

# 図面記号の学習理解

河越正弘・喜多伸之  
(電子技術総合研究所)

## 1. 序 論

図面認識については、近年数多くの研究がなされてきたが、これらの多くは、描画時のうまい制約のつけ方を考えたり、あらかじめ使用されるすべての図面記号の構造を調べて分類ルールを求めたりという、実用的なアプローチを採るものであった。

しかし、計算機に図面を人間と同じように柔軟に扱わせることを目指す、知的インタフェースとしてのアプローチにおいては、人間とのやりとりを通じて、記号を理解したり、特徴と特徴として扱うよう適応したりする能力が必要となる。

そこで、我々は図面を対象とした学習を行なう第一歩として、図面記号の学習について考察し、種々の問題点を洗い出すと共に、それらを解決するた

めのアプローチ及び、実現するためのシステムの枠組み(図1)について既に提案した。

このシステムのデータ・制御の流れを図示すると図2のようになるが、本論では、図中の破線で囲ったデータ記述形式及びElastic Matchingの手法について述べる。

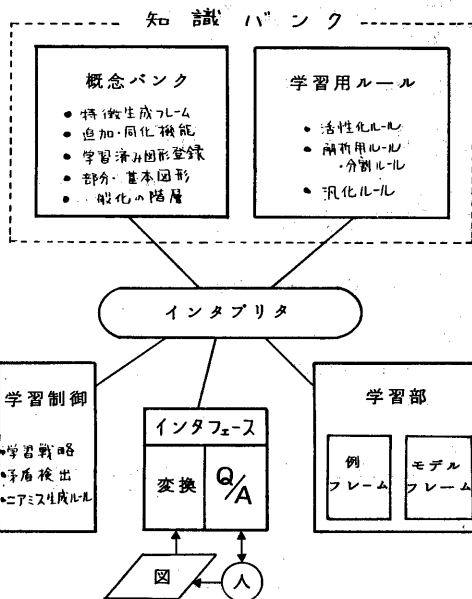


図1 システム構成図

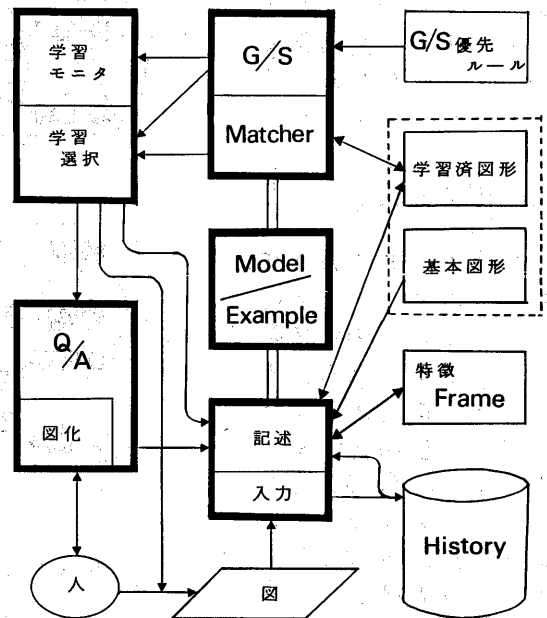


図2 データと制御のフロー

入力された図は、特徴フレームや基本図形フレームを駆動することにより、Exampleとして示される。これと、モデルとをElasticにMatchingさせることにより、一般化/特殊化の争掛りを得ると共に、学習モニタを通じて学習の収束の経度に関する情報を学習選択部に与える。これに基づき、アドバイスや例を人間に要求し学習を進める。

## 2. 図面記号の記述形式

図面を表現するために、リスト、構造関係表、リンク構造、グラフなどがよく用いられる。しかし、これらは記述範囲を柔軟に拡張することができないため、予め設定した範囲外の対象には対応できず、図面学習の立場からは不十分である。そこで、図面をフレームで表現すれば、個々の特徴もそれぞれ一つのフレームで記述されるため、特徴の追加による記述範囲の拡張が容易になり、さらには、図面の階層的な表現も可能となる。以下に、本システムで用いた図面記号のフレームによる記述形式について述べる。

### 2.1 例示記号入力と特徴抽出

画像データ（濃淡値の二次元アレイ）の形で与えられる例示記号から、まず、線分、円弧、円、接続点（交点と端点）を抽出する（図3(a)）。点線、ダット、ハッチングなどは含まないとし、曲線は円弧で近似抽出する。（現在、例示記号は画像データによる入力の替わりに、タブレットを用いてストロークにより入力している。）

将来は、画像データからの特徴抽出に、上位レベルからのフィードバックをかけて、柔軟な特徴抽出を行なえるようにする。

### 2.2 基本特徴フレームによる記述

人間が図面記号を表現するために用いる特徴は非常に多く、図4のように分類できる。この図に示されるように、学習済みの図面記号自身も、他の記号を記述する特徴となりうる。これらの特徴は生成フレーム（図5）の形で知識ベースに蓄えておく。さらに、学習の過程で新たに必要となった特徴や、学習済みの図面記号も、適時、知識で

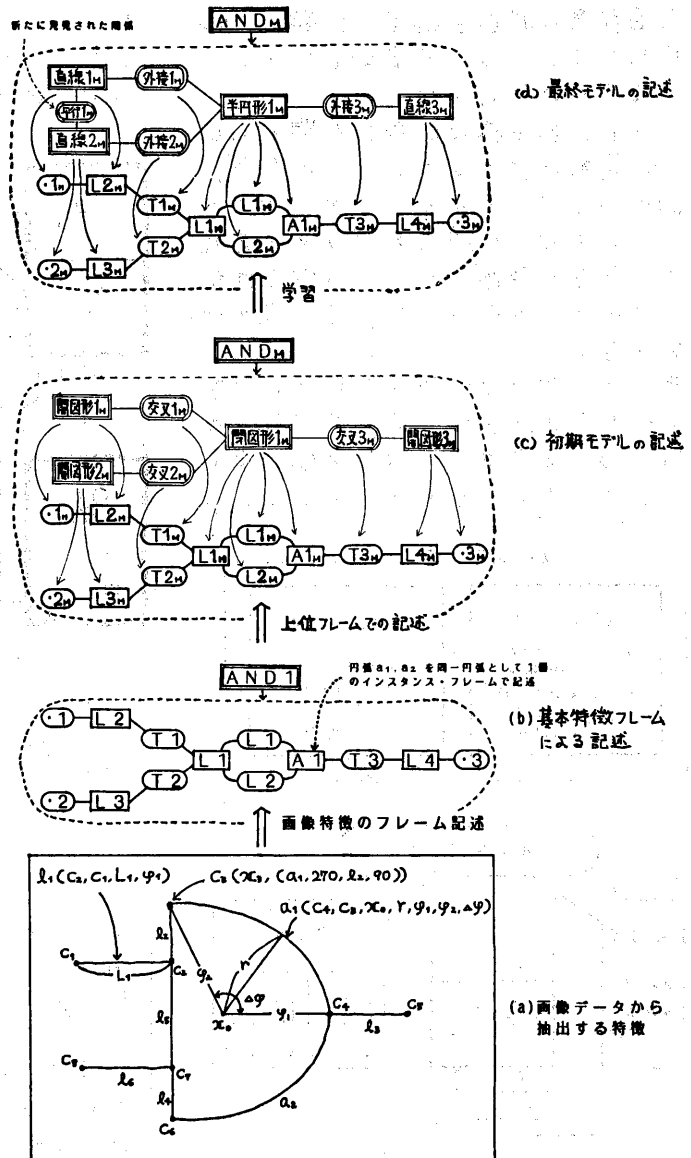


図3 記述形式の例

ースに追加できるものとする。

例示記号から、これらすべての特徴の抽出を試みることは非現実的なので、最初は基本特徴だけを抽出する。基本特徴とは局所的な特徴で、計算機にとって比較的 low コストで抽出できる特徴である。つまり、画像データから抽出した特徴を、図4の基本特徴フレーム(基本要素フレームと接続点フレーム)だけで記述する。

基本要素フレームは、直線フレーム、円弧フレーム、円フレームであり、それぞれ画像データから抽出した線分、円弧、円に対応するが、線分あるいは円弧については、接続点における他の線分あるいは円弧との接続可能性を調べ、接続可能なものについては同一のフレームで記述する(図6)。

接続点フレームは画像データから抽出した接続点を記述するが、端点と、よく現われる4種の接続型は、・型、一型、L型、T型、十型と、特殊化しておく(図7)。

例示記号の基本特徴フレームによる記述の例を図3(b)に示す。

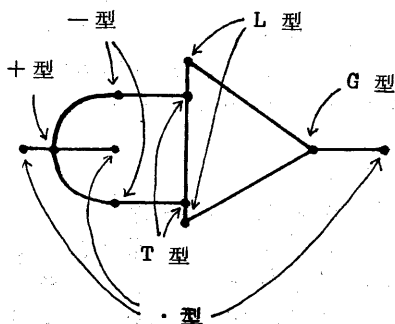


図7 接続点の型

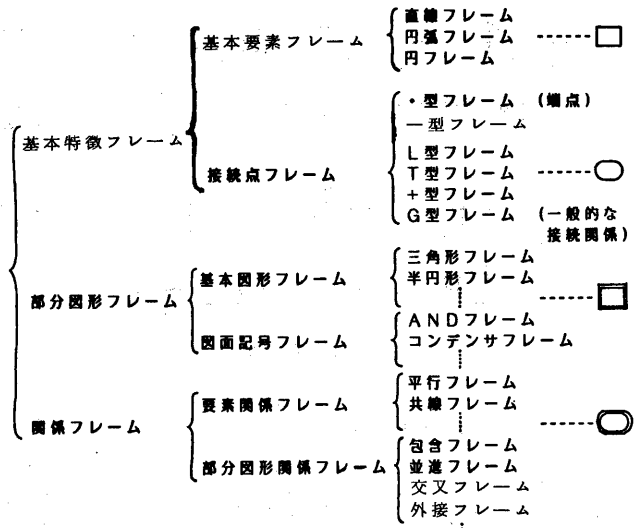


図4 特徴フレームの分類

```
< LINE
  (SELF (VALUE (ELEMENT)))
  (CONTENT (VALUE) (IF-ADDED ($CALC-LINE)))
  (ELEMENT-OF (VALUE))
  (I-RELATION (VALUE))
  (INCLINATION (VALUE) (DEFAULT (0.)))
  (LENGTH (VALUE) (DEFAULT (0.)))
  (NODE-1 (VALUE) (IF-ADDED ($CALC-NODE)))
  (NODE-2 (VALUE) (IF-ADDED ($CALC-NODE))) >
```

```
< T-CONNECT
  (SELF (VALUE (CONNECT)))
  (ANGLE (VALUE))
  (CONTENT (VALUE))
  (N-ELEMENT (VALUE))
  (POSITION (VALUE))
  (RELATION-OF (VALUE))
  (T-ELEMENT (VALUE)) >
```

図5 生成フレームの例

```
< LINE1
  (SELF (VALUE (LINE)))
  (CONTENT (VALUE (L3) (L6) (L4)))
  (ELEMENT-OF (VALUE (AND1)))
  (I-RELATION (VALUE (T-CONNECT0) (T-CONNECT1)))
  (INCLINATION (VALUE (90.0)))
  (LENGTH (VALUE (60.0)))
  (NODE-1 (VALUE (L-CONNECT0)))
  (NODE-2 (VALUE (L-CONNECT1))) >
```

```
< ARCO
  (SELF (VALUE (ARC)))
  (ANG-1 (VALUE (270.0)))
  (ANG-2 (VALUE (90.0)))
  (ANGLE (VALUE (180.0)))
  (CENTER (VALUE ((40.0 50.0)))
  (CONTENT (VALUE (A1) (A2)))
  (ELEMENT-OF (VALUE (AND1)))
  (I-RELATION (VALUE (T-CONNECT2)))
  (NODE-1 (VALUE (L-CONNECT1)))
  (NODE-2 (VALUE (L-CONNECT0)))
  (RADIUS (VALUE (30.0))) >
```

```
< T-CONNECT0
  (SELF (VALUE (T-CONNECT)))
  (ANGLE (VALUE (90.0)))
  (CONTENT (VALUE (C2)))
  (N-ELEMENT (VALUE ((LINE1 90.0)))
  (POSITION (VALUE ((40.0 60.0)))
  (RELATION-OF (VALUE (AND1)))
  (T-ELEMENT (VALUE ((LINED 180.0))) >
```

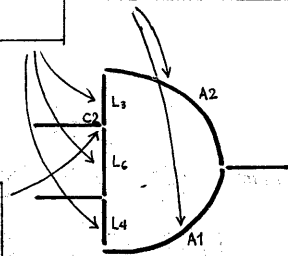


図6 フレームによるデータ記述

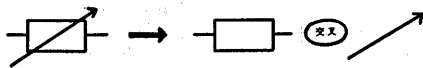


図 8 本質的な特徴による記述例

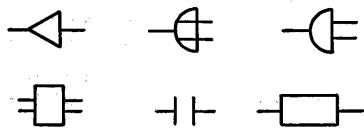


図 9 図面記号の例

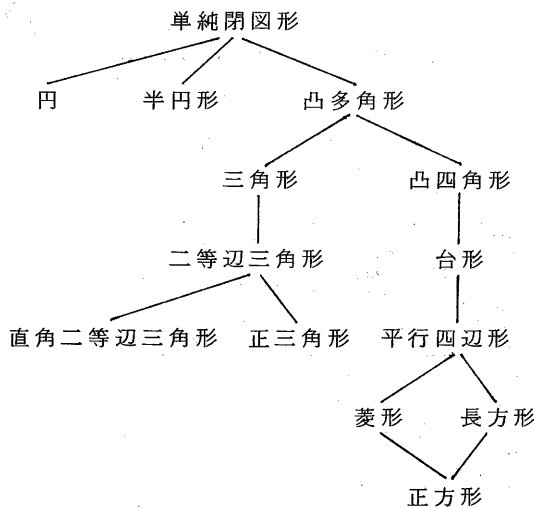


図 10 単純閉図形の束構造

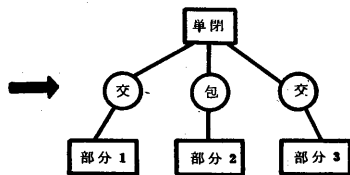
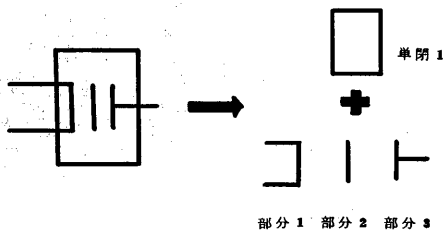


図 11 初期モデルの記述例

## 2.3 図面記号モデルの記述

我々は、記号のモデルを人間がその記号を作ったときに意図した本質的な特徴により記述することを目指す。例えば、図 8 に示すように、可変抵抗記号は、抵抗記号と矢印が交叉したものとして記述したい。このためには、なるべく大きな部分図形の大きな関係で記述することが目安となる。

### ・初期モデルの記述

最初に与えられる例示記号から、学習効率を上げるための手掛りとして単純閉図形(図 10)を抽出したものを初期モデルとする。ここで単純閉図形を手掛りとして選んだ理由は、

- 1) 既知の部分図形である。
- 2) 多くの記号に含まれている(図 9)。
- 3) 抽出アルゴリズムがある(付録 1)。
- 4) 束構造を持つ(図 10)。

などである。

よって初期モデルを作る手順は簡単には次の様である。

1. 基本特徴ツリーによる記述から単純閉図形を抽出する。
2. 残った部分の連結要素をまとめてそれぞれを 1 つの部分ツリーで記述する。
3. 単純閉図形と部分ツリーとの間の接続関係の有無及び包含関係を調べる。

この手順に従って初期モデルを作成した例と図 11 に示す。ただし、単純閉図形がなかったときは、基本特徴ツリーによる記述を初期モデルとする。

### ・モデルの記述

与えられる例示記号と現行モデルとの対応付け結果をもとに、現行モデル記述に一般化/特殊化、あるいは新たな特徴の追加などの修正を加え、最終的には人間の表現に近いモデルが得られる。例は図 3 (d) である。

### 3. 対応付け手法

本システムで必要とされる対応付けには、単に定量的な変化に適応できる能力だけでなく、定性的変化にも適応可能な能力が必要である。しかし、学習過程においては記号図形を同じ向きに提示するという制限を加えることができるので、弛緩整合法を拡張することによりこれを構成した。

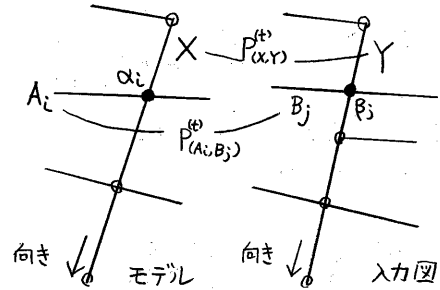


図 12 線分対応の整合性の求め方

#### 3.1 対応付けアルゴリズム

最初に、線分だけを含む場合を対象にして、アルゴリズムの基本部分について説明し、順次、円弧・円、閉図形を含む場合について拡張する。

##### (1) 基本アルゴリズム (線分のみの場合)

・初期対応 例示図形はモデルと大きく角度は変わらないので、モデル線分は自分となる角度が  $\theta_c$  以内のすべての入力線分に初期対応させ、それらの初期確率は同じとする。

・弛緩過程 初期対応をもとに、それぞれの対応付けのもっともらしさを、周りの線分との接続の仕方がどの程度つじつまが合っているかに基づき更新していく。例えば図 12 において、モデル線分  $X$  と例示図形の線分  $Y$  の組を考えると、 $X, Y$  上の接続点列  $\{\alpha_i\}, \{\beta_j\}$  について、まず両者の対応  $(A)$  をとり、それに基づき接続点の組を調べて行き、両方の接続の仕方、及び接続されている線分同士の間対応確率をもとに整合度を求める。さらに  $\sigma$  を変化させて整合度を最大にする  $\sigma_{max}$  を求め、このときの整合度を  $X$  と  $Y$  の自己整合度  $Q_{(x, y)}$  とし、そのときの接続点の対応から隣接整合度  $Q_N$  を求める。こうして求めた  $Q_{(x, y)}, Q_N$  をもとに式 (2) に従って  $X$  と  $Y$  の対応確率を更新する。

以上の手順を繰り返すことにより、モデル線分と例示図形の線分との最適な対応付けが求まる。

### 用語説明

- $P_{(x, y)}^{(t)}$ :  $t$  回繰り返し後の  $X$  と  $Y$  が対応付けられる確率  
 $P_{(A_i, B_j)}^{(t)}$ :  $X, Y$  に接続している  $A_i, B_j$  が対応付けられる確率  
 $\alpha_i, \beta_j$ :  $X, Y$  上の接続点  
 $C(X)$ :  $X$  と対応可能な  $Y$  の集合  
 $Q_{(x, y)}^{(t)}$ :  $X$  と  $Y$  の整合度  $(= Q_{S(x, y)}^{(t)} + \sum Q_{N(x, y)}^{(t)})$   
 $Q_{S(x, y)}^{(t)}$ : 自己整合度。  $A_i, B_j$  の最良対応が  $X, Y$  の対応によるものとする。  
 $Q_N^{(t)}(A_i, B_j)$ : 隣接整合度。  $X$  と  $Y$  に対して  $A_i, B_j$  が最良対応であることの  $A_i, B_j$  の対応によるものとする。  
 $\sigma(\alpha_i, \beta_j)$ : 接続点  $\alpha_i, \beta_j$  の適合係数 (計算法は付録 II 参照)  
 $\sigma(\{\alpha_i\}, \{\beta_j\})$ : 接続点の順序を変化させて、 $\alpha_i$  と  $\beta_j$  を対応させる組み合わせ。  $\sigma=0$  を採用したときの、 $k$  番目の  $\alpha, \beta$  は  $\alpha_i(k, \sigma), \beta_j(k, \sigma)$  と書ける。  
 $P_x^{(t)} \triangleq \{P_{(x, y_1)}^{(t)}, \dots, P_{(x, y_j)}^{(t)}, \dots, P_{(x, \phi)}^{(t)}\}$   
↑  
対応するそのナシの確率

### 確率更新アルゴリズム

$$\begin{cases} Q_{S(x, y)}^{(t)} = \max_{\alpha} \sum_k \sigma(\alpha_i(k, \sigma), \beta_j(k, \sigma)) P_{(A_i, B_j)}^{(t)} \\ Q_N^{(t)}(A_i, B_j) = \sigma(\alpha_i(k, \sigma_{max}), \beta_j(k, \sigma_{max})) P_{(A_i, B_j)}^{(t)} + Q_N^{(t)}(A_i, B_j), \quad k=1, 2, (\sigma_{max}) \end{cases} \quad \text{--- ①}$$

$$\begin{cases} Q_{(x, y)}^{(t+1)} = Q_{S(x, y)}^{(t)} + Q_N^{(t)}(x, y) \\ P_{(x, y)}^{(t+1)} = \frac{Q_{(x, y)}^{(t+1)} \cdot P_{(x, y)}^{(t)}}{\sum_{Y \in C(X)} Q_{(x, Y)}^{(t+1)} P_{(x, Y)}^{(t)}} \end{cases} \quad \text{--- ②}$$

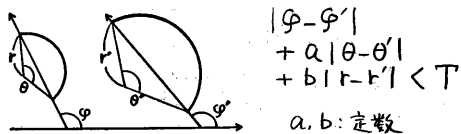


図 13 円弧の対応候補の求め方

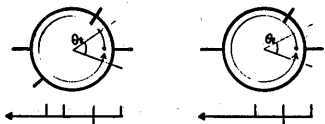


図 14 円の始点の決定法

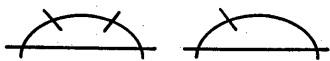


図 15 同一要素の接続点の対応づけ

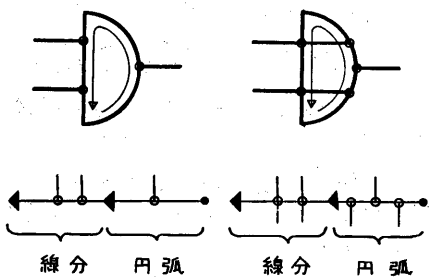


図 16 閉図形を利用した対応づけ

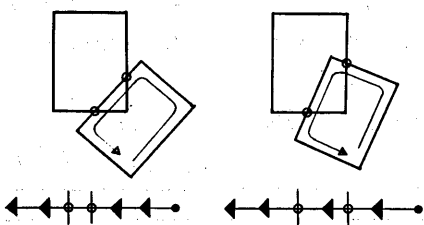


図 17 関係の柔軟化の例

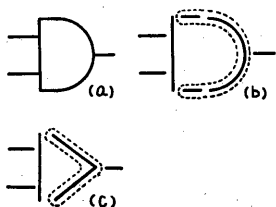


図 18 図形要素の柔軟な対応

## (2) 円弧を含む場合への拡張

- ・円弧の初期対応の相手は図 13 の条件を満たす円弧とする。
- ・円の場合の始点の決め方は図 14 に示す通り、円周を  $\theta_t$  の中で走査し、その中に共通の接続点があったときそこを始点とする。その始点をもとに線分の場合と同様に接続点の対応を付ける。
- ・図 15 のように、2ヶ所まで円弧と、あるいは線分で交わっている時は、該当する接続点を優先的に対応させる。

## (3) 閉図形を利用する場合への拡張

- ・両者に同一の閉図形がある場合、閉図形で記述されたレベルでまず対応付けを行う。
- ・閉図形の辺同士の間隔を付けて、図 16 のように接続点の対応を行なう。

## 3.2 アルゴリズムの柔軟化

今まで述べたアルゴリズムは通常の意味での *Elastic Matching* であるが、関係/要素の一般化/特殊化を組み込むことにより、さらに柔軟に対応付けをすることができる。

### (1) 関係についての柔軟化

図 17 に示すように、閉図形と閉図形の重なりの場合、つまり 2 つの接続点での接続線分が同一図形に属している場合、頂点あるいは、他の線分を越えていても対応可能とみなせるようにする。

### (2) 要素についての柔軟化

例えば図 18 の a) 図と b) 図の対応付けを行なった時、半円部を U 字型に対応付けられるように柔軟化する。この柔軟化により a) 図と c) 図の場合にも同様に、半円と V 字型が対応付けられる。

そのために、閉図形の対応のつかない部分を一つの図形要素としてまとめておく。あるいは、対応のよい部分を全体から取り除いた部分を同様にまとめておく。

#### 4. 対応結果に基づくモデルの更新

前章の対応付け結果に基づいて、以下のようルールに依り関係/要素を、一般化/特殊化してゆく。これにより最初単純な記述であった上位レベルに、人間が持っているような記号に対する記述が構成されてゆく。

このための、適用すべき一般化/特殊化ルールの決め方は、大まかな優先順位と、対応結果とルールの条件部の適合度により、各ルールに動的に与えられる評点に基づく Agenda 型の制御による。

以下に、一般化/特殊化の種類を示し、さらに、これらの一般化/特殊化を行なって、AND 記号の正しい記述をモデルとして構成してゆく過程を図19に示す。

##### ・関係についての一般化/特殊化

△ 同一関係が  $n$  個  $\rightarrow$  任意個の関係に一般化する

▽ 接続個数・種類に関する特殊化

▽ 接続位置・方向に関する特殊化

etc.

##### ・要素についての一般化/特殊化

▽ 直角、辺比などの属性値の特殊化—基本図形フレームに書いておく。

△ 図形の束構造(図10)に基づき一般化 etc.

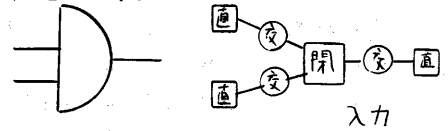
望ましい記述を得るためには、一般化/特殊化だけでは不十分であり、新しい特徴を導入したり、要素や関係をマクロ化する必要がある。新しい特徴の導入は、知識ベースに蓄えられている生成フレームを必要に応じて活性化することにより対応できる。マクロ化はよく出現する複数の図形の組を、まとめて1つとして扱うことで、記述の見通しをよくする。

#### 5. まとめ

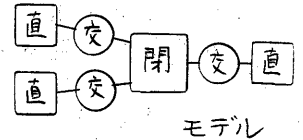
本論では、柔軟な図面記号認識のための、記述形式及び対応付け手法について述べた。

これらは、図2に示されるように、本システムの中核を成す部分であり、この部分

##### 1 典型例入力

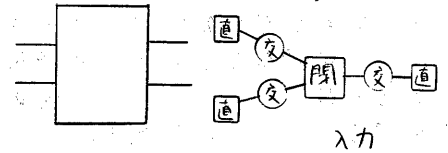


ルール: 最初はそのままだモデルに

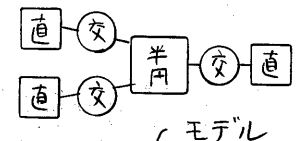


（同じに見える）

##### 2 negative 例

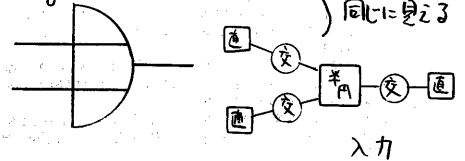


ルール: 閉図形を特殊化

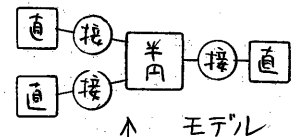


（同じに見える）

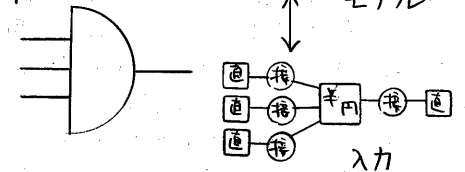
##### 3 negative 例



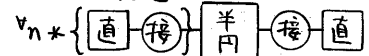
ルール: 閉図形との関係を特殊化



##### 4 positive 例



ルール: 同一関係が同一辺にあれば個数を一般化



モデル

図19 モデル更新過程の例

に人間の感覚をどう入れられるかによって、計算機との図面を介したインタフェースの良し悪しが大きく影響される。その意味で、本システムは、人間が図を見る場合に似た方式を目指したものである。

今後は、これを用いたシステムにおいて、本手法の有効性を確かめると共に、人間とのQ/Aを利用した、より人間のレベルに近い学習を実現することを考えている。

〔謝辞〕 末筆ながら、本研究のために有益な御協力を頂いた情報システム研究室青山技官、弛緩整合法について御助言頂いた画像処理研究室山本博士、日頃御指導頂く佐藤情報システム研究室長、及び本研究の機会を与えられた棟上ソフトウェア部長に感謝致します。

#### 参考文献

1. 横上・佐藤, "図面の自動認識と理解", 情報処理 vol.24 No.9 1086/1094 (1983)
2. 河越・喜多・青山・佐藤, "学習機能を持つ図面理解システムの構想", 情報処理学会第28回全国大会 4 G-5 (1983)
3. 河越・喜多・青山・佐藤, "図面の学習と理解" 電子技術総合研究所イ報 vol.48 no.7 614/624 (1984)
4. 喜多・河越・青山・佐藤, "学習機能を持つ図面理解システムの知識表現", 情報処理学会第28回全国大会 4 G-6 (1983)
5. 山本和彦, "構造解析法による手書き文字認識に関する研究", 電子技術総合研究所研究報告 no.831 94/99 (1983)

#### 付録Ⅰ—単純閉図形抽出アルゴリズム

スタックを1つ用意する。

1. 画面の縁から直線をひき、初めて交わる基本要素を、右方向にその始点から追跡する。
2. 接続点を見つけたら、その接続点がスタックに積まれていないか調べ、積まれていないときは3へ、積まれているときは4へ進む。
3. ① その接続点がT型で、たどってきた道が縦棒だったら、\*印と接続点をスタックに積む。  
② その接続点が・型だったら、なにもしない。  
③ それ以外だったら、その接続点をスタックに積む。  
このあと5に進む。
4. ① その接続点が+型だったら、6へ進む。  
② \*印より深い所へ戻ったら、6へ進む。  
③ それ以外だったら、その接続点をスタックに積んだあと、戻ったところより浅い部分を単純閉図形の候補とし、スタックから取り除く。

このあと5に進む。

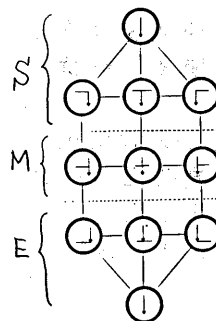
5. スタックの先端の接続点から基本要素の終点の場合、たどってきた道から左回りにみて初めてぶつかる道をたどり、それ以外の場合は、たどってきた道の延長をたどり、2から繰り返す。ただし、それが以前にたどった道であれば、スタックの先端を捨ててスタックが空になれば6へ進む、そうでなければ5を繰り返す。

6. 単純閉図形の候補から、凹図形を取り除いたものを単純閉図形とする。

#### 付録Ⅱ—接続点同志の適合係数の求め方

(適合係数)  $\triangleq$  (接続の種類)の適合度 A-1  
 $\times$  (角度の適合度) A-2  
 $\times$  (位置の適合度) A-3

A-1



$$\sigma_p = 1 - \frac{p}{3}$$

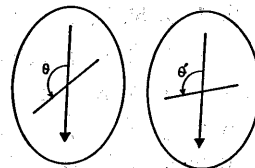
$p(x, y)$ :  $x, y$  の左図上での最短距離

$x \in S, y \in E$  なる

$x, y$  の組は存在しない

$$p(x, y) = 3$$

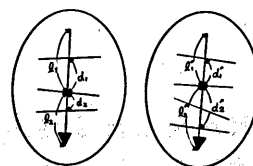
A-2



$$\sigma_\theta = \cos(\Delta\theta)$$

$$\Delta\theta = \theta - \theta'$$

A-3



$$\begin{aligned} \sigma_d &= (\text{自由端を含む点との差}) + (\text{内部接続点からの差}) \\ &= \sigma_f(d_1, d_2, d_3, d_4) + \sigma_i(d_1', d_2', d_3', d_4') \end{aligned}$$

$$\sigma = \sigma_p \cdot \sigma_\theta \cdot \sigma_d \text{ とする。}$$