CASE  $F^k$  OF

primitive-floor: ホールドケース

fold-floor: ホールドケース

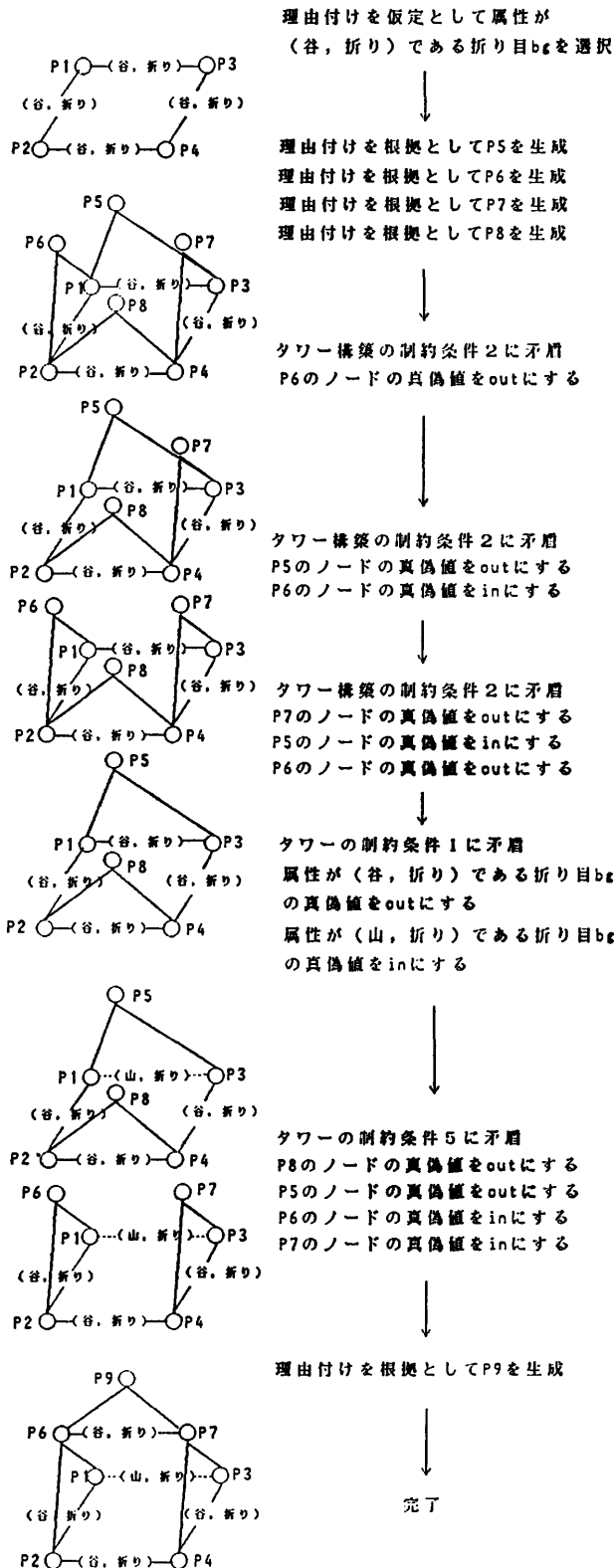


図 12 タワー構築の実行過程

Fig. 12 Process of executing the tower construction.

```

tuck-floor or fold-tuck-floor: タック
    ケース
open-floor or fold-open-floor: オープン
    ケース
END
inf 節点の子孫の節点の裏表属性を反転する。
END
END
procedure ホールドケース
    BEGIN
        任意の一つの fre 節点を inf 節点に変える。
        WHILE (すべての fre 節点に対して) DO
            IF (fre 節点が inf 節点と兄弟関係の弦で隣接)
                その fre 節点を ind 節点に変える。
            IF (fre 節点が ind 節点と兄弟関係の弦で隣接)
                その fre 節点を inf 節点に変える。
            IF (fre 節点が inf 節点と従兄関係の弦で隣接)
                その fre 節点を inf 節点に変える。
            IF (fre 節点が ind 節点と従兄関係の弦で隣接)
                その fre 節点を ind 節点に変える。
        END
        Fk に存在する兄弟関係の弦の順序を k とする。
    END
procedure タックケース
    BEGIN
        tuck 弦に接続している fre 節点を inf 節点とする
        WHILE (すべての fre 節点に対して) DO
            IF (fre 節点が inf 節点と兄弟関係の弦で隣接)
                その fre 節点を ind 節点に変える。
            IF (fre 節点が ind 節点と兄弟関係の弦で隣接)
                その fre 節点を inf 節点に変える。
            IF (fre 節点が inf 節点と従兄関係の弦で隣接)
                その fre 節点を inf 節点に変える。
            IF (fre 節点が ind 節点と従兄関係の弦で隣接)
                その fre 節点を ind 節点に変える。
        END
    END
    
```

隣接)

その fre 節点を ind 節点に変える。

END

$F^*$  に存在する兄弟関係の弦または折り込み弦の順序を  $k$  とする。

END

procedure オープンケース

BEGIN

任意の一つの fre 節点を inf 節点に変える。

WHILE (すべての fre 節点に対して) DO

IF (fre 節点が inf 節点と兄弟関係の弦または開き弦で隣接)

その fre 節点を ind 節点に変える。

IF (fre 節点が ind 節点と兄弟関係の弦または開き弦で隣接)

その fre 節点を inf 節点に変える。

IF (fre 節点が inf 節点と兄弟関係の弦または開き弦で隣接していない)

その fre 節点を inf 節点に変える。

IF (fre 節点が ind 節点と兄弟関係の弦または開き弦で隣接していない)

その fre 節点を ind 節点に変える。

END

$F^*$  に存在する兄弟関係の弦または開き弦の順序を  $k$  とする。

END

## 6. おわりに

本論文では折り紙の展開図を構成する幾何学的要素から、出来上がり図を推論する一方法を提案した。主な結果としては、折り目による図形の変形過程を表現するタワーを提案し、その生成方法と性質を明らかにした。また展開図を構成する幾何学的要素から TMS を用いてタワー構築できることを示した。さらにそのためのタワー構築の制約条件を示した。最後に、タワーを構成するフロアの性質を利用して出来上がり図を決定するアルゴリズムを示した。

なお、ここで示した手法は知的 CAI<sup>4)</sup> を目的とした教授戦略や知的インタフェースの構築などに発展させ得る可能性が期待できよう。

## 参考文献

- 1) 笠原邦彦：おりがみ新世紀，サンリオ (1989)。
- 2) 内田 忠，伊藤英則：折り紙の知識表現とその処理プログラムの作成，第 42 回情報処理学会全国大会論文集，pp. 1-53, 1-54 (1991)。
- 3) Doyle, J.: A Truth Maintenance System, *Artif. Intell.*, Vol. 12, pp. 231-272 (1979)。
- 4) 岡本敏雄，松田 昇：知的 CAI における幾何の証明計画と認識と学習機能について，情報処理学会論文誌，Vol. 30, No. 8, pp. 1046-1057 (1989)。

## 付 録

【付録 1】 鶴の展開図と出来上がり図を示す (図 13)。

【付録 2】 鶴のタワーの一部 (深さ 18 の最終フロア) を示す (図 14)。

(平成 3 年 3 月 25 日受付)

(平成 3 年 10 月 3 日採録)



内田 忠

1967 年生。1989 年名古屋工業大学電気情報工学科卒業。1991 年名古屋工業大学工学研究科博士前期課程電気情報工学専攻修了。同年中部電力(株)入社。論理プログラミング、知識獲得等に興味を持つ。



伊藤 英則 (正会員)

1946 年生。1969 年福井大学電気工学科卒業。1974 年名古屋大学大学院工学研究科博士課程満了。工学博士。同年 4 月より日本電信電話公社横須賀電気通信研究所勤務。1985 年より(財)新世代コンピュータ技術開発機構に研究室長として出向。1989 年 7 月より名古屋工業大学工学部教授。現在知能情報システム学科。オートマトンと言語理論、コンピュータネットワーク通信管理、知識ベースとアルゴリズム、知識表現言語、ファジィ論理制御等の研究に従事。電子情報通信学会，人工知能学会，日本ファジィ学会各会員。

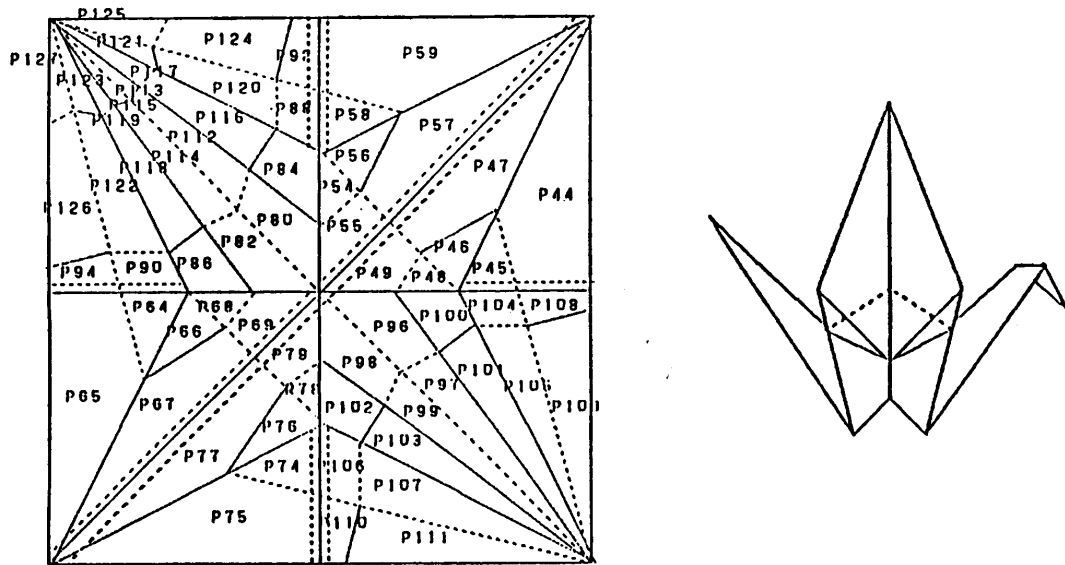


図 13 鶴の展開図と出来上がり図  
 Fig. 13 A crane's crease pattern and resulting shape.

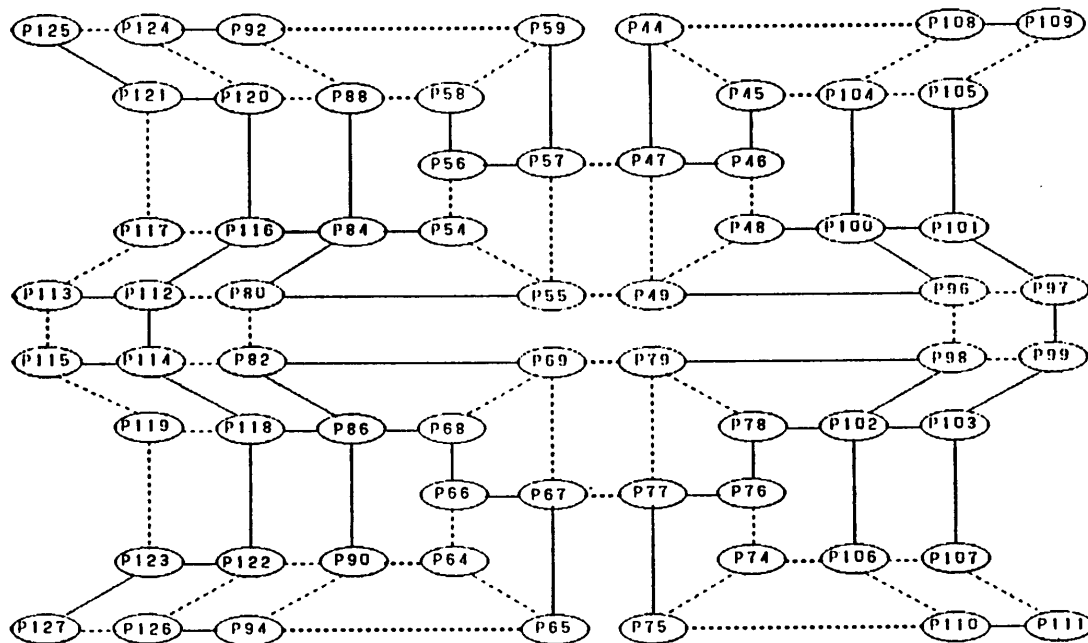


図 14 鶴のタワーの深さ 18 の最終フロア  
 Fig. 14 The 18<sup>th</sup> (bottom) floor of tower for a crane.